Compressive Sensing Approaches for Sensor-based Predictive Analytics in Manufacturing and Service Systems

Kaveh Bastani

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree

of

Doctor of Philosophy
in
Industrial and Systems Engineering

Zhenyu (James) Kong, Chair
Jaime A. Camelio
Julianne Chung
Ran Jin

02/16/2016
Blacksburg, VA

Keywords: Compressive Sensing, Sparse Solution, Predictive Analytics, Sensor-based Quality Assurance

Compressive Sensing Approaches for Sensor-based Predictive Analytics in Manufacturing and Service Systems

Kaveh Bastani

ABSTRACT

Recent advancements in sensing technologies offer new opportunities for quality improvement and assurance in manufacturing and service systems. The sensor advances provide a vast amount of data, accommodating quality improvement decisions such as fault diagnosis (root cause analysis), and real-time process monitoring. These quality improvement decisions are typically made based on the predictive analysis of the sensor data, so called s*ensor-based predictive analytics*. Sensor-based predictive analytics encompasses a variety of statistical, machine learning, and data mining techniques to identify patterns between the sensor data and historical facts. Given these patterns, predictions are made about the quality state of the process, and corrective actions are taken accordingly.

Although the recent advances in sensing technologies have facilitated the quality improvement decisions, they typically result in high dimensional sensor data, making the use of sensor-based predictive analytics challenging due to their inherently intensive computation. This research begins in Chapter 1 by raising an interesting question, *whether all these sensor data are required for making effective quality improvement decisions, and if not, is there any way to systematically reduce the number of sensors without affecting the performance of the predictive analytics?* Chapter 2 attempts to address this question by reviewing the related research in the area of signal processing, namely, compressive sensing (CS), which is a novel sampling paradigm as opposed to the traditional sampling strategy following the Shannon Nyquist rate. By CS theory, a signal can be reconstructed from a reduced number of samples, hence, this motivates developing CS based approaches to facilitate predictive analytics using a reduced number of sensors. The proposed research methodology in this dissertation encompasses CS approaches developed to deliver the following two major contributions, (1) CS sensing to reduce the number of sensors while capturing the most relevant information, and (2) CS predictive analytics to conduct predictive analysis on the reduced number of sensor data.

The proposed methodology has a generic framework which can be utilized for numerous real-world applications. However, for the sake of brevity, the validity of the proposed methodology has been verified with real sensor data associated with multi-station assembly processes (Chapters 3 and 4), additive manufacturing (Chapter 5), and wearable sensing systems (Chapter 6). Chapter 7 summarizes the contribution of the research and expresses the potential future research directions with applications to big data analytics.

DEDICATION

My beloved uncle Mohammad Hossein passed away suddenly in the middle of writing this dissertation. His sudden death disclosed how tough it is living far away from family, especially once you hear terrible news like this, and you do not know how to withstand this situation. I miss you my dear uncle, and I dedicate this dissertation to you and all my family members, especially my dad Pirooz, mom Masi, sister Parastoo, and brother Arash. Love you all.

ACKNOWLEDGEMENTS

I gratefully acknowledge my advisor, Dr. Kong. Without his help, guidance and supervision this research could have not been completed in this quality. And to my other wonderful committee members Drs. Camelio, Chung, and Jin. Thank you for your time, considerations, and helps. I should also thank my former Ph.D. committee members back at the Department of Industrial Engineering and Management at Oklahoma State University (OSU), before I transferred to Virginia Tech. Dear Drs. Bukkapatnam, Balasundaram, and Sheng thank you very much for all your help and consideration.

I would like to thank my lab members at OSU, Drs. Asil Oztekin, Omer Beyca, and Mahmoud Mistarihi, as well as my lab members at Virginia Tech, Dr. Prahalad Rao, Jia Liu, Babak Barazandeh, Chen-ang Liu, and David Roberson for their friendship, help, and unconditional support.

Finally I wish to acknowledge my parents who showed me the importance of education, and supported me consistently throughout my life to reach my goals. I wish to acknowledge my sister Parastoo, brother Arash, and brother in-law Arash for their sincere love, kind supports, and being best friends ever. Finally, I acknowledge one of my best friends Ellie, for her genuine and meaningful support and friendship.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Recent advancements in sensing technologies offer new opportunities for quality improvement, and assurance in manufacturing and service systems. The sensor advances provide a vast amount of data accommodating quality improvement decisions such as fault diagnosis (root cause analysis), and real-time process monitoring. For example, in the realm of quality monitoring and control, given the streaming sensor data, it is desired to determine whether the process is under normal state or not. The importance of quality decisions is vivid as they are directly connected to the costs induced by defects or scraps. Hence, there is a crucial need to make successful and effective quality decisions, in order to reduce the costs and improve the quality of the product/service.

Depending on the nature of the process, the quality improvement decisions shall be made in two ways, namely, (1) offline decision making, and (2) real-time (online) decision making. For example, to do fault diagnosis in some industries, such as assembly processes, it might not be crucial to conduct decision making in a real-time manner [1]. This is because of the fact that detecting the process faults within an acceptable period of time is good enough for these industries, hence, offline decision making is satisfactory. However, in other industries, such as additive manufacturing where minute drifts in process conditions can drastically affect build quality, real-time process monitoring and decision making is required [2].

In general, the quality improvement decisions, whether in offline or online fashion, are made upon the predictive analysis of sensor data. The terminology used to describe this type of analysis is called s*ensor-based predictive analytics*. Sensor-based predictive analytics encompasses a variety of statistical, machine learning, and data mining techniques trying to find patterns between the sensor data and historical facts including (1) engineering knowledge, and (2) historical data. Given the patterns recognized using the above techniques, predictions are made about the quality state of the process, and corrective actions are taken accordingly. The relationship between the sensing, analytics, and processes is depicted in Figure 1-1. This figure illustrates a loop starting with processes, followed by sensing, then analytics, and back to the processes. From Figure 1-1, it can be observed that the sensors are utilized to collect process information and visualize them as sensor data, and then the predictive analytics is carried out on the sensor data to make quality improvement decisions about the process. This loop ensures the continuous quality improvement of the processes.

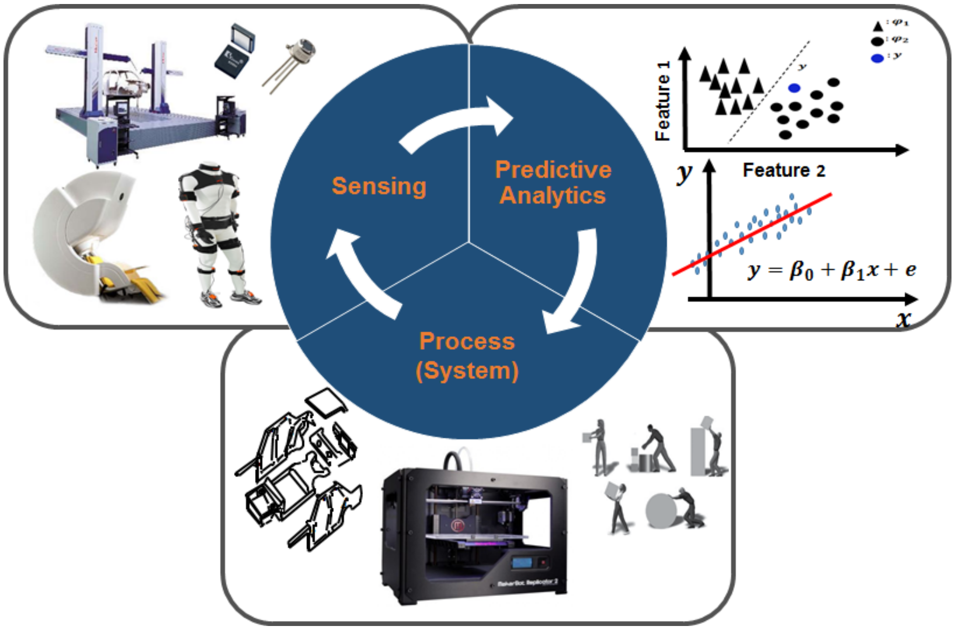

Figure 1-1. Diagram of sensor-based predictive analytics

Although the recent advances in sensing technologies have facilitated the quality improvement decisions, they typically result in high dimensional sensor data, making the use of sensor-based predictive analytics challenging due to their inherently intensive computation. The interesting question to ask is whether all these sensor data are required for making effective quality improvement decisions. In other words, is there any way to systematically reduce the high dimensionality of data by selecting the sensors providing the most representative information from the process. In this way, not only is the high dimensionality of the data reduced, but also the performance of predictive analytics remains intact, or it even might be improved due to analyzing more informative data.

However, the idea of reducing the number of sensors might contradict conventional thought, such that more sensors/data bring more opportunities to account for the complexity of the systems. Most complex engineering systems include a very large number of unknown process variables that may affect the quality state of the system, and unfortunately cannot be directly monitored or sensed. Using more sensors on some variables, which are in our control and can be measured directly, could possibly help gain the required information to estimate the unknown process variables. Consequently, the convention of using more sensors to address the complexity of the system has been founded upon this possibility. However, the validity of this convention was questioned and proven to be wrong in the past decade, due to the rise of compressive sensing (CS) theory in the field of signal processing [3].

CS is a novel sensing paradigm in signal processing to reconstruct signals from much smaller number of measurements. As opposed to the Shannon Nyquist rate, which requires the sampling rate to be at least twice of the bandwidth of the signal, CS proposes that a signal can be reconstructed from a much smaller number of measurements. Regardless of how complex the signal is, as long as it can have a sparse representation, CS theory proves that it can be reconstructed from a much smaller number of measurements than the dimension of the signal [3]. Sparse representation for the signal indicates that by using some specific dictionaries the signal can be transformed and represented as a vector with a very small number of nonzero elements, while most of its elements are zero.

As opposed to the convention mentioned earlier, by using CS theory it would be possible to account for the complexity of the systems using a smaller number of sensors. Although there are a large number of unknown process variables due to the complexity of the system, many of them might not be relevant or effective to the quality improvement decisions, thus they contain a sparse representation. Hence, as motivated from CS theory, estimating the large number of unknown process variables using a limited number of sensors would be possible. In other words, CS theory enables us to utilize a smaller number of sensors, minimize the sensor cost, reduce the high dimensionality of the sensor data, yet carry out the predictive analytics such that quality improvement decisions are made effectively.

This dissertation is mainly focused on CS theory, and proposes CS approaches for sensor-based predictive analytics in manufacturing and service systems. The contributions of the proposed methodology are twofold: (1) it introduces novel approaches to reduce the number of sensors while the most explanatory information are collected, and (2) it develops novel algorithms to estimate the unknown process variables while using a reduced number of sensors. In light of these contributions, sensor-based predictive analytics can be efficiently and effectively carried out. The CS based methodology proposed in this dissertation accommodates offline/online quality improvement decisions relating to a variety of real-world applications, ranging from manufacturing, such as multi-station assembly processes or additive manufacturing, to service systems, including body posture monitoring using wearable sensors.

The rest of the dissertation is organized as follows: in Chapter 2 the related research background on CS theory is presented, followed by the proposed methodology. The proposed methodology has a generic framework which can be utilized for numerous real-world applications. However, for the sake of brevity, the validity of the proposed methodology has been verified with real sensor data associated with multi-station assembly processes (Chapters 3 and 4), additive manufacturing (Chapter 5), and wearable sensing systems (Chapter 6). Chapter 7 summarizes the contribution of the research, and expresses the potential future research directions with applications to big data analytics.

Chapter 2. Compressive Sensing (CS) based Research Framework

The proposed research methodology in this dissertation is founded on CS. Therefore, in this chapter the related research in CS is reviewed in Sec. 2.1, and subsequently Sec 2.2 provides the proposed research methodology based on CS for sensor-based predictive analytics.

2.1 Related Research with Compressive Sensing (CS)

CS is an interdisciplinary area of optimization, statistics, and signal processing which has recently received a large amount of attention due to its advantage in reducing the number of samples for signal reconstruction. CS theory proposes a novel sampling paradigm as opposed to the Shannon Nyquist rate, based on which a signal can be reconstructed from a smaller number of samples. This is certainly important as in many real-world applications it can be very expensive to provide a large number of samples. CS theory encompasses two tasks: (1) *encoding* (forward model), and (2) *decoding* (inverse model). The former task relates to providing a reduced number of samples from the unknown signal, while the latter task is about reconstructing the signal from those samples.

CS encoding introduces the idea that a small number of samples would be enough only if the signal is naturally sparse or sparsable using an appropriate transformation. Indeed, the key idea that CS theory is based on is the sparsity assumption of the signal. Sparsity assumption indicates that either the signal itself or a transformation of the signal is sparse, which means that there are only a few elements containing most of the signal information. Let $\mathbf{x} \in R^n$ represent the signal where n represents the number of signal elements, and $\mathbf{\Psi} = [\,\mathbf{\Psi}_1\ \mathbf{\Psi}_2\ \ldots\ \mathbf{\Psi}_n] \in R^{n\times n}$ represent the transformation matrix at which the signal has a sparse representation, then

$$\mathbf{x} = \mathbf{\Psi s} \tag{1}$$

where $\mathbf{s}_{n\times 1}$ is the vector of coefficients that represents the sparse representation of the original signal $\mathbf{x}$ at matrix $\mathbf{\Psi}$. The signal $\mathbf{x}$ can be represented as a sparse vector if most of the elements of $\mathbf{s}$ are zero. Vector $\mathbf{s}$ is v-sparse if it only has v nonzero elements (v is called the sparsity level). Many of the natural and engineered signals, at some certain basis $\mathbf{\Psi}$ (wavelet, Fourier, etc.), are sparse (note that if $\mathbf{x}$ itself is sparse, then $\mathbf{\Psi} = \mathbf{I}$).

The sparsity assumption is the foundation to reconstruct the signal from fewer samples. In other words, from the sparsity assumption in Eq. (1), there is no need to sample the entire signal; instead sampling those nonzero elements would be enough for signal reconstruction. However, it would be very difficult or even impossible to know which elements are nonzero beforehand. Thus, CS encoding proposes a novel sampling strategy, based on which it is possible to take a smaller number of samples from the signal, and yet ensure that they contain sufficient information to reconstruct the signal. CS encoding states that each sample should be represented as a linear combination of each elements of original signal $\mathbf{x}$. Let $\mathbf{\Phi} = [\,\mathbf{\Phi}_1\ \mathbf{\Phi}_2 \dots \mathbf{\Phi}_n] \in R^{m\times n}$ be a sampling/sensing matrix with $m < n$ representing the number of samples, then the compressive sensing samples can be represented as

$$\mathbf{y} = \mathbf{\Phi x} + \boldsymbol{\varepsilon} \tag{2}$$

where $\mathbf{y} \in R^{m\times 1}$ denotes the samples/measurements, and $\boldsymbol{\varepsilon} \in R^{m\times 1}$ represents the sensing noise. CS encoding explains the conditions and the procedures to provide CS samples $\mathbf{y}$ from the original signal $\mathbf{x}$, hence, it can be realized as a forward modeling approach.

Substituting Eq. (1) into Eq. (2) yields to

$$\mathbf{y} = \mathbf{Ds} + \boldsymbol{\varepsilon} \tag{3}$$

where $\mathbf{D} = \mathbf{\Phi\Psi}$ represents an effective dictionary. From Eq. (3), it can be realized that the measurements $\mathbf{y}$ could be utilized to estimate the sparse representation vector $\mathbf{s}$, as there are only a few nonzero elements that should be estimated. Reducing the number of samples from n to m, from the original signal $\mathbf{x}$, might result in the loss of information, and consequently, accurate signal reconstruction becomes impossible. To avoid the loss of information, CS encoding enforces some conditions on sampling strategy. These conditions are mainly related to the properties of the sensing matrix $\mathbf{\Phi}$, or its variant effective dictionary $\mathbf{D}$. One of these conditions is named *restricted isometry property* (RIP). The RIP condition is met if the isometry constant $\delta_v \in (0,1)$ is such that the inequality of

$$(1-\delta_v) \leq \frac{\|\mathbf{D}_T \mathbf{s}_T\|_2^2}{\|\mathbf{s}_T\|_2^2} \leq (1+\delta_v) \tag{4}$$

holds for all v-sparse vector $\mathbf{s}$ (i.e. $\mathbf{s}_T$), where T denotes any subset from $\{1,..,n\}$ with cardinality v. In other words, the RIP condition requires all the possible subsets of the columns of $\mathbf{D}$ with cardinality v (i.e.$|\mathbf{D}_T|$) be as orthogonal as possible. Note that it cannot be exactly orthogonal as the number of rows m is smaller than the number of columns n.

It has also been shown that a random matrix $\mathbf{\Phi}$ (e.g., sampled from a Gaussian distribution) satisfies the RIP condition with high probability [4]. But for an arbitrary matrix $\mathbf{\Phi}$, it is NP-hard to compute the isometry constant δ_v [5]. This brings a computational issue in designing a proper sensing matrix $\mathbf{\Phi}$ in some real-world application once the random sampling is not practical.

The second criterion to evaluate matrix $\mathbf{D}$ is mutual coherence, which is more computationally tractable than the RIP criterion. Let $\mathbf{G} = \widetilde{\mathbf{D}}^{\boldsymbol{T}}\widetilde{\mathbf{D}}$ be the Gram matrix of $\widetilde{\mathbf{D}}$ computed from matrix $\mathbf{D}$ after normalizing each of its columns [6]. Then, mutual coherence $\mu(\mathbf{D})$ is defined as the largest absolute off-diagonal elements of matrix $\mathbf{G}$, namely,

$$\mu(\mathbf{D}) = \max\left|g_{ij}\right| \ 1 \leq i, j \leq n, i \neq j \tag{5}$$

where g_{ij} represents the i-th element of the j-th column of matrix $\mathbf{G}$. Mutual coherence measures the highest correlation between columns of matrix $\mathbf{D}$. From Eq. (5), clearly, mutual coherence μ can be efficiently computed, as opposed to δ_v which is not computationally tractable. Essentially, a large $\mu(\mathbf{D})$ represents a vulnerability of sensing matrix $\mathbf{\Phi}$ for CS encoding as it results in a loss of information from the signal after reducing the number of samples.

From effective dictionary $\mathbf{D} = \mathbf{\Phi\Psi}$, assuming that matrix Ψ is known beforehand (e.g. it is known that the signal is sparse using wavelet transformation), an optimization opportunity exists in designing a proper sensing matrix $\mathbf{\Phi}$ by optimizing the mutual coherence. Namely, an optimal matrix $\mathbf{\Phi}$ can be computed by minimizing mutual coherence $\mu(\mathbf{D})$. This is certainly useful in our proposed sensor-based predictive analytics, since following this optimization, not only can the number of sensors be reduced, but the loss of information is minimized, hence, the most representative information are collected.

The next task is to reconstruct the signal from the reduced number of samples, which is referred to as CS decoding. This task is essentially related to solving Eq. (3), hence it can be represented as an inverse model as well. In other words, from the reduced number of measurements $\mathbf{y}$, the latent variable of the system should be estimated. In the context of CS theory, CS decoding is about estimating the sparse representation vector $\mathbf{s}$ in Eq. (3). Given the estimate of the sparse vector $\hat{\mathbf{s}}$, the original signal can be reconstructed by using an inverse transformation, namely, $\hat{\mathbf{x}} = \mathbf{\Psi^{-1}}\hat{\mathbf{s}}$. However estimating $\mathbf{s}$ in Eq. (3) is challenging due to its inherently ill-posed ($m << n$), hence, there are infinite possible solutions. The linear equations in Eq. (3) are referred as underdetermined system of linear equations, as the number of equations are smaller than of the unknown variables.

The sparsity assumption of a signal provides reasonable conditions to overcome the underdetermined nature of the sparse representation of signal, and thereby achieve a unique solution. This formulates the estimation task into a sparse estimation/representation problem as follows:

$$\min \|\mathbf{s}\|_0 \quad \text{s.t. } \|\mathbf{y}-\mathbf{Ds}\|_2^2 < \delta \tag{6}$$

where $\|\mathbf{s}\|_0$ is the l_0-norm of the vector $\mathbf{s}$ which simply represents the number of non-zero elements of the vector, and δ is the noise level that controls the accuracy of the solution for the noisy case (i.e., it is assumed that that noise level is known). Unfortunately, solving Eq. (6) is NP-hard [7] as it needs a combinatorial search, i.e., to systematically search all possible subsets of $\mathbf{s}$ with a minimum number of non-zeros. Accordingly, a number of sparse estimation algorithms are reported in literature, such as convex optimization algorithms [8-11], greedy algorithms [12-14], and Bayesian algorithms [15-17].

Recent development in the field of sparse estimation reveals that if $\mathbf{s}$ is sufficiently sparse, then the solution to Eq. (1) is equivalent to that from a relaxed convex problem attained from replacing $\|\mathbf{s}\|_0$ with $\|\mathbf{s}\|_1$, where, $\|\cdot\|_1 = \sum|\cdot|$ is the l_1-norm [18]. The sparse estimation problem on account of the relaxed convex optimization modification is,

$$min\, \|\mathbf{s}\|_1 \quad \text{s.t. } \|\mathbf{y}-\mathbf{Ds}\|_2^2 \leq \delta \tag{7}$$

Convex optimization algorithms, such as Dantzig selector [8], basis pursuit de-noising (BPDN) [9], total variation (TV) regularization [10], and lasso [11] have been developed to solve the relaxed convex optimization problem of Eq. (7). A typical drawback of these algorithms is that, due to their global search, they are computationally intensive for large dimensional problems.

To overcome the computational burden of the aforementioned approaches, so-called Greedy algorithms, such as thresholding algorithm [14], orthogonal matching pursuit (OMP) [14], stagewise orthogonal matching pursuit (STOPM) [12], and least angle regression (LARS) [13] have been proposed. These Greedy algorithms are based on a heuristic search for the support set $S = \{i : \mathbf{s}_i \neq 0, \forall i = 1, \ldots, n\}$, which represents the location of nonzero elements in vector $\mathbf{s}$; they are computationally fast due to their greedy nature. However, there is an inherent tradeoff, the sparse estimation performance of Greedy algorithms often sacrifices accuracy for speed, as they are susceptible to be trapped in a local minima.

While the convex optimization algorithms and greedy algorithms mentioned above provide a point estimate for coefficient $\mathbf{s}$, in contrast, Bayesian algorithms, such as relevance vector machine (RVM) [15, 16], and sparse Bayesian learning (SBL) [17] provide a posterior distribution on the coefficient $\mathbf{s}$. The Bayesian algorithms have recently received more attention than other sparse estimation algorithms, due to their comparatively better (more accurate) sparse estimation performance. These algorithms assume a Gaussian likelihood function, and a conjugate prior promoting sparsity on the coefficient $\mathbf{s}$. By applying the Bayesian rule, the posterior distribution of $\mathbf{s}$ is estimated. The posterior mean of the estimated posterior distribution can be utilized as a point estimator for $\mathbf{s}$. Despite the accurate sparse estimation performance of Bayesian algorithms, their estimation procedures typically require several matrix inversion operations, hence, they are still computationally slow for high dimensional problems.

Given the existing sparse estimation algorithms reviewed above, the sensor-based predictive analytics is doable using a reduced number of samples. In other words, the complexity of the engineering systems can be effectively tackled by the sparse estimation algorithms even if the number of measurements are much smaller than the number of latent variables. This is because of the fact that only a few of these latent variables are active (sparsity assumption). In this dissertation, the CS decoding is proposed as an underlying methodology to enable predictive analytics using a reduced number of sensor data.

2.2 CS Approaches for Sensor-based Predictive Analytics

The proposed methodology in this dissertation contains the integration of CS theory into sensor-based predictive analytics. Following the CS theory, it would be possible to estimate the unknown variables of a system using a reduced number of sensors. As reviewed in Chapter 2, CS theory encompasses CS encoding, and CS decoding. The proposed methodology integrates these two tasks into sensor-based predicative analytics as follows:

- The CS encoding is utilized as an optimization approach in reducing the number of sensors (high dimensionality of the sensor data), while it ensures that the measurements contain the sufficient information from the process. Hence the contribution of the proposed methodology is in the sensing aspect.
- The CS decoding is utilized as the analytical approach to analyze the reduced sensor data, and accommodate effective decision making given complexity of the system. Hence, the contribution of the proposed methodology is in the predictive analytics.

The proposed methodology is a combination of *CS sensing* and *CS predictive analytics* which directly result from incorporation of *CS encoding* and *CS decoding*, respectively. This modifies the diagram of sensor-based predictive analytics in Figure 1-1. Figure 2-1 illustrates the modification where the two entities of sensing and predictive analytics are replaced with CS sensing and CS predictive analytics, respectively. From Figure 2-1, it can be observed that the sufficient process information with a reduced number of dimension can be collected using the sensing matrix $\mathbf{\Phi}$ proposed in CS theory, which is referred as *CS sensing* in the dissertation. The collected information represented as sensor data $\mathbf{y}$ are utilized for conducting predicative analytics based on the sparse estimation problem presented in Eq. (**7**), which is referred as *CS predictive analytics* in the dissertation. The outcome of CS predictive analytics is the estimate of sparse representation vector $\hat{\mathbf{s}}$, and consequently $\hat{\mathbf{x}} = \mathbf{\Psi}^{-1}\hat{\mathbf{s}}$ based on which quality improvement decisions are made accordingly.

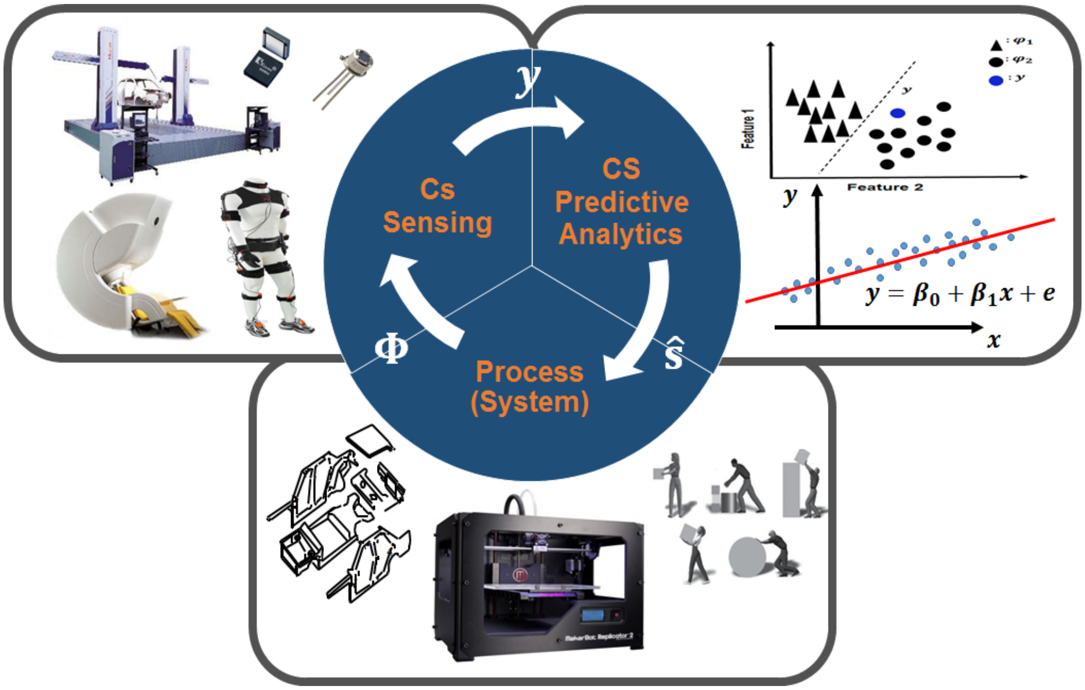

Figure 2-1 Compressive sensing for sensor-based predictive analytics

Following the proposed methodology in Figure 2-1, a number of compressive sensing approaches are developed in the dissertation to accommodate sensor based predictive analytics. As motivated by CS encoding, optimization approaches are developed to design an optimal sensing matrix $\mathbf{\Phi}$ by considering the RIP condition, or mutual coherence. The optimization approaches are referred as sensor placement optimization, and sensor selection optimization in the dissertation. The sensor placement optimization approach indicates that given a reduced number of sensors, the sensors are optimally located in order to collect the most explanatory information from the process. The sensor selection optimization approach relates to the efforts to determine/select the sensors that provide the most relevant information of the process. Both of these optimization approached are directly related to the CS sensing entity in the proposed sensor-based predictive analytics. CS predictive analytics contains the compressive sensing approaches utilized to accommodate predictive analytics using the data collected by CS sensing. Based on the outcome of the CS predictive analytics, quality improvement decisions are made, including fault diagnosis and real-time monitoring. The CS approaches developed for CS predictive analytics contain the contributions to the existing sparse estimation algorithms in the literature, which are utilized for fault diagnosis decision making (offline). Moreover, a novel sparse estimation algorithm is developed in the dissertation that is subsequently utilized for real-time monitoring decision making (online).

The developed CS approaches are generic and can be utilized in different applications such as manufacturing or service systems. However, in this dissertation, each of these approaches are introduced and verified in accordance with a specific application. For example, the proposed sensor placement and fault diagnosis approaches are verified for multi-station assembly processes, the real-time monitoring approach is validated through advanced manufacturing processes such as additive manufacturing, and the sensor selection optimization is studied for wearable sensor technology. An overall research work of this dissertation is presented in Figure 2-2.

The left hand side of the Figure presents the manufacturing systems application including multi-station assembly processes (please see Chapters 3 and 4) and additive manufacturing processes (please see Chapter 5), and the right hand side presents the service system applications such as wearable sensing systems (please see Chapter 6). In Chapter 7, as a part of the future research work, remote health monitoring using multiple sensor data is explained.

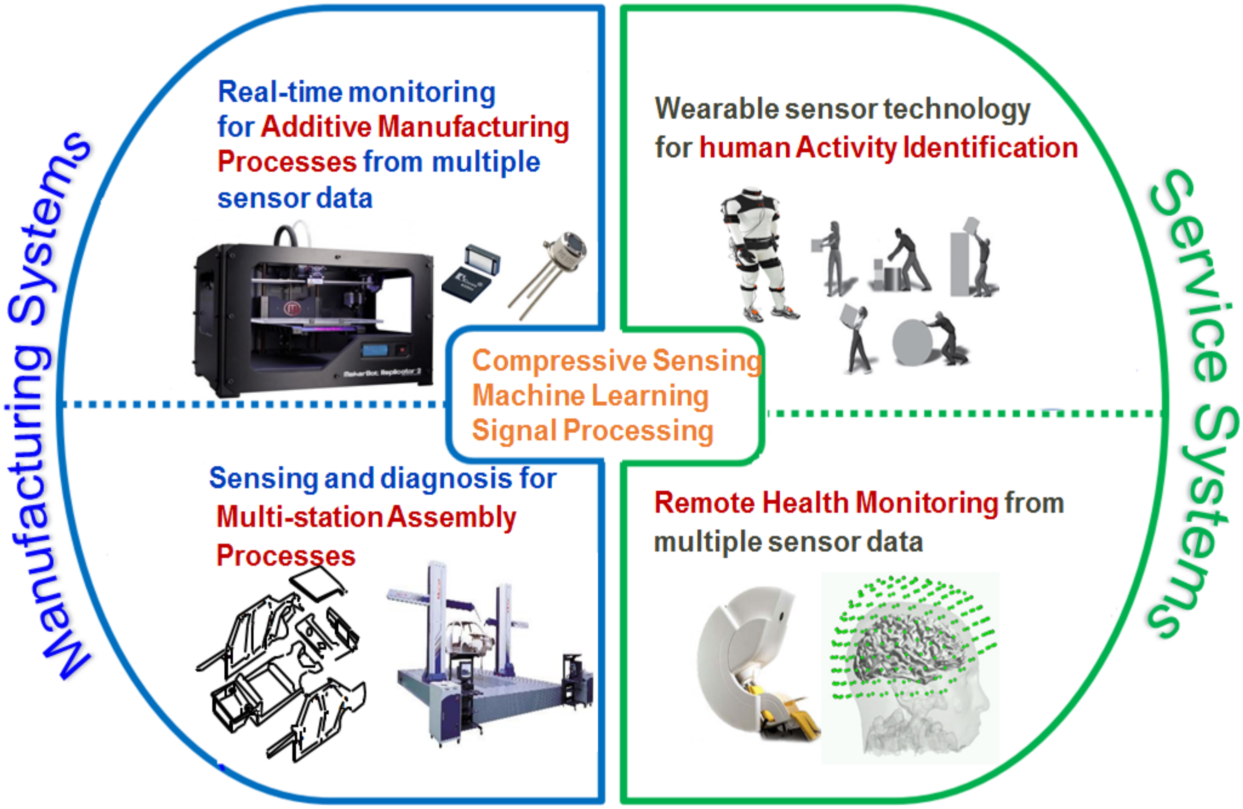
Manufacturing Systems
Service Systems
Real-time monitoring for Additive Manufacturing Processes from multiple sensor data
Wearable sensor technology for human Activity Identification
Compressive Sensing
Machine Learning
Signal Processing
Sensing and diagnosis for Multi-station Assembly Processes
Remote Health Monitoring from multiple sensor data

Figure 2-2. Overall research work and application

Chapter 3. Fault Diagnosis for Multi-station Assembly Processes

Dimensional integrity has a significant impact on the quality of the final products in multi-station assembly processes. A large body of research work in fault diagnosis has been proposed to identify the root causes of the large dimensional variations on products. These methods are based on a linear relationship between the dimensional measurements of the products and the possible process errors, and assume that the number of measurements is greater than that of process errors. However, in practice the number of measurements is often less than that of process errors due to economic considerations. This brings a substantial challenge to the fault diagnosis in multi-station assembly processes since the problem becomes solving an underdetermined system. In order to tackle this challenge, in this chapter, a fault diagnosis methodology is proposed by integrating the state space model with the enhanced relevance vector machine (RVM) to identify the process faults through the sparse estimate of the variance change of the process errors. The results of case studies demonstrate that the proposed methodology can identify process faults successfully[1].

3.1 Introduction

Dimensional quality is a measure of integrity between the actual dimension of the final product and its design nominal. Dimensional integrity has a significant impact on the quality of the final product in multi-station assembly processes because a major part of all quality-related problems belongs to the dimensional problems [19]. Multi-station assembly processes refer to systems that carry out operations on multiple work stations to assemble a final product [20]. Assembly

[1] Bastani, K., Kong, Z. Huang, W. Huo X., and Zhou, Y., 2013, “Fault diagnosis using an enhanced relevance vector machine (RVM) for partially diagnosable multi-station assembly processes,” *IEEE Transactions on Automation Science and Engineering*, 10(1), 124-136.

processes for automotive body, aerospace, and home appliances are typical examples of multi-station assembly processes.

In multi-station assembly processes, dimensional quality control is both critical and challenging since there are a large number of process variables (key control characteristics, i.e., KCCs) which may cause dimensional quality issues of the final assembled products. Some examples of KCCs are fixture locators, locating features on parts, etc. Among the KCCs, the accuracy of the fixture locators that are used to hold the parts throughout the assembly process contributes significantly to the dimensional quality of the final product. A very promising way to maintain or improve the dimensional quality is to reduce the fixture faults (e.g., improper installation and maintenance) [19]. However, this is a challenging task due to the complexity of multi-station assembly processes. For example, the effect of an improper fixture installation in early stations of the assembly process can propagate along the process, making it difficult to identify.

In order to monitor the dimensional integrity of multi-station assembly processes, various data collection techniques such as coordinate measuring machines (CMMs) and optical CMMs have been used to provide the dimensional measurements on key product characteristics (KPCs) of the assembled product. Traditional quality control techniques such as statistical process control charts (SPC) [21] use these dimensional measurements to monitor the dimensional quality of the product, and detect the process change in the multi-station assembly processes. However, SPC itself does not provide diagnostic capability (fault diagnosis, sometimes referred to as root cause identification) due to the nature of SPC methods [22-24]. Targeting the limitation of SPC methods in multi-station assembly processes, there has been a dire need to develop fault diagnosis methodologies which aim to identify the root causes of the large variation on the key product characteristics of the assembled products.

Considerable efforts have been devoted to fault diagnosis based upon an integration of the dimensional measurements with the process/product information [1, 23, 25-30]. In this type of work, the effects of the process errors on the dimensional measurements is represented by a linear model as follows,

$$\mathbf{y} = \mathbf{\Gamma}.\mathbf{u} + \boldsymbol{\varepsilon} \tag{8}$$

where $\mathbf{y}$ is a column vector ($m \times 1$) that represents the dimensional measurement; $\mathbf{u}$ is a column vector of n process errors; $\mathbf{\Gamma}$ is a fault pattern matrix ($m \times n$) obtained from product and the process information; and $\boldsymbol{\varepsilon}$ denotes the noise (a combination of modeling uncertainty and measurement noise). A clarification for the terminologies used in this subchapter is as follows. Process errors ($\mathbf{u}$) are different from process faults. Process errors refer to the variation of all the KCCs. They always exist in the assembly process and their specification limits are determined in the design process. Process faults refer to the process errors whose variations exceed the design specifications. The process faults are in the form of the mean shifts and variance increase of KCCs. The fault diagnosis methods aim at estimating both of them. This subchapter focuses on variance estimation.

In the reported work by Refs. [1, 23, 25-30], the linear model represented by Eq. (8) assumes to have a larger number of measurements than process errors, namely $m > n$. However, due to economic considerations, the measurement equipment/sensors cannot be utilized excessively. Thus, in practice the number of process errors will be larger than the number of measurements ($m < n$). Then the methods developed in Refs. [1, 23, 25-30] may not work for this scenario. This brings a great challenge to fault diagnosis since Eq. (8) becomes an underdetermined system. When the number of equations is less than the number of unknown variables in Eq. (8), mathematically there are an infinitely large number of solutions for the underdetermined system.

By taking into consideration the dependency of equations in Eq. (8), even with $m>n$, if the rank of matrix Γ <n, Eq. (8) is still an underdetermined system. In this subchapter, we call the multi-station assembly process with this property of underdetermined system as partially diagnosable due to the non-existence of a unique solution. Thus, accurate fault diagnosis becomes almost impossible for a partially diagnosable assembly process since there is no unique solution for an underdetermined system.

In order to tackle the challenge above, a further step to enhance fault diagnosis for multi-station assembly processes shall be taken by considering the scenario of underdetermined systems. This viewpoint mathematically translates fault diagnosis into the search for sparse solution. Sparse solution of an underdetermined system in the form of Eq. (8) is the solution to vector u that has few nonzeros among the possible entries in this vector [31]. Sparse solution is in compliance with actual multi-station assembly processes. It can be understood as the probability of having less process faults is higher than the probability of having more process faults, assuming the process faults are independent of each other [30]. Sparse estimation of the process errors that solves the underdetermined system for multi-station assembly processes serves as the key contribution of the proposed method in this subchapter.

3.2 Related Research Background

This subchapter proposes a fault diagnosis method that utilizes the sparse estimate of the underdetermined system for multi-station assembly processes. In the subsequent sections, the related research work in fault diagnosis methodologies and the sparse solution methods is briefly introduced.

3.2.1 Fault Diagnosis Methodologies in Multi-Station Assembly Processes

Research work related to the fault diagnosis methodologies in multi-station assembly processes can be divided into two categories: the single-station case and the multi-station case. Some studies under the single-station case category are based upon the assumption of single fault [26, 30]. Multiple-fault diagnosis for a single-station case is reported in Ref. [1, 25, 27]. Ceglarek and Shi [26] proposed a single-fault diagnosis method for rigid parts in the autobody assembly. Their model is based on fixture geometry and in-line measurements while using principle component analysis (PCA) with pattern recognition for fault mapping process. Rong, et al. [30] extended the work in Ref. [26] to the diagnosis of the single fault in sheet metal assemblies where the compliant characteristics of the parts are considered.

Some methodologies were developed for multiple-fault diagnosis in single-station assembly processes. Apley and Shi [1] developed a diagnosis approach for fixture-related faults in panel assembly. Using geometric information of the panel and fixture, their model utilized a least squares-based algorithm to estimate the faults. Chang and Gossard [27] proposed a computational method for multiple-fault diagnosis in assemblies. By simulating the assembly process, they identified the relationship between measurements and the fixture errors, and created a least squares-based approach for fault diagnosis. To tackle the weakness of least squares methods that are sensitive to the pattern definition among the process faults, Camelio and Hu [25] presented a fault diagnosis method using designated component analysis (DCA). Essentially, DCA is a special case of least-squares method, but the fault patterns are approximated to be orthonormal.

All the above reviewed multiple fault diagnosis methods for single-station assembly processes are all engineering-driven approaches that directly connect the engineering domain knowledge of the process errors with the multivariate measurements on KPC through a mathematical model. Apley and Shi [32] present a data-driven fault diagnosis method which solely relies on the measurement data. They developed a methodology using factor analysis to estimate the fault pattern matrix from the multivariate measurement data. They assume that the fault pattern matrix has a ragged lower triangular form, and the noise is independent and homogeneous.

Fault diagnosis for multi-station processes is another category of the related research work. In this category, both single fault and multiple-fault diagnoses are reported in Refs. [23, 28, 29, 33]. Ding, et al. [28] utilized a novel state space approach for single fault diagnosis in multi-station assembly processes. Their model is based on the product/process design parameters and in-line measurements, and the fault diagnosis is carried out by means of a PCA-based pattern recognition approach.

Multiple-fault diagnosis in a multi-station case seems to be more challenging than other cases. Liu, et al. [34] proposed an engineering-driven factor analysis to identify the multiple faults in multi-station manufacturing processes. They use the engineering domain knowledge of the relationship between the process errors and the KPC to construct some qualitative indicator vectors. These indicator vectors are used as a guide for the factor analysis of the multivariate measurement data. They assume that the noise is independent and homogeneous.

Zhou, et al. [23] applied a mixed linear model to represent the relationship between the measurements and the process faults. Maximum likelihood method was used for mean and variation estimation due to its known statistical properties such as consistency, asymptotic normality and efficiency. They also applied minimum norm quadratic unbiased estimation (MINQUE) as an approximation of the maximum likelihood method for the large sample size. Instead of point estimation, they provided a confidence interval for the estimation results (mean and variance).

Ding, et al. [33] developed a multiple-fault diagnosis method for singular manufacturing systems. Due to the singularity issues, the least squares method cannot be applied to estimate the random deviations of process faults. They reformulated the original variation propagation model as a covariance relation and applied the least squares method to estimate the variance components of the model. Kong, et al. [29] used a state space approach to develop the 3D variation propagation model in multi-station assembly processes. Orthogonal diagonalization was applied to project the covariance matrix of the measurement data onto each axis of the affine space; thus the variance of process errors can be estimated accordingly.

Kong, et al.'s [29] diagnosis method is based on an assumption that the number of measurements is greater than the number of process faults, and the fault pattern matrix has linearly independent columns (i.e., full column rank). Ding, et al. [33] developed a methodology relaxing these assumptions. However, they assumed that the noise term in the variation propagation model has an identical variance. Zhou, et al. [23] developed a more challenging fault diagnosis method compared to Refs. [29, 33]. However, for estimating a large number of variance components, their maximum likelihood approach may lead to severe over-fitting.

As reviewed above, there has been an abundance of fault diagnosis research work in the literature. However, a methodology that is capable of diagnosing the multiple faults in a multi-station assembly process under the scenario of underdetermined systems remains elusive, providing the motivation and main focus of this subchapter.

3.2.2 Sparse Solution Methods

To have a unique solution for underdetermined systems and avoid an infinite number of solutions, the sparse solution assumption is necessary [35]. Solving the sparse solution for an underdetermined system is essentially an optimization problem which aims to minimize the l_0-

norm of the coefficient vector. Here, l_0-norm represents the number of nonzero elements of a vector. Thus, in this optimization problem, the minimum number of nonzero elements of the coefficient vector should be determined. Solving this optimization problem in general is NP-hard [36] and thus approximate solutions are necessary. Several algorithms are reported in the literature to achieve suboptimal solutions [15, 35, 37-49] which can be divided into two groups: deterministic optimization methods [35, 37-42, 44-46], and statistical-based optimization methods [15, 43, 47-49]. Greedy algorithms [39, 44, 45], basis pursuit [35, 38, 40], iterative reweighted least squares methods [42, 46], and iterative thresholding methods [37, 41] are some of the deterministic optimization methods reported in the literature. Some algorithms, such as relevance vector machine (RVM) [15, 43, 47] and maximum a posterior method [48, 49] are the statistical-based optimization algorithms used to find the sparse solution.

Deterministic algorithms [35, 37-42, 44-46] solve a suboptimal sparse solution to the optimization problem in a deterministic manner. These algorithms relax the optimization problem into a more computationally solvable optimization problem and apply some deterministic approaches such as linear programming, greedy methods, and least squares methods to solve the relaxed problem. However, some of these algorithms [35, 38-40, 44, 45] can be applied only for noiseless underdetermined systems (i.e. $\mathbf{y} = \mathbf{\Gamma}.\mathbf{u}$).

In the statistical-based methods [15, 43, 47-49], the sparse solution is considered an estimate of the coefficient vector and thus statistical estimation approaches are applied. They assume a probability density function (mostly zero mean Gaussian distribution) for the noise term and estimate the mean and variance of the coefficient vector. The estimate of the mean of the coefficient vector can be determined by maximizing its posterior distribution [48, 49]. The

Bayesian approach, along with defining hyperprior on the coefficient vector, can also be applied to provide the full posterior on the coefficient vector [15, 47].

The fault diagnosis problem for partially diagnosable multi-station assembly processes is essentially to solve an underdetermined system which is also contaminated with noise. The need to estimate the parameters (mean and variance) of the process errors for quality control is the key reason to choose the statistical based methods. Furthermore, among the statistical based methods, RVM [15] closely matches the objective of our fault diagnosis problem. It is an approach that presents a complete statistical solution for the sparse estimation through using probability density functions. The RVM also provides an online sparse estimate due to its Bayesian framework. The estimate of the noise variance is required in the fault diagnosis and can be effectively carried out by using the RVM method. This subchapter proposes a methodology integrating state space approach with an enhanced RVM to estimate the variance components of the underdetermined multi-station assembly process.

3.2.3 Contribution and Overview

The proposed fault diagnosis methodology for partially diagnosable multi-station assembly processes has two contributions over previous studies: (1) from a fault diagnosis point of view, the concept of sparse solution is applied to identify the root causes of a partially diagnosable system; the assumptions of homogenous noise in refs. [28, 29, 33] and even the assumption of independent noise in Ref. [23] are relaxed to enable a very general case where noise has a general covariance matrix; and (2) an enhanced RVM methodology is developed to tackle the challenge of solving an underdetermined system by relaxing the assumption of iid noise in the existing RVM. Thus, this enhanced RVM methodology is capable of handling fault diagnosis for partially diagnosable systems. Even if only partial process/product information is available, i.e., only partial columns of the fault pattern matrix $\mathbf{\Gamma}$ in Eq. (8) are known, the process faults can still be identified. These contributions fill the vacancy in the related research areas, and also provide very effective tools for the practitioners to identify the root causes of dimensional quality problems in large and complex manufacturing systems. The proposed fault diagnosis method involves three steps: (i) constructing the variation model of the multi-station assembly processes considering the general noise, (ii) converting the variation model from a matrix form into a vector form through a vectorization operator leading, (iii) determining the sparse estimate of the variance change of the process errors which are denoted as the coefficient vector in the Vectorized variance model in step (ii).

This subchapter is organized as follows: In Sec. 3.3, the state space approach used to model the variation propagation is briefly reviewed. Sec. 3.4 proposes the fault diagnosis methodology using an enhanced RVM for underdetermined systems in multi-station assembly processes. In Sec. 3.5, case studies that demonstrate and validate the proposed methodology are presented. Finally, Sec. 3.6 summarizes the subchapter and also provides the concluding remarks.

3.3 Variation Propagation Model for Multi-Station Assembly Process

The multi-station assembly process illustrated in Figure 3-1 [29] has m stations where variable i is the station index. State vector x_i denotes the product quality information (e.g., part dimensional deviations) at station i; u_i is the input variable representing the process errors such as fixture deviations; and the process noise and unmodeled errors are denoted by $\mathbf{w}_i$. The measurements on KPCs are represented by $\mathbf{y}_i$ and it must be noted that y_i is not necessarily available in every station. The variable $\boldsymbol{v_i}$ is used to represent the measurement noise.

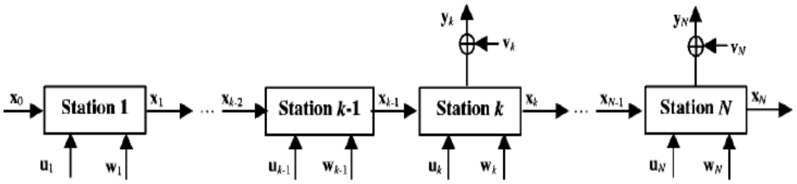

Figure 3-1. Diagram of a mlti-station assembly process

A generic 3-2-1 fixture scheme presented in Figure 3-2 is used in multi-station assembly processes to constrain the degrees of freedom of the rigid parts during the process. NC block locaters (C1, C2, C3), a four-way pin P1, and a two-way pin P2 provide the 3-2-1 fixture layout as presented in Figure 3-2. The deviations of the NC blocks in direction Y, P1 in directions of X and Z, and P2 in direction of Z denote the process/fixture error which consequently cause the deviation of the part to be assembled. The generic 3-2-1 fixture layout is incorporated to the state-space model to present the variation propagation throughout the assembly process.

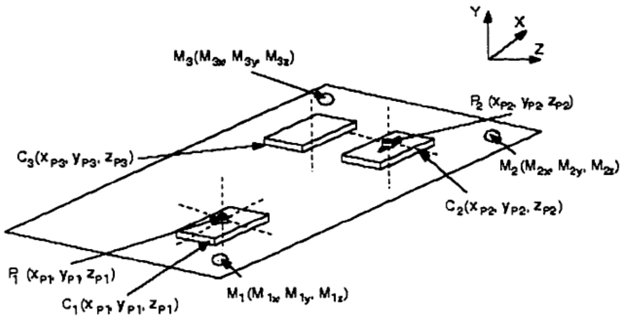
Y
X
Z
$M_3(M_{3x}, M_{3y}, M_{3z})$
$P_2(x_{P2}, y_{P2}, z_{P2})$
$C_3(x_{P3}, y_{P3}, z_{P3})$
$M_2(M_{2x}, M_{2y}, M_{2z})$
$C_2(x_{P2}, y_{P2}, z_{P2})$
$P_1(x_{P1}, y_{P1}, z_{P1})$
$C_1(x_{P1}, y_{P1}, z_{P1})$
$M_1(M_{1x}, M_{1y}, M_{1z})$

Figure 3-2. A generic 3-2-1 fixture scheme

State-space models were adopted to model variation propagation in sheet metal assembly and multi-station manufacturing processes by [50, 51] respectively. Huang, et al. proposed a 3-D variation propagation model using state-space modeling [52, 53]. However, their method can model a datum surface with a single feature. Liu, et al. [54] proposed a state-space modeling to model 3-D variation propagation in multi-station assembly processes which is capable of dealing with a datum surface composed of multiple features on different parts. In this section, the overview of the variation propagation model in multi-station assembly processes developed by [50] is presented.

There are some assumptions in this linear state-space model. The components of u_i are assumed independent of each other. Based on the principles of the 3-2-1 fixture scheme, it is reasonable to assume that the deviation of one of the fixture locator has no effect on that of the others; hence they are independent to each other. w_i and v_i are assumed to be zero mean and all of their elements are independent of each other. Also, it is assumed that the elements of u_i, w_i and v_i are all independent of each other, and the magnitude of w_i is small compared to that of x_i and u_i. Considering all these assumptions together, the state-space model is expressed as follows:

$$\mathbf{x}_i = \mathbf{A}_{i-1}\mathbf{x}_{i-1} + \mathbf{B}_i\mathbf{u}_i + \mathbf{w}_i \qquad i = 1, 2, \dots, p \qquad (9.1)$$

$$\mathbf{y}_i = \mathbf{C}_i\mathbf{x}_i + \mathbf{v}_i \qquad i = 1, 2, \dots, p \qquad (9.2)$$

The part quality information $(\mathbf{x}_i)$ at station i is determined by three terms: (1) $\mathbf{A}_{i-1}\mathbf{x}_{i-1}$ representing the transformation of the product quality information from station i-1 o station i; (2) $\mathbf{B}_i\mathbf{u}_i$ representing the effects of the process errors at station i on the quality information, and (3) $\mathbf{w}_i$ representing process noise and unmodeled errors. Matrices $\mathbf{A}_i, \mathbf{B}_i$ and $\mathbf{C}_i$ are the state matrix, input matrix and observation matrix, respectively. These matrices are constant and determined by the product/process design information.

The state-space model in Eq. (9) using the recursive form of state variables $\mathbf{x}_i$can be reformulated into a linear model representing the relationship between the process errors $\mathbf{u}_i$ and end-of-line KPCs' measurements $\mathbf{y}_p$. The linear model is expressed as follows [50]:

$$\mathbf{y}_p = \sum_{i=0}^{p} \boldsymbol{\gamma}_i \mathbf{u}_i + \varepsilon \tag{10}$$

where $\boldsymbol{\gamma}_i = \mathbf{C}_p \boldsymbol{\phi}_{p,i} \mathbf{B}_i$, $\boldsymbol{\gamma}_0 = \mathbf{C}_p \boldsymbol{\phi}_{p,0}$, and $\boldsymbol{\phi}_{p,i} = \mathbf{A}_{p-1}\mathbf{A}_{p-2} \ldots \mathbf{A}_i$ for $p > i$ and $\boldsymbol{\phi}_{i,i} = \mathbf{I}$. The combination of process noise and measurement noise is denoted by

$$\boldsymbol{\varepsilon} = \sum_{i=1}^{p} \mathbf{C}_p\, \boldsymbol{\phi}_{p,i} \mathbf{w}_i + \mathbf{v}_i \tag{11}$$

It should be noted that for i=0, $\mathbf{u}_0$ represents the initial condition of state space vector such as the fabrication imperfection of the parts before entering the assembly process. Eq. (10) can be written in a matrix notation as Eq. (8) which is rewritten below (for simplification, index p of $\mathbf{y}$ is removed):

$$\mathbf{y} = \boldsymbol{\Gamma}.\mathbf{u} + \boldsymbol{\varepsilon} \tag{12}$$

where $\mathbf{u} = [\mathbf{u}_0\, \mathbf{u}_1\, \ldots\, \mathbf{u}_p]'$ is an $n \times 1$ column vector representing n process errors (in this subchapter we consider fixture locator deviations); $\boldsymbol{\Gamma} = [\,\boldsymbol{\gamma}_0\, \boldsymbol{\gamma}_1 \cdots\, \boldsymbol{\gamma}_p]$ is an $m \times n$ matrix which denotes the fault pattern related to the process errors where m is the number of measurements. Indeed, this linear model effectively links the effect of process errors (KCCs) to the measurements on KPCs of the product.

3.4 Fault Diagnosis Methodology for Multi-Station Assembly Processes

3.4.1 Variation Model Formulation for Multi-station Assembly Processes

As mentioned in Sec. 3.1, process faults are in the form of the mean shift and variance of the process errors **u** in Eq. (12). In this subchapter, **u** is assumed to be zero-mean; hence fault diagnosis can be formulated as a problem of variance estimation of process errors. The deviation model in Eq. (12) links the effects of process errors (KCCs) to the measurements on KPCs. The corresponding variance model is as follows:

$$Cov(\mathbf{y}) = \mathbf{\Gamma} Cov(\mathbf{u}) \mathbf{\Gamma}' + Cov(\boldsymbol{\varepsilon}) \qquad (13)$$

where $Cov(.)$ is the covariance matrix of a random vector. Since it is assumed that all the elements of u are independent of each other, $Cov(\mathbf{u})$ is a diagonal matrix (i.e., $cov(\mathbf{u}) = diag({\sigma_{1E}}^2, {\sigma_{2E}}^2, \dots, {\sigma_{nE}}^2)$) where n is the number of process errors and ${\sigma_{iE}}^2$ for $i = 1, \dots, n$ is the variance of the i-th process error. In this subchapter, we present two different cases for the noise term. First, it is assumed that the noise terms are independent. Thus $Cov(\boldsymbol{\varepsilon}) = diag({\sigma_1}^2, {\sigma_2}^2, \dots, {\sigma_m}^2)$ where m is the number of measurements, and ${\sigma_j}^2$ $(j = 1, \dots, m)$ are different from each other. This consideration relaxes the assumption in Ref. [27] which treats all ${\sigma_j}^2$ as identical. Secondly, we further relax the assumption of independent noise to enable a general covariance matrix. This treatment is more consistent to the state space model represented in Eq. (9).

Process faults can also be interpreted as the large variance of the process errors or KCCs (i.e., beyond the design specification). The process is under normal conditions unless the variance of some process errors (such as fixture locators) exceeds their design specification limits. Thus, variance of a process error exceeding the specification limit is considered as a process fault, which is a root cause for the dimensional quality issue, such as the large variance captured by SPC control charts on the measurement of KPCs. The variation model in Eq. (13) can be reformulated in a way that represents both (1) the design specification limit on variance and (2) the change of variance:

$$Cov(\mathrm{y}) = \Gamma . \begin{pmatrix} {\sigma_{1d}}^2 + \Delta_1 & & \\ & \ddots & \\ & & {\sigma_{nd}}^2 + \Delta_n \end{pmatrix} \Gamma' + \begin{pmatrix} {\sigma_1}^2 & & \\ & \ddots & \\ & & {\sigma_m}^2 \end{pmatrix} \qquad (14)$$

where ${\sigma_{id}}^2$ and Δ_i denote the design specification variance of process error i and its variance change, respectively. It is reasonable to assume that $\Delta_i \geq 0$ since the variance of the process errors should not be less than their design specification limits. Indeed, this formulation is the same as Eq.

(13), with ${\sigma_{iE}}^2$=${\sigma_{id}}^2 + \Delta_i$ (for $i = 1, .., n$). With Eq. (14), a process fault will happen if the corresponding Δ_i> 0 related to the process error i.

In order to simplify the form of Eq. (14), we introduce a vectorization operator $v(K)$ for a $k \times k$ symmetric matrix K, we have

$$v(K) = [\underbrace{K_{1,1}, K_{2,2}, \dots, K_{k,k}}_{\text{Diagonal entries}}, \underbrace{K_{1,2}, \dots, K_{1,k}, K_{2,3}, \dots, K_{k-1,k}}_{\text{Upper triangle entries}}]'$$

where $K_{i,j}$ is the (i,j)th entry of matrix K. The operator $v(.)$ first lists diagonal entries of the matrix K, following by rows in the upper triangle. Lower triangle entries are not listed since matrix K is symmetric. Utilizing $v(.)$, we denote

$$\mathbf{Y} = v\big(Cov(\mathbf{y})\big),$$

$$\mathbf{G}_\mathrm{i} = v\big(\boldsymbol{\Gamma_i}\boldsymbol{\Gamma}_i^{'}\big), \text{ i=1,...,n,}$$

where $\boldsymbol{\Gamma}_i$ is the ith column of $\boldsymbol{\Gamma}$ in Eq. (12), and

$$\boldsymbol{\Sigma}_\mathrm{j} = v\big(\boldsymbol{\Sigma}_j\big), j = 1, \dots, m,$$

where $\boldsymbol{\Sigma}_j$ is an $m \times m$ matrix whose (i,i)th entry is one, and all other entries are zero. Thus, by using the vectorization operation, Eq. (14) in a matrix form can be represented as an equivalent vector form as follows:

$$\mathbf{Y} = \sum_{\mathrm{i}=1}^{\mathrm{n}} ({\sigma_{id}}^2 + \Delta_i)\mathbf{G}_\mathrm{i} + \sum_{j=1}^{m} \sigma_j^2 \boldsymbol{\Sigma}_\mathrm{j} \qquad (15)$$

It can be seen that Eq. (15) is a linear system with $\frac{m\times(m+1)}{2}$ equations and (n+m) unknown variables, i.e., Δ_i (i=1,…,n) and σ_j^2 (j=1,…,m). Based on our assumption that if the number of independent equations, namely, the rank of the matrix of the first term at the right hand side of Eq.

(15) is less than (n+m), then Eq. (15) is an underdetermined system where the sparse estimate of Δ_i, as well as the estimate of variance components of noise term $\sigma_j{}^2$s are intended. This formulation provides a complete utilization of the measurements on the KPCs since both variance and covariance across the KPCs are included in the modeling. It also alleviates the degree of ill-posed system, since now we have $\frac{m\times(m+1)}{2}$ equations as opposed to Eq. (12) where there are m equations, i.e., $\frac{m\times(m-1)}{2}$ more equations are added. This advantage is due to the consideration of the covariance information on the measurement of KPCs.

In this subchapter, a further relaxation is also applied for the noise term. The above formulation in Eq. (15) assumes that the noise term has independent but non-identical variance. For the general case where the covariance matrix of the noise is not diagonal, the corresponding variation model is

$$Cov(y) = \Gamma.\begin{pmatrix} \sigma_{1d}{}^2 + \Delta_1 & & \\ & \ddots & \\ & & \sigma_{nd}{}^2 + \Delta_n \end{pmatrix}\Gamma' + \begin{pmatrix} \sigma_1{}^2 & \dots. & \sigma_{1m} \\ & \ddots & \\ \sigma_{m1} & \dots & \sigma_m{}^2 \end{pmatrix} \qquad (16)$$

where $\sigma_j{}^2$'s are the variance components of the noise, and σ_{ij} denotes the covariance components between the noise terms. Using the vectorization operator $v(.)$, the same as the above derivation, the variation model in Eq. (16), can be simplified in a vector form as

$$\mathrm{Y} = \sum_{\mathrm{i=1}}^{\mathrm{n}}(\sigma_{id}{}^2 + \Delta_i)\mathrm{G_i} + \sum_{i\leq j}^{m}\sigma_{ij}\sum\nolimits_{\mathrm{ij}} \qquad (17)$$

where $\sum_{\mathrm{ij}} = v(\Sigma_{ij})$, and $\mathbf{\Sigma}_{ij}$ is an $m \times m$ matrix whose (i,j)th entry is one and the rest are zeros. In Eq. (17), the variance and covariance components of the noise are denoted by $(\sigma_{ii},\ i = j)$ and $(\sigma_{ij}, i < j)$, respectively. As seen in Eq. (16), we have an underdetermined linear system where estimate of Δ_is, as well as the estimate of variance and covariance components of noise terms σ_{ij}s are intended. However, we have more ill-posed problems since more unknowns should be estimated, which come from the covariance components of the noise.

As mentioned above, the sparse estimate of the Δ_is, as well as the estimate of variance components of noise term σ_{ij}s is the core of the proposed fault diagnosis method for underdetermined system in multi-station assembly processes. In this subchapter, the sparse estimate is accomplished by means of an enhanced approach for the RVM [15]. In Sec. 3.4.2, the fault diagnosis method based on the enhanced RVM is presented in more details.

3.4.2 Fault Diagnosis for Underdetermined Systems using Enhanced RVM

RVM is a probabilistic Bayesian learning framework that achieves sparse solutions for linear regression models. It is based on the normal distribution assumption of the coefficient vector in the linear regression model. In multi-station assembly processes, the process errors are assumed to have a normal distribution. That is the most common distribution considered for the process errors in the literature and the practice. Thus the RVM methodology which assumes the normal distribution and the corresponding conjugate prior over the coefficient vector is in compliance with both literature and practice in multi-station assembly processes. However, the probabilistic Bayesian learning approaches reported in Refs. [15, 47] provide the estimate of the noise term's variance which is assumed to be identically independently distributed (iid). In our proposed method, this assumption is relaxed, i.e., $\sigma_j{}^2$ (i.e., σ_{jj}) in Eq. (17) do not have to be identical. This is more consistent with the actual scenario since different sensors may have different precisions and thus the resulting measurement noises are different. In this subchapter, the noise terms (σ_{ij} in Eqs. (16) and (17)) are still assumed to be independent of each other. This is reasonable since σ_{ij} in Eqs. (16) and (17) represents the variance/covariance of noise in Eq. (12). Along this line, there is a need to enhance the RVM method to estimate the covariance matrix of the noise which is independent, but non-identical. The proposed enhanced RVM method has three steps through which the variance change (Δ_{i}s) and noise term's variance components σ_{ij} in Eqs. (16) and (17) are iteratively estimated. Figure 3-3 illustrates the flowchart of the enhanced RVM method. The details of these steps are presented in the subsequent sections.

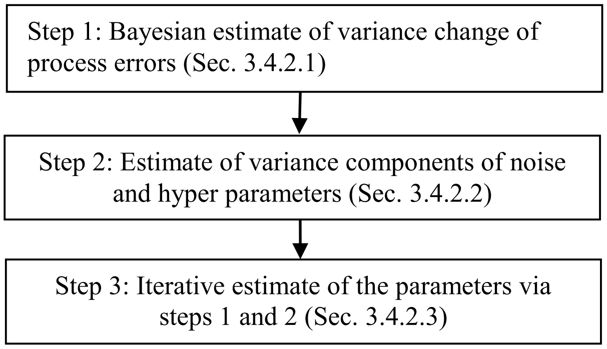
Step 1: Bayesian estimate of variance change of process errors (Sec. 3.4.2.1)
Step 2: Estimate of variance components of noise and hyper parameters (Sec. 3.4.2.2)
Step 3: Iterative estimate of the parameters via steps 1 and 2 (Sec. 3.4.2.3)

Figure 3-3. Procedure of the enhanced RVM method

3.4.2.1 Bayesian Estimate of Variance Change of Process Errors

The linear regression model in Eq. (15), where noise terms are assumed to be independent, can be simplified into

$$\mathbf{Y} - \sum_{\mathrm{i}=1}^{\mathrm{n}} ({\sigma_{id}}^2) \mathbf{G}_{\mathrm{i}} = \sum_{\mathrm{i}=1}^{\mathrm{n}} (\Delta_i) \mathbf{G}_{\mathrm{i}} + \sum_{j=1}^{m} \sigma_j^2 \mathbf{\Sigma}_{\mathrm{j}} \qquad (18)$$

In the equation above, for further simplification, we denote the left hand side term, which contain all the known terms with

$$\mathbf{t} = \mathbf{Y} - \Phi \bullet [{\sigma_{1d}}^2, {\sigma_{2d}}^2, \cdots, {\sigma_{nd}}^2]' \qquad (19)$$

where $\Phi = [\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_{\mathrm{n}}]$ is an $\frac{m \times (m+1)}{2} \times n$ matrix. Column vector $\mathbf{t}$ has the dimension of $\frac{m \times (m+1)}{2}$ that is obtained from the difference between the actual measurements $\mathbf{Y}$ and the KCCs' variance defined by the design specifications (i.e. ${\sigma_{id}}^2{'s}$). Now, Eq. (19) can be simplified into the linear regression model presented below:

$$\mathbf{t} = \mathbf{\Phi w} + \boldsymbol{\xi} \tag{20}$$

where $\mathbf{w} = [\Delta_1, \Delta_2, \cdots, \Delta_n]'$ represents the variance change of each process error; $\boldsymbol{\xi} = [\sigma_1{}^2,\ \sigma_2{}^2, \cdots, \sigma_m{}^2, 0, \cdots, 0]'$ is a column vector with dimension of $\frac{m\times(m+1)}{2}$ and it can be seen that the "0" entries are due to the assumption of independent noise term in Eq. (20) ; and as before $\mathbf{\Phi} = [\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_\mathrm{n}]$ is an $\frac{m\times(m+1)}{2} \times n$ design matrix in this new linear regression formulation. The coefficient vector ($\mathbf{w}$) in Eq. (20) are the change of variance of all process errors, which will be estimated to determine if there are any process faults.

For the case of more general noise ($\boldsymbol{\varepsilon}$ in Eq. (12)) whose independency assumption is relaxed, the linear regression model represented in Eq. (17) can be written the same as Eq. (20) with the only difference as follows:

$$\boldsymbol{\xi} = \left[\sigma_1{}^2,\ \sigma_2{}^2, \cdots, \sigma_m{}^2, \sigma_{12}, \sigma_{13} \cdots, \sigma_{(m-1)m}\right]' \tag{21}$$

As mentioned before, the components in Eq. (21) can be reasonably assumed to be independent because they represent the variance/covariance (not the deviation) of noise in Eq. (12).

In the Bayesian framework, each component of $\boldsymbol{\xi}$ is treated as a random variable. They are assumed to be zero-mean and having Gaussian distribution. Here, we actually relax the assumption of noise variance being iid in RVM [15]. Thus, $\sum_{\sigma^2}$, the covariance matrix of $\boldsymbol{\xi}$ has a diagonal form. The likelihood function for $\mathbf{t}$ has a form of multivariate Gaussian distribution:

$$p(\mathbf{t}|\mathbf{w}, {\textstyle\sum}_{\sigma^2}) =$$

$$(2\pi)^{-\frac{m}{2}} |{\textstyle\sum}_{\sigma^2}|^{-\frac{1}{2}} \exp(-1/2(\mathbf{t} - \mathbf{\Phi w})^T {\textstyle\sum}_{\sigma^2}{}^{-1}(\mathbf{t} - \mathbf{\Phi w})) \tag{22}$$

where $|.|$ represents the determinant of a matrix. Estimation of the model's parameters including $\mathbf{w}$ and noise covariance ($\sum_{\sigma^2}$) in Eq. (22) is the focus of our interest. In the Bayesian analysis, the parameters are considered as random variables that are fully quantified by probability density functions, unlike the frequentist analysis where the parameters are constant values. In our enhanced RVM that follows the Bayesian framework, the full posterior over $\mathbf{w}$ should be identified. Thus, the main task is to estimate the posterior mean and covariance of w along with the covariance matrix ($\sum_{\sigma^2}$) of noise. By choosing a proper conjugate prior on w which is in favor of sparsity properties of the model, and using Bayes rule, posterior mean and covariance of $\mathbf{w}$ can be estimated. A multivariate zero-mean Gaussian prior distribution over w is as below [15]:

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{n} N\,(\mathbf{w}_i|0, \boldsymbol{\alpha}_{\mathbf{i}}{}^{-1}) \qquad (23)$$

where $\boldsymbol{\alpha}$ is a vector of n independent hyperparameters that are inverse variances (also referred to as precision) of $\mathbf{w}$, each of which is used to control the strength of the prior over its corresponding elements of w. RVM uses $\boldsymbol{\alpha}$ as the variable parameters, and infers their values from the actual measurement data. Therefore, there is a need to define additional hyperprior over the values of $\boldsymbol{\alpha}$ and the remaining parameters on the model which are related to the noise covariance matrix ($\sum_{\sigma^2}$). To promote sparsity, a non-informative prior (in this subchapter, a uniform hyperprior is considered), is used as the hyperprior, which implies that there is no information about the parameters before the inference process. Gamma distribution with its parameters fixed to zero is chosen to make the uniform hyperprior. For the normal distribution of the process errors, the variance is modeled with hyperparameters $\boldsymbol{\alpha}_i{}'s$ which are assumed to have a non-informative (uniform) prior. This assumption makes our proposed method more generic. Basically, we let the data speak for itself during the Bayesian updating process without restricting the prior to follow any specific informative distribution. This data driven nature enables a more effective way in handling the complex variability in measurement data. The normality assumption of the process errors and choosing the non-informative prior over the hyperparameters makes our proposed methodology consistent with industrial practice and also more flexible to handle data variability.

The posterior distribution over w given the covariance matrix ($\sum_{\sigma^2}$) of noise $\boldsymbol{\xi}$, hyperparameters $\boldsymbol{\alpha}$, and $\mathbf{t}$ can be computed analytically due to the conjugate prior over $\mathbf{w}$. The likelihood function $p(\mathbf{t}|\mathbf{w}, \sum_{\sigma^2})$ in Eq. (22) (**23**)and the prior distributions $p(\mathbf{w}|\boldsymbol{\alpha})$ in Eq. (23) are both multivariate Gaussian. Hence, using Bayes rule, the posterior distribution of $\mathbf{w}$ is derived as follows:

$$p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \textstyle\sum_{\sigma^2}) = \frac{p(\mathbf{t}|\mathbf{w}, \sum_{\sigma^2})p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{t}|\boldsymbol{\alpha}, \sum_{\sigma^2})} = (2\pi)^{-n/2}|\textstyle\sum|^{-1/2}\exp(-\frac{1}{2}(\mathbf{w} - \mu)^T\textstyle\sum^{-1}(\mathbf{w} - \mu)) \tag{24}$$

where the posterior covariance $\boldsymbol{\Sigma}$ and mean $\boldsymbol{\mu}$ of w are as follows,

$$\boldsymbol{\Sigma} = (\boldsymbol{\Phi}^{\mathbf{T}}\Sigma_{\sigma^2}^{-1}\ \boldsymbol{\Phi}\ + \mathbf{A})^{-1} \tag{25}$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma\Phi}^T\Sigma_{\sigma^2}^{-1}\mathbf{t} \tag{26}$$

where **A** is defined as $Diag(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_n)$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ are functions of hyperparameters $\boldsymbol{\alpha}$, and the covariance matrix Σ_{σ^2} of noise, which still remain unknown at this stage. Thus, estimating these unknown parameters will be necessary, and the procedure is presented in Sec. 3.4.2.2

3.4.2.2 Estimate of Hyperparameters and Variance Components of Noise

Since Σ_{σ^2} is a diagonal matrix, only its diagonal elements, i.e. $\boldsymbol{\beta} = [{\sigma_1}^2, {\sigma_2}^2, \ldots, {\sigma_m}^2]$, need to be estimated. These parameters can be estimated from the measurement data, i.e., $\mathbf{t}$ in Eq. (20) and by maximizing marginal likelihood of $\mathbf{t}$ (evidence maximization) [15]. For mathematical simplicity, we maximized the log of marginal likelihood function of $\mathbf{t}$. It can be carried out by taking the derivative of this function with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and then setting the partial derivatives as zero to identify optimal $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by achieving maximal marginal likelihood of $\mathbf{t}$. The marginal likelihood of $\mathbf{t}$ is a zero mean Gaussian distribution as below:

$$p(\mathbf{t}\ |\boldsymbol{\alpha}, \Sigma_{\sigma^2}) = \int p(\mathbf{t}\ |\mathbf{w}, \Sigma_{\sigma^2})p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w} \tag{27}$$

$$= (2\pi)^{-\frac{m}{2}} |\Sigma_{\sigma^2} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T|^{-\frac{1}{2}}$$

$$\cdot \exp[-\frac{1}{2}\mathbf{t}^T(\Sigma_{\sigma^2} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T)^{-1}\mathbf{t}$$

and its logarithm can be expressed as:

$$L(\alpha, \Sigma_{\sigma^2}) = \ln p(\mathbf{t}\,|\alpha, \Sigma_{\sigma^2}) = -\frac{1}{2}(mln2\pi + \ln|C| + \mathbf{t}^T C^{-1}\mathbf{t}) \tag{28}$$

where $C = \sum_{\sigma^2} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T$ is the covariance matrix for $\mathbf{t}$. By exploiting the determinant identity $\det(\mathbf{X} + \mathbf{D}\mathbf{E}) = det(\mathbf{X})\det(\mathbf{I} + \mathbf{B}\mathbf{X}^{-\mathbf{1}}\mathbf{E})$ (where $\mathbf{X}, \mathbf{D}$ and $\mathbf{E}$ are arbitrary matrices) and some basic algebra, it can be shown that [27]:

$$\ln|\mathbf{C}| = ln|\mathbf{\Sigma}^{-1}| - ln\left|\Sigma_{\sigma^2}^{-1}\right| - ln|\mathbf{A}| \tag{29}$$

Also, using Woodbury inversion identity which is $(\mathbf{M} + \mathbf{U}\mathbf{D}\mathbf{V})^{-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{U}(\mathbf{D}^{-1} + \mathbf{V}\mathbf{M}^{-1}\mathbf{U})^{-1}\mathbf{V}\boldsymbol{M}^{-1}$ [55] for $\mathbf{C}^{-1}$, we formulate the term $\mathbf{t}^T\mathbf{C}^{-1}\mathbf{t}$ as below:

$$\mathbf{t}^T\mathbf{C}^{-1}\mathbf{t} = (\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu})^T\Sigma_{\sigma^2}^{-1}(\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu}) + \boldsymbol{\mu}^T\mathbf{A}\boldsymbol{\mu} \tag{30}$$

Substituting Eqs. (29) and (30) into Eq. (28), we have

$$L(\boldsymbol{\alpha}, \textstyle\sum_{\sigma^2}) = -\frac{1}{2}mln2\pi - \frac{1}{2}ln|\mathbf{\Sigma}^{-1}| + \frac{1}{2}ln\left|\Sigma_{\sigma^2}^{-1}\right| + \frac{1}{2}ln|\mathbf{A}| - \frac{1}{2}(\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu})^T\Sigma_{\sigma^2}^{-1}(\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu}) - \frac{1}{2}\boldsymbol{\mu}^T\mathbf{A}\boldsymbol{\mu} \tag{31}$$

The derivative of $L(\boldsymbol{\alpha}, \Sigma_{\sigma^2})$ with respect to $\boldsymbol{\alpha}$ and $\Sigma_{\sigma^2}$is derived and then is set to zero. The details of the derivation is presented in the Appendix I. The estimate of each element of $\boldsymbol{\alpha}$ and the noise variance is presented as follows:

$$\boldsymbol{\alpha}_{new\ i} = \frac{1-\boldsymbol{\alpha}_i \Sigma_{ii}}{\boldsymbol{\mu}_i{}^2} \tag{32}$$

$$\sigma_j{}^2 = \frac{(\mathbf{t}_j - \boldsymbol{\Phi}_j \boldsymbol{\mu})^2}{1 - Tr\left(\boldsymbol{\Sigma}\boldsymbol{\Phi}^T J_j\ \boldsymbol{\Phi}\right)} \tag{33}$$

where $\sum_{ii}$ is the i-th diagonal element of the covariance matrix $\boldsymbol{\Sigma}$ (Eq. (25)) of the posterior weight $\mathbf{w}$, J_j is an $m \times m$ matrix that all its elements are zero except the j-th diagonal element which is one, $\mathbf{t}_j$ is the j-th element of column vector $\mathbf{t}$, and $\boldsymbol{\Phi}_j$ is the j-th row of $\boldsymbol{\Phi}$ matrix

From Eqs. (32) and (33), it can be seen that although $\boldsymbol{\alpha}$ and $\sigma_j{}^2$ can be estimated, they depend on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, in Eqs. (25) and (26), which, however, depend on $\boldsymbol{\alpha}$ and $\sigma_j{}^2$. Thus, an iterative method will be presented in the next section to iteratively estimate all these variables.

3.4.2.3 Iterative Estimate of the Parameters

In Sec. 3.4.2.1 and 3.4.2.2, it can be seen that the estimate of $\boldsymbol{\alpha}$ and $\boldsymbol{B}$ are the functions of posterior mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of $\mathbf{w}$, and vice versa. Thus, Σ and μ Eqs. (25) and (26)) can be estimated by updating them iteratively through estimate of $\alpha_{new\ i}$ and $\sigma_{new\ j}{}^2$ (Eqs. (32) and (33)) until a convergence criteria is satisfied (e.g. the difference between two consecutive iterations is less than a predefined threshold). The enhanced RVM method after a certain number of iterations will converge to the sparse solution for the linear regression model presented in Eq. (20). The iterative approach presented in this section continues until the convergence criteria is achieved. Indeed, convergence of the iterative algorithm depends upon whether the optimal value for the cost function, i.e. log of the marginal likelihood presented in Eq. (28) is reached. The cost function is a multivariate Gaussian distribution which for sure has a maximum. Thus, taking the first derivation of the cost function with respect to α and the noise covariance and setting them to zero guarantee that the cost function increases at each step and moves toward the maximum. Hence, the iterative approach guarantees the convergence to the optimal value of the cost function. The convergence can be reached through setting a proper threshold that the difference of the all values of $\boldsymbol{\alpha}$ corresponding to the last two consecutive iterations is very small. However, there is no particular rule to set the set the threshold as the stopping criteria in the enhanced RVM method. Apparently when the difference of the two consecutive iterations of the algorithm is small enough we can terminate. Here, as we are looking for the local maxima of Eq. (28), thus, when the relative difference between all the values of $\boldsymbol{\alpha}_{\mathrm{i}}$ of two consecutive iterations is fairly small, we may terminate the iterative process. It means that there is no further relevant coefficient vector can be determined since the change in the values of $\boldsymbol{\alpha}_{\mathrm{i}}$'s is negligible and we have reached the local maxima. Thus, we set the threshold as if the maximum relative difference of the two consecutive iterations of $\boldsymbol{\alpha}_{\mathrm{i}}$'s is less than 0.1% (0.001), i.e. $\max \frac{|\boldsymbol{\alpha}_{\mathrm{k}}-\boldsymbol{\alpha}_{\mathrm{k-1}}|}{\boldsymbol{\alpha}_{\mathrm{k-1}}} \leq 10^{-3}$.

In practice, it can be seen that many of $\boldsymbol{\alpha}_i$ are driven to very large values causing the variances to become zero. Thus, if the posterior probability $p(\mathbf{w}_i|\mathbf{t}, \boldsymbol{\alpha}, \Sigma_{\sigma^2})$ of the corresponding $\mathbf{w}_i$, becomes highly peaked at zero, it means that the corresponding $\mathbf{w}_i$ can be effectively pruned out and the sparsity over $\mathbf{w}$ is realized. Thus, utilizing enhanced RVM, the sparse estimate of $\mathbf{w}$ is achieved whose non-zero values related to their corresponding process errors are the process faults (i.e. $\boldsymbol{\mu}$). Also the estimate of the variance components of the noise term is provided with ${\sigma_{new\,j}}^2$ for $j = 1, \dots, m$.

3.5 Case Studies

An assembly model from a real auto body assembly process is utilized to validate the proposed methodology. As shown in Figure 3-4, the assembled product is a floor pan which is under the driver and the passenger in a car. It has four sub-parts, namely, left floor pan, right floor pan, left bracket, and right bracket. They are assembled in three stations as illustrated in Figure 3-5.

Measurements will be taken at the end of the assembly line. The 4 points M1-M4 as shown in Fig. 4 are measured in X, Y, and Z directions, respectively. Thus, there are a total of 12 measurements (KPCs). The process errors (KCCs) in this case are fixture errors and part mating feature errors (also playing a role as fixture for the part to be assembled). In this assembly process, there are a total of 33 process errors. Therefore, the fault pattern matrix **Γ** in Eq. (8) has dimension of 12 by 33. These 33 KCCs are assumed to have normal distributions. The design specifications of their variability are given in terms of tolerance range which is defined as 6 times of the standard deviation of the process error (6-sigma), such as fixture locator variations.

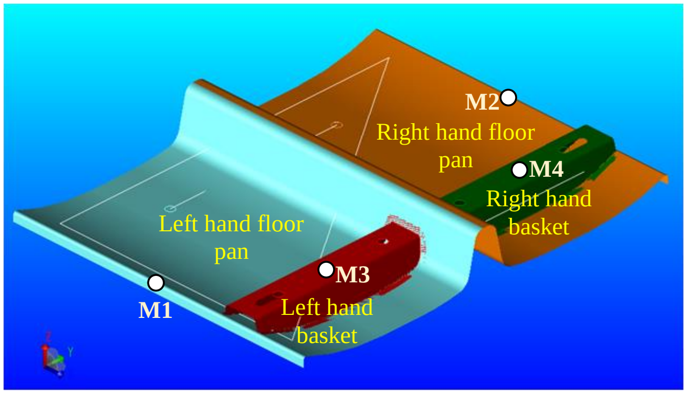

Figure 3-4. The 4-part floor pan assembly model

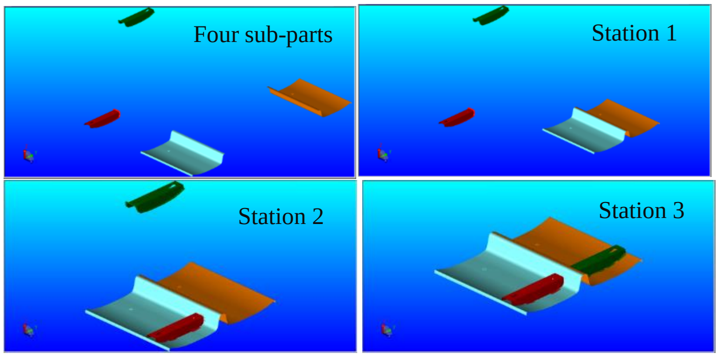

Figure 3-5. Floor-pan assembly in three assembly stations

For KCC1 to KCC30, tolerance ranges are given as 0.5 mm, and KCC31 to KCC33 as 1 mm. By running the Monte Carlo simulation using 3DCSTM software, the assembly process can be accurately simulated. 3DCS has been the leading variation analysis software in this field. The capabilities of 3DCS have been tested and validated by many industrial applications and for this reason we chose 3DCS as yardstick for validating the results from the proposed method in this subchapter.

We simulated the assembly process with the intended process faults which are manipulated in the way that the tolerance ranges of KCC8 to KCC13 are set as 1 mm (beyond their design specification, i.e., 0.5 mm). The number of samples for the simulation is set at 400. After the simulation, the reading of the coordinates of the 12 KPCs of these 400 samples is collected, and they will provide the measurement information Y. Based on the methodology developed in this subchapter, the following two sections demonstrate the fault diagnosis to identity the process faults (KCC8 to KCC13) for two scenarios, namely, with complete fault pattern matrix $\mathbf{\Gamma}$ (Sec. 3.5.1), and partial matrix $\mathbf{\Gamma}$ (Sec. 3.5.2), respectively.

3.5.1 Fault Diagnosis with a Completely Available Fault Pattern Matrix Γ

In this section, we assume that all the process and product information is given. Therefore, the complete fault pattern matrix $\mathbf{\Gamma}$ of the assembly model can be computed and given for fault diagnosis. For this assembly model, matrix $\mathbf{\Gamma}$ in Eq. (8) has a dimension of 12 by 33, corresponding to 12 KPCs and 33 KCCs. By applying the methods presented in our prior research [52, 53, 56], the state space model is built up to represent error propagation in multi-station assembly. The resulting matrix Γ can be obtained. The measurements on the 12 KPCs will provide 12*(12+1)/2=78 variances/covariances; Thus, matrix $\mathbf{\Phi}$ in Eq. (20) is of the dimension of 78 by 33. Since the rank of $\mathbf{\Phi}$ is 25, which is less than 33 (number of KCCs), essentially the problem is still an undetermined system.

Sect. 3.5.2 develops the argument that the posterior mean of **w** that represents the change of the variance of the process errors is iteratively estimated. This iterative estimation continues until the stopping criteria is satisfied during which most of $\boldsymbol{\alpha}_\mathrm{i}$ are driven to very large values, indicating very small variance of the estimated quantity (change of variance). The initial value for all the $\boldsymbol{\alpha}_\mathrm{i}$s is set to $\frac{1}{\mathrm{n}^2} = 9.1827 \times 10^{-4}$, where n=33 (number of KCCs) for this case study. For $\sigma_\mathrm{j}{}^2$ in Eq. (26), we set the initial value to be the variance of vector $\mathbf{t}$. By applying the method presented in Sec. 3.4.2.3, Eqs. (25) and (26) can be iteratively solved with Eqs. (32) and (33). For the purpose of fault diagnosis, we are interested in the mean value $\mathbf{w}$ (Eq. (25)), as well as the variance (inverse of $\boldsymbol{\alpha}$ value (Eq. (32)).

Table 3-1. The iterative computation for alpha values

	Iteration 1	Iteration 5	Iteration 9	Iteration 13	Iteration 17	Iteration 20	Iteration 21
KCC 1	9.182E-04	2.35E+04	2.35E+04	2.35E+04	2.35E+04	2.35E+04	2.35E+04
KCC 2	9.182E-04	2.23E+05	2.23E+05	2.23E+05	2.23E+05	2.23E+05	2.23E+05
……							
KCC 8	9.182E-04	2.32E+03	2.33E+03	2.33E+03	2.31E+03	2.33E+03	***2.33E+03***
KCC 9	9.182E-04	8.56E+02	8.58E+02	8.58E+02	8.73E+02	8.58E+02	***8.58E+02***
KCC 10	9.182E-04	1.65E+03	2.29E+03	2.30E+03	2.31E+03	2.30E+03	***2.30E+03***
KCC 11	9.182E-04	4.45E+03	3.78E+03	3.78E+03	3.86E+03	3.78E+03	***3.78E+03***
KCC 12	9.182E-04	8.57E+02	8.56E+02	8.56E+02	8.54E+02	8.56E+02	***8.56E+02***
KCC 13	9.182E-04	8.57E+02	8.56E+02	8.56E+02	8.54E+02	8.56E+02	***8.56E+02***
……							
KCC 32	9.182E-04	4.14E+05	4.14E+05	4.14E+05	4.14E+05	4.14E+05	4.14E+05
KCC 33	9.182E-04	1.12E+05	1.12E+05	1.12E+05	1.12E+05	1.12E+05	1.12E+05

Table 3-2. The iterative computation for mean value of KCCs

	Iteration 1	Iteration 5	Iteration 9	Iteration 13	Iteration 17	Iteration 20	Iteration 21
KCC1	3.75E-05	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
KCC2	7.24E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
……							
KCC8	1.45E-03	2.07E-02	2.07E-02	2.07E-02	2.08E-02	***2.07E-02***	***2.07E-02***
KCC9	3.03E-04	3.22E-02	3.01E-02	2.80E-02	2.67E-02	***2.62E-02***	***2.62E-02***
KCC10	4.81E-04	2.39E-02	2.09E-02	2.09E-02	2.08E-02	***2.09E-02***	***2.09E-02***
KCC11	1.65E-04	1.32E-02	1.73E-02	1.83E-02	1.81E-02	***1.83E-02***	***1.83E-02***
KCC12	8.87E-04	2.41E-02	2.21E-02	2.21E-02	2.19E-02	***2.19E-02***	***2.19E-02***
KCC13	8.87E-04	2.41E-02	2.32E-02	2.22E-02	2.19E-02	***2.19E-02***	***2.19E-02***
……							
KCC32	2.02E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
KCC33	7.77E-04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

The iterative results are shown in Table 3-1 and Table 3-2. From Table 3-1, it can be seen that the difference of $\boldsymbol{\alpha}$ value between iterations 20 and 21 is much smaller than the predefined value. Thus, the iteration is terminated. As presented in Table 3-2, the computation for the mean value of w is also terminated at iteration 21. In Table 3-1, the $\boldsymbol{\alpha}$ values are quite large for the KCCs 1-7 and KCCs 14-33, indicating they all have very small variances. In Table 3-2 the above KCCs have mean value of zeros. Thus, the values of these KCCs are concentrated about zero with very small variance. Then they can be reasonably considered as zeros because there is no change of variance or process faults. For KCCs 8-13, their values are not very big, indicating that the variances are relatively large. The corresponding mean values are nonzero, which can be considered as nonzero changes of variance of the corresponding KCCs, namely, process faults. Table 3-3 shows the identified nonzero change of variance for KCCs 8-13. From the small relative errors between the estimated tolerance range and actual tolerance range of these KCCs, the effectiveness of our process method can be clearly demonstrated. For the practitioner, the comparison between the estimated tolerance ranges to the corresponding design specifications can be used to determine if there are any process faults. For more rigorous statistical analysis, since α value is the inverse variance of the variance change (Δ_i), a statistical analysis for the confidence of determining process faults ($\Delta_i > 0$ in a statistical sense) can be performed, as shown in Table 3-3, of which the corresponding confidences are listed in the last column.

Table 3-3. Case study 1: identified process faults and the estimate of their variance change and 6-sigma value (unit in mm)

Identified process faults	Tolerance range (6-sigma) of design spec	Estimate of variance change (Δ_i)	Estimated variance of Δ_i (1/α)	Estimated tolerance range (6-sigma) value ($\hat{R}$)	Actual tolerance range (6-sigma) value (R)	Relative* errors (%)	Confidence of the process faults (Δ_i> 0)
8	0.500	0.021	4.29E-04	0.998	1.000	0.239	84%
9	0.500	0.026	1.17E-03	1.093	1.000	-9.303	78%
10	0.500	0.021	4.35E-04	1.001	1.000	-0.048	84%
11	0.500	0.018	2.65E-04	0.953	1.000	4.734	87%
12	0.500	0.022	1.17E-03	1.019	1.000	-1.907	74%
13	0.500	0.022	1.17E-03	1.019	1.000	-1.903	74%

*Relative errors = $\frac{|(R-R)|}{R}$

3.5.2 Fault Diagnosis with a Partially Given Fault Pattern Matrix Γ

In this section, we will demonstrate a case study where if only partial fault pattern matrix **Γ** in Eq. (8) is available, our proposed methodology can still estimate the variance change of the corresponding KCCs, and identify process faults due to our enhanced RVM's ability to handle inhomogeneous noise. If only partial matrix **Γ** is known, then Eq. (8) can be written as $\mathbf{y} = [\boldsymbol{\Gamma}_{known}\ \boldsymbol{\Gamma}_{unknown}].\mathbf{u} + \boldsymbol{\varepsilon}$. Thus, by combining process variation contributed from the unknown gamma matrix, we can have the form of $\mathbf{y} = [\Gamma_{known}].\mathbf{u}_{known} + \boldsymbol{\varepsilon}_{new}$ where $\boldsymbol{\varepsilon}_{new} = \boldsymbol{\Gamma}_{unknown}.\mathbf{u}_{unknown} + \boldsymbol{\varepsilon}$ can be considered the new noise term, including the un-model process errors.

In this case study, we assume that in the fault pattern matrix **Γ**, columns 2, 9, 25, 27, and 31 are unknown due to some limited process/product information. By using Eq. (16), the assumption of independent noise has been relaxed. Thus, with our proposed method, the fault diagnosis can still be carried out with partially known matrix **Γ**. From Table 3-4, it can be seen that even with a partially known fault pattern matrix **Γ**, our proposed method can still identify the process faults with very small relative error between the estimated tolerance ranges and actually tolerance ranges of KCCs 8-13, and a reasonably high confidence of identified process faults. Compared with the case study presented in Sec. 3.5.1, the identified process faults do not include KCC9. This is because the column 9 in matrix $\mathbf{\Gamma}$ corresponding to KCC9 is unknown in this case; thus it is impossible to estimate it. This case study demonstrated the effectiveness of our method even with partially available process/product information.

Table 3-4. Case study 2: Identified process faults and the estimate of their variance change and 6-sigma

Identified process faults	Tolerance range (6-sigma) of design spec	Estimate of variance change (Δ_i)	Estimated variance of Δ_i (1/α)	Estimated tolerance range (6-sigma) value	Actual tolerance range (6-sigma) value	Relative errors (%)	Confidence of the process faults
8	0.500	0.0229	0.00052	1.037	1.000	3.653	84%
10	0.500	0.0243	0.00059	1.061	1.000	6.057	84%
11	0.500	0.0229	0.00052	1.037	1.000	3.653	84%
12	0.500	0.0199	0.00083	0.983	1.000	-1.694	93%
13	0.500	0.0204	0.00089	0.992	1.000	-0.783	93%

3.6 Concluding Remarks

The purpose of fault diagnosis for multi-station assembly process is to effectively identify the root causes of large dimensional variations on the product, a critical component to ensure the overall quality of assembled products. Although much research has been explored in this direction, little research has focused on the partially diagnosable system. This is caused by lack of measurement as opposed to the number of process errors in the assembly process, and essentially is an underdetermined system and thus has no unique solution but with infinite ones. In this subchapter, a fault diagnosis methodology for multi-station assembly processes was developed to tackle the above challenge. The proposed methodology considers the process faults as change (increase) of variance of process errors (KCCs). An enhanced RVM methodology was established by relaxing the assumption of iid noise in the existing RVM method. Thus, it can solve the underdetermined system with sparse solution, which is consistent to the engineering interpretation. Moreover, due to our new development, the enhanced RVM is capable of handling cases where only partial process/product information is available, i.e., only partial columns of the fault pattern matrix are known. Using this methodology the process faults can still be identified effectively.

The provided case studies demonstrate the effectiveness of the proposed methodology. The cases of complete fault pattern matrix (case 1 in Sec. 3.5.1) as well as partially available fault pattern matrix (case 2 in Sec.3.5.2) are investigated. For both cases, satisfactory results are achieved. The relative errors between estimated tolerance ranges using the proposed method and the actual tolerance ranges are between 0.239% and 9.303% for case 1; and for case 2, the relative errors are between 0.783% and 6.057%. The confidence levels for determining the process faults are also reasonable, between 74% and 87% for case 1, and between 84% and 93% for case 2. These results demonstrate the effectiveness of our proposed methodology, which not only fills the gap in the research domain, but also provides effective tools for the practitioner of quality improvement in large and complex manufacturing systems.

Associated with the proposed RVM methodology for fault diagnosis, there is a very critical issue that should be investigated, namely, how to characterize the type of sensor system that will make the proposed RVM method work. This is called diagnosability study, a method to investigate the sensor placement, including determination of number of sensors as well as their location, by which the process faults can be uniquely identified based on the measurements provided by the optimal sensor layout. Zhou et al. [24] defined a diagnosability matrix which is determined by state matrix, input matrix and observation matrix, namely, matrices $\mathbf{A}_{\mathrm{i}}$, $\mathbf{B}_{\mathbf{i}}$ and $\mathbf{C}_{\mathrm{i}}$ in Eq.(9.1). They proved that the diagnosis problem of process faults with variance change is fully diagnosable if the diagnosability matrix is full rank. Since observation matrix $\mathbf{C}_{\mathrm{i}}$ is determined by sensor placement, we can optimize the diagnosability matrix, namely, to maximize its rank, by changing the number and location of sensors. Hence, the desired level of diagnosability can be achieved through optimal sensor placement, and full diagnosability can be achieved if the diagnosability matrix is of full rank. The work reported by Liu, et al. [57] is along this line.

The diagnosability study for underdetermined systems considered in this subchapter is even more challenging since a full rank of the diagnosability matrix is impossible for this case. It still deals with the optimal sensor placement, but the objective is to make a unique sparse solution achievable. This type of diagnosability is different from the one in Zhou, et al. [24] in which the diagnosability is independent of the fault diagnosis methods used. The diagnosability for underdetermined systems is indeed dependent upon the specific fault diagnosis method utilized since different diagnosis methods may have different levels of capabilities to identify the unique sparse solution. In the future, we are going to research optimal sensor placement for underdetermined systems by using the enhanced RVM method proposed in this subchapter. With this study, we are able to characterize what kind of system (necessary number of sensors as well as the sensor location) is diagnosable by using the enhanced RVM method.

Chapter 4. Sensor Placement Optimization for Multi-station Assembly Processes

Developments in sensing technologies have created opportunities to diagnose the process faults in multi-station assembly processes by analyzing the measurement data. Sufficient diagnosability for process faults has been always a challenging issue since the sensors cannot be used excessively. Therefore, there have been a number of methods reported in the literature to optimize the diagnosability of the diagnostic method with given sensor costs for identification of process faults incurred in multi-station assembly processes. However, most of these methods assume that the number of sensors is more than that of the process errors. Unfortunately, this assumption may not hold in many real industrial applications. Thus, the diagnostic methods have to solve underdetermined linear equations. In order to address this issue, we propose an optimal sensor placement method by devising a new diagnosability criterion based on the compressive sensing theory which is able to handle underdetermined linear equations. Our method seeks optimal sensor placement by minimizing the average mutual coherence to maximize the diagnosability. The proposed method is demonstrated and validated through case studies from actual industrial applications[2].

4.1 Introduction and Related Work

Multi-station assembly processes involve a large number of operations in multiple workstations to assemble a final product. Some typical examples of products from multi-station assembly processes include automotive body assembly, aerospace industry, home appliance assembly, etc.

[2] Bastani, K., Kong, Z. Huang, W., and Zhou, Y., 2015, "Compressive sensing based optimal sensor placement for multi-station assembly processes," *IIE Transactions* (in press). DOI: 10.1080/0740817X.2015.1096431.

The dimensional quality of the final product in multi-station assembly processes is affected by process errors, which are also called key control characteristics (KCCs). One example of KCCs is fixture locators that play a critical role in dimensional quality of the assembled product. Fixture locators are used to hold and position the parts during assemblies. Hence, dimensional variation of fixture locators directly causes quality issues of the final products.

To achieve satisfactory dimensional quality of the final product, it is necessary to identify the root causes of large dimensional discrepancies of key product characteristics (KPCs). Fault diagnosis is such a task to ensure a final product with satisfactory dimensional quality standards. There is a large body of literature discussing fault diagnosis for multi-station assembly processes [32, 58-61]. Most of these methods assumed a linear model that links the dimensional deviations of the KCCs to the measurements on KPCs, as follows,

$$\mathbf{y} = \mathbf{\Gamma} \cdot \mathbf{u} + \boldsymbol{\varepsilon} \tag{34}$$

where $\mathbf{y}$ is an $m \times 1$ vector that represents the dimensional deviation of KPCs (taken by measurements); $\mathbf{u}$ is an $n \times 1$ vector that represents the deviations of the process errors (KCCs); $\mathbf{\Gamma}$ is a fault pattern matrix ($m \times n$) obtained from product and process information; $\boldsymbol{\varepsilon}$ denotes the noise (a combination of modeling uncertainty and measurement noise); and m and n are the numbers of measurements and process errors, respectively.

A clarification for the terminologies is as follows. Process errors ($\mathbf{u}$) refer to the variation of all KCCs. They always exist in the assembly process and their specification limits are determined in design process. Process errors are different from process faults. Process faults refer to the process errors whose variations exceed the design specifications. Process faults are typically manifested through the mean shifts and change of the variance. Most of the fault diagnosis methods deal with the variance estimate of the process errors. This is because identifying and eliminating the process faults caused by large variance of KCCs is more challenging than mean shift issues. Therefore, in this subchapter, the fault diagnosis refers to the variance estimate of the process errors.

As shown in Eq. (34), to effectively diagnose the process faults, measurements on KPCs, i.e., **y**, must be taken along the assembly operations. Coordinate measuring machines (CMM) and optical coordinate measuring machines (OCMM) are the typical sensing systems that have been used to provide the dimensional measurements on KPCs in multi-station assembly processes. The proper design of the sensing system is vital for effective measurements of fault diagnosis. The diagnosability for the process faults and the incurred costs for sensors are the main issues in designing an effective sensing system.

Most of the existing research in this direction [57, 62-67] basically considers full diagnosability of a sensing system. A fully diagnosable system is defined as a system that the measurements on the KPCs provide enough information for diagnosis of all process faults. Namely, the variance of all process errors can be uniquely estimated through analysis of the measurements on KPCs [24]. As reported in Ref. [24, 68], a diagnosability matrix is defined based on matrix $\mathbf{\Gamma}$ (Eq. (34)) and the system has full diagnosability if the diagnosability matrix is of full rank since the associated linear equations have a unique solution. Based on the defined diagnosability criteria, some research work [57, 62, 63, 67] developed optimal sensor placement methods to achieve full diagnosability with given sensors.

By taking into consideration of industry practice in which the number of measurements is often less than that of process errors, Zhou et al. [13] extended the diagnosability study in Ref. [68] by studying partially diagnosable systems, which is caused by column rank deficiency of matrix $\mathbf{\Gamma}$ in Eq. (34). Their method identified the process errors that are uniquely diagnosable, as well as the minimal diagnosable sets. The minimal diagnosable set is an interesting concept, which was defined as a group of process errors that are not individually uniquely diagnosable, but their linear combination can be uniquely estimated. By definition, within a minimal diagnosable set, each process error is linearly dependent with others. Due to the rank deficiency induced by this linear dependency, it is impossible to uniquely identify individual process errors in a minimal diagnosable set.

In general, rank deficiency of matrix $\mathbf{\Gamma}$ makes Eq. (34) (or its variant model which is later on utilized for variance change fault diagnosis in this subchapter) become underdetermined linear equations, for which there are infinitely large number of solutions [69]. However, this issue can be tackled by sparsity assumption of process faults. The sparsity assumption is in compliance with multi-station assembly processes since the number of process faults is usually few. This can be justified as the probability of having less process faults is higher than that of having more process faults, assuming the process faults are independent of each other. Following the results of Compressive Sensing theory [70, 71], if certain properties of matrix $\mathbf{\Gamma}$ (e.g., formed by the minimal diagnosable set) are satisfied, unique estimate of the process faults can be achieved with a high probability. Therefore, we propose a compressive sensing based diagnosability criterion to tackle the challenges in diagnosability for underdetermined linear systems. Subsequently, an optimal sensor placement method is developed in this subchapter based upon the proposed diagnosability criterion to improve the diagnosability of multi-station assembly processes.

The rest of the Chapter is organized as follows. Sec. 4.2 introduces overall procedure of the proposed method and highlights the contributions. In Sec. 4.3, the variation propagation model for multi-station assembly processes, along with the vectorization of the variance model, is presented. Sec. 4.4 proposes the diagnosability of multi-station assembly processes with underdetermined systems, based upon which our optimal sensor placement method is proposed. In Sec. 4.5, case studies from industrial application are provided to demonstrate the effectiveness of the proposed method. Finally, Sec. 4.6 summarizes the major findings in this study.

4.2 The Proposed Research Methodology and Contributions

Figure 4-1 shows the procedure of our proposed method for optimal sensor placement for multi-station assembly processes. The contributions of the proposed sensor placement method include:

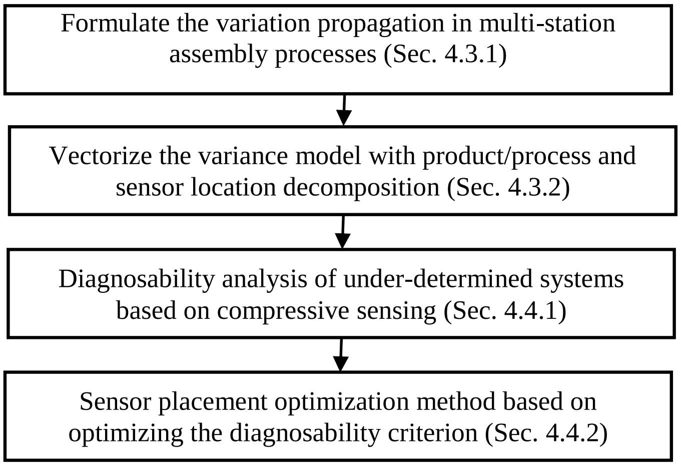
Formulate the variation propagation in multi-station assembly processes (Sec. 4.3.1)
Vectorize the variance model with product/process and sensor location decomposition (Sec. 4.3.2)
Diagnosability analysis of under-determined systems based on compressive sensing (Sec. 4.4.1)
Sensor placement optimization method based on optimizing the diagnosability criterion (Sec. 4.4.2)

Figure 4-1. Procedure of the proposed sensor placement method

(1) we decompose the fault pattern matrix $\mathbf{\Gamma}$ in Eq. (34) into two parts, corresponding to sensor placement and process/product information, respectively. This decomposition enables us to apply compressive sensing theory to study the diagnosability, and (2) we devise a new diagnosability criterion as well as an optimal sensor placement for multi-station assembly processes with underdetermined systems, which has not been well addressed in existing literature.

4.3 Variation Propagation Model in Multi-Station Assembly Processes

In this section, we first introduce the variation propagation model in multi-station assembly processes using state-space modeling (Sec. 4.3.1). Then we derive a vectorized variance model (Sec. 4.3.2) which is used for the diagnosability analysis in Sec. 4.4.

4.3.1 State-space Modeling for Variation Propagation in Multi-Station Assembly Processes

There have been a large body of research work in the literature that utilizes the state-space models to formulate the variation propagation in multi-station assembly processes [72-75]. A brief introduction of the variation propagation model in multi-station assembly processes is presented here. For details, please refer to Ref. [75]. The multi-station assembly process illustrated in Figure 4-2 [59] has M stations where variable i is the station index. State vector $\mathbf{x}_i$ denotes the product quality information (e.g., part dimensional deviations) at station i; $\mathbf{u}_i \in R^{n_i}$ is the input variable representing the process errors at station i (such as fixture deviations) with cardinality $|\mathbf{u}_i| = n_i$, and $\mathbf{w}_i$ represents the process noise and unmodeled errors. The measurements on KPCs at station i are represented by $\mathbf{y}_i \in R^{m_i}$ with cardinality $|\mathbf{y}_i| = m_i$ (it is not necessarily available in every station for which $m_i = 0$). The random vector $\mathbf{v}_i$ is used to model the measurement noise. Note that n_i and m_i denote the number of process errors and measurements at station i, respectively.

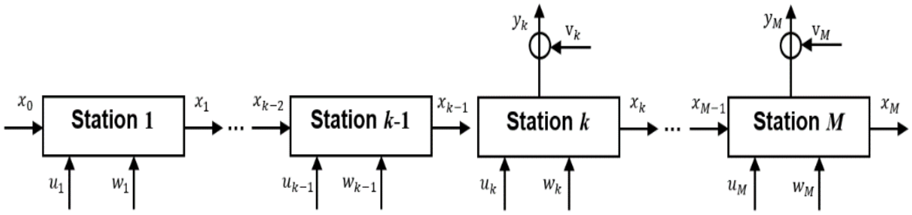

Figure 4-2. Diagram of a multi-station assembly process [3]

The components of $\mathbf{u}_i$ are assumed independent of each other. $\mathbf{w}_i$ and $\mathbf{v}_i$ are assumed to be zero mean and all of their elements are independent of each other. Also, it is assumed that the elements of $\mathbf{u}_i$, $\mathbf{w}_i$ and $\mathbf{v}_i$ are all independent of each other, and the magnitude of $\mathbf{w}_i$ is small compared to that of $\mathbf{x}_i$ and $\mathbf{u}_i$. Considering all these assumptions together, the state-space model is expressed as follows:

$$\mathbf{x}_i = \mathbf{A}_{i-1}\mathbf{x}_{i-1} + \mathbf{B}_i\mathbf{u}_i + \mathbf{w}_i \quad i = 1, 2, \dots, M \tag{35}$$

$$\mathbf{y}_i = \mathbf{C}_i\mathbf{x}_i + \mathbf{v}_i \quad\quad\quad i = 1, 2, \dots, M \tag{36}$$

where $\mathbf{x}_0$ represents the initial condition of state-space vector such as the fabrication imperfection of the parts before entering the assembly process (assumed to be known). The interpretation of matrices $\mathbf{A}_i, \mathbf{B}_i$ and $\mathbf{C}_i$ is summarized in Table 4-1. For details of the formulation of these matrices, please refer to Ref. [75].

Table 4-1. Interpretation of the matrices in Eqs. (35) and (36)

Matrix type	**Interpretation**
State matrix ($\mathbf{A}_i$)	Change of fixture locating layout between stations *i* and *i*-1
Input matrix ($\mathbf{B}_i$)	Fixture locator positions at station *i*
Observation matrix ($\mathbf{C}_i$)	Sensor placement at station *i*

Based on Eqs. (35) and (36), the linear model of Eq. (34) is achieved, and rewritten as Eq. (37) for convenience,

$$\mathbf{y} = \mathbf{\Gamma} \cdot \mathbf{u} + \boldsymbol{\varepsilon} \tag{37}$$

where $\mathbf{y} = [\mathbf{y}_1{}^T\ \mathbf{y}_2{}^T \ldots \mathbf{y}_M{}^T]^T$ is an $m \times 1$ vector which denotes the $m = \sum_{i=1}^{M} m_i$ measurements from station 1 through station M, $\mathbf{u} = [\mathbf{u}_1{}^T\ \mathbf{u}_2{}^T \ldots \mathbf{u}_M{}^T]^T$ is an $n \times 1$ column vector representing $n = \sum_{i=1}^{M} n_i$ process errors (in this subchapter we consider fixture locator deviations and the part mating feature deviation); $\mathbf{\Gamma}$ is the fault pattern matrix with dimension $m \times n$ as follows,

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{C}_1\mathbf{B}_1 & 0 & \ldots & 0 \\ \mathbf{C}_2\boldsymbol{\Phi}_{2,1}\mathbf{B}_1 & \mathbf{C}_2\mathbf{B}_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_M\boldsymbol{\Phi}_{M,1}\mathbf{B}_1 & \mathbf{C}_M\boldsymbol{\Phi}_{M,2}B_2 & \ldots & \mathbf{C}_M\mathbf{B}_M \end{bmatrix} \tag{38}$$

where $\boldsymbol{\Phi}_{j,i} = \mathbf{A}_{j-1}\mathbf{A}_{j-2} \ldots A_i$ for $j > i$ and $\boldsymbol{\Phi}_{i,i} = I$ (we refer readers to Ref. [14] for more details on derivation of matrix $\mathbf{\Gamma}$), and $\boldsymbol{\varepsilon}$ is an $m \times 1$ vector representing the noise term which is a combination of the process noise ($\mathbf{w}_i$) and the measurement noise ($\mathbf{v}_i$) along all the stations in the assembly process. The linear model in Eq. (37) is called the deviation model and presents the effects of the deviation of the KCCs on the deviation of the measurements on KPCs.

Based on Eq. (37), the corresponding variance model is

$$cov(\mathbf{y}) = \mathbf{\Gamma} \cdot cov(\mathbf{u}) \cdot \mathbf{\Gamma}^{\mathrm{T}} + cov(\boldsymbol{\varepsilon}) \tag{39.a}$$

where $cov(\cdot)$ is the covariance matrix of a random vector. Since all the elements of $\mathbf{u}$ are assumed to be independent of each other, $cov(\mathbf{u})$ is a diagonal matrix. As $\mathbf{u}$ is a vector that contains all the KCCs from station 1 through station M, we can represent the covariance matrix of $\mathbf{u}$ as

$$cov(\mathbf{u}) = Diag[cov(\mathbf{u}_1), cov(\mathbf{u}_2), \dots cov(\mathbf{u}_M)] \tag{39.b}$$

where $cov(\mathbf{u}_i)$= $diag\left({\sigma_{1i}}^2, {\sigma_{2i}}^2, \dots, {\sigma_{n_i i}}^2\right)$, ${\sigma_{ji}}^2$ is variance of the j-th KCC at station i, and n_i is the number of KCC's at station i.

4.3.2 Vectorization of the Variance Model in Multi-Station Assembly Processes

In this section, we derive the vectorized variance model for Eq. (39.a). This can be carried out with vectorization operator $Vec(.)$ that transforms a matrix into a vector. If we have an $p \times q$ matrix $\mathbf{Z} = \left[\mathbf{z}_1\, \mathbf{z}_2 \cdots \mathbf{z}_q\right]$ where $\mathbf{z}_i \in R^p$ represents the i-th column of the matrix, $Vec(\mathbf{Z})$ is given as below,

$$Vec(\mathbf{Z}) = [\mathbf{z}_1^T, \mathbf{z}_2^T, \dots, \mathbf{z}_q^T]^T \tag{40}$$

which is an $pq \times 1$ vector. Using $Vec(.)$ on a $q \times q$ symmetric matrix (e.g. $\mathbf{Z}$ is a covariance matrix) yields a vector with dimension $\frac{q(q+1)}{2}$ because the redundant elements in the symmetric matrix can be eliminated.

Vectorization of the variance model provides a linear model where the vectorized $cov(\mathbf{u})$ in Eq. (39.a) denotes the unknown variables (variance of process errors to be estimated), the vectorized

$cov(\mathbf{y})$ in Eq. (39.a) is the known variables (from measurements), and the resulting design matrix is a function of fault pattern matrix $\mathbf{\Gamma}$. We next present Proposition 1 to decompose the design matrix into two parts, i.e., (1) product/process and (2) sensor location information. This formulation enables us to optimize diagnosability of the diagnostic method by using compressive sensing principles [15].

Proposition 1: Vectorized variance model of Eq. (6.a) can be formulated as a linear model

$$\mathbf{t} = \mathbf{\Lambda\Omega b} + \boldsymbol{\xi} \tag{41}$$

where vector $\mathbf{t}$ is from sensor measurements; matrix $\mathbf{\Lambda}$ (the measurement matrix) is only determined by sensor placement, i.e., observation matrix $\mathbf{C}_i$; matrix $\mathbf{\Omega}$ is only determined by the product and process information, i.e., matrices $\mathbf{A}_i$ and $\mathbf{B}_i$; vector $\mathbf{b}$ denotes the variance change of the process errors; and vector $\boldsymbol{\xi}$ represents the variance and covariance elements of the noise term $\boldsymbol{\varepsilon}$. See Table 4-1 for interpretation of matrices $\mathbf{A}_i$, $\mathbf{B}_i$, and $\mathbf{C}_i$.

Proof: Proof is provided in Appendix II.

Recall that the process faults are interpreted as the process errors which have large variance (i.e., beyond their design specification). Thus, vector $\mathbf{b}$ in Eq. (41) includes two parts, i.e., (1) the design specification limit on variance of the process errors, and (2) the change of variance, namely,

$$\mathbf{b} = \boldsymbol{\delta} + \boldsymbol{\sigma} \tag{42}$$

where $\boldsymbol{\sigma} = [{\sigma_{1d}}^2 \ {\sigma_{2d}}^2 \ ... \ {\sigma_{nd}}^2]$ is the design specifications of the *n* process errors in terms of variance, and $\boldsymbol{\delta} = [\boldsymbol{\delta}_1 \ \boldsymbol{\delta}_2 \ ... \ \boldsymbol{\delta}_n]$ denotes the variance change of the *n* process errors. Now, the vectorized variance model in Eq. (41) can be formulated as

$$\mathbf{t} = \mathbf{\Lambda\Omega}(\boldsymbol{\delta} + \boldsymbol{\sigma}) + \boldsymbol{\xi} \tag{43}$$

Since quality of the product usually deteriorates over the time, it is reasonable to assume $\boldsymbol{\delta}_i \geq 0$ $(\forall i = 1, \dots, n)$. For example, a process fault occurs if the corresponding element in $\boldsymbol{\delta}$ is greater than 0. Since the design specification of process errors $\boldsymbol{\sigma}$ is known in the design process, this term can be moved into the left hand side of Eq. (43) and let $\mathbf{f} = \mathbf{t} - \boldsymbol{\Lambda\Omega\sigma}$, thus Eq. (43) can be rewritten as

$$\mathbf{f} = \boldsymbol{\Lambda\Omega\delta} + \boldsymbol{\xi} \tag{44}$$

where $\mathbf{f}$ contains all the known terms regarding the measurements and the process design specifications. Matrices $\boldsymbol{\Lambda}$ and $\boldsymbol{\Omega}$ are also known beforehand in the design process. Vectors $\boldsymbol{\delta}$ and $\boldsymbol{\xi}$ are the unknown variables in the vectorized variance model that should be estimated. Let

$$\mathbf{D} = \boldsymbol{\Lambda\Omega} \tag{45}$$

be named the design matrix, using Eqs. (44) and (45), the estimation process is to estimate $\boldsymbol{\delta}$ (the variance change of the corresponding process faults) and $\boldsymbol{\xi}$ (noise of the system) based on the following linear model,

$$\mathbf{f} = \mathbf{D}\boldsymbol{\delta} + \boldsymbol{\xi} \tag{46}$$

This estimation is the key for fault diagnosis of multi-station assembly processes as it determines which process errors having larger variance (i.e. process faults).

4.4 Diagnosability Study and Optimal Sensor Placement Method using Compressive Sensing

In this section, the diagnosability study of multi-station assembly processes with underdetermined systems is performed based on the compressive sensing theory. Sec. 4.4.1 presents the diagnosability analysis of multi-station assembly processes with underdetermined systems. Sec. 4.4.2 proposes the optimal sensor placement method to optimize the diagnosability criterion. As illustrated in Figure 4-1, these two sections correspond to the third and fourth steps in the overall procedure of the proposed method.

4.4.1 Compressive Sensing based Diagnosability Study

According to Eq. (46), a diagnosable system is defined as the case that the vector $\boldsymbol{\delta}$ can be uniquely estimated. In an actual manufacturing environment, the number of sensors is usually less than that of the process errors; hence, Eq. (46) becomes a set of underdetermined linear equations as the number of equations is less than the unknown variables. Underdetermined linear equations result in non-existence of a unique solution for $\boldsymbol{\delta}$ in Eq. (46) and thus make the system non-diagnosable. To handle the above challenge, we adopt the sparse solution which is the solution that has the minimum number of non-zero elements of $\boldsymbol{\delta}$ in Eq. (46). Seeking sparse solution in the fault diagnosis approach is in compliance with multi-station assembly as justified in Sec. 4.1, namely, the probability of having less process faults is higher than that of having more process faults, assuming the process faults are independent of each other. Compressive sensing [70, 71] enables to achieve sparse solutions for underdetermined linear equations in an effective way. Thus, in this study, we study the sparse solution for Eq. (46) using compressive sensing theory (Sec. 4.4.1.1), and then conduct the diagnosability analysis for multi-station assembly processes (Sec. 4.4.1.2).

4.4.1.1 Sparse Solution using Compressive Sensing

Compressive sensing is a novel sensing approach as opposed to the traditional sampling method that follows the Shannon-Nyquist rate in signal processing. Under the compressive sensing paradigm, the signal can be reconstructed from a smaller number of samples than the Shannon-Nyquist rate [71]. Compressive sensing relies on a fundamental principal, namely, sparsity. Let $\mathbf{x}_{n\times 1}$ be a signal, and $\boldsymbol{\Psi} = [\,\boldsymbol{\psi}_1\,\boldsymbol{\psi}_2 \dots \boldsymbol{\psi}_n]$ be a representation matrix with dimension of $n \times n$, then

$$\mathbf{x} = \mathbf{\Psi s} \tag{47}$$

where $\mathbf{s}_{n\times 1}$ is the vector of coefficients that present the original signal $\mathbf{x}$ at the basis of $\mathbf{\Psi}$. The signal $\mathbf{x}$ can be represented as a sparse vector if most of the elements of $\mathbf{s}$ are zero. Vector $\mathbf{s}$ is v-sparse if it only has v nonzero elements (v is called the sparsity level). Many of the natural and engineered signals, at some certain basis $\mathbf{\Psi}$ (wavelet, Fourier, etc.), are sparse (note that if $\mathbf{x}$ itself is sparse, then $\mathbf{\Psi} = \mathbf{I}$)[70, 71]. The sparsity principle is the foundation to reconstruct the signal from fewer measurements [70, 71]. Let $\mathbf{\Phi}_{m\times n}$ be a sensing matrix (determined by signal sampling scheme) with $m < n$, then the compressive sensing measurements can be represented as

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{e} \tag{48}$$

where $\mathbf{y}_{m\times 1}$ denotes the measurements, and $\mathbf{e}_{m\times 1}$ represents the measurement noise. Substituting Eq. (47) into Eq. (48), we have:

$$\mathbf{y} = \mathbf{\Phi \Psi s} + \mathbf{e} \tag{49}$$

Under the compressive sensing paradigm, the dimension of measurement $\mathbf{y}$ could be smaller than that of the signal $\mathbf{s}$, namely $m < n$, so the linear equations in Eq. (49) are underdetermined. Thus, the sparse solution is utilized. This can be carried out through solving l_0-norm minimization [71] as follows

$$\min \|\mathbf{s}\|_0 \quad \text{s.t. } \|\mathbf{y} - \mathbf{\Phi\Psi s}\|^2 < \gamma \tag{50}$$

where $\|\mathbf{s}\|_0$ is the l_0-norm of the vector $\mathbf{s}$ which simply represents the number of non-zero elements of the vector, and parameter γ is the noise level ($\|\mathbf{e}\|_2 \leq \gamma$) which enables the sparse solution to be robust, namely, small changes in measurements $\mathbf{y}$ result in small changes in the solution. However, solving Eq. (50) is NP-hard [7] as it needs a combinatorial search, i.e., to search all possible subsets of $\mathbf{s}$ with a minimum number of non-zeros. There have been a number of sparse solutions methods reported in the literature to estimate the solution for Eq. (50), such as l_1 norm minimization [9, 76], greedy methods [14, 77, 78], and Bayesian methods [15, 17].

There is a similarity between the vectorized linear model Eq. (44) for multi-station assembly processes and the linear model Eq. (49) in compressive sensing context. The similarity can be highlighted if we let

$$\mathbf{\Phi} = \mathbf{\Lambda} \tag{51}$$

$$\mathbf{\Psi} = \mathbf{\Omega} \tag{52}$$

Proposition 1 in Sec. 4.2.3 proves the measurement matrix $\mathbf{\Lambda}$ is only determined by sensor placement in multi-stations. Since matrix $\mathbf{\Phi}$ in compressive sensing context only presents the sensor sampling scheme of the signal, Eq. (51) can be justified. Moreover, Eq. (52) indicates that the process/product matrix $\mathbf{\Omega}$ is interpreted as the signal representation matrix $\mathbf{\Psi}$ in compressive sensing context. Equation (52) can also be verified through Proposition 1, which proves that the process/product matrix $\mathbf{\Omega}$ is only determined by product and process information.

Due to the large number of process errors and relative small number of measurements, it is reasonable to assume that the vectorized variance model by Eq. (46) is an underdetermined linear system and thus we seek for the sparse solution of $\boldsymbol{\delta}$. The decomposition of matrix $\mathbf{D}$ into product/process matrix $\mathbf{\Omega}$ and measurement matrix $\mathbf{\Lambda}$ shows the clear relationship between the compressive sensing formulation Eq. (49) and our vectorized variance model Eq. (46). Thus, we utilize the criteria exist in compressive sensing theory which depend on the properties of matrix $\mathbf{D}$ in Eq. (45). Thus, by optimizing matrix $\mathbf{\Lambda}$ (in $\mathbf{D} = \mathbf{\Lambda\Omega}$, Eq. (45)), we may achieve the required properties for matrix $\mathbf{D}$. The inherent connections and consistence between our sensor placement problem and the CS formulation are justified through Proposition 1, and Eqs. (51) and (52) (see Appendix II for the proof).

According to the similarities between multi-station assembly processes and compressive sensing presented in Eqs. (51) and (52), fault diagnosis for underdetermined systems can be carried out using the compressive sensing approach. Using the sparse solution concept based on Eq. (50), fault diagnosis (i.e. the solution for Eq. (46)) can be formulated as

$$\min \|\boldsymbol{\delta}\|_0 \quad \text{s.t. } \|\mathbf{f} - \mathbf{D}\boldsymbol{\delta}\,\|^2 < \gamma \tag{53}$$

where $\mathbf{D} = \boldsymbol{\Lambda}\boldsymbol{\Omega}$, and $\|\boldsymbol{\delta}\|_0$ represents the number of non-zero elements, namely, the number of process faults. The sparse solution formulation in Eq. (53) is an NP-hard problem [7]. Therefore, finding the exact solution to Eq. (53) via instinctive computation is practically untenable. Accordingly, some sparse estimation algorithms have been devised, such as convex optimization algorithms [8, 18], greedy algorithms [5, 12, 14] and Bayesian algorithms [15, 17].

Bayesian algorithms, such as the relevance vector machine (RVM) method [15, 17], have recently received more attention than other sparse estimation algorithms, due to their more accurate sparse estimation performance [43]. Furthermore, as opposed to other sparse estimation algorithms, the Bayesian algorithms are able to estimate the noise variance, which is required in the fault diagnosis. Therefore, in Sec. 4.1, based on the RVM method [25], we developed an enhanced RVM algorithm [58] to solve underdetermined linear systems such as Eq. (53). This method has a Bayesian learning framework to estimate the posterior distribution on $\boldsymbol{\delta}$ instead of just a point estimate, and also provides the estimate of the noise $\boldsymbol{\xi}$ in Eq. (46). The key idea of RVM is that it enforces a parameterized prior on $\boldsymbol{\delta}$ that encourages sparsity in its posterior representation, namely, few nonzero values. For more details of the RVM and the enhanced RVM method, please refer Refs. [25] and [2].

4.4.1.2 Diagnosability analysis of Multi-Station Assembly Processes

In general, full diagnosability of process faults ensures that all the process errors can be uniquely estimated. In other words, the unique sparse solution for Eq. (53) can be achievable. The criteria for unique sparse solution of Eq. (53) have been discussed in the literature of Compressive Sensing theory [18], and are manifested through some requirements on the structure of matrix **D** (Eq. (45)). According to Eq. (45), matrix **D** is determined by process/product matrix **Ω** and measurement matrix **Λ** (the results of Proposition 1). Since matrix **Ω** is specified in the design process, it cannot be changed during assembly processes to achieve the required conditions on **D**. However, matrix **Λ** is only determined by the sensor placement (based on Proposition 1). Thus, it is possible to change the sensor placement to achieve an optimal measurement matrix **Λ** and then **D** by which the unique sparse solution can be achieved with a high probability.

The first criterion discussed in Compressive Sensing is represented as *Restricted Isometry Property* (RIP) [70, 71, 76]. However, the challenge with RIP condition is that its evaluation is NP-hard [79]. Therefore, instead of RIP, a computationally efficient condition to evaluate matrix **D** has been investigated, such as the *mutual coherence* in compressive sensing theory [71, 79]. The mutual coherence of matrix **D** is defined as follows,

$$\mu(\mathbf{D}) = \max \frac{\left|\mathbf{d}_{\mathrm{i}}{}^{\mathrm{T}}\mathbf{d}_{\mathrm{j}}\right|}{\|\mathbf{d}_{\mathrm{i}}\|\left\|\mathbf{d}_{\mathrm{j}}\right\|}, 1 < i,j < n \ (i \neq j) \tag{54}$$

where $\mathbf{d}_i$ is the i-th column of the design matrix **D**, and $\|\cdot\|$ represents the l_2-norm of a vector. Indeed, the mutual coherence $\mu(\mathbf{D})$ measures the largest absolute and normalized inner product (correlation) between different columns in matrix **D**. An alternative way to represent the mutual coherence $\mu(\mathbf{D})$ is through using the Gram matrix $\mathbf{G} = \overline{\mathbf{D}}^T\overline{\mathbf{D}}$ [6], where $\overline{\mathbf{D}}$ is the normalized matrix **D**, and the mutual coherence is the largest absolute value of the off-diagonal elements of the Gram matrix **G**. Note that minimizing the correlation between the columns of matrix **D** contributes to toward orthogonality between the columns of matrix $\mathbf{D}$ (according to the inner product concept). Therefore, matrix $\mathbf{\Lambda}$ could be optimized based on the results of Proposition 1, i.e., $\mathbf{D} = \mathbf{\Lambda\Omega}$, to improve mutual coherence of $\mathbf{D}$.

As indicated by Candes and Romberg [70], the unique solution to Eq. (53) is achieved with overwhelming probability if the following condition holds,

$$m \geq C \cdot \mu^2 \, (\mathbf{D}) \cdot v \cdot log \; n \tag{55}$$

where m and n represent the numbers of measurements and signal dimension, respectively, v is the 0sparsity level; and C is some positive constant which can be estimated based on the assumption that the sensing matrix is random [16]. However, in our application, measurement matrix $\mathbf{\Lambda}$ is constructed based on the sensor placement (location and distribution of the sensors), which is not random. Hence, the constant C is unknown in our application; but we can still utilize this theoretical condition as a qualitative relationship to justify the effect of mutual coherence in seeking unique sparse solution. Indeed, for a given multi-station assembly model, where m and n are usually fixed quantities, and the sparsity level v (actual number of process faults) is unknown, then clearly the only way to improve the diagnosability (probability of achieving the unique solution) is to minimize mutual coherence or its variant average mutual coherence which is discussed in the following.

Essentially matrices with low mutual coherence are required for accurate sparse estimation. Some theoretical results in Refs. [80, 81] have shown that if the inequality

$$\|\boldsymbol{\delta}\|_0 \leq \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) \tag{56}$$

holds, then vector $\boldsymbol{\delta}$ can be uniquely estimated from the measurements $\mathbf{f} = \mathbf{D}\boldsymbol{\delta}$ (non-noisy measurements), via some sparse estimation algorithms. From this inequality, it can be seen that minimizing mutual coherence $\mu(\mathbf{D})$ increases the probability of successfully reconstructing $\boldsymbol{\delta}$; and this result is consistent with Eq. (54). However, Elad [6] argues that minimizing the mutual coherence (Eq. (54)) may not improve the actual performance of sparse estimation techniques; but an average measure of mutual coherence does. Elad's claim is based on the fact that the upper bound presented in Eq. (56) has been derived according to a worst-case standpoint (i.e., maximum pairwise correlation between columns of matrix $\mathbf{D}$). In other words, optimizing the mutual coherence only addresses the effect of the columns of matrix $\mathbf{D}$ with maximum pairwise correlation, and ignores the effect of all other columns, which however also have impact on the probability of obtaining the unique solution. This issue can be resolved by the average mutual coherence for optimization, namely, considering the effect of all other columns of matrix $\mathbf{D}$ in sparse estimation performance [4]. The average measure of mutual coherence can be denoted by the average mutual coherence which is defined as follows

$$\mu_{avg}(\mathbf{D}) = \frac{\sum_{i \neq j} |g_{ij}|}{n(n-1)} \tag{57}$$

where g_{ij} represents the element of the Gram matrix $\mathbf{G} = \overline{\mathbf{D}}^T \overline{\mathbf{D}}$ at the i-th row and j-th column, and n is the number of process errors.

The sparse estimation performance is computed as following. Assume we randomly generate r process fault scenarios , namely, $\boldsymbol{\delta}_i$ ($\forall i = 1, \ldots, r$) represents the i-th process fault scenario. As specific, we generate $\boldsymbol{\delta}_i$ with sparsity level of $\|\boldsymbol{\delta}_i\|_0 = v$ for r cases, where the location of the process faults (i.e., location of nonzero values in vector $\boldsymbol{\delta}_i$) has been selected randomly, and the magnitude of process faults (i.e., nonzero values in vector $\boldsymbol{\delta}_i$) has been generated with some considerations of the application case (more details are provided in Sec. 4.5). Let $\widehat{\boldsymbol{\delta}_i}$ be the enhanced RVM estimate of $\boldsymbol{\delta}_i$. The estimation error corresponding to the i-th process fault scenario is denoted by

$$R_i = \left\| \boldsymbol{\delta}_i - \widehat{\boldsymbol{\delta}_\iota} \right\| \qquad \forall i = 1, \dots, r \tag{58}$$

and its relative error is

$$\tau_i = \frac{R_i}{\|\boldsymbol{\delta}_i\|} \qquad \forall i = 1, \dots, r \tag{59}$$

where ‖·‖ denotes a l_2-norm of a vector. Then, we define the diagnosability criterion as the average relative error (averaged over r process fault scenarios) as follows

$$\vartheta_r = E_r(\tau_i) < \theta \tag{60}$$

where θ is a user-defined scalar threshold for ensuring unique sparse solution. Namely, if average relative error ϑ_r is very small enough (determined by θ in Eq. (60)), it means that all the r process fault scenarios have been accurately estimated; hence the unique solutions have been achieved for all the fault scenarios, and consequently the related assembly process is diagnosable. Therefore, we define diagnosability of the system as the following:

Definition 1: A multi-station assembly processes with underdetermined linear equation is called diagnosable if the unique sparse solution to Eq. (53) is achieved, i.e., the condition of Eq. (60) is satisfied.

In the following section, the conditions to achieve the unique sparse solution are investigated. We formulate the condition to achieve the unique sparse solution for Eq. (53) as an optimization problem seeking an optimal measurement matrix $\boldsymbol{\Lambda}$ by which the average mutual coherence $\mu_{avg}(\mathbf{D})$ is minimized.

4.4.2 Optimal Sensor Placement based on Compressive Sensing

The optimal sensor placement in multi-station assembly processes is formulated as an optimization problem which minimizes the average mutual coherence $\mu_{avg}(\mathbf{D})$ defined in Eq. (24) by seeking the optimal sensor placement $\boldsymbol{\Lambda}^*$, namely,

$$\text{Min} \quad \mu_{avg}(\mathbf{D}) \tag{61.a}$$

$$\text{Subject to } \boldsymbol{\alpha} \in F \tag{61.b}$$

$$\mathbf{u} \in S \tag{61.c}$$

where both $\boldsymbol{\alpha}$ and $\mathbf{u}$ are decision variables; $\boldsymbol{\alpha}$ refers the sensor's station assignment; $\mathbf{u}$ denotes the sensor locations (coordinates) on the parts; F and S represent the sensor's possible station assignments and feasible locations on the parts, respectively, which are presented in details as follows. Sensor's station assignment refers the station in which the sensor should be placed and is denoted as

$$\boldsymbol{\alpha} = [\alpha_1 \; \alpha_2 \; \dots \; \alpha_c \,]^T \tag{62}$$

where α_i represents the station to which the i-th sensor is assigned, and c is the total number of sensors to be utilized. For example, if there is an assembly process with three stations, and four sensors are utilized to take the measurements on the parts in the assembly operations. The sensors are distributed as follows: (1) the first and the second sensors are assigned to station 1, (2) the third sensor is assigned to station 2, and (3) the fourth sensor is assigned to station 3. Using Eq. (61.a), the sensors' station assignment for the above assembly processes can be represented by

$$\boldsymbol{\alpha} = [\alpha_1 \; \alpha_2 \, \alpha_3 \;\; \alpha_4 \,]^T = [1 \; 1 \; 2 \; 3]^T.$$

It should be noted that there are some constraints on the first level of design variables Eq. (61.a), i.e., assigning the sensors to some specific stations. This is because that sensor cannot be assigned to the stations when parts to be measured have not arrived at the assembly process. For example, if part number 2 comes to the assembly process at station 2, the measurements for that part cannot be taken before station 2. By considering this constraint, the feasible candidates for α_i in Eq. (61.a) are defined with f_i (for the above example, $f_2 = [2 \;\; 3 \,]^T$) and consequently the constraint on the sensor's station assignment is represented with $\boldsymbol{\alpha} \in F$ where $F = [f_1 \, f_2 \; \dots \; f_c \,]$ presents all the possible sensor assignments, as shown in Eq. (61.a). The sensor location is represented by the coordinates of the sensors on the corresponding part, and is denoted as $\mathbf{u} = [x_1\ y_1\ z_1\ \ldots x_c\ y_c\ z_c]^{\mathrm{T}}$ where $(x_i\ y_i\ z_i)$ represents the coordinate of the i-th sensor. The feasible design space for $\mathbf{u}$ is denoted by S in Eq. ((61.a).c), which is the secondary level of constraints.

The problem in Eq. (61.a) is a mixed integer nonlinear programming model. The objective function $\mu_{avg}(\mathbf{D})$ is a complex nonlinear function of $\mathbf{u}$ and $\boldsymbol{\alpha}$**,** the design space S is a continuous non-convex set, and the sensor assignments feasible region F contains a set of integer values. We solve the problem in two steps. In the first step, we find the feasible region for sensors' station assignment α (integer programming part of the model). In the second step, given the feasible sensors' station assignment $\boldsymbol{\alpha}$, we utilize the sequential space filling methods (adopted from our prior work [82],) to solve the resulting nonlinear programming model for $\mathbf{u}$**.** The sequential space filling method is based on the sampling approaches (such as sequential exponential set, Hammersley set, Monte Carlo, etc.) to search optimal designs. Then the optimal solution is determined by comparing solutions across the assignments, and finding the sensor locations with the minimum $\mu_{avg}(\mathbf{D})$.

This two-step solution approach may only be utilized for small sized problem (i.e., the number of parts in the multi-station assembly processes is small) due to the exhaustive search implemented in the first step. For large sized problems, the metaheuristic Tabu search based algorithm developed in Ref [83] can be utilized. However, description of the working procedures of this algorithm is out of the scope of this subchapter, and the readers are referred to [83] for more details.

When the resulting optimal sensor placement is obtained from Eq. (61.a), the enhanced RVM [58] is run for r process fault scenarios and then the diagnosability criterion in Eq. (60) is applied. If the diagnosability criterion is satisfied, we can conclude that a diagnosable system is achieved through the optimal sensor placement.

4.5 Case Studies

This section demonstrates the proposed sensor placement optimization method with two case studies. In Sec. 4.5.1, the effectiveness of the proposed method is evaluated in improving the diagnosability for a single set of process faults. In Sec. 4.5.2, a multiple sets of process faults are considered for a more general diagnosability study.

4.5.1 Fault Diagnosis for a Single Set of Process Faults

An assembly model from an autobody assembly process is used for demonstrations of the proposed method. The assembled product is a floor pan that has four parts shown in Figure 3-4 For simplification we refer to these four parts (left-hand floor pan (LFP), right-hand floor pan (RFP), left bracket (LB), and right bracket (RB)) as part 1, part 2, part 3, and part 4, respectively. Figure 3-5 illustrates the multi-station assembly processes for the floor pan. The parts are being held by fixtures and part-mating features during the assembly operations. The KCC's are adopted from our previous research work [32]. The locations of the KCCs are presented in Figure 4-3 and Figure 4-4. Fixtures 3-2-1 are utilized in each station to locate the parts and subassembly as illustrated in Figure 4-4. Here, triangles present "3" and straight lines present "2-1".

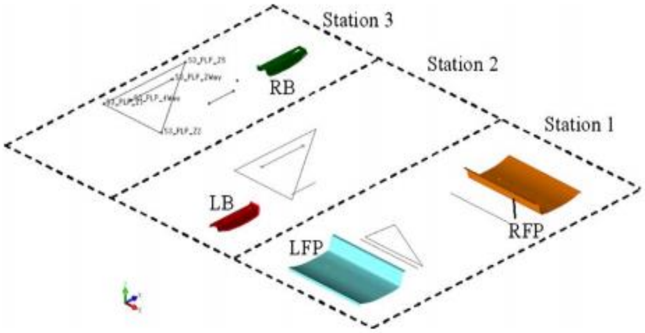

Figure 4-3. The Floor pan assembly process with the fixtures locators

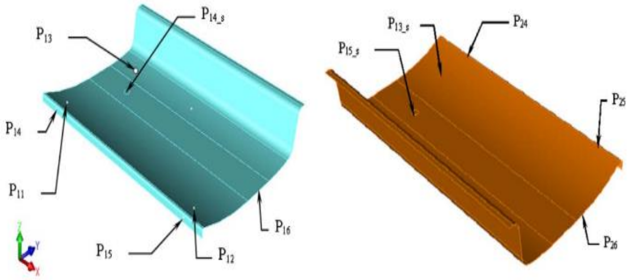

Figure 4-4. KCC locations for left-hand and right-hand floor

Table 4-2 provides the coordinates of KCCs in x-y-z directions. Here for simplification, all the fixture locators are for floor pan left-hand, right-hand or subassembly locators, and not for brackets.

The fixture errors and the part-mating feature errors are considered as the process errors or KCCs. There are 33 KCCs in this assembly process and four measurement points (sensor locations) to be located during the assembly processes. As presented in Figure 3-5, each part has one measurement point: M_1 on part 1, M_2 on part 2, M_3 on part 3, and M_4 on part 4, respectively. These measurements can only be taken in the stations where the corresponding part has been assembled. For example, M_3 on part 3 cannot be taken at station 1 since part 3 has not yet arrived at the assembly process at station 1. With the initial sensor placement shown in Figure 3-5, the design variables has two levels as follows: (1) sensor's station assignment e.g., $\boldsymbol{\alpha} = [2,1,2,3]^T$ representing M_1, M_2, M_3 and M_4 are assigned to stations 2, 1, 2, and 3, respectively, and (2) the sensor locations on the parts $\mathbf{u} = [x_1\ y_1\ \ z_1 \ldots x_4\ y_4\ z_4]^{\mathrm{T}}$denoting the coordinates of the sensors. Since the change in z direction of the parts is very small and their effect on the sensor placement optimization is assumed to be small too, they are removed from the second level of design variables $\mathbf{u}$. The initial $\mathbf{u}$ is presented in Table 4-3.

Table 4-2. The Coordinates for KCCs

KCCs	X	Y	Z
P_{11}	857.66	-591.95	36.09
P_{12}	1719.23	-619.04	36.09
P_{13}	804.10	-207.98	3.72
P_{14}	1129.95	-700.00	35.22
P_{15}	1654.22	-700.00	35.22
P_{16}	1800.00	-482.95	0
P_{24}	734.47	681.35	35.22
P_{25}	1696.78	681.35	35.22
P_{26}	1800.00	389.1549	0
P_{13_s}	963.68	567.76	36.09
P_{14_s}	1402.41	-430.90	0
P_{15_s}	1402.41	333.92	0

Given the floor pan assembly operations, and the initial sensor placement, the diagnosability study of a single set of process faults is carried out. Following the method presented in Sec. 4.2.3.1, the state-space model of the floor pan is constructed. Applying the methods in our prior research [72, 74], fault pattern matrix $\mathbf{\Gamma}$ in Eq. (37) is constructed. There are 33 KCCs and 8 KPCs (measurements) in the floor pan assembly processes, thus fault pattern matrix $\mathbf{\Gamma}$ has dimension of 8×33. The 33 KCCs of the floor pan are assumed to have normal distributions. The design specifications of their variability are given in terms of tolerance range which is defined as six times of the standard deviation of the fixture errors (6-sigma). For KCC_1 to KCC_{30}, the design specification of tolerance ranges are given as 0.5 mm, and KCC_{31} to KCC_{33} as 1 mm. By running the Monte Carlo simulation using 3DCS^{TM} software, the assembly process can be accurately simulated.

The floor pan assembly processes is simulated with a single set of purposely designed process faults which are manipulated in the way that the tolerance ranges of KCC_5, KCC_{10}, KCC_{16}, KCC_{26} are increased to 1 mm, which is beyond their design specification, i.e., 0.5 mm (change of variance

$\Delta = 0.02$). The number of samples for the simulation is set at 400. After the simulation, the measurements of the 8 KPCs of these 400 samples are collected ($\mathbf{y}$ in Eq. (37)).

Following Proposition 1, the vectorized variance model for the initial floor pan assembly processes is constructed. The design matrix $\mathbf{D} = \mathbf{\Lambda\Omega}$ is also constructed. The dimension of matrix $\mathbf{D}$ is 36×33 as there are 8 KPCs (leading to $\frac{8\times 9}{2} = 36$ rows in the matrix that basically accounts for the number of the upper-triangular elements of the covariance matrix of $\mathbf{y}$ in Eq. (39.a)).) and 33 KCCs. The rank of matrix $\mathbf{D}$ is 26, so we have an underdetermined system. Calculating the measurements covariance matrix vectorized in the vector $\mathbf{t}$ in Eq. (43) and using the design specification limits on the KCCs, namely, $\boldsymbol{\sigma}$ in Eq. (42) (i.e., 0.5/6 mm for KCC_1- KCC_{30} and 1/6 mm for the rest of KCCs) from the vector $\mathbf{f}$ (Eq. (44)).

According to Sec. 4.4.1, the compressive sensing based diagnosability study is carried out for the initial floor pan model. The enhanced RVM is used as the sparse solution method. Following the diagnosability analysis presented in Sec. 4.4.1, the enhanced RVM [58] was run, diagnosability criterion $\boldsymbol{\vartheta_r}$ defined in Eq. (60) is verified (note that here r=1 since there is only one process fault scenario is considered). A summary of the reported results of the fault diagnosis for the initial matrix $\mathbf{D}$ is given in

Table 4-4. The user-defined $\boldsymbol{\theta}$ denoting the threshold for uniqueness analysis of sparse solution is set as 5%. The resulting $\boldsymbol{\vartheta_r}$ for the initial sensor layout is 0.870 (87.0%), which is much larger than $\boldsymbol{\theta}$ and indicates that the sparse solution is not unique. Thus, the assembly system is not diagnosable given the current sensor placement (i.e.,$\boldsymbol{\alpha} = [\mathbf{2}, \mathbf{1}, \mathbf{2}, \mathbf{3}]^{\boldsymbol{T}}$

Table 4-3. Design parameters and optimal results for sensor locations

Design variables	Initial location ($\mathbf{u}$)	Design space	Optimal location
x_1	1285.55	(1200,1400)	1223.12
y_1	-643.50	(-550, -643.50)	-561.35
x_2	1285.55	(1200,1400)	1338.39
y_2	662.50	(550, 662.89)	608.76
x_3	1615.14	(1500, 1650)	1571.99
y_3	-442.98	(-350,-450)	-442.20
x_4	1615.14	(1500, 1650)	1591.83
y_4	354.66	(350 ,450)	359.92

Table 4-4. Matrix properties and ϑ_r for the initial and optimal sensor placement

Initial sensor placement		Optimal sensor placement ($\mathbf{D}^*$)	
Avg. mutual coherence	ϑ_r	Avg. mutual coherence	ϑ_r
0.214	0.870	0.136	0.041

Table 4-5. Sensor's station assignments scenarios for the floor pan and the minimum, maximum and the mean of the mutual coherence for each candidate.

Sensor station assignment	Minimum avg. mutual coherence	Mean of the avg. mutual coherence	Maximum avg. mutual coherence
1-1-2-3	0.1641	0.1723	0.1798
1-2-2-3	0.1743	0.1877	0.2009
1-3-2-3	0.1867	0.1972	0.2053
1-1-3-3	0.1360	0.1390	0.1418
1-2-3-3	0.1480	0.1551	0.1605
1-3-3-3	0.1538	0.1621	0.1682
2-1-2-3	0.1924	0.2030	0.2136
2-2-2-3	0.1743	0.1877	0.2009
2-3-2-3	0.1867	0.1972	0.2053
2-1-3-3	0.1739	0.1773	0.1813
2-2-3-3	0.1480	0.1551	0.1605
2-3-3-3	0.1538	0.1621	0.1682
3-1-2-3	0.1987	0.2051	0.2087
3-2-2-3	0.1881	0.1921	0.2007
3-3-2-3	0.1652	0.1721	0.1834
3-1-3-3	0.1714	0.1747	0.1787
3-2-3-3	0.1774	0.1787	0.1807
3-3-3-3	0.1585	0.1622	0.1669

and initial u presented in Table 4-3). In the following, our optimal sensor placement method presented in Sec. 4.4.2 is utilized to increase the possibility of achieving the unique sparse solution, namely, improve the diagnosability for the process faults in the floor pan model. The proposed optimal sensor placement aiming to minimize the average mutual coherence modeled by Eq. (24) includes determining the optimal sensor's station assignment $\boldsymbol{\alpha}^*$ and the sensor locations $\mathbf{u}^*$ in multi-station assembly processes. In our case study, for the sensor locations, we have eight design variables which denote the coordinates of the sensors along the axes x and y. The design space (S in Eq. (61.a)) for these variables is given in the third column of Table 4-3. The mutual coherence values before and after the optimization is shown in Table 4-4. The sensor's feasible station assignment F is provided in Table 4-5 (column 1). There are 18 candidates for the sensor's station assignment.

The number of samples for the sequential space filling is set to 1000, and we have 18 candidates for the sensor sensor's station assignment. Thus, there are a total of 18,000 alternatives. The corresponding design matrix **D** should be constructed for each alternative. The average mutual coherence for each of the constructed matrices **D** is calculated. The results on the minimum, mean, and maximum of the average mutual coherence for each candidate are reported in Table 4-5. From these 18 cases, the sensors' station assignment $\boldsymbol{\alpha}^* = [\mathbf{1\ 1\ 3\ 3}]^{\boldsymbol{T}}$ has the smallest average mutual coherence. The Optimal sensor locations $\mathbf{u}^*$ is provided in the 4th column of Table 4-3.

Given the optimal solutions for Eq. (61.a), i.e., $\boldsymbol{\Lambda}^*$ which results in the least average mutual coherence for matrix **D**, we would like to verify if the resulting optimal sensor layout provides sufficient diagnosability for the floor pan assembly model. Again, the enhanced RVM method is utilized to solve Eq. (53) given the optimal matrix $\mathbf{D}^* = \boldsymbol{\Lambda}^*\boldsymbol{\Omega}$ (from Eq. (45)) and vector **f**. The enhanced RVM is run and the diagnosability criterion ϑ_r is 0.0409 (4%) (Table 4-4), which is smaller than the user-defined θ that is set to 5%. The small value of ϑ_r indicates the existence of a unique sparse solution for Eq. (53) which uniquely identifies the process faults as KCC_5, KCC_{10}, KCC_{16}, and $KCC_{26.}$ These exactly match the faults we simulated to the process.

4.5.2 Fault Diagnosis for Multiple Sets of Process Faults

In the previous case study, the optimal sensor placement method is verified for a specific set of process faults introduced to the assembly process (i.e. KCC_5, KCC_{10}, KCC_{16}, and KCC_{26}). Now, the question that could be asked is that whether the proposed method is only effective for these four process faults. How about another set of process faults with size of four of KCCs (e.g. KCC_2, KCC_7, KCC_{20}, and KCC_{24}). In this case study we demonstrate that the optimal sensor placement is also effective for diagnosis of many other sets of process faults (different scenarios). This case study is originally adopted from the work of Elad [6] where he proposes an optimal sensing matrix for compressive sensing problems. We adopted this case study to our multi-station assembly processes application. This case study includes the following steps.

Step 1- Simulate multiple sets of process faults: In this step we simulate r=200 process fault scenarios including 200 vector $\boldsymbol{\delta}$ of Eq. (53), by randomly picking 4 process errors from KCC_1 through KCC_{33} , and purposely increasing their tolerance ranges (the value of the variance change is randomly selected from a normal distribution with mean of 0.02, and standard deviation of 0.001).

Step 2- Generate the initial case: In this step we choose the initial design matrix **D** as the same as Sec. 4.5.1. Through the simulation process for the floor pan assembly also presented in the previous section, we construct the vector **f** in Eq. (53) for 200 times given the same initial **D**.

Step 3- Test the diagnosability criterion ϑ_r for **D**: In this step, we carried out the diagnosability study for the initial **D**, through using the enhanced RVM for all the 200 cases. In other words, we solve Eq. (53) for all the r =200 scenarios simulated in Step 2 with the enhanced RVM method, and calculate the ϑ_r through Eq. (60).

Step 4- Optimize sensor placement: The optimal sensor placement method proposed in Sec. 4.4 is utilized to determine the optimal measurement matrix $\mathbf{\Lambda}^*$ which consequently provides $\mathbf{D}^* \ = \mathbf{\Lambda}^*\mathbf{\Omega}$ for the provided floor pan assembly process.

Step 5- Test the performance of the proposed method: In this step we simulate vector $\mathbf{f}$ for all the 200 scenarios provided in Step 1 given $\mathbf{D}^*$ and test the performance of the sensor placement method. As in Step 3, we utilize the enhanced RVM method for all the 200 cases generated earlier in this step and calculate ϑ_r.

Table 4-6. $\boldsymbol{\vartheta}_\mathbf{r}$ for multiple sets of process faults with the initial and optimal design matrix

Design matrix	$\mu_{avg}(\mathbf{D})$	ϑ_r (%)	Non-diagnosable systems
Initial case ($\mathbf{D}$)	0.2136	102%	200
Optimal case ($\mathbf{D}^*$)	0.1360	6.8%	4

Following the steps mentioned above, we are able to compare the diagnosability of the initial D with the optimal $\mathbf{D}^*$ in identifying multiple (200) sets of process faults that are randomly simulated. The results for the case study are reported in Table 4-6. According to Table 4-6, after optimization, the diagnosability criterion $\boldsymbol{\vartheta}_r$ the 200 cases is very small: 0.068. However, before optimization, for the initial case, $\boldsymbol{\vartheta}_r$ is very large which does not present a unique solution. Moreover, among the 200 scenarios that the enhanced RVM was run with the optimal sensor placement, only 4 of them have $\boldsymbol{\vartheta}_r$ larger than 7% (the user-defined $\boldsymbol{\theta}$ is set to 7% in this case study), but for the initial case, all the 200 scenarios have larger $\boldsymbol{\vartheta}_r$ than 7%. Therefore, the proposed sensor placement method optimizes the sensor distribution for the floor pan with $\mathbf{\Lambda}^*$ that provides sufficient diagnosability for 196 out of 200 randomly simulated process fault scenarios. This means that for

196 out 200 sets of process faults, the diagnostic method (enhanced RVM) successfully identify all the process faults with the optimized sensor placement.

4.6 Concluding Remarks

This subchapter investigates an optimal sensor placement method for multi-station assembly processes with underdetermined systems. We vectorize the variance model that is derived using the state space model. This formulation helps us draw close connections between the diagnosability study of the vectorized variance model and compressive sensing principles. Average mutual coherence is adopted from the compressive sensing theory to carry out the diagnosability analysis. An optimization method is proposed for optimal sensor placement in a multi-station assembly process based on the average mutual coherence criterion.

Two case studies are presented to demonstrate the effectiveness of the proposed method. In the first case study, a single set of process faults are intentionally introduced to the assembly process of a floor pan. Given the initial sensor placement, the diagnosability analysis of the floor pan model is carried out. The diagnosability criterion is computed as ϑ_r=87%, which demonstrates that the floor pan assembly model is not diagnosable given the initial sensor placement. Then the optimal sensor placement for the given case study is determined through minimizing the average mutual coherence. According to the resulting optimal sensor placement using the proposed method, all four of the process faults in the floor pan assembly model are identified accurately with diagnosability criterion ϑ_r=4.059%. Therefore, the proposed method offers the sufficient diagnosability for the illustrated case study. The second case study illustrates the effectiveness of the proposed method for multiple sets of process faults. 200 scenarios of process faults are randomly generated and introduced to the floor pan assembly model. After applying the proposed optimal sensor placement method, 196 out of 200 scenarios, which were not diagnosable before using the optimal sensor placement, were identified diagnosable. Also, the average of the diagnosability criterion for all 200 cases (ϑ_r decreased from 102% to 6.8%) indicates the effectiveness of the proposed method in improving the diagnosability for the process faults.

In this subchapter we present that by minimizing the average mutual coherence we can reach to a diagnosable multi-station assembly processes. However, the challenging issue that still remains is that how small the average mutual coherence should be. In other words, under which conditions the proposed method could effectively make a fully diagnosable system. Indeed, holding these conditions ensures the unique sparse solution for our under-study underdetermined linear system Eq. (53). There is a need to theoretically investigate these requirements for the enhanced RVM method which is used as the fault diagnosis approach in the manuscript. In the future we are going to research the theoretical conditions that the proposed optimal sensor placement method along with the underlying fault diagnosis method (enhanced RVM) provides the unique sparse solution. Furthermore, It should be mentioned that our optimization model in Eq. (61.a) can be extended to a more general case, by considering the number of measurements as a decision variable, along with α, and u. This will enable us to determine the minimum number of measurements leading to optimal sensor placement. We are going to study this extension in our future research work as well.

Chapter 5. Real-time Monitoring in Advanced Manufacturing Processes from Multiple Sensor Data

The objective of this chapter is to realize real-time monitoring of process conditions in advanced manufacturing using multiple, heterogeneous sensor signals. To achieve this objective we propose an approach invoking the concept of sparse estimation called online sparse estimation-based classification (OSEC). The novelty of the OSEC approach is in representing data from sensor signals as an underdetermined linear system of equations, and subsequently, solving the underdetermined linear system using a new developed greedy Bayesian estimation method. We apply the OSEC approach to two advanced manufacturing scenarios, namely, a fused filament fabrication (FFF) additive manufacturing process (AM); and an ultraprecision semiconductor chemical mechanical planarization (CMP) process. Using the proposed OSEC approach, process drifts are detected and classified with higher accuracy compared to popular machine learning techniques. Process drifts were detected and classified with a fidelity approaching 90% (F-score) using OSEC. In comparison, conventional signal analysis techniques, e.g., neural networks, support vector machines, quadratic discriminant analysis, naïve Bayes, were evaluated with F-score in the range of 40% to 70%[3].

5.1 Introduction

5.1.1 Objective and Motivation

The aim of this work is to achieve real-time monitoring in advanced manufacturing processes using multidimensional, heterogeneous sensor data. This objective is realized by developing a novel

[3] Bastani, K., Rao, P, and Kong, Z., 2015, "An online sparse estimation based classification approach for real-time monitoring in advanced manufacturing processes from heterogeneous sensor data," *IIE Transactions* (in press). DOI: 10.1080/0740817X.2015.1122254.

supervised classification approach termed as online sparse estimation-based classification (OSEC), which is rooted in the areas of sparse estimation and compressive sensing [1]. The imperative need for real-time detection of anomalous process conditions from heterogeneous sensor data, particularly, in advanced manufacturing scenarios is the motivation for this work. This is a practically important problem; for instance, a few seconds of an uncompensated process drift during planarization may result in scrapping of an entire 300 mm semiconductor wafer, leading to prohibitive yield losses [84, 85].

A large body of research in machine learning has been devoted to supervised classification of process conditions (labels) from sensor observations, such as support vector machine (SVM), linear/quadratic discriminant analysis (LDA/QDA), and neural networks (NN), etc. [86]. These algorithms are either computationally intensive or have poor classification accuracy for application to real-time monitoring in advanced manufacturing processes. The motivation of the proposed OSEC approach is to overcome the above shortcoming by achieving the following two aims: (1) improve the classification accuracy; and (2) reduce the computational burden for real-time process monitoring.

5.1.2 Significance and Novelty

The proposed OSEC approach achieves the above objectives based on a novel sparse representation classification framework [87]. This framework directly utilizes the raw sensor data to formulate the supervised classification problem as an *underdetermined* system of linear equations[4]. As further elucidated in the forthcoming literature review section (Sec. 5.2), existing sparse estimation approaches can solve the underdetermined linear system [8-17]. However, most of these existing approaches are not computationally efficient to accommodate real-time

[4] In an underdetermined system, the number of equations is less than the number of unknown variables.

monitoring the large volume of data and high sampling rates of sensors typical to advanced manufacturing scenarios. Therefore, a fast sparse estimation algorithm is further developed in this subchapter, namely, the greedy Bayesian method (GBM) presented in Sec. 5.3. Our proposed OSEC approach utilizes the GBM to efficiently solve the supervised classification problem. There are two advantage accrued from the proposed OSEC approach:

(1) It is equipped with the GBM algorithm, which we demonstrate is accurate, as well as computationally efficient, compared to existing sparse estimation algorithms.

(2) Precludes the need for extraction of statistical features (as opposed to existing supervised classification algorithms), because it directly operates on the raw sensor data. These claims are substantiated with numerically simulated and experimental data in Sec. 5.3.3 (for GBM) and Sec. 5.5 (for OSEC), respectively.

The rest of this subchapter is organized as follows: the related research in the area of sparse estimation is reviewed in Sec. 5.2; we develop a novel sparse estimation algorithm termed as GBM in Sec. 5.3; based on GBM, the OSEC approach is subsequently presented in Sec. 5.4; the effectiveness of the OSEC approach is demonstrated for real-time monitoring of AM and CMP processes in Sec. 5.5; and finally, conclusions and avenues for further work are discussed in Sec. 5.6.

5.2 Research Background and Related Work

The overarching goal of our real-time monitoring approach in this subchapter is to accurately determine the process state to which a new sensor data $\mathbf{y} \in \mathbb{R}^m$ belongs by using prior knowledge based on n training samples acquired from k distinct predefined process states. The training samples are contained in a matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ where m is the sensor dimension and n the number of data points. The distinct predefined process states (training data $\mathbf{\Phi}$) are obtained by extensive experiments and process observation. The OSEC approach attains the above goal by sparse representation of $\mathbf{y}$ using the entire training matrix $\mathbf{\Phi}$. Consequently, the domain of sparse representation/estimation is briefly reviewed in this section to provide the basic foundation of the proposed OSEC approach.

The sparse estimation problem is based on the following relationship,

$$\mathbf{y}_{m\times 1} = \mathbf{\Phi}_{m\times n}\mathbf{w}_{n\times 1} + \boldsymbol{\varepsilon}_{m\times 1}, \tag{63}$$

the aim is to estimate the coefficient vector $\mathbf{w} \in \mathbb{R}^n$ based on measurements $\mathbf{y}_{m\times 1}$, where $\mathbf{\Phi}_{m\times n}$ is the given training (system) matrix with a condition of $m << n$, and $\boldsymbol{\varepsilon}$ represents the measurement noise, This is a challenging problem to solve due to its inherently ill-defined (ill-posed) characteristics – there are infinite possible solutions. In other words, on expanding Eq. (63) as a system of linear equations, it is evident that the number of equations (m) are less than the number of variables (n). Such as system is called an underdetermined linear system.

Assuming that the vector $\mathbf{w}$ is sparse, i.e., it has a small number of nonzero elements, provides reasonable conditions to overcome the underdetermined nature of the sparse estimation formulation, and thereby achieve a unique solution [18]. Accordingly, the sparse estimation problem can be formally stated as,

$$\begin{gathered} min\ \|\mathbf{w}\|_0 \\ \|\mathbf{y} - \mathbf{\Phi w}\|_2^2 \leq \delta \end{gathered} \tag{64}$$

where $\|\mathbf{w}\|_0$ is the l_0-norm of vector $\mathbf{w}$ that represents the number of nonzero elements of the vector, and δ is the noise level ($\|\boldsymbol{\varepsilon}\|_2^2 \leq \delta$) [3]. However, the following challenges are evident for solving Eq. (64).

(1) It needs an exhaustive search over all subsets of vector $\mathbf{w}$, in order to find a particular support set with smallest cardinality satisfying the constraint in Eq. (64).

(2) The sparse estimation formulation in Eq. (64) is an inherently computationally intensive, combinatorial NP-hard problem [7].

Therefore, finding the exact solution to Eq. (64) via instinctive computation is practically untenable. Accordingly, a number of sparse estimation algorithms are reported in literature to solve this problem, such as convex optimization algorithms [8-11], greedy algorithms [12-14], and Bayesian algorithms [15-17].

The above research in the field of sparse estimation reveal that if $\mathbf{w}$ is sufficiently sparse, then the solution to Eq. (64) is equivalent to a form of a relaxed convex optimization problem attained by replacing $\|\mathbf{w}\|_0$ with $\|\mathbf{w}\|_1$, where, $\|\cdot\|_1 = \sum|\cdot|$ is the l_1-norm [6]. The sparse estimation problem on account of the relaxed convex optimization modification is,

$$\begin{gathered} min\,\|\mathbf{w}\|_1 \\ \|\mathbf{y} - \mathbf{\Phi w}\|_2^2 \leq \delta \end{gathered} \tag{65}$$

Convex optimization algorithms, such as Dantzig selector [8], basis pursuit de-noising (BPDN) [9], total variation (TV) regularization [10], and lasso [11] have been developed to solve the relaxed convex optimization problem of Eq. (65). A typical drawback of these algorithms is that, due to their global search, they are computationally intensive for large dimensional problems. Therefore, they are not suitable for real-time process monitoring applications.

To overcome the computational burden, some heuristic algorithms (also called *Greedy* algorithms), such as thresholding algorithm [14], orthogonal matching pursuit (OMP) [14], stage-wise orthogonal matching pursuit (STOPM) [12], and least angle regression (LARS) [13] have been proposed. These Greedy algorithms are based on a heuristic search to estimate the true support set $\Lambda = \{i : \mathbf{w}_i \neq 0, \forall i = 1, \dots, n\}$, which represents the location of nonzero elements in vector $\mathbf{w}$.

One of the most reliable greedy algorithms is OMP, which uses an iterative approach to estimate the true support set Λ of $\mathbf{w}$. Letting S be the estimated support set, OMP starts with the null set $S = \{\}$, and iteratively populates the set S by identifying the columns of matrix $\mathbf{\Phi}$**,** which are maximally correlated with the measurement residual vector $\mathbf{r}$ (for brevity purposes, the OMP approach is detailed in Appendix C). OMP is computationally efficient due to its heuristic nature. However, there is an inherent tradeoff, the sparse estimation performance of OMP often sacrifices accuracy for speed, and consequently, it is susceptible to be confined in a local minima.

While the convex optimization algorithms and greedy algorithms mentioned above provide a point estimate for coefficient $\mathbf{w}$, in contrast, Bayesian algorithms, such as relevance vector machine (RVM) [15, 16], and sparse Bayesian learning (SBL) [17] provide a posterior distribution on the coefficient $\mathbf{w}$. Particularly, RVM has recently received more attention than other sparse estimation algorithms, due to its comparatively better (more accurate) sparse estimation performance. RVM assumes a Gaussian likelihood function, and a conjugate prior promoting sparsity on the coefficient $\mathbf{w}$. By applying the Bayesian rule, the posterior distribution of $\mathbf{w}$ is estimated. The estimated posterior mean is then utilized as the RVM estimate $\mathbf{w}_{\text{RVM}}$ (RVM algorithm is summarized in Appendix D; further details of RVM can be found in Ref. [15]).

5.3 Greedy Bayesian Method (GBM) for Sparse Estimation

As discussed in Sec. 5.2, it is beneficial to integrate the inherent advantages stemming from the Bayesian framework in RVM, and computational efficiency of OMP for estimation of the support set by proposing a novel sparse estimation algorithm known as GBM. The computational efficiency of the proposed algorithm is critical for the success of the process state classification problem studied in this work, recognizing that it is intended for application to real-time monitoring of advanced manufacturing processes, where reticent detection of process drifts can lead to prohibitive losses.

5.3.1 The Proposed GBM Algorithm

Wipf and Nagaragan [88]show that the RVM estimate $\mathbf{w}_{\mathrm{RVM}} = \boldsymbol{\mu}_S$, where $\boldsymbol{\mu}_S$ is a subvector of estimated posterior mean $\boldsymbol{\mu}$ whose elements correspond to the estimated support set S by RVM, satisfies,

$$\underset{\mathbf{w}}{\arg\min} \|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|_2^2 + \sigma^2 g(\mathbf{w}) \tag{66}$$

$$g(\mathbf{w}) = \underset{\boldsymbol{\alpha} \geq \mathbf{0}}{\operatorname{argmin}}\, \mathbf{w}^T \mathbf{B}\mathbf{w} + \log|\sigma^2 \mathbf{I} + \boldsymbol{\Phi}\mathbf{B}\boldsymbol{\Phi}^T| \tag{67}$$

where $\mathbf{B} = Diag(\boldsymbol{\alpha})$, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_{\mathrm{n}}]^T$, $\boldsymbol{\alpha}_{\mathrm{i}}$ represents the hyperparameter associated with the i-th coefficient $\mathbf{w}_{\mathrm{i}}$, and σ^2 is the noise variance in the Gaussian likelihood function $p(\mathbf{y}|\mathbf{w}, \sigma^2)$. However, Solving Eq. (66) is computationally intensive for high dimensional problems as it requires several matrix inversion operations to estimate S. This can be verified through the following analysis of RVM:

Let $\boldsymbol{\mu}_S = \sigma_t^{-2} \boldsymbol{\Sigma}_S \boldsymbol{\Phi}_S^T \mathbf{y}$ and $\boldsymbol{\Sigma}_S = (\sigma_t^{-2} \boldsymbol{\Phi}_S^T \boldsymbol{\Phi}_S + \mathbf{B}_S)^{-1}$ be the posterior mean and posterior covariance of $\mathbf{w}$, and $\boldsymbol{\alpha}_i^t = \frac{1 - \boldsymbol{\alpha}_i^{t-1} \mathrm{N}_{\mathrm{ii}}}{\boldsymbol{\mu}_{\mathrm{i}}{}^2}$, $\sigma_t^2 = \frac{\|\mathbf{y} - \boldsymbol{\Phi}_S \boldsymbol{\mu}_S\|_2^2}{m - \sum_{i \in S}(1 - \boldsymbol{\alpha}_i^t \mathrm{N}_{\mathrm{ii}})}$ be the estimates for hyperparameterts, where N_{ii} is the i-th diagonal element of $\boldsymbol{\Sigma}$, and index t denotes the iteration number. RVM starts with estimated support set $S = \{1, \dots, n\}$, and iteratively updates $\boldsymbol{\mu}_S$, $\boldsymbol{\Sigma}_S$ with respect to $\boldsymbol{\alpha}_i^t$, σ_t^2, and vice versa. By progression of RVM algorithm (after some iterations of RVM), the hyperparameter of irrelevant coefficients gradually start to achieve very large values (e.g. $> 10^6$). Let $Q = \{v\colon \boldsymbol{\alpha}_v^t > \rho = 10^6 \;\; \forall v \in S\}$ be the index set of these coeficients, then the estimated support set is updated by $S = S \backslash Q$. From $\boldsymbol{\Sigma}_S$ it can be seen that one must invert matrices of size $n \times n$ at least for the first few iterations, and then by updating S, the size of matrix inversion can be reduced to

$|S| \times |S|$. Clearly these procedures of estimating support set S is computationally intensive for high dimensional problems (large n).

This shortcoming can be resolved by developing an iterative greedy method to optimize Eq. (66). The proposed algorithm is named GBM, as it utilizes a greedy method in estimating the support set, and is equipped with a Bayesian framework (RVM approach) to estimate the nonzero elements associated with the estimated support set. Mathematically, GBM incrementally selects the columns of matrix $\mathbf{\Phi}$, which locally optimizes Eq. (66). The GBM algorithm starts with an empty support set $S = \{\}$ and populates it iteratively. In specific, GBM determines suitable candidates to be included in the support set S by evaluating columns of matrix $\mathbf{\Phi}$ that have the best fit with respect to the measurement residual vector $\mathbf{r}$ (consistent with OMP iterative approach as explained in Sec. 2). The procedures are enumerated in the following steps (see also Figure 5-1):

```
GBM Algorithm
Step 1. Initialization (t=1)
    Initiate the hyperparameters $\boldsymbol{\alpha}^1 = (\boldsymbol{\alpha}_1^1, \boldsymbol{\alpha}_2^1, \dots, \boldsymbol{\alpha}_\mathrm{n}^1)^T = \mathbf{1}$,and $\sigma_1^2 = 0.01 \times var(\mathbf{y})$
    Set the support set $S = \{\ \}$ and its compliment $S^c = \{1, \dots, n\}$
    Set the initial measurement residual vector $\mathbf{r} = \mathbf{y}$
    Normalize the columns of matrix $\mathbf{\Phi}$ to have a unit norm
Step 2. Estimate the support set:
    solve $i^* = \arg\max\{|\mathbf{r}^T \mathbf{g}_i|\}, \ \forall i \in S^c$
    set $S = S \cup i^*, S^c = S^c \backslash \{i^*\}$
Step 3. Estimate posterior covariance and posterior mean, and Estimate
    hyperparameters ${\boldsymbol{\alpha}_S}^t = \{\boldsymbol{\alpha}_i^t\} \ \forall i \in S$ and $\sigma_t^2$ corresponding to $S$:
    $\boldsymbol{\Sigma}_S = (\sigma_t^{-2} \mathbf{\Phi}_S^T \mathbf{\Phi}_S + \mathbf{B}_S)^{-1}$
    $\boldsymbol{\mu}_S = \sigma_t^{-2} \boldsymbol{\Sigma}_S \mathbf{\Phi}_S^T \mathbf{y}$
    $\boldsymbol{\alpha}_i^t = \frac{1 - \boldsymbol{\alpha}_i^{t-1} \mathrm{N}_{\mathrm{ii}}}{{\boldsymbol{\mu}_\mathrm{i}}^2} \qquad i \in S$
    $\sigma_t^2 = \frac{\|\mathbf{y} - \mathbf{\Phi}_S \boldsymbol{\mu}_S\|_2^2}{m - \sum_\mathrm{i}(1 - \boldsymbol{\alpha}_i^t \mathrm{N}_{\mathrm{ii}})} \ i \in S$
Step 4. Increment $t = t + 1$, let $\widehat{\mathbf{w}} = \boldsymbol{\mu}_S$ and compute the residual
    $\mathbf{r} = \mathbf{y} - \mathbf{\Phi}_\mathrm{S} \widehat{\mathbf{w}}$
    Stop if $t = \vartheta$ (or $\|\mathbf{r}\|_2^2 < \omega$), otherwise go to Step 2
```

Figure 5-1. GBM algorithm for sparse estimation

Step 1: In the initialization step, we set the initial values for the hyperparameters $\boldsymbol{\alpha}^1 = (\boldsymbol{\alpha}_1^1,\ \boldsymbol{\alpha}_2^1, \dots, \boldsymbol{\alpha}_{\mathrm{n}}^1\,)^T = \mathbf{1}$, and $\sigma_1^2 = 0.01 \times var(\mathbf{y})$. Let the initial estimated support set be S = { }, its compliment be $S^c = \{1, \dots, n\}$, and the initial measurement residual vector be $\mathbf{r} = \mathbf{y}$. Matrix $\boldsymbol{\Phi}$ is also normalized to have unit norm, namely $\|\mathbf{g}_i\|_2 = 1$ ($\forall i = 1, \dots, n$), where $\mathbf{g}_i$ represents the i-th normalized column of matrix $\boldsymbol{\Phi}$.

Step 2: To estimate the support set, we derive a greedy rule which is consistent with the OMP algorithm. Unlike RVM that determines the support set S by globally solving Eq. (66) , which is computationally intensive for high dimensional problems, we propose that its upper bound,

$$\arg\min_{\mathbf{w}} \|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|_{\mathbf{2}}^{\mathbf{2}} + \sigma^2\left(\textstyle\sum_{i=1}^{n} g(\mathbf{w}_{\mathrm{i}}) + \log|\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{B}\boldsymbol{\Phi}^{\boldsymbol{T}}|\right) \qquad (68)$$

where $g(\mathbf{w}_{\mathrm{i}}) = \boldsymbol{\alpha}_{\mathrm{i}}\mathbf{w}_i^2$, should be iteratively minimized. The upper bound Eq. (68) can be easily verified by the definition of $g(\mathbf{w}) \leq \sum_{i=1}^{n} g(\mathbf{w}_{\mathrm{i}}) + \log|\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{B}\boldsymbol{\Phi}^{\boldsymbol{T}}|$. Clearly Eq. (68) is convex with respect to $\mathbf{w}$, hence we estimate the support set S by iteratively minimizing Eq. (68). Indeed, we solve the following optimization problem at each iteration (note we dropped the constant term $\log|\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{B}\boldsymbol{\Phi}^T|$ which is not relevant to the optimization),

$$\min_{\mathbf{w}_i} \|\mathbf{r} - \mathbf{g}_i\mathbf{w}_i\|_2^2 + \sigma^2 g(\mathbf{w}_i)\ \forall i \in S^c \qquad (69)$$

by taking the derivative of Eq. (69) with respect to $\mathbf{w}_i$ to 0. The optimal $\mathbf{w}_i{}^*$ is expressed as

$$\mathbf{w}_i{}^* = \frac{\mathbf{g}_i{}^T\mathbf{r}}{1+\sigma^2\ \boldsymbol{\alpha}_{\mathrm{i}}^1}\ \ \forall i \in S^c\ ; \qquad (70)$$

Note that we use the initial value for the hyperparameters $\boldsymbol{\alpha}_{\mathrm{i}}$ ($\forall i \in S^c$) since they are not estimated yet. Substitute Eq. (70) into Eq. (69), evaluate,

$$\|\mathbf{r} - \mathbf{g}_i\mathbf{w}_i{}^*\|_2^2 + \sigma^2\left[\ \boldsymbol{\alpha}_{\mathrm{i}}^1\mathbf{w}_i{}^{*2}\right] = \mathbf{r}^T\mathbf{r} - 2\frac{(\mathbf{g}_i{}^T\mathbf{r})^2}{1+\sigma^2\ \boldsymbol{\alpha}_{\mathrm{i}}^1} \qquad (71)$$
$$+\frac{(\mathbf{g}_i{}^T\mathbf{r})^2}{(1+\sigma^2\ \boldsymbol{\alpha}_{\mathrm{i}}^1)^2}(1+\sigma^2\ \boldsymbol{\alpha}_{\mathrm{i}}^1) = -\frac{(\mathbf{g}_i{}^T\mathbf{r})^2}{1+\sigma^2\ \boldsymbol{\alpha}_{\mathrm{i}}^1} + Constant,$$

and select the index with smallest cost function, which is equivalent to

$$i^* = \arg\max|\mathbf{g}_i{}^T\mathbf{r}|, \quad \forall i \in S^c \tag{72}$$

Then update the support set $S = S \cup \{i^*\}$ as well as $S^c = S^c \backslash \{i^*\}$.

Step 3: In this step, the nonzero coefficients $\hat{\mathbf{w}}$ associated with the estimated support set S are determined. We estimate these nonzero coefficients $\hat{\mathbf{w}}$ by minimizing the original Eq. (66). From the results of Wipf and Nagarajan [88], if the estimated support set is optimal (i.e., $S = \Lambda$), then $\boldsymbol{\mu}_S$ is the globally minimizer of Eq. (66). Following this result, in each iteration t, we utilize Bayesian learning procedures of RVM (presented in Appendix D) to achieve $\hat{\mathbf{w}} = \boldsymbol{\mu}_S$, which requires estimation of $\boldsymbol{\Sigma}_S$, $\boldsymbol{\alpha}_i^t(\mathrm{i} \in S)$, and σ_t^2 accordingly. Note that the procedures of RVM in estimating the support set is different than GBM (i.e. Step 2). However, if the greedy rule of GBM determines the right candidate (i^*) to the estimated support set S at each iteration, then clearly $\hat{\mathbf{w}} = \boldsymbol{\mu}_S$, where S is the estimated support set at the current iteration t, is always a local optimal of Eq. (66). In Theorem 1 presented in Sec. 5.3, we prove some theoretical conditions under which GBM determines the right candidate to the estimated support set S at each iteration. Thus, once the true support set is correctly estimated by GBM, i.e., $S = \Lambda$, then the GBM estimate $\hat{\mathbf{w}} = \boldsymbol{\mu}_S$ will be the globally minimizer of Eq. (66).

Step 4: In this step, the measurement residual vector $\mathbf{r} = \mathbf{y} - \boldsymbol{\Phi}_S\hat{\mathbf{w}}$ is updated. The previous steps (ii) through (iii) are continued either until ϑ iterations (assuming the sparsity level of $\mathbf{w}$ is known), or until $||\mathbf{r}||^2 < \omega$, where ω is a user-defined threshold (e.g.10^{-6}). The GBM algorithm is further summarized in Figure 5-1.

From Eq. (72), which delineates the GBM rule for determination of the support set S, it is evident that the GBM support set estimation is identical to the OMP procedure as outlined in Appendix C (i.e., selecting the columns of matrix $\boldsymbol{\Phi}$ with maximum correlation with measurement residual vector $\mathbf{r}$). However the other terms (in Step 3) are analogous to the Bayesian learning of RVM as evident from see Appendix D. The essential difference between GBM and OMP is in the update rule for residual vector $\mathbf{r}$.

The computational cost of GBM is comparable to OMP as they have the same support set estimation procedure (see Appendix C). According to Step 2 in Figure 5-1, a single candidate i^* is added to the estimated support set S, based on Eq. (72). Thus, the main portion of the GBM computational time relates to the inference part represented in Step 3. From the posterior covariance matrix $\boldsymbol{\Sigma}_S$, the matrix inversion operation has complexity $\mathcal{O}(|S|^3)$. It starts with $|S| = 1$ at the first iteration and progressively increases until convergence. Therefore, as opposed to RVM, GBM never requires inversion of an $n \times n$ matrix. This difference leads the computational cost of GBM to be markedly lower than purely Bayesian algorithms such as RVM.

5.3.2 Performance Guarantee for the Greedy Bayesian Method (GBM)

In the following, we will delineate the performance guarantee conditions for the proposed GBM algorithm. In other words, our aim is to obtain the conditions under which GBM algorithm leads to small sparse estimation error. These conditions mainly depend on certain properties of the training matrix $\boldsymbol{\Phi}$ as delineated from the existing literature on sparse estimation and compressive sensing [3]. One critical criterion condition is called *mutual coherence*; it is utilized to evaluate the suitability of matrix $\boldsymbol{\Phi}$ for successful sparse estimation. Mutual coherence $\tilde{\mu}$ is defined as follows:

$$\tilde{\mu} = \max_{i \neq j} \left| \mathbf{g}_i{}^T \mathbf{g}_j \right| \tag{73}$$

The general idea of mutual coherence $\tilde{\mu}$ is to seek low correlation between any two columns of matrix $\boldsymbol{\Phi}$. It is interesting to provide mutual coherence-based performance guarantee conditions under which GBM algorithm would have a satisfactorily small estimation error. We will show (in Theorem 1) that the performance of GBM under certain conditions, is at least as good an oracle estimator (OE). The OE is known as the best unbiased sparse estimator as it assumes that the true support set of $\mathbf{w}$, i.e. Λ is known as a priori, and uses the least squares (LS) method to estimate the nonzero coefficients of $\mathbf{w}$ whose index correspond to Λ. Let $\mathbf{w}_{\text{OE}}$ be the OE estimator for $\mathbf{w}$, then we have

$$\mathbf{w}_{\text{OE}} = \begin{cases} \mathbf{w}_{\text{OE},\Lambda} = (\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}\,\mathbf{\Phi}_{\Lambda}^{T}\mathbf{y} \\ \mathbf{w}_{\text{OE},\Lambda^{c}} = \mathbf{0} \end{cases} \tag{74}$$

where $\mathbf{w}_{\text{OE},\Lambda}$ represents the OE estimate of the coefficients on the true support set Λ which is given by LS method, and $\mathbf{w}_{\text{OE},\Lambda^{c}}$ is estimate of the coefficients on Λ^{c} which is set to a zero vector $\mathbf{0}$ with appropriate dimension $(n - |\Lambda|)$. OE is not a practical estimator since the true support set Λ is unknown in general. However, OE is typically used to compare the performance of different sparse estimation algorithms. Ben Haim et al. [5] provide an upper bound for OE performance as a function of the sparsity level ϑ, dimension n, true noise variance σ^{2}, and the mutual coherence $\tilde{\mu}$ as:

$$\|\mathbf{w}_{\text{OE}} - \mathbf{w}\|_{\mathbf{2}}^{\mathbf{2}} \leq \epsilon_{0} = 2\vartheta\sigma^{2}(1+\theta)\log n\,\frac{1}{(1-(\vartheta-1)\tilde{\mu})^{2}}.$$

where constant $\theta > 0$ is utilized to control the probability of bounding$\|\mathbf{w}_{\text{OE}} - \mathbf{w}\|_{\mathbf{2}}^{\mathbf{2}}$ (more details are provided in Appendix E, see Lemma 2).

To derive the theoretical results of GBM, the following notations will be used: let GBM estimate $\mathbf{w}_{\text{GBM}} = \begin{cases} \hat{\mathbf{w}} & on\ S \\ \mathbf{0} & otherwise \end{cases}$, where S is the estimated support set after the termination of GBM algorithm (here the termination procedure is based on $t = \vartheta$), $\boldsymbol{\alpha}_{max} = \max\ \boldsymbol{\alpha}_{S}$, where $\boldsymbol{\alpha}_{S}$ is the estimated hyperparameters associated with estimated support set S (for simplification we dropped the index t presented in Step 3 of Figure 5-1), and $\sigma_{\epsilon}^{2} = \sigma_{t}^{2}$ be the estimated noise variance after termination of GBM. In the following, we outline a mathematical proof (Theorem

1 below) delineating the performance guarantee conditions for GBM. We also show that the error bound of GBM is at least as good as that of OE under certain conditions.

Theorem 1. Let $\mathbf{w}_{\text{GBM}}$ be the GBM estimate of an unknown sparse vector $\mathbf{w}$ with true support set $\Lambda = \{i : \mathbf{w}_i \neq 0, \forall i = 1, \dots, n\}$ with known sparsity $|\Lambda| = \vartheta$, and let $\mathbf{y} = \mathbf{\Phi w} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim N(0, \sigma^2\mathbf{I}) \in \mathbb{R}^m$. Define $|\mathbf{w}_{min}|$ and$|\mathbf{w}_{max}|$ be the minimum absolute value and maximum absolute value of nonzero elements in $\mathbf{w}$, respectively, and $\tilde{\mu}$ be the mutual coherence of matrix $\mathbf{\Phi}$. If the dimension of measurement $m > \vartheta + \frac{\vartheta[1+(\vartheta-1)\tilde{\mu}+C][(1+(\vartheta-1)\tilde{\mu})^2\beta-1]}{2[1+(\vartheta-1)\tilde{\mu}]}$, where $C = \frac{m[1-(2\vartheta-1)\tilde{\mu}]^2}{\vartheta(1+\theta)\log(n)}$ for some positive constant θ and $\beta > \frac{4}{(1-(2\vartheta-1)\tilde{\mu})^2}$, and the following conditions hold,

$$|\mathbf{w}_{min}| \geq \vartheta_1 = \frac{2\sigma\sqrt{2(1+\theta)logn}}{1-(2\vartheta-1)\tilde{\mu}} \tag{75}$$

$$|\mathbf{w}_{max}| \leq \vartheta_2 = \sqrt{2\beta\sigma^2(1+\theta)logn} \tag{76}$$

then with a probability of at least

$$(1 - \frac{1}{\mathrm{n}^{\theta}\sqrt{\pi(1+\theta)\log n}}) \cdot (1 - 2e^{-\frac{m}{2}}) \tag{77}$$

GBM determines the true support set Λ of $\mathbf{w}$, and satisfies the quadratic error bound of

$$\|\mathbf{w}_{\text{GBM}} - \mathbf{w}\|_{\mathbf{2}}^{\mathbf{2}} < 2\vartheta\sigma^2(1+\theta)\log n \frac{1}{(1-(\vartheta-1)\tilde{\mu})^2} \tag{78}$$

Proof. The proof for Theorem 1 is provided in Appendix III.

The theoretical results derived in Theorem 1 are intended to indicate that if Eq. (75) holds GBM determines the true support set ($S = \Lambda$) with certain probabilities, and furthermore, under the condition of number of measurements and Eq. (76), the GBM quadratic error bound is at least as good as OE. In the following we emphasize some remarks, which are helpful for delineating the proposed OSEC approach.

Remark 1. In condition $|\mathbf{w}_{min}| \geq \vartheta_1 = \frac{2\sigma\sqrt{2(1+\theta)logn}}{1-(2\vartheta-1)\tilde{\mu}}$, the parameters σ, and ϑ are usually unknown in practice, θ is a constant that controls the probability of determining the true support set (see Lemma 2), and n is the dimension of coefficient vector **w** which cannot be changed. Thus the only parameter which can be changed is the mutual coherence $\tilde{\mu}$. In order to relax the condition ϑ_1 (make this bound smaller), we need to reduce $\tilde{\mu}$.

Remark 2. Condition of number of measurements $m > \vartheta + \frac{\vartheta[1+(\vartheta-1)\tilde{\mu}+C][(1+(\vartheta-1)\tilde{\mu})^2\beta-1]}{2[1+(\vartheta-1)\tilde{\mu}]}$ is interpreted as an upper bound on β. In other words, we relax the condition on m, and instead restrict an upper bound on β. The upper bound is derived as,

$$\beta < \xi = \frac{1}{(1+(\vartheta-1)\tilde{\mu})^2} + \left[\frac{m-\vartheta}{\vartheta}\right]\frac{2}{(1+(\vartheta-1)\tilde{\mu}+C)(1+(\vartheta-1)\tilde{\mu})} \tag{79}$$

Recall that from Theorem 1, we require $\beta > \frac{4}{(1-(2\vartheta-1)\tilde{\mu})^2}$, thus combining the lower bound and the upper bound we can define a range for β as $\frac{4}{(1-(2\vartheta-1)\tilde{\mu})^2} < \beta < \xi$, for which GBM estimate is at least as good as OE with a high probability.

Remark 3. In condition $|\mathbf{w}_{max}| \leq \vartheta_2 = \sqrt{2\beta\sigma^2(1+\theta)logn}$), again σ^2 is unknown, and θ and n are fixed. The only parameter can be changed is β. Hence, to relax $|\mathbf{w}_{max}| \leq \vartheta_2$ (make this bound larger), we need to increase the upper bound on β, namely, ξ. From Eq. (79), ϑ is unknown in practice, and m is fixed, thus the only parameter can be changed to increase ξ is the mutual coherence $\tilde{\mu}$. It can be easily verified that lower mutual coherence $\tilde{\mu}$, makes the upper bound ξ, and consequently ϑ_2 larger.

5.3.3 Performance Demonstration for the Proposed GBM Algorithm

We now test and verify the proposed GBM with respect to its theoretical results in Sec. 5.3.2, as well as its estimation accuracy and computational efficiency. The numerical example in this section is adopted from Ben-Haim, et al. [5], which is a benchmark simulation for performance evaluation of various sparse estimation algorithms compared with OE given their theoretical conditions.

The training matrix $\mathbf{\Phi}$ is chosen as the concatenation of two orthogonal matrices, namely, $\mathbf{H}$, accordingly, $\mathbf{\Phi} = [\mathbf{I}\ \mathbf{H}]$, where $\mathbf{I}$ is an identity matrix with dimension of 256×256, and $\mathbf{H}$ is a Hadamard matrix [5] with the same dimension. The mutual coherence of matrix $\mathbf{\Phi}$ is computed as $\tilde{\mu} = 1/16$. The sparsity level of $\vartheta = 5$ is assigned for sparse vector $\mathbf{w}$. The indices in the true support set Λ of sparse vector $\mathbf{w}$ are randomly chosen. To generate the corresponding nonzero values, two conditions should be held according to Theorem 1. The first condition restricts the minimum absolute value of nonzero elements $|\mathbf{w}_{min}| \geq \vartheta_1$ as presented in Eq. (75) in order to determine the true support set by GBM. The second condition requires Eq. (76) $|\mathbf{w}_{max}| \leq \vartheta_2$to ensure the error bound in Eq. (78). Note we require to have $\beta > \frac{4}{(1-(2k-1)\tilde{\mu})^2} = 20.9$ (by Theorem 1) where $\vartheta = 5$ and $\tilde{\mu} = 1/16$. Consequently, the nonzero values are generated randomly as follows,

$$\mathbf{w}_i = x\{u\vartheta_1 + (1-u)\vartheta_2\} - (1-x)\{u\vartheta_1 + (1-u)\vartheta_2\} \quad \forall i \in \Lambda$$

where $u \sim Unif(0,1)$,and $x \sim Bernoulli(p)$ with $P(x=1) = p$. In this setting, θ (Eqs. (75) and (76)) is assigned as one and equal probability is considered for positive or negative $\mathbf{w}_i$'s (i.e. $p = 1 - p = 0.5$).

From Eq. (76) and the range $\frac{4}{(1-(2\vartheta-1)\tilde{\mu})^2} < \beta < \xi$, we generate different cases by changing ϑ_2 through β values. Ten different values of β within the range of $21 \leq \beta \leq 200$ is considered in our simulation. We keep the noise variance $\sigma^2 = 10^{-2}$ during all the simulation cases. This simulation process is repeated 1,000 times for each case. The estimation performance of sparse vector estimators GBM, and OE are obtained. Mean square error is utilized as the estimation performance criterion to compare the performance of these estimators. The results are provided in Figure 5-2(a). It was observed that, OE slightly outperforms GBM for larger β values, since OE assumes that the true support set Λ is known beforehand. However, it can be observed that at lower values of β, namely in the range of $20.8 \leq \beta \leq 40.1$, the GBM performance is consistently superior to OE (the magnified portion is provided in Figure 5-2(b)).The reason for this superiority is that GBM results in a tighter performance bound given its theoretical conditions. Indeed, this behavior matches the theoretical results as explained in Remark 2, namely, in the range defined in Eq. (79), the GBM estimate is at least as good as OE (note that from Eq. (79), the upper bound ξ is computed 40.1).We note that different case studies can be generated based on changing matrix $\mathbf{\Phi}$ (i.e., mutual coherence $\tilde{\mu}$), sparsity level ϑ in a way that we are able to satisfy the conditions in Theorem 1.

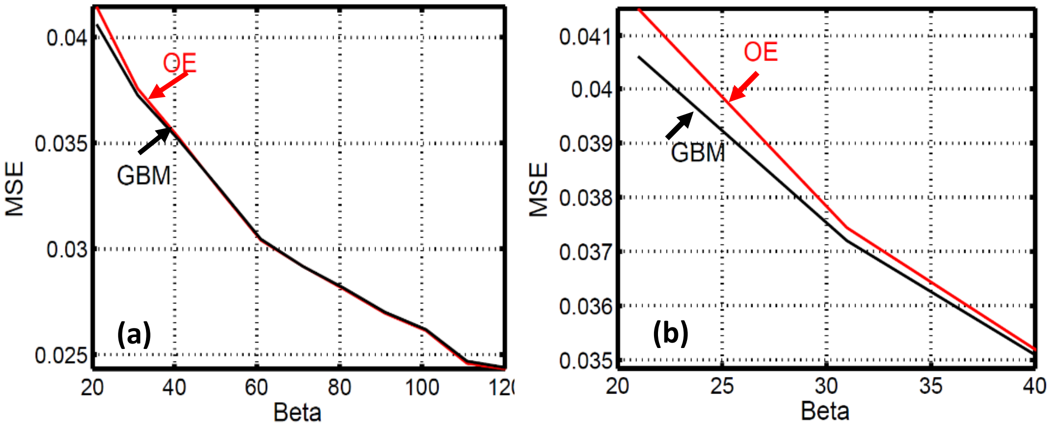

Figure 5-2: Mean square error (MSE) of OE and GBM for sparse estimation problems with matrix $\mathbf{\Phi}$, nonzero elements larger than Eq. (75), (a) $\sigma^2 = 10^{-2}$ and different $\boldsymbol{\beta}$ values, (b) $\boldsymbol{\beta}$ within its theoretical values ($\mathbf{20.8} \leq \boldsymbol{\beta} \leq \mathbf{40}$).

5.4 The Proposed Online Sparse Estimation based Classification Algorithm (OSEC)

The proposed OSEC algorithm consists of the following two phases:

(1) Phase 1 (Sec. 5.4.1) formulates the supervised classification problem into an underdetermined linear system of equations;

(2) Phase 2 (Sec. 5.4.2) applies the developed GBM algorithm (in Sec. 5.3) to perform sparse estimation to efficiently solve the underdetermined linear system obtained in Phase 1, and uses a *classification membership analysis* for classification of system state.

5.4.1 Phase 1: Formulation of Process States as Underdetermined Linear System from Sensor Data

Wright et al. [87] illustrate the use of labeled training data in the context of sparse representation based classification (SRC), i.e., the sensor data captured from the same condition is marked with the same label. SRC formulates the supervised classification problem into an *underdetermined linear system*. An underdetermined linear system is that in which the number of equations is smaller than the number of unknown variables. The approach devised by Wright et al. [87]considers that data belonging to the same label should have some intra-relationship that distinguishes them from data in other labels.

The key idea of SRC is to model this intra-relationship as an underdetermined system of linear equations. Consequently, any data belonging to a specific label can be represented as a linear combination of some of the training elements in that label. Mathematically, this entails, if $\mathbf{\Phi}_i = [\boldsymbol{\varphi}_i^1 \, \boldsymbol{\varphi}_i^2 \dots \boldsymbol{\varphi}_i^{n_i}] \in \mathbb{R}^{m \times n_i}$ $(m << n_i)$ represents a set of data belonging to a particular label (i), where n_i is the number of available data points in that label, and letting $\mathbf{y} \in \mathbb{R}^m$ be the observational data belonging to same label, due to the auto-correlation of data with the same label, we can then formulate $\mathbf{y}$ as a linear combination of the elements of $\mathbf{\Phi}_i$,

$$\mathbf{y} = \sum_{j=1}^{n_i} \boldsymbol{\varphi}_i^j \, \mathbf{w}_i^j \tag{80}$$

where $\boldsymbol{\varphi}_i^j$ represents the j-th data belonging to label i, and $\mathbf{w}_i^j$ are the coefficients utilized for this linear representation. However, two restrictive assumptions limit the practical viability of Eq. (80), namely,

(i) There may be dependencies (inter-relationships) between different labels. Hence, the implicit independence assumption in the formulation of Eq. (80) might be violated.

(ii) Moreover, other unknown factors (noise) could also influence Eq. (80), although their effect is assumed to be inconsequential.

In order to overcome the above shortcomings, Eq. (80) is re-expressed as,

$$\mathbf{y} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \boldsymbol{\varphi}_i^j \mathbf{w}_i^j \quad + \boldsymbol{\varepsilon} \tag{81}$$

In Eq. (81), it can be seen the effect of inter-label dependencies in representation of **y** is accounted with the second summation over index *i*. Indeed, variable *i* represents the label index and it is assumed that there are *k* such labels; also we note that *k* is known due to the intrinsic data sufficiency and observability assumptions of supervised classification. Moreover, the effect of unknown factors is accounted with the error term $\boldsymbol{\varepsilon} \in \mathbb{R}^m$; it is assumed that effect of the so called *noise term* $\boldsymbol{\varepsilon}$ is small. Simplifying Eq. (81) into a matrix form, the supervised classification problem is cast as an underdetermined system of linear equations that have the same form as Eq. (63). For convenience, it is re-written here as follows,

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{w} + \boldsymbol{\varepsilon} \tag{82}$$

where $\mathbf{y} \in \mathbb{R}^m$ denotes the observational data which is to be classified; $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ $(m << n)$ represents the *training matrix*, since it consists of the training data set; $\mathbf{w} \in \mathbb{R}^n$ is a vector containing the classification membership coefficients; and $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ represents the noise due to unknown and inconsequential effects. The training matrix is constructed as follows,

$$\boldsymbol{\Phi} \stackrel{\text{def}}{=} [\boldsymbol{\Phi}_1\ \boldsymbol{\Phi}_2\ \ldots\ \boldsymbol{\Phi}_\mathrm{k}] \tag{83}$$

where $\boldsymbol{\Phi}_i = [\boldsymbol{\varphi}_i^1, \boldsymbol{\varphi}_\mathrm{i}^2 \ldots \boldsymbol{\varphi}_i^{n_i}] \in \mathbb{R}^{m \times n_i}$ contains the training data available on the i-th label ($\forall i = 1, \ldots, k$), n_i is the size of i-th labeled training data, and the relationship between n and n_i is $\sum_{i=1}^{k} n_i = n$. Moreover, the classification membership coefficients are defined as,

$$\mathbf{w} \stackrel{\text{def}}{=} [\mathbf{w}_1{}^T\ \mathbf{w}_2{}^T\ \ldots\ \mathbf{w}_k{}^T]^T \tag{84}$$

where $\mathbf{w}_i = [\mathbf{w}_i^1\ \mathbf{w}_i^2 \ldots \mathbf{w}_i^{n_i}]^T \in \mathbb{R}^{n_i}$ is the vector containing the classification membership coefficients corresponding to the i-th label. For an ideal case, it is evident that if observation **y** truly belongs to label i, then the corresponding $\mathbf{w}_i$ should have nonzero elements, and consequently all other $\mathbf{w}_j$'s ($\forall j = 1,2.., k \backslash i$) should contain zero elements. Based on this intuition, it follows that vector **w** is *inherently sparse*. Hence the name *sparse representation classification* (SRC). Thus, based on Eqs. (82) through (84), a k-way classification can be performed for an observational data **y**. This can be done through estimation of **w**, i.e. $\mathbf{w}^*$. Noting that, Eq. (82) is an underdetermined system, since the number of equations m is smaller than that of the unknown variables n ($m << n$), consequently Eq. (82) has infinitely large number of solutions. The central premise is that vector **w** is sparse, i.e., most of its elements are zero. As explained in Sec. 5.2, starting with this sparsity assumption it is possible to approximate **w** using so-called sparse estimation algorithms; the estimate of **w** obtained therefrom is represented as $\mathbf{w}^*$ in this work. A sizeable focus of this work has been based on developing a novel algorithm (GBM) for approximating **w**, which is not only highly accurate, but also computationally efficient.

5.4.2 Phase 2: Classification Membership Analysis

To accommodate real-time monitoring in an advanced manufacturing scenario, which is typically attributed with high volume of data, and sampling rate of sensors, there is a need to solve Eq. (82) with a computationally efficient algorithm. Therefore, we equip the OSEC approach with the GBM algorithm developed in Sec. 5.3. Letting $\mathbf{w}^*$ be the GBM sparse estimation ($\mathbf{w}^* = \mathbf{w}_{\text{GBM}}$), our objective is to classify the observation $\mathbf{y}$ based upon $\mathbf{w}^*$, i.e., deciding the label (or process state) of an observation $\mathbf{y}$. Let

$$Supp = \{i{:}\, \mathbf{w}_i^* \neq 0, \forall i = 1, \dots, n\}$$

be a set containing the location of nonzero elements in $\mathbf{w}^*$, where $\mathbf{w}_i^*$ represents the i-th element of $\mathbf{w}^*$, and

$$\mathrm{T}_i \overset{\text{def}}{=} \{\textstyle\sum_{j=1}^{i} n_{j-1} + 1, \dots, \sum_{j=1}^{i} n_j\} \tag{85}$$

be the *index set* corresponding to the i-th label in our training data, where $n_{i-1} < n_i \in \mathbb{N}$ ($\forall i = 1, .., k$), and $n_0 = 0$. It can be shown that the index set T_i has the following properties,

$$\begin{aligned} |T_i| &= n_i && \text{(a)} \\ T_1 \cap T_2 \cap \dots \cap T_k &= \emptyset && \text{(b)} \qquad (86) \\ T_1 \cup T_2 \cup \dots \cup T_k &= \{1, \dots, n\} && \text{(c)} \end{aligned}$$

We reiterate, given the supervised classification used in this work, it is explicitly assumed that the labels of the index set T_i are known. In other words, we have *a priori* knowledge of the k labels from which the training data originate. Now supposing observation $\mathbf{y}$ belongs to the i-th label, then it is evident we expect $Supp \subseteq T_i$ (as defined in Eq. (**85**)). This is apparent for an ideal case where $\mathbf{y}$ can be fully represented as a linear combination of the i-th labeled training data. However, as explained previously in phase 1 (Sec. 5.4.1) this situation is improbable in practice due to the presence of inter-relationship between different labels, the presence of measurement noise, and the possible estimation errors resulting from the GBM algorithm. To overcome this drawback and make our approach practically viable, we propose the following *classification membership analysis*. The outline of the analysis is enumerated herewith:

(i) Let M_i be the intersection of $Supp$ with index sets T_i's

$$M_i = Supp \cap T_i \ \ (\forall i = 1,..,k) \tag{87}$$

and, $\mathbf{w}^*_{M_i}$ be a subvector of $\mathbf{w}^*$associated to the set M_i.

(ii) Compute the coefficient membership criterion

$$\widetilde{\mathbf{w}}_{\mathrm{i}} = \frac{\left\|\mathbf{w}^*_{M_i}\right\|_2^2}{\| \mathbf{w}^* \|_2^2} \quad (\forall i = 1,..,k) \tag{88}$$

where $\left\|\mathbf{w}^*_{M_i}\right\|_2^2$ represents the energy of the estimated coefficients corresponding to the i-th label, and $\| \mathbf{w}^* \|_2^2$ is the total energy of the estimated coefficients $\mathbf{w}^*$. Then use Lemma 1 result below.

Lemma1. Let $\widetilde{\mathbf{w}}_{\mathrm{i}}$ be defined in Eq. (88), M_i be defined in Eq. (87). Then M_i and $\widetilde{\mathbf{w}}_{\mathrm{i}}$ $(\forall i = 1,..,k)$ consist of the following properties (proof of Lemma 1 is omitted from the subchapter)

$$M_i \ (\forall i = 1,..,k) \text{ are mutually exclusive} \tag{89}$$

$$\widetilde{\mathbf{w}}_{\mathrm{i}} \geq 0 \ (\forall i = 1,..,k) \tag{90}$$

$$\sum_{i=1}^{k} \widetilde{\mathbf{w}}_{\mathrm{i}} = 1 \tag{91}$$

From Lemma 1, it is evident that $\widetilde{\mathbf{w}}_{\mathrm{i}}$ is akin to a probability mass function. Thus, using Eq. (88), we can interpret the result of GBM estimation as the probability that GBM estimation classifies **y** into label i, by the following rule

$$\lambda = \underset{1 \leq i \leq k}{\operatorname{argmax}} \ \widetilde{\mathbf{w}}_{\mathrm{i}} \tag{92}$$

Essentially, the Eq. (92) entails assigning the index (or label) with the maximum probability for an observation **y**. With the last phase of the proposed OSEC approach thus elucidated, we now proceed to apply the approach to experimentally acquired multi-sensor data.

5.5 Application of the Proposed OSEC for Advanced Manufacturing Processes

This section demonstrates the proposed methodology (OSEC) in practical manufacturing related scenarios. We compare the performance OSEC approach in terms of classification accuracy with popular machine learning approaches as measured using the F-score metric. Accordingly, we use sensor-data two different advanced manufacturing processes.

- Sec. 5.5.1: A fused filament fabrication (FFF) additive manufacturing (AM) process. The data consists of signals from 7 sensors acquired at sampling rate of 2.5 Hz.
- Sec. 5.5.2: A semiconductor chemical mechanical planarization (CMP) process used for polishing blanket copper wafers to specular finish. The data consists of a two axis wireless vibration sensor signal acquired at a sampling rate of 685 Hz.

The data used in these studies is acquired from experiments conducted at our facilities, the experimental details are available in the literature [2, 85]. Each of the datasets presented are analyzed using OSEC, whose performance is compared against five popular machine learning approaches frequently used in the signal analysis and process prognosis domains. The approaches used for comparison are: (i) quadratic discriminant analysis (QDA); (ii) support vector machine (SVM); (iii) neural network (NN); (iv) k nearest neighborhood (k-NN); and (v) naïve Bayes (NB). To ensure equitable comparison, the classification algorithms operate on the same training and validation data sets as OSEC, the classification performance is evaluated based on a common criterion called the F-score [2], which is defined as:

$$F_i = 2 \cdot \frac{(P_i Se_i)}{P_i + Se_i} \tag{93}$$

where P_i and Se_i are *precision* and *sensitivity* of classification, respectively. In our studies, the average F-score obtained from 100 realizations are reported.

5.5.1 Case Study for Fused Filament Fabrication (FFF) Additive Manufacturing (AM) Process

We now apply the OSEC approach to experimental data acquired from a fused filament fabrication (FFF) additive manufacturing (AM) process. The experimental data is relatively high dimensional, consisting of 13 sensor signals, albeit acquired at a low sampling rate of 2.5 Hz.

5.5.1.1 Experimental Studies with FFF Process

Fused Filament Fabrication (FFF) is an additive manufacturing (AM) process in which an object is manufactured by depositing progressive layers of extruded semi-molten material [89, 90](see Figure 5-3). In FFF a thermoplastic material is heated past its glass transition temperature and extruded through a nozzle in a controlled manner (Figure 5-3(a)). In the absence of real-time process monitoring, quality control in FFF is largely limited to offline techniques, leading to high scrap rates [90]. In this work we investigate the means to detect improper extrusion of the material. We instrumented the MakerBot Replicator 2X desktop FFF machine with multiple sensors (see Figure 5-3 (b)). The technical details of the various sensors used in this work are provided elsewhere [2]. We acquired temperature and vibration information at several different locations on the machine (see Figure 5-3 (c)-(e)).

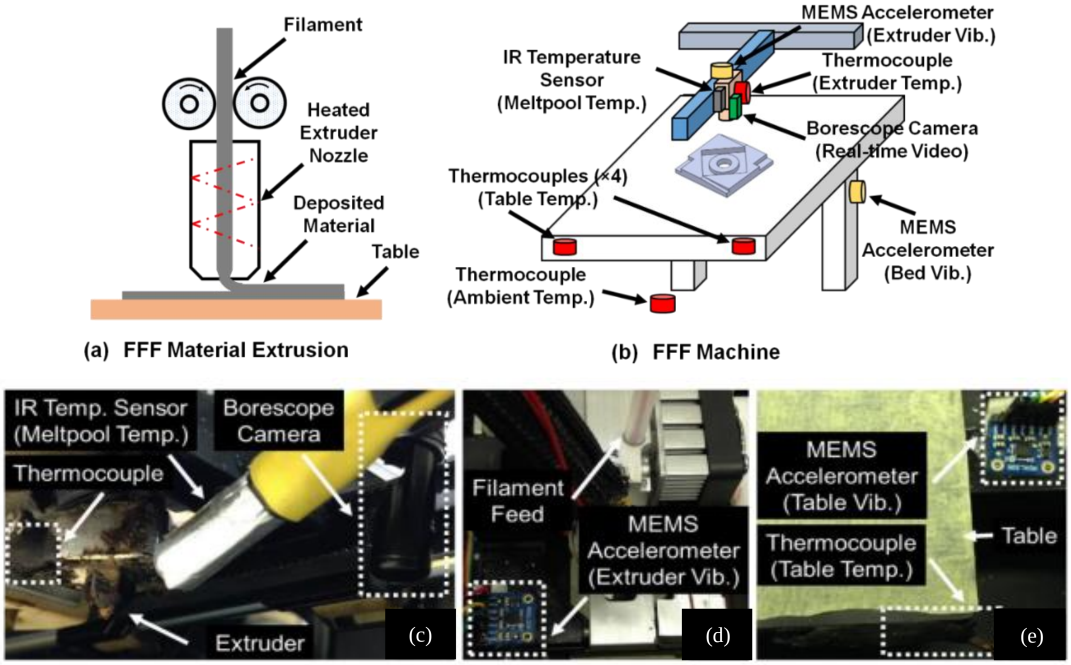
Filament
Heated Extruder Nozzle
Deposited Material
Table
(a) FFF Material Extrusion
MEMS Accelerometer (Extruder Vib.)
IR Temperature Sensor (Meltpool Temp.)
Thermocouple (Extruder Temp.)
Borescope Camera (Real-time Video)
Thermocouples (×4) (Table Temp.)
MEMS Accelerometer (Bed Vib.)
Thermocouple (Ambient Temp.)
(b) FFF Machine
IR Temp. Sensor (Meltpool Temp.)
Borescope Camera
Thermocouple
Extruder
(c)
Filament Feed
MEMS Accelerometer (Extruder Vib.)
(d)
MEMS Accelerometer (Table Vib.)
Thermocouple (Table Temp.)
Table
(e)

Figure 5-3: (a) Schematic of the FFF process. (b) Schematic of the FFF setup instrumented with multiple *in situ* for measuring process conditions in real-time. (c) – (e): Photographs of various sensors used in this study [2].

Data from the six thermocouples and two vibration sensors (three channels each) is collated in real-time. These twelve different sensor channels are synchronized and recorded at a sampling rate of ~ 2.5 Hz. Data from the infrared (IR) meltpool temperature sensor has a sampling rate of 1 Hz. Thus in total, thirteen channels of sensor data are acquired. In this work we will use seven channels of sensor data from the total thirteen, namely, the six vibration channels (three each from bed and extruder) and one IR temperature sensor channel. This is because, the thermocouples are not directly representative of changes in process dynamics.

From our intensive experimental studies [2], we have observed that one of the main quality issues with our FFF machine, which may lead to a failed part, essentially relates to the filament extrusion process. The borescope video (see Figure 5-3(c)) has been used to record the entire printing process for each run. This camera gives a real-time live video feed of the nozzle and extrusion process, which allows us to monitor the extrusion process visually from start to finish. From our intensive experimental studies, we have determined some inappropriate process parameters which lead to the nozzle being clogged, such as printing under low flow rate (rate of extrusion of material) to feed rate (movement of the extruder head) ratio in conjunction with low layer height (thickness of the deposited extrudate). We have carefully used the borescope video to monitor the filament extrusion of the above printing process. Based on the visual analysis with the borescope video, three process states are identified *a priori* (please see the top panel in Figure 5-4): (1) normal operation of the process (Figure 5-4(a)); (2) abnormal operation due to insufficient extrusion (Figure 5-4 (b)); and (3) failure (stoppage) of extrusion due to nozzle clog (Figure 5-4 (c)).

Nozzle clog occurs due to improper selection of process conditions. The onset of extrusion failure due to a nozzle clog is a gradual process. Initially, the process remains in the *normal* state for the initial third of the layers (~ 15 minutes out of 45 minutes) with stable extrusion. Thereafter the extrusion tends to be intermittent; at this stage the extrudate is relatively less consistent (almost stringy), and we term this the *abnormal* process state. Finally, the extrusion process completely stops or *fails* on account of a nozzle clog. It would be valuable to detect the process drift as it progresses from normal to abnormal, so that opportune corrective action can be taken and part build failure can be avoided. The corresponding IR temperature sensor signals acquired during these phases are juxtaposed in the bottom panel of Figure 5-4. These signal trends show a clear decreasing trend vis-à-vis the change in process state as it changes from normal to abnormal and finally, failure (stoppage) state. The overarching aim of this work is to construct a classifier for the three process states delineated above, using the data acquired from different sensors and then to apply this classifier for real-time process monitoring.

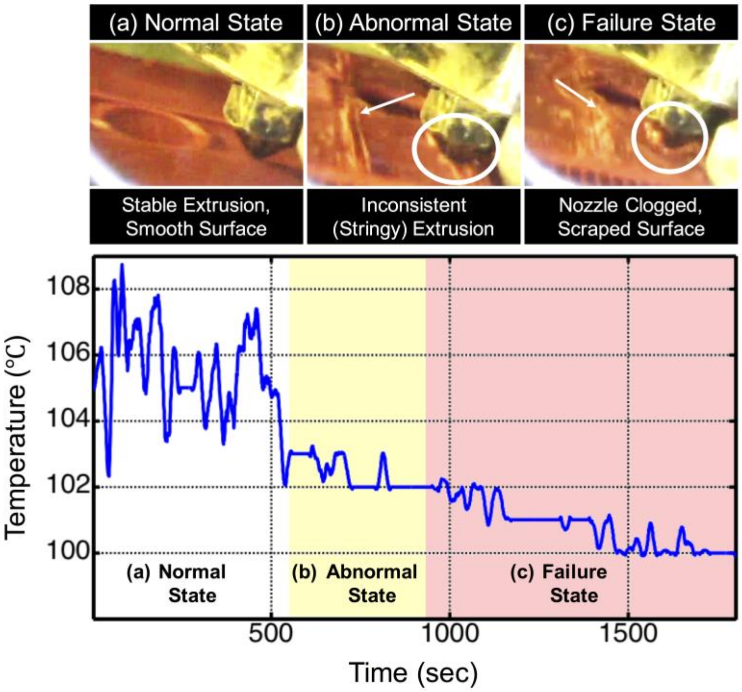

Figure 5-4: The three observed process states in FFF, demarcated as normal, abnormal, and stoppage due to nozzle clog [2]. The bottom panel shows the corresponding IR sensor signals.

5.5.1.2 Application and Verification of OSEC Approach for FFF Process Monitoring

In this section, we present the classification performance for AM process using our OSEC approach. As explained in Sec. 5.5.1.1, there exists three conditions (or labels, i.e., $k = 3$), in AM process, namely, normal extrusion condition, abnormal extrusion condition, and stoppage condition (failure). The OSEC approach is implemented as follows:

Phase 1: We formulate the underdetermined linear system $\mathbf{y} = \mathbf{\Phi w} + \boldsymbol{\varepsilon}$, where $\mathbf{\Phi}$ is the pre-labelled training matrix, given sensor observation $\mathbf{y}$. There are a total of 401, 331, and 901 data available on normal, abnormal, and failure conditions, respectively. We randomly split the available labeled data into training data set and validation data set in the ratio of 70 to 30 percent, respectively. The size of training data for each label are n_1= 280, n_2 = 231, and n_3 = 630. Thus, from Eq. (85)   (**85**), we have $T_1 = \{1, \dots, 280\}, T_2 = \{281, \dots, 511\}, T_3 = \{512, \dots, 1141\}$.

Next, we construct the training matrix as $\mathbf{\Phi} = [\mathbf{\Phi}_1\ \mathbf{\Phi}_2\ \mathbf{\Phi}_3]$, where $\mathbf{\Phi}_1 \in \mathbb{R}^{7\times280}, \mathbf{\Phi}_2 \in \mathbb{R}^{7\times231}$, and $\mathbf{\Phi}_3 \in \mathbb{R}^{7\times630}$ represent the randomly selected training data for labels 1, 2, and 3, respectively (the index 7 represents the number of sensors used in the work). Hence, the sequence of the columns in matrices $\mathbf{\Phi}_{\mathrm{i}}'s$ ($\forall i = 1,2,3$) is not important anymore. Consider a single training data $\boldsymbol{\varphi}_i^j \in \mathbb{R}^m$ randomly selected from the historical data belonging to label i, then it does not matter where we locate $\boldsymbol{\varphi}_{\mathrm{i}}^j$ in matrix $\mathbf{\Phi}_{\mathrm{i}}$ (it can be located as any column of matrix $\mathbf{\Phi}_{\mathrm{i}}$). In this way, we would be able to reduce the correlations in the columns of matrix $\mathbf{\Phi}$ (lower mutual coherence as suggested by Remark 1 and 3).

Phase 2: We solve the underdetermined linear system $\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \boldsymbol{\varepsilon}$ using GBM. In our FFF studies we have 121 + 100 + 271 = 492 data to be used for validation. Let us denote them with $\mathbf{Y} = \{\mathbf{y}^{t1}, \dots \mathbf{y}^{t_q}, \dots, \mathbf{y}^{t_{\mathbf{492}}}\}$, where $\mathbf{y}^{t_q}$ represents the q-th validation data set. Then, in each time, we let $\mathbf{y} = \mathbf{y}^{t_q}$ ($\forall q = 1, ..492$), and solve $\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \boldsymbol{\varepsilon}$ using GBM. From the GBM estimation results $\hat{\mathbf{w}}$, we perform classification membership analysis. In other words, we compute λ by Eq. (92) for all 492 validation cases. The average F-score over 100 replications of the above procedure are reported for each FFF process state. The following inferences can be drawn from the results summarized in Table 5-1.

(1) It is evident that OSEC has the highest average F-score in comparison to other approaches studied. Additionally, we noted the average computational time for classification of a single observation; this is estimated to be close to 0.7 milliseconds for the OSEC. Thus, OSEC can be utilized for real-time monitoring of AM process due to its computationally efficiency, as well as accurate classification performance as reported in Table 5-1. Indeed, the OSEC approach is able to obtain an accurate estimate (F-score > 90%) well within 0.2% of the time period of the signal (sampling rate 2.5 Hz).

(2) The kNN algorithm and NN algorithm perform relatively well on classification of normal and faulty conditions. However their performance for the abnormal condition is poor. This could

be due to the abnormal condition closely resembling both normal condition and extrusion failure condition in its early stage. Detecting the abnormal stage is therefore a challenge for existing classification algorithms to detect successfully.

(3) Although the reported average F-score for the abnormal condition is lower than for other conditions, yet OSEC performs significantly better than the other algorithms against which it is compared for the abnormal process state.

Table 5-1. Average F-score and computational time for classification of FFF process states. (SRC stands for sparse representation classification)

	Process States			
Algorithms Tested	**Normal Extrusion**	**Abnormal Extrusion**	**Failure**	**Computational Time (Milliseconds)**
OSEC (GBM)	**0.942**	**0.891**	**0.950**	**0.71**
SRC (CO)	**0.903**	**0.783**	**0.917**	**140**
SRC (RVM)	**0.921**	**0.822**	**0.942**	**81.1**
SRC (OMP)	**0.875**	**0.81**	**0.89**	**0.83**
QDA	**0.896**	**0.543**	**0.882**	**1.5**
SVM	**0.799**	**0.611**	**0.848**	**11**
NN	**0.911**	**0.614**	**0.896**	**3.4**
kNN	**0.917**	**0.578**	**0.902**	**0.14**
NB	**0.915**	**0.629**	**0.882**	**0.022**

(4) We further extended the classification of FFF process states using other sparse estimation algorithms, namely, convex optimization, OMP, and RVM, which are incorporated into the SRC framework, to solve the underdetermined linear system. They are referred to as SRC (CO), SRC (RVM), and SRC (OMP) in Table 5-1.

(5) From Table 5-1, it is observed that the performance of OSEC is superior to the SRC approach equipped with other sparse estimation algorithms. In addition to that, the computational time (in seconds) related to OSEC is the lowest compared to the SRC approach equipped with other sparse estimation algorithms. This analysis confirms that not only GBM algorithm alleviates

the computational cost associated with real monitoring of FFF process, but also improves the classification performance.

(6) Pertinently, the classification performance in terms of F-score of the OSEC approach and SRC approach, regardless of what sparse estimation algorithm it is equipped with, was found to be superior in comparison with other algorithms studied, such as SVM, QDA, k-NN, NN, and NB.

5.5.2 Case Study for Chemical Mechanical Planarization (CMP) of Copper Wafers

We proceed to apply the OSEC approach to CMP data consisting of two *in situ* wireless vibration signals acquired at 685 Hz.

5.5.2.1 Experimental Studies with CMP process

CMP is a free abrasive process similar to lapping [85, 91, 92], which is widely used in the semiconductor industry for finishing dielectric and metal interconnect layers patterned on semiconductor wafers. The dimensional consistency of these interconnect layers is crucial to the functional performance of semiconductor devices, and should therefore be tightly controlled during manufacture [84, 93]. Since, a typical modern semiconductor integrated circuit (IC) has 7 to 10 interconnect layers each requiring a CMP step, CMP induced variations in layer thickness can severely impede device functionality. Hence, real-time monitoring of CMP process is vital for ensuring functional integrity of semiconductor devices [84, 93, 94].

In this work, sensor data is obtained from a benchtop CMP apparatus (Buehler model Automet 250) instrumented with a wireless two channel MEMS accelerometer (vibration sensor) [85], whose sampling rate is ~ 685 Hz (Figure 5-5(a) and (b)). Cylindrical copper wafers of diameter 40.6250 mm (1.625 in.), and thickness 12.5 mm (0.5 in.) are polished on this apparatus. Scratch free, near-optical finish with R_a ~ 5 nm is reported in this work (Figure 5-5 (c) and (d)). In our previous work, we identified certain optimal CMP process conditions [3].

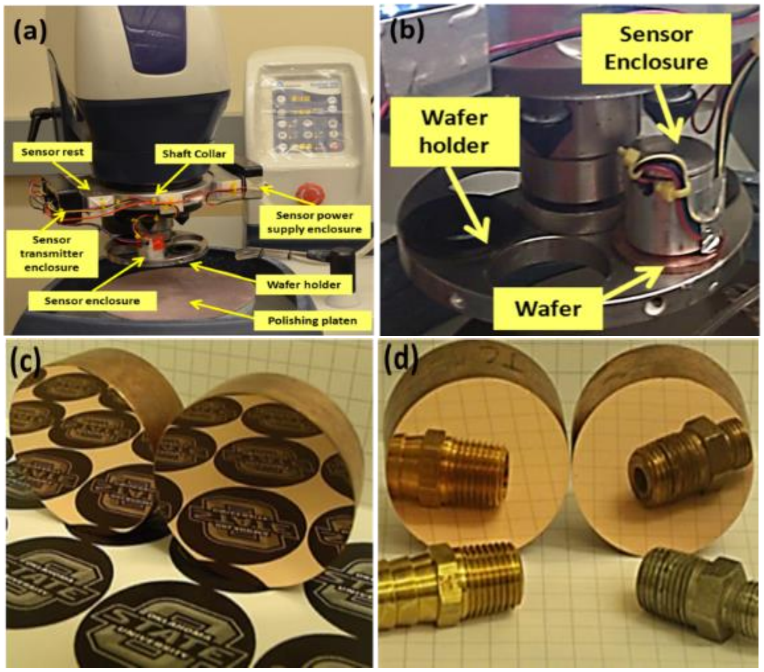

Figure 5-5: Description of CMP apparatus and results. (a) Buehler polishing apparatus instrumented with wireless vibration sensors. (b): Close-up view of the wireless vibration sensing setup. (c) and (d): Specular finished copper wafer with surface roughness in the range of Ra ~ 5 nm.

The vibration data acquired from the CMP process is complex [85]. Thus, it is challenging to detect evanescent process drifts from complex data using traditional statistics and data mining approaches. In this study we analyzed signals from both controlled sub-optimal and optimal CMP process conditions; representative vibration signals for some suboptimal CMP process conditions are shown in Figure 5-6.

The objective is to ascertain the performance of the OSEC approach for classifying CMP various process states. Sub-optimal conditions include, for instance: polishing at high pressure and high velocity; worn out pads (Figure 5-6(a)); insufficient supply of polishing compounds (Figure 5-6 (b)); introduction of foreign particles (Figure 5-6(c)). These suboptimal conditions damage the wafer surface and often lead to deep scratches impeding device yield rates.

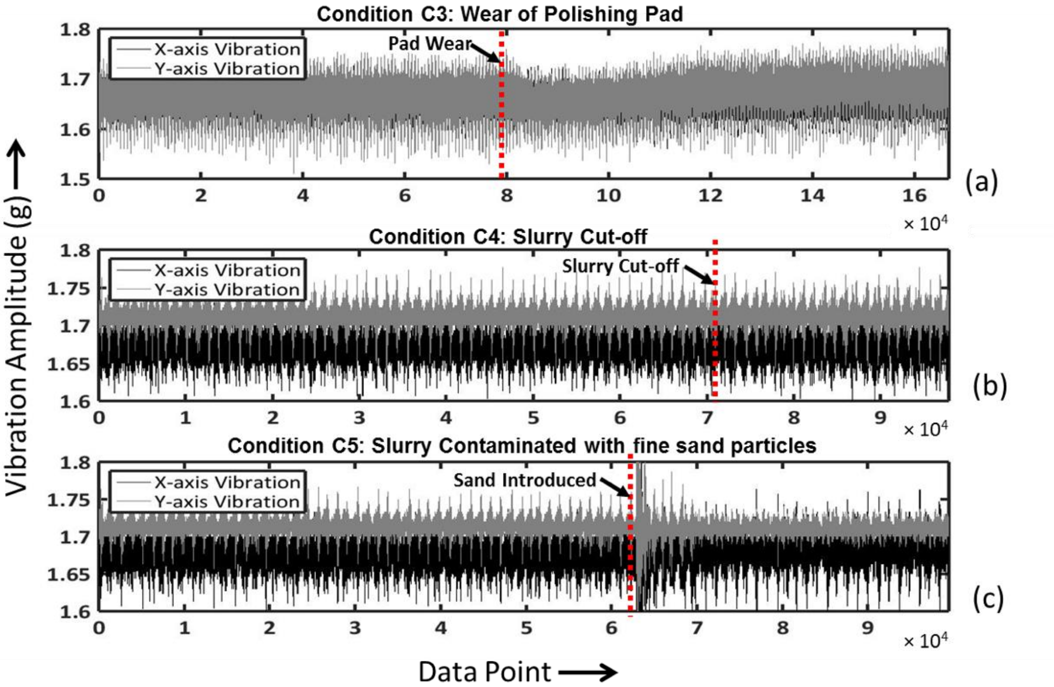

Figure 5-6: MEMS wireless vibration signals (685 Hz sampling rate) in the tangential (x-axis) and radial directions (y-axis) acquired from the experimental copper CMP test bed for three types of process faults. (a) Condition C3: CMP with a worn out polishing pad, note the gradual change in signal amplitude with time. This fault causes deep scratches on the wafer (b) Condition C4: polishing slurry is suddenly cut-off during the process, the changes are not visibly prominent in the time domain, however, subtle change in the frequency domain were observed (not shown). This condition resulted in scratches on the wafer. (c) Condition C5: polishing slurry contaminated with foreign sand particles. Note the sudden spike when the particles contact the wafer. This fault caused deep scratches (> 20 nm deep) on the wafer.

Accordingly, there are in total nine experimental CMP conditions studied (Table 5-2). These are labelled C1 through C9, of which C6 and C8 are considered near-optimal conditions. Each of these experimental conditions is replicated 5 times. For brevity, we abstain from providing extensive details of the experimental conditions; these are detailed in Refs. [85]. The objective is to classify these nine conditions, which is a practically relevant problem, because knowing the status of the process from sensor data will allow an operator to take corrective action. In the forthcoming section, we will demonstrate that the OSEC approach can classify these nine process conditions with higher fidelity compared to conventional machine learning techniques.

Table 5-2: Summary of CMP experimental process conditions to be classified. Barring C6 and C8 all other process conditions are suboptimal.

Condition Label	Condition Summary
C1	Workpiece loses contact with polishing pad (no material removal).
C2	Polishing with excessive load (8 lb.) with worn out pad, leaving scratches on the wafer.
C3	Polishing at low load (2 lb.) condition with worn pad leaving deep scratches on wafer
C4	Polishing at low load (2 lb.), slurry supply is suddenly throttled resulting in suboptimal wafer surface finish and damage to polishing pad.
C5	Polishing at low load (2 lb.) with introduction of fine sand particles in the slurry. Deep scratches are observed on the wafer.
C6	2 lb. load and 30 RPM wafer speed. Optimal condition.
C7	8 lb. load and 30 RPM wafer speed. Suboptimal condition leading to poor wafer uniformity.
C8	2 lb. load and 60 RPM wafer speed. Near optimal condition.
C9	8 lb. load and 60 RPM wafer speed. Highest material removal rate, suboptimal condition leading to poor wafer uniformity and shearing of pad material.

5.5.2.2 Application and Verification of OSEC approach for CMP Process Monitoring

The aim of this section is to classify process states C1 and C9 using the OSEC approach, and compare its performance with existing approaches as in the preceding AM case. As before, we describe the application of the OSEC as stratified in three phase:

Phase 1: We formulate the underdetermined linear system $\mathbf{y} = \mathbf{\Phi w} + \boldsymbol{\varepsilon}$, where $\mathbf{\Phi}$ is the pre-labelled training matrix, given sensor observation $\mathbf{y}$. However, for the CMP case, because the sampling rate approaches 685 Hz, the size of the available dataset is considerable. Consequently, we present our classification study based upon different sizes of training data on each label, namely n_i = {1000, 2000, 5000}. These training data lengths are predicated by the physical characteristics of the sensing system; the wireless sensors used in this work have a transmit-receive acknowledgment cycle of 1 second (685 data points). Hence choosing a training length of less than 685 data points is not physically congruous given system constraints. For each training size n_i's, we define the index set $T_i's$ $(\forall i = 1, \ldots, 9)$ by Eq. (**85**). The training data on each label $\mathbf{\Phi}_i$ are randomly selected from the available data. Hence, the training matrix is formed as $\mathbf{\Phi} = [\mathbf{\Phi}_1 \ \ \mathbf{\Phi}_2 \ldots \ \mathbf{\Phi}_9]$. The validation data set $\mathbf{y}$ are also selected randomly out of the remaining sensor data (with the size of 250 on each label).

Phase 2: We solve the underdetermined linear system $\mathbf{y} = \mathbf{\Phi w} + \boldsymbol{\varepsilon}$ using GBM. In our studies we have $9 \times 250 = 2250$ data to be used for validation. We denote them with $\mathbf{Y} = \{\mathbf{y}^{t1}, \ldots \mathbf{y}^{t_q}, \ldots, \mathbf{y}^{t_{\mathbf{2250}}}\}$, where $\mathbf{y}^{t_q}$ represents the q-th validation data set. Then, in each time, we let $\mathbf{y} = \mathbf{y}^{t_q}$ $(\forall q = 1, \ldots, 2250)$, and solve $\mathbf{y} = \mathbf{\Phi w} + \boldsymbol{\varepsilon}$ using GBM. From the GBM estimation results $\widehat{\mathbf{w}}$, we perform classification membership analysis. In other words, we compute λ by Eq. (92) for all 2250 validation cases.

We repeat all the procedures explained above to achieve 100 realizations for classification analysis of the CMP process. We also analyze the data using other classification algorithms on the same training and validation data sets as above. The average F-score is presented in Table 5-3 along with computational efficiency in Table 5-4. The following inferences can be drawn based on Table 5-3 and Table 5-4:

(1) OSEC is significantly superior in terms of F-score among all other classification algorithms studied. Moreover, increasing the size of training data improves the classification performance of OSEC, at the risk of increasing detection delay.

(2) Indeed, the highest average F-score for the OSEC algorithm is obtained for the training size of $n_i = 5000$. Therefore, increasing the size of the training data could be helpful, albeit at the cost of increased computational load and detection delay incurred by increasing the

Table 5-3: Average F-score for classification algorithms based on different $n_i's$ for CMP process. C6 and C8 are near-optimal CMP process conditions.

F-score for Training Data Size = 1000									
Process Condition	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
OSEC	**0.929**	**0.937**	**0.913**	**0.946**	**0.931**	**0.954**	**0.953**	**0.826**	**0.834**
SRC (CO)	**0.87**	**0.912**	**0.905**	**0.924**	**0895**	**0.956**	**0.96**	**0.813**	**0.789**
SRC (RVM)	**0.889**	**0.872**	**0.941**	**0.932**	**0.922**	**0.965**	**0.941**	**0.851**	**0.862**
SRC (OMP)	**0.825**	**0.812**	**0.892**	**0.785**	**0.842**	**0.941**	**0.875**	**0.812**	**0.812**
QDA	**0.563**	**0.741**	**0.402**	**0.553**	**0.217**	**0.308**	**0.613**	**0.534**	**0.295**
SVM	**0.424**	**0.525**	**0.398**	**0.023**	**< 0.10**	**0.651**	**< 0.10**	**0.676**	**0.572**
NN	**0.548**	**0.754**	**0.438**	**0.622**	**< 0.10**	**0.642**	**< 0.10**	**0.24**	**0.623**
kNN	**0.586**	**0.786**	**0.281**	**0.613**	**0.476**	**0.493**	**0.573**	**0.576**	**0.446**
NB	**0.532**	**0.757**	**0.325**	**0.558**	**0.213**	**0.268**	**0.618**	**0.314**	**0.494**
F-score for Training Data Size = 2000									
OSEC	**0.969**	**0.962**	**0.950**	**0.978**	**0.966**	**0.974**	**0.970**	**0.915**	**0.901**
SRC (CO)	**0.912**	**0.932**	**0.914**	**0.932**	**0.912**	**0.932**	**0.972**	**0.882**	**0.882**
SRC (RVM)	**0.924**	**0.923**	**0.955**	**0.942**	**0.941**	**0.971**	**0.967**	**0.892**	**0.912**
SRC (OMP)	**0.845**	**0.865**	**0.924**	**0.825**	**0.889**	**0.975**	**0.895**	**0.889**	**0.864**
QDA	**0.549**	**0.748**	**0.337**	**0.560**	**0.480**	**0.283**	**0.648**	**0.218**	**0.466**
SVM	**0.442**	**0.604**	**0.373**	**0.039**	**< 0.10**	**0.652**	**< 0.10**	**0.69**	**0.606**
NN	**0.544**	**0.789**	**0.393**	**0.575**	**0.206**	**0.506**	**0.494**	**0.61**	**0.572**
kNN	**0.569**	**0.79**	**0.291**	**0.609**	**0.463**	**0.422**	**0.603**	**0.365**	**0.487**
NB	**0.526**	**0.759**	**0.270**	**0.593**	**0.471**	**0.226**	**0.628**	**0.267**	**0.533**
F-score for Training Data Size = 5000									
OSEC	**0.99**	**0.992**	**0.977**	**0.996**	**0.984**	**0.988**	**0.992**	**0.951**	**0.965**
SRC (CO)	**0.964**	**0.962**	**0.962**	**0.948**	**0.942**	**0.953**	**0.984**	**0.965**	**0.945**
SRC (RVM)	**0.975**	**0.972**	**0.989**	**0.981**	**0.972**	**0.982**	**0.994**	**0.912**	**0.941**
SRC (OMP)	**0.892**	**0.912**	**0.941**	**0.872**	**0.921**	**0.994**	**0.932**	**0.925**	**0.894**
QDA	**0.564**	**0.745**	**0.298**	**0.562**	**0.5**	**0.31**	**0.65**	**0.2**	**0.482**
SVM	**0.422**	**0.634**	**0.393**	**0.069**	**< 0.10**	**0.672**	**< 0.10**	**0.71**	**0.6053**
NN	**0.548**	**0.787**	**0.386**	**0.573**	**0.254**	**0.474**	**0.572**	**0.603**	**0.505**
kNN	**0.58**	**0.786**	**0.297**	**0.61**	**0.438**	**0.451**	**0.591**	**0.354**	**0.491**
NB	**0.542**	**0.759**	**0.242**	**0.5977**	**0.477**	**0.225**	**0.632**	**0.256**	**0.551**

dimension of matrix $\mathbf{\Phi}$. This is especially important for real-time monitoring of CMP process where the sampling rate could be as high as 685 Hz.

(3) Therefore, in order to study the tradeoff between computation time vs training data size, we compared the computational time for the proposed OSEC algorithm given different n_i values.

The average computational cost of a single observation classification is in Table 5-4. It is evident from Table 5-4 that a larger training set size (n_i) incurs higher computational costs. For $n_i = 5000$ OSEC is slowest due to the large dimension of the training matrix yet commensurate with the sampling rate in CMP process (685 Hz). Therefore, the training size of 5000 on each label would be appropriate due to its high classification performance. Additionally, we compared the computational time of OSEC with SRC approach equipped with different sparse estimation algorithms, as well as other supervised classification algorithms such as NN, kNN, and SVM. Our results indicate that the OSEC approach is the fastest among the SRC approach equipped with different sparse estimation algorithms. For example, it was observed that for the smallest training size ($n_i = 1000$), SRC (CO) is more than 1000 times slower than OSEC approach. This is because of the fact the GBM algorithm incorporated into OSEC framework is computationally more efficient than convex optimization algorithm, and even RVM algorithm. This analysis confirms that not only GBM algorithm alleviates the computational cost associated with real monitoring of CMP process, but also improves the classification performance.

(4) Even with a training data set of 5000 data points, one classification task is completed within about 0.9 milliseconds using the proposed OSEC approach, i.e., within 60% the time it takes for a data point to be recorded (1.5 milliseconds, given 685 Hz sampling rate).

Table 5-4: OSEC's computational time given different training size on each label (sensor sampling rate 685 Hz)

	Computational Time (Milliseconds)								
Training Data Size	OSEC	SRC (CO)	SRC (RVM)	SRC (OMP)	QDA	SVM	NN	kNN	NB
1000	0.410	520	23.700	0.690	0.030	7.800	0.510	0.016	0.002
2000	0.591	1080	39.400	0.890	0.041	19.900	1.50	0.024	0.004
5000	0.854	4330	321.9	1.021	1.702	1235.54	103.6	0.824	0.012

5.6 Conclusions and Future Work

In this subchapter, we proposed a supervised classification approach based on the concept of sparse estimation for real-time monitoring applications. The proposed online sparse estimation based classification (OSEC) formulates the supervised classification problem as an underdetermined system of linear equations. To accommodate real-time monitoring in advanced manufacturing scenarios, we further developed a fast sparse estimation algorithm in this subchapter, namely, the greedy Bayesian method (GBM), and incorporated it into the OSEC approach to efficiently solve the underdetermined system.

We validated the OSEC approach for monitoring of two advanced manufacturing processes, namely, a fused filament fabrication (FFF) additive manufacturing process; and a semiconductor chemical mechanical planarization (CMP) process. This is valuable for condition monitoring of advanced manufacturing processes. Specific outcomes from the practical case studies (AM and CMP, Sec. 5.5.1 and Sec. 5.5.2, respectively) illustrated in this work are as follows:

(1) For the fused filament fabrication (FFF) additive manufacturing process, the existing three conditions (labels), namely, normal extrusion condition, abnormal extrusion condition, and extrusion stoppage condition are monitored. The classification performance of OSEC is measured and compared with some of the existing classification algorithms for verification purposes. The classification results show that the proposed OSEC algorithm has superior performance in comparison to other algorithms studied. F-score performance for the proposed OSEC approach was between 89% and 95%.

(2) We further verified the OSEC approach with semiconductor chemical mechanical planarization (CMP) process. Nine CMP process conditions were used for classification. Data from two vibration sensors are gathered with a sampling rate approaching 685 Hz. With the

proposed OSEC algorithm, classification performance consistently approaches 90% (F-score), in contrast the best performance amongst all of the other traditional supervised classification algorithms studied is at most 75%. Moreover, the computational time for OSEC was close to 0.85 milliseconds, which is well within the sampling time (15 milliseconds) of the sensing system. Thus, the OSEC approach is successfully applied to a multiple sensor monitoring scenario having high sampling rates.

In summary, the proposed OSEC algorithm is shown to be computationally efficient, as well as more accurate than traditional machine learning algorithms against which it was compared, viz., k-NN, QDA, NB, NN, SVM) for real-time monitoring of manufacturing processes with high dimensional training matrix (exceeding 45000). The number of sensors utilized for manufacturing applications is usually small (< 25), for the cases where a larger number of sensors, such as in weather monitoring where more than 1000 sensors can be active simultaneously, the complexity of the proposed method can be overwhelming. A prescription could be utilizing a random sampling from the training matrix $\mathbf{\Phi}$ [3]. We are currently augmenting the OSEC approach for multivariate big data mining applications.

Chapter 6. Wearable Sensor Selection Optimization with Applications to Manual Material Handling Tasks

Occupational jobs often involve different types of manual material handling (MMH) tasks. Performing such tasks can be physically demanding, and which may put workers at an increased risk of work-related musculoskeletal disorders (WMSDs). To control and prevent WMSDs, there has been a growing interest in online posture monitoring using wearable sensors. In this Chapter, we utilize OSEC algorithm for monitoring and evaluation of MMH activities, as it was shown in Chapter 5 that it is computationally efficient for online decision making. We further propose an optimization approach to improve classification performance, by differentially weighting sensor, thereby representing the relative of a sensor in classification performance. As such, optimizing these weights enables us to determine the most relevant sensors for classification. A case study using 37 sensors with 111 channels of data was completed to validate performance of the proposed method. With only 30 optimally-selected sensor channels, our method provides high classification accuracy (>84%) and outperforms several benchmark methods, including support vector machine, quadratic discriminant analysis, and neural network[5].

6.1 Introduction

Occupational tasks often involve different types of manual material handling (MMH), requiring workers to, for example, lift, carry, push/pull, and manipulate an object. Performing such tasks can be physically demanding, and contributes substantially to work-related musculoskeletal disorders (WMSDs) reported annually in the United States [95]. WMSDs are an important issue

[5] Bastani, K., Kim, S., Kong, Z., and Nussbaum, M., 2016, “Online classification and sensor selection optimization with applications to human material handling tasks using wearable sensing technologies,” *IEEE Transactions on Human-Machine Systems* (Accepted).

in the workplace, which accounts for a substantial portion (>30%) of lost workday cases [96, 97]. To control and prevent WMSDs, high risk conditions for a given job need to be identified to develop effective interventions [98, 99]. Identification of such conditions ideally requires monitoring and understanding worker behaviors during MMH tasks in an actual work environment, since specific working strategies that individual or groups of workers use to perform MMH tasks can affect the level of physical exposures experienced. Understanding and characterizing MMH strategies used by workers *in situ* can thus aid in the design and evaluation of interventions targeted for high risk work strategies and/or situations, and further help improve the effectiveness of interventions since intervention effectiveness can depend on the individual worker and their specific situations or context [100].

To efficiently monitor and assess working strategies, there is growing interest in human activity and posture monitoring, such as using wearable sensors [101-103] or depth cameras [93, 104]. Self-report and observational methods (e.g., checklists) are often used in practice to monitor activities and postures, yet they are labor-intensive and provide relatively crude outputs [105, 106]. Using wearable sensors such as inertial measurement units (IMUs), along with automatic activity classification, has great potential to enable rapid, comprehensive assessment of physical demands in diverse work settings, providing detailed information on body kinematics (e.g., joint angles) and work demands / strategies (e.g., distribution or frequency of task types during a work shift, or temporal changes in a given task type). Previous studies on activity classification with wearable sensors have reported classification accuracy of ~30% to ~99% [102, 107, 108]. In these studies, poorer performance was generally related to larger movement variability and/or less distinct motion characteristics. These performance-limiting conditions may be common in an occupational setting (e.g., healthcare, construction, and service industries), since workers are often required to perform diverse types of manual tasks.

Therefore, one challenge in promoting the use of IMUs (or similar wearable sensors) for physical exposure assessments is to improve task classification performance. Using such sensors to monitor whole-body kinematics can provide high dimensional time series data, for which accurate and timely online processing is challenging and computationally extensive. To effectively overcome these challenging issues, we recently developed a supervised classification algorithm based on a computationally efficient sparse estimation method to support online decision making for manufacturing processes [109]. Here, we extend the application of this algorithm to improve classification performance for MMH tasks, by incorporating an optimization approach based on sparse estimation performance criteria. Specifically, given that each sensor captures different aspects of body kinematics, we differentiate levels of importance of sensors to improve classification performance. Another advantage of the proposed optimization method is that it can be utilized as a sensor selection methodology to reduce the number of sensors, yet maintain satisfying classification performance.

The remainder of the paper is organized as follows: Sec. 6.2 provides the research background required for MMH task classification. In Sec. 6.3, we propose our optimization approach and its extension as a sensor selection methodology for online MMH task classification. Sec. 6.4 presents the effectiveness of the proposed approaches through case studies. Finally, Sec. 6.5 concludes the paper.

6.2 Research Background

Use of wearable technologies with online MMH task classification can facilitate real-time physical exposure assessments. Some studies have indeed explored a potential of integrating wearable sensors with classification algorithms to monitoring MMH tasks or other occupationally relevant activities. In this section, existing and related studies are reviewed and their limitations are discussed. To overcome those limitations, we utilize OSEC algorithm proposed in Chapter 5 due to its fast computational speed, as well as its accurate classification performance.

Wearable sensors have been widely used to classify physical activities in the fields of healthcare-related research and manufacturing assembly lines [102, 107, 108, 110-120]. In these and related report, diverse supervised classification algorithms have been to recognize and associate patterns in wearable sensor data to specific physical activities. Note, the reader is referred to Preece et al. [119] and references therein for more detailed review. Algorithms used have included k-nearest neighborhood (k-NN) [112, 116, 121], support vector machines (SVM) [110, 120], neural networks (NN) [108, 118], linear discriminant analysis (LDA) [107], and Naïve Bayes classifiers [102, 114, 116, 121].

The working procedures of such classification algorithms vary from simple heuristic based approaches (e.g., k-NN) to more advanced approaches such as NN and SVM. Differences in working procedures of these classifiers likely results in observed differences in classification performance. Hence, there have been a number of studies comparing different classifiers for activity classification in the literature [107, 116, 118, 120]. From such reported results, k-NN and NN often perform better than other classifiers. However, this conclusion cannot be established as a general conclusion for activity classification, since performance of the classification algorithms noted above depend substantially on the preprocessing procedures applied to wearable sensor data (i.e., feature extraction). Input variables to these classifiers are the features extracted initially from the wearable sensor data, typically represented by statistical features, frequency-domain features, time-domain features, etc. [111, 117, 122]. Feature extraction techniques usually result in high dimensional data, with some additional methods used subsequently for dimensionality reduction [113, 115]. The extracted low(er)-dimensional features are then imported into the classifier, to determine the specific physical activity label.

If the features extracted from obtained wearable sensor data are not sufficiently representative, these classification algorithms likely have relatively poor performance. Different feature extraction techniques and dimensionality reduction methods have been utilized to provide the input features [98, 100, 103, 104, 111]. However, there is no consensus on which features lead to better (or worse) classification performance for physical activity identification, and there is little practical guidance to help guide feature selection. Therefore, it would be very useful to apply a classification algorithm for MMH task recognition that is not dependent on the selected features. Furthermore, the computational cost of the classification algorithm should be low enough to be conveniently utilized for online classification (such as MMH tasks, of interest here).

6.3 Optimization Problems for Online MMH Task Classification

To improve the classification performance of the OSEC algorithm, we first need a criterion to measure OSEC classification performance. Since there is no a priori knowledge that a new observation $\mathbf{y}$ correctly belongs to a specific label, we cannot utilize classification error as the criterion. Hence, there is a need for an indirect measure to quantify classification performance. Given the nature of the OSEC algorithm, classification performance depends on how well the sparse estimation of Eq. (82) is carried out. Thus, we consider existing performance criteria used for sparse estimation [71, 79, 123]. These criteria depend mainly on some properties of matrix $\mathbf{\Phi}$, introduced in Sec. 6.3.1. Then, Sec 6.3.2 presents the proposed optimization method to improve sparse estimation/classification performance, by optimizing the weighting of each individual sensor. In Sec. 6.3.3, we extend the proposed optimization problem by incorporating some physical interpretation for the weights assigned to sensors, and which leads to a sensor selection methodology.

6.3.1 Properties of Matrix $\boldsymbol{\Phi}$

Two effective criteria have been reported to evaluate the properties of matrix $\mathbf{\Phi}$ for sparse estimation problems, namely (1) restricted isometry property (RIP) [71, 123], and (2) mutual coherence [79]. It is defined that matrix $\mathbf{\Phi}$ satisfies RIP condition of order υ if we have an isometry constant $0 < \delta_\upsilon < 1$ such that

$$(1-\delta_\upsilon) \leq \frac{\|\mathbf{\Phi}_S\mathbf{w}_S\|_2^2}{\|\mathbf{w}_S\|_2^2} \leq (1+\delta_\upsilon) \tag{94}$$

is satisfied for every index set S with cardinality of υ. The RIP condition requires that any sub-matrix $\mathbf{\Phi}_\mathrm{S}$ from the original matrix $\mathbf{\Phi}$ is nearly orthogonal as δ_υ becomes very small. It has been proved that if the RIP holds then sparse estimation algorithms can approximate $\mathbf{w}$ within a very small error bound with high probability [76, 124]. It has also been shown that a random matrix $\mathbf{\Phi}$ (e.g., sampled from a Gaussian distribution) satisfies the RIP condition with high probability [4]. But for an arbitrary matrix $\mathbf{\Phi}$, it is NP-hard to compute the isometry constant δ_υ [5].

The second criterion to evaluate matrix $\mathbf{\Phi}$ is mutual coherence, and which is more computationally tractable than the RIP criterion. Let $\mathbf{G} = \widetilde{\mathbf{\Phi}}^{\boldsymbol{T}}\widetilde{\mathbf{\Phi}}$ be the Gram matrix of $\widetilde{\mathbf{\Phi}}$ computed from matrix $\mathbf{\Phi}$ after normalizing each of its columns [6]. Then, mutual coherence $\mu(\mathbf{\Phi})$ is defined as the largest absolute off-diagonal elements of matrix $\mathbf{G}$, namely,

$$\mu(\mathbf{\Phi}) = \max\left|g_{ij}\right| \; 1 \leq i,j \leq n, i \neq j \tag{95}$$

where g_{ij} represents the i-th element of the j-th column of matrix $\mathbf{G}$. Mutual coherence measures the highest correlation between columns of matrix $\mathbf{\Phi}$. From Eq. (95), clearly, mutual coherence μ can be efficiently computed, as opposed to δ_k which is not computationally tractable. Essentially, a large $\mu(\mathbf{\Phi})$ represents a vulnerability of matrix $\mathbf{\Phi}$ for sparse estimation, since columns that are highly correlated could easily deteriorate the performance of sparse estimation algorithms [6]. Therefore, matrices with low mutual coherence are required for accurate sparse estimation. Some theoretical results on mutual coherence reported in Refs. [80, 81] prove that if the inequality

$$\|\mathbf{w}\|_0 \leq \frac{1}{2}(1 + \frac{1}{\mu(\mathbf{\Phi})}) \qquad (96)$$

holds, then vector $\mathbf{w}$ is necessarily the sparsest representation of $\mathbf{y} = \mathbf{\Phi w}$ (non-noisy measurements), and some sparse estimation techniques are guaranteed to succeed in reconstruction of $\mathbf{w}$. Thus, given inequality Eq. (96), minimizing mutual coherence $\mu(\mathbf{\Phi})$ yields a wider set of vectors $\mathbf{w}$ to be reconstructed successfully. However, Elad [6] argues that minimizing the proposed mutual coherence (Eq. (95)) may not improve the actual performance of sparse estimation techniques; but, an average measure of mutual coherence does. Based on Elad's study, we consider average mutual coherence defined as follows,

$$\mu_{avg}(\mathbf{\Phi}) = \frac{\sum_{1\leq i,j\leq n, i\neq j} |g_{ij}|}{n(n-1)/2} \qquad (97)$$

Indeed, $\mu_{avg}(\mathbf{\Phi})$ computes the average of absolute off diagonal elements of gram matrix $\mathbf{G}$ as opposed to Eq. (95) which considers the worst case scenario with the highest correlation. As discussed above, it is necessary to minimize $\mu_{avg}(\mathbf{\Phi})$ to improve the sparse estimation performance/classification performance. In Sec. 6.3.2, we present two optimization problems dealing with $\mu_{avg}(\mathbf{\Phi})$, by assigning different weights to different sensor channels.

6.3.2 Optimization for Training Matrix *Φ*

In the proposed linear model (Eq. (82)), the training matrix $\mathbf{\Phi}$ is constructed from time series data associated to each label (see Eq. (83)). Due to the nature of times series data, columns of matrix

$\mathbf{\Phi}$ from the same label should be correlated. Thus, $\mu_{avg}(\mathbf{\Phi})$ is expected to be high. Thus, matrix $\mathbf{\Phi}$ may be vulnerable for sparse estimation, as discussed in Sec. 6.3.1.

We propose an optimization method that aims to minimize $\mu_{avg}(\mathbf{\Phi})$, by assigning different weights to each row (i.e., each sensor channel) of matrix $\mathbf{\Phi}$. Let $\mathbf{z} = \{\mathbf{z}_1\ \mathbf{z}_2 \dots\ \mathbf{z}_\mathrm{m}\}^T$ be the decision variables, where $\mathbf{z}_j$ represents the weight assigned to the j-th row/sensor $(\forall j = 1, \dots, m)$. We define

$$\mathbf{\Phi}(\mathbf{z}) = \begin{bmatrix} \mathbf{z}_1\mathbf{\Phi}_1 \\ \vdots \\ \mathbf{z}_m\mathbf{\Phi}_m \end{bmatrix} \tag{98}$$

where $\mathbf{\Phi}_j$ is the j-th row of matrix $\mathbf{\Phi}$ $(\forall j = 1, \dots, m)$. Each row/sensor weight $\mathbf{z}_j$ is assumed to be between [0, 1]. Now, the optimization problem is formulated as,

$$\begin{gathered} \min \mu_{avg}(\mathbf{\Phi}(\mathbf{z})) \\ \mathbf{z} = \{\mathbf{z}_1\ \mathbf{z}_2 \dots\ \mathbf{z}_\mathrm{m}\}^T \\ \text{S.T.: } 0 \le \mathbf{z}_j \le 1\ \ (\forall j = 1, \dots, m) \end{gathered} \tag{99}$$

To determine the underlying solution approach to Eq. (99), the objective function and constraints should be characterized. The constraints of $0 \le \mathbf{z}_j \le 1\ \ (\forall j = 1, \dots, m)$ are convex, as they are linear, and in the following lemma we prove that $\mu_{avg}(\mathbf{\Phi}(\mathbf{z}))$ is a nonconvex sum-of-ratios function of variable $\mathbf{z}$.

Lemma 1- Let $\mathbf{\Phi}(\mathbf{z}) \in \mathbb{R}^{m \times n}$ be a function of $\mathbf{z} \in \mathbb{R}^m$, as defined in Eq. (98), and $\mu_{avg}(\mathbf{\Phi}(\mathbf{z}))$ be the average mutual coherence of $\mathbf{\Phi}(\mathbf{z})$, then $\mu_{avg}(\mathbf{\Phi}(\mathbf{z}\ ))$ is a nonconvex sum-of-ratios function of variable $\mathbf{z}$.

Proof. Following the definition of average mutual coherence, matrix $\mathbf{\Phi}(\mathbf{z})$ must first be normalized. It can be normalized by introducing the normalization matrix

$$\mathbf{N}(\mathbf{z}) = diag(\textstyle\sum_{i=1}^{m} \mathbf{z}_i^2 {q_{ij}}^2)^{1/2}\ (\forall i = 1, \dots, n) \tag{100}$$

where q_{ij} is the i-th element of matrix $\boldsymbol{\Phi}$ at its j-th column. The normalized matrix is below,

$$\widetilde{\boldsymbol{\Phi}}(\mathbf{z}) = \boldsymbol{\Phi}(\mathbf{z})\mathbf{N}(\mathbf{z}) \tag{101}$$

Then, the Gram matrix of $\widetilde{\boldsymbol{\Phi}}(\mathbf{z})$ is defined as

$$\mathbf{G}(\mathbf{z}) = \widetilde{\boldsymbol{\Phi}}(\mathbf{z})^T \boldsymbol{\Phi}(\mathbf{z}) \tag{102}$$

From the definition, average mutual coherence $\mu_{avg}(\boldsymbol{\Phi}(\mathbf{z}))$ is the average of absolute off-diagonal elements of Gram matrix $\mathbf{G}(\mathbf{z})$. By matrix multiplication and simple algebra, the absolute off-diagonal elements of the Gram matrix can be expressed as

$$g_{ij} = \frac{\left|\sum_{k=1}^{m} \mathbf{z}_k^2 q_{ki} q_{kj}\right|}{\sqrt{\sum_{k=1}^{m} \mathbf{z}_k^2 {q_{ki}}^2}\sqrt{\sum_{k=1}^{m} \mathbf{z}_k^2 {q_{kj}}^2}})\ 1 \leq i,j \leq n, i \neq j \tag{103}$$

Then, average mutual coherence is a simple average of Eq. (103) over $(1 \leq i,j \leq n, i \neq j)$, namely,

$$\mu_{avg}\big(\boldsymbol{\Phi}(\mathbf{z})\big) = \sum_{1 \leq i,j \leq n, i \neq j} \frac{\left|\sum_{k=1}^{m} \mathbf{z}_k^2 q_{ki} q_{kj}\right|}{\sqrt{\sum_{k=1}^{m} \mathbf{z}_k^2 {q_{ki}}^2}\sqrt{\sum_{k=1}^{m} \mathbf{z}_k^2 {q_{kj}}^2}} \tag{104}$$

Note that we dropped the irrelevant term $n(n-1)/2$ as it is constant for our optimization problem. From Eq. (104), it can be seen that it is a sum-of-ratio function of $\mathbf{z}$, where both $\left|\sum_{k=1}^{m} \mathbf{z}_k^2 q_{ki} q_{kj}\right|$ and $\sqrt{\sum_{k=1}^{m} \mathbf{z}_k^2 {q_{ki}}^2}\ \sqrt{\sum_{k=1}^{m} \mathbf{z}_k^2 {q_{kj}}^2}$ are convex with respect to **z.** It is clear that the numerator of Eq. (104) is a convex function in $\mathbf{z}$. The denominator consists of two terms $\sqrt{\sum_{k=1}^{m} \mathbf{z}_k^2 {q_{ki}}^2}$, and $\sqrt{\sum_{k=1}^{m} \mathbf{z}_k^2 {q_{kj}}^2}$ which are realized as weighted norm of vector $\mathbf{z}$. By triangular inequality and the definition of a convex function, it can be shown that each of these terms is convex in $\mathbf{z}$. Since both terms are non-decreasing functions, the product of these two terms is also convex. Thus, the ratios of two convex functions is nonconvex, and this completes the proof. **Q.E.D**.

Following Lemma 1, $\mu_{avg}\big(\mathbf{\Phi}(\mathbf{z})\big)$ is a nonconvex sum-of-ratios in variable $\mathbf{z}$. Thus Eq. (99) becomes a nonconvex optimization problem and is difficult to solve. However, Jong [125] developed a practical global optimization algorithm for nonconvex sum-of-ratios problems with convex constraints. Hence, we utilize this algorithm as the solution approach for Eq. (99). This algorithm transforms the sum-of-ratios problem into a convex parametric programming problem and finds the global solution successfully. Utilizing this algorithm for a given training matrix $\mathbf{\Phi}$, we can always improve (reduce) its average mutual coherence. Thus, performance of the sparse estimation based classification can be improved correspondingly. This can be done by modifying the underdetermined linear system proposed in Eq. (82), leading to an improved OSEC algorithm. The procedures of this algorithm is exactly the same as OSEC algorithm, but the only difference is that instead of solving Eq. (82), improved OSEC algorithm proposes to solve

$$\mathbf{y}(\bar{\mathbf{z}}) = \mathbf{\Phi}(\bar{\mathbf{z}})\mathbf{w} + \boldsymbol{\varepsilon} \tag{105}$$

where $\bar{\mathbf{z}}$ is the optimal solution to Eq. (99), $\mathbf{y}(\bar{\mathbf{z}}) = \begin{bmatrix} \bar{\mathbf{z}}_1\mathbf{y}_1 \\ \vdots \\ \bar{\mathbf{z}}_m\mathbf{y}_m \end{bmatrix}$, and $\mathbf{y}_j$ represents the j-th element of $\mathbf{y}$ ($\forall j = 1, \dots, m$). Matrix $\mathbf{\Phi}(\bar{\mathbf{z}})$ in Eq. (105) benefits from lower average mutual coherence than $\mathbf{\Phi}$ in Eq. (82) Consequently, we can conclude that solving Eq. (105) rather than Eq. (82) is expected to improve the sparse estimation of $\mathbf{w}$. Note that if the noise term was known beforehand, we should have replaced the left hand side of Eq. (105) with $\mathbf{y}' = \mathbf{y}(\bar{\mathbf{z}}) + (1 - \bar{\mathbf{z}})\boldsymbol{\varepsilon}$. Since the noise term $\boldsymbol{\varepsilon}$ is assumed to be small, the difference between $\mathbf{y}'$and $\mathbf{y}(\bar{\mathbf{z}})$ is negligible. Hence, we can still use the solution of Eq. (105) for our classification membership analysis.

6.3.3 Sensor Selection Methodology

In Sec. 6.3.2 we showed that, by assigning different weights ($\mathbf{z}_j$ in Eq. (99)) to each row of the design matrix, the average mutual coherence of the training matrix can be reduced. Here, we would like to select the sensors that contribute more to classification performance. This is particularly useful to reduce the number of wearable sensors, and which could subsequently reduce costs associated with sensors and any potential sensor discomfort to workers.

The sensor selection methodology is formulated as selecting u out of m potential sensors, to minimize average mutual coherence. We propose the sensor selection problem as follows

$$\begin{gathered} \min \mu_{avg}(\mathbf{\Phi}_P) \\ \text{S.T.: } |P| = u \end{gathered} \tag{106}$$

where P is a selection subset of the sensor set $\{1, \dots, m\}$, with cardinality of $|P| = u$; and $\mathbf{\Phi}_P$ is a submatrix of $\mathbf{\Phi}$ where its rows corresponds to P. Let $\mathbf{z} = \{\mathbf{z}_1\ \mathbf{z}_2 \dots \mathbf{z}_\mathrm{m}\}^T$ be the decision variable where $\mathbf{z}_j$ $(\forall j = 1, \dots, m)$ represents whether sensor j-th should be selected ($\mathbf{z}_j = 1$), or not ($\mathbf{z}_j = 0$). Using $\mathbf{z}$ with row-wise multiplication on matrix $\mathbf{\Phi}$ as defined in Eq. (98), we can rewrite Eq. (106) as follows

$$\begin{gathered} \min \mu_{avg}(\mathbf{\Phi}(\mathbf{z})) \\ \text{S.T.: } \mathbf{1}^T\mathbf{z} = u \\ \mathbf{z}_j \in \{0,1\}\ (\forall j = 1, \dots, m) \end{gathered} \tag{107}$$

where $\mathbf{1}$ is a vector with all elements as one, and constraint $\mathbf{1}^T\mathbf{z} = u$ ensures that the number of selected sensor is u. One method to solve Eq. (107) is to try all $\binom{m}{u}$ choices to evaluate performance. However, this is clearly not practical unless m or u is very small. It can be shown that the sensor selection problem is NP-hard [126]. Therefore, to make it solvable in polynomial time, we relax the nonconvex constraints $\mathbf{z}_j \in \{0,1\}(\forall j = 1, \dots, \mathrm{m})$ with convex constraints

$0 \leq \mathbf{z}_j \leq 1$ in Eq. (107). In other words, we assign different weights, between zero and one, to various sensors/rows. Now, the relaxed optimization problem is presented as

$$\begin{gathered} \min \mu_{avg}(\boldsymbol{\Phi}(\mathbf{z})) \\ \text{S.T.}: \mathbf{1}^T\mathbf{z} = u \\ 0 \leq \mathbf{z}_j \leq 1 \;\; (\forall j = 1, \ldots, m) \end{gathered} \tag{108}$$

In Lemma 1, we showed that $\mu_{avg}(\boldsymbol{\Phi}(\mathbf{z}))$ is characterized as a nonconvex sum-of-ratios function of optimization variable $\mathbf{z}$. The constraints added for the purpose of sensor selection are convex. Therefore, the optimization problem in Eq. (108) can still be solved by the solution approach in Ref. [125].

Note that the optimal solution to Eq. (108), denoted by $\mathbf{z}^*$, can be fractional as opposed to integer solution to Eq. (107). Thus, these two problems are not equivalent. Let μ^* be the optimal objective value of Eq. (107). Then, it can be shown that the optimal objective value of Eq. (108), is a *lower bound* on μ^*, namely,

$$L_{\mu^*} = \mu_{avg}(\boldsymbol{\Phi}(\mathbf{z}^*)) \tag{109}$$

This can be realized through the fact the feasible region in Eq. (108) contains the feasible region of Eq. (107), and therefore the optimal objective value of Eq. (108) cannot be larger than μ^*. On the other hand, we can utilize $\mathbf{z}^*$ to achieve a suboptimal solution for Eq. (107), denoted by $\hat{P}$. Indeed $\hat{P}$ is a subset from the set $\{1,2, \ldots, m\}$, which represents the suboptimal sensor selection. Following the approach developed by Joshi and Boyd [127], we define $\hat{P}$ corresponding to the u largest elements of $\mathbf{z}^*$. Let $\mathbf{z}^*$ be sorted in a descending order, then $\hat{P}$ is defined as

$$\hat{P} = \{i_1, i_2, \ldots, i_u\} \tag{110}$$

which is a set representing the indices corresponding to the u largest elements of $\mathbf{z}^*$. Let $\hat{\mathbf{z}} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$ be a binary vector correspond to $\hat{P}$, where

$$\hat{\mathbf{z}}_j = \begin{cases} 1, & j \in \hat{P} \\ 0, & otherwise. \end{cases},$$

then clearly $\hat{\mathbf{z}}$ is a feasible solution to Eq. (108), since it meets the constraints of Eq. (108). Thus, its associated objective function is an upper bound for the original optimal objective value of Eq. (107), i.e.

$$U_{\mu^*} = \mu_{avg}(\mathbf{\Phi}(\hat{\mathbf{z}})) \tag{111}$$

The upper bound U_{μ^*} and lower bound L_{μ^*} of the original optimal objective value can be used to find a gap, defined as

$$\boldsymbol{\delta} = U_{\mu^*} - L_{\mu^*} = \mu_{avg}\left(\mathbf{\Phi}(\hat{\mathbf{z}})\right) - \mu_{avg}\left(\mathbf{\Phi}(\mathbf{z}^*)\right) \tag{112}$$

The ideal case happens at $\boldsymbol{\delta} = 0$, which means that $\hat{\mathbf{z}}$ is optimal for Eq. (107). If this gap is small enough, we can make sure that the selected sensors resulted from solving Eq. (108) are almost equivalent to the results of Eq. (107). The case studies in Sec. 4.3 present the results of sensor selection methodology associated with small gap $\boldsymbol{\delta}$**.**

As discussed above, the sensor selection set is determined by $\hat{P}$ with $\left|\hat{P}\right| = u$, which is computed from $\mathbf{z}^*$ through Eq. (110). Then, the supervised classification can be carried out from the selected sensor data, specifically by solving

$$\mathbf{y}_{\hat{P}} = \mathbf{\Phi}_{\hat{P}}\mathbf{w} + \boldsymbol{\varepsilon} \tag{113}$$

where $\mathbf{y}_{\hat{P}}$ is a subvector of $\mathbf{y}$ associated to $\hat{P}$, and $\mathbf{\Phi}_{\hat{P}}$ is a submatrix of $\mathbf{\Phi}$ whose rows are associated to $\hat{P}$.

6.4 Case Study

We demonstrate the effectiveness of the proposed optimization method by application to online classification of MMH tasks. Sec. 6.4.1 describes experimental procedures using wearable sensors to obtain whole-body kinematics during a simulated job. In Sec. 6.4.2, we demonstrate the effectiveness of the proposed optimization method in reducing the average mutual coherence. Sec. 6.4.3 validates the performance of the improved OSEC algorithm as well as the proposed sensor selection methodology.

6.4.1 Simulated MMH Tasks

In an experiment reported previously [107], 10 gender-balanced young participants (19-29 years old) completed four cycles of a simulated job a laboratory environment. Each job cycle was designed to include major MMH tasks such as lifting/lowering, pushing/pulling, and carrying. Participants were allowed to complete each MMH task using self-selected comfortable styles and speeds. Each of the four job cycles was paced to last 7 minutes, rest was given after each cycle, and the four cycles were repeated three times.

For classification purposes, ground truth MMH task labels were manually assigned by direct observation (via video recordings). MMH task labels include: 1) lifting from the ground (LG), 2) lifting from knuckle height (LK), 3) lowering to the ground (LoG), 4) lowering to knuckle height (LoK), 5) pushing, 6) pulling, 7) carrying, and 8) walking (only as required to perform the tasks).

During the simulated job, whole-body kinematics were monitored at 60 Hz using an inertial motion capture system (MVN BIOMECH, Xsens technologies B.V., Enschede, the Netherlands), with 17 IMUs positioned on the head, sternum and pelvis; and bilaterally on the scapulae, the upper and lower arms, hands, thighs, shanks, and feet (Figure 6-1 for more details, please refer to [128]). Note that all participants indicated that the system was neither intrusive nor distracting. Whole-body kinematics were derived using 37 anatomical bony landmarks, each with three channels (leading to 111 sensor channels in total) required to define bony segments [129, 130]. Time series of the anatomical landmark locations were used as inputs to the proposed OSEC algorithm.

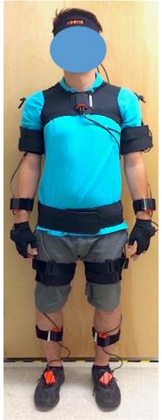

Figure 6-1The MVN™ system setup using 17 inertial measurement units.

6.4.2 Numerical Results for Optimizing Training Matrix $\boldsymbol{\Phi}$

The input matrix $\boldsymbol{\Phi}$ into our optimization problem (Eq. (99)) is the training matrix collected from MMH tasks explained in Sec. 6.4.1. Following the procedures (in Eq. (82)) explained in Chapter 5, matrix $\boldsymbol{\Phi}$ can be constructed. More specifically, we randomly chose 50 training samples from each label. This number was chosen for simplicity of illustration of the proposed optimization method. There were a total of 8 labels in the MMH experiment (see Sec. 6.4.1); thus, matrix $\boldsymbol{\Phi}$ has the dimension of 111 (the total number of sensor channels)×400. We solve the optimization problem in Eq. (99) to achieve the optimal solution $\bar{\mathbf{z}}$.

Figure 6-2 summarizes the absolute values of off-diagonal elements of the gram matrix $\mathbf{G}$ of both the original matrix $\boldsymbol{\Phi}$ and optimized matrix $\boldsymbol{\Phi}(\bar{\mathbf{z}})$. It can be seen that there is a shift towards the origin of the histogram after optimization. This indicates that the absolute values of off-diagonal elements of the gram matrix $\mathbf{G}$ have been reduced, hence the average mutual coherence is reduced accordingly.

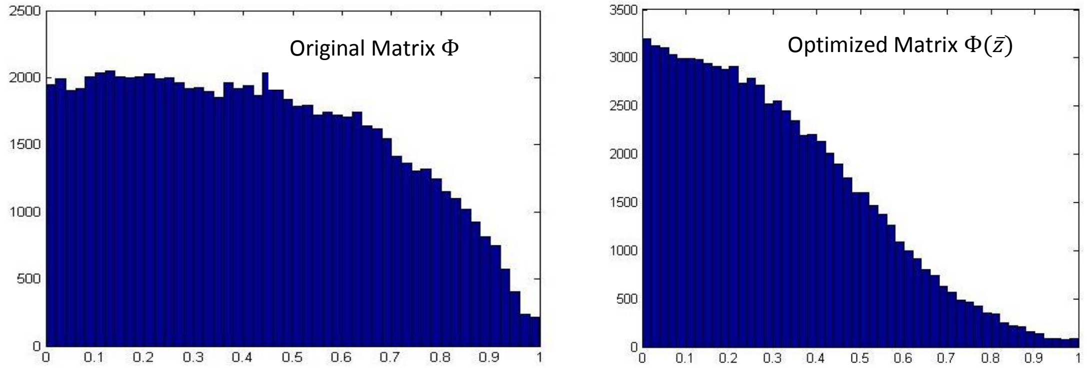

Figure 6-2. Histogram of the absolute off-diagonal elements of the Gram matrix of original matrix$\mathbf{\Phi}$, and optimized matrix $\mathbf{\Phi}(\bar{\mathbf{z}})$. The y axis shows the frequency and the x axis shows the values of mutual coherence. The mean and median of the left and right graphs are (0.4199, 0.4060) and (0.3019, 0.2694), respectively.

6.4.3 Online Classification and Sensor Selection Optimization for MMH Tasks

The classification performance of improved OSEC algorithm, and the effectiveness of the proposed sensor selection optimization for MMH tasks are presented in this section, and subsequently compared with the following widely used supervised classification methods as benchmarks: support vector machine (SVM), quadratic discriminant analysis (QDA), neural network (NN), naïve Bayes (NB), and k-nearest neighborhood (k-NN) ($k = 5$ for our analysis). All these algorithms were coded in Matlab (with the CPU of Intel dual core 2.5 GHz). From our cross validation analysis, the settings for these algorithms were as follows: SVM with Gaussian kernel, NN with one hidden layer containing 30 hidden units, QDA with an empirical prior distribution, and NN approach with a Gaussian distribution and empirical prior.

As explained in Sec. 6.4.1, there are in total 8 labels of sensor data. The first step of improved OSEC is to construct the training matrix $\mathbf{\Phi}$. Therefore, the size of the training data for each label n_i should be determined. Determining training data size in our proposed method is a trade-off between classification accuracy and computational efficiency. In this case study, a training data size of n_i=500 samples on each label (task) was randomly selected from the historical data to construct the training matrix $\mathbf{\Phi}$. In our current work, determining this size was done by numerical experiments. We tried a number of analyses to converge on the training data size of “500”. It was found that this number was sufficient to conduct the classification analysis with very good performance. But, further increasing the number of training data did not substantially improve the classification performance.

The 500 data samples on each label were randomly selected from the data set with a maximum of 100,000 samples on each label (the entire data set obtained from 10 participants doing a simulated job lasting for 7 minutes). Hence, the 500 samples do not correspond to a short duration (of 8.3 seconds). Rather, these points are randomly sampled over a time period of 7 minutes (out of 100,000 sensor data points), and a sample set that we consider to be a good representative for the entire data set. From the remaining historical data, we randomly selected the validation data (test data) to compare the performance of the proposed improved OSEC algorithm with the benchmarks. The validation data (test data) set $\mathbf{y}$ is selected with the size of 250 on each of the 8 labels.

Now, given the validation data set $\mathbf{y}$ and $\mathbf{\Phi}$ we would be able to run our improved OSEC algorithm. As the first step, solving the optimization problem in Eq. (99) leads to the optimal solution$(\bar{\mathbf{z}})$, based on which $\mathbf{\Phi}(\bar{\mathbf{z}})$ and $\mathbf{y}(\bar{\mathbf{z}})$ are computed. Now we would like to perform our classification analysis by applying the sparse estimation of Eq. (105). Following the improved OSEC procedures, we apply our proposed GBM algorithm for sparse estimation purpose. Since there are 250 validation data available on each label, the GBM algorithm is run $8 \times 250 = 2000$ times. Given the GBM estimation, we further perform our classification membership analysis to compute λ by Eq. (92).

For comparison purpose, we also conducted the classification analysis using the benchmark classification algorithms on the same training and validation data sets as explained above. The classification performance of the improved OSEC and other benchmark algorithms are measured based on a common criterion, namely,

$$F_{Score}{}^{i} = 2 \cdot \frac{(P^{i} \cdot S^{i})}{P^{i} + S^{i}} \quad , \forall i = 1, \dots, k \tag{114}$$

where P_i and S_i are precision and sensitivity of classification, respectively. $F_{Score}{}^{i}$ $(\forall i = 1, \dots, 8)$ for each of the classification algorithm was computed. We repeated all the procedures explained above to achieve 1000 realizations for classification analysis of our MMH tasks. The average F_{Score} over 1000 realizations are then reported.

We further study the effectiveness of the proposed sensor selection optimization formulated by Eq. (108). Indeed, we are interested in studying how effective the classification could be, if the number of sensor channels selected is reduced. Thus, we conducted the analysis for different number of sensor channels in range (30-100). For a given number of sensor channels, the optimization problem Eq. (108) is solved and consequently the optimal solution, namely, $\mathbf{\Phi}(\hat{\mathbf{z}})$ or $\mathbf{\Phi}_{\hat{P}}$, is achieved. Then, we performed the classification study by applying the OSEC procedure with Eq. (113).

Table 6-1 presents $F_{Score}{}^{i}$ $(\forall i = 1, \dots, 8)$ for MMH task classification with different numbers of sensor channels selected. Our results show the effectiveness of OSEC with optimized training matrix as well as OSEC with sensor selection optimization module. As observed, OSEC with optimized training matrix (OSEC_111) has the highest F_{Score} for all the labels (> 0.93). Among other classification algorithms, QDA and k-NN give relatively better classification performance than SVM, NB, and NN. However, compared to OSEC_111, their performance is substantially lower. Following sensor selection optimization with different numbers of selected sensor channels, it can be seen that overall classification performance decreases generally with a smaller number of sensor channels selected, yet this loss of classification performance appears minimal. Classification performance of OSEC is therefore considered superior over the benchmarks.

Table 6-1. Average F_{Score} of different classification algorithms for MMH tasks (OSEC_xxx stands for the improved OSEC method with different numbers of sensor channels selected). The final column indicates the average computational time of each classification by each of these algorithms

Algorithms	Carrying	Walking	LoK	LoG	LK	LG	Pulling	Pushing	Time (Sec.)
OSEC_111	1	1	0.971	1	0.990	0.939	0.980	0.960	0.0078
OSEC_100	1	0.990	0.923	1	0.938	0.949	1	0.961	0.0066
OSEC_90	1	0.980	0.941	0.990	0.931	0.949	0.990	0.960	0.0052
OSEC_80	1	0.980	0.923	0.990	0.940	0.948	0.990	0.960	0.0044
OSEC_70	0.990	0.960	0.923	1	0.949	0.939	0.990	0.970	0.0038
OSEC_60	1	0.959	0.942	0.990	0.939	0.920	0.970	0.939	0.0029
OSEC_50	1	0.960	0.904	0.980	0.905	0.922	0.990	0.939	0.0027
OSEC_40	1	0.959	0.923	0.971	0.875	0.960	0.970	0.898	0.0023
OSEC_30	0.980	0.939	0.840	0.990	0.860	0.909	0.960	0.940	0.0019
SVM	0.891	0.179	0.394	0.980	0.512	0.795	0.914	0.402	0.4087
QDA	0.929	0.766	0.661	0.912	0.436	0.716	0.639	0.725	0.0010
k-NN	0.949	0.816	0.561	0.962	0.486	0.766	0.699	0.695	0.0003
NB	0.765	0.571	0.349	0.697	0.244	0.290	0.400	0.189	0.0001
NN	0.943	0.762	0.361	0.918	0.447	0.648	0.588	0.522	0.0120

In Figure 6-3 we show the change of the global average F_{Score} (i.e., $\frac{\sum_{i=1}^{8} {F_{Score}}^{i}}{8}$) as a function of the number of selected sensors. This shows that reducing the number of optimal selected sensors will deteriorate the classification performance of OSEC (global average F_{Score}). The two limits of lowest and highest benchmark performances (i.e. NB, and k-NN, respectively) are provided in Figure 6-3 as well. From the figure, it can be seen that using only 30 sensor channels, which are optimally selected, yields better performance compared to other benchmark algorithms using 111 sensor channels. However, it can be observed that using less than 30 sensor channels also deteriorates the OSEC classification performance. This can be justified from the fact that the sparse estimation problem becomes much more ill-posed, so that a unique solution is not achievable.

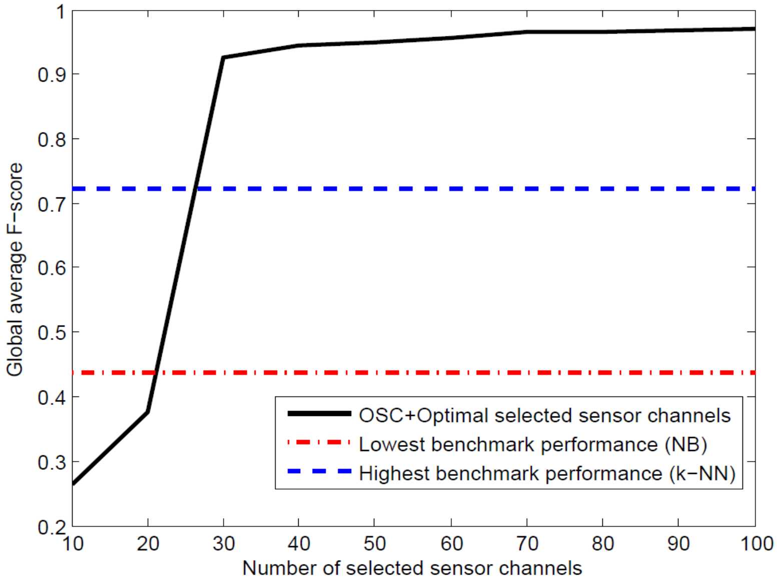

Figure 6-3. Global average F_{Score} results for different number of optimal selected sensors compared to the lowest and highest benchmark performance

The computational time required for each of these algorithms was reported in last column of Table 6-1. This computational time was determined as the mean time taken an algorithm to classify a single test data. In this, there were 1,000 trials, each with 2,000 test data, used to compare classification performance of the algorithms. Hence the computational time was the mean over 2×10^6 test data. From Table 6-1, it can be seen that the fastest algorithms were NB and kNN. The OSEC_111 algorithm was faster than some existing machine learning algorithms such as SVM and NN. Although OSEC-111 algorithm was not as fast as NB and kNN, its classification performance was substantially higher than these. Moreover, the OSEC_111 algorithm, with computational time of 0.0078 seconds, is fast enough for online classification of MMH tasks using wearable sensors at a sampling rate 60 Hz.

The effectiveness of the sensor selection optimization approach in reducing the computational time can be verified as well. The OSEC_30 with computational time of 0.0019 seconds, was the fastest algorithm among all other OSEC_xxx algorithms. Therefore, the developed sensor selection optimization not only reduces the number of sensors, but also achieve higher classification accuracy than other benchmark algorithms in the case study.

It is also useful to evaluate which sensors contribute more to classification of MMH tasks. For this, we have provided the optimal 30 selected sensor channels (due to the space limits, the optimal sensor channels for other cases are not shown). The corresponding optimal values $\mathbf{z}^*$ for these sensors (using Eq. (108)) are also reported in Table 6-2. As can be observed, the z channel (gravitational direction) contributes more than channels x and y to the classification of the MMH tasks. This can be justified through some physical interpretation of the sensors as follows. Due to the nature of MMH tasks, it is expected to see more variation in the z direction of the body kinematics while performing these tasks. Hence, it is expected that in monitoring the MMH tasks, wearable sensors should have more capability in detecting different patterns of data in their corresponding z direction rather than x or y directions. From Table 6-2, it can be observed that the sensor codes (32*z*-37*z*) contain the maximal value of weight, namely, one. Furthermore, it can be seen that the 30 optimal sensor channels out of 111, relate to 26 unique sensors out of the total of 37 sensors. This is certainly a very interesting result, as we could use only these unique sensors to monitor MMH tasks. As a result, the proposed sensor selection methodology could reduce costs induced by physical sensors, computational costs of handling high dimensional data, and avoiding worker discomfort by reducing the number of wearable sensors.

Table 6-2. Optimal selected sensors with their corresponding optimal optimization variable $\mathbf{z}^*$

Sensor Code	Body segment	Anatomical location	Optimal (z^*)	**Sensor Code**	Body segment	Anatomical location	Optimal (z^*)
32z	Left Foot	MT5	1.000	**17z**	Left Arm	Radius	0.491
33z	Left Foot	MT1	1.000	**24z**	Left Leg	Medial Knee	0.469
34z	Left Foot	Heel	1.000	**20z**	Right Arm	Ulnar	0.415
35z	Right Foot	MT5	1.000	**13z**	Pelvis	L5/S1	0.358
36z	Right Foot	MT1	1.000	**13x**	Pelvis	L5/S1	0.316
37z	Right Foot	Heel	1.000	**18z**	Right Arm	Lateral Elbow	0.294
1z	Head	Top	0.903	**15x**	Left Arm	Medial Elbow	0.261
26z	Left Leg	Medial Malleolus	0.897	**21y**	Right Arm	Radius	0.248
31z	Right Leg	Medial Malleolus	0.877	**16x**	Left Arm	Ulnar	0.231
25z	Left Leg	Lateral Malleolus	0.860	**20x**	Right Arm	Ulnar	0.231
30z	Right Leg	Lateral Malleolus	0.784	**19x**	Right Arm	Medial Elbow	0.226
23z	Left Leg	Lateral Knee	0.708	**14x**	Left Arm	Lateral Elbow	0.222
16z	Left Arm	Ulnar	0.658	**12y**	Pelvis	Left PSIS	0.215
29z	Right Leg	Lateral Knee	0.594	**9y**	Pelvis	Right ASIS	0.215
21z	Right Arm	Radius	0.491	**11y**	Pelvis	Right PSIS	0.210

* MT1 and MT5: 1st and 5th metatarsal bones

* L5/S1: Lumbosacral joint

* PSIS: Posterior superior iliac spine

* ASIS: Anterior superior iliac spine

6.5. Conclusions and Future Work

Combining wearable sensors with computationally efficient classification algorithms will facilitate real-time physical exposure assessments, by providing detailed contextual information for MMH tasks. However, wearable sensors yield high dimensional time series data regarding whole-body kinematics, which requires complex and computationally extensive analytics. To overcome the computational issues associated with analyzing these data, we utilized the OSEC algorithm developed in our previous research work. The OSEC algorithm is computationally efficient and benefits from accurate classification performance, and was thus considered potentially effective for online classification of MMH tasks.

Furthermore, we proposed an optimization approach aiming to improve MMH tasks classification performance/sparse estimation performance, by assigning different weights to the sensors. The criterion to measure classification performance is defined as an average of pairwise correlations (i.e., average mutual coherence) between the columns of the training matrix defined in the OSEC algorithm. The optimization problem was formulated as a generic approach to minimize average mutual coherence of any arbitrary matrix, by assigning weights to different rows/sensors. The proposed optimization problem was shown as a promising approach to improve the OSEC classification algorithm, by optimizing the training matrix. We further extended the proposed optimization approach by presenting physical interpretation coming from the importance level of various sensors. This leads to our proposed sensor selection methodology, which aims to select a desired number of sensors while improving classification performance. This is certainly useful since not only is the high dimensionality of the data reduced, but also the costs induced by physical sensors are decreased.

We evaluated the effectiveness of our proposed optimization problems using both numerical studies and motion data for MMH tasks. In the numerical studies, we demonstrated that the proposed optimization problem can be effectively utilized to improve the average mutual coherence of a matrix. Specifically, it was shown that the optimal solution (optimal row weights) significantly reduces the average mutual coherence of an original training matrix constructed from motion data. The obtained real data from MMH tasks was further utilized to evaluate the effectiveness of the proposed sensor selection methodology. From the results, it was observed with at least 30 optimal sensor channels, the MMH classification results of OSEC were always superior to several other common benchmarks approaches. Overall, the OSEC algorithm (improved OSEC algorithm) has the potential to rapidly provide information on work activities (e.g., distribution or frequency of task types during a work shift) at the workplace or changes in these after an ergonomic intervention is introduced. Such information can permit understanding work activity characteristics between workers or across workplaces, and evaluating the effectiveness of ergonomic intervention.

It should be noted, though, that the primary contribution of this paper is the development of a fast and accurate online classification (supervised) algorithm for monitoring of MMH tasks. In future work, to fully utilize this algorithm in practice, additional actions are need. One practical solution could be the use of the sliding window technique (e.g., with a sampling rate of 60 Hz, a sliding window size of 60 samples can be considered). The OSEC algorithm (improved OSEC algorithm) is fast enough to conduct 60 predictions (classifications) per second (this has been verified in Table 6-1). As a result, 60 predictions can be efficiently made within each window. But, there is no need to make 60 decisions. Instead, only one final decision on the task type is made, by accumulating all these 60 predictions. In other words, among all the 60 predictions, the task with the highest frequency is determined as the representative activity out of the sliding window.

While our results are promising, future work is also needed to evaluate classification performance of the OSEC, under more realistic conditions and for longer durations, since a laboratory study (vs. a real world application) likely provides higher classification performance [131]. There is also a need to support detailed physical exposure assessments (e.g., quantifying body kinetics). As such, in addition to the wearable sensors for body kinematics, future work should examine incorporation of a sensor system that can capture external body kinetics (e.g., in-shoe pressure measurement) to quantify task-specific or individual-specific body kinetics, and also to potentially improve MMH classification performance and/or to further reduce the number of wearable sensors required.

Chapter 7. Conclusions and Future Work

Sensor-based predictive analytics has been utilized as a promising approach for making quality improvement decisions for manufacturing and service systems. There has been always an interest of using a large number of sensors to provide comprehensive information from the processes (systems), which may improve the accuracy of the analytics. However, the excessive use of sensors typically results in high sensing costs, and high dimensionality of the sensor data whose analysis require intensive computational costs. The proposed compressive sensing based research methodology in this dissertation proposes approaches to reduce the number of sensors while ensuring the collected information from the processes (systems) is representative enough for predictive analytics.

The applications of the proposed research methodology have been studied in manufacturing and services systems, namely, multi-station assembly processes, advanced manufacturing (such as additive manufacturing), and manual material handling. Sensor placement optimization and fault diagnosis methodologies is developed based on compressive sensing for multi-station assembly processes. As opposed to the existing methodologies in the literature which require to use a large number of sensors, the proposed methodologies are capable of conducting fault diagnosis by a reduced number of sensors which are optimally located to improve the diagnosability of the multi-station assembly processes. The proposed methodologies are validated through real-case studies from auto body assembly processes which are known as typical examples for multi-station assembly processes.

A real-time monitoring methodology based on compressive sensing is further proposed in this dissertation. The proposed methodology is a novel supervised classification method equipped with a computationally fast sparse estimation algorithm to accommodate online quality assurance for advanced manufacturing processes. The performance of the proposed methodology is compared with the existing supervised classification algorithms for real-time monitoring applications in advanced manufacturing processes. The case studies demonstrate the superiority of the proposed methodology over the benchmarks.

Finally a sensor selection optimization methodology is developed based on compressive sensing. The proposed methodology optimally prioritizes the sensors, and selects the sensors with the highest priority, which provide the most relevant information for real-time monitoring applications. The effectiveness of the proposed methodology is studied for real-time body posture monitoring using wearable sensing technologies for manual material handling tasks. From the numerical studies, it is verified that by using the proposed sensor selection optimization methodology, the number of sensors can be effectively reduced, as well as the computational time required for real-time analysis of the data is significantly lowered.

The methodologies developed in this dissertation can be easily applied to big data analytics. Big data are characterized by three components, also known as 3 V’s, namely, (1) volume, (2) velocity, and (3) variety. These three components are presented in Figure 7-1. Volume is related to the high dimensionality of the data, velocity represents how fast the data is collected (i.e. real-time data), and variety relates to the different forms of data.

The proposed methodologies in this dissertation address the volume and velocity components of big data. For example, the sensor selection and sensor placement optimization methodologies tackle the challenge of high dimensionality of big data, and the proposed real-time monitoring methodology can be used to address the high velocity of big data. However, most of the data sets studied in this dissertation include only numerical data. By combining data with different forms (i.e., the variety component), including numerical and categorical, the accuracy of the analytics might be improved. As a part of future research work, CS can be explored and extended to develop methodologies for the sake of addressing all these three components together.

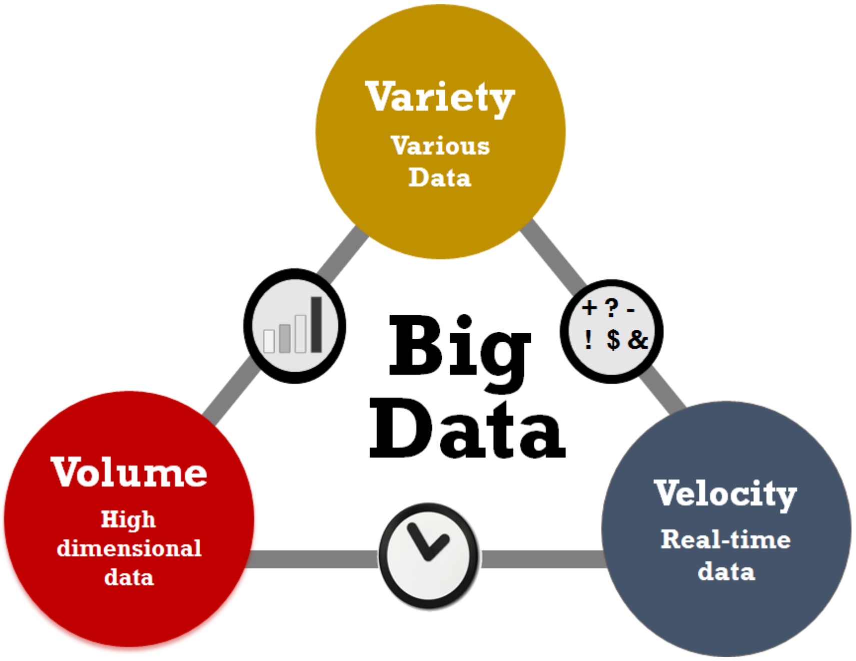

Figure 7-1. Big data analytics

Following the proposed future research in the area of big data analytics, a very promising application is introduced. The terminology given to this area of research is *smart sensor analytics*, which is a combination of three elements, (1) sensors, (2) analytics, and (3) actions. The diagram of smart sensor analytics is presented in Figure 7-2. From this figure, these three elements are connected to each other in the following way. Different types of sensors are utilized to collect structured and unstructured real-time data (i.e., big data). Dimensional reduction methodologies using CS approaches would be useful to extract features, and reduce the high dimensionality of the data. Given the extracted feature, the analytics (predictive analytics) are carried out. With real-time data, there is need to develop computational models enabling real-time analytics on the extracted features. Following the real-time analytics, online decisions can be made regarding the operations improvement or quality improvement. These decisions are considered as the inputs to the actions element of smart sensor analytics. The final task is to take actions and implement these decisions to make improvement in the operations or quality of systems. The paradigm of interconnected sensors, analytics, and actions, result in smart systems based on which training can be done through the system. Any change in the system can be captured by the analysis of real-time data, and improvement actions can be further implemented to sustain the system, and ensure continuous improvement.

In the future, the application of smart sensor analytics will be explored in the areas of advanced manufacturing, healthcare, and other service systems with the aim of reaching to start-up ideas.

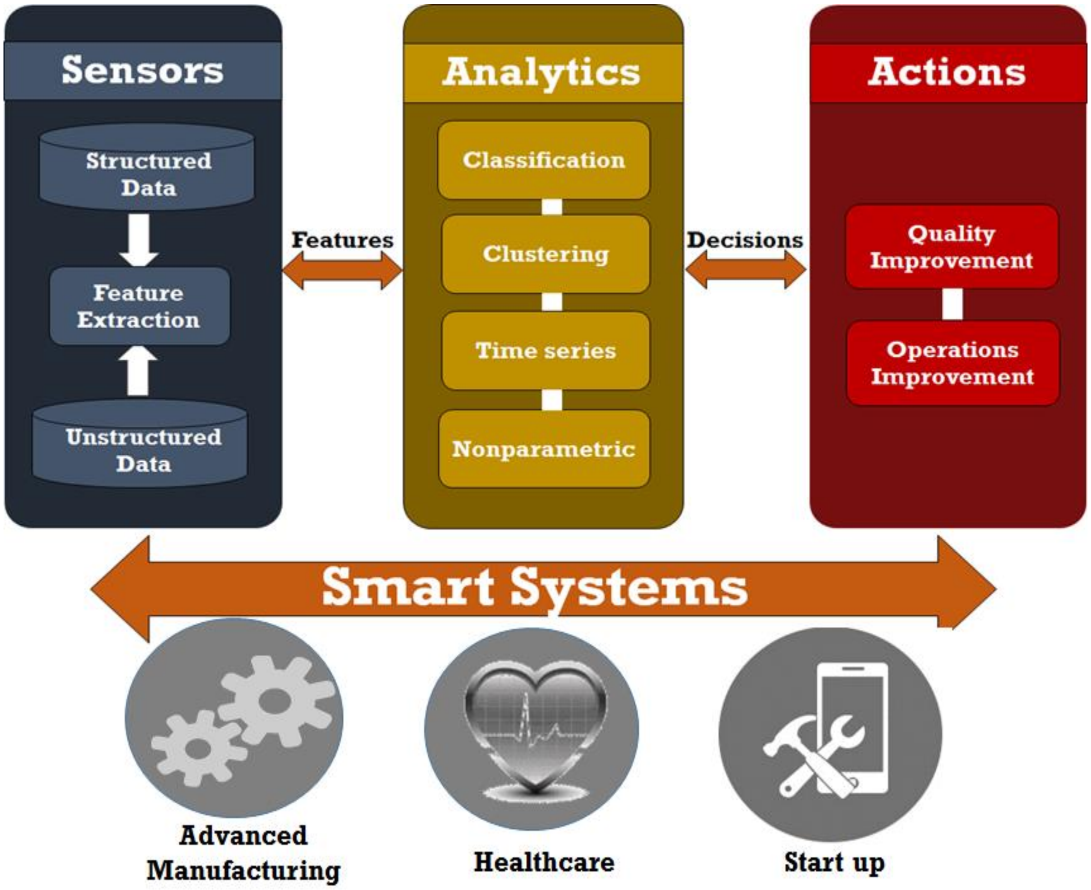

Figure 7-2. Smart Sensor Analytics Diagram

References

1. Apley, D. and J. Shi, *Diagnosis of multiple fixture faults in panel assembly.* Journal of Manufacturing Science and Engineering, 1998. **120**: p. 793.
2. Rao, P.K., et al., *Online Real-time Quality Monitoring in Additive Manufacturing Processes using Heterogeneous Sensors.* Journal of Manufacturing Science and Engineering, 2015.
3. Candè, E.J. and M.B. Wakin, *An introduction to compressive sampling.* Signal Processing Magazine, IEEE, 2008. **25**(2): p. 21-30.
4. Baraniuk, R., et al., *A simple proof of the restricted isometry property for random matrices.* Constructive Approximation, 2008. **28**(3): p. 253-263.
5. Ben-Haim, Z., Y.C. Eldar, and M. Elad, *Coherence-based performance guarantees for estimating a sparse vector under random noise.* Signal Processing, IEEE Transactions on, 2010. **58**(10): p. 5030-5043.
6. Elad, M., *Optimized projections for compressed sensing.* Signal Processing, IEEE Transactions on, 2007. **55**(12): p. 5695-5702.
7. Natarajan, B.K., *Sparse approximate solutions to linear systems.* SIAM journal on computing, 1995. **24**(2): p. 227-234.
8. Candes, E. and T. Tao, *The Dantzig selector: Statistical estimation when p is much larger than n.* The Annals of Statistics, 2007: p. 2313-2351.
9. Chen, S.S., D.L. Donoho, and M.A. Saunders, *Atomic decomposition by basis pursuit.* SIAM journal on scientific computing, 1998. **20**(1): p. 33-61.
10. Li, C., et al., *An efficient augmented lagrangian method with applications to total variation minimization.* Computational Optimization and Applications, 2013. **56**(3): p. 507-530.
11. Tibshirani, R., *Regression shrinkage and selection via the lasso.* Journal of the Royal Statistical Society. Series B (Methodological), 1996: p. 267-288.
12. Donoho, D.L., et al., *Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit.* Information Theory, IEEE Transactions on, 2012. **58**(2): p. 1094-1121.
13. Efron, B., et al., *Least angle regression.* The Annals of statistics, 2004. **32**(2): p. 407-499.
14. Pati, Y.C., R. Rezaiifar, and P. Krishnaprasad. *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*. in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. 1993. IEEE.
15. Tipping, M.E., *Sparse Bayesian learning and the relevance vector machine.* The journal of machine learning research, 2001. **1**: p. 211-244.
16. Tipping, M.E. and A.C. Faul. *Fast marginal likelihood maximisation for sparse Bayesian models*. in *Proceedings of the ninth international workshop on artificial intelligence and statistics*. 2003.
17. Wipf, D.P. and B.D. Rao, *Sparse Bayesian learning for basis selection.* Signal Processing, IEEE Transactions on, 2004. **52**(8): p. 2153-2164.
18. Candès, E.J. and Y. Plan, *Near-ideal model selection by ℓ1 minimization.* The Annals of Statistics, 2009. **37**(5A): p. 2145-2177.
19. Ceglarek, D. and J. Shi, *Dimensional variation reduction for automotive body assembly.* Manufacturing Review, 1995. **8**(2).
20. Ding, Y., J. Shi, and D. Ceglarek, *Diagnosability analysis of multi-station manufacturing processes.* Journal of dynamic systems, measurement, and control, 2002. **124**: p. 1.
21. Woodall, W.H. and D.C. Montgomery, *Research issues and ideas in statistical process control.* Journal of Quality Technology, 1999. **31**(4): p. 376-386.
22. Shi, J. and S. Zhou, *Quality control and improvement for multistage systems: A survey.* IIE Transactions, 2009. **41**(9): p. 744-753.

23. Zhou, S., Y. Chen, and J. Shi, *Statistical estimation and testing for variation root-cause identification of multistage manufacturing processes.* Automation Science and Engineering, IEEE Transactions on, 2005. **1**(1): p. 73-83.

24. Zhou, S., et al., *Diagnosability study of multistage manufacturing processes based on linear mixed-effects models.* Technometrics, 2003. **45**(4): p. 312-325.

25. Camelio, J. and S. Hu, *Multiple fault diagnosis for sheet metal fixtures using designated component analysis.* Journal of Manufacturing Science and Engineering, 2004. **126**: p. 91.

26. Ceglarek, D. and J. Shi, *Fixture failure diagnosis for autobody assembly using pattern recognition.* Journal of Engineering for Industry, 1996. **118**(1): p. 55-66.

27. Chang, M. and D. Gossard, *Computational method for diagnosis of variation-related assembly problems.* International Journal of Production Research, 1998. **36**(11): p. 2985-2995.

28. Ding, Y., D. Ceglarek, and J. Shi, *Fault diagnosis of multistage manufacturing processes by using state space approach.* Journal of Manufacturing Science and Engineering, 2002. **124**: p. 313.

29. Kong, Z., D. Ceglarek, and W. Huang, *Multiple Fault Diagnosis Method in Multistation Assembly Processes Using Orthogonal Diagonalization Analysis.* Journal of Manufacturing Science and Engineering, 2008. **130**: p. 011014.

30. Rong, Q., D. Ceglarek, and J. Shi, *Dimensional fault diagnosis for compliant beam structure assemblies.* Journal of Manufacturing Science and Engineering, 2000. **122**: p. 773.

31. Bruckstein, A., D. Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images.* SIAM review, 2009. **51**(1): p. 34–81.

32. Apley, D.W. and J. Shi, *A factor-analysis method for diagnosing variability in mulitvariate manufacturing processes.* Technometrics, 2001. **43**(1): p. 84-95.

33. Ding, Y., A. Gupta, and D.W. Apley, *Singularity issues in fixture fault diagnosis for multi-station assembly processes.* Journal of Manufacturing Science and Engineering, 2004. **126**: p. 200.

34. Liu, J., J. Shi, and S.J. Hu, *Engineering-driven factor analysis for variation source identification in multistage manufacturing processes.* Journal of Manufacturing Science and Engineering, 2008. **130**: p. 041009.

35. Donoho, D.L., *For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution.* Communications on pure and applied mathematics, 2006. **59**(6): p. 797-829.

36. Natarajan, B.K., *Sparse approximate solutions to linear systems.* SIAM journal on computing, 1995. **24**: p. 227.

37. Blumensath, T. and M. Davies, *Iterative thresholding for sparse approximations.* Journal of Fourier Analysis and Applications, 2008. **14**(5): p. 629-654.

38. Chen, S., D. Donoho, and M. Saunders, *Atomic decomposition by basis pursuit.* SIAM review, 2001. **43**(1): p. 129-159.

39. Donoho, D., et al., *Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit*. 2006: Citeseer.

40. Donoho, D. and M. Elad, *Maximal sparsity representation via l1 minimization.* Proc. Nat. Aca. Sci, 2003. **100**(5): p. 2197-2202.

41. Elad, M., *Why simple shrinkage is still relevant for redundant representations?* Information Theory, IEEE Transactions on, 2006. **52**(12): p. 5559-5569.

42. Gorodnitsky, I. and B. Rao, *Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm.* Signal Processing, IEEE Transactions on, 1997. **45**(3): p. 600-616.

43. Ji, S., Y. Xue, and L. Carin, *Bayesian compressive sensing.* Signal Processing, IEEE Transactions on, 2008. **56**(6): p. 2346-2356.

44. Mallat, S. and Z. Zhang, *Matching pursuits with time-frequency dictionaries.* IEEE Transactions on signal processing, 1993. **41**(12): p. 3397-3415.

45. Pati, Y., R. Rezaiifar, and P. Krishnaprasad. *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*. 2002. IEEE.

46. Rao, B. and K. Kreutz-Delgado, *An affine scaling methodology for best basis selection.* Signal Processing, IEEE Transactions on, 2002. **47**(1): p. 187-200.

47. Wipf, D. and B. Rao, *Sparse Bayesian learning for basis selection.* Signal Processing, IEEE Transactions on, 2004. **52**(8): p. 2153-2164.

48. Zayyani, H., M. Babaie-Zadeh, and C. Jutten. *Source estimation in noisy sparse component analysis*. 2007. IEEE.

49. Zayyani, H., M. Babaie-Zadeh, and C. Jutten, *An iterative Bayesian algorithm for sparse component analysis in presence of noise.* Signal Processing, IEEE Transactions on, 2009. **57**(11): p. 4378-4390.

50. Ding, Y., D. Ceglarek, and J. Shi. *Modeling and diagnosis of multistage manufacturing processes: part I state space model*. 2000. Citeseer.

51. Jin, J., Shi, J., *State space modeling of sheet metal assembly for dimensional control.* Ann Arbor. Ml, 1999. **48109**: p. 2117.

52. Huang, W., et al., *Stream-of-Variation (SOVA) Modeling—Part II: A Generic 3D Variation Model for Rigid Body Assembly in Multistation Assembly Processes.* Journal of Manufacturing Science and Engineering, 2007. **129**: p. 832.

53. Huang, W., et al., *Stream-of-Variation Modeling—Part I: A Generic Three-Dimensional Variation Model for Rigid-Body Assembly in Single Station Assembly Processes.* Journal of Manufacturing Science and Engineering, 2007. **129**: p. 821.

54. Liu, J., J. Jin, and J. Shi, *State space modeling for 3-D variation propagation in rigid-body multistage assembly processes.* Automation Science and Engineering, IEEE Transactions on, 2010. **7**(2): p. 274-290.

55. Higham, N.J., *Accuracy and stability of numerical algorithms*. 2002: Society for Industrial Mathematics.

56. Kong, Z., W. Huang, and A. Oztekin, *Variation Propagation Analysis for Multistation Assembly Process With Consideration of GD&T Factors.* Journal of Manufacturing Science and Engineering, 2009. **131**: p. 051010.

57. Liu, C.Q., Y. Ding, and Y. Chen, *Optimal coordinate sensor placements for estimating mean and variance components of variation sources.* IIE Transactions, 2005. **37**(9): p. 877-889.

58. Bastani, K., et al., *Fault Diagnosis Using an Enhanced Relevance Vector Machine (RVM) for Partially Diagnosable Multistation Assembly Processes.* 2012.

59. Ding, Y., D. Ceglarek, and J. Shi, *Fault diagnosis of multistage manufacturing processes by using state space approach.* Ann Arbor, 2002. **1001**: p. 48109.

60. Kong, Z. and D.H. Ceglarek, *W.(2008),'Multiple fault diagnosis method in multistation assembly processes using orthogonal diagonalization analysis'.* Journal of manufacturing science and engineering. **130**(1): p. 011014-10.

61. Zhou, S., Y. Chen, and J. Shi, *Statistical estimation and testing for variation root-cause identification of multistage manufacturing processes.* Automation Science and Engineering, IEEE Transactions on, 2004. **1**(1): p. 73-83.

62. Camelio, J.A., S.J. Hu, and H. Yim, *Sensor placement for effective diagnosis of multiple faults in fixturing of compliant parts.* Journal of manufacturing science and engineering, 2005. **127**(1): p. 68-74.

63. Ding, Y., et al., *Optimal sensor distribution for variation diagnosis in multistation assembly processes.* Robotics and Automation, IEEE Transactions on, 2003. **19**(4): p. 543-556.

64. Khan, A. and D. Ceglarek, *Sensor optimization for fault diagnosis in multi-fixture assembly systems with distributed sensing.* TRANSACTIONS-AMERICAN SOCIETY OF

MECHANICAL ENGINEERS JOURNAL OF MANUFACTURING SCIENCE AND ENGINEERING, 2000. **122**(1): p. 215-226.

65. Khan, A., D. Ceglarek, and J. Ni, *Sensor location optimization for fault diagnosis in multi-fixture assembly systems.* TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF MANUFACTURING SCIENCE AND ENGINEERING, 1998. **120**: p. 781-792.
66. Khan, A., et al., *Sensor optimization for fault diagnosis in single fixture systems: a methodology.* Journal of manufacturing science and engineering, 1999. **121**(1): p. 109-117.
67. Ren, Y. and Y. Ding, *Optimal sensor distribution in multi-station assembly processes for maximal variance detection capability.* IIE Transactions, 2009. **41**(9): p. 804-818.
68. Ding, Y., Shi, J., and Ceglarek, D., *Diagnosability analysis of multi-station manufacturing processes.* Journal of dynamic systems, measurement, and control,, 2002. **124**: p. 1.
69. Elad, M., *Sparse and redundant representations: from theory to applications in signal and image processing*. 2010: Springer Science & Business Media.
70. Candès, E.J., J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information.* Information Theory, IEEE Transactions on, 2006. **52**(2): p. 489-509.
71. Candès, E.J. and M.B. Wakin, *An introduction to compressive sampling.* Signal Processing Magazine, IEEE, 2008. **25**(2): p. 21-30.
72. Huang, W., et al., *Stream-of-Variation (SOVA) Modeling II: A Generic 3D Variation Model for Rigid Body Assembly in Multistation Assembly Processes.* Journal of Manufacturing Science and Engineering, 2007. **129**(4): p. 832-842.
73. Kong, Z., et al. *Stream of variation analysis for multiple station assembly process with consideration of various tolerance modes*. in *Transactions of the North American Manufacturing Research Conference*. 2006. Society of Manufacturing Engineers.
74. Kong, Z., W. Huang, and A. Oztekin, *Variation propagation analysis for multistation assembly process with consideration of GD&T factors.* Journal of manufacturing science and engineering, 2009. **131**(5).
75. Ding, Y., D. Ceglarek, and J. Shi. *Modeling and diagnosis of multistage manufacturing processes: part I–state space model*. in *Proceedings of the 2000 Japan/USA symposium on flexible automation*. 2000.
76. Candes, E.J., J.K. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements.* Communications on pure and applied mathematics, 2006. **59**(8): p. 1207-1223.
77. Mallat, S.G. and Z. Zhang, *Matching pursuits with time-frequency dictionaries.* Signal Processing, IEEE Transactions on, 1993. **41**(12): p. 3397-3415.
78. Donoho, D.L., et al., *Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit*. 2006: Department of Statistics, Stanford University.
79. Candes, E. and J. Romberg, *Sparsity and incoherence in compressive sampling.* Inverse problems, 2007. **23**(3): p. 969.
80. Donoho, D.L. and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ1 minimization.* Proceedings of the National Academy of Sciences, 2003. **100**(5): p. 2197-2202.
81. Gribonval, R. and M. Nielsen, *Sparse representations in unions of bases.* Information Theory, IEEE Transactions on, 2003. **49**(12): p. 3320-3325.
82. Huang, W., Z. Kong, and A. Chennamaraju, *Robust design for fixture layout in multistation assembly systems using sequential space filling methods.* Journal of computing and information science in engineering, 2010. **10**(4).

83. Exler, O., et al., *A Tabu search-based algorithm for mixed-integer nonlinear problems and its application to integrated process and control system design.* Computers & Chemical Engineering, 2008. **32**(8): p. 1877-1891.

84. Rao, P.K., et al., *A Graph Theoretic Approach for Quantification of Surface Morphology and its Application to Chemical Mechanical Planarization (CMP) Process.* IIE Transactions, 2015(just-accepted): p. 00-00.

85. Rao, P.K., et al., *Process-Machine Interaction (PMI) Modeling and Monitoring of Chemical Mechanical Planarization (CMP) Process Using Wireless Vibration Sensors.* Semiconductor Manufacturing, IEEE Transactions on, 2014. **27**(1): p. 1-15.

86. Hagan, M.T., H.B. Demuth, and M. Beale, *Neural network design, 1996.* PWS Pub, Boston, London.

87. Wright, J., et al., *Robust face recognition via sparse representation.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2009. **31**(2): p. 210-227.

88. Wipf, D. and S. Nagarajan, *Iterative reweighted and methods for finding sparse solutions.* Selected Topics in Signal Processing, IEEE Journal of, 2010. **4**(2): p. 317-329.

89. Gibson, I., D. Rosen, and B. Stucker, *Additive manufacturing technologies: rapid prototyping to direct digital manufacturing. 2010*. Springer.

90. Levy, G.N., R. Schindel, and J.-P. Kruth, *Rapid manufacturing and rapid tooling with layer manufacturing (LM) technologies, state of the art and future perspectives.* CIRP Annals-Manufacturing Technology, 2003. **52**(2): p. 589-609.

91. Kong, Z., et al., *Process performance prediction for chemical mechanical planarization (CMP) by integration of nonlinear Bayesian analysis and statistical modeling.* Semiconductor Manufacturing, IEEE Transactions on, 2010. **23**(2): p. 316-327.

92. Kong, Z., et al., *Nonlinear sequential Bayesian analysis-based decision making for end-point detection of chemical mechanical planarization (CMP) processes.* Semiconductor Manufacturing, IEEE Transactions on, 2011. **24**(4): p. 523-532.

93. Ray, S.J. and J. Teizer, *Real-time construction worker posture analysis for ergonomics training.* Advanced Engineering Informatics, 2012. **26**(2): p. 439-455.

94. Bukkapatnam, S., P. Rao, and R. Komanduri, *Experimental dynamics characterization and monitoring of MRR in oxide chemical mechanical planarization (CMP) process.* International Journal of Machine Tools and Manufacture, 2008. **48**(12): p. 1375-1386.

95. Cheung, Z., et al., *Ergonomic guidelines for manual material handling*, in *NIOSH Publication*. 2007, NIOSH.

96. Statistics, B.o.L., *Nonfatal Occupational Injuries and Illnesses Requiring Days Away From Wrok, 2013, Washington, D.C.: U.S. Department of Labor.* Nov. 2014.

97. Statistics, B.o.L., *Nonfatal Occupational Injuries and Illnesses Requiring Days Away From Wrok, 2012, Washington, D.C.: U.S. Department of Labor, .* Nov. 2013.

98. Lötters, F. and A. Burdof, *Are changes in mechanical exposure and musculoskeletal health good performance indicators for primary interventions?* International archives of occupational and environmental health, 2002. **75**(8): p. 549-561.

99. Westgaard, R.H. and J. Winkel, *Ergonomic intervention research for improved musculoskeletal health: a critical review.* International journal of industrial ergonomics, 1997. **20**(6): p. 463-500.

100. Davis, K.G. and S.E. Kotowski, *Understanding the ergonomic risk for musculoskeletal disorders in the United States agricultural sector.* American journal of industrial medicine, 2007. **50**(7): p. 501-511.

101. Vignais, N., et al., *Innovative system for real-time ergonomic feedback in industrial manufacturing.* Applied ergonomics, 2013. **44**(4): p. 566-574.

102. Ogris, G., et al. *Using a complex multi-modal on-body sensor system for activity spotting*. in *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*. 2008. IEEE.

103. Joshua, L. and K. Varghese, *Accelerometer-based activity recognition in construction.* Journal of Computing in Civil Engineering, 2010. **25**(5): p. 370-379.

104. Yonetsuji, T., et al. *A measurement and evaluation method of a support system to teach how to improve transferring patients*. in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. 2011. IEEE.

105. David, G., *Ergonomic methods for assessing exposure to risk factors for work-related musculoskeletal disorders.* Occupational Medicine, 2005. **55**(3): p. 190-199.

106. Li, G. and P. Buckle, *Current techniques for assessing physical exposure to work-related musculoskeletal risks, with emphasis on posture-based methods.* Ergonomics, 1999. **42**(5): p. 674-695.

107. Kim, S. and M.A. Nussbaum, *An evaluation of classification algorithms for manual material handling tasks based on data obtained using wearable technologies.* Ergonomics, 2014(ahead-of-print): p. 1-12.

108. Parkka, J., et al., *Activity classification using realistic data from wearable sensors.* Information Technology in Biomedicine, IEEE Transactions on, 2006. **10**(1): p. 119-128.

109. Bastani, K., Rao, P., Kong, Z.,, *An Online Sparse Estimation based Classification Approach for Real-time Monitoring in Advanced Manufacturing Processes from Heterogeneous Sensor Data.* IIE Transactions (Submitted), 2014.

110. Doukas, C. and I. Maglogiannis. *Advanced patient or elder fall detection based on movement and sound data*. in *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*. 2008. IEEE.

111. Foerster, F. and J. Fahrenberg, *Motion pattern and posture: correctly assessed by calibrated accelerometers.* Behavior research methods, instruments, & computers, 2000. **32**(3): p. 450-457.

112. Huynh, T. and B. Schiele. *Analyzing features for activity recognition*. in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. 2005. ACM.

113. Huỳnh, T. and B. Schiele, *Unsupervised discovery of structure in activity data using multiple eigenspaces*, in *Location-and Context-Awareness*. 2006, Springer. p. 151-167.

114. Kern, N., B. Schiele, and A. Schmidt, *Multi-sensor activity context detection for wearable computing*, in *Ambient Intelligence*. 2003, Springer. p. 220-232.

115. Mantyjarvi, J., J. Himberg, and T. Seppanen. *Recognizing human motion with multiple acceleration sensors*. in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*. 2001. IEEE.

116. Maurer, U., et al., *Location and activity recognition using eWatch: A wearable sensor platform*, in *Ambient Intelligence in Everyday Life*. 2006, Springer. p. 86-102.

117. Najafi, B., et al., *Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly.* Biomedical Engineering, IEEE Transactions on, 2003. **50**(6): p. 711-723.

118. Pirttikangas, S., K. Fujinami, and T. Nakajima, *Feature selection and activity recognition from wearable sensors*, in *Ubiquitous Computing Systems*. 2006, Springer. p. 516-527.

119. Preece, S.J., et al., *Activity identification using body-mounted sensors—a review of classification techniques.* Physiological measurement, 2009. **30**(4): p. R1.

120. Ravi, N., et al. *Activity recognition from accelerometer data*. in *AAAI*. 2005.

121. Bao, L. and S.S. Intille, *Activity recognition from user-annotated acceleration data*, in *Pervasive computing*. 2004, Springer. p. 1-17.

122. Baek, J., et al. *Accelerometer signal processing for user activity detection*. in *Knowledge-Based Intelligent Information and Engineering Systems*. 2004. Springer.

123. Candes, E.J., *The restricted isometry property and its implications for compressed sensing.* Comptes Rendus Mathematique, 2008. **346**(9): p. 589-592.

124. Bickel, P.J., Y.a. Ritov, and A.B. Tsybakov, *Simultaneous analysis of Lasso and Dantzig selector.* The Annals of Statistics, 2009. **37**(4): p. 1705-1732.
125. Jong, Y., *Practical Global Optimization Algorithm for the Sum-of-Ratios Problem.* arXiv preprint arXiv:1207.1153, 2012.
126. Das, A. and D. Kempe. *Sensor selection for minimizing worst-case prediction error*. in *Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on*. 2008. IEEE.
127. Joshi, S. and S. Boyd, *Sensor selection via convex optimization.* Signal Processing, IEEE Transactions on, 2009. **57**(2): p. 451-462.
128. www.xsens.com/en/mvn-biomech.
129. Wu, G., et al., *Standardization and Terminology Committee of the International Society of Biomechanics. ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part I: ankle, hip, and spine.* J Biomech, 2002. **35**(4): p. 543-8.
130. Wu, G., et al., *International Society of Biomechanics. ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—part II: shoulder, elbow, wrist and hand.* J Biomech, 2005. **38**(5): p. 981-992.
131. van Hees, V.T., et al., *Impact of study design on development and evaluation of an activity type classifier.* Journal of Applied Physiology, 2013.
132. Steeb, W.H. and Y. Hardy, *Matrix Calculus and Kronecker Product: A Practical Approach to Linear and Multilinear Algebra*. 2011: World Scientific Publishing Company.
133. Schott, J.R., *Matrix analysis for statistics.* 2005.
134. Bellman, R., et al., *Introduction to matrix analysis*. Vol. 960. 1970: SIAM.
135. Golub, G.H. and C.F. Van Loan, *Matrix computations*. Vol. 3. 2012: JHU Press.
136. Dasgupta, S. and A. Gupta, *An elementary proof of a theorem of Johnson and Lindenstrauss.* Random structures and algorithms, 2003. **22**(1): p. 60-65.

Appendix

Appendix A

The derivation of $L(\boldsymbol{\alpha}, \Sigma_{\sigma^2})$ (Eq. (**31**)) with respect to $\boldsymbol{\alpha}$ and Σ_{σ^2} is provided here. Firstly, the derivation of $L(\boldsymbol{\alpha}, \Sigma_{\sigma^2})$ is presented. We note that $\frac{dln|X|}{d\boldsymbol{\alpha}} = Tr(X^{-1}\frac{dX}{d\boldsymbol{\alpha}})$. The derivative of a matrix with respect to a vector can be derived in a way that the whole elements of the matrix are derived according to all the elements of the vector, i.e., $\frac{dX}{d\alpha_1}, \frac{dX}{d\alpha_2}, \dots, \frac{dX}{d\alpha_N}$ [132]. The estimate of each element of α is:

$$\boldsymbol{\alpha}_{new\ i} = \frac{1-\boldsymbol{\alpha}_i\Sigma_{ii}}{\boldsymbol{\mu}_i{}^2} \tag{A.1}$$

where $\sum_{ii}$is the i-th diagonal element of the covariance matrix $\boldsymbol{\Sigma}$ (Eq. (25)) of the posterior weight **w**.

Similarly, setting the derivative of $L(\boldsymbol{\alpha}, \sum_{\sigma^2})$ with respect to $\boldsymbol{\beta}$ to zero provides the estimate for each element of $\boldsymbol{B}$. In Eq. (**31**), there are only three terms containing $\boldsymbol{\beta}$, and the rest are just constants. Thus, we have

$$\frac{dL(\boldsymbol{\alpha}, \Sigma_{\sigma^2})}{d\boldsymbol{\beta}} = -\frac{1}{2}\frac{dln|\boldsymbol{\Sigma}^{-1}|}{d\boldsymbol{\beta}} + \frac{1}{2}\frac{dln\left|\Sigma_{\sigma^2}^{-1}\right|}{d\boldsymbol{\beta}} - \frac{1}{2}\frac{d(\mathbf{t} - \boldsymbol{\Phi\mu})^T\Sigma_{\sigma^2}^{-1}(\mathbf{t} - \boldsymbol{\Phi\mu})}{d\boldsymbol{\beta}} \tag{A.2}$$

The derivative of the first term in Eq. (A.2) with respect to the j-th element of $\boldsymbol{\beta}$ is

$$\frac{dln|\Sigma^{-1}|}{d\boldsymbol{\beta}j} = \frac{-1}{\sigma_j{}^2}Tr\left(\boldsymbol{\Sigma\Phi}^T J_j \boldsymbol{\Phi}\right) \tag{A.3}$$

where β_j denotes the j-th element of β, J_j is an $m \times m$ matrix where all its elements are zero except the j-th diagonal element which is one. Please note that we took advantage of the fact that for a general matrix, we have $\frac{dln|X|}{d\boldsymbol{\alpha}} = Tr(X^{-1}\frac{dX}{d\boldsymbol{\alpha}})$. For the second term $ln\left|\Sigma_{\sigma^2}^{-1}\right|$, the derivation with respect to j-th element of B is illustrated as Eq. (A.4)

$$\frac{dln|\Sigma_{\sigma^2}{}^{-1}|}{d\boldsymbol{\beta}j} = \frac{-1}{\sigma_j{}^2} \tag{A.4}$$

Finally, Eq. (A.5) presents the derivation of the third term with respect to B_j, which is

$$\frac{d(\mathbf{t}-\boldsymbol{\Phi\mu})^T {\Sigma_{\sigma^2}}^{-1}(\mathbf{t}-\boldsymbol{\Phi\mu})}{d\boldsymbol{\beta}j} = \frac{-(\mathbf{t}_j-\boldsymbol{\Phi}_\mathbf{j}\boldsymbol{\mu})^2}{({\sigma_j}^2)^2} \tag{A.5}$$

where $\mathbf{t}_j$ is the j-th element of column vector $\mathbf{t}$, and $\boldsymbol{\Phi}_j$ is the j-th row of $\boldsymbol{\Phi}$ matrix. After illustration of all these derivations, Eq. (31 can be written as

$$\frac{dL(\boldsymbol{\alpha},\Sigma_{\sigma^2})}{d\boldsymbol{\beta}} = \frac{1}{2{\sigma_j}^2} Tr\left(\Sigma\boldsymbol{\Phi}^T J_j\ \boldsymbol{\Phi}\right) - \frac{1}{2{\sigma_j}^2} + \frac{(\mathbf{t}_j-\boldsymbol{\Phi}_j\boldsymbol{\mu})^2}{2({\sigma_j}^2)^2} \tag{A.6}$$

The last step is to set Eq. (A.6) to zero to achieve the estimate of the noise variance. The estimate of the noise variance is presented as follows:

$${\sigma_j}^2 = \frac{(\mathbf{t}_j-\Phi_j\mu)^2}{1-Tr\left(\Sigma\Phi^T J_j\ \Phi\right)} \tag{A.7}$$

Appendix B
Proof of Proposition 1

We start the derivation process with expanding the left hand side of Eq. (6.a) as follows

$$cov(\mathbf{y}) = \begin{bmatrix} cov(\mathbf{y}_1) & cov(\mathbf{y}_1,\mathbf{y}_2) & \dots & cov(\boldsymbol{y}_1,\mathbf{y}_M) \\ cov(\mathbf{y}_2,\mathbf{y}_1) & cov(\mathbf{y}_2) & \dots & cov(\mathbf{y}_2,\mathbf{y}_M) \\ \vdots & \vdots & \ddots & \vdots \\ cov(\mathbf{y}_M,\mathbf{y}_1) & cov(\mathbf{y}_M,\mathbf{y}_2) & \dots & cov(\mathbf{y}_M) \end{bmatrix}$$

where cov($\mathbf{y}_\mathrm{i}$) denotes the covariance matrix of the measurements at station *i*, and cov($\mathbf{y}_\mathrm{j},\mathbf{y}_\mathrm{k}$) includes all the covariance elements between station *j* and *k*, i.e.

$$cov(\mathbf{y}_j,\mathbf{y}_k) = \begin{bmatrix} cov(\mathbf{y}_{1j},\mathbf{y}_{1k}) & cov(\mathbf{y}_{1j},\mathbf{y}_{2k}) & \dots & cov(\mathbf{y}_{1j},\mathbf{y}_{m_k k}) \\ cov(\mathbf{y}_{2j},\mathbf{y}_{1k}) & cov(\mathbf{y}_{2j},\mathbf{y}_{2k}) & \dots & cov(\mathbf{y}_{2j},\mathbf{y}_{m_k k}) \\ \vdots & \vdots & \ddots & \vdots \\ cov(\mathbf{y}_{m_j j},\mathbf{y}_{1k}) & cov(\mathbf{y}_{m_j j},\mathbf{y}_{2k}) & \dots & cov(\mathbf{y}_{m_j j},\mathbf{y}_{m_k k}) \end{bmatrix}$$

where $\mathbf{y}_{ik}$ is the *i*-th measurement at station *k*, m_j and m_k are the number of measurements at station *j* and *k*, respectively.

Here we do the vectorization on each element (i.e. $cov(\mathbf{y}_j, \mathbf{y}_k)$, $\forall j \le k = 1,..,M$) and then we stack all of them together to have the final vectorized form of the variance model. From Eqs. (5) and (6), the covariance elements between stations *j* and *k* are presented as,

$$\underset{j \le k}{cov}(\mathbf{y}_j, \mathbf{y}_k) = C_j[\sum_{i=1}^{j} \boldsymbol{\phi}_{\mathrm{j,i}}\mathbf{B}_i\; cov(\,\mathbf{u}_i)(\boldsymbol{\phi}_{\mathrm{j,i}}\mathbf{B}_{\mathbf{i}})^T]\mathbf{C}_k{}^T + cov(\boldsymbol{\varepsilon}_j, \boldsymbol{\varepsilon}_k)$$

(A.8.a)

Let $\mathbf{F}_{\mathrm{ji}} = \boldsymbol{\phi}_{\mathrm{j,i}}\mathbf{B}_i$, and $\mathbf{F}_{\mathrm{ki}} = \boldsymbol{\phi}_{\mathrm{k,i}}\mathbf{B}_i$ (A.8.b)

This simplifies Eq. (A.8.a) into

$$\underset{j \le k}{cov}(\mathbf{y}_j, \mathbf{y}_k) = \mathbf{C}_j[\sum_{i=1}^{j} \mathbf{F}_{ji} cov(\,\mathbf{u}_i)\mathbf{F}_{ki}{}^T]\mathbf{C}_k{}^T + cov(\boldsymbol{\varepsilon}_j, \boldsymbol{\varepsilon}_k) \qquad \text{(A.9)}$$

We expand the right hand side of Eq. (A.9),

$$cov(\mathbf{y}_j, \mathbf{y}_k) = \mathbf{C}_j\mathbf{F}_{j1} cov(\mathbf{u}_1)\mathbf{F}_{k1}{}^T\mathbf{C}_k{}^T + \mathbf{C}_j\mathbf{F}_{j2} cov(\mathbf{u}_2)\mathbf{F}_{k2}{}^T\mathbf{C}_k{}^T + \cdots$$
$$+\mathbf{C}_j\mathbf{F}_{jj} cov(\mathbf{u}_j)\mathbf{F}_{kj}{}^T\mathbf{C}_k{}^T + cov(\boldsymbol{\varepsilon}_j, \boldsymbol{\varepsilon}_k)$$

(A.10)

By Eq. (A.8.b) and some matrix product properties on Eq. (A.8.b), we have

$$cov(\mathbf{y}_j, \mathbf{y}_k) = \mathbf{C}_j[\sum_{i=1}^{n1} \sigma_{i1}{}^2\mathbf{f}_{:i}{}^{j1}\,\mathbf{f}_{i:}{}^{k1}]\mathbf{C}_k{}^T + \mathbf{C}_j[\sum_{i=1}^{n2} \sigma_{i2}{}^2\mathbf{f}_{:i}{}^{j2}\,\mathbf{f}_{i:}{}^{k2}]\mathbf{C}_k{}^T + \cdots$$
$$+\mathbf{C}_j[\sum_{i=1}^{nj} \sigma_{ij}{}^2\mathbf{f}_{:i}{}^{jj}\,\mathbf{f}_{i:}{}^{kj}]\mathbf{C}_k{}^T + cov(\boldsymbol{\varepsilon}_j, \boldsymbol{\varepsilon}_k) \qquad \text{(A.11)}$$

where $\mathbf{f}_{:i}{}^{jr}$ is the *i*-th column in matrix $\mathbf{F}_{jr}$, and $\mathbf{f}_{:i}{}^{kr}$ is the *i*-th row in matrix $\mathbf{F}_{kr}$ ($\forall r = 1,2, \dots j$). For simplification, Let

$$\mathbf{G}_{irjk} = \mathbf{f}_{:i}{}^{jr}\mathbf{f}_{i:}{}^{kr} \; (\forall r = 1,2, \dots j, \forall i = 1,2, \dots n_r\;) \qquad \text{(A.12.a)}$$

then Eq. (11) can be expressed as,

$$cov(\mathbf{y}_j, \mathbf{y}_k) = \mathbf{C}_j[\sum_{r=1}^{j}\sum_{i=1}^{m_r} \sigma_{ir}{}^2\,\mathbf{G}_{irjk}]\mathbf{C}_k{}^T + cov(\boldsymbol{\varepsilon}_j, \boldsymbol{\varepsilon}_k) \qquad \text{(A.12.b)}$$

We use vectorization operator $Vec(\cdot)$ [133] on Eq. (A.12.b) and utilizing the properties of the vectorization operator on the product of matrices. Equation (A.12.b) can be written as,

$$vec\left(cov\left(\mathbf{y}_j,\mathbf{y}_k\right)\right)=\left(\mathbf{C}_k\otimes\mathbf{C}_j\right)\left[\sum_{r=1}^{j}\sum_{i=1}^{m_r}\sigma_{ir}{}^2\, vec\left(\mathbf{G}_{irjk}\right)\right]+vec\left(cov(\boldsymbol{\varepsilon}_j,\boldsymbol{\varepsilon}_k)\right) \tag{A.13}$$

where $\otimes$ is the Kronecker matrix product [133]. To recognize the vectorized form of Eq. (6.a), Let:

$$\mathbf{t}_{jk}= vec\left(cov\left(\mathbf{y}_j,\mathbf{y}_k\right)\right) \tag{A.14.a}$$

$$\boldsymbol{\Lambda}_{jk}=\left(\mathbf{C}_k\otimes\mathbf{C}_j\right) \tag{A.14.b}$$

$$\boldsymbol{\Omega}_{jk}=\left[vec\left(\mathbf{G}_{11jk}\right),\dots,vec(\mathbf{G}_{m_1 1jk}),vec(\mathbf{G}_{12jk}),\dots vec(\mathbf{G}_{m_j jjk})\right] \tag{A.14.c}$$

$$\mathbf{b}_{jk}=\left[\sigma_{11}{}^2,\sigma_{21}{}^2,\dots,\sigma_{m1,1}{}^2,\sigma_{12}{}^2,\dots,\sigma_{mj,j}{}^2\ \right]^T \tag{A.14.d}$$

$$\boldsymbol{\xi}_{jk}=vec(cov(\boldsymbol{\varepsilon}_j,\boldsymbol{\varepsilon}_k)) \tag{A.14.e}$$

then Eq. (A.13) can be expressed as

$$\underset{j\le k}{\mathbf{t}_{jk}}=\boldsymbol{\Lambda}_{jk}\boldsymbol{\Omega}_{jk}\mathbf{b}_{jk}+\boldsymbol{\xi}_{jk}\quad\forall j=1,2,\dots M\ ,\forall k=1,2,\dots M \tag{A.15}$$

with vectorization of $cov\left(\mathbf{y}_j,\mathbf{y}_k\right)$, we can have the vectorization of $cov(\mathbf{y})$ by stacking all the results from Eq. (A.15) as follows,

$$\begin{bmatrix}\mathbf{t}_{11}\\\mathbf{t}_{12}\\\vdots\\\mathbf{t}_{1M}\\\mathbf{t}_{22}\\\vdots\\\mathbf{t}_{MM}\end{bmatrix}=\begin{bmatrix}\boldsymbol{\Lambda}_{11}\\\boldsymbol{\Lambda}_{12}\\\vdots\\\boldsymbol{\Lambda}_{1M}\\\boldsymbol{\Lambda}_{22}\\\vdots\\\boldsymbol{\Lambda}_{MM}\end{bmatrix}\left[\boldsymbol{\Omega}_{11}\quad\boldsymbol{\Omega}_{12}\quad\dots\quad\boldsymbol{\Omega}_{1M}\quad\boldsymbol{\Omega}_{22}\quad\dots\quad\boldsymbol{\Omega}_{MM}\right]\begin{bmatrix}\mathbf{b}_{11}\\\mathbf{b}_{12}\\\vdots\\\mathbf{b}_{1M}\\\mathbf{b}_{22}\\\vdots\\\mathbf{b}_{MM}\end{bmatrix}+\begin{bmatrix}\boldsymbol{\xi}_{11}\\\boldsymbol{\xi}_{12}\\\vdots\\\boldsymbol{\xi}_{1M}\\\boldsymbol{\xi}_{22}\\\vdots\\\boldsymbol{\xi}_{MM}\end{bmatrix} \tag{A.16}$$

Equation (A.16) can be simplified into

$$\mathbf{t}=\boldsymbol{\Lambda}\boldsymbol{\Omega}\mathbf{b}+\boldsymbol{\xi} \tag{A.17}$$

where $\mathbf{t}$=$[\mathbf{t}_{11}{}^{T}\ \mathbf{t}_{12}{}^{T} \cdots \mathbf{t}_{1M}{}^{T}\ \mathbf{t}_{22}{}^{T} \cdots \mathbf{t}_{MM}{}^{T}]^{\mathrm{T}}$, $\mathbf{\Lambda}$=$[\mathbf{\Lambda}_{11}{}^{T}\mathbf{\Lambda}_{12}{}^{T}\ \cdots\ \mathbf{\Lambda}_{1M}{}^{T}\ \ \mathbf{\Lambda}_{22}{}^{T}\ \ \cdot\cdot\ \mathbf{\Lambda}_{MM}{}^{T}]^{\mathrm{T}}$, $\mathbf{\Omega}$=$[\mathbf{\Omega}_{11}\ \mathbf{\Omega}_{12} \cdots \mathbf{\Omega}_{1M}\ \mathbf{\Omega}_{22} \cdots \mathbf{\Omega}_{MM}]$,$\mathbf{b}$=$[\mathbf{b}_{11}{}^{T}\ \mathbf{b}_{12}{}^{T} \cdots \mathbf{b}_{1M}{}^{T}\ \mathbf{b}_{22}{}^{T} \cdots \mathbf{b}_{MM}{}^{T}]^{\mathrm{T}}$, and $\boldsymbol{\xi}$=$[\boldsymbol{\xi}_{11}{}^{T}\ \boldsymbol{\xi}_{12}{}^{T} \cdots \boldsymbol{\xi}_{1M}{}^{T}\ \boldsymbol{\xi}_{22}{}^{T} \cdot\cdot\ \boldsymbol{\xi}_{MM}{}^{T}]^{\mathrm{T}}$. From Eq. (A.14.a), it is clear that vector $\mathbf{t}$ represents the sensor measurements. Matrix $\mathbf{\Lambda}$ is termed as the measurement matrix since it denotes the sensor placement information which is only determined by the observation matrix $\mathbf{C}_i$ in Eq. (A.14.b). Matrix $\mathbf{\Omega}$ is termed as the product/process matrix since it represents the product and process information through multiple stations, which is only determined by matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ considering Eqs. (A.14.c), and (A.14.a). Variable $\mathbf{b}$ denotes the variance of the process errors as shown in Eq. (A.14.d), and $\boldsymbol{\xi}$ represents the variance and covariance elements of the noise terms as in Eq. (A.14.e). **Q.E.D**.

Appendix C

OMP Algorithm

(1) At the first iteration ($t = 1$), the measurement residual vector $\mathbf{r}$ is set as the measurement vector $\mathbf{y}$, and the support set is empty ($S = \{\ \}$).

(2) The index ($i^* = \arg\max\{|\mathbf{r}^T\mathbf{g}_i|\}$) of the maximum correlated column in matrix $\mathbf{\Phi}$ with vector $\mathbf{r}$ is the first candidate to the support set $S = S \cup \{j\}$.

(3) LS is utilized to estimate the nonzero values represented by $\mathbf{w}_{\mathrm{OMP}}^{t} = (\mathbf{\Phi}_S^T\mathbf{\Phi}_S)^{-1}\mathbf{\Phi}_S^T\mathbf{y}$ at iteration t, where $\mathbf{\Phi}_S$ is a submatrix of matrix $\mathbf{\Phi}$, whose columns are associated with S, and $\mathbf{w}_{\mathrm{OMP}}^{t}$ represents the OMP estimate at iteration t. The measurement residual vector is updated based on $\mathbf{r} = \mathbf{y} - \mathbf{\Phi}_S\mathbf{w}_{\mathrm{OMP}}^{t}$.

(4) The above procedure is repeated for ϑ (sparsity level of the sparse vector) times. It is assumed that the sparsity level ϑ of the sparse vector $\mathbf{w}$ is known. In case of unknown ϑ; $\|\mathbf{r}\|_2^2 < \omega$ is

used as the stopping criterion, where ω is a user-defined threshold ($\omega = 10^{-6}$) that terminates the algorithm if the norm of measurement residual vector **r** is small enough.

Appendix D

RVM Algorithm

RVM assumes a Gaussian likelihood function $p(\mathbf{y}|\mathbf{w}, \sigma^2) = N(\mathbf{\Phi w}, \sigma^2\mathbf{I})$, and a conjugate prior on **w,** namely $p(\mathbf{w}|\mathbf{B}) = N(0, \mathbf{B}^{-1})$, where $\mathbf{B} = Diag(\boldsymbol{\alpha})$ with hyperparameters $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_\mathrm{n}]$ (inverse of the variance). By applying the Bayesian rule, the posterior distribution on the sparse vector **w** can be analytically computed due to the conjugacy of the prior. The posterior distribution is estimated $p(\mathbf{w}|\mathbf{y}, \mathbf{B}, \sigma^2) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu} = \sigma^{-2}\mathbf{\Sigma}\mathbf{\Phi}^T\mathbf{y},$ and $\boldsymbol{\Sigma} = (\sigma^{-2}\mathbf{\Phi}^T\mathbf{\Phi} + \mathbf{B})^{-1}$. The remaining task is to estimate the hyperparameters $\boldsymbol{\alpha}$ and noise variance σ^2 from data by maximizing the marginal likelihood. The estimates are given as $\boldsymbol{\alpha}_\mathrm{i}^{new} = \frac{1-\boldsymbol{\alpha}_\mathrm{i}\mathrm{N}_\mathrm{ii}}{\boldsymbol{\mu}_\mathrm{i}^2}$, $\forall i = 1, \dots, n$ and $\sigma^2 = \frac{\|\mathbf{y}-\mathbf{\Phi}\boldsymbol{\mu}\|^2}{m-\sum_\mathrm{i}(1-\boldsymbol{\alpha}_\mathrm{i}\,\mathrm{N}_\mathrm{ii})}$, where N_ii is the i-th diagonal element of $\boldsymbol{\Sigma}$

RVM has an iterative structure in updating the hyperparameters $\boldsymbol{\alpha}$, σ^2, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$. Upon its convergence, it can be observed that many of the hyperparameters $\boldsymbol{\alpha}$ are driven to very large values. Thus, the corresponding posterior variances become nearly zero, which means that there is a high probability that the corresponding coefficients in **w** are peaked around zero. This is advantageous as the resulting posterior mean $\boldsymbol{\mu}$ becomes sparse. The iterative procedures of RVM is presented in Figure A1., and explained in the following:

```
RVM Algorithm
Step 1. Initialization (t=1)
        Initiate the hyperparameters $\boldsymbol{\alpha}^1 = \mathbf{1}$,$\sigma_1^2 = 0.01 \times var(\mathbf{y})$
        Set the support set $S = \{1, 2, \dots, n\}$
Step2. Estimate posterior covariance and posterior mean:
        $\boldsymbol{\Sigma}_S = (\sigma_t^{-2}\boldsymbol{\Phi}_S^T\ \boldsymbol{\Phi}_S\ + \mathbf{B}_S)^{-1}$,
        $\boldsymbol{\mu}_S = \sigma_t^{-2}\boldsymbol{\Sigma}_S\boldsymbol{\Phi}_S^T\mathbf{y}$,
        $\mathbf{B}_S = \mathrm{Diag}(\boldsymbol{\alpha}_S^t)$
Step3. Estimate the hyperparameters corresponds to estimated support set
$\boldsymbol{\alpha}_S^t$, and noise variance $\sigma_t^2$:
        $\boldsymbol{\alpha}_i^t = \frac{1-\boldsymbol{\alpha}_i^{t-1}\mathrm{N}_{\mathrm{ii}}}{\boldsymbol{\mu}_{\mathrm{i}}^2}$ and $\sigma_t^2 = \frac{\|\mathbf{y}-\boldsymbol{\Phi}_S\boldsymbol{\mu}_S\|^2}{m-\sum_{\mathrm{i}}(1-\boldsymbol{\alpha}_i^t\,\mathrm{N}_{\mathrm{ii}})}\ \forall \mathrm{i} \in S$
Step 4. Update support set $S$
        Let $Q = \{v{:}\ \boldsymbol{\alpha}_v^t > \rho\ \forall v \in S)$, then $S = S \setminus Q$
Step 5. Stop if $\max|\boldsymbol{\alpha}_i^t - \boldsymbol{\alpha}_i^{t-1}| < 10^{-6}\ \forall \mathrm{i} \in S$,
        Otherwise increment $t = t + 1$ and go to Step 2
```

Figure A1. RVM algorithm for sparse vector estimation

In the initialization step, i.e., iteration $t = 1$, RVM sets the support set $S = \{1,2,..,n\}$, $\boldsymbol{\alpha}^1 = \mathbf{1}$, and $\sigma_1^2 = 0.01 \times var(\mathbf{y})$. In step 2, RVM estimates the posterior mean $\boldsymbol{\mu}_S = \sigma_t^{-2}\boldsymbol{\Sigma}_S\boldsymbol{\Phi}_S^T\mathbf{y}$ and posterior covariance $\boldsymbol{\Sigma}_S = (\sigma_t^{-2}\boldsymbol{\Phi}_S^T\boldsymbol{\Phi}_S + \mathbf{B}_S)^{-1}$ associated with the support set S, where $\mathbf{B}_S = Diag(\boldsymbol{\alpha}_S^t)$, with $\boldsymbol{\alpha}_S^t$ is a subvector of $\boldsymbol{\alpha}$ associated with estimated support set S at iteration t.

Step 3 of RVM estimates the hyperparameters $\boldsymbol{\alpha}_S{}^{\mathrm{t}}$ and the variance noise σ_t^2 from data by maximizing the marginal likelihood. The update for the hyperparameters $\boldsymbol{\alpha}_S^t$ contains the elements $\boldsymbol{\alpha}_i^t = \frac{1-\boldsymbol{\alpha}_i^{t-1}\mathrm{N}_{\mathrm{ii}}}{\boldsymbol{\mu}_{\mathrm{i}}^2}$ $(\forall i \in S)$ where N_{ii} is the diagonal element of $\boldsymbol{\Sigma}_S$ corresponds to index i, and the noise variance is given by $\sigma_t^2 = \frac{\|\mathbf{y}-\boldsymbol{\Phi}_S\boldsymbol{\mu}_S\|^2}{m-\sum_{\mathrm{i}}(1-\boldsymbol{\alpha}_i^t\,\mathrm{N}_{\mathrm{ii}})}$ at each iteration t.

In Step 4, let $Q = \{v{:}\ \boldsymbol{\alpha}_v^t > \rho\ \forall v \in S)$ (e.g. $\rho = 10^6$), then RVM updates the support set $S = S \setminus Q$. Note that updating the support set is very essential in reducing the computational time of the algorithm. The reason is that in practice it can be seen that after a few iterations some of the hyperparameters get very large values such that $\boldsymbol{\alpha}_v^t > \rho$ $(\forall v \in S)$ (their corresponding indices are denoted by Q). Then we can simply remove them from our computations, and accordingly update the support set by $S = S \setminus Q$. Thus, there will be less number of parameters to be updated. Finally, in Step 5, RVM stops if $\max\left|\boldsymbol{\alpha}_i^t - \boldsymbol{\alpha}_i^{t-1}\right| < 10^{-6}$ ∀i ∈ S, otherwise increments $t = t + 1$ and goes to Step 2.

Appendix E

Proof of Theorem 1

We provide some Lemmas along with their proofs, which is used for proof of Theorem 1.

Lemma 2. (see Ref. [5]): Suppose $\boldsymbol{\varepsilon} \sim N(0, \sigma^2\mathbf{I}) \in \mathbb{R}^m$, $\mathbf{g}_i$ is the i-th normalized column of matrix $\boldsymbol{\Phi} \in \mathrm{R}^{m\times n}$, then the event $U = \{max\ |{\mathbf{g}_i}^T\boldsymbol{\varepsilon}| < \tau,\ 1 \leq i \leq n\,\}$ where $\tau = \sigma\sqrt{2(1+\theta)logn}$ and $\theta > 0$, occurs with a probability of at least

$$1 - \frac{1}{\mathrm{n}^{\theta}\sqrt{\pi(1+\theta)logn}} \tag{A.17}$$

Lemma 2 will be used to prove that GBM determines the right support set S with at least the probability specified by Eq. (A.17**)**. Note that θ is a constant controls our confidence about event U, namely, larger θ results in higher probability (confidence).

Lemma 3. (see Ref. [5])*:* Let $\mathbf{w}$ be an unknown vector having a support set $\Lambda = \{i: \mathbf{w}_i \neq 0\ \forall i = 1, \cdots, n\}$ with cardinality of $|\Lambda| = \vartheta$, and $\mathbf{y} = \boldsymbol{\Phi}\mathbf{w} + \boldsymbol{\varepsilon}$ for some noise $\boldsymbol{\varepsilon}$ (defined in Lemma 2). Define $|\mathbf{w}_{max}|$ as the maximum absolute value of nonzero elements in sparse vector $\mathbf{w}$, and suppose that

$$|\mathbf{w}_{max}| \geq \frac{2\tau}{1 - (2\vartheta - 1)\tilde{\mu}} \tag{A.18}$$

where $\tau = \sigma\sqrt{2(1+\theta)logn}$, and $\tilde{\mu}$ denotes the mutual coherence of matrix $\boldsymbol{\Phi}$. Then if the event U defined in Lemma 2 holds, we have

$$\max_{i\in\Lambda} |{\mathbf{g}_i}^T\mathbf{y}| > \max_{i\notin\Lambda} |{\mathbf{g}_i}^T\mathbf{y}| \tag{A.19}$$

Based on Lemma 3, we can show that given the condition in Eq. (A.18), GBM algorithm determines the right candidate to estimated support set S at the first iteration. Equation (A.19) indicates that there is at least one index on the true support set Λ whose corresponding column in matrix $\mathbf{\Phi}$ has larger correlation with $\mathbf{y}$ than that of other indices not belonging to the true support set ($i \notin \Lambda$). Recall that the GBM algorithm at the first iteration $t = 1$ selects the index of columns $\mathbf{g}_i$'s which is maximally correlated with $\mathbf{y}$, namely $i^* = \arg\max_{i \in S^c}\{|\mathbf{g}_i{}^T\mathbf{y}|\}$, where $S^c = \{1,2, \ldots, n\}$. Thus, if Eq. (A.19) holds, then $i^* \in \Lambda$.

Lemma 4. Let $\mathbf{w}$ be an unknown vector having a support set $\Lambda = \{i: \mathbf{w}_i \neq 0 \; \forall i = 1, \cdots, n\}$ with cardinality of $|\Lambda| = \vartheta$, $\mathbf{w}_{\mathrm{GBM}}^{t-1}$be the GBM estimate of $\mathbf{w}$ at the $(t-1)$-th iteration$(1 < t \leq \vartheta)$ having a support set S with cardinality of $|S| = t - 1$, $\bar{\mathbf{w}}^{t-1} = \mathbf{w} - \mathbf{w}_{\mathrm{GBM}}^{t-1}$, and $\mathbf{r} = \mathbf{\Phi}\bar{\mathbf{w}}^{t-1} + \boldsymbol{\varepsilon}$ denotes the current measurement residual vector . Define $\bar{\mathbf{w}}_{max}^{t-1}$ as the maximum absolute value of $\bar{\mathbf{w}}^{t-1}$, and suppose that

$$\bar{\mathbf{w}}_{max}^{t-1} \geq \frac{2\sigma\sqrt{2(1+\theta)logn}}{1-(2\vartheta-1)\tilde{\mu}} \tag{A.20}$$

then if event U in Lemma 2 holds, and $S \subset \Lambda$, we have

$$\max_{i \in \Lambda}|\mathbf{g}_i{}^T\mathbf{r}| \; > \max_{i \notin \Lambda}|\mathbf{g}_i{}^T\mathbf{r}| \tag{A.21}$$

Proof: Following the same idea in proof of Lemma 3 (Ref. [5]), we first derive an upper bound for $\max_{i \notin \Lambda}|\mathbf{g}_i{}^T\mathbf{r}|$. By the definition of **r,**

$$\max_{i \notin \Lambda}|\mathbf{g}_i{}^T\mathbf{r}| \; = \max_{i \notin \Lambda}|\mathbf{g}_i{}^T\mathbf{\Phi}\bar{\mathbf{w}}^{t-1} + \mathbf{g}_i{}^T\boldsymbol{\varepsilon}|.$$

Based on the condition that $S \subset \Lambda$, $\mathbf{g}_i{}^T\mathbf{\Phi}\bar{\mathbf{w}}^{t-1} = \sum_{v\in\Lambda} \mathbf{g}_i{}^T\mathbf{g}_v\bar{\mathbf{w}}_v^{t-1}$ where $\bar{\mathbf{w}}_v^{t-1}$ represents a coefficient corresponds to an index $v \in \Lambda$ in $\bar{\mathbf{w}}^{t-1}$. Then we have

$$\begin{aligned} &\max_{i \notin \Lambda}|\mathbf{g}_i{}^T\mathbf{r}| \; = \max_{i \notin \Lambda}\left|\sum_{v\in\Lambda}\mathbf{g}_i{}^T\mathbf{g}_v\bar{\mathbf{w}}_v^{t-1} + \mathbf{g}_i{}^T\boldsymbol{\varepsilon}\right| \\ &\leq \max_{i \notin \Lambda}|\mathbf{g}_i{}^T\boldsymbol{\varepsilon}| + \max_{i \notin \Lambda}\left|\sum_{v\in\Lambda}\mathbf{g}_i{}^T\mathbf{g}_v\bar{\mathbf{w}}_v^{t-1}\right| \end{aligned} \tag{A.22}$$

$$\begin{aligned}&\le \max_{i\notin\Lambda}|\mathbf{g}_i{}^T\boldsymbol{\varepsilon}| + \sum_{v\in\Lambda}\max_{i\notin\Lambda}|\mathbf{g}_i{}^T\mathbf{g}_v||\bar{\mathbf{w}}_v^{t-1}|\\ &< \tau + \vartheta\tilde{\mu}\bar{\mathbf{w}}_{max}^{t-1}\end{aligned}$$

where $\bar{\mathbf{w}}_{max}^{t-1} = \max|\bar{\mathbf{w}}^{t-1}|$. Note that the first term at the end of Eq. (A.22) has been derived based on Lemma 2 showing that event $U = \{max\ |\mathbf{g}_i{}^T\boldsymbol{\varepsilon}| < \tau,\ 1 \le i \le n\}$ is held with a probability of at least Eq. (A.17). The second term at the end of Eq. (A.22) can be verified through the definition of mutual coherence, i.e. $|\mathbf{g}_i{}^T\mathbf{g}_v| \le \tilde{\mu}$.

Now we derive a lower bound on $\max_{i\in\Lambda}|\mathbf{g}_i{}^T\mathbf{r}|$ as follows:

$$\begin{aligned}\max_{i\in\Lambda}|\mathbf{g}_i{}^T\mathbf{r}| &= \max_{i\in\Lambda}\left|\bar{\mathbf{w}}_i^{t-1} + \sum_{v\in\Lambda-\{i\}}\mathbf{g}_i{}^T\mathbf{g}_v\bar{\mathbf{w}}_v^{t-1} + \mathbf{g}_v{}^T\boldsymbol{\varepsilon}\right|\\ &\ge \max_{i\in\Lambda}\left|\bar{\mathbf{w}}_i^{t-1}\right| - \max_{i\in\Lambda}\left|\sum_{v\in\Lambda-\{i\}}\mathbf{g}_i{}^T\mathbf{g}_v\bar{\mathbf{w}}_v^{t-1} + \mathbf{g}_v{}^T\boldsymbol{\varepsilon}\right|\\ &> \bar{\mathbf{w}}_{max}^{t-1} - \tau - (\vartheta-1)\tilde{\mu}\bar{\mathbf{w}}_{max}^{t-1}\\ &= \bar{\mathbf{w}}_{max}^{t-1} - (2\vartheta-1)\tilde{\mu}\bar{\mathbf{w}}_{max}^{t-1} - \tau + \vartheta\tilde{\mu}\bar{\mathbf{w}}_{max}^{t-1}\end{aligned} \tag{A.23}$$

Now comparing the two bounds in Eqs. (A.22) and (A.23), we conclude

$$\max_{i\in\Lambda}|\mathbf{g}_i{}^T\mathbf{r}| > \bar{\mathbf{w}}_{max}^{t-1} - (2\vartheta-1)\tilde{\mu}\bar{\mathbf{w}}_{max}^{t-1} - 2\tau + \max_{\forall i\notin\Lambda}|\mathbf{g}_i{}^T\mathbf{r}| \tag{A.24}$$

if the condition presented in Eq. (A.20) holds, QED.

GBM has an iterative way in determining the support set as presented in Figure 5-1. Lemma 4 proves that, given Eq. (A.20) and the condition $S \subset \Lambda$, we always have $\max_{i\in\Lambda}|\mathbf{g}_i{}^T\mathbf{r}| > \max_{i\notin\Lambda}|\mathbf{g}_i{}^T\mathbf{r}|$ with high probability. At iteration t, GBM should solve $i^* = \arg\max_{i\in S^c}\{|\mathbf{g}_i{}^T\mathbf{r}|\}$, where $\mathbf{r}$ is the current measurement residual vector. Since we already showed that $\max_{i\in\Lambda}|\mathbf{g}_i{}^T\mathbf{r}| > \max_{i\notin\Lambda}|\mathbf{g}_i{}^T\mathbf{r}|$, hence i^* always belongs to the true support set Λ, simply because at least there is one index on the true support set Λ whose corresponding correlation with $\mathbf{r}$ is always larger than that of all other indices which do not belong to the true support set ($i \notin \Lambda$). Therefore GBM determines the right candidate to the estimated support set at the t-th iteration ($1 < t \le \vartheta$).

Lemma 5. Let $\mathbf{\Phi}_{\Lambda} \in R^{m\times\vartheta}$ with $\text{rank}(\mathbf{\Phi}_{\Lambda}) = \vartheta$, where $\Lambda = \{i: w_i \neq 0\ \forall i = 1,\cdots,n\}$ with cardinality of $|\Lambda| = \vartheta$, and let $\mathbf{M} = \sigma_{\in}^{2}\mathbf{B}_{\Lambda} = \sigma_{\in}^{2}Diag(\boldsymbol{\alpha}_{\Lambda})$ where $\boldsymbol{\alpha}_{\Lambda} = \{\boldsymbol{\alpha}_i: i \in \Lambda\}$, $\boldsymbol{\alpha}_{max} = \max(\boldsymbol{\alpha}_{\Lambda})$, $\sigma_{\in}^{2}$ be the estimated variance noise from GBM, and $\lambda_{max}[\cdot]$ denotes the maximum eigenvalue of a matrix. Then,

$$\lambda_{max}[(\mathbf{I} + (\ \mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}\mathbf{M})^{-2}] \leq \left\{\frac{1 + (\vartheta - 1)\tilde{\mu}}{1 + (\vartheta - 1)\tilde{\mu} + \sigma_{\in}^{2}\ \boldsymbol{\alpha}_{max}}\right\}^{2} \quad \text{(A.25)}$$

$$\lambda_{\max}[\mathbf{M}(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-2}\mathbf{M}] \leq \left\{\frac{\sigma_{\in}^{2}\ \boldsymbol{\alpha}_{max}}{1 - (\vartheta - 1)\tilde{\mu}}\right\}^{2} \quad \text{(A.26)}$$

Proof: By Courant-Fischer Theorem [134], we can show that the minimum eigenvalue of the matrix $\mathbf{I} + (\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}\mathbf{M}$ is larger than the sum of minimum eigenvalues of $\mathbf{I}$ and $(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}\mathbf{M}$

$$\lambda_{\min}[\mathbf{I} + (\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}\mathbf{M}\ ] \geq 1 + \lambda_{\min}[(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}\mathbf{M}] \quad \text{(A.27)}$$

Using Corollary 11 in Ref. [28] we have

$$\lambda_{\min}[(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}\mathbf{M}\ ] \geq \lambda_{\min}[(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}]\lambda_{\max}[\mathbf{M}] \geq \frac{\sigma_{\in}^{2}\boldsymbol{\alpha}_{max}}{1 + (\vartheta - 1)\tilde{\mu}} \quad \text{(A.28)}$$

where we have utilized the Gershgorin disc theorem [135], by which it can be verified that $\lambda_{\min}[(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-1}] \geq \frac{1}{1+(\vartheta-1)\tilde{\mu}}$. Utilizing Eqs. (A.27) and (A.28), and the fact that $\lambda_{max}[\mathbf{V}^{-2}] \leq (\frac{1}{\lambda_{min}[\mathbf{V}]})^{2}$ where $\mathbf{V}$ is an arbitrary matrix, we can show Eq. (A.25). Now again using Corollary 11 in [29] then

$$\begin{aligned}\lambda_{\max}[\mathbf{M}(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-2}\mathbf{M}]\ &\leq \lambda_{\max}[\mathbf{M}]\lambda_{\min}[(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-2}\mathbf{M}] \\ &\leq \lambda_{\max}[\mathbf{M}]\lambda_{\max}[(\mathbf{\Phi}_{\Lambda}^{T}\mathbf{\Phi}_{\Lambda})^{-2}]\lambda_{\max}[\mathbf{M}] \quad \text{(A.29)}\end{aligned}$$

and this can be simplified into Eq. (A.26), QED.

We will utilize Lemma 5 to find the upper bounds for $\lambda_{\max}[(\mathbf{I}+(\boldsymbol{\Phi}_\Lambda^T\boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M})^{-1}]$ and $\lambda_{\max}[\mathbf{M}(\boldsymbol{\Phi}_\Lambda^T\boldsymbol{\Phi}_\Lambda)^{-2}\mathbf{M}]$ that will be used in Lemma 6 to derive the quadratic error bound for GBM estimate.

Lemma 6. Let $\mathbf{w}$ be an unknown vector with support set Λ =$\{i\colon w_i \neq 0\ \forall i=1,\cdots,n\}$, $\mathbf{y}=\boldsymbol{\Phi}\mathbf{w}+\boldsymbol{\varepsilon}$ for some noise $\boldsymbol{\varepsilon}$ (defined in Lemma 2), and let $\mathbf{w}_{GBM}$ be the GBM estimate, and $\boldsymbol{\alpha}_{max}$ and σ_ϵ^2 be the maximum precision and noise variance estimated from GBM, respectively. Assume GBM has estimated the true support set, namely $S=\Lambda$ under the event U in Lemma 2, and

$$|\mathbf{w}_{max}| \leq \vartheta_2 = \sqrt{2\beta\sigma^2(1+\theta)logn} \tag{A.30}$$

for some constant $\beta > \frac{4}{(1-(2\vartheta-1)\tilde{\mu})^2}$. If the dimension of measurements m satisfies the following

$$m > \vartheta + \frac{\vartheta[1+(\vartheta-1)\tilde{\mu}+C][(1+(\vartheta-1)\tilde{\mu})^2\beta-1]}{2[1+(\vartheta-1)\tilde{\mu}]} \tag{A.31}$$

where $C=\frac{m[1-(2\vartheta-1)\tilde{\mu}]^2}{\vartheta(1+\theta)\log(n)}$ for some positive constant θ, then with a probability of at least

$$(1-\frac{1}{\mathrm{n}^\theta\sqrt{\pi(1+\theta)\log n}})\cdot(1-2e^{-\frac{m}{2}}) \tag{A.32}$$

the error bound of $\mathbf{w}_{\mathrm{GBM}}$ is bounded as

$$\|\mathbf{w}_{\mathrm{GBM}}-\mathbf{w}\|_2^2 < 2\vartheta\sigma^2(1+\theta)\mathrm{log}n\frac{1}{(1-(\vartheta-1)\tilde{\mu})^2} \tag{A.33}$$

Proof: We assume that GBM has estimated the true support set namely $S=\Lambda$, then $\mathbf{w}$ and $\mathbf{w}_{\mathrm{GBM}}$ have the same support set, thus

$$\|\mathbf{w}_{\mathrm{GBM}}-\mathbf{w}\|_2^2 = \|\widehat{\mathbf{w}}-\mathbf{w}_\Lambda\|_2^2 = \|\boldsymbol{\mu}_\Lambda-\mathbf{w}_\Lambda\|_2^2 \tag{A.34}$$

Recall that $\boldsymbol{\mu}_\Lambda$ is the estimated posterior mean from GBM, and let $\boldsymbol{\Sigma}_\Lambda=(\boldsymbol{\Phi}_\Lambda^T\boldsymbol{\Phi}_\Lambda+\sigma_\epsilon^2\mathbf{B}_\Lambda)^{-1}$, then it can be shown that $\boldsymbol{\mu}_\Lambda=\boldsymbol{\Sigma}_\Lambda\boldsymbol{\Phi}_\Lambda^T\mathbf{y}$. Here we just moved the noise variance σ_ϵ^2 term into the inverse matrix for simplification purpose. Now $\boldsymbol{\mu}_\Lambda$ can be written in terms of the oracle estimator $\mathbf{w}_{\mathrm{OE},\Lambda}$ Eq. (74), i.e.

$$\boldsymbol{\mu}_\Lambda = \boldsymbol{\Sigma}_\Lambda {\boldsymbol{\Phi}_\Lambda}^T \boldsymbol{\Phi}_\Lambda \mathbf{w}_{\mathrm{OE},\Lambda} = (I + (\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M})^{-1}\mathbf{w}_{\mathrm{OE},\Lambda} \tag{A.35}$$

where $\mathbf{M} = \sigma_\in^2 \mathbf{B}_\Lambda$, and then,

$$\begin{aligned}
\|\boldsymbol{\mu}_\Lambda - \mathbf{w}_\Lambda\|_2^2 &= \left\|(\mathbf{I} + (\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M})^{-1}\mathbf{w}_{\mathrm{OE},\Lambda} - \mathbf{w}_\Lambda\right\|_2^2 \\
&= \left\|(\mathbf{I} + (\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M})^{-1}\left[\mathbf{w}_{\mathrm{OE},\Lambda} - (\mathbf{I} + (\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M})\mathbf{w}_\Lambda\right]\right\|_2^2 \\
&\leq \lambda_{max}[(\mathbf{I} + (\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M})^{-2}]\left\|\mathbf{w}_{\mathrm{OE},\Lambda} - \mathbf{w}_\Lambda - (\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M}\mathbf{w}_\Lambda\right\|_2^2 \\
&\leq \lambda_{max}\left[\left(\mathbf{I} + \left({\boldsymbol{\Phi}_S}^T \boldsymbol{\Phi}_S\right)^{-1}\mathbf{M}\right)^{-2}\right]\left\{\left\|\mathbf{w}_{\mathrm{OE},\Lambda} - \mathbf{w}_\Lambda\right\|_2^2 + \left\|(\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M}\mathbf{w}_\Lambda\right\|_2^2\right\}
\end{aligned} \tag{A.36}$$

where $\lambda_{max}(.)$ denotes the maximum eigenvalue of a matrix. Ben-Haim et al. [5] provides an upper bound for the oracle estimator $\mathbf{w}_{\mathrm{OE},\Lambda}$ error as follows,

$$\|\mathbf{w}_{\mathrm{OE}} - \mathbf{w}\|_2^2 = \left\|\mathbf{w}_{\mathrm{OE},\Lambda} - \mathbf{w}_\Lambda\right\|_2^2 \leq \epsilon_0 = 2\vartheta\sigma^2(1+\theta)\log n \frac{1}{(1-(\vartheta-1)\tilde{\mu})^2} \tag{A.37}$$

Using the result of Lemma 5 (Eqs. (A.25) and (A.26)), and Eq. (A.37) the upper bound on Eq. (A.36) is derived as follows

$$\|\boldsymbol{\mu}_\Lambda - \mathbf{w}_\Lambda\|_2^2 \leq \left\{\frac{1+(\vartheta-1)\tilde{\mu}}{1+(\vartheta-1)\tilde{\mu}+\sigma_\in^2\alpha_{max}}\right\}^2 \cdot \left\{\epsilon_0 + \left\|(\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M}\mathbf{w}_\Lambda\right\|_2^2\right\}$$

$$\begin{aligned}
\|\boldsymbol{\mu}_\Lambda - \mathbf{w}_\Lambda\|_2^2 &\leq \left\{\frac{1+(\vartheta-1)\tilde{\mu}}{1+(\vartheta-1)\tilde{\mu}+\sigma_\in^2\alpha_{max}}\right\}^2 \cdot \left\{\epsilon_0 + \left\|(\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-1}\mathbf{M}\mathbf{w}_\Lambda\right\|_2^2\right\} \\
&\leq \left\{\frac{1+(\vartheta-1)\mu}{1+(\vartheta-1)\tilde{\mu}+\sigma_\in^2\alpha_{max}}\right\}^2 \cdot \left\{\epsilon_0 + \lambda_{max}[\mathbf{M}(\boldsymbol{\Phi}_\Lambda^T \boldsymbol{\Phi}_\Lambda)^{-2}\mathbf{M}]\|\mathbf{w}_\Lambda\|_2^2\right\} \\
&\leq \left\{\frac{\mathbf{1}+(\boldsymbol{\vartheta}-\mathbf{1})\tilde{\boldsymbol{\mu}}}{\mathbf{1}+(\boldsymbol{\vartheta}-\mathbf{1})\tilde{\boldsymbol{\mu}}+\boldsymbol{\sigma}_\in^2\boldsymbol{\alpha}_{\mathbf{max}}}\right\}^2 \left\{\boldsymbol{\epsilon}_\mathbf{0} + \left\{\frac{\boldsymbol{\sigma}_\in^2\boldsymbol{\alpha}_{max}}{\mathbf{1}-(\boldsymbol{\vartheta}-\mathbf{1})\tilde{\boldsymbol{\mu}}}\right\}^2 \|\mathbf{w}_\Lambda\|_2^2\right\}
\end{aligned} \tag{A.38}$$

From the assumption in Lemma 5, $|\mathbf{w}_{max}|$ is bounded as below

$$|\mathbf{w}_{max}| \leq \sqrt{2\beta\sigma^2(1+\theta)logn} \tag{A.39}$$

with $\beta > \frac{4}{(1-(2\vartheta-1)\tilde{\mu})^2}$ which is necessary to meet the condition Eq. (75) on the minimum absolute value of nonzero element of the sparse vector $\mathbf{w}$. From the definition of l_2-norm, it can be verified that

$$\|\mathbf{w}_\Lambda\|_\mathbf{2}^\mathbf{2} \leq \vartheta(|\mathbf{w}_{max}|)^2 \leq \beta 2\vartheta\sigma^2(1+\theta)logn \tag{A.40}$$

Substitute Eq. (A.40) as an upper bound on $\|\mathbf{w}_\Lambda\|_\mathbf{2}^\mathbf{2}$ in Eq. (A.38) to derive

$$\|\boldsymbol{\mu}_\Lambda - \mathbf{w}_\Lambda\|_\mathbf{2}^\mathbf{2} \leq (1+\beta(\sigma_\in^2\boldsymbol{\alpha}_{\max})^2)\cdot\left\{\frac{1+(\vartheta-1)\tilde{\mu}}{1+(\vartheta-1)\tilde{\mu}+\sigma_\in^2\boldsymbol{\alpha}_{\max}}\right\}^2 \epsilon_0 \tag{A.41}$$

Let $\rho = 1+(\vartheta-1)\tilde{\mu})$ and $r = \sigma_\in^2\boldsymbol{\alpha}_{\max}$, and use Eqs. (A.33) and (A.41), then the quadratic error bound on GBM estimate is derived as

$$\|\mathbf{w}_{\text{GBM}} - \mathbf{w}\|_\mathbf{2}^\mathbf{2} \leq (1+\beta r^2)\cdot\left\{\frac{\rho}{\rho+r}\right\}^2 \epsilon_0 \tag{A.42}$$

Recall that ϵ_0 is the error bound on OE (see Eq. (A.37)) Now, given the error bound in Eq. (A.42), we would like to show that the GBM estimate is at least as good as OE, namely, $\|\mathbf{w}_{\text{GBM}} - \mathbf{w}\|_\mathbf{2}^\mathbf{2} < \epsilon_0$. In other words we need to verify

$$(1+\beta r^2)(\frac{\rho}{\rho+r})^2 < 1 \tag{A.43}$$

Clearly β and ρ depends on the given conditions of the sparse estimation problem, i.e., the sparsity level ϑ and mutual coherence $\tilde{\mu}$. But r is determined by the GBM algorithm. Equation (A.43) is equivalent to the following condition

$$r < \frac{2(1+(\vartheta-1)\tilde{\mu})}{\beta(1+(\vartheta-1)\tilde{\mu})^2-1} \tag{A.44}$$

Therefore, Eq. (A.44) is the sufficient condition to verify that $\|\mathbf{w}_{\text{GBM}} - \mathbf{w}\|_\mathbf{2}^\mathbf{2} < \epsilon_0$. We then use proof by contradiction as the following: We first assume that

$$\|\mathbf{w}_{\text{GBM}} - \mathbf{w}\|_2^2 \geq \epsilon_0 \tag{A.45}$$

, and later we show that Eq. (A.45) leads to a bound on $r = \sigma_{\in}^2 \boldsymbol{\alpha}_{\max}$ which satisfies Eq. (A.44). Hence this is a contradiction, and $\|\mathbf{w}_{\text{GBM}} - \mathbf{w}\|_2^2 < \epsilon_0$ must hold then.

We start the proof by deriving the bound on $r = \sigma_{\in}^2 \boldsymbol{\alpha}_{\max}$. Recall that we have assumed GBM has estimated the true support set $S = \Lambda$, then according to the estimate $\boldsymbol{\alpha}_{\text{i}}$'s ($i \in \Lambda$) and variance noise $\sigma_{\in}^2$ presented in step 3 of Figure 5-1 we have

$$r = \boldsymbol{\alpha}_{max}\sigma_{\in}^2 = \left[\frac{1 - \boldsymbol{\alpha}_{max}^{t-1}\text{N}_{idid}}{\boldsymbol{\mu}_{id}{}^2}\right] \cdot \left[\frac{\|\mathbf{y} - \boldsymbol{\Phi}_\Lambda \boldsymbol{\mu}_\Lambda\|_2^2}{\text{m} - \sum_{i\in\Lambda}(1 - \boldsymbol{\alpha}_i^t\ \text{N}_{\text{ii}}) \leq \vartheta}\right] \tag{A.46}$$

where id represents the index corresponds to the maximum of estimated $\boldsymbol{\alpha}_{max}$, and N_{idid} is the diagonal element corresponds to the index id on the estimated posterior covariance matrix $\boldsymbol{\Sigma}_\Lambda$. From [15] it is known that in general $1 - \boldsymbol{\alpha}_i^{t-1}\text{N}_{\text{ii}} \leq 1$ ($i \in \text{S} = \Lambda$), thus $1 - \boldsymbol{\alpha}_{max}^{t-1}\text{N}_{idid} \leq 1$, and $\sum_{i\in\Lambda}(1 - \boldsymbol{\alpha}_{\text{i}}\text{N}_{\text{ii}}) \leq \vartheta$. Let $\boldsymbol{\mu}_{min}$ be the minimum absolute value of $\boldsymbol{\mu}_\Lambda$, which means $\boldsymbol{\mu}_{id}{}^2 \geq \boldsymbol{\mu}_{min}{}^2$. Now we can bound r in Eq. (A.46) as follows,

$$r \leq \frac{\|\mathbf{y} - \boldsymbol{\Phi}_\Lambda \boldsymbol{\mu}_\Lambda\|_2^2}{(m-\vartheta)\boldsymbol{\mu}_{min}{}^2} = \frac{\|\boldsymbol{\Phi}_\Lambda \mathbf{w}_\Lambda + \boldsymbol{\varepsilon} - \boldsymbol{\Phi}_\Lambda \boldsymbol{\mu}_\Lambda\|_2^2}{(m-\vartheta)\boldsymbol{\mu}_{min}{}^2}$$
$$= \frac{\|\boldsymbol{\Phi}_\Lambda(\mathbf{w}_\Lambda - \boldsymbol{\mu}_\Lambda) + \boldsymbol{\varepsilon}\|_2^2}{(m-\vartheta)\boldsymbol{\mu}_{min}{}^2} \leq \frac{\lambda_{\max}(\boldsymbol{\Phi}_\Lambda^T\boldsymbol{\Phi}_\Lambda)\ \|\mathbf{w}_\Lambda - \boldsymbol{\mu}_\Lambda\|_2^2 + \|\boldsymbol{\varepsilon}\|_2^2}{(m-\vartheta)\boldsymbol{\mu}_{\min}{}^2}. \tag{A.47}$$

where we used the definition of $\mathbf{y} = \boldsymbol{\Phi}\mathbf{w} + \boldsymbol{\varepsilon} = \boldsymbol{\Phi}_\Lambda \mathbf{w}_\Lambda + \boldsymbol{\varepsilon}$. From [5], it can be shown that $\lambda_{max}(\boldsymbol{\Phi}_\Lambda^T\boldsymbol{\Phi}_\Lambda) \leq 1 + (\vartheta - 1)\tilde{\mu}$, then we have

$$r \leq \frac{[1 + (\vartheta - 1)\tilde{\mu}]\ \|\mathbf{w}_\Lambda - \boldsymbol{\mu}_\Lambda\|_2^2 + \ \|\boldsymbol{\varepsilon}\|_2^2}{(m-\vartheta)\mu_{\min}{}^2}$$
$$\leq \frac{\vartheta}{m-\vartheta} \cdot \frac{[1 + (\vartheta - 1)\tilde{\mu}]\ \|\mathbf{w}_\Lambda - \boldsymbol{\mu}_\Lambda\|_2^2 + \ \|\boldsymbol{\varepsilon}\|_2^2}{\|\mathbf{w}_\Lambda - \boldsymbol{\mu}_\Lambda\|_2^2} \tag{A.48}$$
$$= \frac{\vartheta}{m-\vartheta} \cdot \left[1 + (\vartheta - 1)\tilde{\mu} + \frac{\|\boldsymbol{\varepsilon}\|_2^2}{\|\mathbf{w}_\Lambda - \boldsymbol{\mu}_\Lambda\|_2^2}\right].$$

Recall that $\|\mathbf{w}_{\Lambda} - \boldsymbol{\mu}_{\Lambda}\|_2^2 = \|\mathbf{w}_{\mathrm{GBM}} - \mathbf{w}\|_2^2$, then substituting Eq. (A.45) into Eq. (A.48) results in

$$r \leq \frac{\vartheta}{m-\vartheta}\left[1 + (\vartheta - 1)\tilde{\mu} + \frac{1}{2\vartheta(1+\theta)\log n \frac{1}{(1-(\vartheta-1)\tilde{\mu})^2}} \cdot \frac{\|\boldsymbol{\varepsilon}\|_2^2}{\sigma^2}\right]. \tag{A.49}$$

In Eq. (A.49), let $\boldsymbol{z} = \frac{\boldsymbol{\varepsilon}}{\sigma} \sim N(0, \mathbf{I})$ where $\boldsymbol{\varepsilon} \in \mathbb{R}^m$, and $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 \mathbf{I})$, thus it can be shown that $\|\mathbf{z}\|^2 = \frac{\|\boldsymbol{\varepsilon}\|^2}{\sigma^2} \sim {\chi^2}_m$ with

$$rF_{{\chi^2}_m}(2m) \geq 1 - 2e^{-\frac{m}{2}}. \tag{A.50}$$

where $F_{{\chi^2}_m}(.)$ is the cumulative distribution function (CDF) of ${\chi^2}_m$ with m degree of freedoms [136]. Therefore, with probability of at least Eq. (A.50), we have $\|\mathbf{z}\|^2 \leq 2m$ which consequently leads to

$$r \leq \frac{\vartheta}{m - \vartheta}\left[1 + (\vartheta - 1)\tilde{\mu} + \frac{m(1 - (\vartheta - 1)\tilde{\mu})^2}{\vartheta(1 + \theta)\log n}\right] \tag{A.51}$$

Let $C = \frac{m(1-(\vartheta-1)\tilde{\mu})^2}{\vartheta(1+\theta)\log n}$ and simplify Eq. (A.51) as follows

$$r \leq \frac{\vartheta}{m - \vartheta}[1 + (\vartheta - 1)\tilde{\mu} + C] \tag{A.52}$$

From the condition in Lemma 6 (Eq. (A.31)) it can be easily verified that

$$\frac{\vartheta}{m - \vartheta}[1 + (\vartheta - 1)\tilde{\mu} + C] < \frac{2(1 + (\vartheta - 1)\tilde{\mu})}{\beta(1 + (\vartheta - 1)\tilde{\mu})^2 - 1} \tag{A.53}$$

In other words the assumption $\|\mathbf{w}_{\mathrm{GBM}} - \mathbf{w}\|_2^2 \geq \epsilon_0$ has led to the bound in Eq. (A.52) which is smaller than our sufficient condition in Eq. (A.44). There is a contradiction here, and $\|\mathbf{w}_{\mathrm{GBM}} - \mathbf{w}\|_2^2 < \epsilon_0$ must hold then. By this, we showed that under the conditions of Eqs. (A.30) and (A.31), GBM estimate is at least as good as OE $\|\mathbf{w}_{\mathrm{GBM}} - \mathbf{w}\|_2^2 < \epsilon_0$ with a probability of $(1 - \frac{1}{\mathrm{n}^{\theta}\sqrt{\pi(1+\theta)\log n}}) \cdot (1 - 2e^{-\frac{m}{2}})$, since this result is under the joint probability of the event of U

(to determine the true support set) as well as $\|\mathbf{z}\|^2 \leq 2m$ (see Eq. (A.51) to bound the performance bound).This completes the proof, Q.E.D.

We use Lemma 6 to derive an upper bound on the quadratic error bound for GBM, and verify that it is tighter than that of the OE estimator. Now in the following, we prove Theorem 1 based on Lemmas 2 to 6.

Proof for Theorem 1: Based on Lemma 2 and Lemma 3 we show that GBM correctly determines the first candidate to the estimated support set S with at least the probability specified by Eq. (A.17). Now following the same approach (i.e., by induction) as Ben Haim, *et al.* (See Ref. [5]), we show that GBM can correctly estimate the support set in ϑ iterations (i.e. $S = \Lambda$) with certain probabilities. Suppose that we are at the t-th iteration$(1 < t \leq \vartheta)$, and have correctly determined $(t-1)$ candidates to the estimated support set S so far. If we show that at iteration t, we can also determine the right candidate then the proof is completed. The proof is as the following:

Let $\mathbf{w}_{\text{GBM}}^{t-1}$ be the GBM estimate at iteration $t-1$, therefore $\left\|\mathbf{w}_{\text{GBM}}^{t-1}\right\|_0 = t-1$ with support set $S \subset \Lambda$. From Step 2 of Figure 5-1, the GBM algorithm at the t-th iteration is to find $\mathbf{g}_i$ which satisfies

$$i^* = \arg\max_{i \in S^c}\{|\mathbf{g}_i{}^T\mathbf{r}|\} \tag{A.54}$$

where $= \boldsymbol{\Phi}\bar{\mathbf{w}}^{t-1} + \boldsymbol{\varepsilon}$, $\bar{\mathbf{w}}^{t-1} = \mathbf{w} - \mathbf{w}_{\text{GBM}}^{t-1}$, and $S^c = \{1,2,\dots,n\} \setminus S$. In Lemma 4, we showed that if Eq. (A.20) holds, the GBM rule presented in Eq. (A.54) determines the right candidate to the estimated support set since $\max_{i\in\Lambda}|\mathbf{g}_i{}^T\mathbf{r}| > \max_{i\notin\Lambda}|\mathbf{g}_i{}^T\mathbf{r}|$. However, Eq. (A.20) automatically is satisfied if Eq. (75) in Theorem 1 holds. The reason is that, since we have determined the right candidates to the estimated support set S so far, the support set for $\bar{\mathbf{w}}^{t-1}$ is a subset of true support set Λ with cardinality of less than or equal ϑ. Thus, there is at least an element in $\bar{\mathbf{w}}^{t-1}$ equals to the corresponding element in $\mathbf{w}$ which means

$$\max\left|\bar{\mathbf{w}}^{j-1}\right| \geq |\mathbf{w}_{\min}| \geq \frac{2\sigma\sqrt{2(1+\theta)logn}}{1-(2\vartheta-1)\tilde{\mu}} \tag{A.55}$$

Recall that $\max\left|\bar{\mathbf{w}}^{j-1}\right| = \bar{\mathbf{w}}_{max}^{t-1}$, hence $\bar{\mathbf{w}}_{max}^{t-1} \geq \frac{2\sigma\sqrt{2(1+\theta)logn}}{1-(2\vartheta-1)\tilde{\mu}}$. Therefore by induction, we proved that if the condition Eq. (75), GBM determines the right support set, namely $S = \Lambda$ in ϑ iterations. Now, given the right support set estimated by GBM, we utilize Lemma 5 and Lemma 6 to derive the GBM quadratic error bound. From the conditions in Lemma 6 (Eqs. (A.30) and (A.31)), it is can be verified $\|\mathbf{w}_{\mathrm{GBM}} - \mathbf{w}\|_{\mathbf{2}}^{\mathbf{2}} < \epsilon_0$. This means the GBM estimate is at least as good as OE, (i.e., the error bound of GBM is tighter than that of the OE). Note that all the procedures of this proof is under the joint probability of the event of U (to determine the true support set) as well as $\|\mathbf{z}\|^2 \leq 2m$ (see Eq. (A.51) to bound the performance bound), namely,

$$1 - \frac{1}{\mathrm{n}^{\theta}\sqrt{\pi(1+\theta)logn}} . 1 - 2e^{-\frac{m}{2}}$$

This completes the proof, QED.