IISE Transactions

ISSN: 2472-5854 (Print) 2472-5862 (Online) Journal homepage: https://www.tandfonline.com/loi/uiie21

Multi-Sensor Prognostics Modeling for Applications with Highly Incomplete Signals

Xiaolei Fang, Hao Yan, Nagi Gebraeel & Kamran Paynabar

To cite this article: Xiaolei Fang, Hao Yan, Nagi Gebraeel & Kamran Paynabar (2020): Multi-Sensor Prognostics Modeling for Applications with Highly Incomplete Signals, IISE Transactions, DOI: 10.1080/24725854.2020.1789779

To link to this article: https://doi.org/10.1080/24725854.2020.1789779

Accepted author version posted online: 01 Jul 2020.

Multi-Sensor Prognostics Modeling for Applications with Highly Incomplete Signals

Xiaolei Fang

Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC

Hao Yan

School of Computing, Informatics, & Decision Systems Engineering, Arizona State University, Tempe, AZ

Nagi Gebraeel

H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA

Kamran Paynabar

H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA

Corresponding author Xiaolei Fang xfang8@ncsu.edu

Abstract

Multi-stream degradation signals have been widely used to predict the residual useful lifetime of partially degraded systems. To achieve this goal, most of the existing prognostics models assume that degradation signals are complete, i.e., they are observed continuously and frequently at regular time grids. In reality, however, degradation signals are often (highly) incomplete, i.e., containing missing and corrupt observations. Such signal incompleteness poses a significant challenge for the parameter estimation of prognostics models. To address this challenge, this paper proposes a prognostics methodology that is capable of using highly incomplete multi-stream degradation signals to predict the residual useful lifetime of partially degraded systems. The method first employs multivariate functional principal

components analysis to fuse multi-stream signals. Next, the fused features are regressed against times-to-failure using (log)-location-scale regression. To estimate the fused features using incomplete multi-stream degradation signals, we develop two computationally efficient algorithms: *subspace detection* and *signal recovery*. The performance of the proposed prognostics methodology is evaluated using simulated datasets and a degradation dataset of aircraft turbofan engines from NASA repository.

Key words. RUL, degradation modeling, multi-stream signal fusion, missing data

1 Introduction

Inexpensive sensor technology has allowed many original equipment manufacturers to install numerous sensors on their products, especially capital-intensive assets. These sensors are used to detect faults and determine the severity of an asset's degradation state through condition monitoring. Prognostics is the process of transforming raw condition monitoring data into high-fidelity degradation signals to predict the residual useful lifetime (RUL) of an asset. Several types of multi-stream prognostics models have been proposed in the literature. Examples include neural network models (Xu et al., 2014), neuro-fuzzy methods (Gouriveau and Zerhouni, 2012), parametric models that utilize data aggregation and fusion methods (Liu et al., 2013; Liu and Huang, 2014; Liu et al., 2015; Yan et al., 2016; Chehade et al., 2017, 2018; Song et al., 2017; Song and Liu , 2018), and functional principal component analysis (Liao and Sun, 2011; Fang et al., 2017a,b). Most of these models were developed on the premise that degradation signals are observed *with high fidelity at frequent time steps*, which we refer to as *complete signals* (see Figure 1(a) for an example). In reality, however, many of these complex assets operate in harsh environments that often have a significant impact on the quality of the raw data due to errors in data acquisition, communication, read/write operations, etc. In addition, different systems may have different data acquisition/sampling frequencies. Consequently, the resulting degradation

signals often contain significant levels of missing and corrupt observations (a.k.a. *incomplete signals*) as illustrated in Figure 1 (b). Such signal incompleteness results in inaccurate or even intractable parameter estimation for most of the existing prognostics models, and thus compromises their predictability. To address this challenge, this paper focuses on developing a multi-sensor prognostics methodology for capital-intensive assets with highly incomplete degradation signals.

The prognostics model in this paper is developed by integrating multivariate functional principal component analysis (MFPCA) and (log)-location-scale (LLS) regression. MFPCA has been used in prognostics (Fang et al., 2017a,b). It is a nonparametric functional data analysis technique that captures the joint variation of multi-stream functional data (i.e., degradation signals in our paper). It reduces the dimensionality of multi-stream degradation signals and provides low-dimensional fused features known as FPC-scores. The FPC-scores are then regressed against time-to-failure (TTF) using LLS regression. LLS regression has been widely used in reliability engineering and survival analysis since their response variables are general and include a variety of time-to-failure (TTF) distributions, such as (log)-normal, (log)-logistics, smallest extreme value (SEV), and Weibull (Doray, 1994). The regression coefficients of the LLS regression-based prognostics model can be estimated using maximum likelihood estimation (MLE) given the FPC-scores (Doray, 1994). FPC-scores can be calculated by projecting degradation signals onto the centered degradation signal matrix's (left) singular vectors, which can be estimated using singular value decomposition (SVD) or eigen decomposition (ED). However, SVD and ED are not suitable for incomplete degradation signals. To address this challenge, this paper proposes two new algorithms to estimate FPC-scores in applications involving (highly) incomplete degradation signals.

The first algorithm, which we designate as *subspace detection*, is inspired by the fact that the singular vectors of the centered degradation signal matrix can be computed in different coordinate systems. In other words, in different coordinate systems, the singular vectors of the centered degradation signal

matrix remain unchanged except that they have different coordinate values. Therefore, we will first build a new coordinate system (relative to the default coordinate system that degradation signals are in). Next, the incomplete degradation signals are projected into the new coordinate system to get complete projected signals. Third, SVD is applied to the projected signals (in the new coordinate system) to get singular vectors, which are then transformed back to the default coordinate system. In detail, the proposed *subspace detection* method comprises the following steps: (1) use the incomplete signals to extract a set of orthogonal basis vectors that span the column space of degradation signals (these orthogonal basis vectors define the new coordinate system); (2) expand the incomplete degradation signals as a linearly weighted combination of the basis vectors. By doing so, the degradation signal from each system is represented by a weight vector, which contains new coordinate values of the signal in the new coordinate system. (3) construct a weight matrix using the weight vectors and center the matrix; (4) apply SVD on the centered matrix to compute singular vectors; (5) carry out an inverse transformation to transform the singular vectors back to the default coordinate system. The second algorithm, which we refer to as *signal recovery*, starts with recovering complete degradation signals using their incomplete observations. Next, the recovered signals from all the sensors are concatenated to construct a signal matrix. Finally, the signal matrix is centered and the centered matrix is used to compute singular vectors and FPC-scores via a newly developed incremental SVD algorithm, which is computationally efficient and memory economic.

The remainder of the paper is organized as follows. In Section 2, we review some of the relevant papers that also focus on developing prognostics models for missing data applications. In section 3 we present the prognostics methodology. We then discuss the *subspace detection* algorithm in Section 4.1 and the *signal recovery* algorithm in Section 4.2. The performance of our model is evaluated using simulated datasets in Section 5 and an aircraft turbofan engine degradation dataset in Section 6. Finally, Section 7 concludes.

2 Literature review

Many prognostics models have been developed in the literature (Xu et al., 2014; Gouriveau and Zerhouni, 2012; Liu et al., 2013; Liu and Huang, 2014; Liu et al., 2015; Yan et al., 2016; Chehade et al., 2017, 2018; Song et al., 2017; Song and Liu , 2018). However, most of them assume that degradation signals are complete. Several key papers have investigated degradation modeling in the context of missing observations (Song et al., 2019; Zhou et al., 2011, 2012; Fang et al., 2015; Liao and Sun, 2011; Sun et al., 2014). Song et al. (2019) developed a supervised classification-based prognostics model, which can be used for applications with incomplete multi-sensor degradation signals. Specifically, the authors first used random-effect models to estimate the underlying signal paths. The estimated underlying signal paths were then utilized to estimate a multi-dimensional failure surface via supervised classification. The method proposed by Song et al. (2019) is an interesting and novel attempt to predict the residual lifetime of systems with multi-stream degradation signals and the numerical studies indicated that the proposed method is promising. However, the method in Song et al. (2019) is based on random-effect models, which are challenging to use for many applications since it is often difficult to determine the parametric form of underlying degradation paths (Zhou et al., 2011). This challenge is significantly augmented if degradation signals are highly incomplete, which is the case considered by this paper. For example, it is usually difficult to determine an appropriate parametric form of a signal when only 5% of its observations are available. Unlike Song et al. (2019), our proposed method is semi-parametric and does not need a parametric model to characterize the underlying degradation process.

Zhou et al. (2011), Zhou et al. (2012), and Fang et al. (2015) focused on prognostics models for applications with fragmented and sparsely observed degradation signals. Liao and Sun (2011) and Sun et al. (2014) proposed methodologies to recover missed observations of incomplete signals. All the methodologies in Zhou et al. (2011), Zhou et al. (2012), Fang et al. (2015), Liao and Sun (2011), and Sun et al. (2014) were based on functional principal

component analysis (FPCA) and kernel smoother. FPCA was employed to reduce the dimensionality of degradation signals and provide low-dimensional fused features (i.e., FPC-scores), which were estimated from incomplete degradation signals using kernel smoother algorithms (Yao et al., 2005). A kernel smoother is a statistical technique to estimate a real-valued function as the weighted average of neighboring observed data. The weight is defined by a kernel function, such that closer points are given higher weights. Since all the prognostics models in Zhou et al. (2011), Zhou et al. (2012), Fang et al. (2015), Liao and Sun (2011), and Sun et al. (2014) used FPCA and kernel smoother, they share some common limitations. The first limitation is the assumption that degradation signals share the same time domain–that is– signals have the same length. In reality, however, signals have different lengths due to truncation. To be specific, a system is usually stopped for repair or replacement when its degradation signal crosses a predefined " failure threshold," and, thus, no further observations can be acquired beyond that point. In such scenarios, using FPCA results in a significantly biased estimate of the mean and covariance functions (Zhou et al., 2012). To address this challenge, Zhou et al. (2012) proposed a procedure that relied on axis transformation. However, the approach was limited to strictly monotonic signals with very low noise levels. An alternative approach, time-varying regression, was proposed by Fang et al. (2015) to address the time-domain challenge. The framework iteratively selects the training systems and truncates their signals so that the truncated training signals and the test signal have the same length. One limitation of the time-varying structure in Fang et al. (2015) is that it is computationally expensive since it chooses new training datasets at different time points, which requires re-estimating the model each time a new observation is observed from a fielded system. In addition, it does not make full use of the available dataset. To address this challenge, this paper proposes a signal transformation methodology based on polar-domain transformation. The transformation of degradation signals from the time domain to the polar domain allows them to have the same lengths. The details of the polar transformation method are discussed in Section 3.1. The second common limitation for the prognostics models in Zhou et al. (2011), Zhou et

al. (2012), Fang et al. (2015), Liao and Sun (2011), and Sun et al. (2014) is that they are computationally expensive. The computational burden mainly results from the kernel smoothers, which are used to estimate the signal features (i.e., FPC-scores). Specifically, a one-dimensional kernel smoother is used to estimate the mean function, and a two-dimensional kernel smoother is utilized to smooth the covariance function. It is well-known that kernel smoothers are computationally intensive (Yao et al., 2005), especially for large-scale signal matrices/covariance matrices. To reduce the computational burden, this paper presents computational efficient estimation algorithms, which will be discussed in detail in Section 4.

3 Degradation modeling and prognostics framework

This paper focuses on developing a prognostics model for systems that are monitored by multiple sensors. We assume that data from each sensor is synthesized into one type of degradation signal. The prognostics model is established by using functional (log)-location-scale (LLS) regression, in which the covariate is the multi-stream degradation signals and the response is TTF. We denote the degradation signal for the pth sensor of system i as $x_{i,p}(t)$, for $i = 1,\ldots,N$ and $p = 1,\ldots,P$. Here, N is the number of systems (we use a system to refer an asset, piece of equipment, or machine), P is the number of sensor equipped on each system, and $t \in [0,T]$ is degradation time. Denote the TTF of system i as $\tilde{y}_i$, we build the following functional LLS regression to model the relationship between the TTF and degradation signals:

$$y_i = \gamma_0 + \int_0^T \boldsymbol{\gamma}(t)^{\cdot} \boldsymbol{x}_i(t)dt + \sigma\grave{o}_i \quad (1)$$

where $y_i = \tilde{y}_i$ for a location-scale model and $y_i = \ln(\tilde{y}_i)$ for a log-location-scale model. γ_0 is the intercept, $\boldsymbol{\gamma}(t) = (\gamma_1(t),\ldots,\gamma_P(t))^{\cdot}$ is the regression coefficient function and $\boldsymbol{x}_i(t) = (x_{i,1}(t),\ldots,x_{i,P}(t))^{\cdot}$ is the concatenated degradation signal from all P sensors. σ is the scale parameter and $\grave{o}_i$ is the random noise term with a standard location-scale density $f(\grave{o})$. For example, $f(\grave{o}) = \exp(\grave{o} - \exp(\grave{o}))$ for SEV distribution and $f(\grave{o}) = 1/\sqrt{2\pi}\exp(-\grave{o}^2/2)$ for

normal distribution. Consequently, y_i has a density in the form of $\frac{1}{\sigma} f\left(\frac{y_i - \gamma_0 - \int_0^T \boldsymbol{\gamma}(t)^{\cdot} \boldsymbol{x}_i(t)dt\rangle}{\sigma}\right)$.

3.1 Polar-domain transformation of degradation signals

Functional LLS regression in Equation (1) requires that the degradation signals from all the systems share the same time domain. In reality, however, degradation signals are observed in different time domains since they usually are truncated by a predefined failure threshold. Specifically, a system is considered to be failed if its degradation signal crosses the failure threshold. Once failed, the system is stopped for repair or replacement, and thus no further observation can be acquired beyond that point. As an illustration, Figure 2(a) shows degradation signals from five systems measured by one sensor. In the figure, only the solid portions can be observed. To address this challenge, we propose a polar coordinate transformation method. It can be seen from Figure 2(a) that the observable portions of all the degradation signals share the same polar domain, $[0, \pi/2]$. This inspires us to re-express degradation signals use their polar coordinates. To be specific, we apply the following transformation on observation $(t, x_{i,p}(t))$:

$$\begin{cases} \theta = \arctan\left(\dfrac{t}{m_p - x_{i,p}(t)}\right) \\ r_{i,p}(\theta) = \sqrt{t^2 + (m_p - x_{i,p}(t))^2}, \end{cases} \quad (2)$$

where m_p is the failure threshold for sensor p. By applying the transformation in Equation 2, all the transformed signals from sensor p, i.e., $\{r_{i,p}(\theta)\}_{i=1}^N$, share the same domain $[0, \frac{\pi}{2}]$ (see Figure 2(b)). As a result, the functional LLS regression model in Equation (1) can be expressed as follows:

$$y_i = \alpha_0 + \int_0^{\frac{\pi}{2}} \boldsymbol{\alpha}(\theta)^{\cdot} \boldsymbol{r}_i(\theta)d\theta + \sigma\grave{o}_i \quad (3)$$

where α_0 is the intercept, $\boldsymbol{\alpha}(\theta) = (\alpha_1(\theta), \ldots, \alpha_P(\theta))^{\cdot}$ is the regression coefficient and $\boldsymbol{r}_i(\theta) = (r_{i,1}(\theta), \ldots, r_{i,P}(\theta))^{\cdot}$ is the transformed degradation signals for system *i*. Note that although the polar coordinate transformation method requires the existence of a failure threshold for sensor *p*, the value of the threshold (i.e., m_p) is not required to be known. In reality, the maximum value of degradation observations from sensor *p* is used as the threshold for the coordinate transformation.

3.2 Multi-stream degradation signal fusion

The estimation of the functional LLS regression model in Equation (3) is nontrivial due to the existence of an integral. To address this challenge, we follow the method proposed in (Fang et al., 2017a). Specifically, we employ MFPCA to transform functional LLS regression to classic LLS regression. MFPCA is an extension of FPCA. It works by concatenating different types of degradation signals into a single vector, and FPCA is then applied to the concatenated vector in a conventional manner. One benefit of MFPCA is that it is capable of capturing the auto- and cross-correlation within/among signal streams and providing low-dimensional fused features known as FPC-scores.

Let the mean function of the degradation signals (i.e., $\{\boldsymbol{r}_i(\theta)\}_{i=1}^N$) be $\boldsymbol{\mu}(\theta) = (\mu_1(\theta), \ldots, \mu_P(\theta))^{\cdot}$ and its covariance function be $\boldsymbol{C}(\theta, \theta')$. Then $\boldsymbol{C}(\theta, \theta')$ is a $P \times P$ block matrix, where the (*g*, *h*)th block is the covariance function between sensors *g* and *h*, for $g = 1, \ldots, P$ and $h = 1, \ldots, P$, with $\theta, \theta' \in [0, \frac{\pi}{2}]$. Using Mercer's theorem, $\boldsymbol{C}(\theta, \theta')$ can be decomposed as $\boldsymbol{C}(\theta, \theta') = \sum_{k=1}^{\infty} \eta_k \boldsymbol{\psi}_k(\theta) \boldsymbol{\psi}_k(\theta')^{\cdot}$, where $\eta_1 \geq \eta_2 \geq \ldots,$ are eigenvalues, and $\boldsymbol{\psi}_k(\theta) = (\psi_{k,1}(\theta), \ldots, \psi_{k,P}(\theta))^{\cdot}$ for $k = 1, 2, \ldots$ are the corresponding eigenfunctions. Thus, we can rewrite $\boldsymbol{r}_i(\theta)$ as follows:

$$\boldsymbol{r}_i(\theta) = \boldsymbol{\mu}(\theta) + \sum_{k=1}^{\infty} \zeta_{i,k} \boldsymbol{\psi}_k(\theta), \quad (4)$$

where $\zeta_{i,k} = \int_0^{\frac{\pi}{2}} (\boldsymbol{r}_i(\theta) - \boldsymbol{\mu}(\theta))^{\cdot} \boldsymbol{\psi}_k(\theta) d\theta$ are the FPC-scores. It is often sufficient to use a few eigenfunctions corresponding to the largest eigenvalues to approximate signals with a reasonable accuracy. Using only K eigenfunctions, equation (4) can now be rewritten as $\boldsymbol{r}_i(\theta) = \boldsymbol{\mu}(\theta) + \sum_{k=1}^{K} \zeta_{i,k} \boldsymbol{\psi}_k(\theta)$. K can be determined using fraction-of-variance explained (FVE) or cross-validation (Yao et al., 2005). Since the set of eigenfunctions $\{\boldsymbol{\psi}_k(\theta)\}_{k=1}^{\infty}$ forms a complete set of orthonormal basis functions vectors, $\boldsymbol{\alpha}(\theta)$ can be expanded to $\boldsymbol{\alpha}(\theta) = \sum_{k=1}^{\infty} \beta_k \boldsymbol{\psi}_k(\theta)$ (Yao et al., 2011). Therefore, the functional LLS model in Equation (3) can be expressed as follows (details of the derivation can be found in (Fang et al., 2017a)):

$$y_i = \beta_0 + \boldsymbol{\beta}^{\cdot} \boldsymbol{\zeta}_i + \sigma\grave{o}_i, \quad (5)$$

where β_0 is the intercept, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)^{\cdot} \in \mathbb{R}^K$ is the coefficient and $\boldsymbol{\zeta}_i = (\zeta_{i,1}, \ldots, \zeta_{i,K})^{\cdot} \in \mathbb{R}^K$ is the FPC-score vector for system i, which can be estimated from degradation signals. Given $\boldsymbol{\zeta}_i$, the parameters $(\beta_0, \boldsymbol{\beta}, \sigma)$ in Equation (5) can be estimated using maximum likelihood estimation (Doray, 1994).

3.3 Fused feature estimation with complete signals

In this subsection, we discuss how to estimate the FPC-scores ($\boldsymbol{\zeta}_i$ in Equation (5)) when the degradation signals are complete. Denote the discrete observation time point in the polar domain for sensor p as $\{\Theta_{p,1}, \Theta_{p,2}, \ldots, \Theta_{p,J_p}\}$, where J_p is the number of observations for sensor p. Then, the discrete observations for sensor p of system i are $\boldsymbol{l}_{i,p} = (\boldsymbol{r}_{i,p}(\Theta_{p,1}), \boldsymbol{r}_{i,p}(\Theta_{p,2}), \ldots, \boldsymbol{r}_{i,p}(\Theta_{p,J_p}))^{\cdot} \in \mathbb{R}^{J_p}$. Thus, the concatenated signals from all the P sensors of system i is $\boldsymbol{s}_i = (\boldsymbol{l}_{i,1}, \boldsymbol{l}_{i,2}, \ldots, \boldsymbol{l}_{i,P})^{\cdot} \in \mathbb{R}^M$, where $M = \sum_{p=1}^{P} J_p$. Then, the observed degradation signal matrix from all the P sensors and N systems is represented by $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N) \in \mathbb{R}^{M \times N}$. Without loss of generality, we assume $N < M$. The FPC-scores in Equation (5) can be estimated as follows:

(i) Estimating signal mean. The signal mean is computed by taking the average over all the systems, i.e., $\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{s}_i$, where $\boldsymbol{s}_i$ is the *i*th column of $\boldsymbol{S}$.

(ii) Centralizing signal matrix. This is achieved by subtracting the signal of each system to the mean signal estimated in (i): $\tilde{\boldsymbol{s}}_i = \boldsymbol{s}_i - \hat{\boldsymbol{\mu}}$, for $i = 1, \ldots, N$. As a result, $\boldsymbol{S} = (\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \ldots, \tilde{\boldsymbol{s}}_N)$ is the centered signal matrix.

(iii) Estimating eigenvectors. Eigenvectors are estimated by solving the eigen equation $\boldsymbol{S}\boldsymbol{\psi} = \eta\boldsymbol{\psi}$, which can be achieved by applying SVD on matrix $\boldsymbol{S}$. The resulting eigenvalues associated with the eigenvectors are denoted as $\{\hat{\eta}_k, \boldsymbol{\psi}_k\}$ for $k = 1, \ldots, N$. Fraction-of-variance explained (FVE) is used to select the first *K* eigenvectors as follows: $K = \inf_k \{F_k \geq D\}$, where $F_k = \sum_{j=1}^{k} \eta_j^2 / \sum_{j=1}^{N} \eta_j^2$ and $D \in (0,1]$ is the FVE threshold (Fang et al., 2017a).

(iv) Computing the FPC-scores. The *k*th FPC-score of system *i* is $\zeta_{i,k} = \tilde{\boldsymbol{s}}_i^{\bullet} \boldsymbol{\psi}_k$, for $k = 1, \ldots, K$ and $\boldsymbol{\zeta}_i = (\zeta_{i,1}, \zeta_{i,2}, \ldots, \zeta_{i,K})^{\bullet}$.

Steps (i)-(iv) estimate the FPC-scores of multi-stream degradation signals only if the signals are complete. If the signals are incomplete, Steps (i)-(iv) cannot be used. In the following section, we develop two algorithms, namely *subspace detection* and *signal recovery*, to estimate FPC-scores from incomplete multi-stream degradation signals.

4 Fused feature estimation using incomplete degradation signals

In this section, we develop two new algorithms to estimate FPC-scores from incomplete multi-stream degradation signals. Both algorithms are inspired by the maximum margin matrix factorization (MMMF) method proposed in Srebro et al. (2005). MMMF is a widely used collaborative filtering (also known as matrix completion) method that focuses on recovering the unobserved entries of a partially observed matrix (i.e., the incomplete degradation signal matrix in this paper). It is formulated as a semi-definite program (SDP) that finds two low-norm latent factor matrices to simultaneously approximate the observed entries under some loss measure and predict the unobserved entries. From

the statistical learning point of view, MMMF can be seen as a signal expansion method, and the two factor matrices can be treated as a weight matrix and a basis matrix respectively. This is because each column of the recovered matrix can be expressed as a linearly weighted combination of a set of basis vectors, where the rows of the second factor matrix are the basis vectors and the corresponding column of the first factor matrix are the weights. The set of basis vectors in the second factor matrix spans the low-dimensional subspace that the recovered matrix lies in. Note that the low-dimensional subspace is the same subspace spanned by the left singular vectors of the signal matrix. This inspires us to propose our first algorithm, i.e., *subspace detection*, which computes the singular vectors of the signal matrix using a set of its general basis vectors from MMMF and computes FPC-Scores by projecting incomplete signals to the singular vectors. Our second algorithm, i.e., *signal recovery*, directly computes the singular vectors and FPC-scores using the recovered signal matrix from MMMF. To accomplish this, instead of using regular SVD, which is computationally intensive and memory expensive, we propose a novel incremental SVD that computes the singular vectors by adding one column (or row) of the signal matrix at a time.

Suppose that a historical dataset is available for model training and parameter estimation. The dataset contains incomplete multiple degradation signals from N systems. Denote the discrete observation time point (in polar domain) for sensor p as $\{\theta_{p,1}, \theta_{p,2}, \ldots, \theta_{p,J_p}\}$, where J_p is the number of unique observation time points for sensor p. We construct the degradation signal matrix $\boldsymbol{S}$ by concatenating multi-stream signals from all sensors (illustrated in Table 1).

The dimensionality of the signal matrix is $M \times N$, where $M = \sum_{p=1}^{P} J_p$ is the total number of unique observation timestamps from all sensors. The cross markers in $\boldsymbol{S}$ represent the entries whose values were observed and blank entries are the missing observations. We let $\Omega_i \subseteq \{1, 2, \ldots, M\}$ be the index of the entries observed in the ith row of $\boldsymbol{S}$ and $\boldsymbol{\Omega} = \{(i, j) : j \in \Omega_i, i = 1, \ldots, N\}$ be the index of observed entries of $\boldsymbol{S}$. Following the notations defined above, we

denote the incomplete signal matrix as $\boldsymbol{S}^{\Omega}$ and the concatenated incomplete degradation signals from system i as $\boldsymbol{s}_i^{\Omega_i}$.

To recover the missing entries of signal matrix $\boldsymbol{S}$ using its incomplete observations $\boldsymbol{S}^{\Omega}$, MMMF considers the following matrix factorization: $\boldsymbol{S} = \boldsymbol{XY}$, where $\boldsymbol{X} \in \mathbb{R}^{M \times K}$ and $\boldsymbol{Y} \in \mathbb{R}^{K \times N}$ are two factor matrices. K is the rank of signal matrix $\boldsymbol{S}$. Given the incomplete observed signal matrix $\boldsymbol{S}^{\Omega}$, MMMF focuses on finding two factor matrices such that their product has the smallest rank and best matches the observed entries of $\boldsymbol{S}^{\Omega}$. Mathematically, it can be formulated as the following optimization problem (Srebro et al., 2005; Mazumder et al., 2011):

$$\min_{X,Y} \| \boldsymbol{S}^{\Omega} - (\boldsymbol{XY})^{\Omega} \|_F^2 + \lambda(\| \boldsymbol{X} \|_F^2 + \| \boldsymbol{Y} \|_F^2), \quad (6)$$

where $\|\cdot\|_F^2$ is the Frobenius norm and λ is the tuning parameter. The rank K and tuning parameter λ can be selected using cross-validation. The first term in optimization criterion (6) guarantees that the recovered matrix (i.e., $\boldsymbol{XY}$) best matches the observed incomplete signal matrix $\boldsymbol{S}^{\Omega}$ and the second term guarantees the recovered matrix has the smallest rank. Many algorithms have been developed in the literature to solve optimization criterion (6), such as the ones in Toh and Yun (2010), Balzano et al. (2010), and Srebro et al. (2005). Solving optimization problem (6) gives factor matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, from which we can recover the signal matrix $\boldsymbol{S}$. In the following subsections, we introduce two algorithms to compute the FPC-Scores in Equation (5) using the factor matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$.

4.1 The subspace detection algorithm

As mentioned earlier, the solution of MMMF can be seen as a signal expansion technique that decomposes each degradation signal as a weighted linear combination of a set of basis vectors, where both the weights and basis vectors are estimated from the degradation signals' incomplete observations. For example, the signal from system i (i.e., $\boldsymbol{s}_i$, the ith column of signal matrix $\boldsymbol{S}$) is expanded as a weighted linear combination of the columns of $\boldsymbol{X}$, i.e., $\boldsymbol{s}_i = \boldsymbol{X}\boldsymbol{y}_i$, where the weights $\boldsymbol{y}_i \in \mathbb{R}^K$ is the ith column of $\boldsymbol{Y}$. Therefore, $\boldsymbol{X}$ is the

basis matrix and $\boldsymbol{Y}$ is the weight matrix. Since the columns of $\boldsymbol{X}$ constitute a set of basis vectors for the signal matrix $\boldsymbol{S}$, they span the column space of $\boldsymbol{S}$. In addition, since the signal matrix $\boldsymbol{S}$ and the centered signal matrix S have the same column space, the columns of $\boldsymbol{X}$ also span the column space of S.

Recall that MFPCA first computes the (left) singular vectors of the centered signal matrix S and then projects each column of S onto the singular vectors to compute FPC-scores. Therefore, computing the singular vectors of S is a crucial step for FPC-scores estimation. The unknown singular vectors also span the column space of S. In other words, both the unknown singular vectors and the columns of $\boldsymbol{X}$ span the same linear space. This inspires us to develop the following new algorithm to compute the singular vectors of S using $\boldsymbol{X}$:

Algorithm 1: Computing the SVD of S using $\boldsymbol{S}$ and $\boldsymbol{X}$.

Input: An uncentered matrix $\boldsymbol{S} \in \mathbb{R}^{M \times N}$ and one set of its basis vectors $\boldsymbol{X} \in \mathbb{R}^{M \times K}$

(1) Orthonormalize $\boldsymbol{X}$ to get a set of orthonormal basis $\boldsymbol{Q} \in \mathbb{R}^{M \times K}$:

This can be achieved by applying Gram-Schmidt orthonormalization, QR decomposition, or SVD on $\boldsymbol{X}$.

(2) Compute the projected weight matrix:

$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N) \in \mathbb{R}^{K \times N}$, where $\boldsymbol{w}_i = \boldsymbol{Q}^{\cdot} \boldsymbol{s}_i \in \mathbb{R}^{K}$ for $i = 1, \ldots, N$.

(3) Centralize the weight matrix:

$\boldsymbol{W} = \boldsymbol{W} - \overline{\boldsymbol{W}}$, where $\overline{\boldsymbol{W}} = (\overline{\boldsymbol{w}}, \overline{\boldsymbol{w}}, \ldots, \overline{\boldsymbol{w}})$ and $\overline{\boldsymbol{w}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{w}_i$.

(4) Apply SVD on the centered weight matrix:

$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\bullet}$, where $\boldsymbol{U} \in \mathbb{R}^{K\times K}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{K\times N}$ and $\boldsymbol{V} \in \mathbb{R}^{N\times N}$

(5) Set $\tilde{\boldsymbol{U}} = \boldsymbol{Q}\boldsymbol{U}, \tilde{\Sigma} = \Sigma, \tilde{\boldsymbol{V}} = \boldsymbol{V}$.

Output: SVD of the centered matrix $\boldsymbol{S} \in \mathbb{R}^{M\times N}$, which is $\boldsymbol{S} = \tilde{\boldsymbol{U}}\tilde{\Sigma}\tilde{\boldsymbol{V}}^{\bullet}$.

Proposition 1. *Given an uncentered matrix $\boldsymbol{S}$ and one set of its basis vectors $\boldsymbol{X}$, Algorithm 1 computes the SVD of the centered matrix $\boldsymbol{S}$.*

Proposition 1 suggests that Algorithm 1 indeed computes the singular vectors of $\boldsymbol{S}$ using $\boldsymbol{S}$ and $\boldsymbol{X}$ (the proof can be found in the appendix). From a geometrical point of view, these singular vectors are in a coordinate system (denoted by coordinate system I, which is also the coordinate system that the degradation signals lie in) in the column space of $\boldsymbol{S}$. To compute the singular vectors, Step (1) of Algorithm 1 constructs another coordinate system (denoted by coordinate system II) in the same column space. Step (2) calculates the coordinates of each system's degradation signal (i.e, each column of $\boldsymbol{S}$) in coordinate system II. Step (3) centers the coordinated signals and Step (4) applies SVD on the centered coordinated signals. In other words, Steps (2)–(4) implement SVD on the centered signals and compute the singular vectors in coordinate system II. Finally, Step (5) applies a coordinate system transformation, which transforms the singular vectors from coordinate system II to coordinate system I.

Algorithm 1 computes the singular vectors of the centered signal matrix $\boldsymbol{S}$ using the uncentered signal matrix $\boldsymbol{S}$ and a set of its basis vectors $\boldsymbol{X}$. However, Algorithm 1 is designated for complete signals only. This is because Step (2) computes the projection weights by using $\boldsymbol{w}_i = \boldsymbol{Q}^{\bullet}\boldsymbol{s}_i$, which requires $\boldsymbol{s}_i$ to be completely observed. In this paper, the observed degradation signal matrix $\boldsymbol{S}^{\Omega}$ is incomplete, and thus cannot directly be used to compute the projection weights. We notice that $\boldsymbol{s}_i$ is a weighted linear combination of the columns of $\boldsymbol{Q}$. Therefore, to address the missing data challenge, we compute the projected weights using $\min_{\boldsymbol{w}_i} \| \boldsymbol{Q}^{\Omega_i}\boldsymbol{w}_i - \boldsymbol{s}_i^{\Omega_i} \|^2$, where matrix $\boldsymbol{Q}^{\Omega_i} \in \mathbb{R}^{|\Omega_i|\times K}$

consists the $|\Omega_i|$ rows of matrix $\boldsymbol{Q}$ indexed by the set Ω_i. This results in the solution $\boldsymbol{w}_i = (\boldsymbol{Q}^{\Omega_i \bullet} \boldsymbol{Q}^{\Omega_i})^{-1} \boldsymbol{Q}^{\Omega_i \bullet} \boldsymbol{s}_i^{\Omega_i}$. Therefore, we revise the weight computation method in Step (2) such that Algorithm 1 works with an incomplete signal matrix.

Using matrix $\boldsymbol{X}$ from MMMF and the incomplete signal matrix $\boldsymbol{S}^{\Omega}$, we compute the SVD of the centered matrix $\boldsymbol{S} = \tilde{\boldsymbol{U}} \tilde{\Sigma} \tilde{\boldsymbol{V}}^{\bullet}$ via Algorithm 1. Then, the FPC-scores can be computed by using $\boldsymbol{Z} = \tilde{\boldsymbol{U}}^{\bullet} \boldsymbol{S} = \tilde{\Sigma} \tilde{\boldsymbol{V}}^{\bullet}$, where $\boldsymbol{Z} = (\zeta_1, \zeta_2, \ldots, \zeta_N) \in \mathbb{R}^{K \times N}$ and ζ_i is the FPC-score vector for system i (see Equation (5) for details).

The proposed subspace detection algorithm is computationally efficient and memory economic. To be specific, the main computational burden of the subspace detection algorithm comes from the computation of the SVD of matrix $\boldsymbol{W}$ with a dimensionality of $\mathbb{R}^{K \times N}$. However, the classic algorithm in Section 3.3 computes the SVD of matrix $\boldsymbol{S}$ with a dimensionality of $\mathbb{R}^{M \times N}$, which is computationally more intensive than the subspace detection algorithms since $K << \min(M, N)$.

4.2 The signal recovery algorithm

MMMF recovers the missing entries of the signal matrix $\boldsymbol{S}$. To compute the FPC-scores, a straightforward way is to center the recovered matrix and then apply SVD on the centered matrix directly. However, SVD is computationally expensive and memory intensive. To address this challenge, we develop an incremental SVD algorithm that computes the SVD of the centered signal matrix by adding one column at a time.

Many incremental SVD algorithms have been developed in the literature (Brand, 2002; Bunch, 1978; Balzano and Wright, 2013). However, these algorithms are designed for either full-rank or low-rank matrices. A matrix is full-rank if its rank equals the number of its rows or columns, whichever is smaller. That is, a matrix with m rows and n columns is full-rank if its rank $K = \min\{m, n\}$, or equivalently, $\sum_{k:\sigma_k \neq 0} 1 = \min\{m, n\}$, where σ_k is the singular

value of the matrix and $\sum_{k:\sigma_k \neq 0} 1$ represents the number of nonzero singular values. For a full rank matrix, Bunch (1978) develop an incremental SVD algorithm, which computes the SVD of the matrix by adding one column at a time. The singular vector matrices and singular value matrix are updated and their size grows as columns are added. Balzano and Wright (2013) point out that the algorithm in Bunch (1978) can be modified such that it works for low-rank matrices (i.e., matrices whose rank $K < \min\{m,n\}$, or equivalently, $\sum_{k:\sigma_k \neq 0} 1 < \min\{m,n\}$). This can be achieved by avoiding adding extra dimensions to the singular vector matrices and singular value matrix if the newly added column already lies in the space spanned by the singular vectors.

Unfortunately, the degradation signal matrix considered in this paper is neither low-rank nor full-rank but approximately low-rank. Approximately low-rank means that some of the singular values are very small and approximately but not exactly equal zeros. A degradation signal matrix is usually approximately low-rank because degradation signals are contaminated with noise. The small singular values that are approximately zeros represent the variation resulting from the signal noise. For matrices with approximately low-rank, the existing incremental SVD algorithms do not work. To address this challenge, we propose an incremental SVD algorithm, which is summarized in Algorithm 2.

Algorithm 2: Incremental SVD for matrices with approximately low-rank

---

Input: Matrix $\boldsymbol{S}_{[M\times N]}$, $\varepsilon = 1\times e^{-6}$

Output: SVD of $\boldsymbol{S}$, i.e., $\boldsymbol{S}_{[M\times N]} = \boldsymbol{U}_{[M\times M]}\boldsymbol{\Sigma}_{[M\times N]}\boldsymbol{V}^{\cdot}_{[N\times N]}$

(1) **Initialization:** $d = 1,\ \boldsymbol{U}_{[M\times d]} := \frac{\tilde{\boldsymbol{s}}_1}{\|\tilde{\boldsymbol{s}}_1\|},\ \boldsymbol{\Sigma}_{[d\times d]} := \|\tilde{\boldsymbol{s}}_1\|,\ \boldsymbol{V}_{[d\times d]} := 1$

(2) for *i = 2 to N* **do**

(3) $\boldsymbol{w}_{[d\times 1]} := \boldsymbol{U}^{\cdot}_{[M\times d]}\tilde{\boldsymbol{s}}_{i[M\times 1]}$ % weight vector

(4) $\boldsymbol{p}_{[M\times 1]} := \boldsymbol{U}_{[M\times d]}\boldsymbol{w}_{[d\times 1]}$

(5) $\boldsymbol{e}_{[M\times 1]} := \tilde{\boldsymbol{s}}_{i[M\times 1]} - \boldsymbol{p}_{[M\times 1]}$ % residual vector

(6) $\begin{bmatrix} \boldsymbol{\Sigma}_{[d\times d]} & \boldsymbol{w}_{[d\times 1]} \\ \boldsymbol{0}_{[1\times d]} & \|\boldsymbol{e}\|_{[1\times 1]} \end{bmatrix} = \boldsymbol{U}_{[(d+1)\times(d+1)]}\boldsymbol{\Sigma}_{[(d+1)\times(d+1)]}\boldsymbol{V}^{\cdot}_{[(d+1)\times(d+1)]}$

(7) $\boldsymbol{U}_{[M\times(d+1)]} := \begin{bmatrix} \boldsymbol{U}_{[M\times d]} & \dfrac{\boldsymbol{e}_{[M\times 1]}}{\|\boldsymbol{e}\|_{[1\times 1]}} \end{bmatrix}\boldsymbol{U}_{[(d+1)\times(d+1)]}$

(8) $\boldsymbol{\Sigma}_{[(d+1)\times(d+1)]} := \boldsymbol{\Sigma}_{[(d+1)\times(d+1)]}$

(9) $\boldsymbol{V}_{[(d+1)\times(d+1)]} := \begin{bmatrix} \boldsymbol{V}_{[d\times d]} & \boldsymbol{0}_{[d\times 1]} \\ \boldsymbol{0}_{[1\times d]} & \boldsymbol{1}_{[1\times 1]} \end{bmatrix}\boldsymbol{V}_{[(d+1)\times(d+1)]}$

(10) **if** $\|\boldsymbol{e}\| < \varepsilon$ **then**

(11) $\boldsymbol{U}_{[M\times d]} := \boldsymbol{U}_{[M\times(d+1)]}(1:M, 1:d)$ %delete the last column

(12) $\boldsymbol{\Sigma}_{[d\times d]} := \boldsymbol{\Sigma}_{[(d+1)\times(d+1)]}(1:d, 1:d)$ %delete both the last row and last column

(13) $\boldsymbol{V}_{[(d+1)\times d]} := \boldsymbol{V}_{[(d+1)\times(d+1)]}(1:d+1, 1:d)$ %delete the last column

(14) **else**

(15) $d := d+1$

(16) **end**

(17) end

Algorithm 2 starts with the first column of the centered signal matrix $\boldsymbol{S}_{[M\times N]}$, where the subscript in [] is the dimensionality of the matrix. When a new column $\tilde{\boldsymbol{s}}_{i[M\times 1]}$ is added, we first expand it using the current singular vectors in

matrix $\boldsymbol{U}_{[M\times d]}$, and compute the weight vector $\boldsymbol{w}_{[d\times 1]}$ and the residual vector $\boldsymbol{e}_{[M\times 1]}$. Here, the residual vector $\boldsymbol{e}_{[M\times 1]}$ represents the information that cannot be explained using the current singular vectors in $\boldsymbol{U}_{[M\times d]}$. In other words, $\boldsymbol{e}_{[M\times 1]}$ is a vector that is perpendicular to the subspace spanned by the vectors in $\boldsymbol{U}_{[M\times d]}$. The newly added column $\tilde{\boldsymbol{s}}_{i[M\times 1]}$ and the current SVD results (i.e., $\boldsymbol{U}_{[M\times d]}\boldsymbol{\Sigma}_{[d\times d]}\boldsymbol{V}^{\cdot}_{[(d+1)\times d]}$) have the following relationship:

$$\left[\boldsymbol{U}_{[M\times d]}\boldsymbol{\Sigma}_{[d\times d]}\boldsymbol{V}^{\cdot}_{[(d+1)\times d]} \quad \tilde{\boldsymbol{s}}_{i[M\times 1]}\right]=\left[\boldsymbol{U}_{[M\times d]} \quad \frac{\boldsymbol{e}_{[M\times 1]}}{\|\boldsymbol{e}\|_{[1\times 1]}}\right]\begin{bmatrix}\boldsymbol{\Sigma}_{[d\times d]} & \boldsymbol{w}_{[d\times 1]}\\ \boldsymbol{0}_{[1\times d]} & \|\boldsymbol{e}\|_{[1\times 1]}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_{[d\times d]} & \boldsymbol{0}_{[d\times 1]}\\ \boldsymbol{0}_{[1\times d]} & \boldsymbol{1}_{[1\times 1]}\end{bmatrix}. \quad (7)$$

Here, $\|\cdot\|$ is the ℓ_2 norm. Therefore, we may apply SVD on the matrix $\begin{bmatrix}\boldsymbol{\Sigma}_{[d\times d]} & \boldsymbol{w}_{[d\times 1]}\\ \boldsymbol{0}_{[1\times d]} & \|\boldsymbol{e}\|_{[1\times 1]}\end{bmatrix}$ and use the decomposed results to update matrices $\boldsymbol{U}_{[M\times d]}, \boldsymbol{\Sigma}_{[d\times d]}$ and $\boldsymbol{V}^{\cdot}_{[(d+1)\times d]}$ (rows (6)–(9) in Algorithm 2). Comparing to the algorithms in Bunch (1978) as well as Balzano and Wright (2013), the novelty of our algorithm is that we use the norm of the residuals to determine whether an extra dimension should be added to the subspace or not. To be specific, we check the norm of the residuals $\boldsymbol{e}_{[M\times 1]}$. If the norm is small enough (i.e., $\|\boldsymbol{e}\|<\varepsilon$), the newly added vector lies almost in the space spanned by the vectors in $\boldsymbol{U}_{[M\times d]}$, and thus there is no need to add an extra dimension to the dimension of the space (Steps (11)–(13)). Algorithm 2 is computationally efficient and memory economic because it involves applying SVD on matrices with a dimensionality of $(d+1)\times(d+1)$, where $(d+1)<N$ is usually a small number. Therefore, Algorithm 2 can be used to compute the FPC-scores of a large-size signal matrix.

5 Numerical study

In this section, we validate the effectiveness of the proposed prognostics methodology using simulated datasets. We compare the performance of our method to some benchmarks in terms of computational time and the accuracy of predicting the RULs at various levels of signal-noise-ratio, data incompleteness, and sampling schemes.

5.1 Data generation

In this simulation study, we consider 500 identical systems, each of which is monitored by 100 sensors. We begin by simulating the underlying degradation signal of system *i* using the following functional form: $s_i(t) = -c_i / \ln(t)$, for $i = 1, \ldots, 500$; where $c_i \sim N(1, 0.25^2)$ and $0 \le t < 1$. We compute the TTF as the first time that $s_i(t)$ crosses the soft failure threshold *D*, where *D* = 2. In addition, we add an i.i.d. noise to the logarithmic TTF of each system–that is– the logarithmic TTF of system *i* is $\ln(\tilde{y}_i) = -c_i / D + \varepsilon_i$, where $\varepsilon_i \sim N(0, 0.025^2)$. Since the logarithmic TTF (i.e., $\ln(\tilde{y}_i)$) is a linear function of c_i with a normally distributed noise term, LLS regression with a lognormal distribution is the most suitable model to capture the relationship between TTFs and the fused features. Next, we generate the degradation signals from the 100 sensors. The degradation signal from the *p*th sensor of system *i* is generated using the following functional form: $s_{i,p}(T_i) = -c_i / \ln(T_i) + \grave{o}_{i,p}(T_i)$, where $p = 1, \ldots, 100, \grave{o}_{i,p}(T_i) \sim N(0, \sigma_p^2)$ and the discrete observation time points T_i are evenly spaced between 0 and $\tilde{y}_i$ (i.e., the TTF of system *i*) with an incremental 0.001. Here, $\sigma_p = \dfrac{\sum_i \sum_{T_i} s_{i,p}(T_i)}{500 \times \delta}$ and δ is the signal-noise-ratio (SNR).

5.2 Benchmarks and validation settings

We compare the performance of our prognostics methodology with two feature extraction approaches, *Subspace detection* and *Signal recovery*, with three benchmarks. The first baseline model, which we refer to as *Signal recovery (SVD)*, is similar to our proposed *Signal recovery* method except that regular SVD is applied to the recovered signals to extract FPC-scores. For our methods and the first benchmark, we first apply the polar transformation to the generated degradation signals. Next, we use the GROUSE algorithm in Balzano et al. (2010) to solve the optimization problem (6). Finally, the three methods are respectively used to extract FPC-scores for prognostics. Note that when degradation signals are highly incomplete in the time domain, the polar transformation method may result in a signal matrix $\boldsymbol{S}$ with a large

number of columns and also a high level of incompleteness. This may negatively affect the performance of the GROUSE algorithm (Balzano et al., 2010) and thus our prognostics methods. To address this challenge, we round the angular values from polar transformation (θ in Equation (5)) to control the size of the signal matrix and also the level of incompleteness. To be specific, we divide the angular domain $[0, \pi/2]$ into $Q+1$ equal parts, which gives us the following set of tick values $\{0, \frac{\pi}{2Q}, \frac{2\pi}{2Q}, \frac{3\pi}{2Q}, \ldots, \frac{\pi}{2}\}$. Then we round each angular value to its nearest tick value. By doing so, the column size of the signal matrix after polar coordinate transformation is set to $Q+1$. In this section and Section 6, Q is set to 500.

The second baseline model, which we designate as *Kernel smoother*, is an extension of the prognostics model proposed by Fang et al. (2017a). In Fang et al. (2017a), the authors use Hierarchical FPCA to fuse multi-stream degradation signals, and the fused features are then regressed against TTFs via LLS regression in a similar manner to the regression framework used in this paper. Hierarchical FPCA works by first applying FPCA to the degradation signals from each sensor (i.e., degradation signasl are grouped by each sensor) individually to extract their FPC-scores. Next, the FPC-scores from different sensors are concatenated, and regular PCA is applied to the concatenated FPC-scores to extract fused features. The numerical studies in Fang et al. (2017a) indicate that Hierarchical FPCA performs almost the same as MFPCA (which is used in this paper) on fusing multi-stream signals. The prognostics model proposed in Fang et al. (2017a) works only for complete data. Here, we revise it by employing a kernel smoother (Yao et al., 2005) so that it works for incomplete data. To be specific, we first apply kernel smoother to the pooled data from the same sensor of all systems to estimate its signal mean and covariance matrix. The covariance matrix is then decomposed using eigen decomposition and eigen vectors are provided. Using the eigen vectors and the incomplete observations, the FPC-scores can be computed via an algorithm called principal analysis by conditional expectation (PACE) (Yao et al., 2005) (please refer to Yao et al. (2005) more details about the algorithm). Next, the FPC-scores from all the sensors are

concatenated and regular PCA is applied on the concatenated vector to extract the fused features (i.e., ζ_i in Equation (5)). Finally, the prognostics model is built by regressing the fused features against TTFs using lognormal regression.

The third benchmark, designated *B Spline*, uses B spline to impute the missing observations. Specifically, we apply B spline to the incomplete observations from each sensor of each individual system to fit a degradation trajectory. The order of spline basis and the number of the knots are selected using generalized cross-validation (GCV). The imputed degradation signals are then concatenated and MFPCA is used to extract fused features, which are then regressed against TTFs using lognormal regression.

We evaluate the performance of our proposed methods and the benchmarking models using data generated under two SNRs: $\delta = 1$ (high noise) and $\delta = 2$ (low noise). For each SNR, we consider the following four scenarios: (i) Complete data, (ii) Random sampling, (iii) Nonuniform sampling, and (iv) Imbalanced sampling. In the first scenario, all the observations are used for model training and RUL prediction. In the second scenario, we randomly sample the observations from each degradation signal at four levels of data incompleteness: 20%, 40%, 60%, and 80%. For example, 20% means that we randomly select 20% of the observations from each signal and use the selected observations to construct a dataset for model training and RUL prediction. The third scenario also samples data at four levels of incompleteness (i.e., 20%, 40%, 60%, and 80%). However, it randomly samples a sequence of continuous observations from each degradation signal. Taking the 20% level as an example, a fragmented signal piece, the length of which is 20% of the total length of the degradation signal is randomly selected. In the last scenario, we consider four imbalanced sampling combinations: “10% + 90%,” “20% + 80%,” “30% + 70%” and “40% + 60%.” Here, “10% + 90%” means that we randomly select 50 sensors and randomly choose 10% degradation observations of these sensors, and randomly choose 90% observations of the remaining 50 sensors to construct a dataset.

For each scenario above, we randomly choose 400 systems for model training and the remaining 100 systems are used for model testing. The number of FPC-scores is chosen by setting FVE at 0.95 (see Section 3.3 for details). We repeat the whole simulation process for 10 times and compute the prediction errors using the following equation:

$$\text{Prediction Error} = \frac{|\text{Estimated TTF} - \text{True TTF}|}{\text{True TTF}}. \quad (8)$$

The simulation scenarios were performed using MATLAB 2016b in a 64-bit Unix system with the Xeon X5560 CPU @2.80 GHz processor and 32.0 GB RAM.

5.3 Results and analysis

5.3.1 Computational time

We first report the computational time of our algorithms and the baseline models when 80% of the observations from each sensor are available for model training in Table 2. Table 2 indicates that *Kernel smoother* is the computationally most intensive method, the computational time of which is more than 2 hours. This is reasonable since Kernel smoothing smooths/imputes each observation individually using a local regression, which implies that a large number of regression models need to be estimated. The second time consuming model is *Signal recovery (SVD)*, which is 480 seconds. The computational time for our model *Signal recovery (ISVD)* is 39 seconds, which is much smaller than *Signal recovery (SVD)*. This suggests that the incremental SVD algorithm developed in this paper is computationally efficient. We also observe that the computational time of the proposed method *Subspace detection* is 38 seconds, which is also computationally efficient. In addition, the computational time of the benchmark *B Spline* is 68 seconds, which is comparable to our methods and much faster than *Kernel smoother* and *Signal recovery (SVD)*.

5.3.2 High noise data

For the generated high noisy data, we report the prediction errors of our methods and the benchmarks for complete signals in Figure 3. Please notice that even if the degradation signals are complete in the time domain, they are incomplete after we transform them to the polar domain. This is because although degradation signals from different systems have been observed at the same time points in the time domain, they have different angular coordinates in the polar domain after transformation.

Figure 3 illustrates that our methods *Signal recovery (ISVD)* and *Subspace detection* achieve smaller prediction errors than the benchmarking models *Kernel smoother* and *B Spline*. We believe this is because our methods use all available observations–that is–observations from all sensors and all systems, for missing data recovery. Unlike our methods, the two benchmarks utilize only partial of the available data to impute the missing observations. For example, *Kernel smoother* imputes a missing observation by using its neighboring data; *B spline* uses a system's available data to recovery that system's missing observations. In addition, Figure 3 indicates that our proposed method *Signal recovery (ISVD)* has similar prediction accuracies to *Signal recovery (SVD)*. This implies that the incremental SVD algorithm proposed in this paper has similar performance to regular SVD.

Figure 4 shows the prediction errors for incomplete signals that are randomly selected at four levels of data incompleteness. Figure 5 presents the prediction errors for imbalanced incomplete signals. Both Figures 4 and 5 indicate that the prediction accuracy of our proposed methodologies consistently outperform the two benchmarks *Kernel smoother* and *B Spline* at all levels of data incompleteness and all imbalanced sampling combinations. For example, in Figure 4, when 80% of the observations are available, the median absolute prediction errors (and the interquartile range, i.e., IQR) for *Kernel smoother*, *B Spline*, *Signal recovery (ISVD)*, and *Subspace detection*, are $0.08(0.12)$, $0.58(0.08)$, $0.05(0.06)$, and 0.05(0.06), respectively. When 20% observations available, they respectively are $0.18(0.3)$, $0.15(0.3)$, $0.1(0.1)$, and 0.1(0.1). We again believe this is because that our methods use all available observations for missing data imputation, whereas the two benchmarks utilize

only partial of them. From Figures 4 and 5, we also observe that *Signal recovery (ISVD)* and *Signal recovery (SVD)* perform similarly in terms of prediction accuracy and precision. This again suggests our proposed incremental SVD algorithm performs similarly to classic SVD.

Figure 6 illustrates the prediction errors at four levels of data incompleteness using a nonuniform sampling scheme. Similar to Figures 3, 4, 5, we observe that our methods consistently perform better than the benchmarking models *Kernel smoother* and *B Spline*, which again indicates the superiority of our models. Moreover, we again observe that *Signal recovery (ISVD)* and *Signal recovery (SVD)* have similar prediction performance, which again validates the effectiveness of our proposed incremental SVD algorithm. In addition to the above observations, Figure 6 shows that *B spline* works worse than *Kernel smoother* when the data incompleteness level is 20%, 40%, and 60%. We believe this is because when data is highly incomplete and non-uniformly sampled, *B spline* is not able to accurately capture the trend of the degradation trajectory. Consequently, it cannot accurately impute the missing observations, and thus its predictability is compromised.

5.3.3 Low noise data

We report the prediction errors of our methods and the baseline models using the low noise data in Figures 7, 8, 9 and 10.

Figure 7 summarizes the prediction errors using complete signals. From Figure 7, we observe that the median absolute prediction error (and IQR) of *Kernel smoother*, *B Spline*, *Signal recovery (ISVD)*, and *Subspace detection* are $0.031(0.041)$, $0.029(0.038)$, $0.024(0.037)$, and 0.024(0.037), respectively. This suggests that the prediction accuracy and precision of our proposed methods are a little bit higher than the two benchmarks. Figure 8 illustrates the prediction errors for data with four levels of data incompleteness. It illustrates that our methods perform similarly to the two baseline models when 80%, 60%, and 40% observations are available. However, when the level of data incompleteness is 20%, our methods achieve higher prediction accuracy and precision than the benchmarking models (please refer to Figure 8(d) for

details). Similar results can also be observed in Figures 9 and 10. Therefore, we conclude that if degradation signals are with the low level of noise, the performance of the proposed methods is at least comparable to the baseline models when the percentage of available observations is high enough. When degradation signals are highly incomplete (for example, only 20% observations are available), our methods work better than the benchmarks. Moreover, Figure 10 indicates that *B spline* works worse than *Kernel smoother* when the data incompleteness levels are 20%, 40%, and 60%. Again, we believe this is because *B spline* cannot accurately capture the trend of a degradation trajectory if data is highly incomplete and non-uniformly sampled. As a result, it cannot accurately impute the missing observations, which compromises its prediction performance.

From Figures 7, 8, 9 and 10, we also observe that the prediction errors of *Signal Recovery (ISVD)* and *Signal Recovery (SVD)* are similar, which again implies that the incremental SVD developed in this paper performs similarly to regular SVD.

6 Case study

In this section, we use multi-sensor degradation data from aircraft turbofan engines provided by NASA (Saxena et al., 2008) to evaluate the performance of our model. The dataset is comprised of the following; (i) degradation signals from 100 training engines that were run to failure, (ii) degradation signals from an additional 100 test engines whose operation was prematurely terminated at random time points prior to their failure time, and (iii) the real TTFs of the 100 test engines. We refer the readers to Saxena et al. (2008), Fang et al. (2017a), and Fang et al. (2017b) for a more detailed introduction of the dataset.

In the dataset, each engine is monitored using 21 sensors, some of which are non-informative. Following the suggestion of Fang et al. (2017b), we choose 4 sensors (i.e., Total temperature at LPT outlet, Bypass Ratio, Bleed Enthalpy, and HPT coolant bleed) to build a prognostics model under lognormal distribution. Specifically, both the training and test datasets are first

transformed into the polar coordinate system (discussed in Section 3). The transformed training dataset is then used for training and the transformed test dataset is used to evaluate the TTF prediction performance. For each system in the test dataset, we predict its failure times at the time points pooled from its incomplete observations of the four sensors. For example, if the observation time points for a test system are {1, 2, 5}, {1, 2, 6}, {2, 5, 6}, and {1, 3, 5} for sensors 1, 2, 3 and 4, respectively, the pooled time points are {1, 2, 3, 4, 5, 6}. Therefore, the failure times of this system will be predicted at time points 1, 2, 3, 5 and 6. The prediction errors are computed using Equation (8). Similar to the simulation study in Section 5, we compare the performance of our method with three baseline models (*Kernel smoother*, *B Spline*, and *Signal recovery (SVD)*) under four scenarios: (i) Complete data, (ii) Random sampling, (iii) Nonuniform sampling, and (iv) Imbalanced sampling.

6.1 Computational time

We first compare the computational time of our methods and the benchmarking models. The computational time when 80% observations are available is shown in Table 3. Similar to the simulation study, we observe that *Kernel smoother* is the computationally most expensive method. The computational time of our proposed methods *Signal recovery (ISVD)* and *Subspace detection* are similar and comparable to *B spline*. In addition, *Signal recovery (ISVD)* is computationally efficient than *Signal recovery (SVD)*, which implies the incremental SVD developed in this paper can significantly speed up the computation process of regular SVD.

6.2 Prediction results and analysis

We report the prediction errors of our methods and the benchmarks using complete signals in Figure 11. We observe that the median of the absolute prediction errors (and IQR) for *kernel smoother*, *B Spline*, *Signal recovery (ISVD)*, and *Subspace detection* are respectively $0.85(0.7), 0.3(0.35), 0.15(0.2)$, and 0.15(0.2). This implies that the methodologies developed in this paper outperform the baseline models in terms of both prediction accuracy and

precision. We believe this is because that our proposed methods use the available observations from all the systems and all the sensors for data imputation, whereas the benchmarks only use partial of the available observations. Moreover, Figure 11 indicates that *Signal recovery (ISVD)* and *Signal recovery (SVD)* have similar prediction errors, which suggests that the incremental SVD algorithm developed in this paper performs similarly to regular SVD.

Figure 12 summaries the prediction errors for data randomly sampled at four incompleteness levels, and Figure 13 reports the prediction errors for data randomly sampled with four imbalance combinations. Both Figures 12 and 13 indicate that our proposed methods work better than the baseline models in terms of prediction accuracy and precision at all data incompleteness scenarios. Again, we believe this is because our methods use more observations than the benchmarking models for missing observation imputation. Furthermore, Figures 12 and 13 again confirm that our proposed incremental SVD works similarly to regular SVD since there is no significant difference between the prediction errors of *Signal recovery (ISVD)* and *Signal recovery (SVD)*.

The prediction errors for nonuniformly sampled data are reported in Figure 14, from which we observe similar results as Figures 11, 12, and 13. That is, our methods perform consistently better than *Kernel smoother* and *B Spline*, and *Signal recovery (ISVD)* works similar to *Signal recovery (SVD)*. This again confirms the superiority of our methods. One more thing we observe from Figure 14 is that *B Spline* achieves lower prediction accuracy and worse prediction precision than *Kernel smoother*. We believe this is reasonable since *B Spline* fits each degradation signal individually by using its available observations. Therefore, it cannot accurately capture the trend of the signal and thus cannot impute the missing observations well if the available observations are not uniformly sampled from the signal trajectory.

7 Conclusions

This paper developed a prognostics model that can fuse incomplete degradation signals from multiple sensors to predict a system's failure time in real-time. The prognostics model is based on functional LLS regression and MFPCA. MFPCA focuses on fusing the incomplete multi-stream degradation signals and providing fused features known as FPC-scores, which are then regressed against TTF using LLS regression. To address the challenge of estimating FPC-scores using incomplete signals, we proposed two feature extraction algorithms based on MMMF. The first algorithm–*Subspace detection*–uses a set of general basis vectors provided by MMMF to compute the singular vectors of the incomplete degradation signal matrix and then the FPC-scores of each system's incomplete degradation signals. The second algorithm–*Signal recovery*–proposes an incremental SVD algorithm to compute the singular vectors of the signal matrix recovered from MMMF and then the FPC-scores. The incremental SVD algorithm, which computes the SVD of an approximately low-rank matrix by adding one row/column at a time, is computationally efficient and memory economic.

A simulation study and a multi-sensor degradation data from aircraft turbofan engines were used to evaluate the performance of our proposed method and several baseline models. The results indicated that the methods developed in this paper worked consistently better than the benchmarks in terms of computational time, prediction accuracy and prediction precision, especially when data was highly incomplete. In addition, the results also suggested that the proposed incremental SVD algorithm performed similarly to regular SVD but possessed much higher computational efficiency. The model developed in this paper only focuses on time series-based incomplete degradation signals. The development of a prognostics methodology for high-order incomplete degradation signals (such as profiles and images) is an important topic for future research.

A Proof for Proposition 1

Since the columns of $\boldsymbol{Q}$ constitute a set of orthonormal basis vectors of $\boldsymbol{S}$, $\boldsymbol{Q}^{\cdot}\boldsymbol{Q} = \mathrm{I}$. We have the following:

$$\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \boldsymbol{S},$$
$$\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \boldsymbol{S}.$$

Recall that the weight matrix is defined as $\boldsymbol{W} = \boldsymbol{Q}^{\cdot}\ \boldsymbol{S}$. Since the centered weight matrix $\boldsymbol{W} = \boldsymbol{W} - \overline{\boldsymbol{W}}$ and $\overline{\boldsymbol{W}} = \boldsymbol{Q}^{\cdot}\ \overline{\boldsymbol{S}}$, we have the following:

$$\begin{aligned}\boldsymbol{S} &= \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \boldsymbol{S} \\ &= \boldsymbol{Q}\boldsymbol{W} \\ &= \boldsymbol{Q}(\boldsymbol{W} + \overline{\boldsymbol{W}}) \\ &= \boldsymbol{Q}\boldsymbol{W} + \boldsymbol{Q}\overline{\boldsymbol{W}} \\ &= \boldsymbol{Q}\boldsymbol{W} + \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \overline{\boldsymbol{S}}.\end{aligned}$$

As a result, we have

$$\begin{aligned}& \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \boldsymbol{S} = \boldsymbol{Q}\boldsymbol{W} + \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \overline{\boldsymbol{S}} \\ \Rightarrow\ & \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \boldsymbol{S} - \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \overline{\boldsymbol{S}} = \boldsymbol{Q}\boldsymbol{W} \\ \Rightarrow\ & \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ (\boldsymbol{S} - \overline{\boldsymbol{S}}) = \boldsymbol{Q}\boldsymbol{W} \\ \Rightarrow\ & \boldsymbol{Q}\boldsymbol{Q}^{\cdot}\ \boldsymbol{S} = \boldsymbol{Q}\boldsymbol{W} \\ \Rightarrow\ & \boldsymbol{S} = \boldsymbol{Q}\boldsymbol{W}.\end{aligned}$$

Since the SVD for the centered weight matrix is $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\cdot}$, we can conclude that $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\cdot} = \hat{\boldsymbol{U}}\hat{\Sigma}\hat{\boldsymbol{V}}^{\cdot}$, where $\hat{\boldsymbol{U}} = \boldsymbol{Q}\boldsymbol{U}, \hat{\Sigma} = \Sigma, \hat{\boldsymbol{V}} = \boldsymbol{V}$.

References

Balzano, L., Nowak, R., & Recht, B. (2010). Online identification and tracking of subspaces from highly incomplete information. *In Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on (pp. 704-711). IEEE.*

Balzano, L., & Wright, S. J. (2013, December). On GROUSE and incremental SVD. *In 2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) (pp. 1-4). IEEE.*

Brand, M. (2002, May). Incremental singular value decomposition of uncertain data with missing values. *In European Conference on Computer Vision* (pp. 707-720). Springer, Berlin, Heidelberg.

Bunch, J. R., and Nielsen, C. P. (1978). *Updating the singular value decomposition. Numerische Mathematik*, 31(2), 111-129.

Candès, E. J., & Plan, Y. (2010). *Matrix completion with noise. Proceedings of the IEEE*, 98(6), 925-936.

Cai, J. F., Candès, E. J., & Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4), 1956-1982.

Chehade, A., Bonk, S., & Liu, K. (2017). Sensory-based failure threshold estimation for remaining useful life prediction. *IEEE Transactions on Reliability*, 66(3), 939-949.

Chehade, A., Song, C., Liu, K., Saxena, A., & Zhang, X. (2018). A data-level fusion approach for degradation modeling and prognostic analysis under multiple failure modes. *Journal of Quality Technology*, 50(2), 150-165.

Dai, W., & Milenkovic, O. (2010, March). SET: An algorithm for consistent matrix completion. *In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on (pp. 3646-3649). IEEE.*

Doray, L. (1994). IBNR reserve under a loglinear location-scale regression model. *In Casualty Actuarial Society Forum Casualty Actuarial Society* **2**, 607–652.

Fang X., Gebraeel N., & Paynabar K. (2017). Scalable prognostic models for large-scale condition monitoring applications. *IISE Transactions*, 49(7), 698-710.

Fang, X., Paynabar, K., & Gebraeel, N. (2017). Multistream sensor fusion-based prognostics model for systems with single failure modes. *Reliability Engineering & System Safety*, 159, 322-331.

Fang, X., Zhou, R., & Gebraeel, N. (2015). An adaptive functional regression-based prognostic model for applications with missing data. *Reliability Engineering & System Safety*, 133, 266-274.

Gouriveau, R., & Zerhouni, N. (2012). Connexionist-systems-based long term prediction approaches for prognostics. *IEEE Transactions on Reliability*, 61(4), 909-920.

Karhunen, K. (1947), “Uber lineare Methoden in derWahrscheinlichkeitsrechnung”, *Annales Academiae Sciientiarum Fennicae, Series AI: Mathematica-Physica*, 37, 3-79.

Keshavan, R. H., Montanari, A., & Oh, S. (2010). Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6), 2980-2998.

Lee, K., & Bresler, Y. (2010). Admira: Atomic decomposition for minimum rank approximation. *IEEE Transactions on Information Theory*, 56(9), 4402-4416.

Liao, H., & Sun, J. (2011). Nonparametric and semi-parametric sensor recovery in multichannel condition monitoring systems. *IEEE Transactions on Automation Science and Engineering*, 8(4), 744-753.

Liu, K., Gebraeel, N. Z., & Shi, J. (2013). A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Transactions on Automation Science and Engineering*, 10(3), 652-664.

Liu, K., & Huang, S. (2014). Integration of data fusion methodology and degradation modeling process to improve prognostics. *IEEE Transactions on Automation Science and Engineering*, 13(1), 344-354.

Liu, K., Chehade, A., & Song, C. (2015). Optimize the signal quality of the composite health index via data fusion for degradation modeling and prognostic analysis. *IEEE Transactions on Automation Science and Engineering*, 14(3), 1504-1514.

Jain, P., Meka, R., & Dhillon, I. S. (2010). Guaranteed rank minimization via singular value projection. *In Advances in Neural Information Processing Systems* (pp. 937-945).

Ma, S., Goldfarb, D., & Chen, L. (2011). Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1), 321-353.

Mazumder, R., Hastie, T., & Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug), 2287-2322.

Moghaddass, R., & Zuo, M. J. (2014). An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process. *Reliability Engineering & System Safety*, 124, 92-104.

Ramsay, J. O. (2004). *Functional data analysis*. Encyclopedia of Statistical Sciences, 4.

Rennie, J. D., & Srebro, N. (2005, August). Fast maximum margin matrix factorization for collaborative prediction. *In Proceedings of the 22nd international conference on Machine learning (pp. 713-719). ACM.*

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008, October). Damage propagation modeling for aircraft engine run-to-failure simulation. *In Prognostics and Health Management, 2008. PHM 2008. International Conference on (pp. 1-9). IEEE.*

Song, C., Liu, K., & Zhang, X. (2017). Integration of data-level fusion model and kernel methods for degradation modeling and prognostic analysis. *IEEE Transactions on Reliability*, 67(2), 640-650.

Song, C., & Liu, K. (2018). Statistical degradation modeling and prognostics of multiple sensor signals via data fusion: A composite health index approach. *IISE Transactions*, 50(10), 853-867.

Song, C., Liu, K., & Zhang, X. (2019). A generic framework for multisensor degradation modeling based on supervised classification and failure surface. *IISE Transactions*, 51(11), 1288-1302.

Srebro, N., Rennie, J., & Jaakkola, T. S. (2005). Maximum-margin matrix factorization. *In Advances in neural information processing systems* (pp. 1329-1336).

Sun, J., Zuo, H., Yang, H., & Pecht, M. (2010, January). Study of ensemble learning-based fusion prognostics. *In Prognostics and Health Management Conference*, 2010. PHM'10. (pp. 1-7). IEEE.

Sun, J., Liao, H., & Upadhyaya, B. R. (2014). A robust functional-data-analysis method for data recovery in multichannel sensor systems. *IEEE transactions on cybernetics*, 44(8), 1420-1431.

Toh, K. C., & Yun, S. (2010). An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of optimization*, 6(615-640), 15.

Xu, J., Wang, Y., & Xu, L. (2014). PHM-oriented integrated fusion prognostics for aircraft engines based on sensor data. *IEEE Sensors Journal*, 14(4), 1124-1132.

Yan, H., Liu, K., Zhang, X., & Shi, J. (2016). Multiple sensor data fusion for degradation modeling and prognostics under multiple operational conditions. *IEEE Transactions on Reliability*, 65(3), 1416-1426.

Yao, F., Müller, H. G., & Wang, J. L. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470), 577-590.

Yao, F., Fu, Y., & Lee, T. C. (2011). Functional mixture regression. *Biostatistics*, 12(2), 341-353.

Zhou, R., Serban, N., & Gebraeel, N. (2011). Degradation modeling applied to residual lifetime prediction using functional data analysis. *The Annals of Applied Statistics*, 1586-1610.

Zhou, R., Gebraeel, N., & Serban, N. (2012). Degradation modeling and monitoring of truncated degradation signals. *IIE Transactions*, 44(9), 793-803.

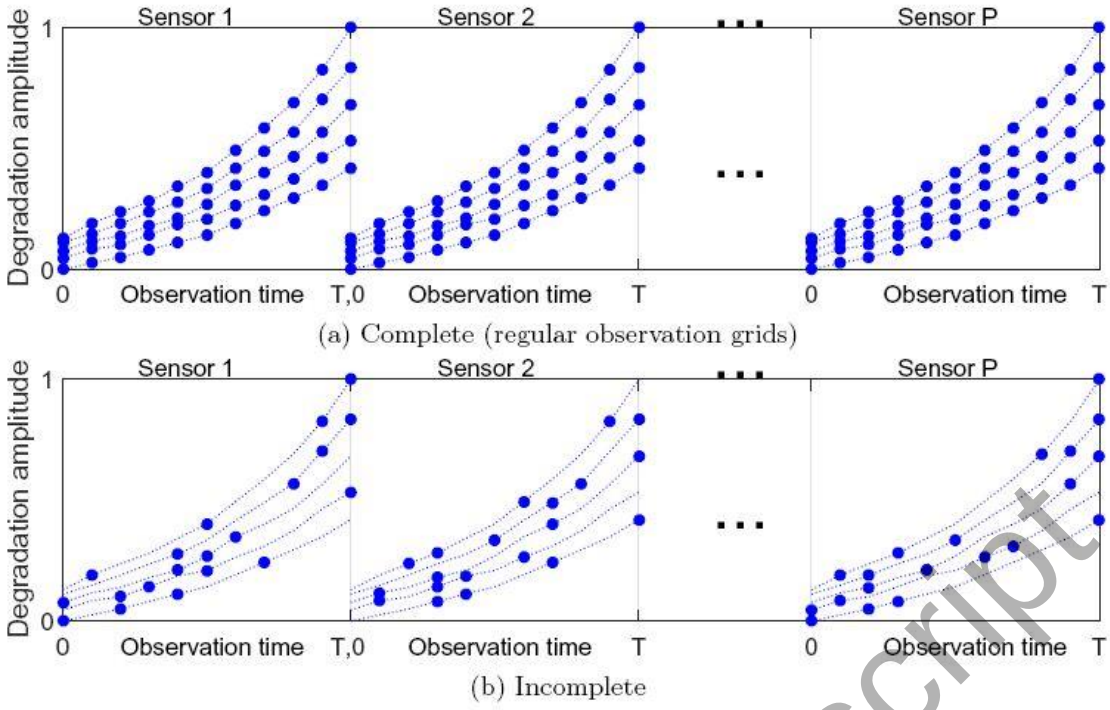
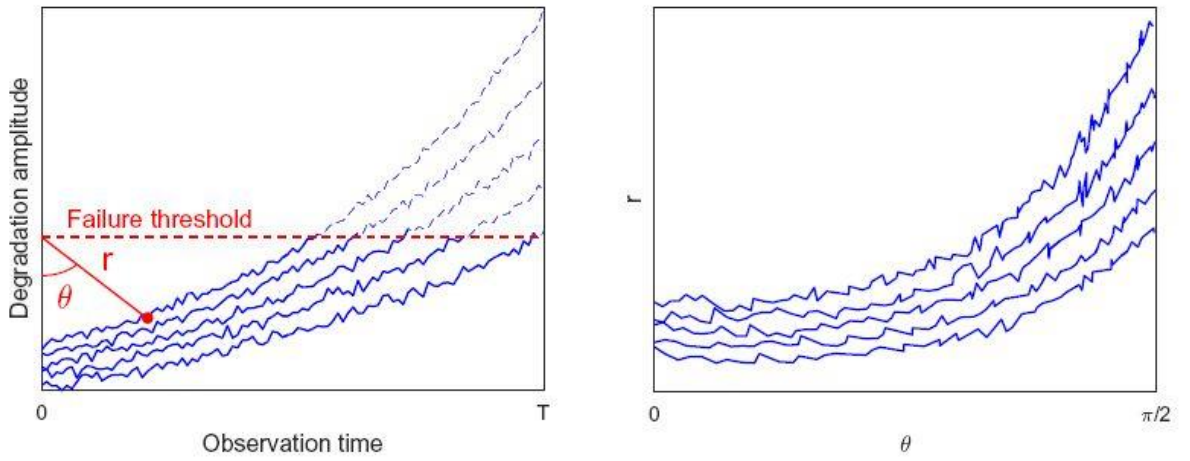


Fig. 1 An illustration of complete and incomplete multi-stream degradation signals.



(a) Truncation of degradation signals (b) Transformed signals in the polar coordinate

Fig. 2 The truncation of degradation signals and polar coordinate transformation.

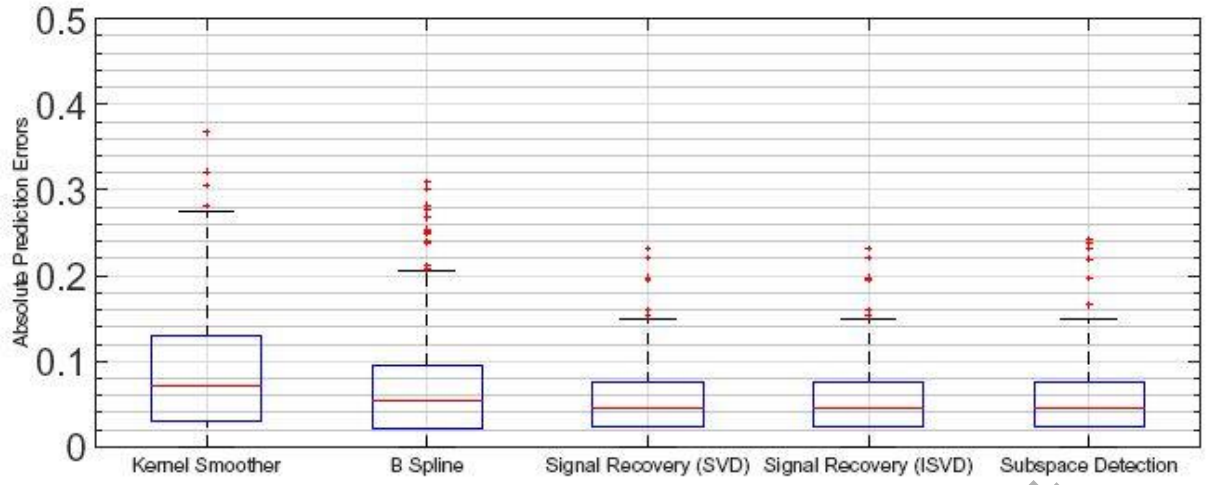


Fig. 3 Prediction errors for complete signals (High noise).

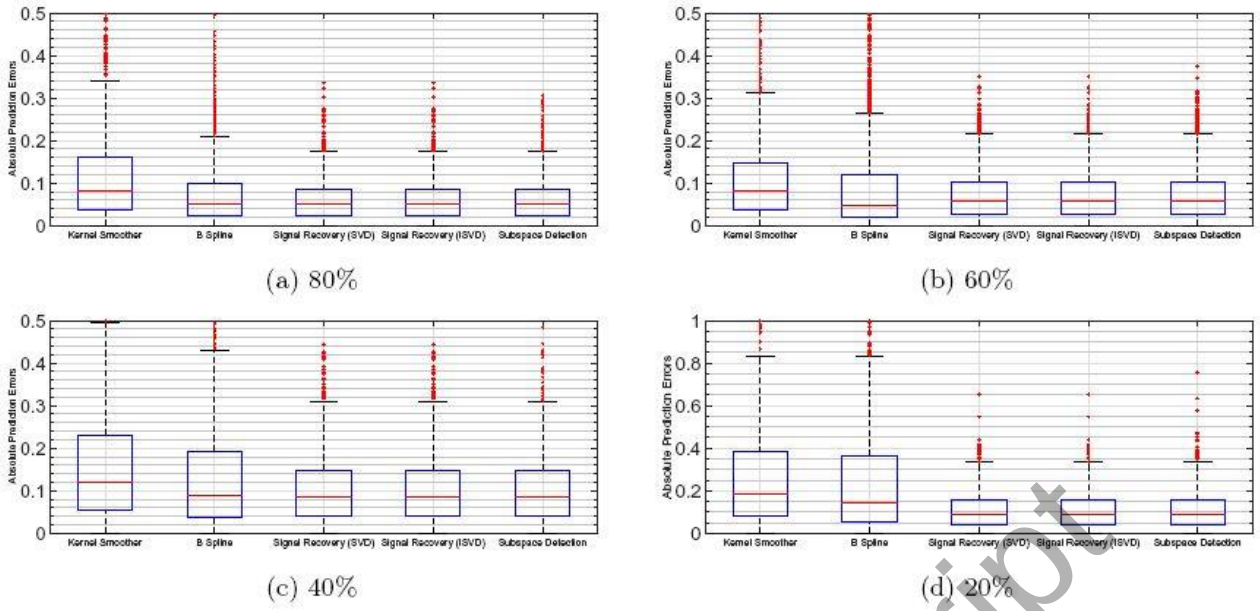
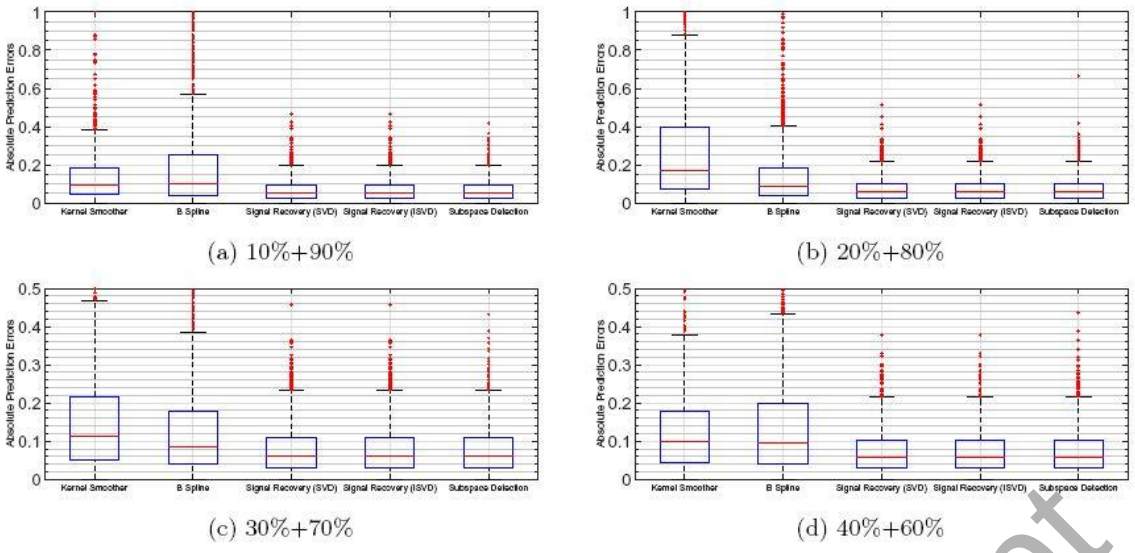


Fig. 4 Prediction errors for incomplete signals (High noise + Random sampling).



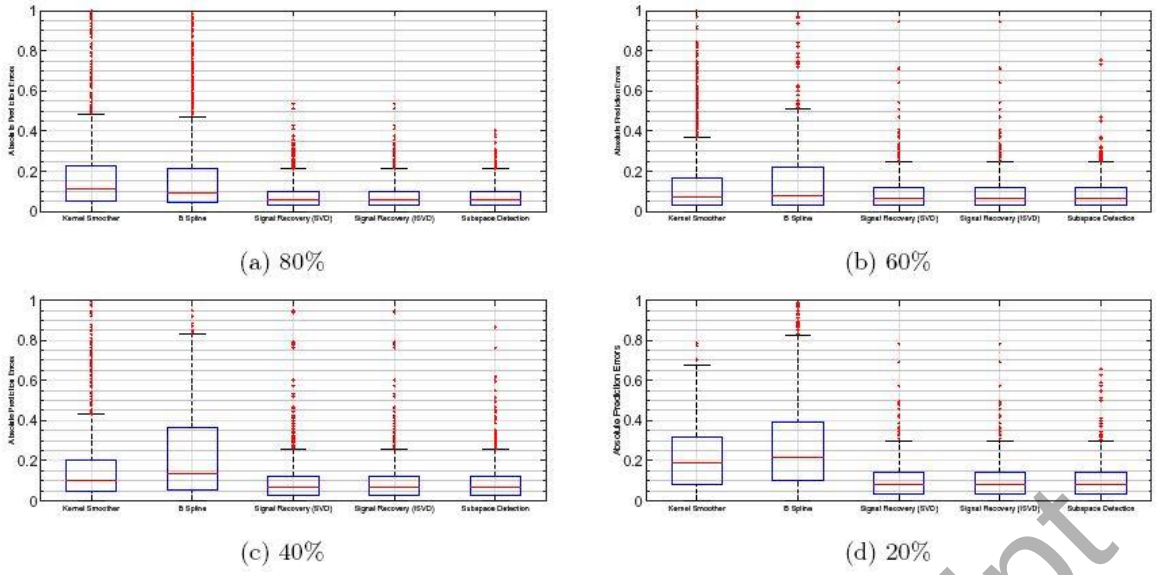
(a) 10%+90%

(b) 20%+80%

(c) 30%+70%

(d) 40%+60%

Fig. 5 Prediction errors for incomplete signals (High noise + Imbalanced sampling)



(a) 80% (b) 60%

(c) 40% (d) 20%

Fig. 6 Prediction errors for incomplete signals (High noise + Nonuniform sampling)

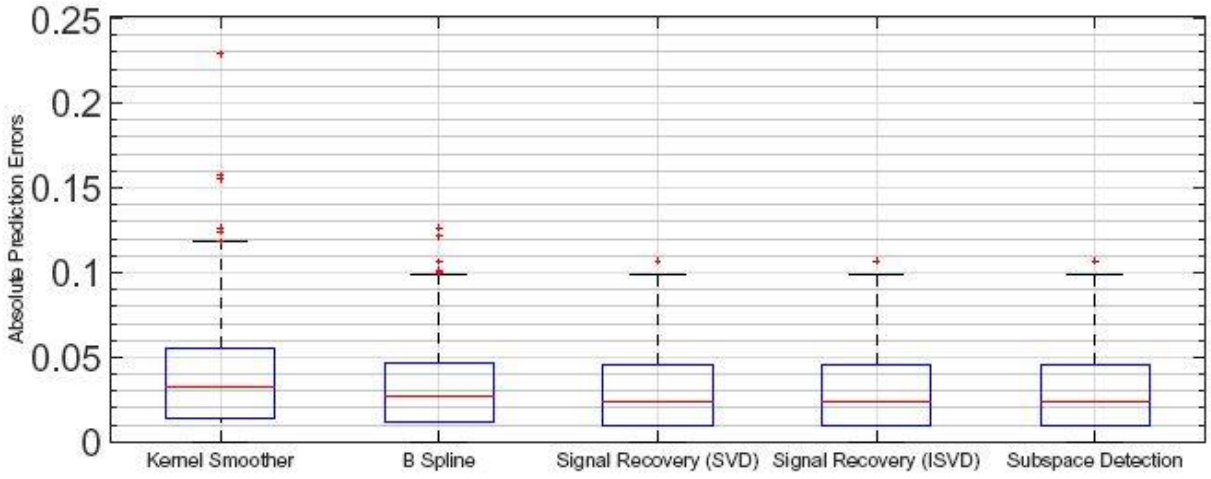


Fig. 7 Prediction errors for complete signals (Low noise).

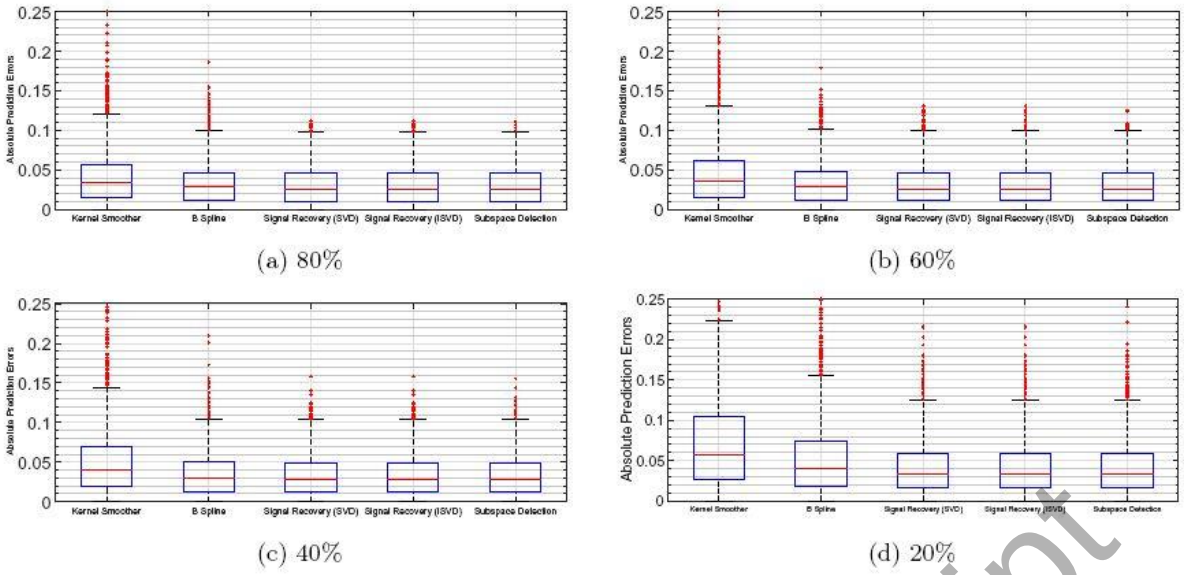
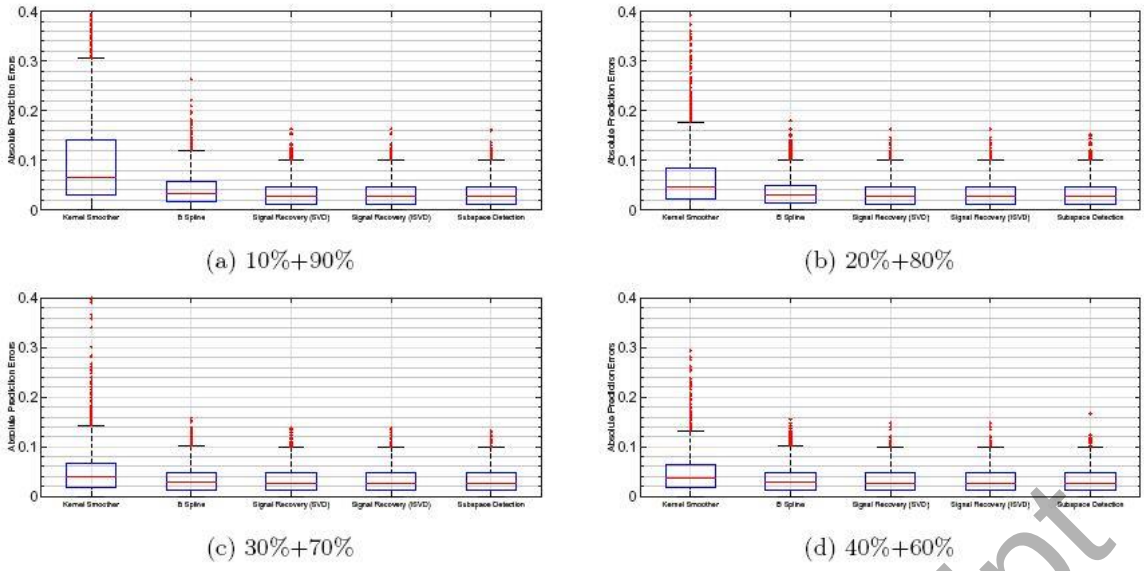


Fig. 8 Prediction errors for incomplete signals (Low noise + Random sampling).



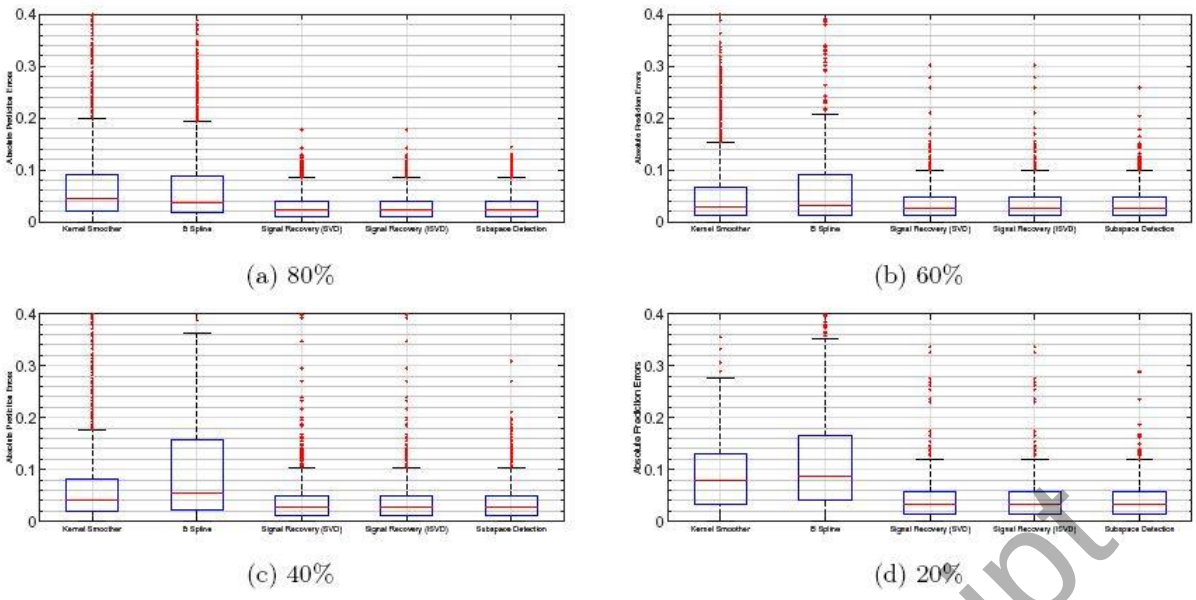
(a) 10%+90%

(b) 20%+80%

(c) 30%+70%

(d) 40%+60%

Fig. 9 Prediction errors for incomplete signals (Low noise + Imbalanced sampling)



(a) 80%

(b) 60%

(c) 40%

(d) 20%

Fig. 10 Prediction errors for incomplete signals (Low noise + Nonuniform sampling).

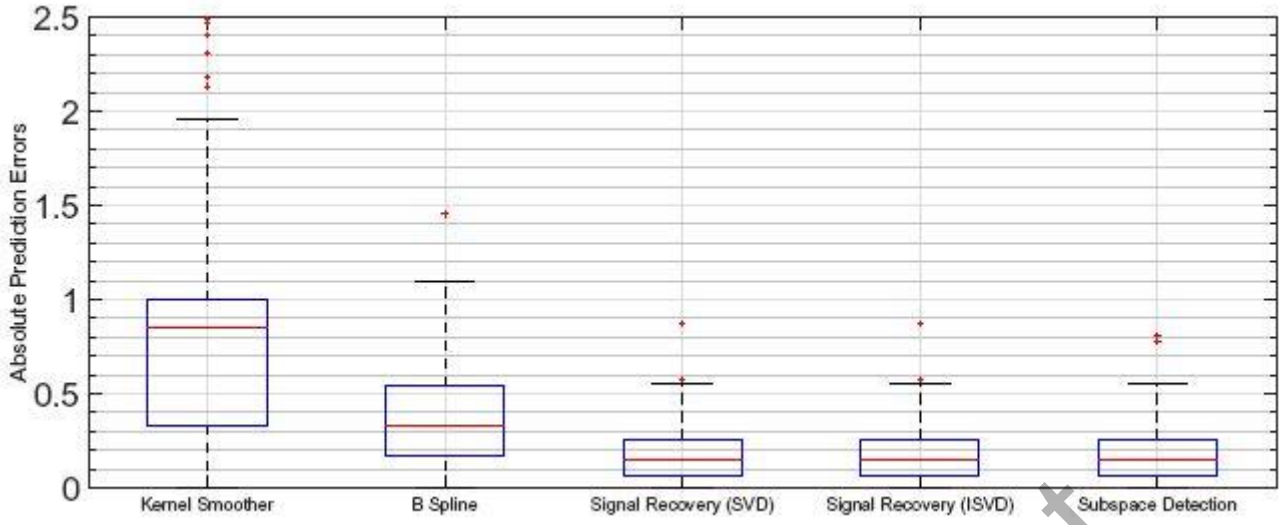


Fig. 11 Prediction errors for complete signals.

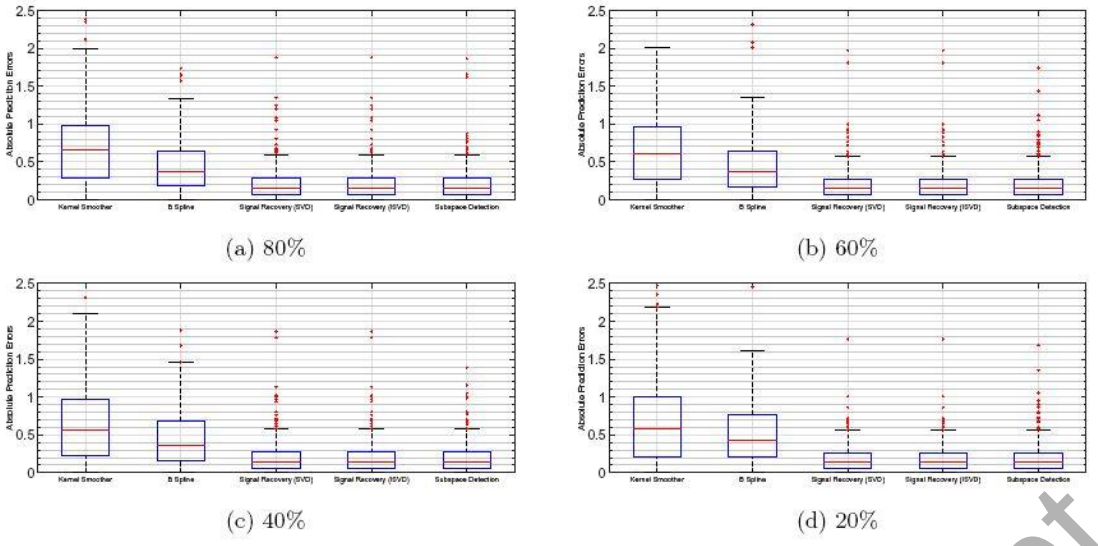


Fig. 12 Prediction errors for incomplete signals (Random sampling).

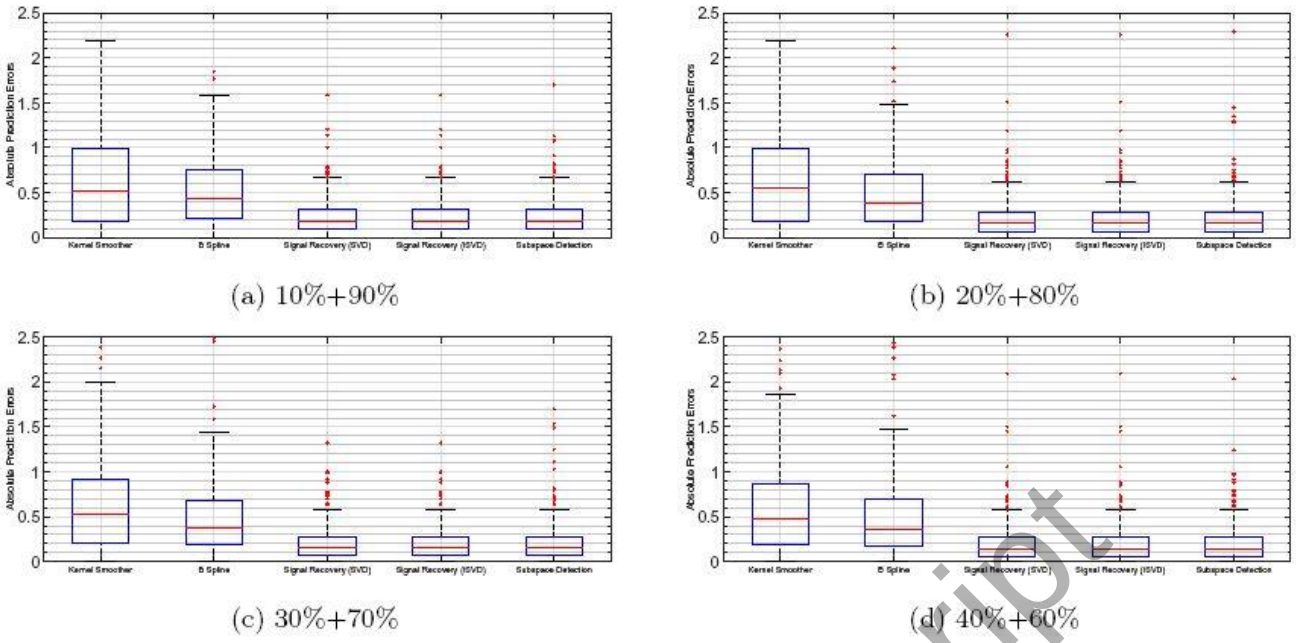


Fig. 13 Prediction errors for incomplete signals (Imbalanced sampling).

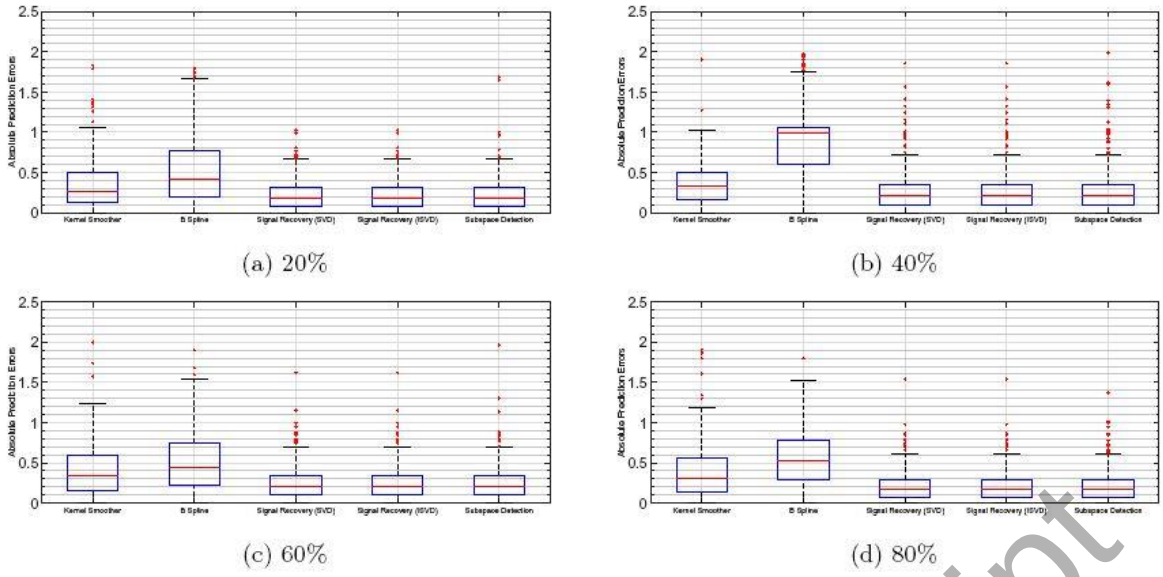


Fig. 14 Prediction errors for incomplete signals (Nonuniform sampling).

Table 1 An illustration of signal matrix $S^{\bullet}$.

		Sensor 1				Sensor 2				⋯	Sensor P			
		$\theta_{1,1}$	$\theta_{1,2}$	⋯	θ_{1,J_1}	$\theta_{2,1}$	$\theta_{2,2}$	⋯	θ_{2,J_2}		$\theta_{P,1}$	$\theta_{P,2}$	⋯	θ_{P,J_P}
System Index	1			⋯	×		×	⋯		⋯			⋯	×
	2	×		⋯		×	×	⋯		⋯	×		⋯	×
	⋮		×	⋯	×			⋯	×	⋯		×	⋯	
	N	×		⋯		×		⋯		⋯	×		⋯	

Table 2 Computational time when 80% observations are available (unit: second).

Kernel smoother	B Spline	Signal recovery (SVD)	Signal recovery (ISVD)	Subspace detection
> 7200	68	480	39	38

Table 3 Computational time when 80% observations are available (unit: second).

Kernel smoother	B Spline	Signal recovery (SVD)	Signal recovery (ISVD)	Subspace detection
38, 500	8	12	6	4