The International Journal of Advanced Manufacturing Technology (2019) 103:4659–4669
https://doi.org/10.1007/s00170-019-03912-x

ORIGINAL ARTICLE

Monitoring and diagnosis of multi-channel profile data based on uncorrelated multilinear discriminant analysis

Feng Ye[1] · Zhisheng Zhang[1] · Zhijie Xia[1] · Yifan Zhou[1] · Hui Zhang[1]

Received: 6 January 2019 / Accepted: 20 May 2019 / Published online: 3 June 2019

Abstract
Sensors are being widely used in many industrial practices and contain rich information which can be analyzed to detect system anomalies. The outputs of sensors are time-ordered data known as waveform signals, which are also called profiles. Many monitoring methods only focus on a single profile. However, multiple profiles are recorded by different sensor channels in many processes. It is crucial to study methods to analyze the multi-channel profiles. In this paper, uncorrelated multilinear discriminant analysis is suggested for fault detection and diagnosis. Then, the algorithm combined with tensor-to-tensor projection is proposed to make it get better performance in improving the accuracy of detection and reducing the fluctuation of the results in analyzing multi-channel profiles. The proposed method is applied directly to the multi-channel profiles. Discriminative and uncorrelated features are extracted, which are then fed into classifiers to identify different fault types. The effectiveness of the developed method is demonstrated by using both the simulation and a real-world case study. The real profiles in the case study are from a sensor fusion application in multiple forging operation processes, where multi-channel profiles are monitored to detect the faults of missing parts.

Keywords Fault detection and diagnosis · Feature extraction · Data classification · Sensor fusion · Linear discriminant analysis

1 Introduction

In complicated manufacturing processes, sensors are widely used to collect real-time process data. In many practical applications, the output of an automatic sensor and data capturing technology is represented by spatial- or time-ordered functional data known as waveform signals or profiles. There are many examples of profiles collected from the industrial processes, such as the force signals used to press seats into engine head during an assembly process [1] and the power signals in ultrasonic metal welding process [2]. These data are affected by the properties of the materials, process setups, etc. The profile data can provide rich information for monitoring product quality and fault diagnosis. One of the challenges is that many industrial processes contain some operations which produce small signals. Although these signals provide valuable information about the industrial process, their changes are hard to detect. The corresponding profile is called "weak signal." To avoid the complexity of profile data analysis, some simple statistical characteristics like the maximum or mean of the profile data are often used for online process monitoring. Although these methods are useful, they fail to utilize the rich information of data and cannot meet the demand in some industrial applications. For example, in the forging process, multiple dies work together to produce a full product. Missing parts from some dies will not only produce defective products but also damage the expensive dies.

Many researchers have studied the modeling and monitoring of profile data [3–5]. In the area of profile monitoring and fault detection, Paynabar et al. [6] applied a nonlinear parametric regression model to develop a system which was robust and insensitive to the variations of temperature in real-world production practice. Some methods combined with artificial networks (ANN) and machine learning were proposed to monitor profiles [7, 8]. A variable moving window kernel principal component analysis (VMWKPCA) algorithm was developed to monitor the nonlinear process in [9]. Lu et al. [10] proposed a nonlinear adaptive dictionary learning algorithm to achieve early fault detection of bearing elements. Lei

✉ Zhijie Xia
zhijie.xia@gmail.com

[1] School of Mechanical Engineering, Southeast University, No. 2 Southeast University Road, Nanjing 211189, China

et al. [11] added up all channel profiles and applied principal component analysis (PCA) to the aggregated tonnage profiles to extract features. For the purpose of feature extraction, a set of low-dimensional monitoring features were extracted from the high-dimensional profile data to detect tool wear [12–14]. An adaptive sensor fusion combined with signal processing techniques to extract features to estimate of tool wear was proposed in [15]. Paynabar et al. [16] combined multidimensional PCA with change-point models for the construction of monitoring statistics. Grasso et al. [17] combined PCA with empirical mode decomposition to monitor health condition in waterjet cutting. A robust incremental online feature extraction method based on PCA, RIPCA (Robust Incremental Principal Component Analysis), is proposed in [18].

Most literature analyzed individual profile (single signal). However, in many processes, multiple profiles are recorded by different sensor channels. For instance, as shown in Fig. 1, four sensors are installed on different uprights of a forging machine, and each sensor collects the tonnage forces exerted on dies. The outputs of these sensors are called multi-channel profiles. An individual sensor signal does not make full use of rich information of multi-channel profiles and only provides a partial state of the process. The analysis of multi-channel profiles presents challenges to researchers. One solution is to reshape the multi-channel profiles into a high-dimensional vector. However, this significantly increases the dimension and computational complexity. In addition, it breaks the original data structure and may lose some useful information. Hence, multilinear feature extraction methods need to be studied. Paynabar et al. [19] applied uncorrelated multilinear principal component analysis (UMPCA) [20] for profile monitoring and fault diagnosis by accounting for the interrelationship of different profile channels. Grasso et al. [21] applied multiway PCA to reduce the profile dimensionality in order to improve the efficiency of the profile analysis system and monitored multi-channel profiles with control charts. Pacella [22] applied two multilinear extensions of PCA to extract features in an emission control system.

Linear discriminant analysis (LDA) is a classical algorithm for feature extraction. However, regular LDA cannot be operated on multi-channel profiles directly because it can only be applied for vectors. One solution is to reshape the multi-channel profiles into a high-dimensional vector, which is known as Vectorized-LDA (VLDA). This approach, again, does not make full use of rich information of multi-channel profiles. An uncorrelated multilinear discriminant analysis (UMLDA) [23] was proposed for face recognition and image processing. UMLDA operates on multidimensional data directly and extracts uncorrelated discriminant features by tensor-to-vector projection. Compared to the UMPCA algorithm which is unsupervised, UMLDA is a supervised multilinear feature extractor which will consider class information when extracting features and could be more suitable for face recognition. Although there is some exploratory research on the applications of UMLDA to face recognition and image processing. Little research has been reported in the literature on using the UMLDA technique to analyze multi-channel profiles for fault detection and diagnosis in manufacturing systems.

In this paper, we first applied UMLDA to analyze multi-channel profiles, which is compared with previous methods to show its effectiveness in process monitoring and fault diagnosis. UMLDA is sometimes not stable depending on projection orders. We then proposed an improved UMLDA (I-UMLDA) method with tensor-to-tensor projection to reduce the effects of projection order to improve the accuracy of detection and reduce the fluctuation of the results. The proposed method is applied to analyze multi-channel profiles by obtaining a set of extracted features, which are used later to classify different working conditions and provide fault diagnosis results. The effectiveness of the improved method is demonstrated by using both the simulation and a real-world example of a multi-operation forging process.

The rest of the paper is organized as follows. Section 2 reviews the UMLDA algorithm for analysis and feature extraction on multi-channel profiles and proposes the I-UMLDA with tensor-to-tensor projection. In Sect. 3, the performance of the proposed I-UMLDA is evaluated and compared with UMLDA and other competing methods using simulations. The proposed method is applied to a case study on a multi-operation forging process in Sect. 4. Finally, this paper is concluded in Sect. 5.

2 Dimension reduction of multi-channel profiles by UMLDA and I-UMLDA

In this section, the notations relating to the tensor representation and some basic multilinear algebra are reviewed in Sect. 2.1. The implementations of I-UMLDA combined with tensor-to-tensor projection are presented in Sect. 2.2.

2.1 Basic definitions and tensor projection

An Nth-order tensor is denoted as $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$, and each I_n represents the n-mode of tensor $\mathcal{X}, n=1,\ldots,N$. The n-mode vector of $\mathcal{X}$ are the vectors whose dimension is I_n got from original tensor $\mathcal{X}$ by varying the index i_n while fixing all the other indices. In multilinear algebra, the n-mode ($n = 1, 2, \ldots, N$) product of the tensor $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ by the matrix $\mathrm{Z}\in\mathbb{R}^{D_n\times I_n}$ denoted by $\mathcal{X}\times_n\mathrm{Z}$ is a tensor with entries $(\mathcal{X}\times_n\mathrm{Z})(i_1,\ldots,i_{n-1},d_n,i_{n+1},\ldots,i_N)=\sum_{i_n}\mathcal{X}(i_1,\ldots,i_N)\cdot\mathrm{Z}(d_n,i_n)$.

The UMLDA and I-UMLDA proposed in this paper takes a tensor subspace approach of feature extraction. There are two common methods of tensor projection when projecting a

tensor into a subspace: the tensor-to-vector projection (TVP) and the tensor-to-tensor projection (TTP) [24]. The TVP consists of multiple elementary multilinear projection (EMPs). The EMPs consist of unit projection vector per mode. A tensor $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ can be projected to a scalar y by an EMP $\left\{v^{(1)^T}, v^{(2)^T}, \ldots, v^{(N)^T}\right\}$ as $y = \langle\mathcal{X},\mathcal{V}\rangle = \langle\mathcal{X}, v^{(1)}\circ v^{(2)}\circ, \ldots, \circ v^{(\ N)}\rangle = \mathcal{X}\times_1 v^{(1)^T}\times_2 v^{(2)^T}\ldots\times_N v^{(N)^T}$, $\|v^{(n)}\| = 1$ for $n = 1, 2, \ldots, N$, where $\|\cdot\|$ is the Euclidean norm for vectors, and $\mathcal{V} = v^{(1)}\circ v^{(2)}\circ, \ldots, \circ v^{(N)}$, and $\circ$ denotes the outer product. The TVP of a tensor $\mathcal{X}$ to a L-dimensional vector $\boldsymbol{y}\in\mathbb{R}^L$ consists of L EMPs $\left\{v_l^{(1)^T}, v_l^{(2)^T}, \ldots, v_l^{(N)^T}\right\}, l = 1, \ldots, L$. It can be denoted as $\mathcal{X}\times_{n=1}^{N}\left\{v_l^{(n)^T}, n = 1, \ldots, N\right\}_{l=1}^{L}$, in which the lth element of $\boldsymbol{y}$ is got by the lth EMP as $y(l) = \mathcal{X}\times_1 v_l^{(1)^T}\times_2 v_l^{(2)^T}\ldots\times_N v_l^{(N)^T}$.

In the multi-operation forging process, there is a set of multi-channel profiles which can be denoted as $\mathcal{X}\in\mathbb{R}^{C\times K\times M}$ with M samples (sample index $m = 1, \ldots, M$), and the C is the number of channels or sensors (channel or sensor index $c = 1, \ldots, C$), and the K is the number of data points measured in each channel (data index $k = 1, \ldots, K$).

2.2 I-UMLDA

LDA is a classical algorithm for dimensionality reduction and data classification, while UMLDA is its multilinear extension. In the multi-operation forging process, the accuracy of fault detection may fluctuate when the UMLDA algorithm was implemented on multi-channel profiles because of the projection order and different initialization. The original tensor space can be projected into another tensor subspace which captures most of the variations in the input high-dimensional data by TTP before using UMLDA, and the data transformed in the subspace is arranged according to the importance in each order. TTP is initialized by the full projection which can ease the effects of different order on accuracy to reduce the fluctuation and improve the accuracy of detection.

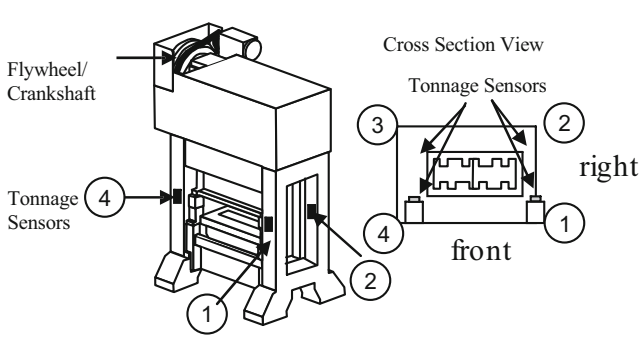

Fig. 1 Sensor distributions in the forging machine

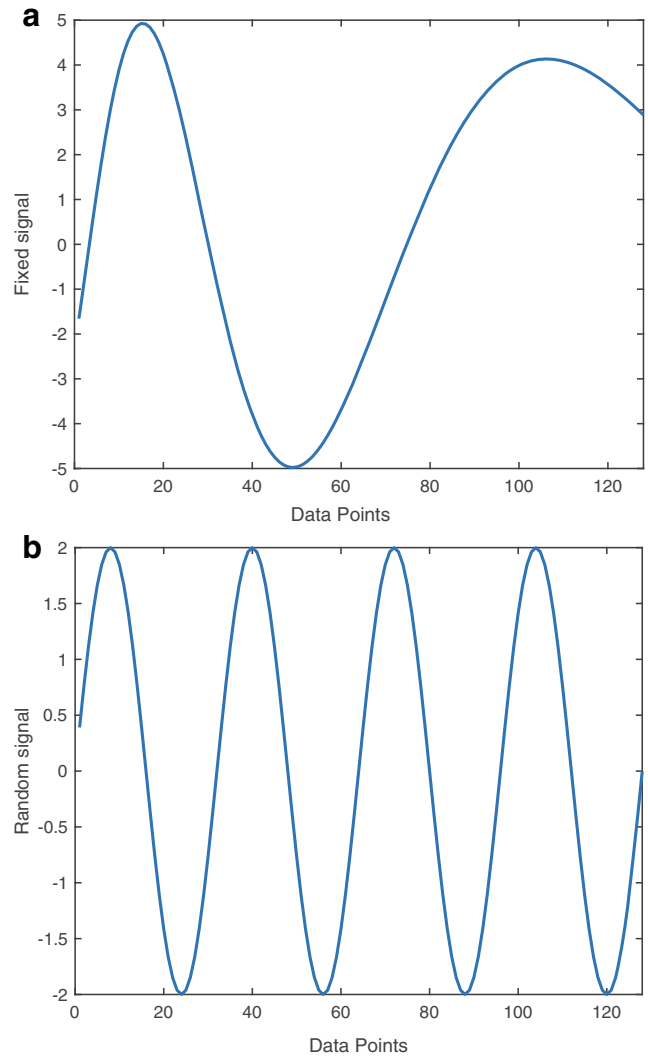

Fig. 2 Basic signals in generation. **a** Fixed-effect signals. **b** Random-effect signals

The TTP consists of a multilinear transformation or projection matrices $\left\{Z^{(n)}\in\mathbb{R}^{I_n\times D_n}, \mathrm{n} = 1, \ldots, N\right\}$ that projects the original tensor data $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ to another tensor data $\mathcal{Y}\in\mathbb{R}^{D_1\times D_2\times\ldots\times D_N}: \mathcal{Y} = \mathcal{X}\times_1 Z^{(1)^T}\times_2 Z^{(2)^T}\ldots\times_N Z^{(N)^T}$. The projection of an n-mode vector of $\mathcal{X}$ by $Z^{(n)^T}$ is computed as the inner product between the n-mode vector and the rows of $Z^{(n)^T}$. The projection matrices $\left\{Z^{(n)}\in\mathbb{R}^{I_n\times D_n}, \mathrm{n} = 1, \ldots, N\right\}$ need to be determined to maximize the total tensor scatter $\Psi_{\mathcal{Y}}$:

$$\left\{Z^{(n)}, \mathrm{n} = 1, 2\right\} = argmax\ \Psi_{\mathcal{Y}} \tag{1}$$

where $\Psi_{\mathcal{Y}} = \sum_{m=1}^{M}\|\mathcal{Y}_m - \overline{\mathcal{Y}}\|_F^2$, $\|\cdot\|_F$ is the Frobenius norm, and $\overline{\mathcal{Y}}$ is the mean tensor: $\overline{\mathcal{Y}} = \frac{1}{M}\sum_{m=1}^{M}\mathcal{Y}_m$. To solve $Z^{(n)}$, for all the other given projection $Z^{(1)}, \ldots, Z^{(n-1)}, Z^{(n+1)}, \ldots, Z^{(N)}$, the $Z^{(n)}$ consists of the D_n eigenvectors corresponding to the largest D_n eigenvalues of the matrix

$$\Phi^{(n)} = \sum_{m=1}^{M} \left(X_m^n - \overline{X}^n\right) \cdot Z_{\Phi^{(n)}} \cdot Z_{\Phi^{(n)}}^T \cdot \left(X_m^n - \overline{X}^n\right)^T \quad (2)$$

where

$$Z_{\Phi^{(n)}} = \left(Z^{(n+1)} \otimes Z^{(n+2)} \otimes \ldots \otimes Z^{(N)} \otimes Z^{(1)} Z^{(2)} \otimes \ldots Z^{(n-1)}\right) \quad (3)$$

$\otimes$ denotes the Kronecker product and the X_m^n is the n-mode unfolded matrix of the mth tensor $\mathcal{X}_m$. An iterative procedure can be utilized to solve this problem with initializations through full projection. The full projection refers to the multilinear projection with $D_n = I_n, \quad n = 1, \ldots, N$. In the full projection, $Z_{\Phi^{(n)}} \cdot Z_{\Phi^{(n)}}^T$ is an identity matrix which was proved from the pertinent lemma listed in the Appendix. Then, $\Phi^{(n)}$ in (2) turns to

$$\Phi^{(n)} = \sum_{m=1}^{M} \left(X_m^n - \overline{X}^n\right) \cdot \left(X_m^n - \overline{X}^n\right)^T \quad (4)$$

In this case, $\Phi^{(n)}$ is only decided by the original tensor $\mathcal{X}$ and independent of other projection matrices. Therefore, $\Phi^{(n)}$ is not affected by the order of projection which can reduce the effects of projection order in UMLDA. Then, UMLDA is applied to the new tensor data $\mathcal{Y}$ to determine a set of EMPs for feature extraction. The classical Fisher Discriminant Criterion (FDC) in LDA is defined as the scatter ratio for vector samples, which is trying to maximize the between-class distance and minimize the with-class distance simultaneously. The goal of UMLDA is trying to calculate L EMPs $\left\{v_l^{(n)^T}, n = 1, \ldots, N\right\}_{l=1}^{L}$ which can maximize the scatter ratio while the produced features are uncorrelated. Let $\{y_{m_l}, m = 1, \ldots, M\}$ denote the lth projected (scalar) features, where $y_{m_l} = \mathcal{Y}_m \times_1 v_l^{(1)^T} \times_2 v_l^{(2)^T}$ is the projection of the mth profile sample using the lth EMP. In addition, let h_l denote the set of coordinate vectors. The mth projected sample signal using the lth EMP is equivalent to the mth element of the lth coordinate vectors: $h_l(m) = y_{m_l}$. Accordingly, the between-class scatter $S_{B_l}^y$ and the within-class scatter $S_{W_l}^y$ are

$$S_{B_l}^y = \sum_{c=1}^{C} N_c \left(\overline{y}_{c_l} - \overline{y}_l\right)^2 \qquad S_{W_l}^y = \sum_{m=1}^{M} \left(y_{m_l} - \overline{y}_{c_{ml}}\right)^2 \quad (5)$$

where C is the number of profile classes, N_c is the number of samples for class c, c_m is the class label for the mth profile sample, $\overline{y}_l = \left(\frac{1}{M}\right) \sum_m y_{m_l} = 0$, and $\overline{y}_{c_l} = \left(\frac{1}{N_c}\right) \sum_{m, c_m = c} y_{m_l}$. The FDC for the lth scalar samples $F_l^y = \frac{S_{B_l}^y}{S_{W_l}^y}$. Therefore, the objective function for the lth EMP is

$$\left\{v_l^{(n)^T}, n = 1, 2\right\} = argmax\ F_l^y \quad (6)$$

$$\text{Subject to} \quad v_l^{(n)^T} v_l^{(n)} = 1 \quad \frac{h_l^T h_j}{\|h_l\| \|h_j\|} = \delta_{lj}, \quad l, j = 1, \ldots, L$$

$$\delta_{lj} = \begin{cases} 1, \text{if } l = j \\ 0, \text{otherwise} \end{cases} \quad (7)$$

3 Performance comparison between UMLDA and I-UMLDA using simulations

In this section, the performances of the UMLDA and I-UMLDA with TTP implemented on multi-channel profiles are compared using simulations for fault detection and diagnosis. Fixed and random-effect models with benchmark signals are utilized to generate random nonlinear profile data.

In the simulation of multi-operation forging process, four sensors are mounted on a forging machine to record 4-channel profiles during each cycle ($C = 4$), each sensor consists of 128 data points ($K = 128$). 200 samples ($M = 200$) are generated which can be denoted as tensor $\mathcal{X} \in \mathbb{R}^{C \times K \times M}$. It can be represented as follows:

$$y_m^c = \alpha_m^c \boldsymbol{x_1} + \beta_m^c \boldsymbol{x_2} + b_m^c + \varepsilon_m^c$$
$$m = 1, 2, \ldots, 200; \quad c = 1, 2, 3, 4 \quad (8)$$

where y_m^c is the $K \times 1$ vector of profile data for channel c, α_m^c is the fixed-effect coefficients for profile channel c and sample m, and $\alpha \sim MVN(\mu_\alpha, \Sigma_\alpha)$, $\mu_\alpha = [1, 2, 3, 1]$, $\Sigma_\alpha = diag(0.5, 0.5, 0.5, 0.5)$; $\boldsymbol{x_1}$ is the $K \times 1$ vector of fixed-effect signals and a benchmark signals "Doppler" with a range of K is selected, which is illustrated in Fig. 2a; β_m^c is the random-effect coefficients for profile channel c and sample m, and $\beta \sim MVN(\mu_\beta, \Sigma_\beta)$, $\mu_\beta = [1, 2, 3, 1]$, $\Sigma_\beta = diag(0.5, 0.5, 0.5, 0.5)$; $\boldsymbol{x_2}$ is the $K \times 1$ vector of random-effect signals and sine function with $T = \frac{K}{4}$ is selected: $\boldsymbol{x_2} = 2\sin\left(\frac{2\pi}{T} k\right)$ (Fig. 2b), where k is the data points index; $b \sim MVN(\mu_b, \Sigma_b)$, $\mu_b = [5, 10, 15, 12]$, $\Sigma_b = diag(0.5, 0.5, 0.5, 0.5)$ and b_m^c is the $K \times 1$ vector of signal difference between different channels; $\varepsilon_m^c \sim N(0, 0.5)$ is the random noise.

The simulations generated different out-of-control conditions from in-control working conditions. The out-of-control working conditions are related to some faults like part missing from dies or die weight not uniformly distributed etc. A set of 1000 profiles are generated and each class has 200 samples including one in-control condition and four out-of-control conditions (1)–(4). All situations are illustrated in Fig. 3.

More concretely, the other four out-of-control conditions are listed as follows:

1. The shift of the fixed effect signals $\boldsymbol{x_1}$: $\boldsymbol{x_1} \rightarrow \boldsymbol{x_1} + 2\sigma_{x_1} \boldsymbol{I}_K$, where the σ_{x_1} is the standard deviation of fixed effect signals $\boldsymbol{x_1}$, and $\boldsymbol{I}_K$ is a $K \times 1$ unit vector.

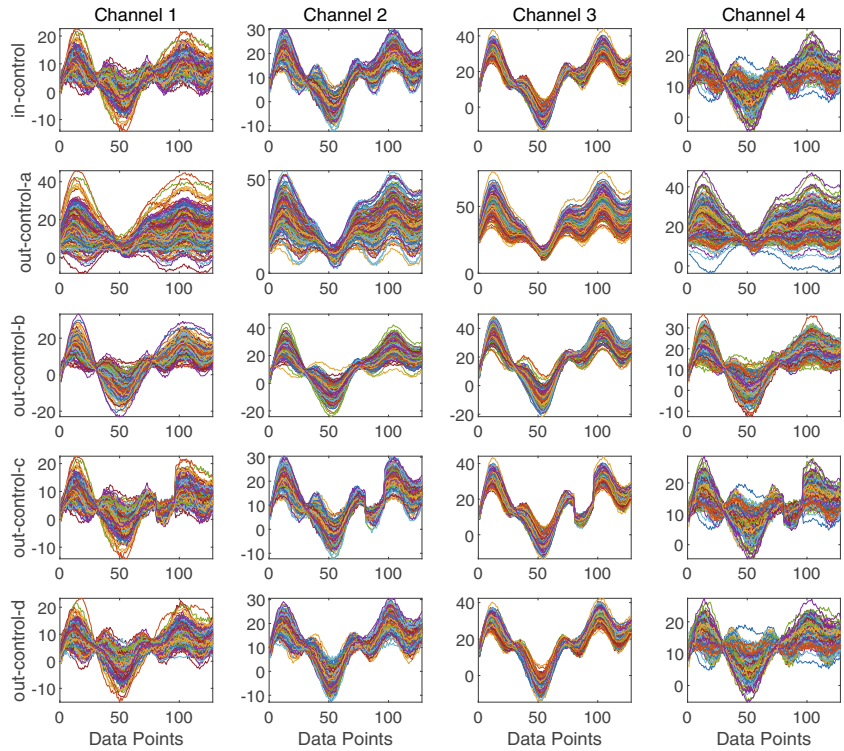

Fig. 3 Simulation dataset: 1000 samples in 5 classes for each channel

2. Variation of the fixed-effect coefficients α: $\mu_\alpha = \mu_\alpha + 2\sigma_\alpha$, $\sigma_\alpha = 2\sigma_\alpha$, where μ_α and σ_α are the mean and standard deviation of the fixed effect coefficients α.
3. Weak of part range signals $\boldsymbol{x}_1$: $\boldsymbol{x}_1 \rightarrow \begin{cases} \boldsymbol{x}_1 & 1 \le k \le \frac{5}{8}K \text{ and } \frac{6}{8}K < k \le K \\ \boldsymbol{x}_1 - \sigma_{x_1} \boldsymbol{I}_{\frac{K}{8}} & \frac{5}{8}K < k \le \frac{6}{8}K \end{cases}$ where σ_{x_1} is the standard deviation of fixed-effect signals $\boldsymbol{x}_1$, and $\boldsymbol{I}_{\frac{K}{8}}$ is a $\frac{K}{8} \times 1$ unit vector.
4. Phase differences of the random-effect signals: sine function $\boldsymbol{x}_2 = 2sin\left(\frac{2\pi}{T}k + \frac{\pi}{6}\right)$

3.1 Methods in comparison

The procedure of multi-channel profile monitoring and fault detection is shown in Fig. 4. Prior to analyzing the multi-channel profiles, each profile sample is denoised using a wavelet-based soft-thresholding method which was also used before applying UMPCA-based method in [19]. The soft-thresholding approach is widely used in signal denoising and is applied with the following thresholding rule:

$$\eta(\mathbf{b}) = \text{sign}(\mathbf{b})\ (|\mathbf{b}| - t)_+ \tag{9}$$

where the $\eta(\cdot)$ is the soft-thresholding function, sign $(\cdot)$ is the sign function, $\mathbf{b}$ is the wavelet coefficients, and t is the threshold [25]. The multi-channel profile can be represented as a tensor object. The feature extractor, e.g., UMLDA, will transform the tensor data into new features, which will be fed into some classifiers like nearest neighbor classifier (NNC) for classification. Finally, the performance will be assessed by correct classification rates.

The performance of improved algorithms against different other algorithms is compared. VPCA is referred to as vectorized-principal component analysis (PCA) [26], which reshapes the tensor data into a vector first and then applies the regular PCA. Unlike VPCA, MPCA and UMPCA deal

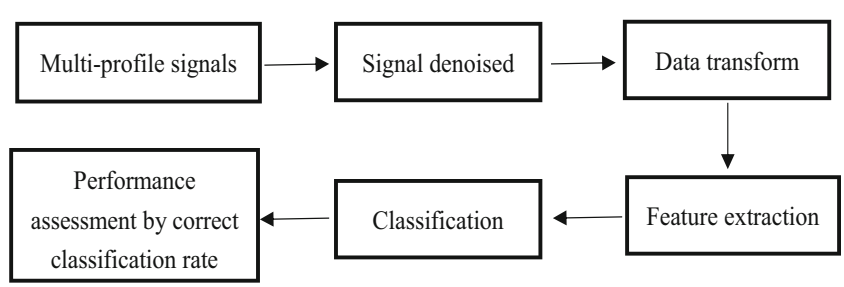

Fig. 4 Framework of profile monitoring and fault detection

Table 1 Correct classification rate (CRR) (mean ± Std %)

Number	VPCA	MPCA	UMPCA	UMLDA	R-UMLDA	R-UMLDA-A	I-UMLDA
1	27.8 ± 4.42	27.5 ± 4.29	27.4 ± 4.44	72.7 ± 5.55	76.7 ± 6.00	91.5 ± 1.14	84.6 ± 5.88
2	28.8 ± 4.66	28.9 ± 4.35	33.7 ± 8.84	75.6 ± 7.99	80.1 ± 7.64	91.5 ± 1.13	87.1 ± 5.83
3	29.6 ± 3.57	29.6 ± 3.41	45.6 ± 8.44	80.0 ± 8.72	83.7 ± 7.16	92.4 ± 1.38	87.4 ± 5.68
4	31.1 ± 2.04	30.9 ± 1.86	52.2 ± 4.83	78.3 ± 8.60	84.8 ± 7.60	91.6 ± 1.11	87.9 ± 4.67
5	38.3 ± 4.97	35.0 ± 3.94	–	78.9 ± 9.61	85.1 ± 6.99	92.1 ± 1.19	88.7 ± 3.21
6	44.0 ± 4.66	37.8 ± 3.47	–	80.8 ± 9.65	84.6 ± 7.14	91.7 ± 1.00	89.3 ± 3.74
7	48.8 ± 4.00	41.2 ± 3.83	–	81.5 ± 8.00	84.2 ± 7.31	92.1 ± 1.39	90.6 ± 2.97
8	52.2 ± 3.05	44.0 ± 3.02	–	80.6 ± 9.08	85.3 ± 7.14	91.9 ± 1.39	91.8 ± 2.27
9	56.2 ± 2.66	56.1 ± 2.90	–	81.2 ± 8.15	84.5 ± 7.57	91.8 ± 1.24	91.8 ± 2.15
10	57.9 ± 2.11	58.4 ± 2.34	–	81.0 ± 8.19	85.0 ± 7.05	91.9 ± 1.67	91.7 ± 1.56
20	61.8 ± 1.62	61.7 ± 2.19	–	87.0 ± 4.66	85.2 ± 6.58	91.7 ± 1.24	92.2 ± 1.25
30	63.4 ± 1.74	64.0 ± 2.07	–	85.8 ± 5.17	87.3 ± 4.92	91.8 ± 1.06	92.1 ± 0.95

with multidimensional (tensor) data rather than vectors. Both MPCA and UMPCA try to find projections which capture most of the variations in the input high-dimensional data. UMLDA and its developed methods are supervised multilinear methods, which take the classification labels into considerations. Both the MPCA and UMPCA are unsupervised methods. In some applications, the dimensionality of the profile is high when the number of samples is limited. It tends to minimize the within-class scatter towards zero so that F_l^y can reach the maximum of infinity. But the real within-class scatter is bigger than the estimated within-class scatter because of the limited sample number. Hence, the regularization parameter γ is introduced to improve the UMLDA method, which is called Regularized UMLDA(R-UMLDA). γ in this paper is empirically set to 10^{-5}. However, R-UMLDA is not very stable because of different initialization and regularization. The regularized UMLDA with aggregation (R-UMLDA-A) is proposed to aggregate A different initialized and regularized UMLDA to ease the effects of different initialization. A is empirically set to 20 and regularization parameter γ ranges from 10^{-7} to 10^{-2}.

3.2 Simulation results

According to the simulation described above, the multi-channel profile data is generated and different algorithms are applied and compared with each other. The features extracted from those algorithms are fed into the nearest neighbor classifier (NNC) [27]. Let $\boldsymbol{y} \in R^L$ denote the L-dimensional features extracted from multi-channel profile data and $\boldsymbol{y}_i = \left(y_i^1, y_i^2, \ldots, y_i^L\right)^T, \boldsymbol{y}_j = \left(y_j^1, y_j^2, \ldots, y_j^L\right)^T$. The output of NNC is a class label which is assigned to the class of the nearest neighbor measured by L_p distance:

Table 2 Average time spent in one experiment (s)

Number	VPCA	MPCA	UMPCA	UMLDA	R-UMLDA	R-UMLDA-A	I-UMLDA
1	0.12	0.77	0.25	1.65	1.76	6.50	3.03
2	0.14	0.48	0.35	2.02	2.01	10.44	3.16
3	0.13	0.50	0.46	2.46	2.50	15.78	3.22
4	0.13	0.48	0.54	2.92	2.85	20.64	3.40
5	0.14	0.52	–	3.23	3.20	25.88	3.45
6	0.14	0.52	–	3.76	3.72	30.57	3.48
7	0.14	0.49	–	4.21	4.52	39.01	3.69
8	0.13	0.49	–	4.57	4.82	46.43	3.66
9	0.14	0.48	–	5.25	5.12	50.10	4.38
10	0.14	0.48	–	5.53	5.53	56.01	3.79
20	0.14	0.48	–	9.55	9.60	108.49	4.84
30	0.14	0.48	–	13.82	13.84	162.09	5.69

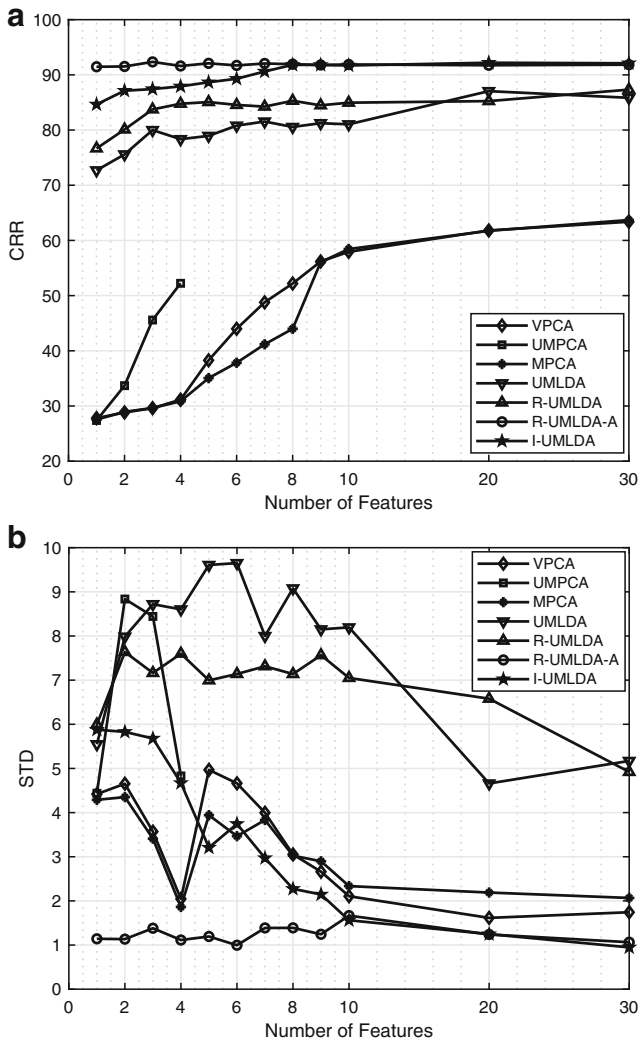

Fig. 5 Classification performance of NNC for different feature extractors. **a** Mean of the correct classification rate (CRR). **b** Standard deviation of CRR (STD)

$$L_p = \left(\sum_{l=1}^{L} \left| y_i^l - y_j^l \right|^p \right)^{\frac{1}{p}} \tag{10}$$

when $p = 2$, L_p distance is Euclidean distance which are usually used in NNC.

It should be noted that this paper focuses on multilinear feature extraction and mapping the original space to the new subspace where profile data has the greatest separability. In other words, the results of performance are mainly contributed by the feature extraction algorithms rather than the classifier. The classification accuracy of the proposed method can be improved if a more sophisticated classifier such as the support vector machine (SVM) is used instead of the NNC which is a simple classifier. However, such an experiment is out of the scope of this paper. The performance of these algorithms is evaluated and compared based on the following criteria:

1. Correct classification rate (CRR): $R = \frac{\sum_{m=1}^{M_{test}} M_{\tilde{c}_m = c_m}}{M_{test}}$, where M_{test} is the total number of test samples, and $M_{\tilde{c}_m = c_m}$ is the test sample which the predicted class label $\tilde{c}_m$ equals to true class label c_m.
2. The standard deviation of CRR (σ_{CRR}): $\sigma_{CRR} = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(CRR_i - \overline{CRR} \right)^2}$, where the N_{test} is the number of experiments, and CRR_i is the Correct classification rate of ith experiment. $\overline{CRR}$ is the mean of CRR: $\overline{CRR} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} CRR_i$.
3. Average time spent in one experiment.

In order to study the performance with different dimensions of extracted features, 100 experiments were performed for each projection with the projected number of features $N = 1, 2, \ldots, 10, 20, 30$. The tenfold cross-validation is applied to better evaluate the performance of different methods. Tables 1 and 2 show the results.

It should be noted that UMPCA produces up to four features which are uncorrected because the number of features extracted by UMPCA is upper-bounded by $min\{min_n I_n, M\}$,

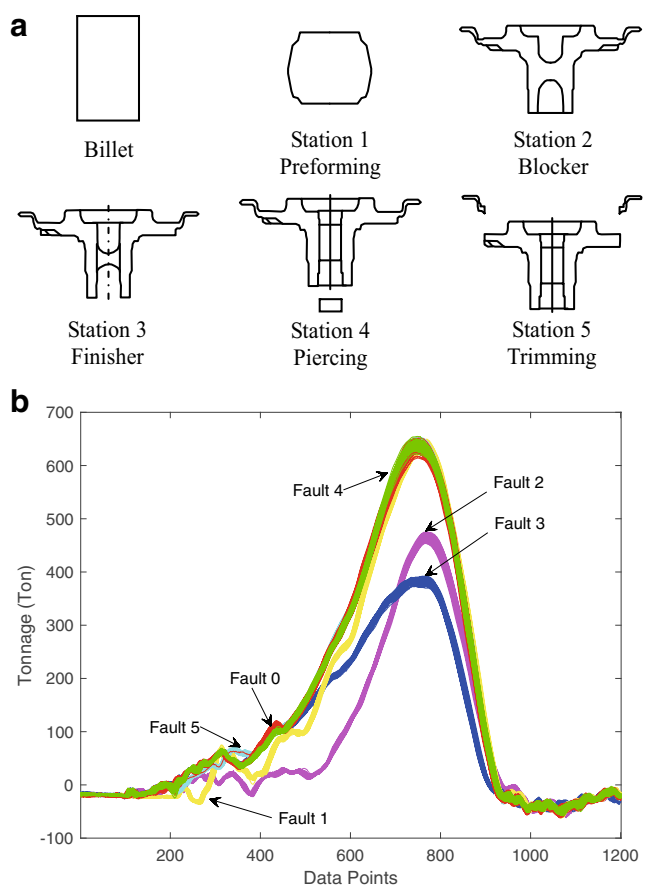

Fig. 6 **a** Shape of workpieces at each operation. **b** Overlapping samples of aggregated tonnage profiles for normal and missing operations

Table 3 Tonnage profile segments [19]

	Segment								
	1	2	3	4	5	6	7	8	9
Interval	[1153]	[154,212]	[213,296]	[297,447]	[448,560]	[561,635]	[636,816]	[817,865]	[866,1200]

where I_n is the n-mode dimensionality and M is the number of samples.

The plotted results are shown in Fig. 5. It is apparent that the results of VPCA and UMPCA are worse than methods based on LDA. It is consistent with our understanding that LDA-based methods take class information into classification when reducing dimensionality while PCA-based methods only seek projections to maximize the variability. It also shows that the I-UMLDA outperforms UMLDA and R-UMLDA not only in the mean of CRR but also in the standard deviation of CRR. The mean of CRR using I-UMLDA is higher than that using UMLDA and R-UMLDA in Fig. 5a. The standard deviation of CRR using I-UMLDA decreases with the increase of dimension and also smaller than these two methods which are shown in Fig. 5b. The results of R-UMLDA-A are slightly better than I-UMLDA. However, in Table 2, the cost of R-UMLDA-A is also larger in that the average time spent is significantly larger than I-UMLDA. When the dimension of projected features is 30, the running time of R-UMLDA-A is about 31 times longer than that of I-UMLDA.

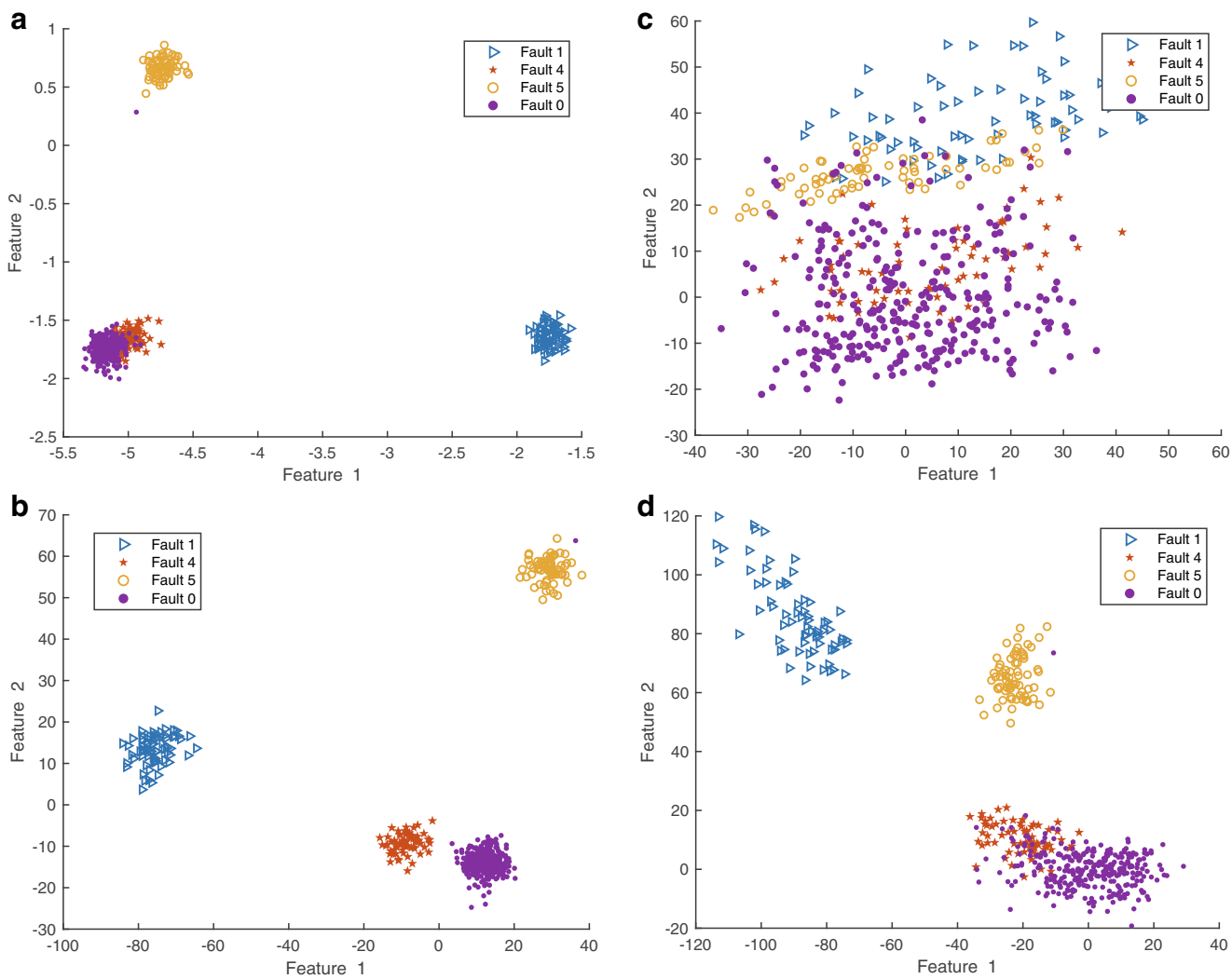

Fig. 7 Scatter plots across different fault groups for features extracted from different methods. **a** UMLDA. **b** I-UMLDA. **c** UMPCA. **d** MPCA

4 Case study

In this section, the I-UMLDA with TTP and other different algorithms are applied to a real-world multi-operation forging process. As illustrated in Fig. 1, four sensors are installed on the different uprights of the forging machine, which record the force exerting on dies represented as four-channel profile data. Five different dies are working together to produce a final product which performs five operations in a sequence of (1) performing, (2) blocking, (3) finishing, (4) piercing, and (5) trimming. A shape sketch of raw billet, intermediate and final parts after each operation of the selected forging process are shown in Fig. 6a. The blocking and finishing operations generate large signals because they make significant shape changes on the product while the piercing and trimming operations generate small signals which can also be called weak signals. It is more difficult to detect missing parts in such operations.

The improved algorithm will be used for missing parts detection with the four-channel profile data. Each profile data contains rich information about the product quality which can be used for process conditions classification. The recorded profiles will be classified as either a normal working production without missing parts in all stations or a fault condition due to a missing part. A training multi-channel profile dataset including six groups is collected. Fault i ($i = 1, \ldots, 5$) is a faulty condition with a missing part in station i and fault 0 is the normal working production. The overlapping samples of aggregated multi-channel profile data for normal working conditions with 308 samples and 69 samples under each 5 faulty conditions are depicted in Fig. 6b. It can be seen that fault 2 and fault 3 can be easily detected from the fault 0 (normal working condition) even by visual inspection of a profile. Hence, these two faults are excluded from the analysis. In addition, the profile signals partially change at the specific signal segment when a missing part happens. The profile signals can be segmented into several parts which specify the working boundary of each operation. This paper will extract features and classify faults by the corresponding signal segments rather than the whole cycle of the signals in order to increase the detection sensitivity and robustness. Lei et al. [11] added up all channel profiles and applied the vectorized method to the aggregated tonnage profiles to extract features. Paynabar et al. [19] did not break the tensor structure of multi-channel profile data and applied UMPCA to analyze multi-channel profiles that considered the interrelationship of different profile channels. Each profile can be divided into nine segments based on their study. Each boundary corresponding to each segment of a profile is shown in Table 3. UMPCA cannot classify fault 1, fault 4, and fault 5 from fault 0 by one segment for the reason that UMPCA do not consider class information when extracting features. Therefore, more segments were used to classify different faults. Segment 4 was selected for detecting fault 4 from fault 0, while segment 3 was chosen for detecting fault 1 and fault 5 from fault 0 in [19]. In this paper, only segment 4 is needed to detect fault 1, fault 4, and fault 5 from fault 0 with UMLDA-based method, which greatly improved the efficiency.

As described in simulation, the procedures in Fig. 4 are executed step by step for feature extraction and classification. The algorithms discussed in simulations are applied to these multi-channel profiles. As can be seen in Fig. 7a, b, signals for fault 4, which are "weak signals" mentioned in the introduction, are difficult to detect. The features extracted from UMLDA lead to worse results than features extracted by I-UMLDA. In contrast, fault 1, fault 4, and fault 5 can be completely detected from fault 0 using I-UMLDA. And from Fig. 7c, d, the features extracted by PCA-based methods are hard to separate as expected.

As shown in Fig. 7a, b, it should also be mentioned that the scale of new feature space mapped by I-UMLDA is much larger than the scale of new feature space mapped by UMLDA which will improve the data separability. The extracted features were classified with NNC classifier. A tenfold cross-validation method was applied in order to evaluate the performance of the classification. Table 4 shows the overall confusion matrices of NNC classifiers for UMLDA and I-UMLDA features obtained from tenfold cross-validation.

As expected from the scatter plots in Fig. 7, faults 1 and 5 can be detected and classified using both I-UMLDA and UMLDA. However, fault 4 is hard to detect from fault 0, and the features extracted from I-UMLDA lead to better results than that obtained by UMLDA. On average, 93.01% of the missing parts can be accurately classified using UMLDA while 99.81% of the missing parts can be accurately classified using I-UMLDA. The results indicate that the proposed I-UMLDA has a better performance of fault detection and diagnosis than that of UMLDA.

Table 4 Confusion matrix of NNC for UMLDA and I-UMLDA features

	Classified as			
	Fault 0	Fault 1	Fault 4	Fault 5
UMLDA				
Actual				
Fault 0	292	0	15	1
Fault 1	0	69	0	0
Fault 4	20	0	49	0
Fault 5	0	0	0	69
I-UMLDA				
Actual				
Fault 0	307	0	0	1
Fault 1	0	69	0	0
Fault 4	0	0	69	0
Fault 5	0	0	0	69

5 Conclusion

An I-UMLDA method based on TTP is proposed to analyze multi-channel profile (tensor) data. The original tensor data can be projected into another tensor which can capture most of the variations in the input high-dimensional data before using UMLDA method. The data transformed in the subspace is arranged according to the importance in each order. It can reduce the effects of different order on the accuracy and achieve better fault detection performance in reducing the fluctuation of accuracy and improving the accuracy of detection. The simulation was implemented to assess the performance of the improved algorithm, which shows the proposed method has better performance than UMLDA as well as other competing algorithms. The results indicated that the features extracted from I-UMLDA are more separable and the correct classification rates are less fluctuant than UMLDA and competitor methods. These methods were also applied to a real-world case study of a multi-operation forging process for fault detection and diagnosis. The results showed that I-UMLDA outperforms UMLDA in classifying different faulty types. Further research can be carried out in developing tensor-based algorithms for data fusion, fault detection, and classification.

Funding This study was supported by the National Natural Science Foundation of China (NSFC) under Grant No.51775108.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

In (2), when $D_n = I_n$, $n = 1, \ldots, N$, $Z_{\Phi^{(n)}} \cdot Z^T_{\Phi^{(n)}}$ is an identity matrix.

By successive application of the transpose property of the Kronecker product $(\mathrm{M} \otimes \mathrm{N})^{\mathrm{T}} = \mathrm{M}^{\mathrm{T}} \otimes \mathrm{N}^{\mathrm{T}}$ [28].

$$Z_{\Phi^{(n)}}{}^T = \left(Z^{(n+1)^T} \otimes Z^{(n+2)^T} \otimes \ldots \otimes Z^{(N)^T} \otimes Z^{(1)^T} Z^{(2)^T} \otimes \ldots Z^{(n-1)^T}\right) \quad (11)$$

By the Kronecker product theorem $(\mathrm{M} \otimes \mathrm{N})(\mathrm{X} \otimes \mathrm{Y}) = (\mathrm{MX}) \otimes (\mathrm{XY})$ [28],

$$Z_{\Phi^{(n)}} \cdot Z_{\Phi^{(n)}}{}^T = \left(Z^{(n+1)} Z^{(n+1)^T} \otimes \ldots \otimes Z^{(N)} Z^{(N)^T} \otimes Z^{(1)} Z^{(1)^T} \otimes \ldots Z^{(n-1)} Z^{(n-1)^T}\right) \quad (12)$$

For all n, when $D_n = I_n$, $Z^{(n)}$ is a square matrix, $Z^{(n)T} Z^{(n)} = \boldsymbol{I}_{I_n}$, where $\boldsymbol{I}_{I_n}$ is an $I_n \times I_n$ identity matrix. Thus, $Z_{\Phi^{(n)}} \cdot Z_{\Phi^{(n)}}{}^T = \boldsymbol{I}_{I_1 \times I_2 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N}$.

References

1. Paynabar K, Jin J (2011) Characterization of non-linear profiles variations using mixed-effect models and wavelets. IIE Trans 43(4):275–290. https://doi.org/10.1080/0740817x.2010.521807
2. Lee SS, Shao C, Kim TH, Hu SJ, Kannatey-Asibu E, Cai WW, Spicer JP, Abell JA (2014) Characterization of ultrasonic metal welding by correlating online sensor signals with weld attributes. J Manuf Sci E T ASME 136(5):051019. https://doi.org/10.1115/1.4028059
3. Noorossana R, Saghaei A, Amiri A (2011) Statistical analysis of profile monitoring. Wiley series in probability and statistics. Wiley, Hoboken
4. Guo W, Shao C, Kim TH, Hu SJ, Jin J, Spicer JP, Wang H (2016) Online process monitoring with near-zero misdetection for ultrasonic welding of lithium-ion batteries: an integration of univariate and multivariate methods. J Manuf Syst 38:141–150. https://doi.org/10.1016/j.jmsy.2016.01.001
5. Grasso M, Colosimo BM, Tsung F (2017) A phase I multi-modelling approach for profile monitoring of signal data. Int J Prod Res 55(15):4354–4377. https://doi.org/10.1080/00207543.2016.1251626
6. Paynabar K, Jin J, Agapiou J, Deeds P (2012) Robust leak tests for transmission systems using nonlinear mixed-effect models. J Qual Technol 44(3):265–278
7. Zhang Z, Wang Y, Wang K (2013) Intelligent fault diagnosis and prognosis approach for rotating machinery integrating wavelet transform, principal component analysis, and artificial neural networks. Int J Adv Manuf Technol 68(1–4):763–773. https://doi.org/10.1007/s00170-013-4797-0
8. Lahdhiri H, Said M, Abdellafou KB, Taouali O, Harkat MF (2019) Supervised process monitoring and fault diagnosis based on machine learning methods. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-019-03306-z
9. Fazai R, Taouali O, Harkat MF, Bouguila N (2016) A new fault detection method for nonlinear process monitoring. Int J Adv Manuf Technol 87(9–12):3425–3436. https://doi.org/10.1007/s00170-016-8745-7
10. Lu Y, Xie R, Liang SY (2019) Bearing fault diagnosis with nonlinear adaptive dictionary learning. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-019-03455-1
11. Lei Y, Zhang Z, Jin J (2010) Automatic tonnage monitoring for missing part detection in multi-operation forging processes. J Manuf Sci E T ASME 132(5):051010. https://doi.org/10.1115/1.4002531
12. Zerehsaz Y, Shao C, Jin J (2019) Tool wear monitoring in ultrasonic welding using high-order decomposition. J Intell Manuf 30(2):657–669. https://doi.org/10.1007/s10845-016-1272-4
13. Xie Z, Li J, Lu Y (2019) Feature selection and a method to improve the performance of tool condition monitoring. Int J Adv Manuf Technol 100(9):3197–3206. https://doi.org/10.1007/s00170-018-2926-5
14. Duo A, Basagoiti R, Arrazola PJ, Aperribay J, Cuesta M (2019) The capacity of statistical features extracted from multiple signals to predict tool wear in the drilling process. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-019-03300-5
15. Bhattacharyya P, Sengupta D (2009) Estimation of tool wear based on adaptive sensor fusion of force and power in face milling. Int J Prod Res 47(3):817–833. https://doi.org/10.1080/00207540701403376
16. Paynabar K, Zou C, Qiu P (2016) A change-point approach for phase-I analysis in multivariate profile monitoring and diagnosis. Technometrics 58(2):191–204. https://doi.org/10.1080/00401706.2015.1042168

17. Grasso M, Pennacchi P, Colosimo BM (2014) Empirical mode decomposition of pressure signal for health condition monitoring in waterjet cutting. Int J Adv Manuf Technol 72(1–4):347–364. https://doi.org/10.1007/s00170-014-5671-4
18. Kong X, Chang J, Niu M, Huang X, Wang J, Chang SI (2018) Research on real time feature extraction method for complex manufacturing big data. Int J Adv Manuf Technol 99(5):1101–1108. https://doi.org/10.1007/s00170-016-9864-x
19. Paynabar K, Jin J, Pacella M (2013) Monitoring and diagnosis of multichannel nonlinear profile variations using uncorrelated multilinear principal component analysis. IIE Trans 45(11):1235–1247. https://doi.org/10.1080/0740817x.2013.770187
20. Lu H, Plataniotis KN, Venetsanopoulos AN (2009) Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning. IEEE Trans Neural Netw 20(11):1820–1836. https://doi.org/10.1109/tnn.2009.2031144
21. Grasso M, Colosimo BM, Pacella M (2014) Profile monitoring via sensor fusion: the use of PCA methods for multi-channel data. Int J Prod Res 52(20):6110–6135. https://doi.org/10.1080/00207543.2014.916431
22. Pacella M (2018) Unsupervised classification of multichannel profile data using PCA: an application to an emission control system. Comput Ind Eng 122:161–169. https://doi.org/10.1016/j.cie.2018.05.029
23. Lu H, Plataniotis KN, Venetsanopoulos AN (2009) Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition. IEEE Trans Neural Netw 20(1):103–123. https://doi.org/10.1109/tnn.2008.2004625
24. Lu H, Konstantinos NP, Venetsanopoulos AN (2008) MPCA: multilinear principal component analysis of tensor objects. IEEE Trans Neural Netw 19(1):18–39. https://doi.org/10.1109/tnn.2007.901277
25. Donoho DL, Johnstone IM (1995) Adapting to unknown smoothness via wavelet shrinkage. J Am Stat Assoc 90(432):1200–1224. https://doi.org/10.2307/2291512
26. Bharati MH, Liu JJ, MacGregor JF (2004) Image texture analysis: methods and comparisons. Chemom Intell Lab Syst 72(1):57–71. https://doi.org/10.1016/j.chemolab.2004.02.005
27. Zhang M, Zhou Z (2007) ML-KNN: a lazy learning approach to multi-label leaming. Pattern Recogn 40(7):2038–2048. https://doi.org/10.1016/j.patcog.2006.12.019
28. Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Rev 51(3):455–500. https://doi.org/10.1137/07070111x