Computers & Industrial Engineering 122 (2018) 161–169

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Unsupervised classification of multichannel profile data using PCA: An application to an emission control system

Massimo Pacella

Dipartimento di Ingegneria dell'Innovazione, Università del Salento, Piazza Tancredi 7, 73100 Lecce, Italy

ARTICLE INFO

Keywords:
Unsupervised classification
Data clustering
Principal component analysis
Fault diagnosis

ABSTRACT

Modern sensing technologies have facilitated real-time data collection for process monitoring and fault diagnosis in several research fields of industrial engineering. The challenges associated with diagnosis of multichannel (multiple) profiles are yet to be addressed in the literature. Motivated by an application of fault diagnosis of an emission control system, this paper proposes an approach for efficient and interpretable modeling of multichannel profile data in high-dimensional spaces. The method is based on unsupervised classification of multichannel profile data provided by several sensors related to a fault event. The final goal is to isolate fault events in a restricted number of clusters (scenarios), each one described by a reference pattern. This can provide practitioners with useful information to support the diagnosis and to find root cause. Two multilinear extensions of principal component analysis (PCA), which can analyze the multichannel profiles without unfolding the original data set, are investigated and compared to regular PCA applied to vectors generated by unfolding the original data set. The effectiveness of multilinear extensions of PCA is demonstrated using an experimental campaign carried out on an emission control system. Results of unsupervised classification show that the multilinear extension of PCA may lead to a classification with better compactness and separation of clusters.

1. Introduction

Modern sensing technologies have facilitated real-time data collection for process monitoring and fault diagnosis in several research fields of industrial engineering. Profile or functional data is one of the common data types collected by sensing systems. Process monitoring and fault diagnosis using profile data remains an important and challenging problem in statistical process control (SPC) (Wang & Huang, 2017). Although profile monitoring and fault diagnosis has been extensively studied in the SPC literature, the challenges associated with monitoring and diagnosis of multichannel (multiple) profiles are yet to be addressed (Grasso, Colosimo, & Pacella, 2014; Paynabar, Jin, & Pacella, 2013; Paynabar, Qiu, & Zou, 2016; Zhang, Ren, Yao, Zou, & Wang, 2015).

For example, a combustion engine converts chemical energy of fuel by a redox reaction with oxygen, producing gas with polluting substances such as nitrogen oxides (NO_x). The emission control issue has assumed an important role in industrial engineering, as well as in the context of transportation (Salehi, Jalalian, & Siar, 2017). In order to fulfill legislation requirements, a high number of on-board sensors have been introduced in order to monitor both the combustion and the exhaust gas aftertreatment process. This opens the way for the collection of a large amount of real-time data coming from different sensors (multichannel profiles). Nowadays, the compliance of combustion engine with stringent emission regulations make necessary to have more efficient and more reliable emission control systems.

The NO_x Storage Catalyst (NSC) is an emission control system in which a exhaust gas aftertreatment process reduces emissions by alternating two phases: (i) adsorption and (ii) regeneration. During adsorption, the NO_x is captured by an adsorber that traps the NO_x molecules. During regeneration, when the adsorber is saturated, the stored NO_x is reduced in a catalytic process. In this phase, the NSC system may be affected by a specific fault that reduces the efficiency of NO_x conversion. This fault is manifested as an excessive deviation of one characteristic parameter of the system from the standard condition (Isermann & Ball, 1997). In particular, it consists of a relative air/fuel ratio (λ) that falls below an acceptability threshold (λ-undershoot fault). Technical *diagnosis* of the λ-undershoot fault of an NSC system (Soliman, Rizzoni, & Kim, 1999), i.e. the determination of the kind of causes for the fault detected by the λ-sensor, assumes an extreme importance in order to fulfill legislation requirements.

An unsupervised classification of profile data provided by several sensors during a fault event, should be considered to support fault diagnosis. The unsupervised classification is based on partitional clustering of profile data with the objective to isolate the fault events in a restricted number of scenarios, each one described by a reference

E-mail address: massimo.pacella@unisalento.it.

https://doi.org/10.1016/j.cie.2018.05.029
Received 16 January 2018; Received in revised form 16 May 2018; Accepted 18 May 2018
Available online 01 June 2018
0360-8352/

pattern. Then, this pattern could be then analyzed by an expert for decision-making, i.e. to find root cause. The high dimensionality of the data set provides a considerable challenge to clustering approaches. First, the curse of dimensionality can make algorithms for clustering slow, and second, the existence of many irrelevant features may not allow the identification of the relevant underlying structure in the data (Boutsidis, Drineas, & Mahoney, 2009). An approach to this problem consists to apply dimensionality reduction techniques to the raw data set in order to obtain a more compact representation of the original high-dimensional data that nonetheless captures all the information necessary for decision-making. The application of a dimensionality reduction technique consists of: (i) projecting data to a low-dimensional space; (ii) extracting relevant features from data for fault diagnosis; (iii) eliminating noise; and (iv) compressing the data to reduce storage requirements (Jiang, Jia, Hu, & Xu, 2009).

Dimensionality reduction methods include principal component analysis (PCA) (Jolliffe, 2002), multi-dimensional scaling (Cox & Cox, 1994) and independent component analysis (Hyvärinen & Oja, 2000). All of these methods can be applied to multichannel profile data by implementing a vectorization operation, i.e. by unfolding the original data set. Several authors have pointed out the inefficacy of standard dimensionality reduction methods in these situations (Paynabar et al., 2013; Yan, Paynabar, & Shi, 2015). This issue motivates the utilization of two multilinear extensions of PCA, which can analyze the multichannel profiles without unfolding the original data set. In particular, the multilinear principal component analysis (MPCA) (Lu, Plataniotis, & Venetsanopoulos, 2008) and the uncorrelated multilinear principal component analysis (UMPCA) (Lu, Plataniotis, & Venetsanopoulos, 2009). These methods can preserve the original correlation structure of the multistream structured data. Therefore, they can provide more efficient dimension reduction and feature extraction compared to a vectorization operation. Regular PCA based on vectorized data (VPCA) is used as benchmark in our study (Paynabar et al., 2013).

The remainder of the paper is organized as follows. In Section 2, the dimensionality reduction applied to multichannel profile issue is introduced. In Section 3, the unsupervised classification method is presented, while the benchmark indexes for clustering validation are described in subsequent Section 4. In Section 5, the approach is applied to real case of study related to a specific fault-event analysis (λ-undershoot) of the NSC system. The performance of VPCA, MPCA and UMPCA are then compared. Finally, Section 6 provides concluding remarks.

2. Dimensionality reduction of multichannel profiles

The objective of PCA is to reduce the dimensionality of a large number of interrelated variables of one-dimensional arrays (vectors) by linearly transforming them to a new set of variables, called principal components (PCs), which are uncorrelated and ordered so that the first few retain most of the original data variability. However, PCA cannot be used directly to two-dimensional arrays (matrices) or higher-dimensional arrays, unless such data are reshaped into one-dimensional arrays. There are two drawbacks to this reshaping into a one-dimensional array (vectorization): (i) spatial locality information may be lost (e.g. positions which are neighbors in the original higher-dimensional representation, may result far away from each other in the vectorized representation), (ii) the alignment leads to overparameterization causing higher time and space costs due to the high number of variables in the unfolded vectors.

The *Multilinear Principal Component Analysis* (MPCA) is a generalization of the PCA where the input can be not only vectors, but also tensors, i.e. a multidimensional array of data (Lu et al., 2008). Tensor-based PCA methods, such as MPCA, directly analyze a tensor without reshaping it into a vector, thus preserving the tensor structure of the original data set. The PCs obtained from MPCA are correlated contrary to PCA. In Lu et al. (2009) an extension of the MPCA named *Uncorrelated Multilinear Principal Component Analysis* (UMPCA) was proposed, which introduces the zero-correlation constraint among features. The use of both MPCA and UMPCA for modeling multichannel profile data have been presented in the literature (Grasso et al., 2014; Masoud, Zerehsaz, & Jin, 2017; Paynabar et al., 2013). Recently, in Pacella and Colosimo (in press) the use of MPCA has been proposed for modeling cylindrical surfaces obtained from a manufacturing process and for control chart implementation.

2.1. Basic notation and definition

1-order arrays (vectors) are denoted by lower-case boldface letters ($\mathbf{x}$), 2-order arrays (matrices) by upper-case boldface ($\mathbf{X}$) and 3rd-order arrays (tensors) by calligraphic letters ($\mathcal{X}$). Their elements are denoted with indexes in brackets, where the indexes are denoted by lower-case letters and span the range from 1 to the upper-case letter index (e.g. $\mathbf{x}(d)$, $\mathbf{X}(d, n)$, $\mathcal{X}(d, n, m)$ where $d = 1, ...,D$, $n = 1, ...,N$ and $m = 1, ...,M$).

Consider the case of a N-channel signal profile, in which each profile consists of D data points. A data set of M samples is stored in a 3rd-order tensor denoted as $\mathcal{X} \in \mathbb{R}^{D\times N\times M}$ and addressed by 3 indexes, d, n and m related to the so-called 1-mode, 2-mode and 3-mode of $\mathcal{X}$. By definition, the l-mode vectors ($l = 1, 2, 3$) of $\mathcal{X}$ are defined as the vectors obtained by varying the l-th index while keeping all the others fixed.

2.2. Singular value decomposition and PCA

The Vectorized PCA (VPCA) represents the application of regular PCA to tensor data, after reshaping tensors into vectors. In the case of multichannel profiles, the N-channel signal profile of index $m = 1, ...,M$, which is measured on D points, is stored in matrix $\mathbf{X}_m \in \mathbb{R}^{D\times N}$, or alternatively it can be reshaped into vector $\mathbf{x}_m \in \mathbb{R}^{D\cdot N}$, where $\mathbf{x}_m = \mathrm{vec}(\mathbf{X}_m)$ (is the vectorized operator and transforms a matrix into a vector by stacking all the columns of this matrix one underneath the other).

Let $\bar{\mathbf{x}} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{x}_m$ be the average vector of unfolded data and let $\mathbf{x}_m^c = \mathbf{x}_m - \bar{\mathbf{x}}$ be the centered vector obtained from $\mathbf{x}_m$ by subtracting the average vector. The entire data set can be represented by matrix $\mathbf{X}^c \in \mathbb{R}^{P\times M}$ (the rows represent the $P = D\cdot N$ variables and the columns are the M samples $\mathbf{x}_m^c$).

The covariance matrix is $\hat{\boldsymbol{\Sigma}} = \frac{1}{M-1}\mathbf{X}^c\mathbf{X}^{c^T}$, and is obtained re-scaling by a constant $1/(M-1)$ the *scatter* matrix of $\mathbf{X}^c$ defined as $\mathbf{X}^c\mathbf{X}^{c^T}$. The *Singular Value Decomposition* (SVD) of $\mathbf{X}^c$ is as follows.

$$\mathbf{X}^c = \mathbf{USZ}^T = \mathbf{U}^{(1)}\mathbf{SU}^{(2)^T} = \mathbf{S}\times_1\mathbf{U}^{(1)}\times_2\mathbf{U}^{(2)}. \quad (1)$$

On the right-hand in Eq. (1), the *j-mode product* ($j = 1, 2$) of matrix $\mathbf{S} \in \mathbb{R}^{P\times M}$ by matrices $\mathbf{U}^{(1)} \in \mathbb{R}^{P\times P}$ and $\mathbf{U}^{(2)} \in \mathbb{R}^{M\times M}$, is denoted by $\mathbf{S}\times_j\mathbf{U}^{(j)}$ and results in a matrix with entries $(\mathbf{S}\times_1\mathbf{U}^{(1)})(j_1, i_2) = \sum_{i_1=1}^{P}\mathbf{S}(i_1, i_2)\mathbf{U}^{(1)}(j_1, i_1)$ and $(\mathbf{S}\times_2\mathbf{U}^{(2)})(i_1, j_2) = \sum_{i_2=1}^{M}\mathbf{S}(i_1, i_2)\mathbf{U}^{(2)}(j_2, i_2)$.

$\mathbf{U} \in \mathbb{R}^{P\times P}$ in Eq. (1) is a unitary matrix (i.e. it has columns which form an orthonormal basis of $\mathbb{R}^P$), whose columns are the eigenvectors of the scatter matrix $\mathbf{X}^c\mathbf{X}^{c^T}$ (and of $\hat{\boldsymbol{\Sigma}}$). These eigenvectors are called the *left singular vectors* of $\mathbf{X}^c$. $\mathbf{S} \in \mathbb{R}^{P\times M}$ is pseudodiagonal and contains the *singular values* of $\mathbf{X}^c$. Without loss of generality, the elements in $\mathbf{S}$ are assumed arranged in a decreasing order (columns of $\mathbf{U}$ and $\mathbf{Z}$ are correspondingly arranged). $\mathbf{Z} \in \mathbb{R}^{M\times M}$ is a unitary matrix, whose columns are the eigenvectors of the scatter matrix $\mathbf{X}^{c^T}\mathbf{X}^c$, also called the *right singular vectors* of $\mathbf{X}^c$. As the columns of matrix $\mathbf{U} = \mathbf{U}^{(1)}$ form a unitary basis for the space $\mathbb{R}^P$, the coordinates in this new basis for the data set stored in $\mathbf{X}^c$ are $\mathbf{U}^T\mathbf{X}^c = \mathbf{SZ}^T$, i.e. $\mathbf{X}^c\times_1\mathbf{U}^{(1)^T} = \mathbf{S}\times_2\mathbf{U}^{(2)}$.

The aim of PCA is to solve the problem of approximating the data matrix $\mathbf{X}^c$ with another matrix $\widetilde{\mathbf{X}}^c$ which has a lower rank ($rank(\widetilde{\mathbf{X}}^c) < rank(\mathbf{X}^c)$), where the approximation objective is to minimize the distance between $\mathbf{X}^c$ and $\widetilde{\mathbf{X}}^c$:

$$\min_{\widetilde{\mathbf{X}}^c}(\|\mathbf{X}^c-\widetilde{\mathbf{X}}^c\|_F) \quad s.t. \quad rank(\widetilde{\mathbf{X}}^c) \leqslant P'. \tag{2}$$

P' is a given upper bound for the rank of matrix $\widetilde{\mathbf{X}}^c$ ($P' < P$). By the SVD in Eq. (1), matrix $\widetilde{\mathbf{X}}^c$ can be obtained as $\widetilde{\mathbf{X}}^c = \widetilde{\mathbf{S}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)}$ where $\widetilde{\mathbf{S}}$ is the same matrix as $\mathbf{S}$ except that it contains only the P' larger singular values (the other singular values are replaced by zero).

Hence, denoting by $\widetilde{\mathbf{U}}^{(1)}$ the matrix formed by the first P' columns of $\mathbf{U}^{(1)}$, which correspond to the first P' larger singular values of $\mathbf{X}^c$, a data sample vector of P points $\mathbf{x}_m$ ($m = 1, ...,M$) is projected to a feature space as $\widetilde{\mathbf{U}}^{(1)^T}(\mathbf{x}_m-\overline{\mathbf{x}}) = (\mathbf{x}_m-\overline{\mathbf{x}}) \times_1 \widetilde{\mathbf{U}}^{(1)^T}$. This is the vector of P' coordinates $\mathbf{y}_m = (y_{m1}, ...,y_{mP'})$ which represent the so-called *scores* (PC-features) of vector $\mathbf{x}_m$.

2.3. High order singular value decomposition

In the targeted application the data set is summarized in a 3rd-order tensor with real entries. Let $\mathscr{X} \in \mathbb{R}^{D\times N\times M}$ represent the tensor addressed by 3 indexes, d, n and m related to the so-called 1-mode, 2-mode and 3-mode of $\mathscr{X}$. By definition, the j-mode *slice* ($j = 1, 2, 3$) of the 3rd-order tensor $\mathscr{X}$ is the matrix obtained by fixing the j-mode index to a constant value. Hence, matrices $\mathbf{X}_m$ are the 3-mode slices of data tensor $\mathscr{X}$. Let $\mathscr{X}^c$ represent the centered data tensor obtained from $\mathscr{X}$ by subtracting the matrix $\overline{\mathbf{X}} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{X}_m$ from each sample $\mathbf{X}_m$ ($m = 1, ...,M$).

Tensor $\mathscr{X}^c$ is decomposed using the generalization of the SVD in Eq. (1) called *Higher Order Single Value Decomposition* (HOSVD) (Lathauwer, Lathauwer, Moor, & Vandewalle, 2000a). HOSVD is also known as *Tucker* decomposition in multilinear algebra. Tucker decomposition decomposes a tensor into a *core tensor* multiplied by a matrix along each mode. In particular, the centered 3rd-order tensor $\mathscr{X}^c$ can be decomposed as follows.

$$\mathscr{X}^c = \mathscr{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}, \tag{3}$$

where $\mathscr{S} \in \mathbb{R}^{D\times N\times M}$, $\mathbf{U}^{(1)} \in \mathbb{R}^{D\times D}$, $\mathbf{U}^{(2)} \in \mathbb{R}^{N\times N}$ and $\mathbf{U}^{(3)} \in \mathbb{R}^{M\times M}$. The j-mode product ($j = 1, 2, 3$) of $\mathscr{S}$ by $\mathbf{U}^{(j)}$ is denoted as $\mathscr{S} \times_j \mathbf{U}^{(j)}$ and has entries equal to $(\mathscr{S} \times_1 \mathbf{U}^{(1)})(d, n, m) = \sum_{i_1=1}^{D}\mathscr{S}(i_1, n, m)\mathbf{U}^{(1)}(d, i_1)$, $(\mathscr{S} \times_2 \mathbf{U}^{(2)})(d, n, m) = \sum_{i_2=1}^{N}\mathscr{S}(d, i_2, m)\mathbf{U}^{(2)}(n, i_2)$, and $(\mathscr{S} \times_3 \mathbf{U}^{(3)})(d, n, m) = \sum_{i_3=1}^{M}\mathscr{S}(d, n, i_3)\mathbf{U}^{(3)}(m, i_3)$ respectively.

$\mathscr{S}$ is the *core tensor* for the HOSVD of $\mathscr{X}^c$. It is a 3rd-order tensor that contains the so-called 1-mode, 2-mode and 3-mode *singular values* of $\mathscr{X}^c$, which are defined as the *Frobenius norm* of the 1-mode, 2-mode and 3-mode slices of tensor $\mathscr{S}$ respectively. By definition, the Frobenius norm of an array is the squared root of the sum of each squared entry of its argument. The entries of $\mathscr{S}$ are real but not necessarily positive (by definition, the singular values of $\mathscr{X}^c$ obtained from the slices of $\mathscr{S}$ are non-negative, as the norm of a matrix is non-negative). The essential difference with the SVD is that $\mathscr{S}$ is, in general, a full tensor, instead of being pseudodiagonal as $\mathbf{S}$ in Eq. (1) (this means that nonzero off-diagonal elements could occur in 1-mode, 2-mode and 3-mode slices of tensor $\mathscr{S}$).

$\mathbf{U}^{(1)} = (\mathbf{u}_1^{(1)}\cdots\mathbf{u}_d^{(1)}\cdots\mathbf{u}_D^{(1)})$, $\mathbf{U}^{(2)} = (\mathbf{u}_1^{(2)}\cdots\mathbf{u}_n^{(2)}\cdots\mathbf{u}_N^{(2)})$and $\mathbf{U}^{(3)} = (\mathbf{u}_1^{(3)}\cdots\mathbf{u}_m^{(3)}\cdots\mathbf{u}_M^{(3)})$ in Eq. (3) are unitary matrices in $\mathbb{R}^{D\times D}$, $\mathbb{R}^{N\times N}$ and $\mathbb{R}^{M\times M}$ respectively (i.e. they have columns which form an orthonormal basis of $\mathbb{R}^D$, $\mathbb{R}^N$ and $\mathbb{R}^M$ respectively). Tensor product in Eq. (3) can be represented equivalently by a *Kronecker product*, i.e. $\mathrm{vec}(\mathscr{X}^c) = (\mathbf{U}^{(3)} \otimes \mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)})\mathrm{vec}(\mathscr{S})$.

Considering the columns of matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$, Eq. (3) can be also rewritten as follows.

$$\mathscr{X}^c = \sum_{d=1}^{D}\sum_{n=1}^{N}\sum_{m=1}^{M}\mathscr{S}(d, n, m)\cdot(\mathbf{u}_d^{(1)}\circ\mathbf{u}_n^{(2)}\circ\mathbf{u}_m^{(3)}), \tag{4}$$

where $(\mathbf{u}_d^{(1)}\circ\mathbf{u}_n^{(2)}\circ\mathbf{u}_m^{(3)})$ denotes the *outer product* of vectors $\mathbf{u}_d^{(1)}$, $\mathbf{u}_n^{(2)}$ and $\mathbf{u}_m^{(3)}$ and results in a 3rd-order tensor in which the generic element of indices i_1, i_2, i_3 is obtained as $\mathbf{u}_d^{(1)}(i_1)\cdot\mathbf{u}_n^{(2)}(i_2)\cdot\mathbf{u}_m^{(3)}(i_3)$ for all values of indices i_1, i_2 and i_3.

Eqs. (3) and (4) are also refereed to as the *Tucker3* model as it is a three-way, which specifies a relationship between factors in the three modes spanned by the data array. The generality of the Tucker3 model, and the fact that it covers other models such as the Candecomp/PARAFAC (or canonical polyadic decomposition) as a special case (Louwerse & Smilde, 2000), has made it an often used model for decomposition, compression, and interpretation in many applications (Rato et al., 2016). More details on the Tucker decomposition and other multi-way analysis models can be found in Smilde, Bro, and Geladi (2004).

In our study, the *Higher-order orthogonal iteration algorithm* proposed in Lathauwer, Lathauwer, Moor, and Vandewalle (2000b) and based on the MATLAB implementation described in Kolda (2001) and Bader and Kolda (2006) is exploited for computing the Tucker decomposition in Eqs. (3) and (4).

2.4. MPCA algorithm

The simplest method to implement dimensionality reduction in the multilinear case is based on truncation of the unitary matrices in Eq. (3). The HOSVD of $\mathscr{X}^c$ can be truncated by keeping $D' \leq D$ orthonormal columns for the unitary matrix $\mathbf{U}^{(1)}$ in the 1-mode and, simultaneously, $N' \leq N$ orthonormal columns for the unitary matrix $\mathbf{U}^{(2)}$ in the 2-mode, such that the projected tensor captures most of variation observed in the original data set.

Through matrices $\widetilde{\mathbf{U}}^{(1)} \in \mathbb{R}^{D\times D'}$ and $\widetilde{\mathbf{U}}^{(2)} \in \mathbb{R}^{N\times N'}$ a basis of $D'\cdot N'$ tensors for the space $\mathbb{R}^{D\times N}$ can be derived. Thus, a centered data sample $(\mathbf{X}_m-\overline{\mathbf{X}})$ ($m = 1, ...,M$), is projected to a lower-dimension feature matrix as follows.

$$\mathbf{Y}_m = (\mathbf{X}_m-\overline{\mathbf{X}}) \times_1 \widetilde{\mathbf{U}}^{(1)^T} \times_2 \widetilde{\mathbf{U}}^{(2)^T} \tag{5}$$

where $\mathbf{Y}_m \in \mathbb{R}^{D'\times N'}$ is the projected matrix. In analogy to regular PCA, this matrix can be represented by a vector of $P' = D'\cdot N'$ coordinates $\mathbf{y}_m = (y_{m1}, ...,y_{mP'})$ which represent the so-called *scores* (PC-features) of matrix $\mathbf{X}_m$.

Similarly to the optimization problem in Eq. (2) for the case of regular PCA, the problem can be formulated in terms of an approximation of tensor $\mathscr{X}^c$ with another tensor $\widetilde{\mathscr{X}}^c$, where the objective is to minimize the norm of the distance between them. In particular, assuming that the targeted dimensionality (D' and N') is specified in advance, the optimization problem can be formulated as follows:

$$\min_{\widetilde{\mathscr{X}}^c}(\|\mathscr{X}^c-\widetilde{\mathscr{X}}^c\|_F) \quad s.t. \quad rank(\widetilde{\mathbf{X}}_{(1)}^c) \leqslant D', rank(\widetilde{\mathbf{X}}_{(2)}^c) \leqslant N', \tag{6}$$

where $\widetilde{\mathbf{X}}_{(1)}^c$ and $\widetilde{\mathbf{X}}_{(2)}^c$ represent the matrices whose column vectors are the 1-mode and 2-mode vectors of tensor $\widetilde{\mathscr{X}}^c$.

Differently from the optimization problem in Eq. (2), the optimal solution for Eq. (6) is not univocally given by a truncation of matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$. In fact, $\mathscr{S}$ is a full tensor, instead of being pseudodiagonal as $\mathbf{S}$ in Eq. (1). In Yan et al. (2015), it was showed that the optimization problem of Eq. (6) is equivalent to maximizing the variation of the projected low-dimensional tensor, known as Multi-linear Principal Component Analysis (MPCA) method proposed in Lu et al. (2008). MPCA is based on an iterative algorithm for finding the projection matrices $\widetilde{\mathbf{U}}^{(1)}$ and $\widetilde{\mathbf{U}}^{(2)}$. This algorithm, called *Sequential Mode Truncation*, can be used when the targeted (reduced) dimensionality in each mode (D', N') is not specified in advance.

2.5. Uncorrelated multilinear PCA

Differently from regular PCA, the extracted features by MPCA may be correlated. For this reason, other multilinear low-rank reduction techniques applicable to tensor data were also considered in the literature (Paynabar et al., 2013, 2016; Yan et al., 2015). In particular, Lu et al. (2009) proposed an extension of the MPCA named Uncorrelated

Multilinear PCA (UMPCA), which introduces the zero-correlation constraint among features derived from an iterative procedure aimed at finding directions capturing maximum variance.

The Elementary Multilinear Projection (EMP) (Lu et al., 2009) projects matrix $\mathbf{X}_m \in \mathbb{R}^{D\times N}$ to a scalar y equal to $y = \mathbf{X}_m \times_1 \mathbf{u}^{(1)^T} \times_2 \mathbf{u}^{(2)^T}$, where $\mathbf{u}^{(1)^T} \in \mathbb{R}^D$, $\mathbf{u}^{(2)^T} \in \mathbb{R}^N$. The UMPCA objective is to determine a set of P' EMPs $\{\mathbf{u}_p^{(1)^T}, \mathbf{u}_p^{(2)^T}\}_{p=1}^{P'}$ that maximize the variance captured while producing uncorrelated feature. In analogy to PCA, the scalars of P' projections form the vector $\mathbf{y}_m = (y_{m1}, ...,y_{mP'})$ which represent the so-called *scores* (PC-features) of matrix $\mathbf{X}_m$.

The zero-correlation constraint introduces a limitation on the maximum number of features that may be extracted. In particular, the number of uncorrelated features P' that can be extracted by UMPCA is upper-bounded by the minimum of tensor mode dimension. This implies that UMPCA may be more appropriate for high-resolution tensor objects where the dimensionality in each mode is high enough to enable the extraction of sufficient number of uncorrelated features.

3. Unsupervised classification

Unsupervised classification, also called cluster analysis, consists in dividing a collection of objects into subgroups or clusters so that elements within each group are similar to each other and dissimilar to those of other groups. A popular approach is hierarchical cluster analysis, which aims at identifying a series of clusters within a nested structure. Despite the conceptual simplicity of the method, there are many situations in which the interest is to identify a certain predefined number of groups. In these situations, partitional clustering methods may be more appropriate. In the targeted application consisting of a large number of variables, partitional clustering methods may be computationally faster than hierarchical clustering. The *k-means* is one of the most used partitional clustering methods.

Let $\mathbf{y}_m = (y_{m1}, ...,y_{mP'})$ be the data vector of P' PCA features for sample of index $m = 1, ...,M$. The similarity measure considered in this work is the squared Euclidean distance $d(\mathbf{y}_m, \mathbf{y}_{m'}) = ||\mathbf{y}_m - \mathbf{y}_{m'}||^2$. Let K be the number of clusters and let C_k be the cluster of index $k = 1, ...K$. The overall within-cluster variance is the sum of the squared distance between each element $\mathbf{y}_m$ and its "closed centroid". It is defined as follows.

$$SSW = \sum_{k=1}^{K} N_k \sum_{\mathbf{y}_m \in C_k} d(\mathbf{y}_m, \overline{\mathbf{y}}_k), \tag{7}$$

where the centroid $\overline{\mathbf{y}}_k = (\overline{y}_{k1}, ...,\overline{y}_{kP'})$ is the mean vector of the k-th cluster and N_k is the number of elements of the k-th cluster. The overall between-cluster variance is defined as follows.

$$SSB = \sum_{k=1}^{K} N_k d(\overline{\mathbf{y}}_k, \overline{\overline{\mathbf{y}}}), \tag{8}$$

where $\overline{\overline{\mathbf{y}}}$ is the mean vector of all K centroids.

The objective of partitional clustering is to find cluster centroids that minimize the within-cluster variance *SSW* (tight clusters) and maximize the between-cluster variance *SSB* (clusters well separated). In order to solve such a problem, which is NP-hard, the *k-means* consists in positioning K points (which represent the centroids of the K clusters) and in repositioning them iteratively so that the clusters stabilize. The only input required by *k-means* clustering is the number of clusters to create. The *k-means* starts with an arbitrary number $K \leq M$ of centroids, which are typically chosen uniformly at random from the data point. At each iteration (until cluster assignments remain unchanged), the algorithm computes the distances between data vectors and centroids, assign each data vector to the closed centroid, and set each cluster centroid to the mean of its assigned cluster.

A variant of the clustering algorithm proposed in Arthur and Vassilvitskii (2007), named *k-means++*, specifies a procedure to initialize the cluster centers before proceeding with the standard *k-means* optimization iterations. The *k-means++* was reported to yield a much better clustering with a lower value of *SSW* than the original *k-means*. In this work, the *k-means++* was implemented for cluster analysis of the P' PCA feature vectors of the M profile sample.

3.1. Selecting the number of clusters by elbow method

The *k-means* algorithm requires to decide in advance the number of clusters K. The basic idea behind *k-means* clustering, is to define clusters such that the total within-cluster sum of square $SSW(K)$ is minimized. The total $SSW(K)$ measures the compactness of the clustering, which should be as small as possible. In order to find the best partition of the data, the clustering algorithm with different values of K is implemented ($K_{min} \leq K \leq K_{max}$). The overall within-cluster variance $SSW(K)$ has a monotonic decreasing function of K. The elbow method looks at the total $SSW(K)$ as a function of the number of clusters K and selects the number of clusters so that adding another cluster does not improve much better the total $SSW(K)$. On a diagram representing the index values against the number of selected clusters, this corresponds to an "elbow".

For any continuous function, the elbow point could be mathematically defined as the point of maximum curvature; however, $SSW(K)$ is not a continuous curve and curvature is not well-defined for discrete data curves. A formal definition for an elbow in discrete data sets based on the mathematical definition of curvature for continuous functions by authors on Satopaa, Albrecht, Irwin, and Raghavan (2011), which presented a new algorithm, named *Kneedle*, for approximating these elbow (or knee) points in discrete data sets. In this work, the *Kneedle* algorithm was used in order to decide the optimal number of cluster K for the *k-means* algorithm.

4. Clustering validation

Clustering validation, which evaluates the goodness of clustering results, can be divided into two categories: external and internal. The main difference is whether or not external information is used for validation. On one hand, external validation measures know the "true" clusters in advance. On the other hand, internal validation measures are based only on information in the data. In many practical applications of clustering, external information such as class labels is often not available. In these situations, internal validation measures are the only option for clustering validation.

Since the aim of clustering is to make objects within the same cluster similar and objects in different clusters distinct, internal validation measures are based on the two criteria: (i) compactness and (ii) separation. The first criterion measures how closely related the objects in a cluster are. Lower within-cluster variance or lower average distance of objects from the cluster centroid indicate better compactness. The second criterion measures how distinct or well-separated a cluster is from other clusters. For example, the pairwise distances between cluster centroids or the pairwise minimum distances between objects in different clusters are used as measures of separation.

4.1. Internal clustering validation measures

A number of internal clustering validation measures for clustering were proposed in the literature. Recently, in Liu et al. (2013) an experimental study has been presented, which investigated the validation properties of a suite of 11 existing internal clustering validation measures in several aspects. The first aspect is the monotonicity of the measure, i.e. it reaches its optimal value (maximum or minimum) as the number of clusters equals to the true cluster number. A second aspect is the noise in data, which can have a significant impact on the performance of an internal validation measure. Another aspect is whether clusters with various density and of unequal size affects the performance of the internal validation measures. By considering these aspects, the following 3 internal clustering validation measures were

selected in this work.

- The Caliński-Harabasz index (Caliński & Harabasz, 1974) evaluates the cluster validity based on the average between- and within-cluster variance. In formula, it can be defined as follows.

$$CH = \frac{M-K}{K-1} \cdot \frac{SSB}{SSW} \tag{9}$$

where M is the total number of objects, K the number of clusters in which objected have been partitioned, SSW is the overall within-cluster variance and SSB is the overall between-cluster variance. In practice, the CH index measures separation based on the maximum distance between cluster centroids, and measures compactness based on the sum of distances between objects and their cluster centroid. A greater CH index shows a better clustering result.

- The Davies-Bouldin index (Davies & Bouldin, 1979) is calculated as follows. For each cluster, the similarities between it and all other clusters are computed, and the highest value is assigned as its cluster similarity. Then the measure index can be obtained by averaging all the cluster similarities. In formula, the DB index can be defined as follows.

$$DB = \frac{1}{K}\sum_{k=1}^{K} \max_{k' \neq k} \frac{(\bar{d}_k + \bar{d}_{k'})}{d_{k,k'}} \tag{10}$$

where K the number of clusters, $\bar{d}_k$ is the average distance between each object in the k-th cluster and the centroid of the k-th cluster, $\bar{d}_{k'}$ is the average distance between each point in the k-th cluster and the centroid of the k'-th cluster and $d_{k,k'}$ is the distance between the centroids of the k-th and k'-th clusters. In practice, $\bar{d}_k$ ($\bar{d}_{k'}$) is a measure of dispersion of cluster k (k'), while $d_{k,k'}$ is a measure of separation between cluster k-th and k'-th. Since the goal is to achieve the minimum within-cluster dispersion and maximum between-cluster separation, the DB index should be minimized. The Euclidean distance is commonly adopted in the calculation of DB index. This implies that the estimation of separation between clusters is reduced to the Euclidean distance between representative points (the centroids), regardless of the distances between the rest of the data.

- The Silhouette index (Rousseeuw, 1987) validates the clustering performance based on the pairwise difference of between- and within-cluster distances. Let $a(\mathbf{y}_m)$ denote the within-cluster mean distance defined as the mean distance of point $\mathbf{y}_m$ to the other points of the cluster it belongs to. Similarly, let us evaluate the mean distance of point $\mathbf{y}_m$ to the points of each of the other clusters. Let $b(\mathbf{y}_m)$ denote the smallest of these mean distances. For each point $\mathbf{y}_m$, consider the following ratio:

$$s(\mathbf{y}_m) = \frac{b(\mathbf{y}_m) - a(\mathbf{y}_m)}{\max(a(\mathbf{y}_m),\ b(\mathbf{y}_m))}, \tag{11}$$

which is called the silhouette width of the point $\mathbf{y}_m$. It is a quantity between -1 and 1: a value near 1 indicates that the point $\mathbf{y}_m$ is assigned to the right cluster whereas a value near -1 indicates that the point should be assigned to another cluster. The mean of the silhouette widths for a given cluster C_k is called the cluster mean silhouette. Finally, the global silhouette index say S is the mean of the mean silhouettes through all the clusters:

$$S = \frac{1}{K}\sum_{k=1}^{K} \frac{1}{N_k} \sum_{\mathbf{y}_m \in C_k} s(\mathbf{y}_m) \tag{12}$$

K is the number of clusters and N_k is the number of elements of the k-th cluster. The optimal cluster partition is determined by maximizing the value of this index. Furthermore, the distribution of the silhouette index among all objects belonging to a given cluster is capable of providing a metric for the quality of clustering.

According to the experimental study presented in Liu et al. (2013) the measures CH, DB and S perform well with reference to the monotonicity aspect and in cases of clusters in a data set with various density and of unequal size. Furthermore, the DB and S measures also influenced by noise in a less sensitive way, when compared to CH.

5. Application to an emission control system

NSC is an emission control system by which the exhaust gas is treated in a two-phase process: (i) adsorption (molecules of NO_x in the exhaust gas are captured by an adsorber) and (ii) regeneration (the stored NO_x is reduced in a catalytic process).

The regeneration phase starts when the NO_x adsorber is saturated. Reliable solid-state NO_x sensors are available and can be used to determine when the adsorber is saturated. During the regeneration phase, of duration ranging between 30 and 90 s, the engine control unit (ECU) is programmed to maintain the combustion process in a rich air-to-fuel status. This status is related to the amount of oxygen present during the combustion process, the relative air/fuel ratio (λ) measured upstream of the NSC (λ_{US-NSC}). During regeneration, λ_{US-NSC} should assume values in the set-point interval 0.92–0.95. However, faults could occur, which are detected by a λ_{US-NSC} ratio falling below an acceptability threshold (0.8–0.9). This kind of fault, which is called λ-undershoot, worsens NSC performance during the regeneration phase. Although λ-undershoot fault can be easily detected by monitoring λ_{US-NSC}, the root cause behind this behavior may changes and are not clearly defined.

In a modern automobile vehicle, the actual vehicle status can be monitored by several on-board sensors collecting real-time data related both to the combustion and the exhaust gas aftertreatment processes. In Fig. 1, 20 on-board sensors of a modern diesel engine are schematically represented. These sensors provide together a large amount of real-time multichannel profile data, which can be used to support diagnosis of the λ-undershoot fault, i.e. the determination of the kind of causes for the poor regeneration performance of the NSC. The high dimensionality of the data sets provided by on-board sensors requires to apply dimensionality reduction techniques to the raw data set in order to obtain a more compact representation of the original high-dimensional data, which nonetheless captures all the information necessary for fault diagnosis.

5.1. Multichannel profile data set

An experimental campaign was carried out on a diesel vehicle in order to collect a data set of multichannel profile data related to the λ-undershoot fault. During the experimental campaign, an instrumented car performed some missions in an area with no relevant elevation changes along a normal traffic route. The collected data set consisted of profile signals, recorded by 20 sensors. Each channel of the data set is referred to a specific sensor. The list of 20 channels is reported in Table 1. The analysis concerns multi-sensor heterogeneous profile data, as the various sensors measures different variables.

The final objective of the proposed approach is to isolate fault events in a restricted number of clusters (scenarios), each one described by a reference pattern. This can provide practitioners with an efficient and interpretable model of multichannel profile data in high-dimensional spaces to support the diagnosis and for finding root cause. The approach is based on unsupervised classification of multichannel profile data provided by several sensors during a fault. The proposed approach is schematically represented by Fig. 2.

For each channel, a profile refers to a specific variable sampled in a time-window of 2 s with a sample rate of 100 Hz. The sampled time-windows for each channel were based on a λ-undershoot event, which was triggered by a value of the λ_{US-NSC} (channel No. 12) below an acceptability threshold. In particular, the center point of each window of 2 s, was set equal to the instant in which a λ-undershoot event

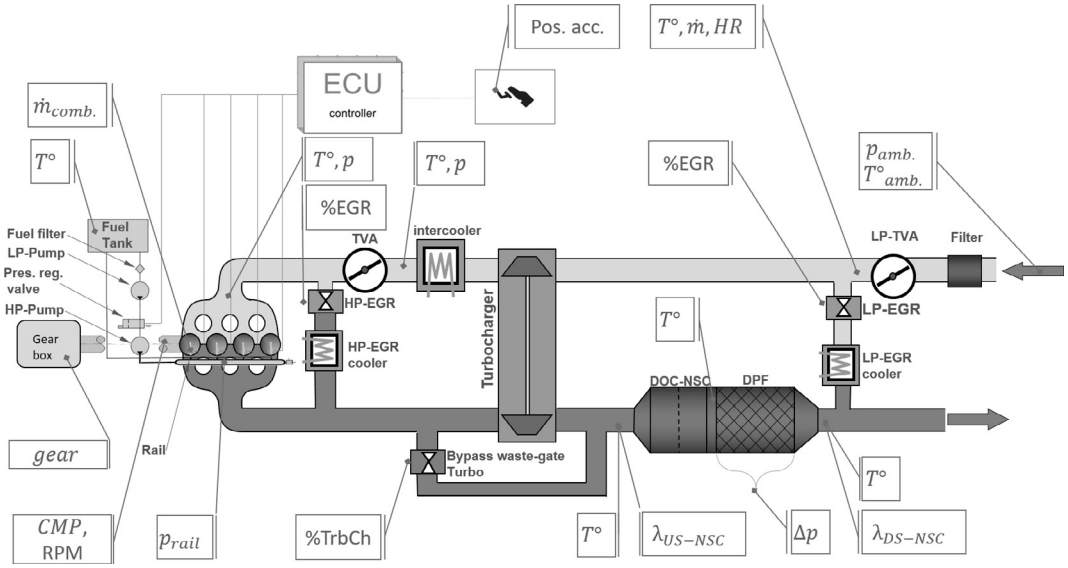

Fig. 1. 20 on-board sensors of a diesel engine monitoring the combustion (pipeline in light gray) and the exhaust gas aftertreatment process (pipeline in dark gray).

Table 1
List of channels. For each channel is reported the description, processing label and unit.

#	Description	Label	Unit
1	air aspirated per cylinder	Ch01	[mg/s]
2	accelerator pedal position	Ch02	–
3	aspirated air temperature	Ch03	[°C]
4	air inlet temperature	Ch04	[°C]
5	low pressure EGR valve	Ch05	–
6	high pressure EGR valve	Ch06	–
7	environmental pressure	Ch07	[mbar]
8	environmental temperature	Ch08	[°C]
9	engine rotational speed	Ch09	[rpm]
10	fuel in the 2nd pre-injection	Ch10	[mg/s]
11	total quantity of fuel injected	Ch11	[mg/s]
12	lambda upstream NSC (λ_{US-NSC})	Ch12	–
13	downstream intercooler pressure	Ch13	[mbar]
14	inner torque	Ch14	[Nm]
15	rail pressure	Ch15	[mbar]
16	aperture ratio of inlet valve	Ch16	–
17	gear index	Ch17	–
18	turbo charger position	Ch18	–
19	swirl valve	Ch19	–
20	vehicle velocity	Ch20	[km/h]

occurred (one second before and one after the fault event). The number of λ-undershoot events (i.e. samples) observed in our experiment was 342.

The number of data points in each profile was set equal to 203. Therefore, the data set can be represented by a third-order tensor object $\mathscr{X} \in \mathbb{R}^{D\times N\times M}$ where $D = 203$, $N = 20$ and $M = 342$. Profiles for each channel was linearly scaled into the range [0, 1] and centered by subtracting the average profile of the channel, say $\mathscr{X}^c_{[0,1]} \in \mathbb{R}^{203\times 20\times 342}$ the resulting tensor object. This tensor object was analyzed by VPCA, MPCA and UMPCA.

It should be noted that in terms of between-profile variability, the stationarity assumption (at least in a weak sense) is required for VPCA, MPCA and UMPCA modeling. In terms of within-profile variability, PCA-based approaches can also be applied if the stationarity assumption is violated (although in this case the noise within a profile may result in a lower performance of the approach). In this work, prior to analyzing the multichannel profile data, each profile sample was denoised using low-pass filtering.

5.2. Dimensionality reduction by means of PCA

Both the VPCA and MPCA were performed for different values of variability explained (say ψ) that is a measure of the information ratio

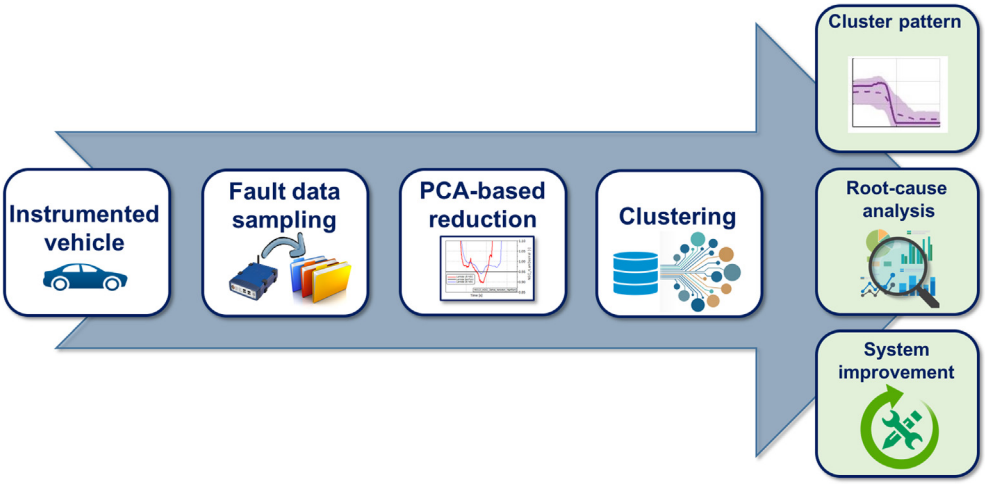

Fig. 2. The approach proposed in this work.

Table 2
Results of the PCA-based techniques on the dateset. Variability explained (ψ), number of retained features (P') and model MSE.

Dataset Original Dimension $D \times N \times M$	PCA based technique	Variability Explained $\psi(\%)$	No. Features Retained P'	Model *MSE*
		97	13	1.285E − 03
		98	17	1.075E − 03
	VPCA	99	26	7.715E − 04
		99.9	98	2.688E − 04
		99.99	179	8.918E − 05
		99.999	288	2.464E − 05
203 × 20 × 342				
		97	27	1.301E − 03
		98	36	1.185E − 03
	MPCA	99	55	8.302E − 04
		99.9	210	2.700E − 04
		99.99	1037	7.204E − 05
		99.999	2772	2.127E − 05
	UMPCA	50.437	20	5.809E − 03

from the processed data set respect the original one (in %). Values of ψ ranging between 97% and 99.999% were considered in the analysis. The variability explained by the model influences the number of retained features P' (the greater ψ is, the greater P' is). In the case of VPCA, the number of retained features in the model, which are uncorrelated, is limited to the maximum tensor mode dimension (corresponding to sample number $M = 342$ in the reference case study). In case of MPCA, the number retained features, which may be correlated, is limited to the product of $D = 203$ (datapoins of each profile) and $N = 20$ (number of channels), i.e. $203 \times 20 = 4060$ features. In the case of UMPCA, P' is limited by the minimum tensor mode dimension (which corresponds to the number of channels $N = 20$ in the reference case study).

Results of the PCA-based techniques on the dateset are summarized in following Table 2, where for each level of variability explained ψ, the number of features retained P' is showed along with the mean squared error (*MSE*) of the model. The MSE of the model was computed by comparing the original tensor data set $\mathscr{X}^{c}_{[0,1]}$ to the approximation tensor data set $\widetilde{\mathscr{X}}^{c}_{[0,1]}$ obtained from the PCA-based model.

From Table 2, it can be noted that given a value of variability explained, the MSE of VPCA and MPCA models are comparable. Furthermore, while VPCA and MPCA allows the model to explain up to 99.999% of variability in the original of the original data set, with MSE of about 2E − 05 and a number of retained features equal to 288 and 2772 respectively, UMPCA can explain no more than 50.437% of variability (MSE equal to 5.8E − 03) since the retained features, which are uncorrelated, are limited to 20. By comparing MPCA to VPCA for a given value of variability explained, it can be observed that the number of retained features is greater than that of the VPCA. This can be interpreted by considering that the MPCA features may be statistically correlated, so they may share some common information of the original variation of data.

5.3. Unsupervised classification validation

For each model and for each value of the variability explained, unsupervised classification was applied on the retained features of each sample. The results of unsupervised classification are reported in following Table 3, where K is the optimal cluster number obtained by the *Kneedle* procedure implemented in order to minimize the within-cluster sum of square $SSW(K)$, which measures the compactness of the clustering. In the same table, the clustering execution time has been reported for each case.

By considering the results of Table 3, it can be observed that the high number of features retained in the MPCA model has a great impact

Table 3
Unsupervised classification. Optimal number of clusters (K) and execution time.

PCA based technique	P'	Clustering Exc. time	Clustering K
$VPCA_{\psi\approx 97}$	13	00 h 11 min 27 s	27
$VPCA_{\psi\approx 98}$	17	00 h 15 min 13 s	19
$VPCA_{\psi\approx 99}$	26	00 h 11 min 26 s	23
$VPCA_{\psi\approx 99.9}$	98	00 h 31 min 46 s	23
$VPCA_{\psi\approx 99.99}$	179	00 h 56 min 21 s	20
$VPCA_{\psi\approx 99.999}$	288	01 h 34 min 56 s	26
$MPCA_{\psi\approx 97}$	27	00 h 11 min 57 s	11
$MPCA_{\psi\approx 98}$	36	00 h 16 min 51 s	8
$MPCA_{\psi\approx 99}$	55	00 h 25 min 34 s	8
$MPCA_{\psi\approx 99.9}$	210	01 h 21 min 19 s	7
$MPCA_{\psi\approx 99.99}$	1037	05 h 47 min 57 s	7
$MPCA_{\psi\approx 99.999}$	2772	14 h 24 min 20 s	7
UMPCA	20	00 h 10 min 52 s	13

on the clustering computational time. MPCA presented longer processing time when compared to VPCA for the same value of variability explained. Furthermore, it can be observed that the optimal number of clusters K is influenced by the PCA-based model implemented. In the case of VPCA, the number of clusters ranges between 19 and 27. By using MPCA, the number of clusters is lesser than that of VPCA, for each value of the explained variability. In particular, it ranges between 7 and 11 showing that MPCA allows to obtain the minimum within-cluster sum of squares in a smaller number of clusters. It is worth noting that more clusters produced by VPCA implies that the clusters are smaller, which makes the number of data points in the clusters smaller, while in the case of MPCA the number of data points in the clusters are greater. Since clustering was implemented in order to achieve an optimal compactness, it is fairly to conclude that by MPCA the retained features are more representative of the original data set.

Internal validation measures were used to assess the quality of the clustering they produced. Clustering validation results based on the three internal measures reported in Section 4.1 is summarized in Table 4. From Table 4, it can be observed that the MPCA approach leads to the best results of Caliński-Harabasz (*CH*), Davies-Bouldin (*DB*) and Silhouette (*S*) indexes. In particular, the clustering validation measures show that by using the MPCA approach the result of classification of multichannel samples achieves a better within-cluster dispersion along with a better between-cluster separation. As result, the use of MPCA

Table 4
Clustering validation.

PCA based Techniques	Internal Index *CH*	*DB*	*S*
$VPCA_{\psi\approx 97}$	52.98	1.1171	0.4148
$VPCA_{\psi\approx 98}$	54.00	1.2615	0.3523
$VPCA_{\psi\approx 99}$	49.53	1.2304	0.3560
$VPCA_{\psi\approx 99.9}$	47.52	1.3074	0.3516
$VPCA_{\psi\approx 99.99}$	49.73	1.3995	0.3402
$VPCA_{\psi\approx 99.999}$	46.55	1.2380	0.3663
$MPCA_{\psi\approx 97}$	212.92	1.1598	0.4860
$MPCA_{\psi\approx 98}$	315.40	1.0185	0.5155
$MPCA_{\psi\approx 99}$	353.23	1.1952	0.4656
$MPCA_{\psi\approx 99.9}$	664.75	0.8814	0.5889
$MPCA_{\psi\approx 99.99}$	776.50	0.8065	0.6203
$MPCA_{\psi\approx 99.999}$	782.06	0.8028	0.6216
UMPCA	201.10	0.8590	0.4993

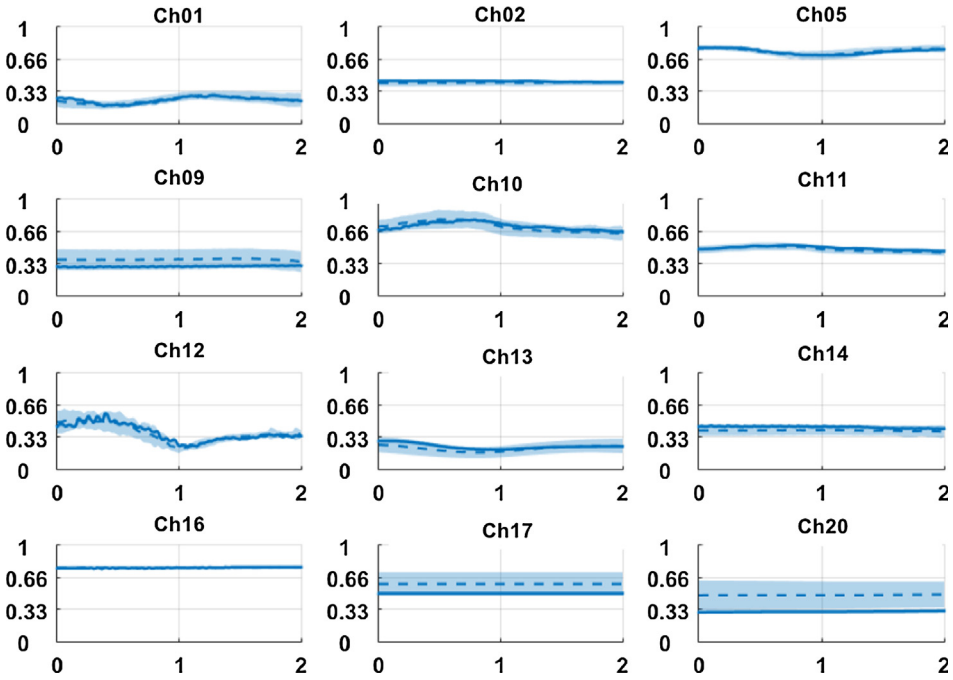

Fig. 3. 12 channels for "Cluster A". Each panel depicts the centroid profile (continues line) the average profile (dashed line) the variability area.

allows to identify more interesting structure in the data, while in contrast, the PCs obtained from VPCA are not correct features for separating fault samples in groups. Finally, results in Table 4 show that although UMPCA produces a number of uncorrelated features limited by the number of channels in the data set, and hence it is unable to model to an arbitrary level of detail, the method leads to a better clustering compactness and separation when compared to VPCA.

5.4. An example of clusters assessment

Once multichannel profile data have been partitioned into clusters (scenarios), practitioners analyze cluster patterns to get more insight of data for fault diagnosis. Fig. 3 depicts 12 out of the 20 channels in Table 1 for a specific cluster (named "Cluster A") resulting from the analysis of experimental multichannel profile data. For each channel, the centroid profile (continuous line), the mean profile (dashed line), and the variability area around the mean profile, are reported in the same figure. Each channel refers to a time-window of 2 s. Moreover, as the data set is composed by heterogeneous data, each channel is linearly scaled into the range [0, 1]. Fig. 4 depicts similar information for a different cluster of experimental multichannel profile data (named "Cluster B").

By evaluating the shape of centroids and mean profiles in each channel of the clusters useful information can be obtained for detecting fault root cause. In particular, from Figs. 3 and 4, it can be observed that while in "Cluster A" the velocity is quite constant (`Ch20`), there is no change in the accelerator pedal position (`Ch02`) or in the gear index (`Ch17`), "Cluster B" clearly refers to an acceleration phase during the vehicle driving. As consequence, the fluctuations of the intake air mass (`Ch01`) during the NSC regeneration are more relevant in cluster of Fig. 4 than in the cluster of Fig. 3. Additionally, from Fig. 4 it can be observed in "Cluster B" that λ_{US-NSC} value (`Ch12`) presents an increasing trend in the final part of the time window. This indicates the ending of the NSC regeneration phase, which is due to changes in the engine operating conditions. In fact, in "Cluster B" it can be also noted not only an increase of air mass per cylinder (`Ch01`) but also changes in the aperture ratios of inlet valve (a.k.a. throttle valve, labeled as `Ch16`), EGR valve (`Ch05`) along with a reduction of the amount of injected fuel charge in the second post-injection (`Ch10`).

6. Conclusions

The emission control issue has assumed an important role in industrial engineering, as well as in transportation where energy is still provided by internal combustion engines. In order to fulfill legislation requirements on emissions, a high number of on-board sensors are introduced in order to monitor both combustion and exhaust gas aftertreatment process. In this scenario, large sets of real-time multichannel profile data need to be analyzed.

In this study, two multilinear extensions of PCA, which can analyze the multichannel profiles without unfolding the original data set, have been investigated with application to an emission control system.

By comparing the MPCA to regular PCA on vectors generated by unfolding the original multichannel profile data set (VPCA), the MPCA appeared to be a more efficient method as it can capture the cross-correlation among different signals and important variation patterns more effectively than VPCA. The MPCA results in a multilinear projection that projects the original tensor objects into a lower-dimensional tensor subspace while preserving the variation in the original data. This is especially important when an high number of data points or channels are involved or in the presence of more complex data structures.

Clustering validation based on three internal indexes (*CH*, *DB* and *S*) shows that unsupervised classification may also benefit of the MPCA method. In fact, MPCA may lead to obtain a given within-cluster dispersion with a better between-cluster separation. In practice, using MPCA may allow to identify more interesting structure in the data, while in contrast, VPCA appears to be unable to extract correct features for separating data in clusters. UMPCA, which produces uncorrelated features, is unable to model to an arbitrary level of detail, because the number of extracted features is limited by the number of channels. Nevertheless, when compared to VPCA, UMPCA leads to better performance in unsupervised classification.

As a summary, the topic of classification of multichannel profile data via the multilinear dimension reduction techniques applicable to tensor data, such is MPCA and UMPCA, appeared to be a promising field of research for the future. Considering that the approach proposed in this paper is quite general, one direction of future research consists in testing and comparing the approach to different existing procedures for unsupervised classification. Furthermore, as fault diagnosis and root

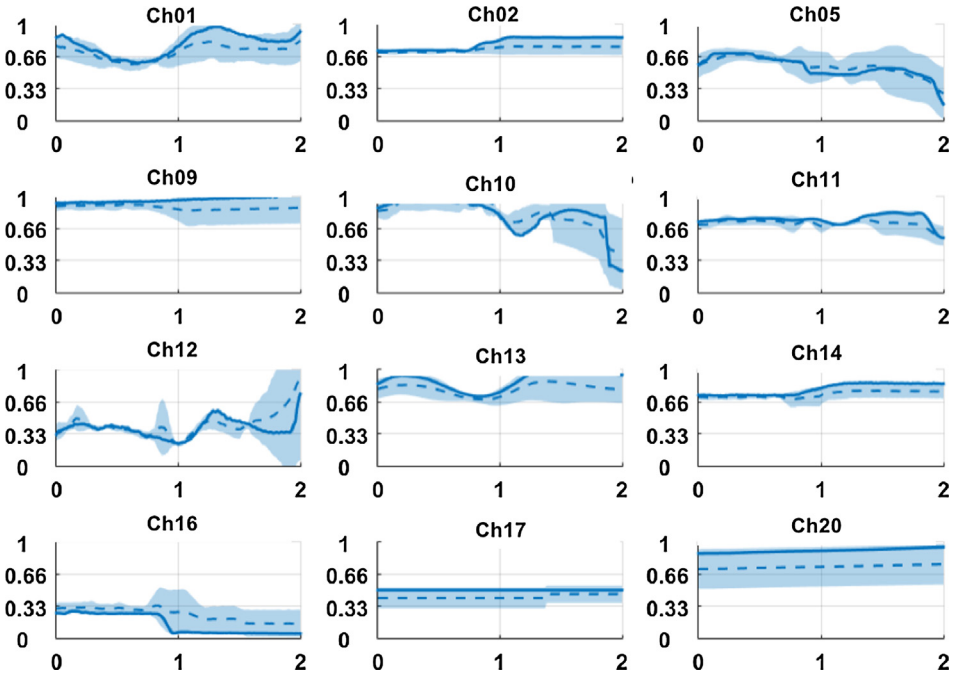

Fig. 4. 12 channels for "Cluster B". Each panel depicts the centroid profile (continues line) the average profile (dashed line) the variability area.

cause identification is an important task in several application fields, extensions of the proposed approach to supervised classification is an important, yet challenging research topic that deserves further study.

Acknowledgements

This research is supported by Ministry of Education, University and Research of Italy (MIUR). The author also thanks the editor and referees for their insightful comments.

References

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, SODA '07* (pp. 1027–1035). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Bader, B. W., & Kolda, T. G. (2006). Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software, 32*(4), 635–653.

Boutsidis, C., Drineas, P., & Mahoney, M. W. (2009). Unsupervised feature selection for the k-means clustering problem. In *Advances in neural information processing systems.* (Vol. 22, pp. 153–161). Curran Associates, Inc.

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics, 3*(1), 1–27.

Cox, T. F., & Cox, M. A. (1994). *Multi-dimensionalscaling.* Chapman and Hall.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1,* (2), 224–227.

Grasso, M., Colosimo, B., & Pacella, M. (2014). Profile monitoring via sensor fusion: The use of PCA methods for multi-channel data. *International Journal of Production Research, 52*(20), 6110–6135.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks, 13*(4), 411–430.

Isermann, R., & Ball, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice, 5*(5), 709–719.

Jiang, Q., Jia, M., Hu, J., & Xu, F. (2009). Machinery fault diagnosis using supervised manifold learning. *Mechanical Systems and Signal Processing, 23*(7), 2301–2311.

Jolliffe, I. T. (2002). *Principal component analysis. Springer Series in Statistics.* New York: Springer.

Kolda, T. G. (2001). Orthogonal tensor decompositions. *SIAM Journal of Matrix Analysis and Applications, 23*(1), 243–255.

Lathauwer, L. D., Lathauwer, J. V. D., Moor, B. D., & Vandewalle, J. (2000a). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications, 21*(4), 1253–1278.

Lathauwer, L. D., Lathauwer, J. V. D., Moor, B. D., & Vandewalle, J. (2000b). On the best rank-1 and rank-(R_1, R_2, ...,R_N) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications, 21*(4), 1324–1342.

Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J., & Wu, S. (2013). Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics, 43*(3), 982–994.

Louwerse, D., & Smilde, A. (2000). Multivariate statistical process control of batch processes based on three-way models. *Chemical Engineering Science, 55*(7), 1225–1235.

Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2008). MPCA: Multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks, 19*(1), 18–39.

Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2009). Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning. *IEEE Transactions on Neural Networks, 20*(11), 1820–1836.

Masoud, H. I., Zerehsaz, Y., & Jin, J. J. (2017). Analysis of human motion variation patterns using UMPCA. *Applied Ergonomics, 59*(6), 401–409.

Pacella, M., & Colosimo, B. M. (2016). Multilinear principal component analysis for statistical modeling of cylindrical surfaces: A case study. *Quality Technology & Quantitative Management* 1–19 (in press, doi: 10.1080/16843703.2016.1226710).

Paynabar, K., Jin, J., & Pacella, M. (2013). Monitoring and diagnosis of multichannel nonlinear profile variations using uncorrelated multilinear principal component analysis. *IIE Transactions, 45*(11), 1235–1247.

Paynabar, K., Qiu, P., & Zou, C. (2016). A change point approach for phase-I analysis in multivariate profile monitoring and diagnosis. *Technometrics, 58*(2), 191–204.

Rato, T. J., Rendall, R., Gomes, V., Chin, S.-T., Chiang, L. H., Saraiva, P. M., et al. (2016). A systematic methodology for comparing batch process monitoring methods: Part I-assessing detection strength. *Industrial & Engineering Chemistry Research, 55*(18), 5342–5358.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*(1), 53–65.

Salehi, M., Jalalian, M., & Siar, M. M. V. (2017). Green transportation scheduling with speed control: Trade-off between total transportation cost and carbon emission. *Computers & Industrial Engineering, 113*, 392–404.

Satopaa, V., Albrecht, J., Irwin, D., & Raghavan, B. (2011), Finding a Kneedle in a Haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops* (pp. 166–171).

Smilde, A., Bro, R., & Geladi, P. (2004). *Multi-way analysis with applications in the chemical sciences.* Chichester-UK: Wiley.

Soliman, A., Rizzoni, G., & Kim, Y. (1999). Diagnosis of an automotive emission control system using fuzzy inference. *Control Engineering Practice, 7*(2), 209–216.

Wang, Y.-H. T., & Huang, W.-H. (2017). Phase II monitoring and diagnosis of auto-correlated simple linear profiles. *Computers & Industrial Engineering, 112*, 57–70.

Yan, H., Paynabar, K., & Shi, J. (2015). Image-based process monitoring using low-rank tensor decomposition. *IEEE Transactions on Automation Science and Engineering, 12*(1), 216–227.

Zhang, J., Ren, H., Yao, R., Zou, C., & Wang, Z. (2015). Phase I analysis of multivariate profiles based on regression adjustment. *Computers & Industrial Engineering, 85*, 132–144.