Journal of Electronic Research and Application

Research Article

Multiple Profiles Sensor-based Engineering Data Processing

Tianlu Xu

The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract: Nowadays big data is widely adopted in industry field. In an advanced manufacturing system hundreds of sensors are deployed to collect key variables for system performance and the real-time data would be used for further monitoring and anomaly detection. However, there are many challenges for applying the sensor-based data directly, including the profile data has unsynchronized different length for different samples, the existence of obvious long-term drift, strong correlation of sensor clusters and the particular feature extraction. To solve these problems this invention presents a multiple profiles sensor-based engineering data processing system, including (1) preprocessing the signals to align the data and remove long-term drift, (2) clustering the sensors which have strong correlations, and (3) extracting particular features from different sensor clusters.

Keywords: *Big data, Sensor-based, Multiple profiles*

Publication date: May, 2019

Publication online: 31 May 2019

Corresponding author: Tianlu Xu, liaoquanneng@ivygate.cn

1 Introduction

1.1 Background of invention

Industrial big data refers to big data generated in industrialized information applications. The connection of Information technology and global industrial systems bring profound changes to global industry and innovate new patterns of enterprise production, operation, marketing and management. Nowadays big data is widely adopted in an industry system to test and evaluate the manufacturing system performance.

Hundreds of sensors are placed in the manufacturing system for real-time data collection. The high-dimension data stream generated from sensors will be further used for monitoring and anomaly detection.

However, there are many challenges for developing such an online data processing system, including the profile length varies from sample to sample due to the self-controlled process, data profiles have long-term drift which needed to be removal and its magnitude varies in different time segments, how to classify different sensors to find strongly correlated sensor clusters, extract highly heterogeneous sensor features from particular sensor clusters for further anomaly and detection.

To address these challenges, this invention presents a real-time multiple sensor-based data processing system, to solve synchronization problem we do data alignment for different samples and calculating the signal correlation of different time points for every sensor. However, even for the same sensor profile, its long-term drift magnitude varies in different time segments. It's necessary for estimating long-term drifts for different time segments separately and removing the long-time drift. Many sensors show strongly within-profile correlation, it motivates us to classify the sensors into sensor clusters according to the data correlation of them for further exploration. Applying multichannel functional principle component analysis(MFPCA) related methods to extract features from particular sensor clusters could make us convenient for further anomaly detection. (Figure 1)

2 Summary

In order to preprocess the raw sensor signals to remove these system inherent variations and extract particular features from sensor clusters for accurate comparison and analysis, this invention proposes an engineering

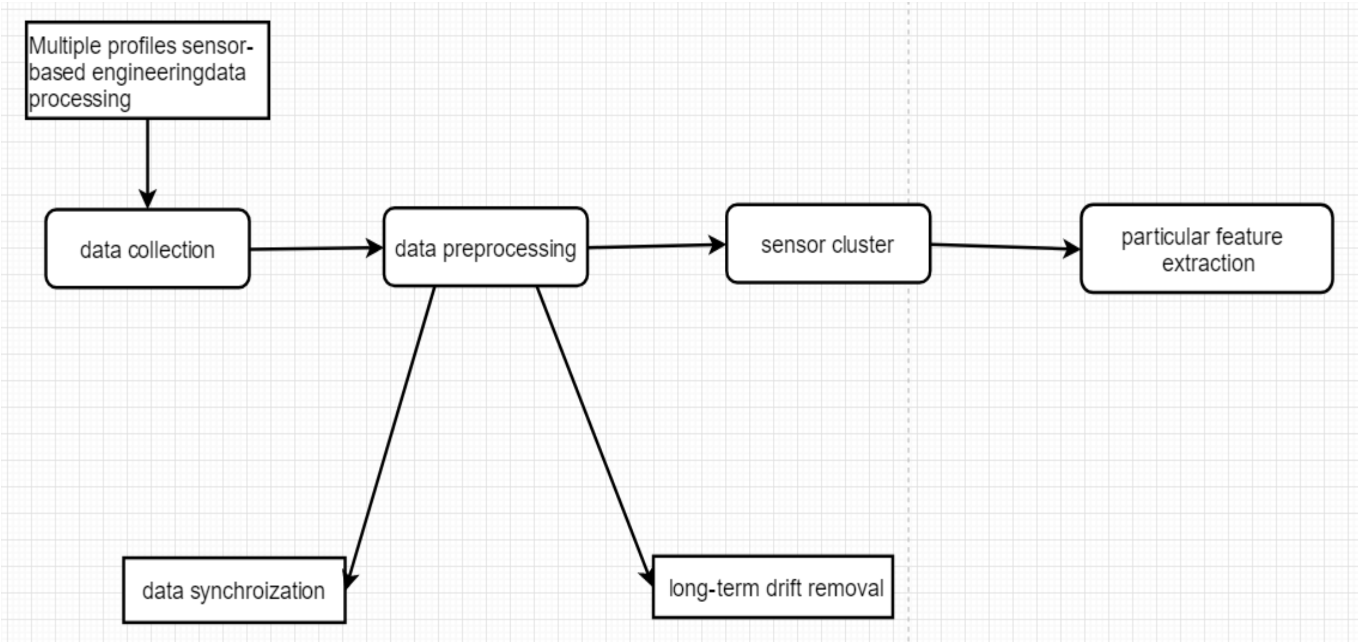


Figure1. The structure of the data processing system

data analyzing system which could align the data generated by different sensors, remove long-term drift, classify sensors into groups and extract particular features for further analyze.

In order to uniform the profile data length and align them we would like to use Dynamic Time Warping technique[1]. This method is applied to synchronize the time stamp. The sensor signal is represented as Y_{ijt},t $=1,...,T_i$, and the reference sample as Y_{1jt},t $=1,...,T_1$. The data alignment can be processed creating a T_1-by-T_i distance matrix, where (t_1, t_i) denotes the square distance, using Dynamic Time Wrapping method to determine the optimal path $(t_{1l}, t_{il})_{l=1,...,L}$ which gives the lowest warping cost(Figure 2 and 3). Use Sakoe-Chiba band constraint to limit the search space and determine the optimal path, it would be wise to write a program and let computer do the complex calculations. Finally, the optimal path represented as $(t_{1l}, t_{il})_{l=1,...,L}$ indicates a mapping from the i^{th} sample to the reference sample, which minimize the lowest wrapping cost.

Generally the long-term drift depends on the system

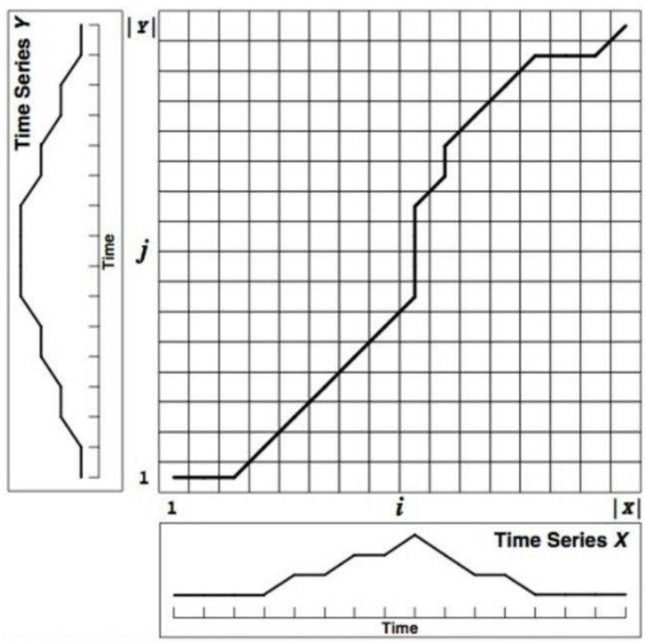


Figure 2. Dynamic time wrapping

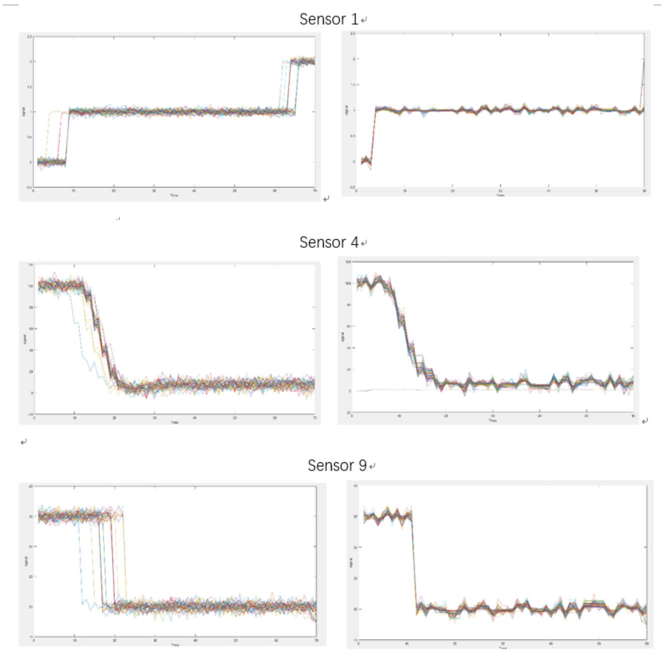


Figure 3. Data Alignment

status which is influenced by the "on-off" operations and even for the same sensor profile its magnitude varies in different time segments. To remove the long-time drift we would like to estimate long-term drift for different time segments separately. Also the "on-off" operations has an obvious effect on system stability. Based on it, the time points between two switching operations could be regarded as a nature segment. Because the time length T is equal for all samples after synchronization, these T profiles can be divided into M steps, with Y_{ijt} represented as

$$Y_{ij} = \mu_{jt} + b_{ijm} + \epsilon_{ijt}$$

Where m is the corresponding segment to which t^{th} profile points belongs. In the equation, μ_{jt} is the templet signal of the j^{th} sensor at the t^{th} time point. b_{ijm} is considered as the constant drift of the j^{th} variable in the m^{th} time segment of the i^{th} sample. ϵ_{ijt} represents the remaining individual signal from t=1 to T with the mean 0 and variance σ^2_{jt}. According to Lee et al.[2], μ_{jt} and b_{ijm} can be calculated by minimizing the weighted sum of squared errors based on weighted least squares algorithm.

$$\min_{\mu_{jt}, b_{ijm}} WWSE_{jm} = \sum_{i=1}^{N} \sum_{t=t_{m-1}+1}^{t_m} \frac{1}{\sigma^2_{jt}} (Y_{ijt} - \mu_{jt} - b_{ijm})^2$$

$$\text{s.t.} \sum_{i=1}^{N} b_{ijm} = 0$$

for j=1 to P and m=1 to M. However, one thing worth noticing is that the $\hat{b}_{ijm}$ is not always suitable to be the long-term drift because a drift should always be monotone while in this case $\hat{b}_{ijm}$ is purely the difference of the i^{th} sample from the template profile in the m^{th} segment and cannot be guaranteed to be monotone. To solve this problem, a threshold s_{jm} is set. For each time the cumulative drift magnitude between neighbor samples $|\hat{b}_{Njm} - \hat{b}_{1jm}|$ is compared with the threshold. If it is greater than the threshold the $\hat{b}_{ijm}$ is considered valuable. Otherwise, the $\hat{b}_{ijm}$ is set to zero which suggests that the difference is negligible.

$$\tilde{b}_{ijm} = \begin{cases} \hat{b}_{ijm}, & if |\hat{b}_{Njm} - \hat{b}_{1jm}| > s_{jm} \\ 0, & otherwise \end{cases}$$

The s_{jm} is set to $3 * \text{std}(\hat{b}_{Njm})$ as it gives the best performance in practice.

At the very beginning, the σ^2_{jt} can be calculated by $\sigma^2_{jt} = \sum_{i=1}^{N} \frac{(Y_{ijt} - \hat{\mu}_{jt} - \tilde{b}_{ijm})^2}{N}$. However, after several times of "on-off" operation, the profile is not changed significantly. Hence, these serial segments can be combined as one to simplify analysis. The segments may be represented in another way when taking $\{t_m | \hat{\sigma}^2_{jt_m} > s\}$ as the segment boundaries. Here s is a pre-defined threshold with the value of $3 * \text{median}(\hat{\sigma}^2_{jt})$. Finally, $\hat{b}_{ijm}$, $\tilde{b}_{ijm}$, $\hat{\sigma}^2_{jt}$ and t_m can be calculated via iterating the procedures where M is flexible during iterations(Figure 4 and 5).

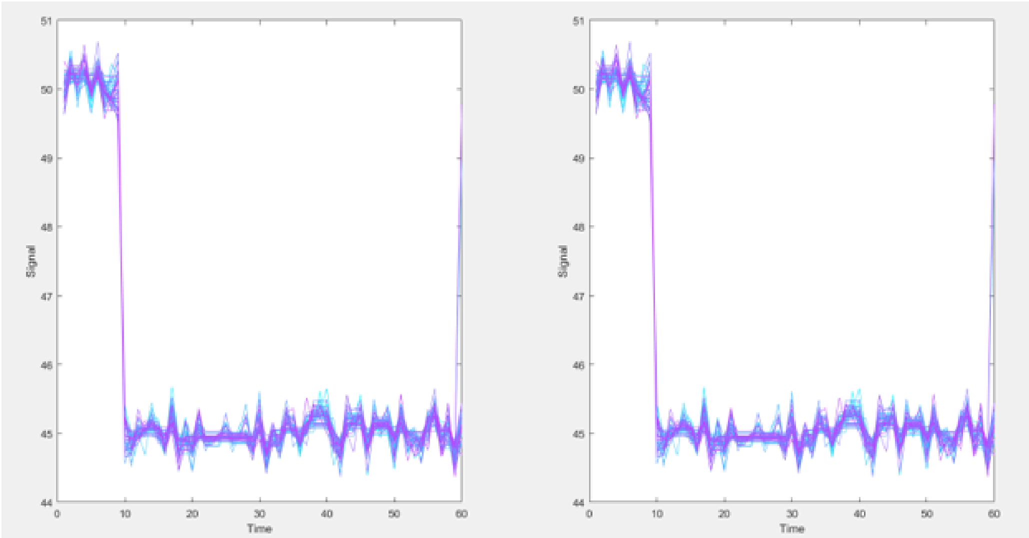
Different sensors have strong within-profile correlations hence its convenient to classify them into sensor clusters for further analyze. As we mentioned before the sensor profiles can be naturally clustered according to the sensor correlation matrix. Therefore, agglomerative hierarchical correlation clustering method could be used classification. Firstly classify each of the sensor as a cluster by itself and then merges them into larger clusters according to the sensor correlations. In each step of hierarchical clustering, it finds the closest pair of clusters and then merges them into a new parent cluster. It is repeated until only one cluster is formed after N−1 iterations where N is the number of sensors. Here the Pearson's correlation is used to measure the similarity between different sensors. Then the distance (dissimilarity) between different sensors can be defined. We can see that the more correlated the two sensors are, the shorter their distance will be. In addition to the distance between two individual sensor signals, cophenetic distance with the average linkage is used to measure the between-cluster distance. In every step, according to the distance of the sensors we can select two most similar clusters and merge them into one. Dendrogram can be used to draw such sequential merging procedure and based on the diagram we can finally get the sensor information of every cluster. And by reordering the sensors, the strong within-cluster correlations and slight between-cluster correlations can be clearly shown. (Figure 6, 7, 8, 9, 10)

3 Description of preferred embodiment

3.1 Data alignment

Since there is self-controlled process when manufacturing, the profile data of every sample has various lengths without synchronization. We use Dynamic Time Warping[1]. This method can help to synchronize the time stamp. The sensor signal is represented as Y_{ijt} , t = 1,...,T_i, and the reference sample as Y_{1jt},t = 1,...,T_1. The data alignment can be done by doing following procedures.

Sensor 15



Sensor 32

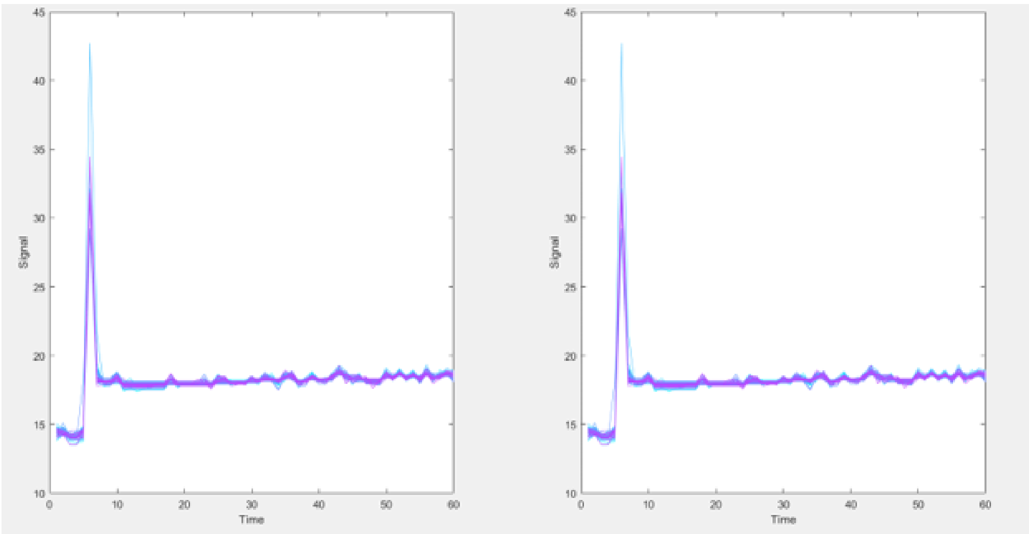
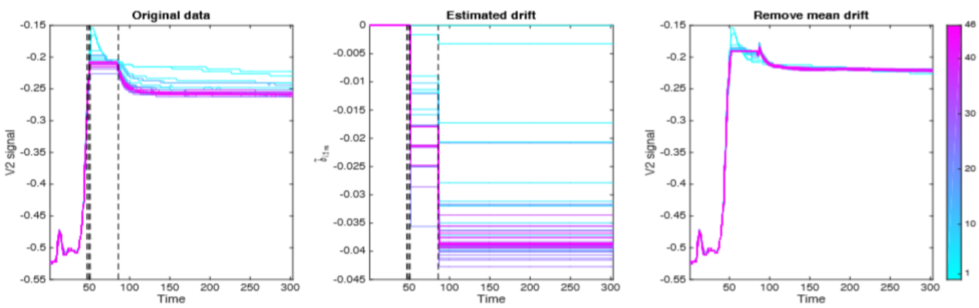


Figure 4. Long-term drift



(a) Original data (b) Estimated drift (c) Data after removing the drift

Figure 5. Long-term drift from another sample

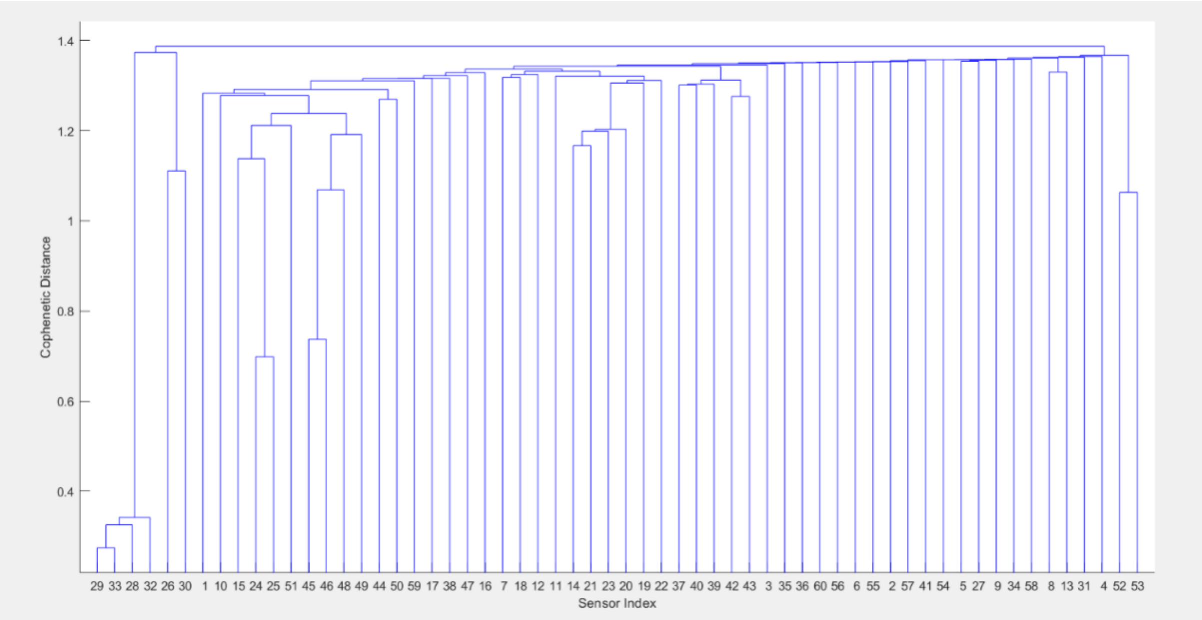


Figure 6. Hierarchical Clustering for sensors

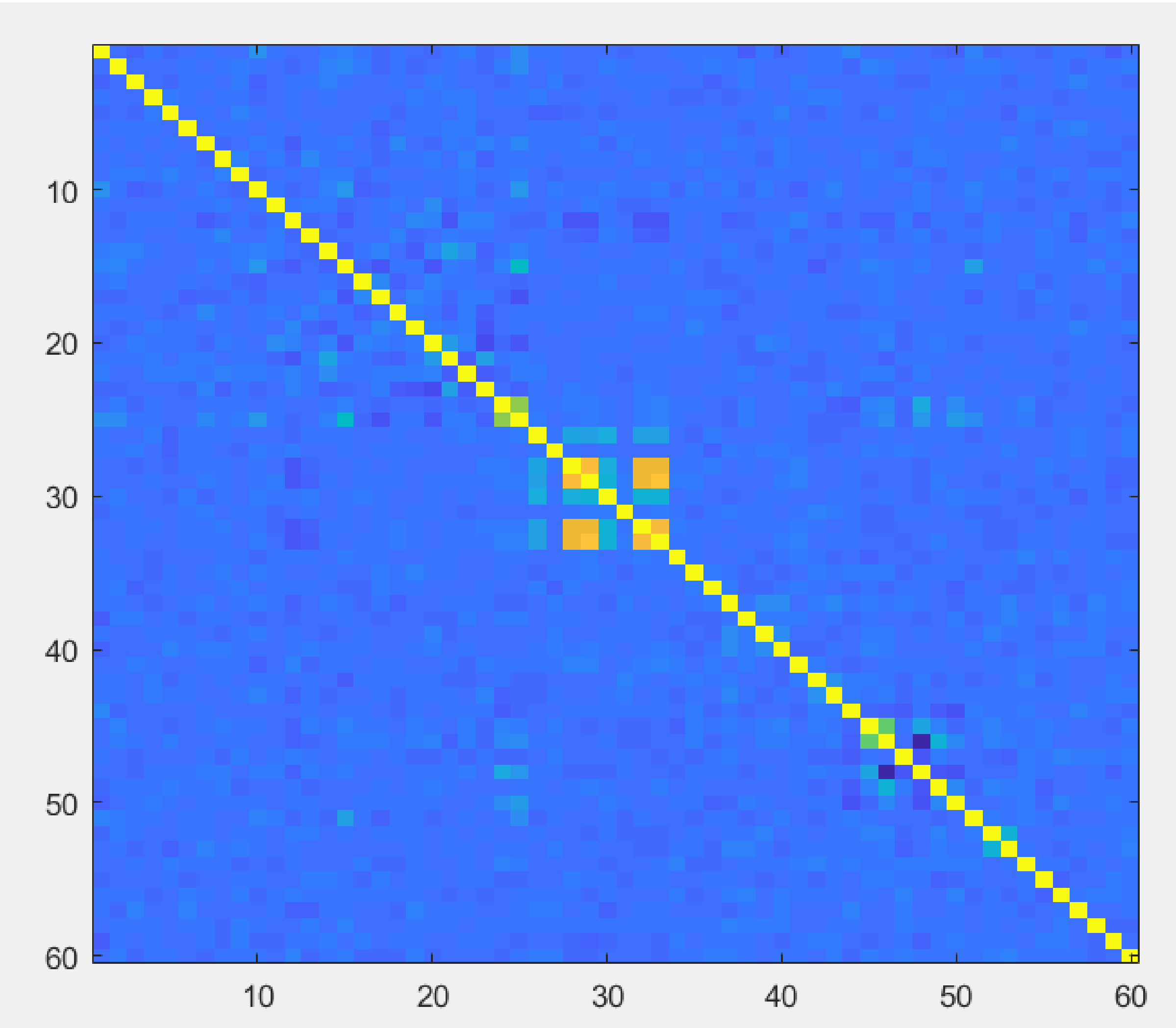


Figure 7. The correlation structure of reordered sensors based on the clustering result

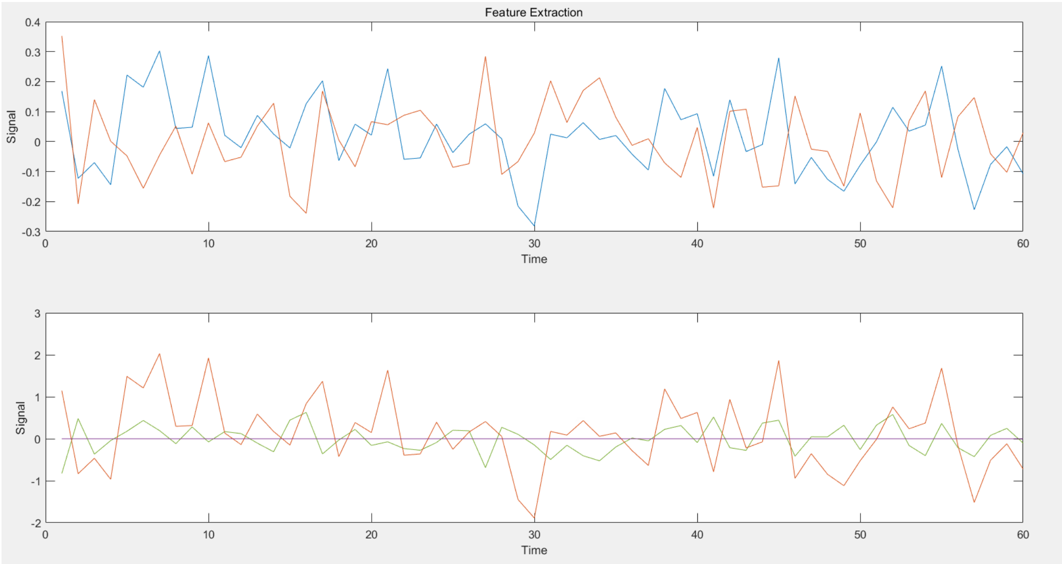


Figure 8. Features extraction from cluster1 sensor20

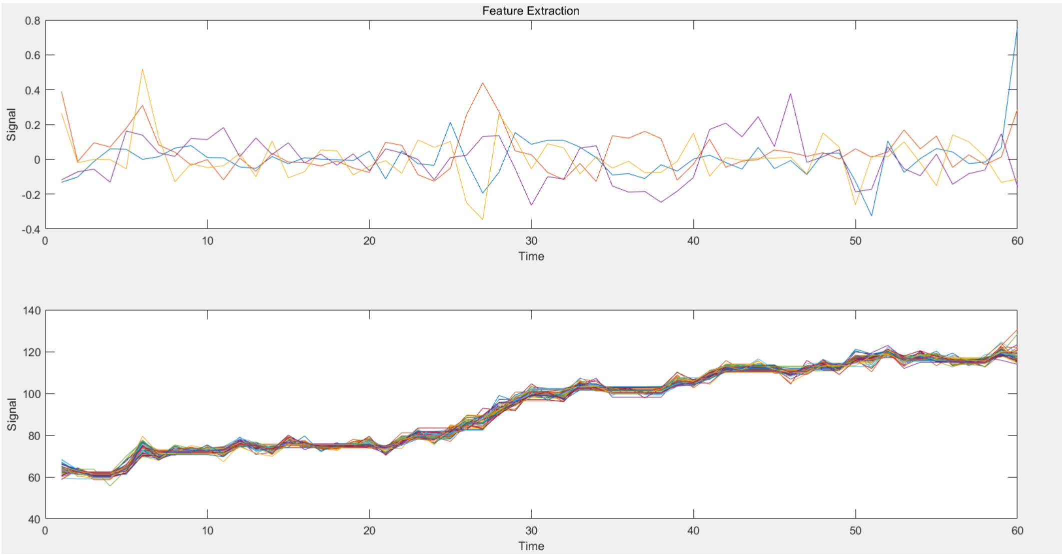


Figure 9. Features extraction from cluster2 sensor51

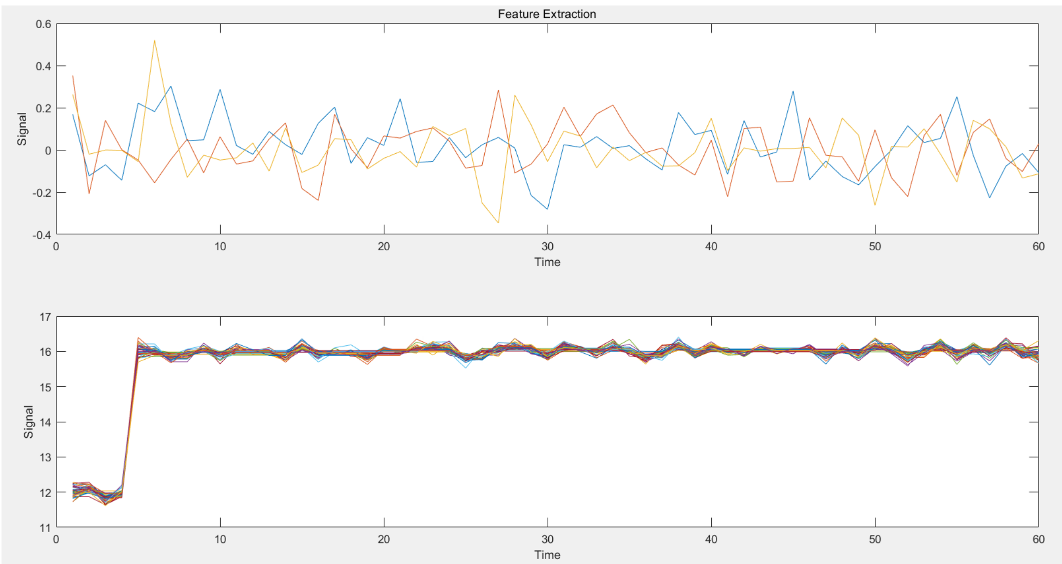


Figure 10. Features extraction from cluster31 sensor47

(1)Create a T_1-by-T_i distance matrix, where (t_1,t_i) denotes the square distance:

$$d(t_1,t_i) = (Y_{1j(t_1)} - Y_{ij(t_i)})^2$$

(2)Use DTW to determine the optimal path $(t_{1l}, t_{il})_{l=1,\ldots,L}$ which gives the lowest warping cost.

$$\text{DTW}\left(Y_{1j(t_1)},Y_{ij(t_i)}\right) = \min \quad \{\sum_{l=1}^{L} d(t_{1l},t_{il})\}$$

(3)Use Sakoe-Chiba band constraint to limit the search space, which is

$$|t_{1l} - t_{il}| \leq \max(0.2T_1, 0.2T_i, |T_1 - T_i|), l = 1,\ldots,L.$$

To find the optimal path, it is recommended to write a program and let computer do the complex calculations. Finally, the optimal path represented as $(t_{1l},t_{il})_{l=1,\ldots,L}$ indicates a mapping from the i^{th} sample to the reference sample, which provides the lowest wrapping cost.

In this experiment we collect some figures which clearly show the difference between the original data and the data after alignment.

3.2 Long-term drift removal

In general, the Long-term Drift is mainly caused by unstable system status. In addition, the "on-off" operations has a non-negligible effect on system stability. Based on it, the time points between this two switching operations could be regarded as a nature segment. Because the time length T is equal for all samples after synchronization, these T profiles can be divided into M steps, with Y_{ijt} represented as

$$Y_{ijt} = \mu_{jt} + b_{ijm} + \epsilon_{ijt}$$

Where m is the corresponding segment to which t^{th} profile points belongs. In the equation, μ_{jt} is the templet signal of the j^{th} sensor at the t^{th} time point. b_{ijm} is considered as the constant drift of the j^{th} variable in the m^{th} time segment of the i^{th} sample. ϵ_{ijt} represents the remaining individual signal from t=1 to T with the mean 0 and variance σ^2_{jt}. According to Lee et al.[2], μ_{jt} and b_{ijm} can be calculated by minimizing the weighted sum of squared errors based on weighted least squares algorithm.

$$\min_{\mu_{jt},b_{ijm}} WWSE_{jm} = \sum_{i=1}^{N} \sum_{t=t_{m-1}+1}^{t_m} \frac{1}{\sigma^2_{jt}} (Y_{ijt} - \mu_{jt} - b_{ijm})^2$$

$$\text{s.t.} \sum_{i=1}^{N} b_{ijm} = 0$$

for j=1 to P and m=1 to M. However, one thing worth noticing is that the $\hat{b}_{ijm}$ is not always suitable to be the long-term drift because a drift should always be monotone while in this case $\hat{b}_{ijm}$ is just the difference of the i^{th} sample from the template profile in the m^{th} segment and cannot be guaranteed to be monotone. To solve this problem, a threshold s_{jm} is set. For each time the cumulative drift magnitude between neighbor samples $|\hat{b}_{Njm} - \hat{b}_{1jm}|$ is compared with the threshold. If it is greater than the threshold the $\hat{b}_{ijm}$ is considered valuable. Otherwise, the $\hat{b}_{ijm}$ is set to zero which suggests that the difference is negligible.

$$\tilde{b}_{ijm} = \begin{cases} \hat{b}_{ijm}, & if|\hat{b}_{Njm} - \hat{b}_{1jm}| > s_{jm} \\ 0, & otherwise \end{cases}$$

The s_{jm} is set to $3 * \text{std}(\hat{b}_{Njm})$ as it gives the best performance in practice.

At the very beginning, the σ^2_{jt} can be calculated by $\sigma^2_{jt} = \sum_{i=1}^{N} \frac{(Y_{ijt}-\hat{\mu}_{jt}-\tilde{b}_{ijm})^2}{N}$. However, after several times of "on-off" operation, the profile is not changed significantly. Hence, these serial segments can be combined as one to simplify analysis. The segments may be represented in another way when taking $\{t_m|\hat{\sigma}^2_{jt_m} > \text{s}\}$ as the segment boundaries. Here s is a pre-defined threshold with the value of $3 * \text{median}(\hat{\sigma}^2_{jt})$. Finally, $\hat{b}_{ijm}$, $\tilde{b}_{ijm}$, $\hat{\sigma}^2_{jt}$ and t_m can be calculated via iterating the procedures where M is flexible during iterations.

In our experiment there is no obvious long-term drift as figure shows hence we use a graph from another experiment. It is clearly to find the color difference between the original data and the data after removing the long-term drift.

3.3 Sensor cluster and feature extraction

Previously, we performed data alignment and canceled long-term drift on the data generated by all the sensors. At the same time, we can find that the sensors are related to each other. Therefore, in this part we classify the sensors and perform feature extraction to perform sensors by hierarchical classification. Correlation analysis of data sets to facilitate monitoring and anomaly analysis of data sets of each sensor in the later stage. We aggregate the data we collected before and construct SMFPCA for each collection. This part mainly contains the two aspects:

Firstly taking the data set generated by each sensor as a single cluster, and then merge them into a larger cluster based on the correlation between the sensors. In this way, we merge the similar classes of these sensors into each other until all the sensors are merged into the same cluster. This is what we call hierarchical clustering.

In the data analysis of the sensor, we can regard the data set as a continuous function $Y_{ij}(t)$, where the j^{th} data in the i^{th} sensor is produced for $t \in [a , b]$, which [a , b] The time interval, $Y_{ijt}(t=1, ..., T)$ is the time sampling grid point. The function can be expressed as $Y_{ij}(t)=\mu_i(t)+\varphi_{ij}(t)$, i=1...P, j=1...N, the first term in the expression is the function of the ith sensor The template, the latter term is a random error with respect to $E(\varphi_{ij}(t))=0$.

Through this function analysis we can see that there is correlation between the sensors, hence we can easily analyze the correlation between data sets using the multivariate SPC scheme. Meanwhile it is also important to consider the heterogeneous characteristics between the sensors, because different sensors may measure data of different process variables.

We propose an algorithm for sensor clustering and feature extraction to analyze cluster-like structures with sparse interphase between sensors. The main technical solution is to divide multiple sensors into multiple sensor clusters to ensure that the sensors in the same cluster have similar characteristics, and then cluster the relevant clusters into sub-clusters into the same cluster through multiple clusters, so that P sensors pass through. After multiple clustering, it becomes a cluster, and the correlation between sensors can be expressed as:

$$\rho_{ii'} = \frac{\sum_{j=-m_0+1}^{0}\sum_{t=1}^{T}\left(Y_{ijt}-\bar{Y}_{it}\right)\left(Y_{i'jt}-\bar{Y}_{i't}\right)}{\sqrt{\sum_{j=-m_0+1}^{0}\sum_{t=1}^{T}\left(Y_{ijt}-\bar{Y}_{it}\right)^2}\sqrt{\sum_{j=-m_0+1}^{0}\sum_{t=1}^{T}\left(Y_{i'jt}-\bar{Y}_{i't}\right)^2}}, i,i' = 1,\cdots,p$$

Define different sensors:

$$d_{ii'} = 1-\left|\rho_{ii'}\right|, i,i' = 1,\cdots,P$$

It can be seen that the distance between the two sensors decreases as the correlation increases. In addition to the distance between two individual sensor signals, cophenetic distance with the average linkage is used to measure the between-cluster distance of clusters r and k, where N_r and N_k are the number of sensors in clusters r and k. Then in every step, according to the distance of r and k we can select two most similar clusters and merge them in two one.

We also consider two other extreme scenarios with G=1 or G=60 for mere comparison, but not for recommendation. In particular, the chart with G=1 treats all sensors as one cluster, which is exactly the multichannel functional PCA (MFPCA) based monitoring scheme (Paynabar et al.2016) extended to Phase II. The chart with G=60 treats each single sensor as a cluster, which is the functional PCA (FPCA) based monitoring scheme for every profile channel separately(Yu et al. 2012). With regard to the number of R in the monitoring scheme, according to the engineering domain knowledge, we know that the maximum number of changed sensors is not larger than 40. Based on the dendrogram, we divide the 40 sensors into several clusters to ensure that the minimum correlation between two sensors in the same cluster is bigger than 1.2. And by reordering the sensors, the strong within-cluster correlations and slight between-cluster correlations can be clearly shown. Then we extract the particular features from certain clusters and when we meet new sensor profiles we could classify these signals into special clusters according to the features extracted before.

References

[1] Keogh E., Ratanamahatana C.A. "Exact Indexing of Dynamic Time Warping". Knowledge and Information Systems, 2005, 7(3):358–86.

[2] Tibshirani R., Walther G., Hastie T. "Estimating the Number of Clusters in a Dataset via the Gap Statistic," Journal of the Royal Statistical Society, Series B, 2001, 63:411–23.