Complex System Fault Detection Using Factor Analysis

Yilu Zhang

GM R&D Center, Warren, MI 48090, USA
yilu.zhang@gm.com

ABSTRACT

In this paper, we propose to use a data modeling technology, Factor Analysis, in the application of complex system fault diagnosis and failure prognosis. Factor Analysis captures the dominant dependency underlying observable measurements of physical systems, and is sensitive to their changes, as demonstrated by the preliminary experimental results on two real-world datasets. Comparison studies show that Factor Analysis has advantages over two related techniques, Principal Component Analysis and K-Means. [*]

1 INTRODUCTION

The quality of modern life enjoys continuous improvement due to the extensive use of intelligent systems, such as robotic assistants, automobiles, health-care device, and airplanes. At the same time, the increasing sophistication of these systems causes concerns regarding their robustness and reliability. Consequently, fault diagnosis and failure prognosis (D&P) has become a key component of many complex systems.

In general terms, D&P algorithms transfer the signals measured from the target system into the knowledge regarding the system health status. In particular, the goal of diagnosis is to detect anomalies and isolate root cause of faults. The goal of prognosis is to identify failure precursors and predict the remaining useful life of the system, which allows adequate maintenance to be provided in advance.

At the heart of D&P technologies is the understanding of system failure modes, including the classification, the symptoms, the underlying causes, the progressive patterns, and the effects. However, the proliferating system complexity makes it difficult to exhaust every potential failure mode. The mounting time-to-market pressure reduces the development time, and poses further challenges to fully explore the failure mechanisms of newly developed systems.

Effective and efficient data analytical methodologies are being called for to compensate the popular D&P methods, such as model-based approaches that require thorough knowledge of the target system, and data-driven approaches that usually need labeled failure data (Schwabacher and Goebel, 2007).

Towards this end, this paper presents some initial results of utilizing a data modeling technology called *Factor Analysis* (FA) to support the identification of new/unknown failure modes and the signature of incipient failures, given limited prior knowledge of the system. The rest of this paper is organized as follows. Section II introduces the standard FA, including the model, the interpretation of the model parameters, the parameter estimation algorithm, and a comparison of FA with a related technique, Principal Component Analysis (PCA). Section III presents an extension of FA to Mixture of Factor Analysis (MFA), and the comparison of MFA with a related technique, K-Means. Section IV presents some preliminary experimental results of using MFA for fault detection. Section V draws the conclusion and discusses future works.

2 Factor Analysis

FA is a statistical technique to represent observed random variables in terms of a much smaller number of latent variables, i.e., hidden random variables. The latter usually reveal the key driving factors or relationships underlying the observations. Therefore, these hidden random variables are usually termed as *factors.* The use of FA was pioneered by the psychologist Charles Spearman in his 1904 paper in the

field of intelligence research. Since then, FA has been widely used in areas such as business and economics, meteorology, and medical science, all of which deal with large quantities of data (Nokhandan et al., 2009; Safdari et al., 2005; Chaijaruwanich et al., 2006). We have not seen much previous work using FA in system diagnosis and prognosis though.

Physical systems are governed by fundamental physical laws. For example, an electrical system is governed by the ohm's law. As a result, the signals measured from a physical system are usually closely related, and have an underlying structure. While the specifics of the structure are different from one system to another, the structure remains the same as long as the system is in its normal operation. When the system is at fault, the underlying structure will change, and the change will be reflected in the measured signals. Capturing this change allows the capture of fault signature. The challenge for real-world applications, especially for those complex engineering systems, is that the structural dependency among signals is usually complicated, and not easy to fully understand and model. In addition, the dependency is likely to be contaminated by sensor noise. Being a correlation analysis technique, FA has a potential to address this challenge. By modeling the covariance matrix, a representation of the intrinsic structure of the measured signals, FA captures the dominant signal dependency, and de-couples the underlying structures of measured data from noise. While standard FA handles only linear systems, an extension of FA, namely mixture of factor analysis (MFA), addresses non-linear systems.

In FA, a *d*-dimensional real-valued random vector x is modeled using a *k*-dimensional random vector z, where k is generally much smaller than d. The model is written as,

$$x = \mu + \Lambda z + e \tag{1}$$

where x is the vector of *observed variables*, z is the vector of *latent variables* or *factors*, μ is the mean, Λ is a matrix called *factor loading matrix*, and e is a d-dimensional vector representing the noise. z is assumed to be $N(0, \mathrm{I})$ distributed, which means the factors in z are mutually independent with a zero mean and a unit deviation. The noise model is channel independent and Gaussian, *i.e.*, $e \sim N(0, \Psi)$, where Ψ is a diagonal matrix. The goal of FA is to identify parameters $\{\mu, \Lambda, \Psi\}$ that best fit the observed data. Since k is typically smaller than d, FA promises to reduce the redundant information.

2.1 Understanding the parameters

In general, FA identifies the hidden factors in z that govern the underlying mutual dependence among the observed variables in x.

The factor loading matrix, Λ, represents the mapping from the factor space to the original observation space. An element λij of Λ is called the *loading* of the *i*th observed variable on the *j*th factor, which, in a general sense, means the correlation between the observed variable and the factor. If multiple loadings in one column of Λ take relative high values, the corresponding factor will have high contributions (or impacts) to multiple observed variables. The number of high loadings and the value of loadings determine how important the corresponding factor is.

To get more insight of the factor loading matrix, Λ, we conduct a simple transformation on Eq. (1), and end up with,

$$\begin{aligned} \Phi(x,x) &= Var(x-\mu) \\ &= Var(\Lambda z + e) \\ &= Var(\Lambda z) + 2Cov(\Lambda z, e) + Var(e) \\ &= \Lambda\Lambda^T + \Psi \end{aligned} \tag{2}$$

Eq. (2) shows that the covariance structure of the observed variables can be decomposed into two parts. The first part is characterized by the factor loading matrix, Λ. It captures the variance shared by the different observed variables, which can be considered as the mutual dependency among observed variables. The second part Ψ is a diagonal matrix, and therefore, captures the unique variance of each observed data channel. Since FA requires Ψ to be diagonal, it forces the shared variance of the observed data into Λ. Therefore, FA favors factors that have high contribution to the shared variance. In other words, FA tends to identify the common factors underlying the observed variables.

The factor vector z can be considered as the projection of the observation vector x in the factor space. The realization of the factors in z is called *score*. FA model assumes the factors in z are mutually independent. At the first glance, this assumption sounds to impose an unnecessary limitation over the representation power of FA. In practice, this assumption is very useful to many real-world applications. For example, in prognosis applications, the key is to uncover the underlying factors that control the degradation process. Decoupling the controlling

factors avoids the evaluation of overlapping factors, and allows a clear insight of the degradation process. By ranking the independent factors, the degradation index can be developed over major factors to assess the degradation, and eventually predict the remaining useful life.

FA models the noise e separately from the factors z, which is different from a related modeling technology called *Principle Component Analysis* (PCA). In the context of D&P applications, each observation channel of FA represents a sensor. The separation of noise and factors avoids the fault signature, as represented by the factor loadings, to be corrupted by the sensor noise. This is one of the advantages of FA over PCA, as will be shown in the following sections. It should be noted that the assumption of independent noise between data channels or sensors is not a strong one in real-world applications.

2.2 Parameter estimation algorithm

The parameter estimation for FA is not straightforward due to the existence of the latent variables (the factors). To find the maximum-likelihood estimation (MLE), a class of algorithms known as *Expectation-Maximization* (EM) is usually used.

An EM algorithm starts with some initial value of the model parameters θ, and iteratively goes through the Expectation step (E step) and the Maximization step (M step) to find the model parameters that maximize the overall likelihood for the model to generate the observed data $X = \{x_n, n = 1...N\}$. Specifically, in the E step, EM assumes that the current estimate of the model parameters $\theta^{(t)}$ is the true value, and computes the expectation $(Q(\theta \mid \theta^{(t)}))$ of the log likelihood $(\log L(\theta; x, z))$ of generating the observed data, with respect to the conditional distribution of the latent variable given the observed variables. That is,

$$Q(\theta \mid \theta^{(t)}) = E_{z|x,\theta^{(t)}}[\log L(\theta; x, z)] = \int p(z \mid x; \theta^{(t)}) \cdot \log L(\theta; x, z) dz \quad (3)$$

Then in the M-step, the expectation is maximized to obtain the new estimate $\theta^{(t+1)}$. That is,

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta \mid \theta^{(t)}) \quad (4)$$

The iteration continues until no value can be found to increase $Q(\theta \mid \theta^{(t)})$. EM algorithms guarantee the monotonic increase of the likelihood. However, depending on the initial value, EM algorithms may not converge to the global maximum.

For the FA model in Eq. (2), an EM algorithm is proposed by (Ghahramani and Hinton, 1996), where the corresponding $Q(\theta \mid \theta^{(t)})$ is,

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= E\left[\ln \prod_n (2\pi)^{d/2} |\Psi|^{-1/2} \exp\{-\frac{1}{2}(x_n - \mu - \Lambda z)^T \Psi^{-1}(x_n - \mu - \Lambda z)\} \right] \\ &= -\frac{1}{2}\sum_n E\left[(x_n - \mu - \Lambda z)^T \Psi^{-1}(x_n - \mu - \Lambda z)\right] - \frac{N}{2}\ln|\Psi| + C \\ &= -\frac{1}{2}\sum_n \begin{pmatrix} (x_n - \mu)^T \Psi^{-1}(x_n - \mu) - 2(x_n - \mu)^T \Psi^{-1} \Lambda E[z \mid x_n; \theta^{(t)}] \\ + tr\left(\Lambda^T \Psi^{-1} \Lambda E[zz^T \mid x_n; \theta^{(t)}]\right) \end{pmatrix} \\ &\quad - \frac{N}{2}\ln|\Psi| + C \end{aligned} \quad (5)$$

where $\theta = \{\mu, \Lambda, \Psi\}$, $\theta^{(t)} = \{\mu^{(t)}, \Lambda^{(t)}, \Psi^{(t)}\}$, tr is the trace operator with $tr(A) = \sum_i a_{ii}$, and C is a constant.

To run the EM algorithm, one needs to specify the total number of factors to be learned†, and randomly generate the initial values for $\theta^{(0)}$ ($\theta^{(0)} = \{\mu^{(0)}, \Lambda^{(0)}, \Psi^{(0)}\}$). Then, the following E-step and M-step are executed until the likelihood is not increasing. The computations involved in the **E step** are,

$$E[z \mid x; \theta^{(t)}] = \beta x \quad (6)$$

and,

$$\begin{aligned} E[zz^T \mid x; \theta^{(t)}] &= Var(z \mid x) + E(z \mid x)E(z \mid x)^T \\ &= I - \beta\Lambda + \beta x x^T \beta^T \end{aligned} \quad (7)$$

where $\beta = \Lambda^{(t)^T}(\Psi^{(t)} + \Lambda^{(t)}\Lambda^{(t)^T})^{-1}$. The **M step** is,

$$\Lambda^{(t+1)} = \left(\sum_{n=1}^{N} x_n E[z \mid x; \theta^{(t)}]^T\right)\left(\sum_{n=1}^{N} E[zz^T \mid x_n; \theta^{(t)}]^T\right)^{-1} \quad (8)$$

and,

$$\Psi^{(t+1)} = \frac{1}{N} diag\left\{\sum_{n=1}^{N} x_n x_n^T - \Lambda^{(t+1)} E[z \mid x_n; \theta^{(t)}]^T x_n^T\right\} \quad (9)$$

where the $diag$ operator sets all the off-diagonal elements of a matrix to zero.

2.3 Comparison with PCA

FA is similar to PCA in that both are dimension-reduction techniques that combine data channels based on the data's covariance structure.

However, FA is fundamentally different from PCA. FA belongs to the group of approaches called *generative model*. With the generative models, a stochastic process is mathematically formulated to represent the observation data. The stochastic process

† How to specify the optimal total number of factors is still an open problem. In the experiments presented in this paper, we picked the number with a limited trial-and-error process.

typically has hidden parameters that need to be identified. If the observed data is indeed generated by the model, fitting the parameters of the generative model to maximize the data likelihood provides an optimal solution to interpret the data. The "generative" aspect of FA makes it particularly suitable for the applications such as D&P. In typical D&P applications, there is an underlying physical process governing the system behavior, which is reflected in the sensor measurements. By finding the underlying model that controls the process, and fitting the parameters with the observed data, FA is sensitive to the model change (including parameter and structure changes) due to system faults and, therefore, warrants the capture of the faults.

On the other hand, PCA empirically identifies the components (directions) along which the data has major variance, regardless of the underlying model that governs the generation of the observed data. By maximizing the variance of the input data, PCA may retain unwanted variations. In the context of D&P, there is a potential risk that PCA picks up non-fault-related data variance, such as sensor noise, instead of system fault related variance, which may lead to incorrect diagnostics decisions.

To show the difference between FA and PCA, we did an experiment on a synthetic dataset that is generated by the following model,

$$x = \mu + \Lambda z + e, \tag{10}$$

where, $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \sim N(0, I)$, $e \sim N(0, \Psi)$,

$$\mu = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ \Lambda = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \ \Psi = \begin{bmatrix} 0.1 & & \\ & 0.1 & \\ & & 4 \end{bmatrix}.$$

As can be seen, the observed data x is controlled by two underlying factors, where the first factor is a dominant factor since it impacts two data channels x_1 and x_2, while the second factor only impacts the 3rd data channel. At the same time, the 3rd data channel is deliberately set to have a large noise.

Two thousand data points were generated and fed to both FA and PCA. Fig. 1 shows the results. For better illustration, Fig. 1 shows both the data with noise (green crosses) and without noise (blue dots). The blue line shows the direction of the actual dominant factor. It is aligned with the major spreading direction of the data without noise (blue dots). The black line and the red line are the directions of the dominant factor recovered by PCA and FA, respectively. As can be seen, FA correctly recovers a dominant direction (red line) very close to the actual one (blue line). On the other hand, PCA is confused by the noise, and recovers a dominant direction (black line) along the 3rd data channel x_3, which has the highest noise.

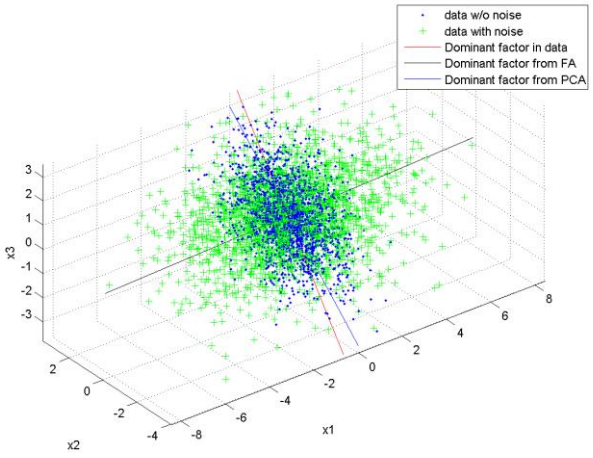


Fig. 1: Comparison of FA and PCA in identifying the dominant factor

3 Mixtures of Factor Analysis

In real-world applications, the linear relationship of Eq.(1) may not always hold. For example, shown in Fig. 2 are battery voltage and current during multiple vehicle cranking events that happened at different battery ages during an accelerated-ageing experiment. As can be seen, as the battery ages, the relationship between its voltage and current changes too. The overall relationship can no longer be described by one linear model in Eq. (1).

An extension of standard FA is to model the data in a piece-wise manner. That is, as shown in Fig. 2, to divide the data into clusters, and fit each cluster with a separate FA model. The objective is that the overall mixture model has the maximized likelihood of generating all the observed data. This extension is called, *Mixtures of Factor Analysis* (*MFA*). Each FA model in the overall mixtures of models is usually called an *analyzer*, which models one cluster of the data.

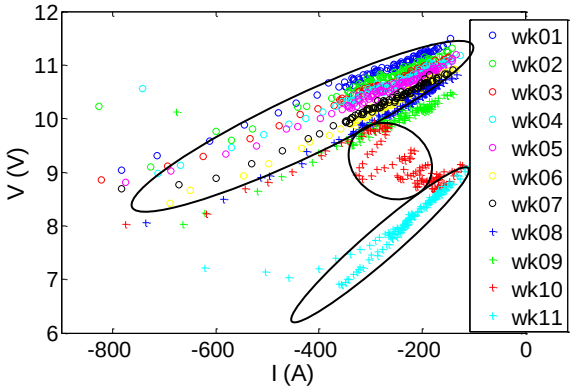


Fig. 2: An illustration of MFA: the observed data is battery voltage and current during multiple vehicle cranking events happened at different battery ages. The data is divided into clusters as indicated by circles, each of which is modeled by a standard FA model.

An MFA model is written as

$$x = \mu_m + \Lambda_m z + e \tag{11}$$

where μ_m is the mean of the *m*-th analyzer (corresponding to the *m*-th cluster), Λ_m is the loading matrix of the *m*-th analyzer, and $m = 1..M$. Similar to the FA model, the factors in z are assumed to be $N(0, \mathrm{I})$ distributed, and the *d*-dimensional random variable e is $N(0, \Psi)$ distributed, where Ψ is a diagonal matrix. From Eq. (11), we have,

$$P(x \mid z, \omega_m) = N(\mu_m + \Lambda_m z, \Psi) \tag{12}$$

where ω_m represents the *m*-th cluster in MFA. Therefore, the generative model of MFA can be shown in the following mixture distribution,

$$P(x) = \sum_{m=1}^{M} \int P(x \mid z, \omega_m) P(z \mid \omega_m) P(\omega_m) dz \tag{13}$$

where, $P(\omega_m) = \pi_m$ is the prior probability of the *m*-th cluster. The parameters of this model are,

$$\theta = \{\mu_m, \Lambda_m, \pi_m, \omega_{mn}, \Psi; m = 1..M, n = 1..N\},$$

where $\omega_{mn} = 1$, if x_n belongs to the *m*-th cluster. Compared to the FA model, there is an extra latent viable, ω_{mn}, which specifies the membership of each data point with respect to the clusters. Note that the clusters share the same Ψ to preserve the interpretation of e as the sensor noise.

The key of MFA is to divide the data into clusters, each of which has data with similar covariance structure. Similar to standard FA, EM algorithms have been developed to estimate the parameters in MFA (Ghahramani and Hinton, 1996).

3.1 Comparison with K-Means

While FA is a dimension reduction technique, MFA, the extension of FA, is fundamentally a clustering technique, as illustrated in Fig. 2. It is useful to compare MFA with another popular clustering technique called *K-Means* (Jain *et al.*, 1999).

K-Means is also an iterative algorithm. In each step, it starts with multiple cluster centers, where the initial cluster centers are selected randomly. The data points are assigned to different clusters based on their distance from the cluster centers. The cluster center is then re-estimated using the updated cluster assignment.

It should be noted that the iteration process of K-Means is very similar to the EM algorithms used in MFA. The main difference is that the clustering assignment in K-Means is based on the distance metric, while that in MFA is based on the similarity of the covariance structure.

To better understand the difference, we show some comparison results on a synthetic dataset. The synthetic dataset consists of two groups of data, generated by the following model,

$$x = \mu_m + \Lambda_m z + e, \tag{14}$$

where, $m = 1,2$, $z \sim N(0, I)$, $e \sim N(0, \Psi)$,

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \Lambda_1 = \begin{bmatrix} 0.9 \\ 0.5 \\ 0.1 \end{bmatrix}, \quad \Lambda_2 = \begin{bmatrix} 0.2 \\ 0.9 \\ 0.5 \end{bmatrix},$$

$$\Psi = \begin{bmatrix} 0.15 & & \\ & 0.15 & \\ & & 0.15 \end{bmatrix}.$$

Note that each group has one underlying factor. We generated 5000 samples for the first group and 100 samples for the second group. The unbalance of the group size is designed to simulate the situation in fault detection, where the normal operation data usually out-numbers the abnormal operation data. Fig. 3 (a) shows the synthetic dataset, with data from the first group in blue and that from the second group in red.

Without specifying the group ID, the data is fed into both the EM algorithm for MFA proposed by Ghahramani and Hinton (Ghahramani and Hinton, 1996), and the K-Means implemented by MATLAB. Fig. 3 (b) shows the data labeled by MFA as group 1 (blue), and Fig. 3 (c) shows the data labeled by MFA as group 2 (red). Similarly, the results of K-Means are shown in Fig. 3 (d) and (e). As can be seen clearly, using the covariance structure as the similarity metric, MFA recovered the group ID with much better

performance than K-Means.

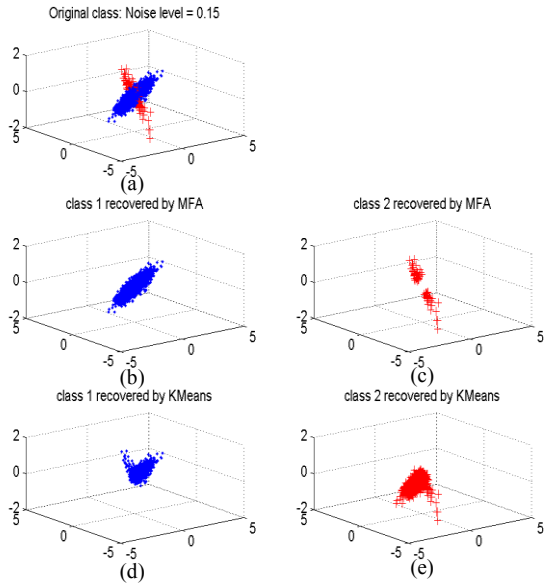


Fig. 3: Clustering results for MFA and K-Means.

4 Preliminary Experimental Results on Fault Detection

In this section, we present some initial experiment results on fault detection using MFA. It should be pointed out that fault classification with MFA is not addressed in this paper.

4.1 Robot execution failure detection

The first set of data, Robot Execution Failures, is obtained from UC-Irvine Machine Learning Data Repository (http://archive.ics.uci.edu/ml/). There are 5 datasets in total. Each dataset corresponds to one particular robot maneuvering. Each dataset consists of multiple segments of force and torque measurements. Some of the segments are collected during normal operations. The other are collected right after a failure. Each segment lasts 315ms, and contains 15 data samples collected at regular intervals. Each sample has 3 force measurements and 3 torque measurements.

Due to the space limit, reported here are the results on the first dataset, "LP1: approach to grasp position." The results of MFA and K-Means on other datasets of Robot Execution Failure are similar. The total number of segments in LP1 is 88, whose breakdown is as follows, 24% normal, 19% collision, 18% front collision, and 39% obstruction.

The data is fed into the MFA with 4 analyzers and 3 factors in each analyzer. After the learning algorithm converges, the membership of each data sample, ω_{mn}, is generated by the learned MFA model. The results for the dataset LP1 are shown in Fig. 4. For better illustration, in Fig. 4, we group data segments of the same operation together. The solid vertical lines are the group boundaries. The dashed vertical lines are segment boundaries. The y axis is the membership of the data samples generated by MFA. One interesting observation from Fig. 4 is that MFA groups all of the normal operation data samples into one cluster (cluster 3), while the membership of the data samples from the failure operations jumps among all four clusters. A close examination of the results shows that every failure operation segment has at least one data sample with the membership other than 3. Based on this, the fault detection algorithm can be as simple as,

"If at least one data point within the segment has a membership other than 3, a fault is detected."

At segment level, this algorithm yields zero false alarm rate and zero false negative rate in detecting failure operations for this dataset. Keep in mind that this is achieved by simply feeding the raw data into the MFA model without any manual intervention such as labeling.

As a comparison, the data is fed into K-Means, and the results are shown in Fig. 5. It is clear that K-Means can differentiate the failure operations of obstruction from normal operations, collisions, or fr-collisions. However, it cannot differentiate normal operations from collisions or fr-collisions, because the membership 2 dominants all these three types of operations.

4.2 Vehicle battery fault detection

The second set of data was collected by GM R&D in developing battery state of health estimation algorithms (Zhang et al., 2009). In this data collection effort, 12 batteries from different suppliers were aged from fresh to the end of life through an accelerated ageing process. The battery age varies from 13 to 44 weeks. During the ageing process, weekly cranking tests were conducted on a test vehicle for each battery after it was conditioned to a high state of charge (SOC) and the temperature of 25C. Battery current, battery voltage, and engine RPM were collected at a sampling

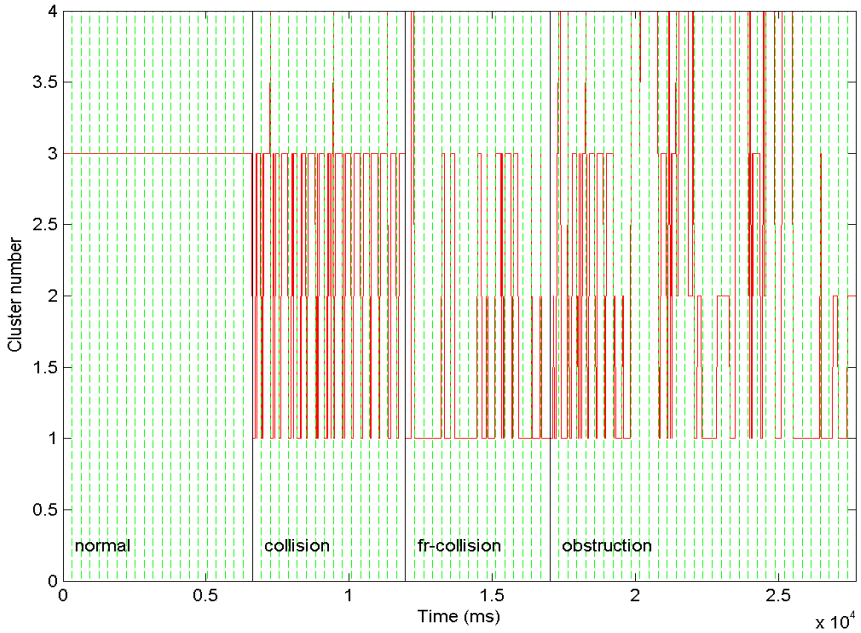


Fig. 4: Results for MFA on Robot Execution Failure dataset LP1

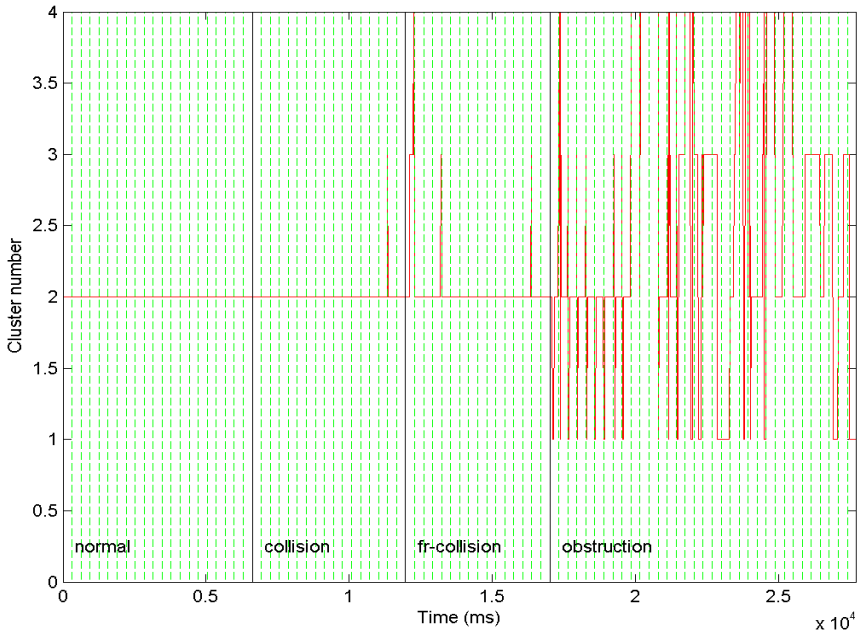
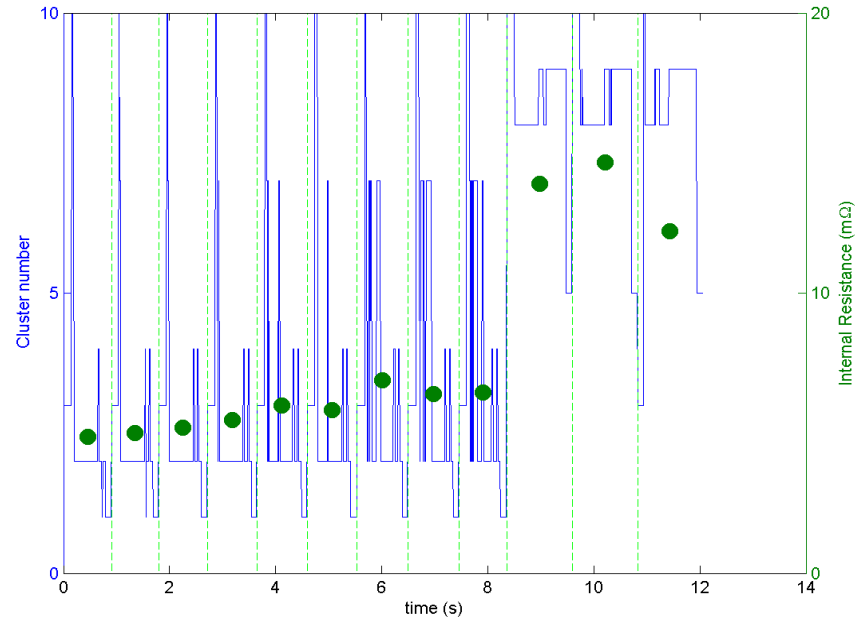


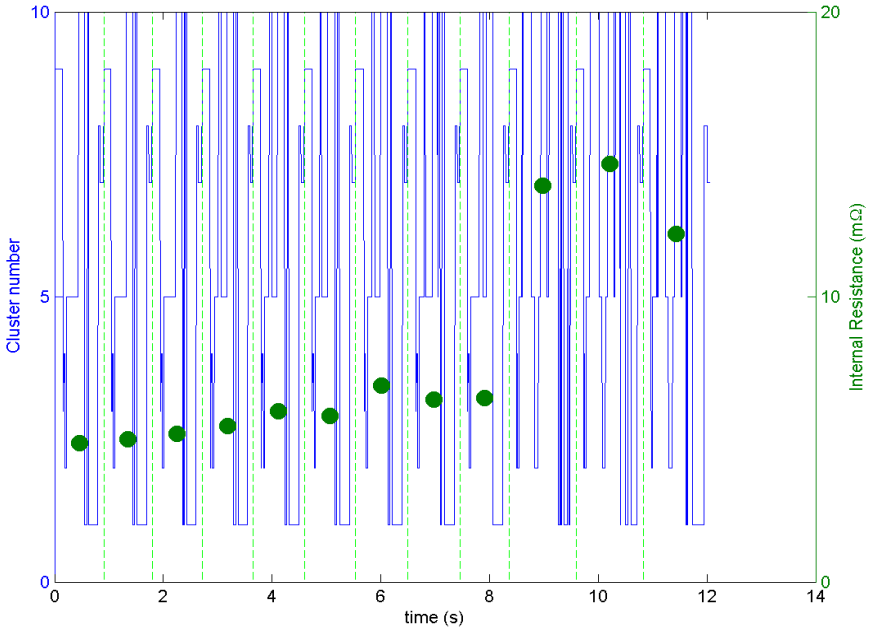
Fig. 5: Results for K-Means on Robot Execution Failure dataset LP1

frequency of 125Hz during cranking. After data cleaning, there are totally 279 cranking data files that have adequate data. Each cranking lasts a little bit less than 1 second. Therefore, each data file has about 100 data samples.

The data is fed to the MFA with 10 analyzers and 1 factor for each analyzer. After the learning algorithm converges, the membership of each data sample, ω_{mn}, is generated by the learned MFA model. Fig. 6 (a) shows the result of one representative battery. The green vertical lines specify the boundaries of the cranking data files. The cranking data files are ordered by increasing battery age from left to right. It can be seen from Fig. 6 (a) that there is a clear pattern change in the cluster number sequence, as the battery is getting close to the end of life. This means that MFA has



(a)



(b)

Fig. 6: Results for (a) MFA and (b) K-Means on battery cranking data

detected an underlying change in the covariance structure of the data sample as battery ages.

A related data provides more insight to the underlying change indicated by the MFA results. In Fig. 6 (a), the green dots plotted in the middle of each segment indicate the estimated internal resistance of the battery, whose value increases significantly towards the end of the battery life. Consider a simplified battery model (Plett, 2004),

$$V = V_{oc} - I \cdot R \tag{15}$$

where V_{oc} is the battery open circuit voltage, R is the internal resistance, and V and I are battery terminal voltage and current. It is understandable that the internal resistance is the underlying dependency

between battery terminal voltage and current. MFA captures the change of this underlying dependency without prior knowledge of the system, *i.e.* the battery model.

For comparison, the same set of data is fed into K-Means, and the results are shown in Fig. 6 (b). It is clear that K-Means is not able to capture the change in battery characteristics.

5 Conclusions and Future Works

In this paper, we discuss a data modeling technology, Factor Analysis, and propose its applications in system diagnosis and prognosis. The preliminary yet promising experimental results on two real-world datasets suggest that FA is sensitive to intrinsic system changes caused by system faults, and can capture the change without thorough prior knowledge about the system. This property makes FA an excellent candidate technology to facilitate the fast analysis and detection of new system failure modes, which is critical to enhance the robustness and reliability of complex intelligent systems.‡ While this paper presents some initial results on fault detection, yet to be explored is the feasibility of using Factor Analysis for fault isolation and failure prediction.

REFERENCES

(Chaijaruwanich *et al*., 2006) J. Chaijaruwanich, J. Khamphachua, S. Prasitwattanaseree, S. Warit, and P. Palittapongarnpim, "Application of Factor Analysis on Mycobacterium Tuberculosis Transcriptional Responses for Drug Clustering, Drug Target, and Pathway Detections," in *Lecture Notes in Computer Science: Advanced Data Mining and Applications*, Vol. 4093, pp. 835-844, 2006.

(Ghahramani and Hinton, 1996) Ghahramani, Z. and G.E. Hinton, "The EM Algorithm for Mixtures of Factor Analyzers." CRG-TR-96-1, University of Toronto, 1996.

(Jain *et al.*, 1999) A.K. Jain, M.N. Murty, P.J. Flynn, "Data Clustering: A Review," in *ACM Computing Surveys*, Vol. 31, No. 2, pp. 264-323, 1999.

(Nokhandan *et al.*, 2009) M.H. Nokhandan, G.A. F. Ghalhary and M. Mousavi-Baygi, "The application of factor analysis and artificial neural networks in predicting spring precipitation by means of climatic parameters of the upper levels of atmosphere," in *Trends in Applied Sciences Research*, Vol. 4, Issue 2, pp 85-97, 2009.

(Plett, 2004) G.L. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs, Part 2, modeling and identification", in *Journal of Power Sources*, Vol. 134, pp. 262-276, 2004.

(Safdari *et al.*, 2005) Cyrus Safdari, Nancy J. Scannell, Rubina Ohanian, "A Statistical Approach to Peer-Groupings; The Case of Banks in Armenia," in *The Journal of American Academy of Business, Cambridge*, Vol. 6, No. 2, March 2005.

(Schwabacher and Goebel, 2007) M. Schwabacher, K. Goebel, "An survey of artificial intelligence for prognostics," in *Proceedings of AAAI Fall Symposium*, Arlington, VA, 2007.

(Zhang *et al.*, 2009) Xiaodong Zhang, Ryan Grube, Kwang-Keun Shin, Mutasim Salman, Robert Conell, "Automotive Battery State-of-Health Monitoring: A Parity Relation Based Approach," in *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (Safeprocess 2009)*, Barcelona, Spain, June 30-July 3, 2009.

Yilu Zhang received his B.S., and M.S. degrees in Electrical Engineering from Zhejiang University, China, in 1994, and 1997, respectively; and his Ph.D. degree in Computer Science from Michigan State University in 2002. He joined the R&D center of General Motors Co. at Warren, Michigan in 2002, and currently holds a position of Staff Researcher.

Dr. Zhang's research interests include statistical pattern recognition, machine learning, signal processing, and their applications such as integrated system health management and human machine interactions. Dr. Zhang is a Senior Member of IEEE. He is a winner of 2008 "Boss" Kettering Award – the highest technology award in GM – for his contribution to Connected Vehicle Battery Monitor, a remote vehicle diagnostics technology.

‡ It should be noted that the physical interpretation of the detected failure modes is still up to the domain experts.