Research Article

Adaptive Fault Detection for Complex Dynamic Processes Based on JIT Updated Data Set

Jinna Li,[1, 2] Yuan Li,[3] Haibin Yu,[2] Yanhong Xie,[1] and Cheng Zhang[1]

[1] *Department of Science, Shenyang University of Chemical Technology, Liaoning, Shenyang 110142, China*

[2] *Lab of Industrial Control Networks and Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Liaoning, Shenyang 110016, China*

[3] *College of Information Engineering, Shenyang University of Chemical Technology, Liaoning, Shenyang 110142, China*

Correspondence should be addressed to Jinna Li, lijinna_721@yahoo.com.cn

A novel fault detection technique is proposed to explicitly account for the nonlinear, dynamic, and multimodal problems existed in the practical and complex dynamic processes. Just-in-time (JIT) detection method and k-nearest neighbor (KNN) rule-based statistical process control (SPC) approach are integrated to construct a flexible and adaptive detection scheme for the control process with nonlinear, dynamic, and multimodal cases. Mahalanobis distance, representing the correlation among samples, is used to simplify and update the raw data set, which is the first merit in this paper. Based on it, the control limit is computed in terms of both KNN rule and SPC method, such that we can identify whether the current data is normal or not by online approach. Noted that the control limit obtained changes with updating database such that an adaptive fault detection technique that can effectively eliminate the impact of data drift and shift on the performance of detection process is obtained, which is the second merit in this paper. The efficiency of the developed method is demonstrated by the numerical examples and an industrial case.

1. Introduction

Fault has been a constant topic of research for several decades [1–4]. Several fault detection methods have been developed to solve problems since there exists a growing need for fault detection in the real process engineering not only from the plant's safety perspective but also from considering the quality of the process products [5–7]. Moreover, the existing methods used to fault detection have been applied into a broad range of areas such as chemical process,

networked control systems and semiconduction process, and so forth [8–11]. The dynamic change, multiple mode, and nonlinearity exist objectively in the most of the aforementioned process, such as semiconduction process, tank reactors, and so forth [9–11], which has brought new challenges to the analysis and implementation of fault detection. Therefore, they must be taken into account carefully in developing a high-performance and adaptive fault identification method to detect the abnormal cases as early as possible.

As summarized in [12], the technologies of process analysis and operation are derived broadly into two categories: model-based approach and data-based approach. In the model-based approach, static or dynamic models are built for the process under normal operating situation. The difference between the actual process output and nominal model's output is monitored to determine whether any fault occurs or not [5, 13–18]. Noted that many control processes are data rich but information poor, which senses the data-based method is strongly needed to obtain a flexible and high-efficiency detection manger systems. Among the reported results in the literature, to mention a few, data-driven KNN fault detection was addressed in [9]. Based on it, [10] proposed an improved principal component analysis (PCA) KNN technique to implement the fault identification. Noted that data-based fault monitoring and identification methods were also investigated in [11, 18, 19].

It should be pointed out the nonlinearity, dynamics, and multimode are the inevitable obstacles for either model-based approach or data-based approach when process detection is required in the real-world applications. Various methods, including statistical process control (SPC), multivariate statistical process control (MSPC), qualitative knowledge-based methods, artificial intelligence, and various integrated methods, have been developed and performed in the available literature [20–23]. Well known that PCA is used in MSPC widely, the existing nonlinear PCA method [24, 25], dynamic PCA method [26], and independent component analysis PCA method [18] have been presented to address the nonlinearity, dynamics, and multimode faced by fault detection, respectively. Since JIT is inherently adaptive in nature, which is achieved by storing the current measured data in the database [27, 28], and is capable of detecting and diagnosing whether the query is normal or not by on-line and adaptive approach [19], as shown in Figure 1, data-based JIT method has attracted much more scholars' attention in the recent years (see [12]). Reference [12] proposed a data-based JIT-SPC detection and identification technique, in which the distance was calculated and checked every time when fault detection need to be conducted. References [9, 10] applied KNN rule to fault detection for semiconductor manufactory process, in which k-nearest neighbors were used to tackle the multimodal problem. However, though nonlinear PCA method [24, 25], dynamic PCA method [26], independent component analysis PCA method [18], data-based JIT [12, 18, 19], and KNN methods [9, 10] have been reported, the efficient method which can naturally handle nonlinear correlation among the variables, dynamic change of the systems and multimodal batch trajectories have not been fully investigated so far.

As a matter of fact, it is crucial for realizing the desired performance in detection to determine an appropriate normal operation data stored in database, since too much data will have heavy load on both storage cost and computation, while, less data naturally effect on the efficacy of detection technique. How to determine an appropriate raw data set so that cost, computation complexity, and performance of detection can be compromised is a challenge. Based on the Mahalanobis distance between samples, the simplification of raw data set is studied in [29], in which the simplifying procedure is terminated in terms of the desired number of samples. However, how to determine number of samples that is closely related to quality of detection was not investigated. Well known to all, data drift and shift exist
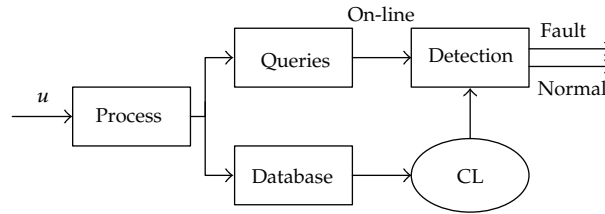
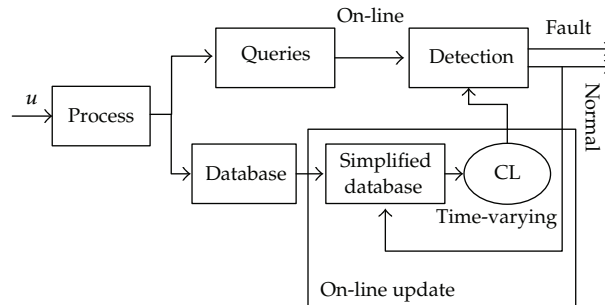**Figure 1:** On-line detection scheme.

Figure 2: Simplified database-based on-line detection scheme.

in the practical complex dynamic process inevitably, which can be produced due to much more insuperable causes, such as aging of instrument, variation of temperature, effect of environment, and difference in the coming materials, and so forth [30]. In this case, some of queries with drift and shift that might be normal are mistaken for faults, or that might be fault are falsely identified as normalities. Then, it is of paramount importance to actively regulate the control limit (CL) by on-line approach, which will surely improve the quality of fault detection. However, to the best of authors' knowledge, how to simplify and update the raw samples set to light the computation load and realize high performance have not been investigated fully to date. Especially for data-driven fault detection with time-varying control limit, few results have been available in the literature so far, which motivates the present study.

The overall goal in this paper is to propose a flexible, adaptive JIT scheme of fault detection. By computing the Mahalanobis distance between normal samples, as well as Mahalanobis distance from the query to the normal operation data stored in the database, the raw data set is simplified and updated. Moreover, the updated database-based time-varying CL is derived in terms of KNN rule combined with SPC method, such that the judgment is conducted every time when fault detection is required as shown in Figure 2. Moreover, the integration of JIT method and KNN technique is well suited for the nonlinear, dynamic and multimodal fault detection process. Finally, simulation results and industrial case illustrate the efficiency of the method proposed.

2. Simplifying the Raw Data Set

In this section, we will describe the method of simplifying the raw data set.

In the practical detection of the process, the huge amounts of raw data bring the serious calculation load and cost spending for fault detection, identification, and diagnosis.

Since the closer Mahalanobis distance between two samples is, the more similar their basic features are, we take mean of them to retain the common characteristics, and this mean is put into the raw data set to replace the original two samples, which can keep the characteristics of raw data set to the greatest extent [29]. Let $\mathbf{X}$ denotes the raw data matrix with n samples (rows) and m variables (col021umms). The specific procedure is as follows.

Algorithm 2.1.

  Step 1. Let $\mathbf{Z}(n \times m) = \mathbf{X}(n \times m)$, and the covariance matrix of $\mathbf{X}$ is defined as: $\mathbf{V} = (v_{ij}) = ((x_i - \overline{x}_i)'(x_j - \overline{x}_j)/(n-1))$. Where x_i, x_j $(i, j = 1, 2, \ldots, m)$ denote the stochastic variables, and $\overline{x}_i$ and $\overline{x}_j$ denote the mean of them.

  Step 2. Calculate the Mahalanobis distance between every sample and all the other samples stored in data set $\mathbf{Z}$, which is denoted as md_{ij} $(i, j = 1, 2, \ldots, n)$ and is placed in Mahalanobis distance matrix $\mathbf{MD}(n \times n) = (md_{ij})$.

  Step 3. Find out the minimum and nonzero element in each raw in the matrix $\mathbf{MD}$, which is placed in row vector $\mathbf{v}(1 \times n)$, and the place (column number) of each minimum element in each row is recorded in row vector $p(1 \times n)$. Based on it, finding out the minimum value in $\mathbf{v}(1 \times n)$, and if it's place in $\mathbf{v}(1 \times n)$ is i and No. i element is j in vector $p(1 \times n)$, then md_{ij} is the minimum value in the matrix $\mathbf{MD}(n \times n)$, which presents the minimum Mahalanobis distance between the sample i and the sample j in raw data set.

  Step 4. Let $\mathbf{M} = \mathbf{Z}$, and we use the mean of the sample i and the sample j to replace the sample i, and delete the sample j, then the row number of the matrix $\mathbf{Z}$ is reduced a line. Define $\Gamma = ((z_i - \overline{z}_i)'(z_j - \overline{z}_j)/(l-1))$, where z_i, z_j $(i, j = 1, 2, \ldots, m)$, and l denotes the row number of the matrix $\mathbf{Z}$. If $1 \geq \mathrm{Trace}(\Gamma)/\mathrm{Trace}(\mathbf{V}) \geq \epsilon$, where $0 < \epsilon \leq 1$, return to Step 2; otherwise, $\mathbf{M}$ is the simplified data matrix. Exit.

  Pseudocode 1 is given.

Remark 2.2. Motivated by PCA [10, 19, 31], where combination of variables that capture the largest amount of information in data set is found, the inequality $1 \geq \mathrm{Trace}(\Gamma)/\mathrm{Trace}(\mathbf{V}) \geq \epsilon$ realizes the preservation and update of the original information to the greatest extent in Algorithm 2.1. Note that $\mathrm{Trace}(\Gamma)$ and $\mathrm{Trace}(\mathbf{V})$ denote the trace of covariance matrix of simplified or update data set and original data set, respectively, and well known that $\mathrm{Trace}(\mathbf{V})$ is equal to the sum of all the eigenvalues of matrix $\mathbf{V}$. In this sense that the threshold ϵ is selected to maximize the retention of the originally statistical information. For example, $\epsilon = 0.98$ means that 98% of the variance in raw data set is represented by the new simplified data set $\mathbf{Z}$. Obviously, Algorithm 2.1 is a kind of logical and promising way for the data-driven fault detection.

Remark 2.3. Different from [29], in which though the raw data is reduced based on Mahalanobis distance between samples, the subjectively determined number of simplified data has an essential effect on high-performance detection process. In this paper, the threshold that limits the extent of reducing the raw data is given, which is a preferable way that compromises the accuracy of fault detection and the computation complexity as well as lower cost.

Remark 2.4. Apparently, simplification and update scheme of raw data set presented for fault detection also apply to data-driven fault monitoring, diagnosis, and isolation, since the proper number of raw data still need to be determined and data drift and shift still needs to be

```
for i = 1 : m
    for j = 1 : m
        cov(i, j) = sum((x(:, i) − mean(x(:, i))).
        *(x(:, j) − mean(x(:, j)))) /(size(x, 1) − 1);
    end
end
  Z = X;
  ss1 = eig(cov);
  [sm1, sn1] = size(ss1);
  lmd1 = 0
for i = 1 : sm1
    lmd1 = lmd1 + ss1(i);
end
while 1
[p, q] = size(Z)
for i = 1 : q
    for j = 1 : q
        cov(i, j) = sum((z(:, i) − mean(z(:, i))).
        *(z(:, j) − mean(z(:, j)))) /(size(z, 1) − 1);
    end
end
  ss = eig(cov);
  [sm, sn] = size(ss)
  lmd = 0
for i = 1 : sm
    lmd = lmd + ss(i);
end
if lmd/lmd1 < ϵ||lmd/lmd1 > 1
    break
end
  [n, m] = size(Z);
  d = zeros(n, n);
  M = Z;
for i = 1 : n
    for j = 1 : n
        d(i, j) = d(i, j) + (z(i, :) − z(j, :)) * inv(cov)
        *(z(i, :) − z(j, :))^T
    end
end
  md = zeros(n − 1, n − 1);
for i = 1 : n
    for j = 1 : n
        if d(i, j) > 0
            md(i, j) = d(i, j);
        else
            md(i, j) = 10000;
        end
    end
end
  [The_Min,The_Min_ID] = min(md')
  v = The_Min;
  p = The_Min_ID;
  [The_Min, The_Min_ID] = min(v);
  i = The_Min_ID;
  j = p(i);
  z(i, :) = (z(i, :) + z(j, :))/2;
  z(j, :) = [ ];
end
```

Pseudocode 1

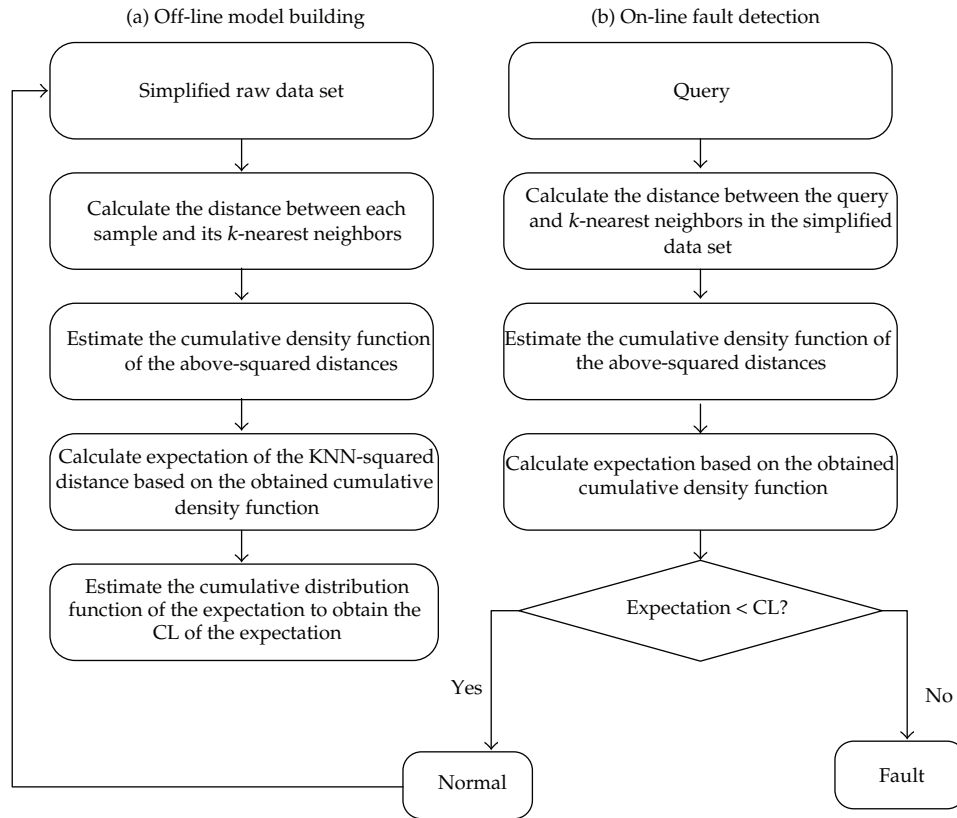(a) Off-line model building

(b) On-line fault detection

Figure 3: Flow chart of proposed detection method.

coped with. In this sense that the simplification and update technique suggested in this paper is quite general.

3. Detection Method

In this section, we will give the fault detection method including off-line and on-line cases as shown in Figure 3.

3.1. Off-Line Model Building

Algorithm 3.1.

    Step 1. Set $i = 1$, and s is the number of simplified data set.

    Step 2. Find k nearest neighbors (see [30]) for each sample i in the simplified data set and calculate the distances between sample i and its k-nearest neighbors.

    Step 3. Estimate the cumulative density function of the above-squared distances by the function "ksdensity" in Matlab, which is denoted as $f_i(x)$.

    Step 4. Calculate expectation of the KNN-squared distance based on the obtained cumulative density function in terms of the definition of expectation, and the expectation is denoted as $\overline{X}_i$.

Step 5. Set $i = i + 1$. If $i \leq s$, go to Step 2; otherwise, go to Step 6.

Step 6. Estimate the cumulative distribution function of the expectation to obtain the CL.

Remark 3.2. At Step 2, Euclidean distance is used, which is simple and easy, but any other distance is also suitable for the method proposed. The obtained CL in Algorithm 3.1 is based on the statistical test concept in the same way as SPC, in this sense off-line model is constructed by the KNN rule-based SPC approach.

Remark 3.3. In general, the estimation of probability density function in multidimensional space is difficultly derived [12]. To overcome this difficulty, we try to estimate the probability density functions of squared distance and expectation of squared distance in Step 3 and Step 4 from stochastic variable point of view. In addition, expectation of squared distance can be also obtained by taking an average over squared distances.

Remark 3.4. Since there are similar statistical characteristics for the normal samples and the distance between the fault sample and the nearest neighboring samples must be greater than the normal sample's distance to the nearest neighboring samples [9, 10, 12], setting CL to detect faults in terms of cumulative distribution function of the expectation is sound and effective. CL proposed in this paper means that the expectation values of vast majority distance for the normal samples do not exceed it. For example, 95% control limit means that the value within which 95% of population of normal operation data (expectation values) is included. Here, 95% is also called confidence level based on probability and statistical theory.

3.2. On-Line Fault Detection

Algorithm 3.5.

Step 1. Calculate the distance between the query and k nearest neighbors in the simplified data set.

Step 2. Estimate the cumulative density function of the above-squared distances.

Step 3. Calculate expectation based on the obtained cumulative density function.

Step 4. The query is abnormal if the expectation is beyond its CL, otherwise, this query is normal.

Step 5. If the query is normal, to update, it can be put into the normal samples database, which will be also simplified by using the technique described in Algorithm 2.1. In this case, the updated database is used to compute the new CL to continue to identify the next query in Step 1.

Remark 3.6. Compared with [9, 10, 12, 19], the technique that simplifies and updates raw data set is a main contribution in this paper. More importantly, the time-varying CL can be derived, such that adaptive fault detection can be implemented by on-line approach, which will eliminate the impact of data drift and shift on the quality of fault detection. Different from [18] wherein just-in-time-learning (JITL) along with two-step independent component analysis and principal component analysis was studied, in this paper, the database is updated and simplified. Moreover, note that the amount of database will not randomly increase when normal queries are added to it. The reason is that Algorithm 2.1 is implemented once the on-line detection process is completed. In other words, the updated database still can be simplified

Table 1: Control limits under the different thresholds.

Thresholds	Number of left samples	Maximum Mahalanobis distance	CL
0.99	292	$4.8747e - 005$	$8.1465e - 004$
0.95	271	$3.0298e - 004$	$8.7031e - 004$
0.6	124	0.0028	0.0026

by virtue of Algorithm 2.1. Obviously, high fault detection capability and low cost can be owned due to the usage of Algorithms 2.1–3.5.

Remark 3.7. In fact, the difficulties posed by nonlinearity, dynamics, and multiple modes of control process on fault detection have been addressed explicitly by the detection method proposed, which comes as no surprise, since the KNN technique, SPC method as well as on-line and update scheme are integrated.

4. Numerical Examples

In this section, two examples are given to show the effectiveness of the fault detection technique. The first example aims at the single modal case to show the efficacy of simplified data set based detection procedure presented in this paper. The second example is used in the multimodal case to compare with JIT method [12].

Example 4.1. Consider the following dominant nonlinear process mode [9, 10]:

$$x_1 = x_2^2 + \text{noise}. \tag{4.1}$$

Firstly, 30 normal runs are operated for verifying the method of simplifying raw data set. Here threshold $\epsilon = 0.95$ is set. By Algorithm 2.1, the 28 samples are left. Figures 4 and 5 show the raw data set and the simplified data set, respectively.

Continue to operate the system (4.1), we obtain 300 normal data used for the raw data, 5 normal runs used for validation, and 5 faults introduced, which is shown in Figure 6. The number of nearest neighbors k is set to be 10, and the confidence level is chosen as 99% to obtain the CL. Table 1 gives the number of left raw data set used for training, the maximum Mahalanobis distance, and the CL under the different thresholds ϵ, and the histogram of simplified raw data and fault detection is shown in Figure 7, where the percentages of left date to raw data and detected faults to total faults are clearly seen. Here, the maximum Mahalanobis distance means that the samples with smaller distance than it can be merged based on Algorithm 2.1. Correspondingly, the detection results under thresholds 0.99, 0.95, and 0.6 are also shown in Figures 8, 9, and 10, respectively. As illustrated in Figures 7–10, the amount of left data become gradually less and less and the CL is increasing with the decrease in threshold, consequently, the effect of fault detection becomes bad as expected.

Note that the threshold ϵ decided has a significant impact on the detection results, obviously, the bigger the threshold is, the more accurate detection operates. Simulation results presented illustrate that defection performance does not suffer degradation by virtue of the simplified data set. FD-KNN [9] is applied into this nonlinear case, and the detection result is shown in Figure 11. It should be pointed out though a better detection result is also obtained
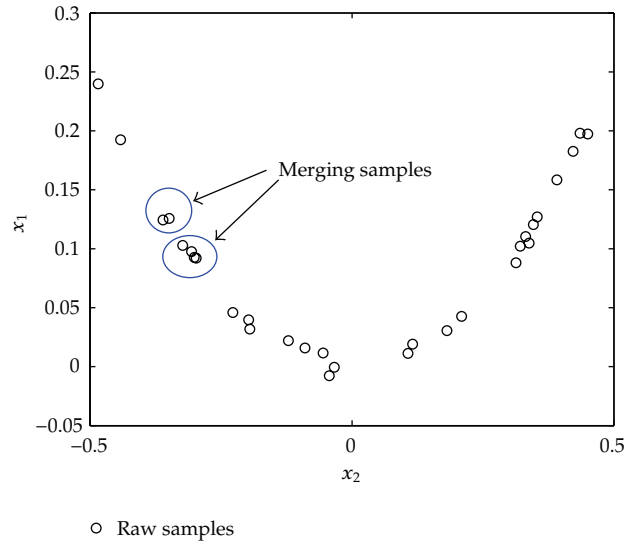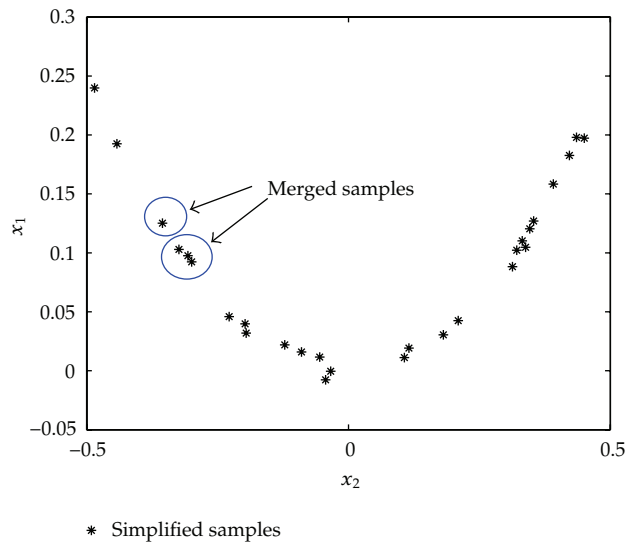
Figure 4: Raw data set.

Figure 5: Simplified data set.

by using FD-KNN approach [9], the raw data set is simplified before implementing the detection in this paper, which will contribute to the saving storage space and reducing the computational complexity.

Example 4.2. Considering the following bimodal case [9, 10]:

$$
\begin{aligned}
A \quad & y_1 = 2y_2 + \text{noise}, \\
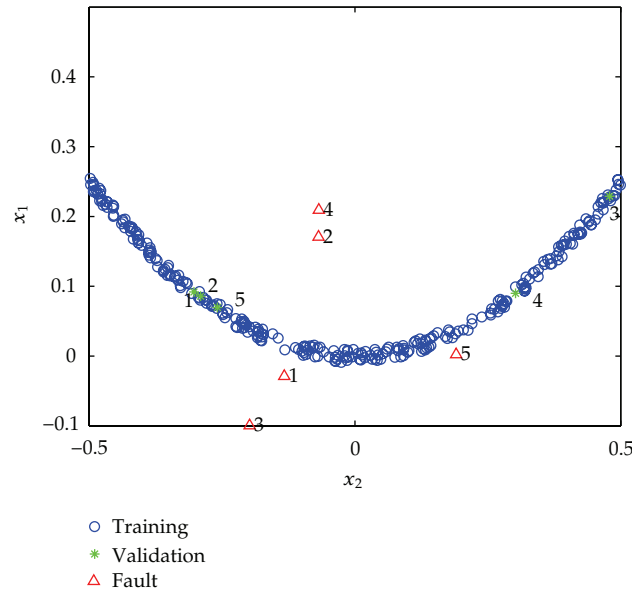B \quad & y_1 = 1.5y_2 + 6 + \text{noise}.
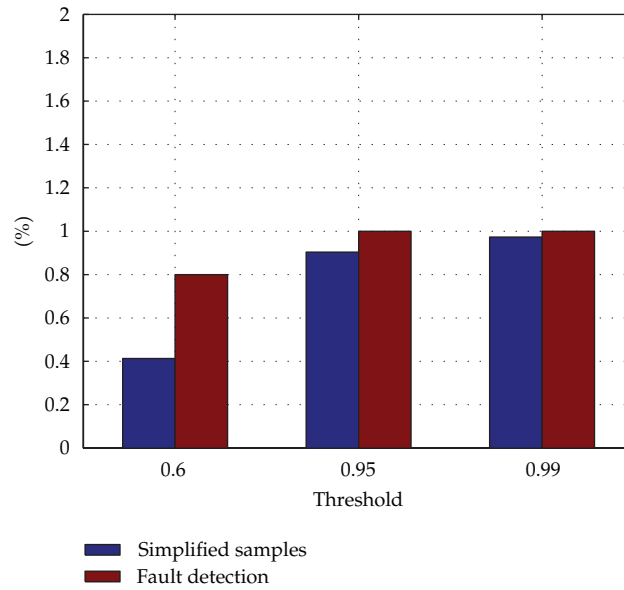\end{aligned}
\tag{4.2}
$$

Figure 6: Scatter plot of samples.

Figure 7: Histogram of detection results under different thresholds.

The above two cases are operated to produce 200 normal samples, respectively, and continue to be operated to produce 100 normal samples and 10 faults that are used for the validation and fault defection, respectively, which is given in Figure 12. For comparative analysis, 5 normal data and 10 fault data are marked. Similar to Example 4.1, the number of nearest neighbors k is set to be 10, and the confidence level is chosen as 99% to obtain the CL.
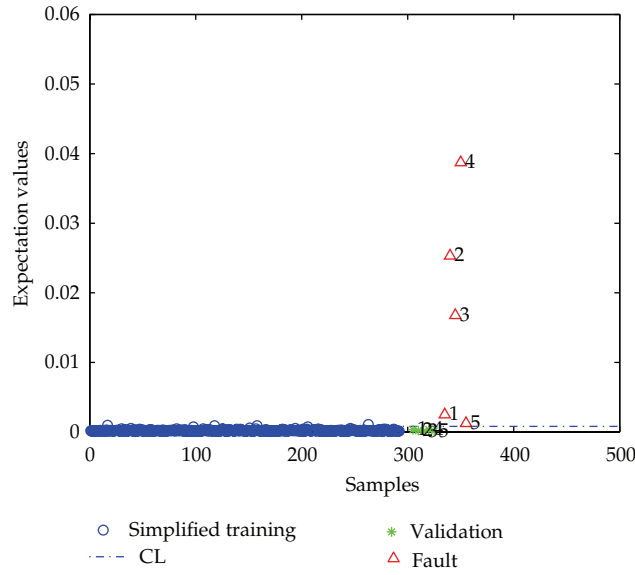
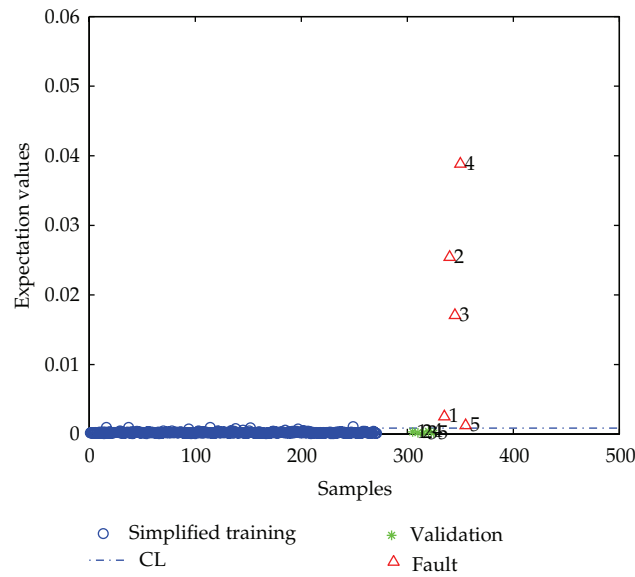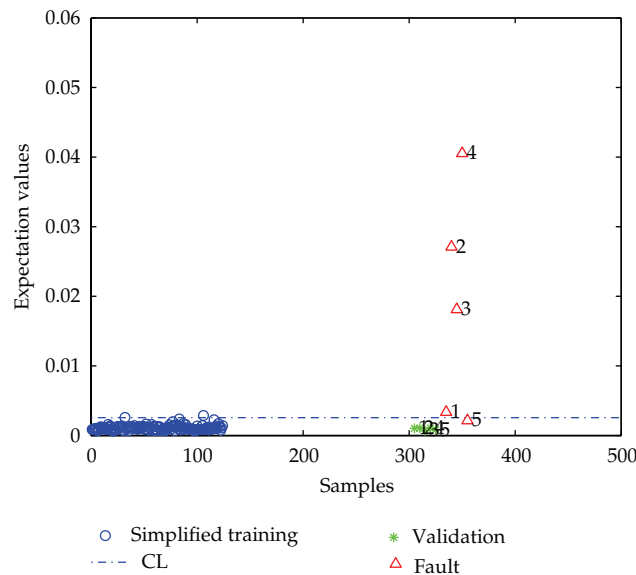**Figure 8:** Detection result under the threshold $\epsilon = 0.99$.

Figure 9: Detection result under the threshold $\epsilon = 0.95$.

By Algorithm 2.1, the raw data set is simplified. Moreover, the simplified data set is updated by on-line approach when the 100 normal data and 10 faults are detected. As shown in Table 2, at the end of the detection, the number of left raw data is 358, it is obvious that the amount of data set is not increased unlimitedly due to the threshold $\varepsilon = 0.9$. There is no doubt that the on-line and update method proposed in this paper can surely reduce the cost of data storage and computation load. The detection results by the method in this paper and JIT method [12] are presented in Figures 13 and 14, respectively. Moreover, the embedded son

Table 2: Summary of parameters in Example 4.2.

Raw data	400
Threshold	0.99
Left data	358
Confidence level	99%
CL	0.0067
Normal data	100
Fault data	10
Updated data set	344

Figure 10: Detection result under the threshold $\epsilon = 0.6$.

figures in Figures 13 and 14 are used to emphasize the CL and the verification of training data and validation. From Figure 12, the CL is time-varying as the normal and fault data are identified. Note that since only normal data 33, 34, and 47 are mistaken for faults by the method proposed, whereas normal data 33, 34, 47, 6, and 62 are mistaken for faults by the JIT method [12], the better detection result is obtained in this paper using the original and update CL than the one in [12]. Here, the original CL means CL that is obtained in terms of simplified database by off-line approach. Comparatively speaking, the threshold determined subjectively in advance during calculating the sparse distance might partially degrade the performance of fault detection. Likewise, the tradeoff of storage cost and high detection performance is realized.

5. Case Study

In this section, an AL stack etch process was performed on a commercial scale Lam 9600 plasma etch tool at Texas Instrument, Inc. [9, 32, 33]. The data are taken from MACHINE_DATA, OES_DATA, and RFM_DATA during three experiments [33]. As pointed out by [9],
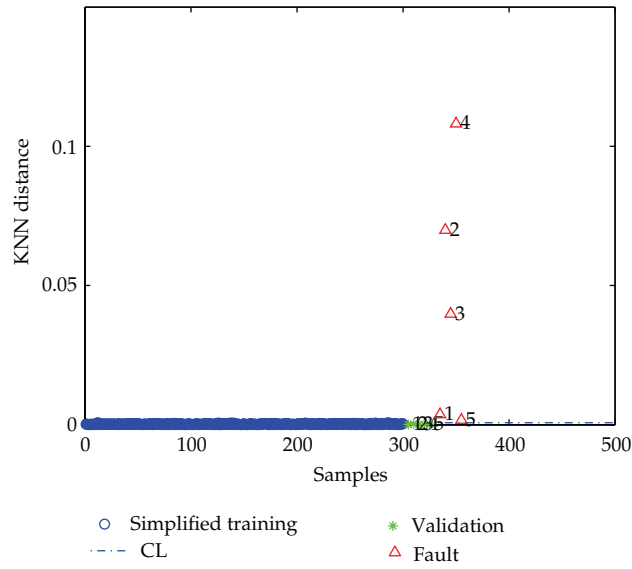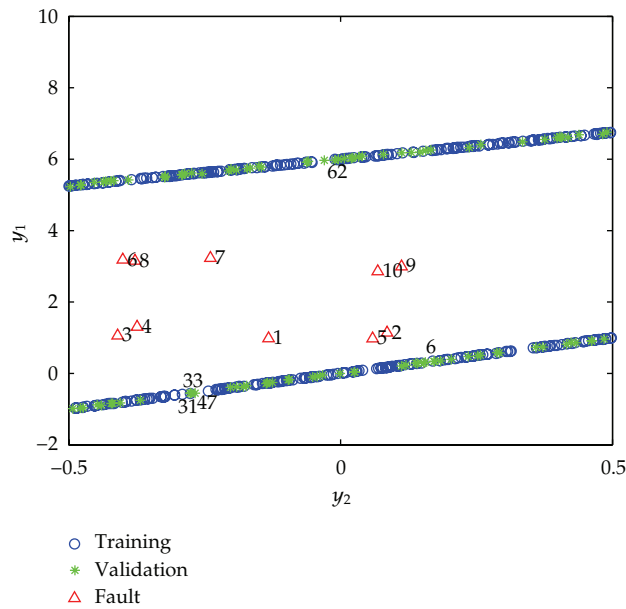
Figure 11: Fault detection by FD-KNN method.

Figure 12: Scatter plot of samples.

the unique characteristics associated with semiconductor process different from other production processes include the unequal batch duration, unequal step duration, and process drift and shift, therefore, data from the different experiments of the same resource have different mean and different covariance structures, which can be seen more clearly in the case of
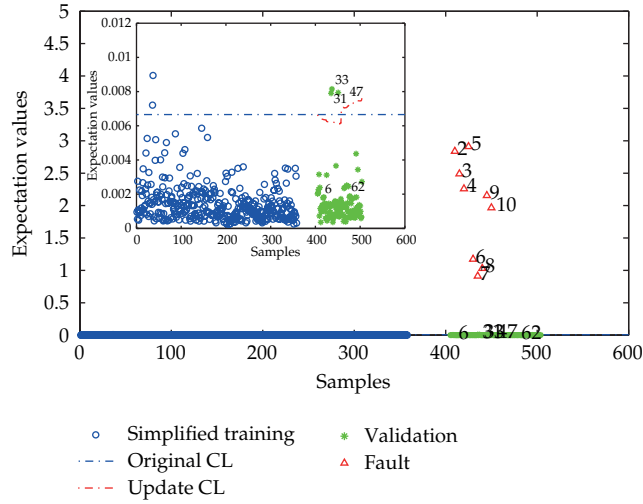
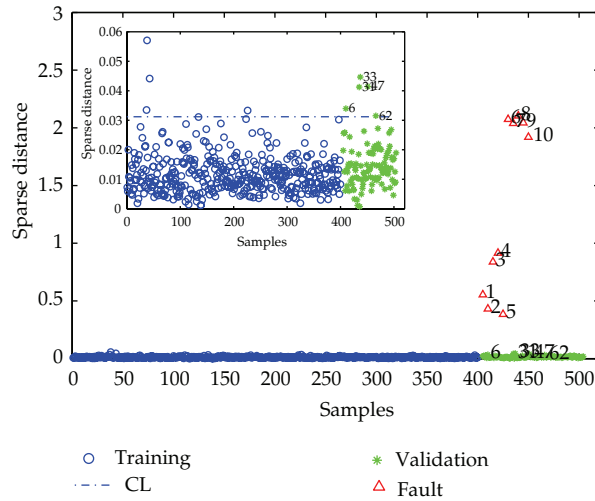**Figure 13:** Fault detection by the method proposed in this paper.

**Figure 14:** Fault detection by JIT method.

different resources. Due to the multimodal characteristic, the detection process based on the method proposed in this paper is as follows.

Step 1. Data preprocessing: we choose 10 normal batches and 10 process variables selected from the three data resources, such that the meaningful results can be obtained. Then, 30 normal batches are stored in the raw database. Further, the data is unfolded as the 2-D array by using the way in [9].

Step 2. The data set is simplified by Algorithm 2.1.

Step 3. The CL is computed by Algorithm 3.1.

Step 4. Another 15 normal batches are selected for the validation from the three different resources, in which 5 normal batches are used from each of resources. Moreover, 5 faults are also chosen from the intentionally induced during the experiments in the above
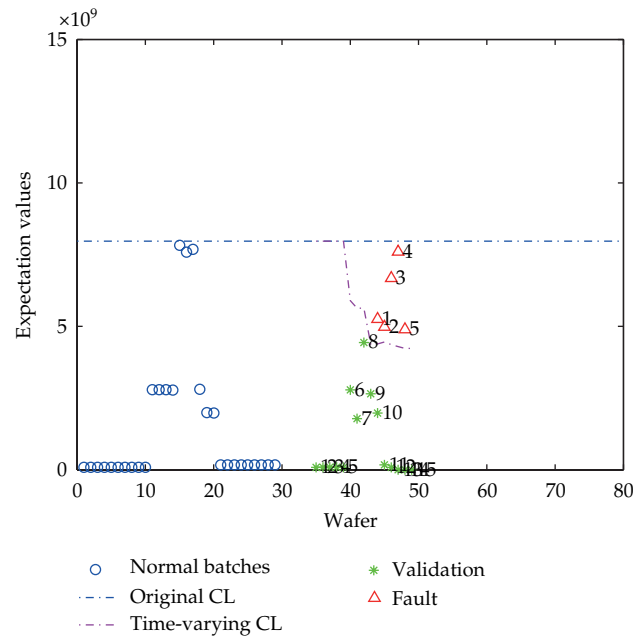
Figure 15: Fault detection for etch process.

resources for detection. Note that Algorithm 2.1 still can be used to undate the raw data set to produce the time-varying CL.

Following the aforementioned steps, threshold $\varepsilon = 0.9$ is determined in the detection for the etch process, and 29 normal batches can be obtained after Step 2. Moreover, the CL $7.9682e + 009$ is computed as original CL under confidence level 0.99. Known that if the query is normal, then it will be put into the simplified data set during identifying the normal data and fault data. It is worth pointing out the number of data in the database will be not surely increasing unlimitedly at the same time of updating the database due to Step 4. In this case, the final number of samples in the database is 36. The detection result is shown in Figure 15, where all of faults are identified correctly under the time-varying control limit by the method proposed in this paper. However, note that if raw data set is not updated, all of 5 faults can not be detected under the original CL computed. It is certain that some original faults with small data drift due to the effect of temperature and environment can be identified more easily in terms of the method of updating raw data set by online approach than the methods without applying the technology of update data set.

6. Conclusion

This research presented aims at highlighting the following two aspects: on one hand, the raw data set is simplified and updated by JIT approach based on the Mahalanobis distance between samples; on the other hand, combining the KNN rule with SPC method, the time-varying CL can be obtained to solve the nonlinear, multimodal, and data drift and shift problems existed in the practical case study. Numerical examples and an industrial case study show that the method proposed is an adaptive, flexible, and high-performance fault detection technique.

Acknowledgments

References

[1] M. Wang and T. Zhan, "Analysis of the fault attack ECDLP over prime field," *Journal of Applied Mathematics*, vol. 2011, Article ID 580749, 11 pages, 2011.

[2] Z. Gao, T. Breikin, and H. Wang, "Reliable observer-based control against sensor failures for systems with time delays in both state and input," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 38, no. 5, pp. 1018–1029, 2008.

[3] Z. Gao, X. Shi, and S. X. Ding, "Fuzzy state/disturbance observer design for T-S fuzzy systems with application to sensor fault estimation," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 38, no. 3, pp. 875–880, 2008.

[4] Z. Gao, X. Dai, T. Breikin, and H. Wang, "Novel parameter identification by using a high-gain observer with application to a gas turbine engine," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 4, pp. 271–279, 2008.

[5] X. B. Yang, X. Q. Jin, Z. M. Du, and Y. H. Zhu, "A novel model-based fault detection method for temperature sensor using fractal correlation dimension," *Building and Environment*, vol. 46, no. 4, pp. 970–979, 2011.

[6] L. F. Mendonça, J. M. C. Sousa, and J. M. G. Sá da Costa, "An architecture for fault detection and isolation based on fuzzy methods," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1092–1104, 2009.

[7] P. M. Frank, E. Alcorta García, and B. Köppen-Seliger, "Modelling for fault detection and isolation versus modelling for control," *Mathematics and Computers in Simulation*, vol. 53, no. 4–6, pp. 259–271, 2000.

[8] F. Li, B. R. Upadhyaya, and L. A. Coffey, "Model-based monitoring and fault diagnosis of fossil power plant process units using group method of data handling," *ISA Transactions*, vol. 48, no. 2, pp. 213–219, 2009.

[9] Q. P. He and J. Wang, "Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 20, no. 4, pp. 345–354, 2007.

[10] Q. P. He and J. Wang, "Principal component based k-nearest-neighbor rule for semiconductor process fault detection," in *Proceedings of the American Control Conference (ACC '08)*, pp. 1606–1611, Washington, USA, June 2008.

[11] J. Park, I. H. Kwon, S. S. Kim, and J. G. Baek, "Spline regression based feature extraction for semiconductor process fault detection using support vector machine," *Expert Systems with Applica-tions*, vol. 38, no. 5, pp. 5711–5718, 2011.

[12] K. Manabu, S. Takeaki, and H. Shinji, "Just-in-time statistical process control for flexible fault management," in *Proceedings of the SICE Annual Conference*, pp. 1482–1485, Taipei, Taiwan,, August 2010.

[13] Z. Gao and H. Wang, "Descriptor observer approaches for multivariable systems with measurement noises and application in fault detection and diagnosis," *Systems and Control Letters*, vol. 55, no. 4, pp. 304–313, 2006.

[14] Z. Gao, S. X. Ding, and Y. Ma, "Robust fault estimation approach and its application in vehicle lateral dynamic systems," *Optimal Control Applications and Methods*, vol. 28, no. 3, pp. 143–156, 2007.

[15] C. C. Feng, "Robust fault-tolerant control for systems with extended bounded-sensor-faults," in *Proceedings of the 17th World Congress of the International Federation of Automatic Control*, pp. 2582–2587, Seoul, Korea, 2008.

[16] D. M. Himmelblau, *Fault Detection and Diagnosis in Chemical and Petrochemical Processes*, Elsevier, Amsterdam, The Netherland, 1978.

[17] S. Simani and C. Fantuzzi, "Dynamic system identification and model-based fault diagnosis of an industrial gas turbine prototype," *Mechatronics*, vol. 16, no. 6, pp. 341–363, 2006.

[18] Z. Ge and Z. Song, "Online monitoring of nonlinear multiple mode processes based on adaptive local model approach," *Control Engineering Practice*, vol. 16, no. 12, pp. 1427–1437, 2008.

[19] C. Cheng and M. S. Chiu, "Nonlinear process monitoring using JITL-PCA," *Chemometrics and Intelligent Laboratory Systems*, vol. 76, no. 1, pp. 1–13, 2005.

[20] S. Yoon and J. F. MacGregor, "Statistical and causal model-based approaches to fault detection and isolation," *AIChE Journal*, vol. 46, no. 9, pp. 1813–1824, 2000.

[21] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: part I: quantitative model-based methods," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[22] S. Mahadevan and S. L. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *Journal of Process Control*, vol. 19, no. 10, pp. 1627–1639, 2009.

[23] D. Fischer, *Artificial Intelligence Techniques Applied to Fault Detection Systems [Ph.D. Dissertation]*, Electrical and Computer Engineering, McMaster University, Ontario, Canada, 2004.

[24] D. Dong and T. J. Mcavoy, "Nonlinear principal component analysis—based on principal curves and neural networks," *Computers and Chemical Engineering*, vol. 20, no. 1, pp. 65–78, 1996.

[25] H. G. Hiden, M. J. Willis, M. T. Tham, and G. A. Montague, "Non-linear principal components analysis using genetic programming," *Computers and Chemical Engineering*, vol. 23, no. 3, pp. 413–425, 1999.

[26] W. Ku, R. H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 30, no. 1, pp. 179–196, 1995.

[27] G. Bontempi, H. Bersini, and M. Birattari, "The local paradigm for modeling and control: from neuro-fuzzy to lazy learning," *Fuzzy Sets and Systems*, vol. 121, no. 1, pp. 59–72, 2001.

[28] R. D. Nowak, "Nonlinear system identification," *Circuits, Systems, and Signal Processing*, vol. 21, no. 1, pp. 109–122, 2002.

[29] X. Gao, G. Wang, and J. H. Ma, "An approach of building simplified multivariate statistical model based on mahalanobis distance," *Information and Control*, vol. 30, no. 7, pp. 676–680, 2001.

[30] A. Ison and C. J. Spanos, "Robust fault detection and fault classification of semiconductor manufacturing equipment," in *Proceedings of the Fifth International Symposium on Semiconductor Manufacturing*, pp. 1–4, Tokyo, Japan, October 1996.

[31] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive PCA for adaptive process monitoring," *Journal of Process Control*, vol. 10, no. 5, pp. 471–486, 2000.

[32] B. M. Wise, N. B. Gallagher, S. W. Butler, D. D. White, and G. G. Barna, "A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process," *Journal of Chemometrics*, vol. 13, no. 3-4, pp. 379–396, 1999.

[33] E. R. Inc, "Metal etch data for fault detection evaluation," 1999, http://software.eigenvector.com/Data/Etch/index.html.