Abstract: A new fault detection method is proposed to realize adaptive high-performance monitoring and efficient maintenance of the system. The proposed method is data-driven and called Just-In-Time Statistical Process Control (JIT-SPC). JIT-SPC focuses on the distance from the current operation data to the normal operation data stored in the database, because fault detection depends essentially on whether normal data exist near the current data or not. Since JIT-SPC is a model-free technique without impractical assumptions that conventional methods make, it has a potential for realizing practical, adaptive, high-performance monitoring. In addition, fault identification can be done through contribution plot in the framework of JIT-SPC. The usefulness of JIT-SPC and its contribution plot is demonstrated through a case study of the vinyl acetate monomer process. The results show that JIT-SPC can cope with changes in operating condition and can detect faults earlier than the conventional MSPC.

Keywords: Fault detection, Fault diagnosis, Multivariate statistical process control, Adaptive algorithms, Just-in-time modeling, Maintenance, Vinyl acetate monomer process.

1. INTRODUCTION

In order to operate manufacturing processes efficiently, it is crucial in any industry to detect malfunction of equipments and abnormal quality of products as early as possible, to identify the cause of such faults, and to take appropriate measures. Multivariate statistical process control (MSPC) has been widely accepted in industry. The conventional MSPC uses $T^2$ and $Q$ (or SPE) statistics derived through principal component analysis (PCA) (Jackson and Mudholkar (1979); Kresta et al. (1991); Kano and Nakagawa (2008)).

Although MSPC can cope with correlation among variables, it is difficult to apply the conventional MSPC technique to processes that manufacture various products and processes in which operating conditions or process characteristics are changed. This is because the conventional MSPC technique is based on the assumption that the process has one operating mode. To cope with multi-product processes and multiple operation modes, multi-model approaches have been proposed (Hwang and Han (1999); Zhao et al. (2004)). On the other hand, to cope with changes in process characteristics or operating conditions, recursive modeling approaches such as recursive PCA have been investigated by many researchers (Dayal and MacGregor (1997); Wang et al. (1997); Rannar et al. (1998); Li et al. (2000); Lee and Vanrolleghem (2004); Choi et al. (2006)).

Obviously, a problem of the multi-model approach is the cost of building and maintaining many models. On the other hand, the recursive modeling approach cannot adapt models to abrupt changes in process characteristics. Such changes are usually caused just after the maintenance of production equipments. In such a case, the deterioration of fault detection performance is unavoidable.

Although a lot of researches have been performed in the area of fault detection, the industry is not satisfied with the existing methods. This is because a lot of advanced methods use complicated modeling methods or impractical assumptions. Therefore, the present work has focused on how we can realize a simple, intuitive, and high-performance fault detection system. The basic concept of SPC is to detect a fault by judging whether normal data exist near the current data or not. Statistically, this judgment is based on the probability density of normal data at the point where the current data exist. In general, however, the estimation of probability density function in multi-dimensional space is very difficult. Thus, in the present work, in order to avoid estimating the multi-dimensional probability density function and to cope with the nonlinear and time-varying nature of processes, a new fault detection method based on the Just-In-Time (JIT) technique is proposed.

Several JIT fault detection methods have been proposed (Cheng and Chiu (2005); Ge and Song (2008)). However, they are not simple and have the same problems that conventional methods have, because they build local models by using PCA or support vector regression (SVR). On the other hand, a fault detection method based on the nearest-neighbor method has been proposed (Inoue and Yamamoto (2006)). This method is simple and intuitive, but statistical characteristics of the method is not investigated and the procedure after fault detection, i.e. fault isolation or fault identification, is not investigated.
In this research, an adaptive and high-performance fault detection method, referred to as Just-In-Time Statistical Process Control (JIT-SPC), is proposed. In JIT-SPC, the sparse distance is introduced to avoid estimating the probability density function in multi-dimensional space and to judge whether normal data exist near the current data. In addition, in order to achieve higher performance, a technique for adapting the control limit to operating condition is also proposed. Furthermore, contribution plot for JIT-SPC is proposed for fault identification. The usefulness of JIT-SPC and its contribution plot is demonstrated through a realistic case study of the vinyl acetate monomer process.

2. JUST-IN-TIME STATISTICAL PROCESS CONTROL

The proposed JIT-SPC focuses on the distance from the current operation data to the normal operation data stored in the database. The distance is calculated and checked every time when fault detection should be conducted. The judgment is based on the statistical test concept in the same way as conventional SPC. Thus, the proposed method is called JIT-SPC (Kano et al. (2010)).

2.1 Primitive Algorithm

First, the algorithm of JIT-SPC without adaptation of control limit is explained. This method is referred to as JIT-SPC(F).

The off-line procedure for building a fault detection system is as follows:

(1) Store normal operation data in a database. Set \( i = 1 \).
(2) Calculate distance between the \( i \)th sample and all the other samples stored in the database.
(3) Estimate the cumulative distribution function of the distance.
(4) Calculate the distance at which the cumulative distribution function reaches the threshold \( \phi_f \) determined in advance. This distance is called sparse distance \( D_s \).
(5) Set \( i = i + 1 \) and go to step 2 till \( i \) reaches the number of samples.
(6) Estimate the cumulative distribution function of \( D_s \).
(7) Set a control limit of \( D_s \) on the basis of its cumulative distribution function.

The above-mentioned procedure can be repeated and the control limit can be updated on-line. This update is straightforward because statistical modeling such as PCA is not necessary in the proposed method.

At step 2, L2 norm is used as distance in the following case study, but any distance can be used. In general, a method with lighter computational load is desirable because the computational load would become a problem in JIT techniques (Shigemori et al. (2011)).

At step 3, the kernel density estimation is used in the following case study to estimate the cumulative distribution function. When a query is normal, it is expected that many data exist near the query. On the other hand, there exist few data near the query when it is abnormal. Therefore, the cumulative distribution function should rise more sharply for normal queries than that for abnormal queries.

In order to evaluate the sharpness of the rise, the threshold is set at step 4 against the cumulative distribution function. For example, the sparse distance means the distance within which 2% of the population of normal operation data is included when the threshold is 0.02. Although this method based on the estimated cumulative distribution function is precise, the sparse distance can be defined as the distance within which 2% of the samples of normal operation data is included if the simplicity is more important than the accuracy.

The sparse distance becomes short in the area where normal samples exist densely; on the other hand, it becomes long in the area where normal samples exist sparsely. Therefore, faults can be detected by setting a control limit against the sparse distance. Thus, the cumulative distribution function of the sparse distance is estimated at step 6, and the control limit of the sparse distance is determined at step 7.

The on-line fault detection procedure is as follows:

(1) Calculate distance between the query and all the samples stored in the database.
(2) Estimate the cumulative distribution function of the distance, and calculate the sparse distance \( D_{sq} \).
(3) The query is abnormal if \( D_s \) is beyond its control limit.

2.2 Adaptation of Control Limit

In order to improve the performance of JIT-SPC(F), a technique for adapting its control limit to operating condition is proposed. This method is called JIT-SPC.

The on-line fault detection procedure of JIT-SPC is as follows:

(1) Calculate distance between any pairs of samples stored in the database. Calculate new distance when a new sample is obtained.
(2) Calculate distance between the query and all the samples stored in the database.
(3) Estimate the cumulative distribution function of the distance.
(4) Calculate the distance at which the cumulative distribution function reaches the threshold \( \phi_q \). This distance is called sparse distance of the query \( D_{aq} \).
(5) Select the samples located within \( D_{aq} \) from the query. Set \( i = 1 \).
(6) Estimate the cumulative distribution function of the distance between the \( i \)th sample, selected in step 5, and all the other samples stored in the database.
(7) Calculate the distance at which the cumulative distribution function reaches the threshold \( \phi_q \). This distance is called sparse distance of the dataset \( D_{sk} \).
(8) Set \( i = i + 1 \) and go to step 6 till \( i \) reaches the number of the selected samples.
(9) Estimate the cumulative distribution function of \( D_{sk} \).
(10) Set a control limit \( CL_{D_{sq}} \) on the basis of the cumulative distribution function of \( D_{sk} \).
(11) The query is abnormal if \( D_{aq} \) is beyond its control limit \( CL_{D_{sq}} \).
The major difference between two algorithms is that JIT-SPC can adaptively change its control limit on the basis of the probability density of normal data around the query.

### 2.3 Contribution Plot

In addition to fault detection, the concept of a contribution plot, which is well-known in the area of MSPC, can be used to identify measured variables that contribute to out-of-control signals (Nomikos (1996)). In the proposed system, contribution is determined as follows:

1. Select normal samples that exist within the sparse distance from the query.
2. Decompose the distance between the query and selected normal samples along each measured variable.
3. Calculate the sum of the decomposed distance for each measured variable. This sum is defined as contribution.

### 3. CASE STUDY

In order to validate the practicability of the proposed JIT-SPC(F) and JIT-SPC and compare them with the conventional MSPC, they are applied to the benchmark vinyl acetate monomer production plant (Yumoto et al. (2010); Seki et al. (2010)), whose process flow diagram is shown in Fig. 1.

The process consists of several basic unit operations, which include a vaporizer, a catalytic plug-flow reactor, a feed-effluent heat exchanger, a vapor/liquid separator, a gas compressor, an absorber, a CO$_2$ removal system, an azoetric distillation column with a decanter, and a buffer tank for the liquid feed. There are two major recycle loops, one for the gas reactant recovery and the other for the liquid reactant recovery. There are seven chemical components involved in this process. Ethylene (C$_2$H$_4$), oxygen (O$_2$), and acetic acid (HAc) are provided as fresh feeds and they are converted into vinyl acetate (VAc) with water (H$_2$O) and carbon dioxide (CO$_2$) as byproducts. In the fresh C$_2$H$_4$ stream, an inert component ethane (C$_2$H$_6$) is contained. The reactor effluent is sent to the separator, where gas and liquid are separated. The vapor from the separator goes to the compressor and the liquid stream becomes a part of the feed to the distillation column. The gas from the compressor is recycled back to the reactor through the absorber and the CO$_2$ removal system. The liquid product, VAc and water, are withdrawn from the decanter, and the bottom product, HAc and water, is recycled back through the HAc tank.

A rigorous dynamic simulator for this process was developed on the commercial software package Visual Modeler (Omega Simulation Co., Ltd.), which is based on rigorous first-principles models and has abundant industrial applications for operator training. The simulator employs pressure flow calculations and considers non-ideality in the process equipments, so that more realistic simulations are made possible than the conventional simulators. In addition, the developed simulator will be available in the public domain with a free limited license.

In this case study, two faults are investigated: valve stiction at the effluent liquid flow of the separator and increase of differential pressure in the distillation column. The samples stored in the database, representing the normal operating condition, are obtained when the product flow rate $F_{prod}$ is changed as shown in Fig. 2. In the simulation, reactor exit temperature $T$ is used to change the product flow rate, and feed flow rate $F_{C2H4}$ changes accordingly. In order to reduce computational load, the total number of samples in the database is reduced from 25000 to 10000. The sampling interval is one minute. The database and the

---

**Fig. 1.** Process flow diagram of vinyl acetate monomer production plant.

**Fig. 2.** Time series data obtained under normal operating condition.
Table 1. Parameters.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\phi_q$</th>
<th>$\phi_s$</th>
<th>UCL</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIT-SPC</td>
<td>$100/N$</td>
<td>$100/(N-1)$</td>
<td>max $D_{sk}$</td>
<td>1.025</td>
</tr>
<tr>
<td>JIT-SPC(F)</td>
<td>$0.001$</td>
<td></td>
<td></td>
<td>1.106</td>
</tr>
<tr>
<td>MSPC</td>
<td></td>
<td></td>
<td>99%</td>
<td>$0.954$</td>
</tr>
</tbody>
</table>

Table 2. Percentage of samples exceeding control limit under normal operating condition.

<table>
<thead>
<tr>
<th>Run</th>
<th>JIT-SPC</th>
<th>JIT-SPC(F)</th>
<th>$T^2$</th>
<th>$Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.06</td>
<td>0.21</td>
<td>0.98</td>
<td>0.64</td>
</tr>
<tr>
<td>2</td>
<td>0.18</td>
<td>0.10</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>3</td>
<td>0.58</td>
<td>0.51</td>
<td>0.00</td>
<td>0.05</td>
</tr>
<tr>
<td>4</td>
<td>0.16</td>
<td>0.53</td>
<td>0.73</td>
<td>0.83</td>
</tr>
<tr>
<td>5</td>
<td>0.47</td>
<td>0.06</td>
<td>0.00</td>
<td>0.10</td>
</tr>
<tr>
<td>6</td>
<td>0.16</td>
<td>0.00</td>
<td>0.00</td>
<td>0.15</td>
</tr>
<tr>
<td>7</td>
<td>0.11</td>
<td>0.18</td>
<td>0.33</td>
<td>0.23</td>
</tr>
<tr>
<td>8</td>
<td>0.32</td>
<td>0.09</td>
<td>0.30</td>
<td>0.38</td>
</tr>
<tr>
<td>9</td>
<td>0.43</td>
<td>0.01</td>
<td>0.00</td>
<td>0.06</td>
</tr>
<tr>
<td>10</td>
<td>0.34</td>
<td>1.12</td>
<td>0.44</td>
<td>0.36</td>
</tr>
<tr>
<td>Average</td>
<td>0.28</td>
<td>0.28</td>
<td>0.28</td>
<td>0.28</td>
</tr>
</tbody>
</table>

control limit of JIT-SPC are updated whenever new data is obtained.

For fair comparison, the percentage of samples exceeding the control limit in the normal operating condition is made equivalent for every method by adding a fixed value $\delta$ to the control limit. The operating condition is regarded as fault when the monitored index exceeds its control limit for two consecutive points. Control limits of JIT-SPC, JIT-SPC(F) and MSPC are set by using the database, in which the numbers of measured variables and samples are 93 and 10000, respectively. Table 1 shows parameters used in this case study. $\phi_q$ and $\phi_s$ are the thresholds in JIT-SPC, $N$ is the number of samples in the database, and $\phi_f$ is the threshold in JIT-SPC(F). UCL denotes upper control limit. The parameters $\phi_q$, $\phi_s$, $\phi_f$, and $\delta$ are determined by trial and error. The number of principal components (PCs) is corresponding to the number of eigenvalues larger than one.

3.1 Monitoring in Normal Operating Condition

First, JIT-SPC, JIT-SPC(F) and conventional MSPC are applied to 10 test data sets obtained under the normal operating condition. From Table 2, it is confirmed that the percentage of samples exceeding the control limit in the normal operating condition is made equivalent for every method.

Each test data set has 10000 samples, which are obtained when product flow rate is changed as shown in Fig. 3. This figure also shows the monitoring results. The dotted lines are control limits. In JIT-SPC, the difference $D_{sq} - CL_{D_{sq}}$ can be used for fault detection with the fixed control limit of zero.

In the normal operating condition, $D_{sq}$ of JIT-SPC, $D_s$ of JIT-SPC(F), and $T^2$ and $Q$ of MSPC should not exceed their control limits. However, $Ds$, $T^2$, and $Q$ exceed their control limits when product flow rate is changed.

3.2 Detection of Valve Stiction

In this scenario, JIT-SPC, JIT-SPC(F), and conventional MSPC are applied to 10 test data sets obtained when valve stiction occurs. The valve stiction means that the valve opening (valve position) does not follow its setpoint. The valve for separator liquid flow is stuck at 501 min, and the number of samples in each data set is 1500. The process is operated under the normal operating condition with different product flow rate until the valve stiction occurs.

Table 3 shows the fault detection results: time [min] needed to detect the fault in each data set and its average, i.e., average time to signal (ATS). ATS is similar to average run length (ARL), which is the average number of points that must be plotted before a point indicates an out-of-control condition (Montgomery (1997)). These two performance indicators have the following relation:

$$ATS = ARL \times (\text{sampling interval}) \quad (1)$$

The results show that JIT-SPC detects the fault earlier than JIT-SPC(F) and MSPC. In particular, ATS of JIT-SPC is much shorter than that of JIT-SPC(F); thus, the superiority of adaptive control limit over fixed control limit was confirmed.
Table 3. Fault detection results in the case of stiction of valve for separator liquid flow. Average time to signal (ATS) [min] is calculated from 10 data sets.

<table>
<thead>
<tr>
<th>Run</th>
<th>JIT-SPC</th>
<th>JIT-SPC(F)</th>
<th>$T^2$</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>58</td>
<td>95</td>
<td>200</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>63</td>
<td>84</td>
<td>120</td>
<td>64</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>43</td>
<td>78</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>76</td>
<td>128</td>
<td>310</td>
<td>84</td>
</tr>
<tr>
<td>5</td>
<td>85</td>
<td>104</td>
<td>138</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td>60</td>
<td>78</td>
<td>117</td>
<td>56</td>
</tr>
<tr>
<td>7</td>
<td>26</td>
<td>43</td>
<td>88</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>82</td>
<td>106</td>
<td>134</td>
<td>78</td>
</tr>
<tr>
<td>9</td>
<td>55</td>
<td>78</td>
<td>117</td>
<td>58</td>
</tr>
<tr>
<td>10</td>
<td>33</td>
<td>53</td>
<td>101</td>
<td>31</td>
</tr>
</tbody>
</table>

ATS 56.9 81.2 140.3 59.3

Fig. 4. Fault detection results in the case of stiction of valve for separator liquid flow.

The signals $D_{sq}$ and $D_{sq} - C_L D_{sq}$ of JIT-SPC, $D_s$ of JIT-SPC(F), and $T^2$ and $Q$ of MSPC are shown in Fig. 4. It should be noted that each plot is the result of one particular run and it does not represent the average performance of each technique.

3.3 Detection of Differential Pressure Change

In the next scenario, JIT-SPC, JIT-SPC(F), and conventional MSPC are applied to 10 test data sets obtained when differential pressure increases in distillation column. The differential pressure increases at 501 min, and the number of samples in each data set is 1500. The process is operated under the normal operating condition with different product flow rate until the fault occurs.

Table 4 shows the fault detection results; JIS-SPC(F) and $T^2$ of MSPC cannot detect the increase of differential pressure efficiently. In addition, JIT-SPC can detect the fault significantly earlier than $Q$ of MSPC. The signals $D_{sq}$ and $D_{sq} - C_L D_{sq}$ of JIT-SPC, $D_s$ of JIT-SPC(F), and $T^2$ and $Q$ of MSPC are shown in Fig. 5. Through these case studies, the superiority of JIT-SPC over the other methods was confirmed.

Table 4. Fault detection results in the case of differential pressure change in distillation column. Average time to signal (ATS) [min] is calculated from 10 data sets. n/a means no out-of-control signal is observed.

<table>
<thead>
<tr>
<th>Run</th>
<th>JIT-SPC</th>
<th>JIT-SPC(F)</th>
<th>$T^2$</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17</td>
<td>n/a</td>
<td>n/a</td>
<td>37</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>n/a</td>
<td>n/a</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>n/a</td>
<td>n/a</td>
<td>84</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>n/a</td>
<td>n/a</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>n/a</td>
<td>n/a</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>n/a</td>
<td>n/a</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>25</td>
<td>n/a</td>
<td>34</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>25</td>
<td>n/a</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>n/a</td>
<td>n/a</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>38</td>
<td>n/a</td>
<td>11</td>
</tr>
</tbody>
</table>

ATS 14.8 n/a n/a 30.3

Fig. 5. Fault detection results in the case of differential pressure change in distillation column.

3.4 Contribution Plot

Figs. 6 and 7 show the contribution plots when out-of-control signals are detected. In Fig. 6, the contributions of variables 8 and 34 are significant. These two variables are directly related to the cascade control system for the liquid level of the separator. The variable 8 is manipulated variable (MV) of slave flow controller for effluent liquid (FIC201), and the variable 34 is MV of master liquid level controller (LIC201). It is clear that this contribution plot suggests the valve for effluent liquid flow of the separator might be abnormal.

In Fig. 7, the contributions of variables 29 and 41 are significant. The variable 29 is bottoms flow rate of the distillation column, which is a controlled variable of FIC404, and the variable 41 is liquid level of the reflux drum of the distillation column, which is a controlled variable of LIC402. Both identified variables are directly related to the distillation column. Therefore, this contribution plot suggests that the operating condition of distillation column might be abnormal.

These results clearly show that the proposed contribution plot is useful for fault diagnosis when JIT-SPC is used.
Contribution
Variable
Fig. 6. Contribution plot in the case of stiction of valve for separator liquid flow.

Contribution
Variable
Fig. 7. Contribution plot in the case of differential pressure change in distillation column.

4. CONCLUSIONS

In the present work, JIT-SPC was proposed to realize adaptive high-performance fault detection, and the usefulness was demonstrated through a practical case study. The results have shown that JIT-SPC can detect faults earlier than JIT-SPC(F) and conventional MSPC and also that contribution plot for JIT-SPC is useful for fault diagnosis.

ACKNOWLEDGEMENTS

The simulator used in this research was developed by the Workshop No.27 (Process Control Technology) of the Japan Society for the Promotion of Science (JSPS) 143rd Committee on Process Systems Engineering. The authors express their special gratitude to Mr. Shigeki Ootakara of Mitsui Chemicals Inc. and Mr. Takamasa Yamoto of Omega Simulation Co. Ltd. for their contributions to the development.

REFERENCES


