ELSEVIER

Chemical Engineering Science 61 (2006) 6109–6132

Chemical Engineering Science

www.elsevier.com/locate/ces

Online fault diagnosis and state identification during process transitions using dynamic locus analysis

Rajagopalan Srinivasan*, Ming Sheng Qian

Laboratory for Intelligent Applications in Chemical Engineering, Department of Chemical and Biomolecular Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore

Received 12 February 2005; received in revised form 22 May 2006; accepted 28 May 2006
Available online 6 June 2006

Abstract

Chemical plants operate in a variety of states; some of these are steady states while others including grade changes, startup, shutdown, and maintenance operations are transitions. Transition operations are usually challenging and more prone to abnormalities. Therefore, automated process monitoring during transitions is important. In this paper, we propose a new signal comparison-based approach, called dynamic locus analysis, for online state identification and fault diagnosis during process transitions. Dynamic locus analysis is an extension of Smith and Waterman's [1981. Identification of common molecular subsequence. Journal of Molecular Biology 147, 195–197] discrete sequence comparison algorithm to continuous signals. It uses dynamic programming to efficiently identify the portion of a long reference signal that best matches another signal. During online application, signals from real-time sensors are compared with those from prior process runs to identify the current process state as well as estimate its progress. Run-to-run variations between the reference and online signals are accounted for by using dynamic time warping (DTW) for signal comparison. Dynamic locus analysis can be directly used for multivariate temporal signals and has the computational efficiency needed for real-time application. Extensive testing on three case studies—the Tennessee Eastman challenge problem, a lab-scale distillation column, and a simulated fluidized catalytic cracking unit—reveal that the proposed method can quickly identify normal as well as abnormal process states.

Keywords: Process supervision; Abnormal situation management; Signal comparison; Dynamic time warping; Unsteady state operation; Sequence comparison

1. Introduction

Modern chemical plants are large in scale and highly complex. Due to market forces, increasingly, these plants are operated in a multitude of states and frequently change between them to accommodate varying raw materials, product grades, demands, etc. Transitions also arise during startup, shutdown, and maintenance activities. Although essential, transitions are uneconomical states and there is a constant push to decrease the duration of the transition. Operator involvement in the operation of the plant is also higher through the course of transitions; consequently the occurrence of human error is higher during these periods (Nimmo, 1993). Since present day process automation applications like alarm management and advanced control systems are usually configured for a single operating state—typically a steady state mode—and these do not function effectively during transitions (Srinivasan et al., 2005a), operators do not have decision support applications during transitions. There is therefore a need to develop intelligent systems that can detect occurrence of faults during multiple state operations. Timely and accurate fault diagnosis would enable the operator to correct errors and recover from abnormal situations which may otherwise compromise the safe and economic functioning of the plant.

Fault diagnosis has received a lot of attention in literature (Jain et al., 2000; Venkatasubramanian et al., 2003). However, most online fault detection and diagnosis methods in the literature assume that the process is in a known, well characterized state (e.g., a normal steady state) when diagnosis is started. This assumption becomes untenable for agile processes that undergo transitions. The following are some key challenges during online fault diagnosis and state identification

* Corresponding author. Tel.: +65 65168041; fax: +65 67791936.
E-mail address: chergs@nus.edu.sg (R. Srinivasan).

0009-2509/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ces.2006.05.037

during process transitions:

(1) Unknown initial state: since agile processes are frequently in dynamic states, the operator cannot assume that the process is in an a priori known state when diagnosis is started. It is necessary to first identify the current state of the process from the online signals before evaluating if the process is in a normal condition from that state.
(2) Partial information: since state identification needs to be performed as early as possible without having to wait for the current state to end or a new state to begin, the method should work even if it is initiated in the middle of a transition and based on incomplete information such as unknown start times.
(3) Run-to-run variations: the method should be robust to changes in process signals induced by different strategies of different operators. Signal synchronization is therefore critical.
(4) Process complexities: transitions involve large changes which bring forth process nonlinearities, discontinuities, non-stationary signals, as well as variable time-delays.
(5) Low computation cost: the computational load should be modest, so that the method can be deployed online even for large-scale processes with thousands of sensors.

In this paper, we propose a signal comparison-based approach to state identification and fault diagnosis during process transitions. We use a short segment of online sensor signals as a fingerprint and develop a methodology inspired by sequence comparison approaches in bioinformatics to quickly and computationally inexpensively identify the current process state. The rest of this paper is organized as follows: we review previous research in process state identification and signal comparison next. In Section 2, we describe the dynamic programming-based discrete sequence comparison algorithm that provides the motivation for this work. The dynamic locus analysis methodology for online state identification is proposed in Section 3. In Section 4, we illustrate the proposed approach using three case studies—the Tennessee Eastman (TE) challenge process, a lab-scale distillation column, and a simulation of a fluidized catalytic cracking unit. Conclusions from this work are presented in Section 5.

1.1. *Previous work in process monitoring and fault diagnosis*

Due to significant advances in data collection and storage, vast amount of historical data is becoming commonly available. This data is a rich source of information about the process that can be used to improve plant operation. Multivariate statistics such as principal components analysis (PCA) have been widely used for process data classification, fault detection, and diagnosis (Chiang and Braatz, 2003; Kano et al., 2001; Chen and Liao, 2002). PCA reduces the dimensionality of data with minimum loss of information by projecting high dimensional data onto uncorrelated vectors. The projections are chosen so that the maximum amount of information, measured in terms of its variability, is retained in the smallest number of dimensions.

Krzanowski (1979) defined the PCA similarity factor for measuring the similarity of different groups of data. The PCA similarity factor compares the reduced subspaces:

$$S_{\text{PCA}} = \frac{1}{k} \sum_{p=1}^{k} \sum_{q=1}^{k} \cos^2 \theta_{pq} \tag{1}$$

where θ_{pq} is the angle between the pth principal component of dataset X and qth principal component of dataset Y. Raich and Cinar (1997) used the PCA similarity factor for diagnosing process disturbances. Singhal and Seborg (2002) modified the PCA similarity factor by weighing each principal component with the square root of its eigenvalue, λ.

$$S_{\text{PCA}}^{\lambda} = \frac{\sum_{p=1}^{k} \sum_{q=1}^{k} \lambda_p^S \lambda_q^{\text{T}} \cos^2 \theta_{pq}}{\sum_{p=1}^{k} \lambda_p^S \lambda_p^{\text{T}}}. \tag{2}$$

A major limitation of the classical PCA-based approaches is that the PCA model is time invariant. A number of modifications have been developed to overcome this limitation. Nomikos and Macgregor (1994) presented a multi-way PCA method which organizes time-varying data from multiple runs first into a time-ordered three-dimensional array. The array is then unfolded into a two-dimensional matrix, and a statistical model for the deviation of process variables between runs is built. Srinivasan et al. (2004) proposed a dynamic PCA-based similarity factor $S_{\text{DPCA}}^{\lambda}$ for comparing non-stationary signals that accounts for the temporal evolution of signals. One strong assumption of PCA-based methods is that all batches have equal duration and all are synchronized.

Another family of data-driven approaches use signal comparison in order to overcome this. These approaches are based on the precept that the same types of faults or disturbances show similar features in the process signal. This assumption is satisfied by most common normal and faulty operating states in chemical processes (excluding unstable operating states such as sustained high frequency oscillations). By comparing the online signal with a database of signals corresponding to the different fault classes, any fault in the process can be identified. The challenge in these methods is that it is normal for two similar signals to be slightly different and not match each other perfectly. This synchronization problem can be overcome using dynamic time warping (DTW).

Let X and Y denote two time-sampled signals of lengths m and n, and let j and i denote the time index of their trajectories, respectively. DTW finds a sequence F^* of P points on an n^*m grid such that a total distance measure between the two trajectories is minimized as shown in Fig. 1.

$$F^* = \{c(1), c(2), \ldots, c(p), \ldots, c(P)\}, \tag{3}$$

$$c(p) = [i(p)\, j(p)]. \tag{4}$$

The minimum normalized distance $D^*(n, m)$ between the signals is found by warping their time axis which can be

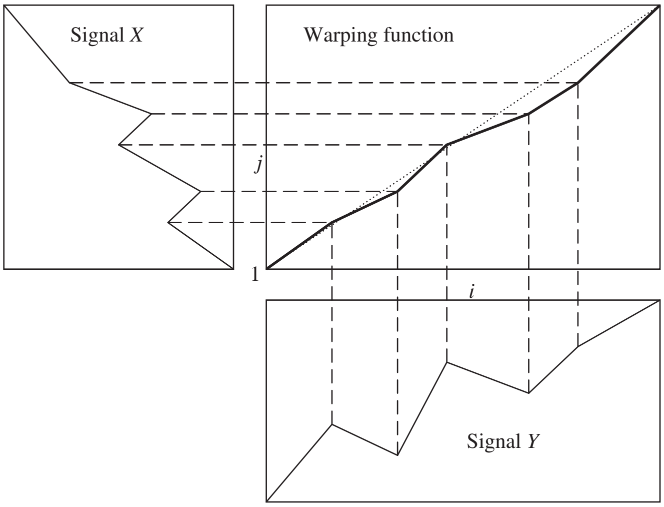

Fig. 1. DTW of signal T on signal R.

formulated as

$$D^*(n,m)=\min_F[D(n,m)], \tag{5}$$

$$D(n,m)=\frac{1}{N(w)}\sum_{p=1}^{P} d[i(p),j(p)]^* w(p). \tag{6}$$

Here, $D(n,m)$ is the normalized total distance between the two signals, $d[i(p),j(p)]$ is the local distance between the point $j(p)$ of Y and point $i(p)$ of X, $w(p)$ is a nonnegative weight coefficient, and $N(w)$ (usually $=\sum_{p=1}^{P} w(p)$) a normalization factor. $w(p)$ provides the flexibility to differently weigh horizontal and vertical steps in the DTW path. The optimal path $F^*$is found as $F^*=\mathrm{argmin}_F\,[D(n,m)]$ subject to constraints derived from the physics of the problem. One such constraint is the Itakura local constraint (Itakura, 1975) which defines the set of predecessors allowed for (i,j) as $(i-1,j)$, $(i-1,\ j-1)$, and $(i-1,\ j-2)$ and results in a local slope in [1/2 2]. DTW is thus reduced to the optimization algorithm:

$$D_A(i,j)=\min\left\{\begin{matrix} D_A(i-1,j)+d(i,j) \quad or \\ [\infty\ if\ Condition(A^*)] \\ D_A(i-1,j-1)+d(i,j) \\ D_A(i-1,j-2)+d(i,j) \end{matrix}\right\}. \tag{7}$$

Condition (A*) indicates that the predecessor of point $(i-1,j)$ is the point $(i-2,j)$.

DTW has been used for fault detection and diagnosis in chemical processes by Kassidas et al. (1998). However, DTW is computationally intensive (in both time and memory) and is seldom suitable for online signal comparison. To overcome these limitations, Colomer et al. (2002) combined DTW with qualitative representation of signals. Each signal was first decomposed into episodes which provided a higher-level representation of the signal. DTW was then used to find the optimal match between the episodes of the two signals. Srinivasan and Qian (2005) augmented DTW with landmarks in the signal, called singular points, to minimize the search space and improve the computational performance.

Approaches based on trend analysis adopt a different strategy to improve the computational performance of signal comparison. Rather than compare the raw signal, their abstraction based on qualitative features—such as increasing trend, decreasing trend, etc—is analyzed. Rengaswamy and Venkatasubramanian (1995) used syntactic pattern recognition methods to compare the trends and identify abnormal situations during steady state operations. As an extension to multi-state operations, Sundarraman and Srinivasan (2003) proposed the enhanced trend analysis approach which considers additional semiquantitative features such as duration and magnitude of trends. However, the qualitative abstraction results in loss of information, subsequently the time resolution of state identification results becomes coarser. During online state identification, we are interested in finding the segment of a long reference signal that is most similar to a given real-time signal. This is similar to the bioinformatics problem of identifying the maximally homologous (similar) subsequence from a set of long discrete sequences. In the following, we outline a popular approach for this problem.

2. Dynamic programming approaches to discrete sequence comparison

The discrete sequence comparison problem is generally formulated as follows: given two long (molecular) sequences, find a pair of segments—one from each sequence—such that there are no other pair of segments with greater similarity. The search seeks not only contiguous subsequences but also allows for small variations among the two, including mismatches and insertion/deletions.

Several heuristic (Needleman and Wunsch, 1970) as well as mathematically rigorous approaches have been proposed in the literature. One such is the dynamic programming approach of Smith and Waterman (1981). Let $A=\{a_1,a_2,a_3,\ldots,a_n\}$ and $B=\{b_1,b_2,b_3,\ldots,b_m\}$ be the two sequences to be compared. A similarity measure between sequences elements a and b is defined as $s(a,b)$, where $s(a,b)>0$ if $a=b$ and $s(a,b)<0$ for at least some cases of $a\neq b$. Insertions or deletions of length k receive weight of $-w_k$.

To find parts of segments with high degree of similarity, we setup a matrix H whose values $H_{i,j}$ are the maximum similarity of two segments *ending* in a_i and b_j, respectively. The similarity algorithm is started with

$$H_{i,0}=H_{0,j}=0, \quad 1\leqslant i\leqslant n,\quad 1\leqslant j\leqslant m. \tag{8}$$

Other elements of H are calculated as

$$H_{i,j}=\max\{0;\ S(a_x a_{x+1}\ldots a_i, b_y b_{y+1}\ldots b_j)\},$$
$$1\leqslant x\leqslant i,\quad 1\leqslant y\leqslant j$$

which can be rewritten in recursive form as

$$H_{i,j}=\max\{0,\ H_{i-1,j-1}+s(a_i,b_j),\ F_{i,j},\ G_{i,j}\}, \tag{9}$$

where

$$F_{i,j} = \max_{1 \leqslant k \leqslant i} \{H_{i-k,j} - w(k)\}, \tag{10}$$

$$G_{i,j} = \max_{1 \leqslant k \leqslant j} \{H_{i,j-k} - w(k)\}. \tag{11}$$

In the above, $H_{i,j}$ allows for the various possibilities for ending the segments at any a_i and b_j. $H_{i-1,j-1} + s(a_i, b_j)$ considers the case where a_{i-1} and b_{j-1} have been associated previously and a_i and b_j with similarity $s(a_i, b_j)$ are being associated; $F_{i,j}$ and $G_{i,j}$ consider the possibilities of deletions in sequence A and sequence B, respectively. Finally, the zero is included in (9) to prevent similarity from becoming negative and indicates no similarity between a_i and b_j.

The pair of segments with maximum similarity is found by first locating the maximum element of H. The other matrix elements leading to this maximum are then sequentially traced back until an element of H with value 0 is found. This procedure thus identifies the maximal similarity segment as well as produces the corresponding alignment. The pair of segments with the next best similarity can be found by applying the same procedure to the second largest element of H not associated with the first trace back. Waterman and Eggert (1987) extended the above algorithm to identify all non-intersecting similar subsequences with similarity above a pre-specified threshold.

Next, we illustrate the above procedure with a simple example. Consider the comparison of two DNA sequences A = AAUGCCAUUGACGG and B=CAGCCUCGCUUAG. In this example, we define $s(a_i, b_j) = 1$ if $a_i = b_j$ and $s(a_i, b_j) = -\frac{1}{3}$ otherwise. $w_k = 1 + \frac{k}{3}$. The Hmatrix shown in Table 1 is constructed following (8)–(11). The maximal value of 3.3 indicates that the matching ends at $(a_{10}, b_8)$and matching segments are GCCAUUG and GCCUCG as highlighted in the table. It can be noted that although the two segments differ through a missing element and a mismatch (fourth and sixth positions in A), this segment has the maximum match among all possible segments of a and b. This algorithm provides not only a mathematically rigorous basis for searching for maximally similar segments, but it can be efficiently programmed with low computational complexity.

In this paper, we extend the above discrete sequence comparison algorithm to the *continuous* domain and online signal comparison. Real-time fault diagnosis and state identification are shown to be equivalent to locating the best match of a short signal segment derived from real-time sensor readings among a number of long historical reference signals. The reference signal that yields the minimal difference reveals the process state (e.g., normal vs. abnormal, identity of transition, etc) as well as the extent of progression (from the relative position in the reference signal).

3. Dynamic locus analysis

Consider the signal segment $X = \{x_1, x_2, x_3, \ldots, x_m\}$ which is the last m samples from an online sensor. This signal has to be compared with a collection of K reference signals to identify the one that best matches X. Let $Y = \{y_1, y_2, y_3, \ldots, y_n\}$ be one of the reference signals. Here m is called the evaluation window. $m \ll n$ since only a small segment of the online signal is used for state identification.

Definition. For every segment of Y, say $Z = \{y_l, y_{l+1}, \ldots, y_j\}$, a segment $Z^* = \{y_{l^*}, y_{l^*+1}, \ldots, y_{j^*}\}$, is called the locus of X *iff* $D^*(X, Z^*) = \min_{l,j}\{D^*(X, Z)\}$. Also, y_{l^*} is called the *corresponding point* of x_1 and y_{j^*} the corresponding point of x_m.

During online state identification, since the real-time signal is generated dynamically, it is incomplete and its corresponding start time vis-à-vis the reference signal is not known a priori. Therefore, two aspects are essential in identifying the current state of a process—(1) finding the corresponding start point in the reference signal (that is, linking x_1 to y_{l^*}); and (2) comparing the real-time signal segment and every possible segment in the reference signal (that is, calculating the difference between two segments). The brute force approach of considering each possible starting point and performing an independent comparison will result in an unacceptable computation load. We devise an efficient approach using a concept similar to the discrete sequence comparison algorithm above so that the computational performance is suitable even for real-time usage.

Let i and j be the time index of X and Y, respectively. The locus of X is identified using a *dissimilarity matrix*, D_S. The (i, j) element of D_S measures the minimal difference between the subsegment $\{x_1, x_2, x_3, \ldots, x_i\}$ in X and the subsegment $\{y_l, y_{l+1}, y_{l+2}, \ldots, y_j\}$ in Y. In the general case, l is unknown and is determined using dynamic programming.

$$D_S(i, j) = \min_F \left\{ \sum_{d=1}^{i} \Delta(x_d, y_{j(d)}) \right\},$$

$$c(d) = (d, j(d)),$$

$$F = \{c(1), c(2), \ldots, c(i)\}, \tag{12}$$

where F is a sequence of matches between X and Z. $y_{j(d)}$ is the time warped point that matches with x_d, and $\Delta(x_d, y_{j(d)})$ is the difference between x_d and $y_{j(d)}$. While a variety of metrics can be used to calculate Δ, in this paper we use the Euclidean metric

$$\Delta(x_i, y_j) = |y_j - x_i|. \tag{13}$$

Note that $D_S(i, j)$ is not the total minimum distance between $\{x_1, x_2, x_3, \ldots, x_i\}$ and $\{y_1, y_2, \ldots, y_j\}$, rather it is the total minimum distance between X and its locus in Y.

The optimal match between X and the locus in Y is given by$D_S(m, j(m))$. As mentioned earlier, there are two problems in calculating $D_S(m, j(m))$. First, the point in Y that corresponds to x_1 is not known a priori. Second, X and Y would not match exactly due to noise and run-to-run differences. Therefore, there can be synchronization and magnitude differences between X and Y. We solve the former using Smith and Waterman's (1981) dynamic programming approach and the latter through DTW.

Table 1
H matrix for comparing sequences A = AAUGCCAUUGACGG and B = CAGCCUCGCUUAG

	Δ	C	A	G	C	C	U	C	G	C	U	U	A	G
Δ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
A	0.0	0.0	1.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.7
U	0.0	0.0	0.0	0.7	0.3	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.7
G	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.7	1.0	0.0	0.0	0.7	0.7	1.0
C	0.0	1.0	0.0	0.0	2.0	1.3	0.3	1.0	0.3	2.0	0.7	0.3	0.3	0.3
C	0.0	1.0	0.7	0.0	1.0	3.0	1.7	1.3	1.0	1.3	1.7	0.3	0.0	0.0
A	0.0	0.0	2.0	0.7	0.3	1.7	2.7	1.3	1.0	0.7	1.0	1.3	1.3	0.0
U	0.0	0.0	0.7	1.7	0.3	1.3	2.7	2.3	1.0	0.7	1.7	2.0	1.0	1.0
U	0.0	0.0	0.3	0.3	1.3	1.0	2.3	2.3	2.0	0.7	1.7	2.7	1.7	1.0
G	0.0	0.0	0.0	1.3	0.0	1.0	1.0	2.0	3.3	2.0	1.7	1.3	2.3	2.7
A	0.0	0.0	1.0	0.0	1.0	0.3	0.7	0.7	2.0	3.0	1.7	1.3	2.3	2.0
C	0.0	1.0	0.0	0.7	1.0	2.0	0.7	1.7	1.7	3.0	2.7	1.3	1.0	2.0
G	0.0	0.0	0.7	1.0	0.3	0.7	1.7	0.3	2.7	1.7	2.7	2.3	1.0	2.0
G	0.0	0.0	0.0	1.7	0.7	0.3	0.3	1.3	1.3	2.3	1.3	2.3	2.0	2.0

In dynamic locus analysis, the optimal matching path between $\{x_1, x_2, x_3, \ldots, x_i\}$ and the corresponding part of *Y* must be found. The dissimilarity matrix is used for the purpose. The dissimilarity matrix is constructed based on the principle of optimality (Williams, 1970) using dynamic programming. Since the optimal search should allow for compression and elongation in *Y* relative to *X*, time warping with Itakura local constraint is used to synchronize *X* and *Y*. Here, *X* is considered as the key signal and *Y* as the warping signal. The main steps in dynamic locus analysis are

Step 1: Dissimilarity matrix initialization. The first column and row of the dissimilarity matrix are initialized at this stage. The first column of D_s considers all possible matches for x_1 in *Y*.

$$D_S(1, j) = \Delta(x_1, y_j), \quad j \in [1\ n]. \tag{14}$$

The first row of D_s is calculated based on the Itakura local constraint as follows:

$$D_S(2, 1) = D_S(1, 1) + \Delta(x_2, y_1),$$

$$D_S(i, 1) = \infty, \quad i \in [3\ m]. \tag{15}$$

Step 2: Matrix propagation. Other elements of D_S are constructed recursively using the Itakura local continuity constraint as follows:

$$D_S(i, j) = \min\{D_S(i-1, j-1) + \Delta(x_i, y_j), F_{i,j}, G_{i,j}\},$$
$$i \in [2\ m], \quad j \in [2\ n], \tag{16}$$

$$F_{i,j} = D_S(i-1, j-2) + \Delta(x_i, y_j), \tag{17}$$

$$G_{i,j} = D_S(i-1, j) + \Delta(x_i, y_j) \quad \text{or } \infty \text{ if } G^* \tag{18}$$

G^* indicates that the predecessor of point $(i-1, j)$ is the point $(i-2, j)$.

Since we seek the segment in *Y* that matches the complete *X*, the minimal distance between *X* and *Y* is obtained as the smallest value in the last column, i.e., the minimal distance is $D_S(m, j^*)$ where

$$j^* = \operatorname*{argmin}_j\{D_s(m, j)\}, \quad j \in [1\ n]. \tag{19}$$

In the following, the time at which $y(j^*)$ occurred in the reference signal is notated as τ_y and the time of $x(m)$ as τ_x.

From the principle of optimality (Myers et al., 1980), an optimal policy must follow two rules—(1) If F^* is an optimal path which goes through point (i, j), then the optimal path to point (i, j) is part of F^*, and (2) the optimal path to point (i, j) depends only on previous points. The above recursive relationship satisfies both the rules, therefore, here the *optimal* dissimilarity $D_S(i, j)$ is found since all possible predecessors are considered and the one that minimizes the total dissimilarity is selected. The local search space at each point is schematically shown in Fig. 2a and the resulting total search space for dynamic locus analysis is shown in Fig. 2b. It should be noted that the dynamic locus analysis search space for calculating $D_S(m, j^*)$is larger than DTWs search space for $D(X, Z^*)$.

Step 3: Tracing the locus of *X* in *Y*. To recreate the optimal path, we track the predecessor to each element in $D_S(i, j)$ while constructing D_S using the *parent matrix P*. Each element in $P(i, j)$gives the relative position of its predecessor.

$$P(i, j) = \begin{cases} 0 & \text{if } D_S(i, j) = G_{i,j}, \\ 1 & \text{if } D_S(i, j) = D_S(i-1, j-1) + \Delta(x_i, y_j), \\ 2 & \text{if } D_S(i, j) = F_{i,j}. \end{cases} \tag{20}$$

The optimal matching path is identified by tracing the parent matrix starting from (m, j^*) and recursively identifying the optimal predecessor $(m-1, \ j^* - P(m, j^*))$. Note that, because of the local continuity constraint, $j^* - 2 * m \leqslant l^* \leqslant j^* - \frac{1}{2} * m$.

Once the locus has been determined, the *normalized difference* η between *X* and *Y* can be calculated as

$$\eta = \frac{\sum_{d=1}^{n} |y_{j(d)} - x_d|}{n} \tag{21}$$

which is the DTW distance between *X* and its locus.

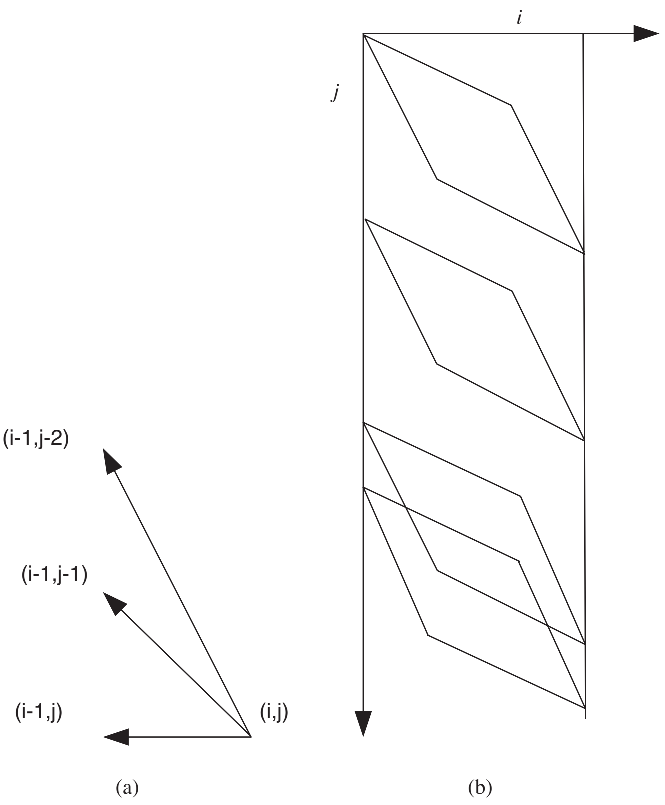

Fig. 2. Search space of (a) Itakura local constraint; (b) dynamic locus analysis.

Step 4: Online fault diagnosis and state identification. Once the locus of X in Y is known, the current state of the process can be determined. For this, consider a suitably annotated reference database that has historical data of common transitions in the process (such as startup, shutdown, grade change, etc) as well as from periods when the process underwent abnormal operations (due to faults, human errors, etc). During online monitoring, the real-time signal is compared with every reference signal in the database. If the best matching reference signal in the database corresponds to an abnormal event, the fault can be flagged and diagnosed using the associated annotations in the reference database. If the normalized difference of the best matching reference signal (i.e., one with the smallest η) is greater than a user defined threshold $\eta_{\max}$, the process can be said to be in an novel state. The specificity of online disturbance identification can be measured using *inseparability* α, which is defined as the ratio of the normalized differences of the best matching reference signal and that of the second-best one. A small value of α (≈ 0) implies that the real-time signal clearly matches a specific reference pattern while $\alpha \approx 1$ implies that the real-time signal cannot be clearly differentiated from two or more reference patterns. An online state can be identified when the inseparability decreases below a user-specified *minimum inseparability threshold* $\alpha_{\min}$.

The *coherence metric* indicates the smooth progression of the corresponding point in the reference signal as the process evolves in real-time. It is measured as the standard deviation of $\tau_y - \tau_x$ within the evaluation window. A coherence metric close to zero indicates a smooth progression of τ_y with respect to τ_x while a large value indicates that the real-time process does not match the reference signal. The application of these metrics is illustrated using case studies in Section 4.

3.1. Illustration of dynamic locus identification

In this section, we illustrate the dynamic locus analysis algorithm described above using the X and Y signals shown in Fig. 3.

Step 0: Signal normalization. Normalize X and Y to [0 1] based on the range of the sensor (Srinivasan et al., 2004).

$$x_i' = \frac{|x_i - x_L|}{|x_H - x_L|}, \tag{22}$$

where x_i' is the normalized variable value, and x_H and x_L are the high and low limits of the sensor range, respectively. In this example, the sensor range is [0 100]. The dissimilarity matrix is next calculated.

Step 1: Dissimilarity matrix initialization. The first column of D_S is calculated following (14). For example, $D_S(1,1) = |0.4070 - 0.3988| = 0.0082$ as shown in Table 2. At the end of this stage, $D_S(1,36) = |0.4070 - 0.4071| = 0.0001$ has the smallest value indicating that x_1 best matches y_{36}. The first row of D_s is calculated following (15). Thus, $D_S(2,1) = D_S(1,1) + |0.4310 - 0.3988| = 0.0404$. Also, $D_S(3,1) = D_S(4,1) = \cdots = D_S(8,1) = \infty$.

Step 2: Dissimilarity matrix propagation. All other elements of D_S are calculated using (16)–(18) which is similar to the Itakura local constraint in DTW. For example,

$$D_S(2,25) = \min\{D_S(1,24) + \Delta(x_2, y_{25}), G_{2,25}, F_{2,25}\}.$$

Here, $D_S(1,24) + \Delta(x_2, y_{25}) = 0.0167 + 0.0037 = 0.0204$, $G_{2,25} = D_S(1,25) + \Delta(x_2, y_{25}) = 0.0277 + 0.0037 = 0.0314$, and $F_{2,25} = D_S(1,23) + \Delta(x_2, y_{25}) = 0.0415 + 0.0037 = 0.0452$. Therefore $D_S(2,25) = 0.0204$ and $P(2,25) = 1$ as shown in Tables 2 and 3, respectively. All other elements in the second column are similarly calculated. This process is repeated for columns 3–8 as well.

The optimal match for Y is found from the last column. In this example, the minimum value in this column, i.e., the minimal distance between X and the locus of Y, is $D_S(8,32) = 0.0656$, so y_{32} is the corresponding point to x_8.

Step 3. Tracing the locus of X. The optimal path can then be identified from Table 3 starting from $P(8,32)$. As shown there, $P(8,32) = 1$, which implies that the optimal predecessor is (7, 31). By applying this recursively, the optimal path is $\{(1,24)\ (2,25)\ (3,26)\ (4,28)\ (5,28)\ (6,30)\ (7,31)\ (8,32)\}$ and $\{y_{24} \cdots y_{32}\}$ is the locus of X in Y. The search map is shown schematically in Fig. 4.

The reader may note from Table 2 that parts of X may better match other segments of Y. For example, from the fifth column, $D_S(5,12) = 0.0277$ and lower then $D_S(5,28) = 0.0474$. This indicates that the subsegment $\{y_9, y_{10}, y_{11}, y_{12}\}$ better matches with $\{x_1, \ldots, x_5\}$ as can be confirmed from Fig. 3. However, since $\{x_6, \ldots, x_8\}$ does not have a good match in the

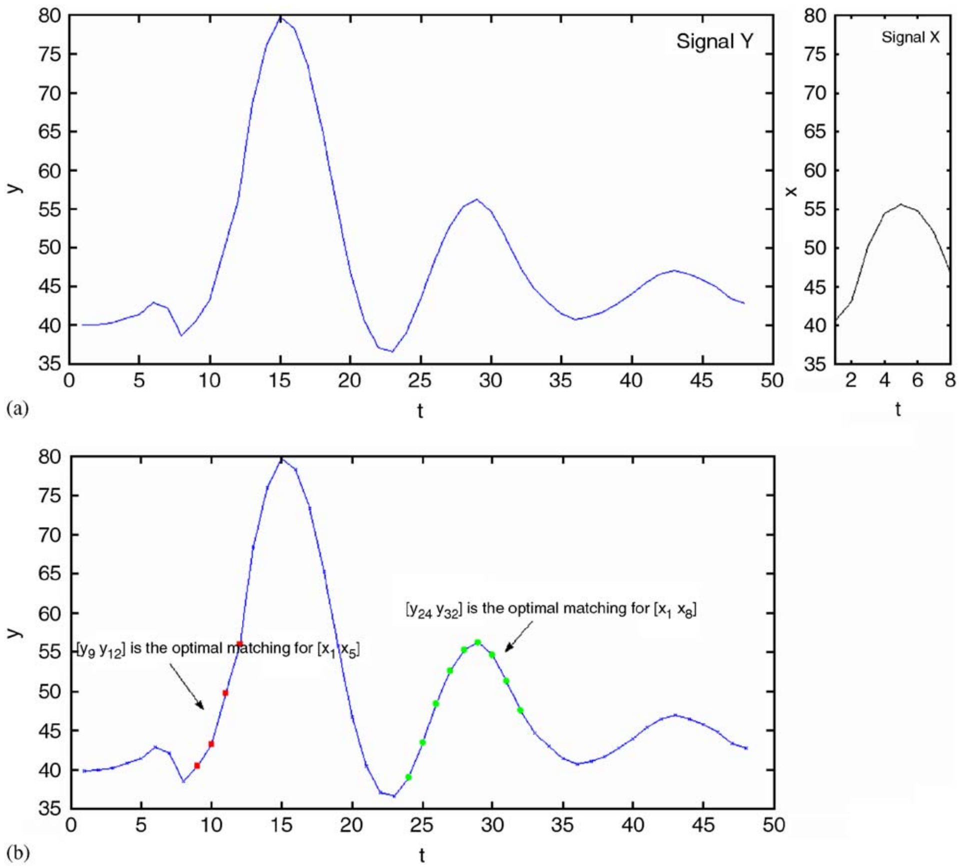

Fig. 3. (a) Reference signal Y and real-time signal X for illustrative example. (b) Corresponding points as identified by dynamic locus analysis.

neighborhood of y_{12}, this subsegment does not serve as the locus for the complete X. The best segment match from this has a dissimilarity of $D_S(8, 14) = 0.6201$ which is almost 10 times larger than $D_S(8, 32)$.

The *stability* of the locus identified by this method is also apparent from Table 2. The five next larger values in the final column of D_S are in the immediate neighborhood of $D_S(8, 32)$. Thus, even if there is significant local variation between X and Y due to noise, a similar segment in the same neighborhood will be identified as the locus. The stability and coherence of the dynamic locus analysis method is further illustrated in Section 4.

While the above example illustrated the dynamic locus analysis method using a single variable, the extension to the multivariate case is straightforward if X and Y are considered as Q-dimensional vectors of all the sensors to be monitored in the process. In the general case, different sensors can be weighted differently, so the distance metric of (13) has to be replaced by:

$$\Delta(x_i, y_j) = \sum_{c=1}^{Q} (w_c |y_{j,c} - x_{i,c}|), \tag{23}$$

where $x_{i,c}$ is the value of the cth) variable at time i; $y_{j,c}$ its value in reference signal Y at time j; and w_c the weight for the variable. In this paper, since all variables are normalized based on the sensor range (Eq. (22)), we use the same weight for all variables. Similarly, while the time warping method can be generalized to consider different warping for different variables, for multivariate cases we have selected to use the same warping for all variables and ignored cross-variable desynchronization in X. In the next section, we evaluate the dynamic locus analysis approach using three case studies.

4. Case studies

4.1. Case study 1: online disturbance identification in Tennessee Eastman plant

The Tennessee Eastman TE plant is a popular test bed for process control, fault diagnosis and signal comparison. In this section, data from this simulated plant is used to test the accuracy of the proposed method. The TE process produces two products (G and H) and a byproduct (F) from reactants A, C, D, and E. The process flowsheet is shown in Fig. 5. The process has five units: a two-phase reactor, a product condenser, a flash separator, a recycle compressor, and a product stripper. There are 53 variables in the TE plant: 22 of these are process measurements variables, 19 are component compositions, and 12 are process-manipulated variables. The closed-loop process simulator used here was developed by Singhal (2003) based on

Table 2
Dissimilarity matrix for illustrative example

$D_S(i,j)$	i	1	2	3	4	5	6	7	8
j	y_j x_i	0.4070	0.4310	0.4999	0.5441	0.5556	0.5477	0.5214	0.4677
1	0.3988	0.0082	0.0404	∞	∞	∞	∞	∞	∞
2	0.3996	0.0074	0.0385	0.1385	0.2831	∞	∞	∞	∞
3	0.4015	0.0055	0.0351	0.1366	0.2793	0.4372	0.5835	∞	∞
4	0.4080	0.0010	0.0240	0.1270	0.2631	0.4269	0.5666	0.6969	0.7566
5	0.4134	0.0064	0.0186	0.1051	0.2577	0.3998	0.5612	0.6691	0.7512
6	0.4287	0.0217	0.0033	0.0745	0.2205	0.3474	0.5188	0.6116	0.7081
7	0.4211	0.0141	0.0163	0.0821	0.1975	0.3319	0.4740	0.5742	0.6581
8	0.3855	0.0215	0.0596	0.1306	0.2331	0.3675	0.4941	0.6098	0.6564
9	0.4040	0.0030	0.0300	0.1555	0.2222	0.3491	0.4756	0.5914	0.6379
10	0.4319	0.0249	0.0039	0.0719	0.2428	0.3458	0.4616	0.5651	0.6009
11	0.4967	0.0897	0.0687	0.0072	0.0546	0.3017	0.3527	0.4863	0.5153
12	0.5601	0.1531	0.1540	0.0641	0.0232	0.0277	0.3141	0.3528	0.5787
13	0.6844	0.2774	0.3430	0.3385	0.1474	0.1519	0.1643	0.3273	0.5695
14	0.7605	0.3535	0.4826	0.6036	0.5549	0.2281	0.2405	0.4034	0.6201
15	0.7973	0.3903	0.6437	0.7800	0.5917	0.7966	0.4015	0.4402	0.6569
16	0.7829	0.3759	0.7054	0.9267	0.8424	0.7822	1.0174	0.6631	0.7186
17	0.7336	0.3266	0.6292	0.9391	0.9695	1.0204	0.9681	1.1803	0.9290
18	0.6529	0.2459	0.4678	0.7822	0.8911	1.0668	1.1257	1.0996	0.8483
19	0.5578	0.1508	0.2776	0.5257	0.5394	0.8932	0.9033	1.0045	1.0946
20	0.4675	0.0605	0.0971	0.3099	0.3865	0.6275	0.7076	0.9572	0.9574
21	0.4048	0.0022	0.0283	0.1921	0.3314	0.5373	0.6801	0.8242	0.8871
22	0.3704	0.0366	0.0628	0.1578	0.3315	0.5166	0.6939	0.8311	0.9215
23	0.3655	0.0415	0.0676	0.1627	0.3364	0.5215	0.6988	0.8360	0.9264
24	0.3903	**0.0167**	0.0575	0.1724	0.3117	0.4770	0.6740	0.8052	0.9086
25	0.4347	0.0277	**0.0204**	0.0856	0.2818	0.4028	0.5900	0.6767	0.8382
26	0.4844	0.0774	0.0701	**0.0359**	0.0957	0.3531	0.4164	0.6271	0.6437
27	0.5262	0.1192	0.1228	0.0467	0.0539	0.0833	0.3746	0.3794	0.6855
28	0.5525	0.1455	0.1988	0.1754	**0.0443**	**0.0474**	0.0881	0.1192	0.4642
29	0.5619	0.1549	0.2501	0.2609	0.1932	0.0506	0.0617	0.1022	0.2134
30	0.5463	0.1393	0.2546	0.2965	0.1776	0.1869	**0.0489**	0.0737	0.1808
31	0.5123	0.1053	0.1866	0.2670	0.2926	0.2209	0.0860	**0.0579**	0.1026
32	0.4754	0.0684	0.1128	0.2112	0.2799	0.3729	0.2592	0.1320	**0.0656**
33	0.4468	0.0398	0.0556	0.1659	0.2632	0.3887	0.3218	0.3339	0.0788
34	0.4302	0.0232	0.0240	0.1253	0.2392	0.3886	0.4904	0.4130	0.1696
35	0.4140	0.0070	0.0240	0.1099	0.2400	0.3808	0.5145	0.5978	0.3875
36	0.4071	0.0001	0.0240	0.1168	0.2469	0.3877	0.5214	0.6288	0.4736
37	0.4106	0.0036	0.0205	0.1099	0.2434	0.3850	0.5179	0.6253	0.6549
38	0.4162	0.0092	0.0149	0.0987	0.2378	0.3772	0.5165	0.6218	0.6769
39	0.4276	0.0206	0.0070	0.0793	0.2152	0.3432	0.4974	0.5912	0.6619
40	0.4396	0.0326	0.0177	0.0673	0.1719	0.3312	0.4394	0.5792	0.6074
41	0.4538	0.0468	0.0434	0.0638	0.1541	0.2737	0.3676	0.5070	0.5209
42	0.4650	0.0580	0.0666	0.0782	0.1429	0.2334	0.3564	0.4127	0.5097
43	0.4700	0.0630	0.0858	0.0965	0.1380	0.2236	0.3112	0.3626	0.4150
44	0.4650	0.0580	0.0921	0.1206	0.1573	0.2285	0.3062	0.3626	0.3653
45	0.4575	0.0505	0.0769	0.1345	0.1832	0.2554	0.3138	0.3702	0.3728
46	0.4478	0.0408	0.0575	0.1291	0.2170	0.2910	0.3284	0.3799	0.3825
47	0.4337	0.0267	0.0295	0.1237	0.2340	0.3388	0.3694	0.4161	0.4041
48	0.4269	0.0199	0.0240	0.1025	0.2196	0.3627	0.4118	0.4638	0.4569

the base control structure of McAvoy and Ye (1994). During the simulation, variables are sampled every minute. The 22 process measurements used in this paper for process state identification are shown in Table 4.

Here we use dynamic locus analysis to identify unknown process disturbances online and estimate their progress. Five disturbances, henceforth referred to as XD1–XD5, that affect the A feed flowrate, reactor pressure, reactor level, reactor temperature, and compressor work are considered. These are representative of setpoint changes or servo control problems in real industrial process operations. Different instances (runs) of the same disturbance class have different start times, duration, and magnitude. For example, during XD1-A, the flowrate of A feed from upstream is increased from the base case value of 0.25052 to 0.3902 kscm h (a 60% change) in three steps starting at $t = 51$ min as shown in Table 5(a). After the process recovers from these, the inverse change, decreasing the A feed flow is introduced at $t = 651$ min. The process is then allowed to return to a steady state. The effect on the A flow rate (XMEAS(1)) and the downstream pressures (XMEAS(13) and XMEAS (16)) is shown in Fig. 6. Two other instances XD1-B and XD1-C with changes of magnitude 55% and 50% were also introduced.

Table 3
Parent matrix for illustrative example

$D_S(i, j)$	i	1	2	3	4	5	6	7	8
j	$y_j x_i$	0.4070	0.4310	0.4999	0.5441	0.5556	0.5477	0.5214	0.4677
1	0.3988	1	0	–	–	–	–	–	–
2	0.3996	1	1	1	0	–	–	–	–
3	0.4015	1	0	2	0	1	0	–	–
4	0.4080	1	0	1	0	1	0	1	0
5	0.4134	1	1	0	1	0	1	0	1
6	0.4287	1	2	0	1	0	1	0	1
7	0.4211	1	2	1	1	0	1	0	1
8	0.3855	1	1	1	2	1	1	1	1
9	0.4040	**1**	0	1	2	2	2	2	2
10	0.4319	1	**1**	0	2	1	0	1	0
11	0.4967	1	2	**1**	0	1	0	1	0
12	0.5601	1	2	2	**1**	**0**	1	0	1
13	0.6844	1	2	1	2	1	**1**	**0**	1
14	0.7605	1	2	1	1	2	2	1	**1**
15	0.7973	1	2	1	2	1	2	2	2
16	0.7829	1	2	1	2	2	0	1	2
17	0.7336	1	0	1	2	1	1	0	1
18	0.6529	1	0	1	0	1	1	1	2
19	0.5578	1	0	1	0	1	0	2	0
20	0.4675	1	0	1	0	1	0	1	0
21	0.4048	1	0	1	0	1	0	1	0
22	0.3704	1	1	1	0	1	0	1	1
23	0.3655	1	2	2	1	2	1	2	2
24	0.3903	**1**	0	2	2	0	2	0	2
25	0.4347	1	**1**	0	1	0	1	0	1
26	0.4844	1	2	**1**	0	1	0	1	0
27	0.5262	1	2	2	1	0	1	0	1
28	0.5525	1	2	1	**2**	**0**	1	0	1
29	0.5619	1	2	1	1	1	1	0	1
30	0.5463	1	0	1	2	0	**2**	0	1
31	0.5123	1	0	1	2	1	2	**1**	0
32	0.4754	1	0	1	0	1	2	1	**1**
33	0.4468	1	0	1	0	1	2	1	2
34	0.4302	1	0	1	0	1	2	1	2
35	0.4140	1	0	1	0	1	0	1	2
36	0.4071	**1**	0	1	1	2	1	1	2
37	0.4106	1	1	0	2	2	2	2	2
38	0.4162	1	2	0	1	0	1	0	1
39	0.4276	1	2	0	1	0	1	0	1
40	0.4396	1	2	1	0	1	0	1	0
41	0.4538	1	2	1	0	1	0	1	0
42	0.4650	1	2	1	1	0	1	0	1
43	0.4700	1	2	1	2	0	1	0	1
44	0.4650	1	2	1	2	1	1	0	1
45	0.4575	1	0	1	2	1	2	1	1
46	0.4478	1	0	1	2	1	2	2	2
47	0.4337	1	0	1	0	1	2	1	2
48	0.4269	1	0	1	0	1	2	1	1

As summarized in Table 5, similar changes were introduced to bring forth disturbance classes XD2–XD5. In general, the five disturbance classes have very similar effects on the process and are difficult to distinguish based on the online measurements (Srinivasan and Qian, 2005). They therefore serve as good illustration of the DLA approach.

Like any process history-based method, the accuracy of the dynamic locus analysis method depends on the representativeness of the reference signal. A good reference signal should adequately capture the dynamics of the key variables including not only their magnitudes and trends, but also the various sub-states, their time of occurrence, and sequence. The latter aspect is discussed further in Section 4.3. In this case study, XD1-B, XD2-B, XD3-B, XD4-B and XD5-B were used as the references for the five disturbances. In these, the samples which correspond to signals from normal operation were omitted, resulting in a length of 1220 samples. A separate reference of normal operation, notated as XD0-B was also added to the

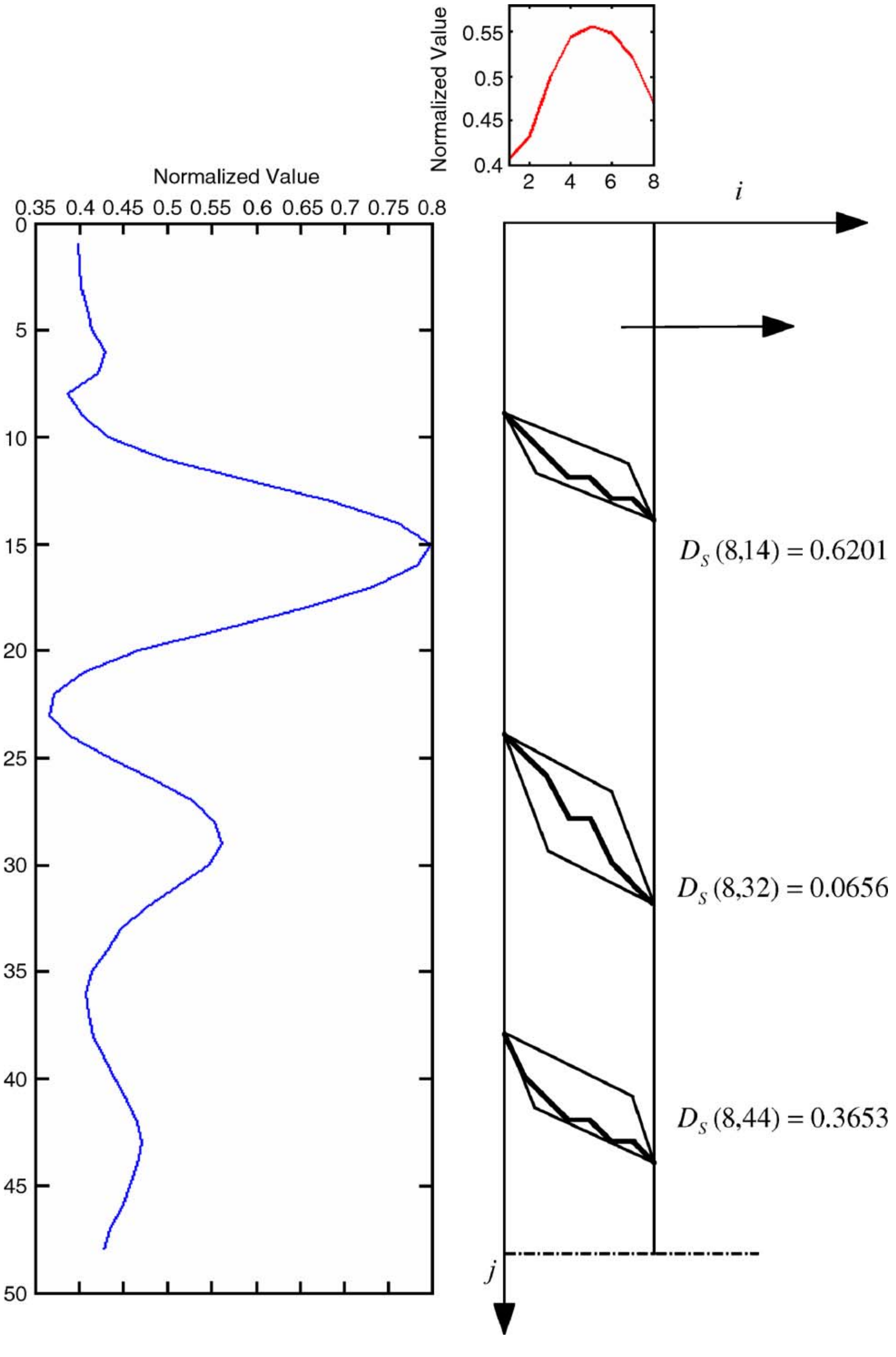

Fig. 4. Search path for illustrative example.

reference database. Data from other runs were then used for testing.

The dynamic locus analysis method is used for online disturbance identification as follows. Analysis starts at $t = 1$ with the process in an unknown state. With an evaluation window of $m = 8$ samples, the locus can be identified in each reference signal starting at $t = 8$ min when the first real-time signal segment is available. The normalized signal differences, corresponding points, and inseparability can then be calculated. At every subsequent time point, the evaluation window is moved forward by one sample (rolling evaluation window) and the analysis repeated. In one run, a fault was introduced into the process at $t = 51$ min. Fig. 7 shows the results from dynamic locus analysis of real-time signals from time $\tau_x = 8$ samples to 108 samples when compared with reference signal XD0-B. Initially the process is in a normal steady state as can be seen

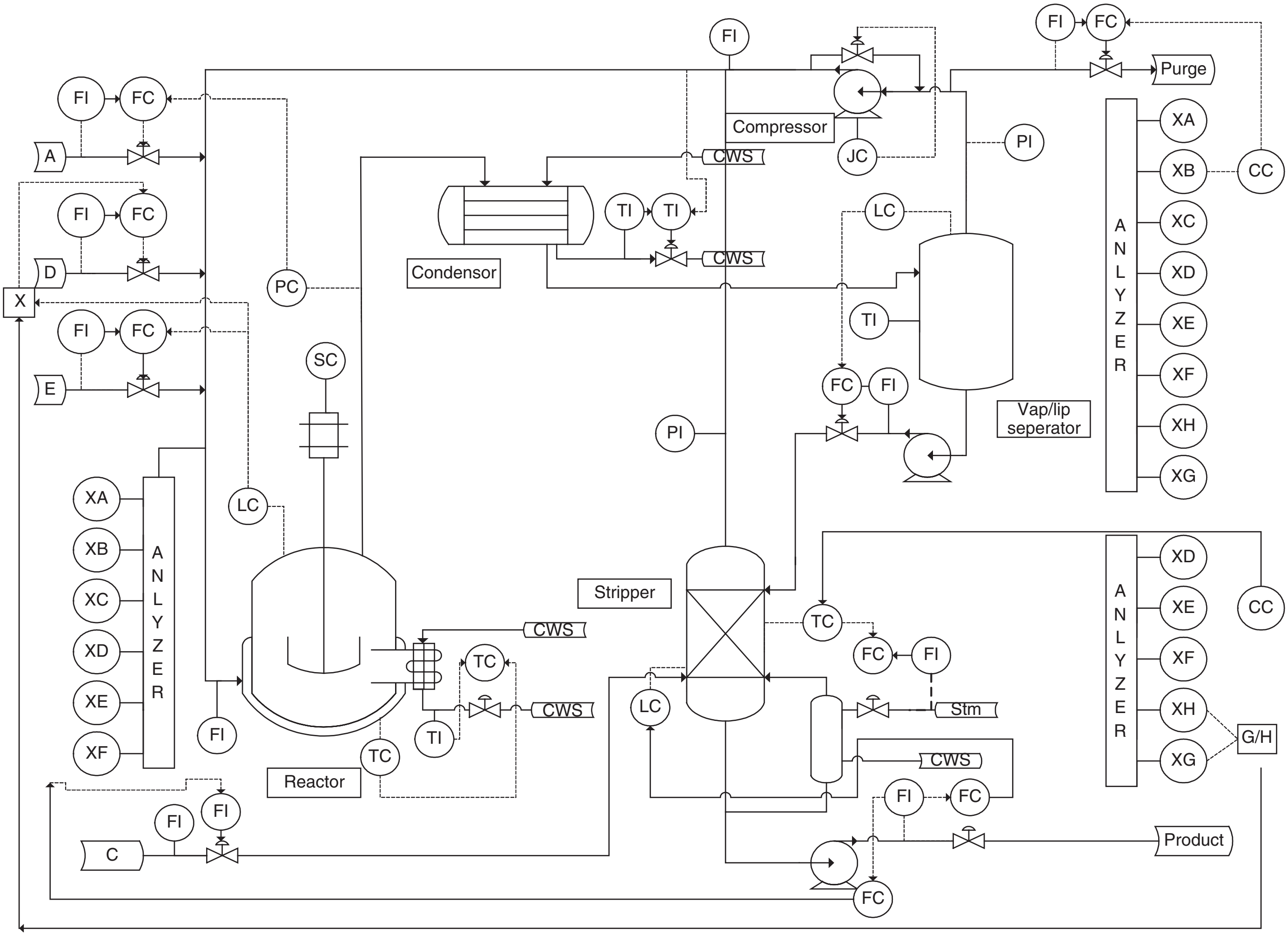

Fig. 5. Flowsheet of Tennessee Eastman process.

Table 4
TE process measurements and their base value

Variable name	Variable number	Base case value	Units
A feed (stream 1)	XMEAS (1)	0.25052	kscm h
D feed (stream 2)	XMEAS (2)	3664.0	kgh^{-1}
E feed (stream 3)	XMEAS (3)	4509.3	kg h^{-1}
A and C feed (stream 4)	XMEAS (4)	9.3477	kscm h
Recycle flow (stream 8)	XMEAS (5)	26.902	kscm h
Reactor feed rate (stream 6)	XMEAS (6)	42.339	kscm h
Reactor pressure	XMEAS (7)	2705.0	kPa gauge
Reactor level	XMEAS (8)	75.0	%
Reactor temperature	XMEAS (9)	120.40	°C
Purge rate (stream 9)	XMEAS (10)	0.33712	kscmh
Product separator temperature	XMEAS (11)	80.109	°C
Product separator level	XMEAS (12)	50.000	%
Product separator pressure	XMEAS (13)	2633.7	kPa gauge
Product separator underflow (stream 10)	XMEAS (14)	25.160	m^3 h^{-1}
Stripper level	XMEAS (15)	50.000	%
Stripper pressure	XMEAS (16)	3102.2	kPa gauge
Stripper underflow (Stream 11)	XMEAS (17)	22.949	m^3 h^{-1}
Stripper temperature	XMEAS (18)	65.731	°C
Stripper steam flow	XMEAS (19)	230.31	kg h^{-1}
Compressor work	XMEAS (20)	341.43	kW
Reactor cooling water outlet temperature	XMEAS (21)	94.599	°C
Condenser cooling water outlet temperature	XMEAS (22)	77.297	°C

Table 5
Disturbance profiles for TE process resulting from changes in base values of: (a) a feed during XD1; (b) reactor pressure during XD2; (c) reactor level during XD3; (d) reactor temperature during XD4; (e) compressor work during XD5

	Target	Time (min)	Target	Time (min)	Target	Time (min)	Target	Time (min)
(a)								
XD1-A	1.20*Base value	51	1.40*Base value	61	1.60*Base value	71	1.0*Base value	651
XD1-B	1.15*Base value	51	1.35*Base value	65	1.55*Base value	79	1.0*Base value	711
XD1-C	1.10*Base value	51	1.30*Base value	69	1.50*Base value	87	1.0*Base value	771
(b)								
XD2-A	1.03*Base value	51	1.05*Base value	61	1.07*Base value	71	1.0*Base value	891
XD2-B	1.025*Base value	51	1.045*Base value	65	1.065*Base value	79	1.0*Base value	891
XD2-C	1.02*Base value	51	1.04*Base value	69	1.06*Base value	87	1.0*Base value	951
(c)								
XD3-A	1.05*Base value	51	1.10*Base value	61	1.15*Base value	71	1.0*Base value	651
XD3-B	1.045*Base value	51	1.09*Base value	65	1.135*Base value	79	1.0*Base value	711
XD3-C	1.04*Base value	51	1.08*Base value	69	1.12*Base value	87	1.0*Base value	771
(d)								
XD4-A	1.05*Base value	51	1.10*Base value	61	1.15*Base value	71	1.0*Base value	651
XD4-B	1.045*Base value	51	1.09*Base value	65	1.135*Base value	79	1.0*Base value	711
XD4-C	1.04*Base value	51	1.08*Base value	69	1.12*Base value	87	1.0*Base value	771
(e)								
XD4-A	0.95*Base value	51	0.90*Base value	61	0.85*Base value	71	1.0*Base value	651
XD4-B	0.955*Base value	51	0.91*Base value	65	0.865*Base value	79	1.0*Base value	711
XD4-C	0.96*Base value	51	0.92*Base value	69	0.88*Base value	87	1.0*Base value	771

from the small normalized difference η. Starting at $\tau_x = 54$ samples, the η with XD0-B increases as does the inseparability ratio α shown in Fig. 8 indicating a process abnormality. When the real-time signal is compared with the other references in the database, from the 55th sample η is the smallest with XD2 (see Fig. 9). At $\tau_x = 56$ samples, α falls below $\alpha_{\min}$ identifying that the process is undergoing disturbance XD2 (see Fig. 8). The location of minimum η in these plots indicate the progression of the disturbance—thus $\tau_x = 56$ samples correspond to $\tau_y =$ sixth sample in the XD2-B. The progression of τ_y with time is coherent only with reference XD2-B as shown in Fig. 10; the erratic progression with all other references also bears confirmation that the disturbance in this run is XD2.

To further test the accuracy of the approach, starting at $\tau_x = 250$ samples, we introduced a major variation in both duration and magnitude during this run as compared to XD2-B. This results in the increase in α between $\tau_x = 258$ to 400 samples. However, even during this period, α is substantially below the $\alpha_{\min}$ of 0.75. Thus while run-to-run variations will also lead to an increase in η with XD2-B, this effect is small when compared to η with other types of disturbances. Fig. 11 reveals the clear separability of the disturbance throughout the run. A summary of the result for this run is reported as Run-3 in Table 6. A similar study was performed for nine other runs. Details of these are presented in Qian (2005) and only a summary is presented here. In all cases, the proposed method correctly identified the disturbance with an average delay of five samples. The computation time for the complete analysis at each time sample is 2.26 s (on a Pentium IV, 2.4 GHz cpu).

4.1.1. *Effect of parameters on dynamic locus analysis results*

The performance of dynamic locus analysis is affected by two main parameters—the evaluation window m and the minimum inseparability threshold $\alpha_{\min}$. In this section, we test the parametric sensitivity of dynamic locus analysis.

4.1.1.1. Evaluation window To test the effect of m, we repeated the above ten test runs with various m values as shown in Table 7. In all cases the proposed method identified the correct disturbance. The results reveal the following: increasing the evaluation window leads to an increase in the detection delay. This is because the normalized difference is calculated over the entire window. For large m, the inclusion of samples from the initial normal periods decreases the sensitivity during calculation of η. The time cost of dynamic locus analysis also increases (almost linearly) with increase in m. The benefit of increasing m is however apparent from the improvement in coherence metric for larger values as shown in Fig. 12. Smaller evaluation windows can lead to suboptimal matches since only a short segment of the real-time signal has to match the reference signal. In the face of noise, the time progression of τ_y versus τ_x will have fluctuations. At large m values, since a longer signal segment has to match with the reference, noise induced variations in τ_y is minimal and the coherence metric is low. We have found that, in all cases, $m = 8$ samples provides an acceptable detection delay, low computational cost, as well as adequate coherence.

4.1.1.2. Minimum separability threshold $\alpha_{\min}$ affects the detection delay as shown in Fig. 13. Increasing $\alpha_{\min}$ decreases the detection delay but at the cost of accepting a smaller

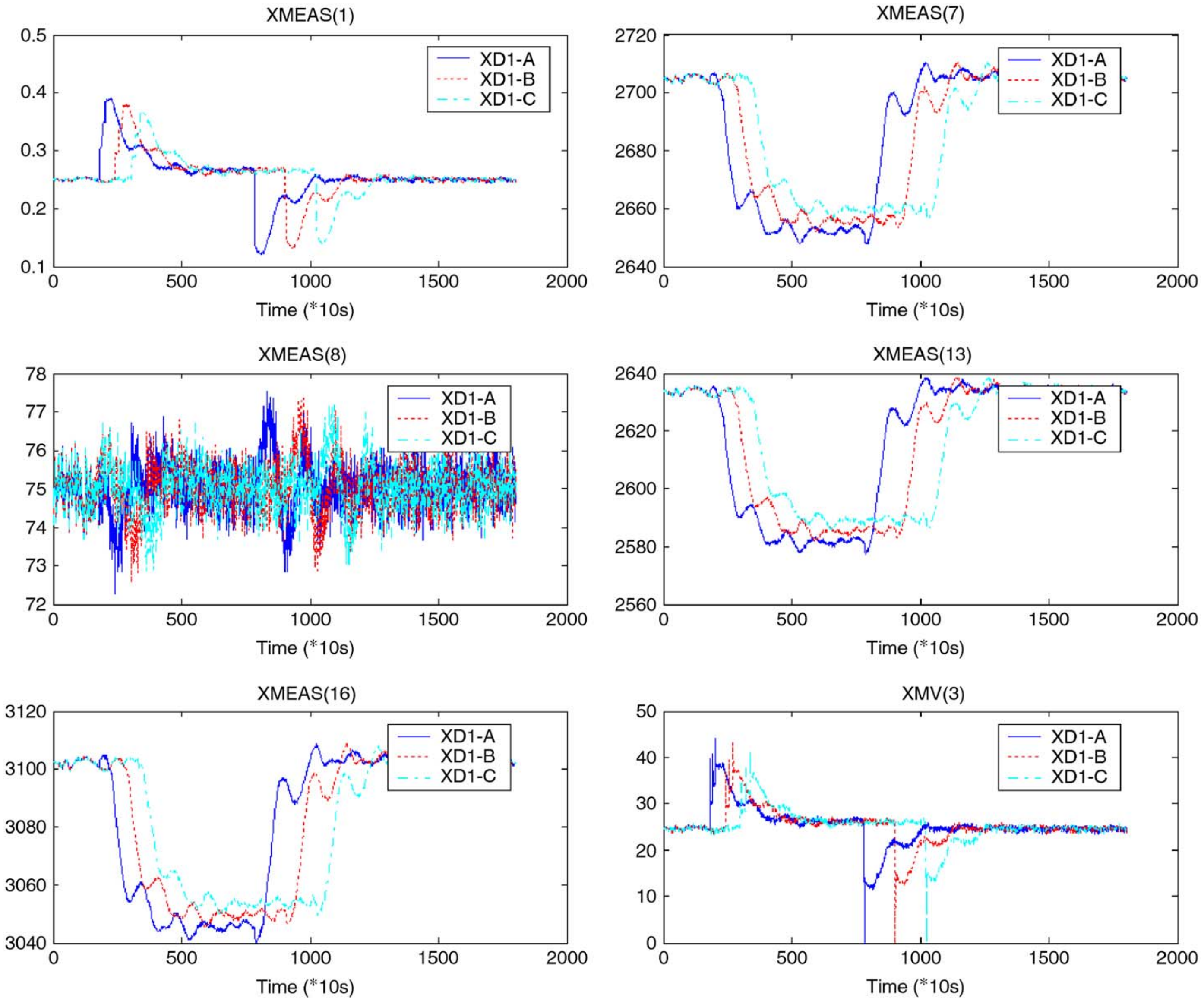

Fig. 6. Three runs of XD1 with different magnitudes and duration.

difference between the best matching and the next-best matching reference signals as the basis for the result. We have found $\alpha_{\min} = 0.75$ as a suitable threshold for all case studies.

The effect of increasing noise in the samples was also studied. Table 8 shows the change in the identification delay as noise is varied from 0% to 10%. As can be seen there, on average the identification delay increases from 5 samples to only 7.7 samples despite the large increase in measurement noise. As can be seen from the above studies, the dynamic locus analysis approach is accurate, computationally efficient and robust to a wide range of parameter settings.

4.2. Case study 2: fault diagnosis during startup of a lab-scale distillation column

In this section, the proposed methodology is illustrated on a lab-scale distillation unit. The schematic of the unit is shown in Fig. 14. The distillation column is of 2 m height and 20 cm inner-diameter and has 10 trays. The feed enters at tray 4. The system is well integrated with a control console and data acquisition system. Nineteen variables comprising of all tray temperatures, reboiler and condenser temperature, reflux ratio, top and bottom column temperatures, feed pump power, reboiler heat duty, and cooling water inlet and outlet temperatures are measured at 10-s intervals. Cold startup of the distillation column with ethanol–water 30% v/v mixture is performed following the standard operating procedure shown in Table 9. The feed passes through a heat exchanger before being fed to the column. The startup normally takes 2 h and different faults such as sensor fault, failure to open pump, too high a reflux ratio etc., can be introduced at different states of operation. For fault diagnosis using dynamic locus analysis, the reference database is first populated using data from 11 runs of the process—one normal startup and the 10 faults summarized in Table 10.

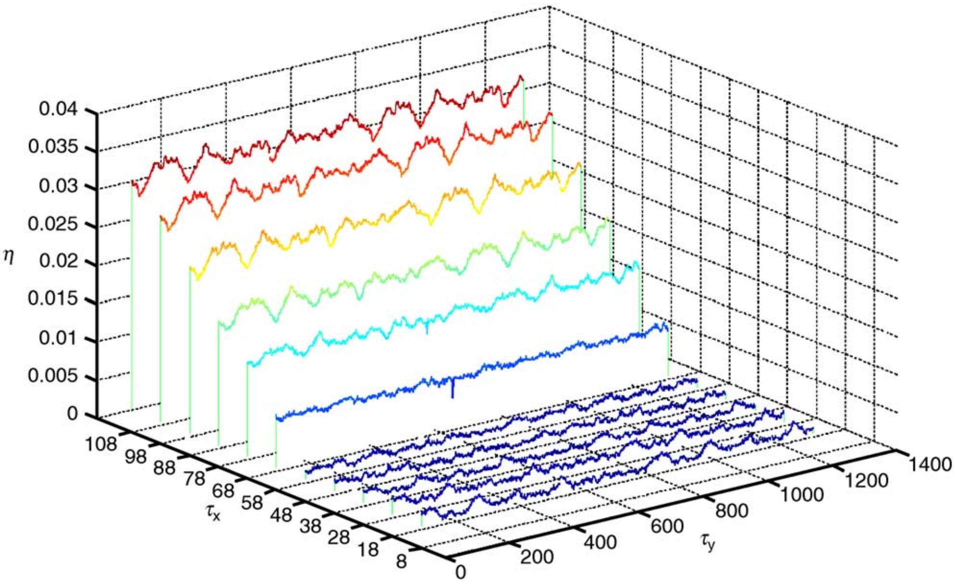

Fig. 7. Online comparisons of real-time signal and reference signal XD0-B at 10 snapshots from $\tau_x = 8$ to 108 samples.

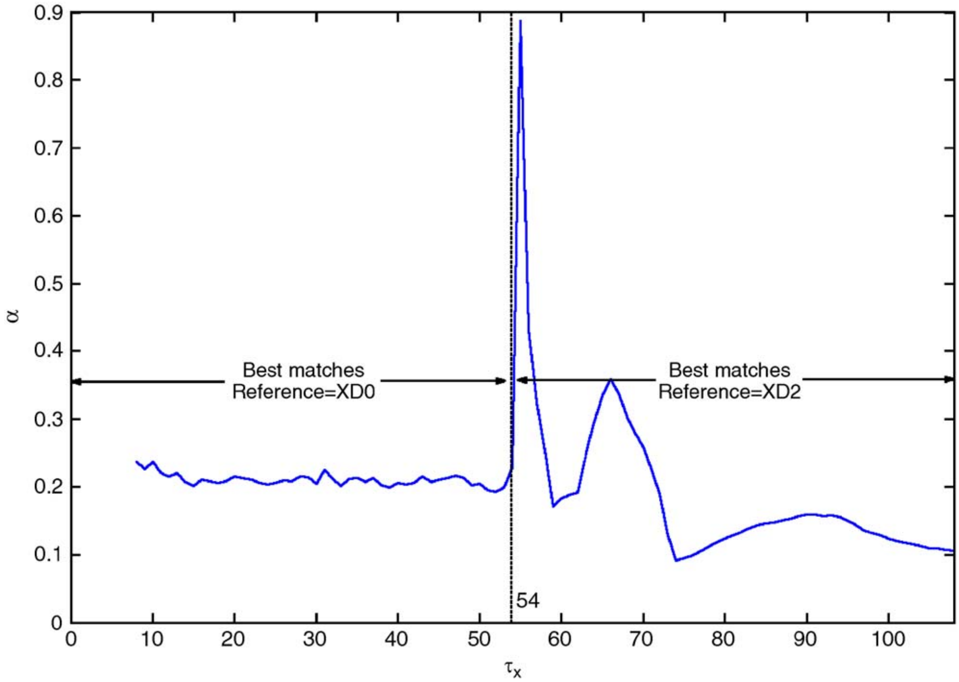

Fig. 8. Inseparability ratio during the first 110 min of Run-3 of TE process.

For example, a higher than normal pumping rate was induced in the run for DST03. This causes instability in the column and there is a drastic drop in the column temperatures as can be seen in Fig. 15.

The online dynamic locus analysis algorithm was used for fault diagnosis and decision support during subsequent startups of the column. Consider one run (Run-3) when a fault was introduced at $t = 3590$ s. Results from this run are shown in Fig. 16 which shows the normalized difference between the real-time signal and the 11 reference signals in the database throughout the startup. As can be seen there, the real-time signal is close to normal till about 3700 s. The difference between real-time signal and all other references is much higher. Starting around $t = 3700$ s, the difference between real-time signal and the normal reference increases indicating that there is a fault during the startup operation. The difference between real-time signal and DST03 decreases and falls below that of the normal reference. At $t = 3710$ s, α falls below $\alpha_{\min}$ and identifies the

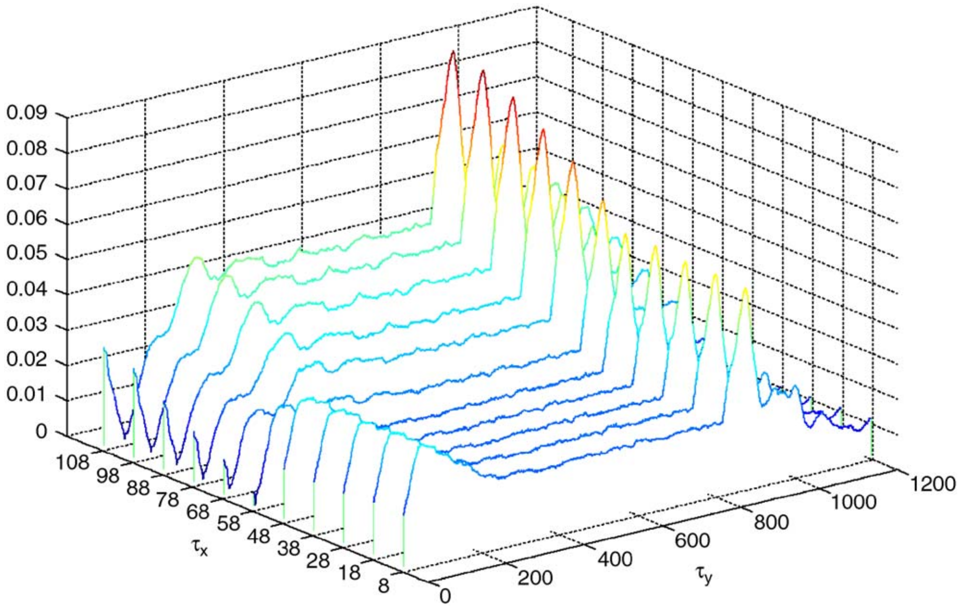

Fig. 9. Online comparisons of real-time signal and reference signal XD2 at 10 sample snapshots from $\tau_x = 8$ to 108 samples.

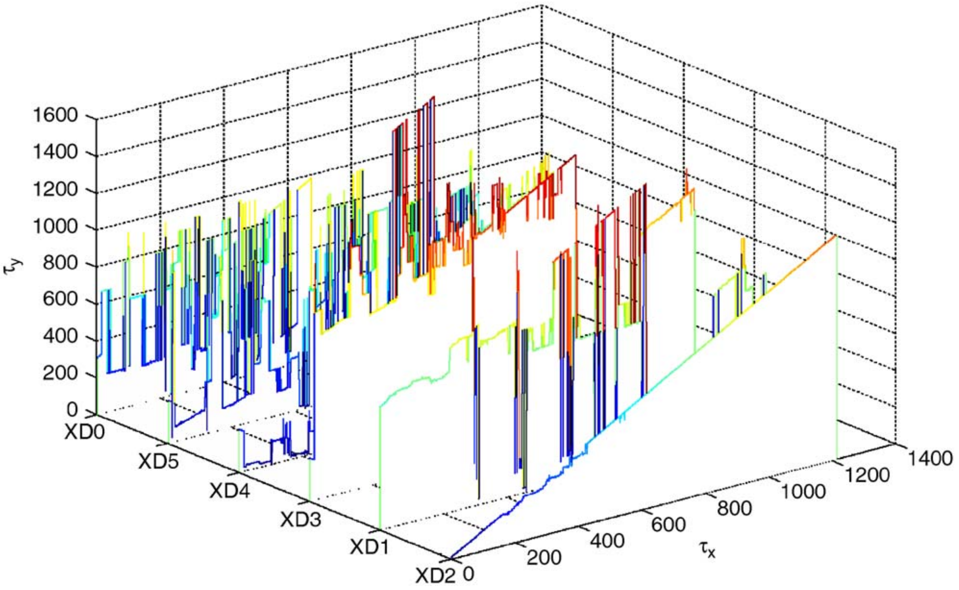

Fig. 10. Time progression of corresponding points when real-time signal is compared with all the reference signals throughout the run.

fault as of class DST03. The accuracy of the identification is evident from Fig. 17 which tracks τ_x over time and shows that the evolution of the current state continuously matches DST03. The slight fluctuations around t=5000 s are due to measurement noise and run-to-run differences.

Similar tests were done for all other cases. In the interest of space, only a summary of the findings is presented in Table 11. All faults could be accurately identified within an average of 43 s of their occurrence (about four samples). The maximum identification delay was about 120 s (12 samples). The average time cost at each sample was only 0.468 s. The evolution of the faults was also identified clearly with an average coherence metric of about 1.5 sample. A parametric sensitivity study was conducted for this case study as well and revealed the same trends as the previous case. In the interest of space, only the results from varying the evaluation window are reported here. As can be seen in Table 12, an increase in the evaluation window leads to a small increase in the identification delay and a *linear* increase in the computational cost.

4.3. Case study 3: process state identification during multimodal operation of FCCU

Dynamic locus analysis can be used for identifying the state of multi-mode process operations such as during startup, shutdown and other transitions in a fluidized catalytic cracking unit

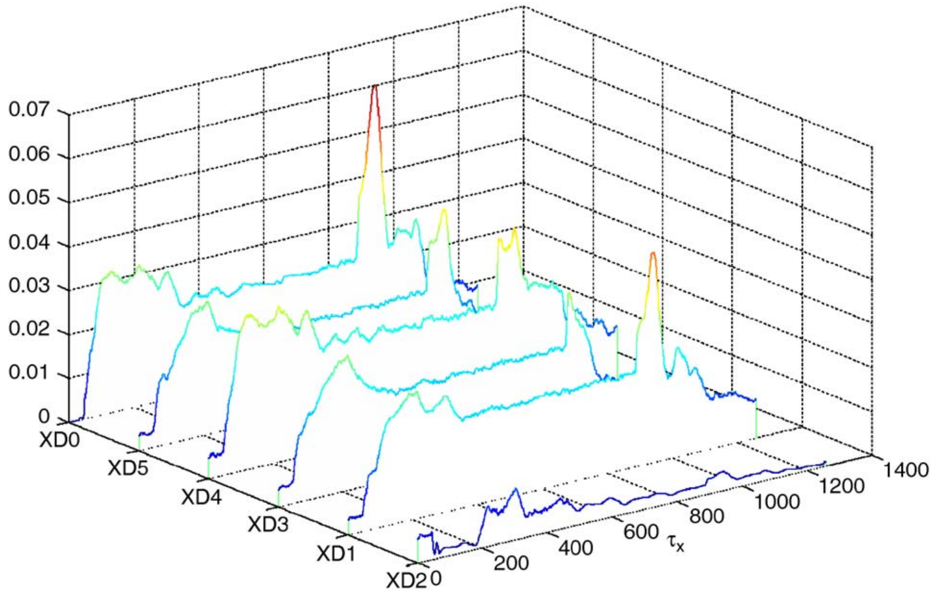

Fig. 11. Online comparisons of real-time signal with all reference signals at 10 sample snapshots from $\tau_x = 8$ to 1220 samples.

Table 6
Online process disturbance detection in TE process

	η at time of identification							Best-matching reference	τ_y (sample)	τ_x (sample)	Delay (sample)	Time cost (s)
	XD1	**XD2**	**XD3**	**XD4**	**XD5**	**XD0**	α					
Run-1	**0.0018**	0.0056	0.0051	0.0056	0.0039	0.0029	0.6151	XD1	6	56	5	2.24
Run-2	**0.0015**	0.0049	0.0043	0.0050	0.0033	0.0023	0.6686	XD1	6	56	5	2.25
Run-3	0.0061	**0.0023**	0.0075	0.0075	0.0054	0.0051	0.4457	XD2	6	56	5	2.26
Run-4	0.0045	**0.0020**	0.0057	0.0060	0.0039	0.0035	0.5729	XD2	6	56	5	2.26
Run-5	0.0051	0.0072	**0.0019**	0.0067	0.0053	0.0044	0.4261	XD3	6	56	5	2.25
Run-6	0.0044	0.0064	**0.0016**	0.0060	0.0046	0.0034	0.4737	XD3	6	56	5	2.25
Run-7	0.0055	0.0072	0.0067	**0.0019**	0.0054	0.0042	0.4483	XD4	6	56	5	2.25
Run-8	0.0050	0.0063	0.0058	**0.0017**	0.0046	0.0040	0.4205	XD4	6	56	5	2.28
Run-9	0.0038	0.0049	0.0050	0.0052	**0.0017**	0.0029	0.5914	XD5	6	56	5	2.30
Run-10	0.0034	0.0045	0.0041	0.0048	**0.0015**	0.0025	0.5933	XD5	6	56	5	2.25

Table 7
Effect of evaluation window on identification delay and time cost in TE case study

m	Identification delay (samples)						Time cost (s)					
	4	8	12	16	20	24	4	8	12	16	20	24
Run-1	3	5	8	11	14	17	1.19	2.24	3.19	4.16	5.13	6.13
Run-2	3	5	9	12	15	20	1.21	2.25	3.18	4.15	5.12	6.04
Run-3	3	5	7	10	13	15	1.21	2.26	3.25	4.20	5.15	6.14
Run-4	3	5	8	12	15	20	1.22	2.26	3.21	4.15	5.14	6.06
Run-5	3	5	7	10	12	15	1.21	2.25	3.25	4.21	5.20	6.14
Run-6	3	5	8	11	13	16	1.21	2.25	3.22	4.18	5.15	6.11
Run-7	3	5	7	9	12	14	1.21	2.25	3.26	4.25	5.19	6.16
Run-8	3	5	7	10	12	14	1.21	2.28	3.25	4.20	5.19	6.18
Run-9	3	5	8	10	13	16	1.21	2.30	3.22	4.20	5.16	6.11
Run-10	3	5	8	11	14	17	1.21	2.25	3.21	4.17	5.15	6.09
Average	3	5	8	11	13	16	1.21	2.26	3.23	4.19	5.16	6.12

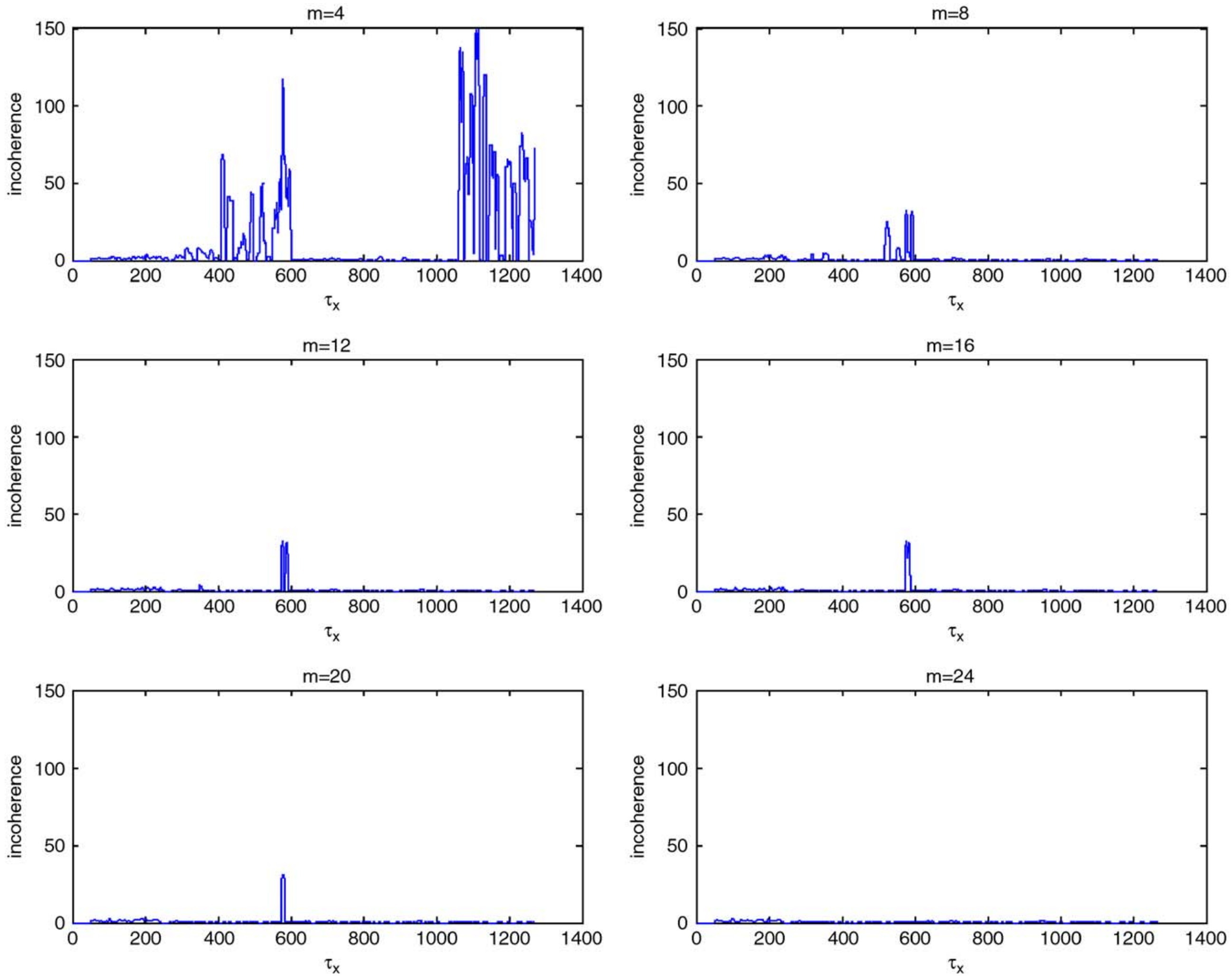

Fig. 12. Effect of evaluation window on coherence metric during Run-3 of TE process.

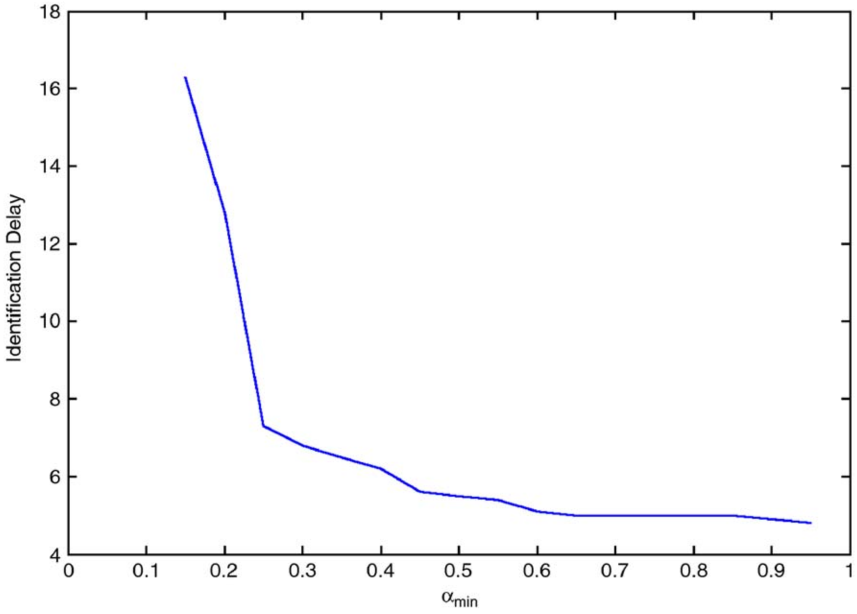

Fig. 13. Effect of minimum inseparability threshold $\alpha_{\min}$ on identification delay in TE process.

(FCCU). The FCCU converts a mixture of heavy oils into more valuable light products and is the dominant conversion process and the major contributor to value addition in the refining process. Successful operation of the FCCU is critical to the operating success of most refineries. The FCCU can be operated to maximize the yield of gasoline or middle distillate. This flexibility allows the refiner to tune the product slate to change in demands. This results in transitions during the FCCU operation. A high-fidelity dynamic simulator of a FCCU, called ShadowPlant, is used here (Honeywell, 2001). The ShadowPlant consists of five main sections as shown in Fig. 18:

1. feed preheater;
2. riser/Regenerator;

Table 8
Effect of measurement noise on identification delay in TE case study

Noise	Identification delay (samples)					
	0%	2%	4%	6%	8%	10%
Run-1	5	5	5	6	6	6
Run-2	5	5	6	6	6	6
Run-3	5	5	5	6	6	6
Run-4	5	6	6	6	7	12
Run-5	5	5	5	6	6	10
Run-6	5	5	6	6	6	8
Run-7	5	5	6	7	7	9
Run-8	5	5	6	6	7	8
Run-9	5	5	5	6	6	6
Run-10	5	5	5	6	6	6
Average	5	5.1	5.5	6.1	6.3	7.7

Table 9
Standard operating procedures (SOP) for startup

Distillation column startup SOP	
1.	Set all controllers to manual
2.	Fill reboiler with liquid bottom product
3.	Open reflux valve and operate the column on full reflux
4.	Establish cooling water flow to condenser
5.	Start the reboiler heating coil power
6.	Wait for all of the temperatures to stabilize
7.	Start feed pump
8.	Activate reflux control and set reflux ratio
9.	Open bottom valve to collect product
10.	Wait for all the temperatures to stabilize

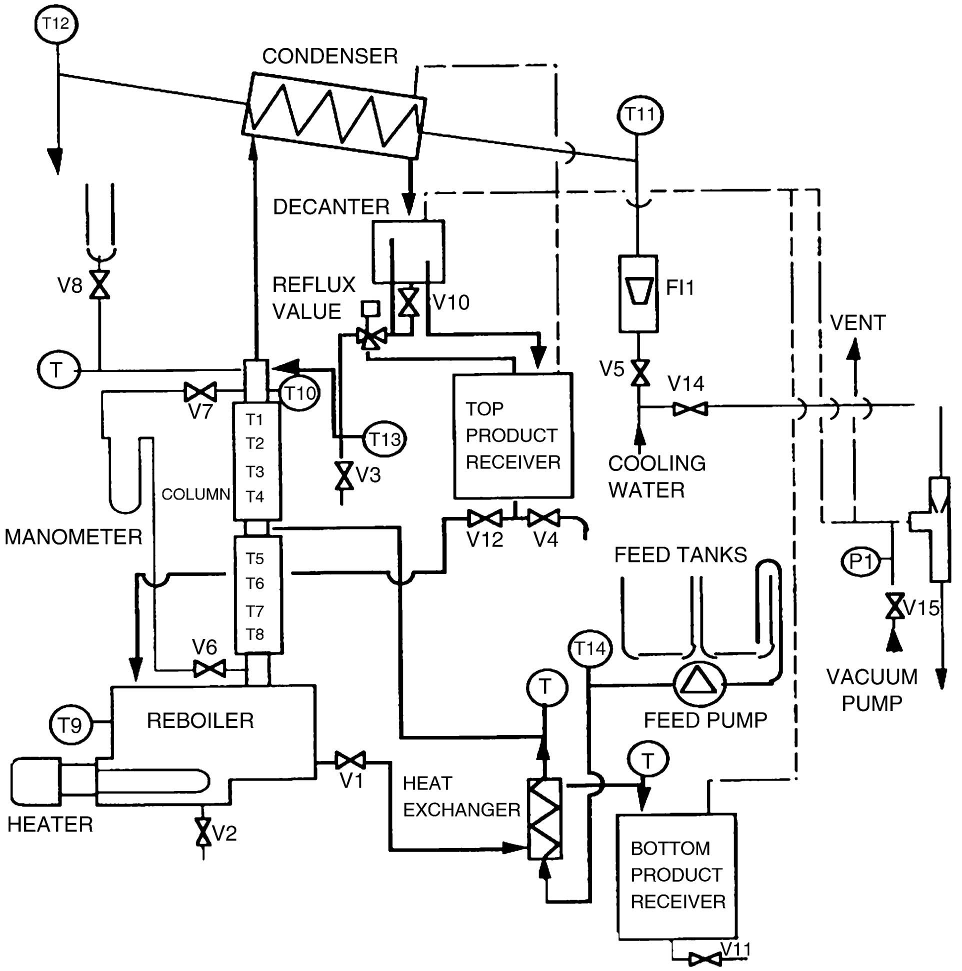

Fig. 14. Schematic of lab-scale distillation column.

3. main fractionator;
4. waste heat boiler;
5. air-preheater.

Table 10
Process disturbances for the distillation column operation

Case	Disturbance	Type
DST01	Reboiler power low	Step
DST02	Reboiler power high	Step
DST03	Feed pump high	Step
DST04	Feed pump low	Step
DST05	Tray temperature sensor T6 fault	Random variation
DST06	Reflux ratio high	Step
DST07	Reflux ratio low	Step
DST08	Bottom valve	Sticking
DST09	Cooling water	Slow drift
DST10	Low cooling water flow and feed pump malfunction	Step

Among the main operational states of the ShadowPlant, the startup is one of the most difficult transitions. One of the first steps in the startup of the ShadowPlant is the startup of the air-preheater section. The air blower is started slowly and operated at a steady state speed of about 4000 rpm. Air is heated in the preheater furnace to about 370 °C and is used to initially increase the temperature of reactor/regenerator. Once this operation is stabilized, torch oil flow is started in the regenerator. This supplies the necessary heat and the fuel gas supply to the air-preheater is stopped. Catalyst is then added to the regenerator gradually and the temperature maintained around 400 °C by manipulating the torch oil flow rate. When the catalyst level in the riser reaches 60%, the regenerator slide valve is opened and the catalyst moves through the riser to the reactor. Once the regenerator/reactor section has reached an intermediate steady state, with the regenerator temperature around 600 °C, the fractionator is flushed with kerosene and started up with fresh feed.

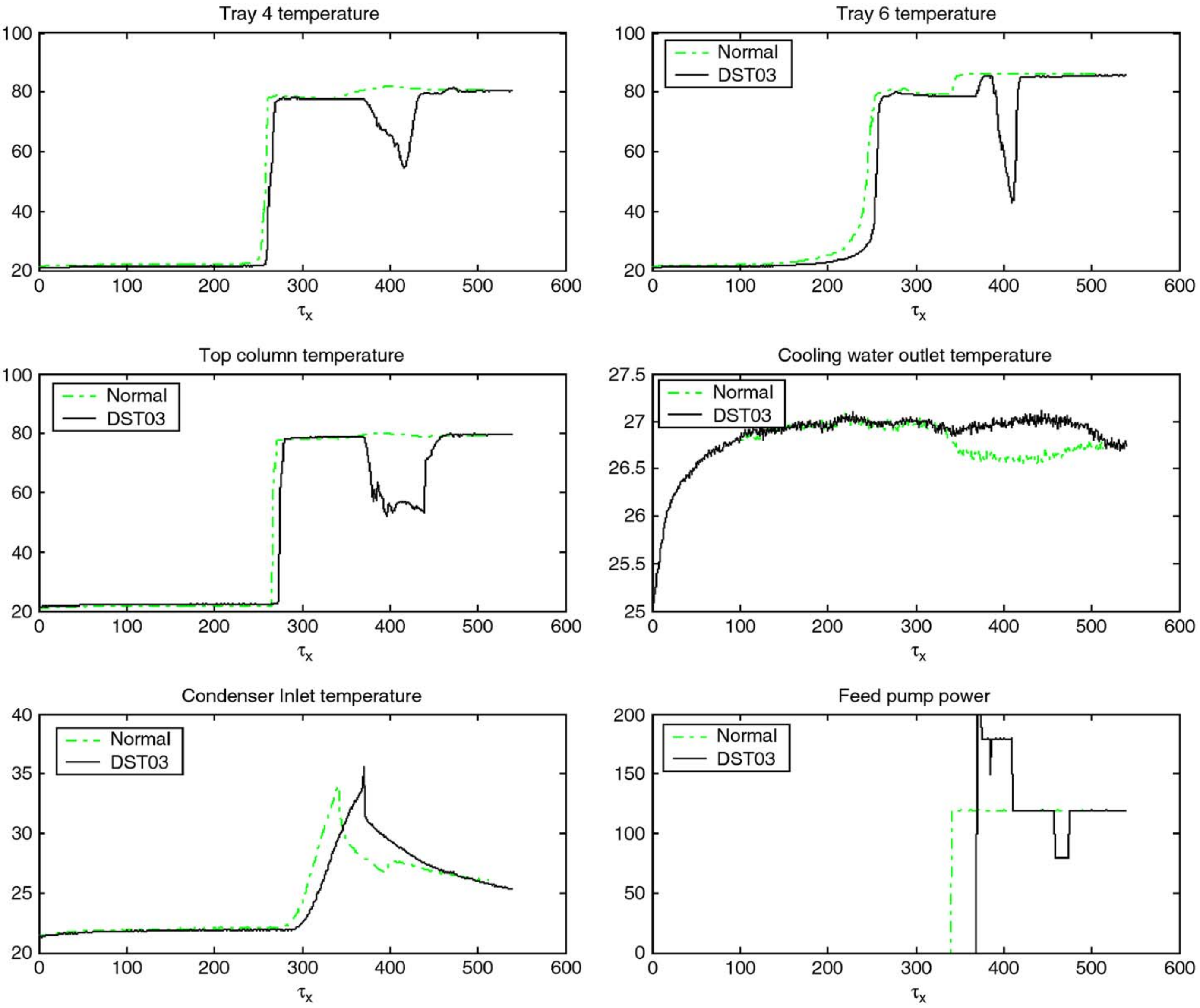

Fig. 15. Process signals for Run-03 of lab-scale distillation column.

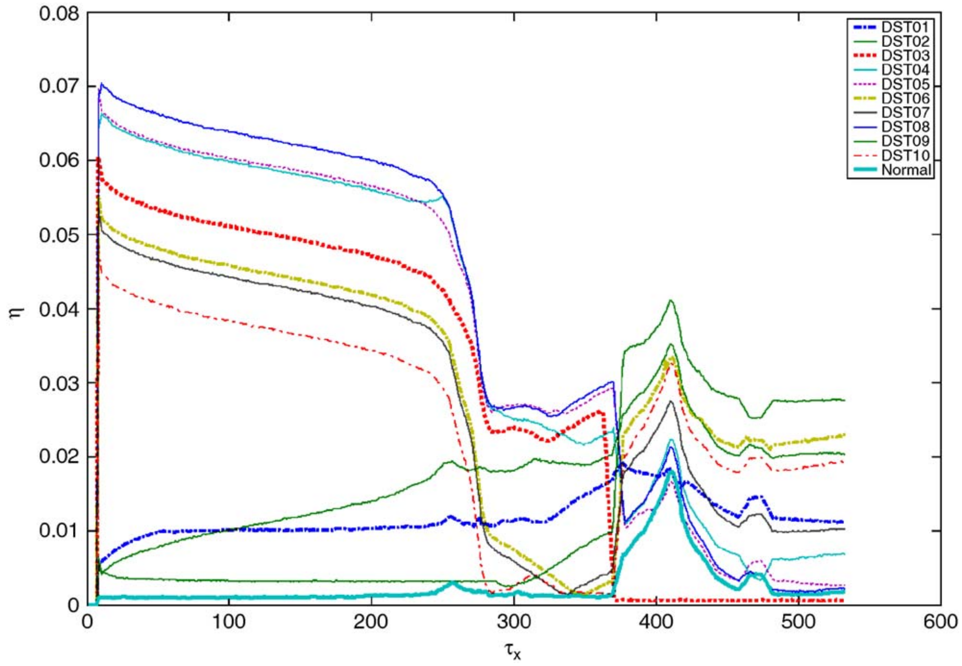

Fig. 16. Normalized difference with all reference signals during Run-3 of lab-scale distillation column startup.

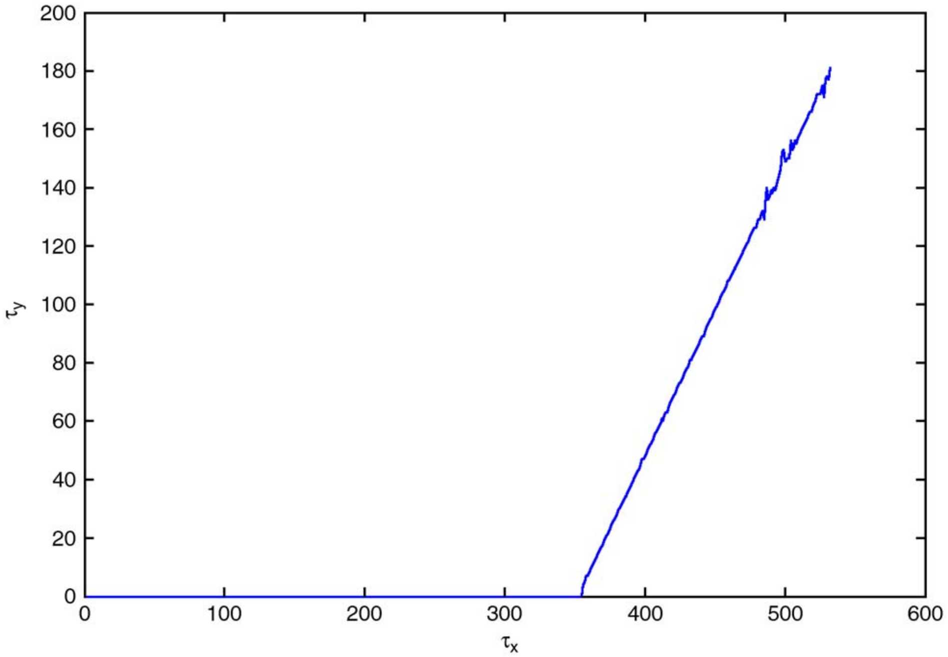

Fig. 17. Time evolution of progression of fault between $t = 0$ to 550 samples during Run-3 of lab-scale distillation column startup.

When the fractionator reaches a steady state, it is connected to the reactor/regenerator. Feed is started through the reactor and cracked products are fed to the main fractionator. Throughput is gradually increased and steady state flow established. Details of the unit and the startup transition are reported by Srinivasan et al. (2004, 2005a).

Several runs of the startup were performed following the standard operating procedure and data for all 335 measured variables collected at 10 s intervals. Random noise was also added to the measured variables to simulate measurement noise in the real process. While the procedure for starting up the FCCU was the same in all the runs, minor differences in the magnitudes and duration were introduced between the runs.

Three runs—G_5, G_6 and G_7—with average startup duration of about 23 h are considered in detail here. Data from G_7 was used as the reference signals while data from G_5 and G_6 are

Table 11
Faults diagnosis results for lab-scale distillation column

Case	Time fault introduced (s)	Detection time (sample)	Detection delay (sample)	Identification time (sample)	Identified fault	Identification delay (sample)	Coherence metric (sample)	Time cost (s)
Run-01	10	6	5	6	DST01	5	1.612	0.473
Run-02	10	6	5	6	DST02	5	1.544	0.468
Run-03	3590	370	11	371	DST03	12	0.855	0.470
Run-04	3560	356	0	357	DST04	1	0.995	0.464
Run-05	4250	427	2	429	DST05	4	0.279	0.467
Run-06	3500	352	2	352	DST06	2	2.391	0.470
Run-07	3450	346	1	346	DST07	1	3.609	0.470
Run-08	4700	472	2	472	DST08	2	1.612	0.465
Run-09	10	6	5	6	DST09	5	1.618	0.464
Run-10	3000	301	1	306	DST10	6	0.988	0.467
Average	–	–	3.4	–		4.3	1.550	0.468

Table 12
Effect of evaluation window on identification delay and time cost in distillation column case study

m	Identification delay (samples)						Time cost (s)					
	6	8	10	12	14	16	6	8	10	12	14	16
Run-1	5	7	9	11	13	15	0.473	0.637	0.803	0.968	1.132	1.295
Run-2	5	7	9	11	13	15	0.468	0.631	0.796	0.957	1.117	1.273
Run-3	12	13	13	14	14	14	0.470	0.636	0.802	0.966	1.129	1.291
Run-4	1	1	2	3	3	4	0.464	0.635	0.801	0.965	1.127	1.288
Run-5	4	4	5	5	6	5	0.467	0.636	0.802	0.967	1.131	1.293
Run-6	2	3	3	3	3	4	0.470	0.635	0.802	0.965	1.128	1.289
Run-7	1	1	1	1	1	1	0.470	0.634	0.798	0.962	1.124	1.283
Run-8	2	2	2	2	2	2	0.465	0.636	0.800	0.963	1.126	1.289
Run-9	5	7	9	11	13	15	0.464	0.636	0.802	0.967	1.131	1.292
Run-10	6	6	6	6	8	9	0.467	0.627	0.787	0.944	1.099	1.253
Average	4.3	5.1	5.9	6.7	7.6	8.4	0.468	0.634	0.799	0.962	1.124	1.285

used for testing the dynamic locus analysis method. Using prior knowledge, the nine main substates listed in Table 13 and shown in Fig. 19 were identified as important from an operations perspective. Their times of occurrence in the reference signals were also stored in the database so that the corresponding points can be identified during real-time runs. It should be noted that the two test runs differ in the start time, duration and variable magnitudes during each substate. Dynamic locus analysis was used to identify the progression of the transition in real-time. Results for online substate identification are given in Table 13. In all cases, the substates were correctly identified within an average delay of 1.7 samples (or 17 s) since the inception of the substate. This compares favorably with other approaches such as the trend-based state identification reported by Srinivasan et al. (2005a), where the average delay was 20 samples (about 3 min). This decrease in detection delay can be attributed to the use of the original signal (rather than its abstracted qualitative equivalent); hence there is no loss of information and process operations like opening of valves and setpoint changes that are direct indicators of specific substates can be easily and accurately identified even in the presence of noise and run-to-run variations. Despite the large-scale of the case study the average time cost for each sample was only 6.6 s. An evaluation of the effect of number of variables was also performed. A change in the number of variables in each sample from 16 to 335 (a factor of 21) leads to an increase in the computational time by a factor of only 1.4. This is because the computational complexity of DLA is independent of the number of variables and the small increase in the computational load with larger sample sizes is solely due to the resulting increase in the vector lengths.

5. Discussion

The online identification of the process state is an important and challenging problem which occurs in process supervision, especially during transition operations. The use of small windows of data to identify temporal process states has been previously demonstrated using other data-driven approaches—statistical methods (Srinivasan et al., 2004) and neural networks (Srinivasan et al., 2005b, c). In this paper, we have developed a signal comparison-based approach to this problem. Existing signal comparison approaches have been

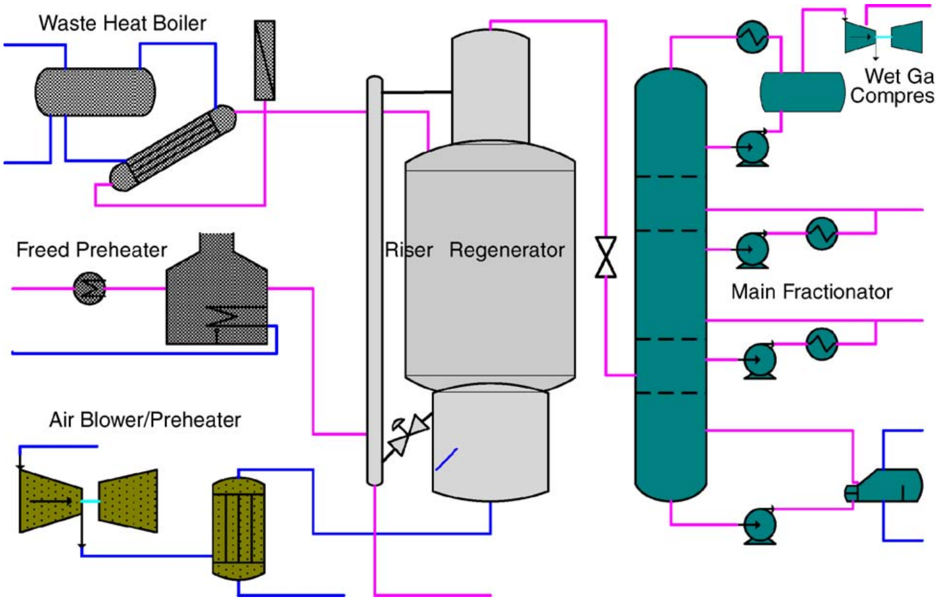

Fig. 18. Schematic of ShadowPlant FCCU.

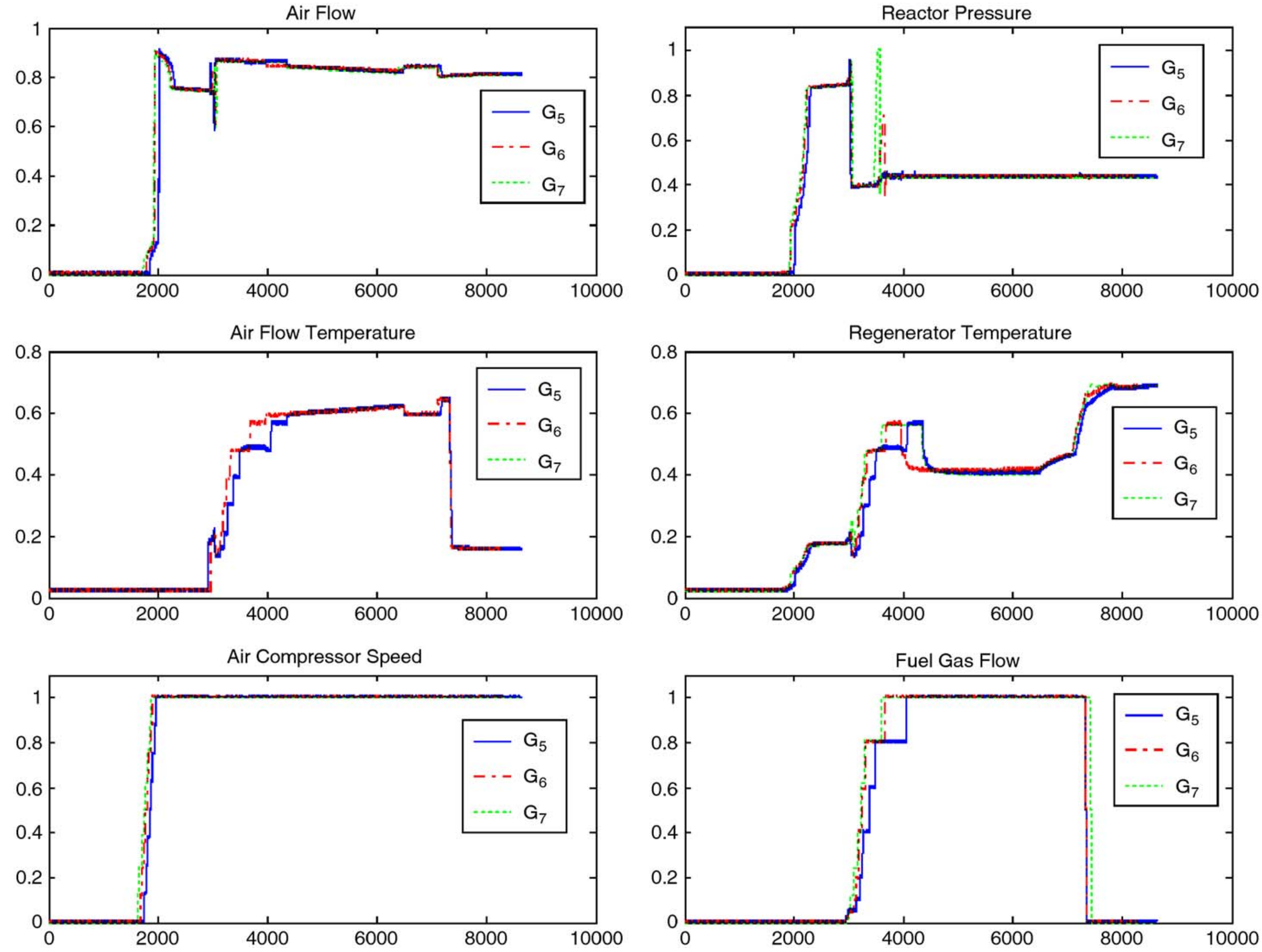

Fig. 19. Real-time and reference signals during startup of ShadowPlant.

Table 13
State identification results during startup transition in shadowPlant

Sub-state of startup	Reference signal		Test run G_5			Test run G_6		Time cost for each data sample (s)
	Occurrence time (×10 s)	Occurrence time (×10 s)	Detection time (×10 s)	Detection time (×10 s)	Occurrence time (×10 s)	Detection time (×10 s)	Detection delay (×10 s)	
Unit checking (S_S-1)	11	11	11	0	11	11	0	6.453
Waste heat boiler startup (S_S-2)	352	325	326	1	382	383	1	6.625
Preparation for air-blower startup (S_S-3)	1471	1642	1643	1	1568	1569	1	6.610
Startup of main air-blower (S_S-4)	1632	1753	1754	1	1687	1688	1	6.656
Operation adjustment (S_S-5)	1880	1967	1967	0	1893	1894	1	6.609
Flue gas lighting (S_S-6)	2983	2931	2932	1	2963	2964	1	6.687
Temperature ramp-up (S_S-7)	2998	2952	2953	1	2979	2980	1	6.688
Catalyst loading (S_S-8)	4346	4365	4367	2	3969	3972	3	6.625
Catalyst transfer (S_S-9)	7234	7258	7266	8	7227	7229	2	6.656
Average		–	–	1.667	–	1.222	–	6.623

designed largely for offline use where data from one complete run is compared with that from another—the essential requirement being that the start- and end-points of the signals to be compared should coincide. These approaches cannot be used for online applications since the point in the historical database that should be matched with the starting point of the online signal is unknown. The approach proposed in this paper uses dynamic programming to efficiently identify the segment in historical data that best matches a given online multivariate signal. During this search, all the potential matching endpoints in the reference signal are considered. Synchronization differences between the two signals are also accommodated by using DTW for signal comparison.

Results from three different case studies show that both the state type and progress can be identified accurately. This is particularly beneficial during transitions in large-scale agile processes where, because of simultaneous manual and automated control actions in different sections of the plant, operations personnel need information about the current state of each plant section. The corresponding point calculated by dynamic locus analysis can be directly used to provide this as illustrated in Section 4.3.

The accuracy of identification as well as the computation time increase with the evaluation window, therefore selection of a suitable window length is essential. This approach is based on the insight that a small evaluation window can sufficiently serve as a fingerprint for state identification since the occurrence of similar temporal, multivariate patterns at different process states is rare in practice. This precept has been confirmed using three different case studies. The small evaluation window leads to one of the key advantages of this method—a computational performance that is suitable for online usage, even in large-scale case studies. Computational time and memory requirements are only weak functions of the number of variables to be monitored as shown in Section 4.3. The current signal comparison approach in DLA does not seek to analyze novel states other than calculating their normalized distance. Future work will seek to further extend the analysis of novel states including identifying of locus from linear and nonlinear combinations of reference signals.

Notation

A	sequence $A=\{a_1,a_2,a_3,\ldots,a_i\}$
B	sequence $B=\{b_1,b_2,b_3,\ldots,b_j\}$
$D(n,m)$	normalized DTW distance between signals X and Y
$D^*(nm)$	minimum DTW distance between signals X and Y
$D_A(i,j)$	minimum accumulated distance from point $(1,1)$ to point (i,j)
D_S	dissimilarity matrix between signals X and Y
K	collection of reference signals
k	position in reference signal which has the minimum $D(x_m,y_k)$
m	Length of current segment = length of evaluation window
n	length of reference signal
P	parent matrix
Q	number of variables in process
X	real-time signal segment $X=\{x_1,x_2,x_3,\ldots,x_m\}$
x_H	high limit of sensor range
x_L	low limit of sensor range
x_i'	normalized value of x_i
Y	reference signal $Y=\{y_1,y_2,y_3,\ldots,y_n\}$
y_{j^*}	point in reference signal that corresponds to x_m
Z	segment of signal Y, $Z=\{y_l,y_{l+1},\ldots,y_j\}$

Subscripts

c	index of variable
i	time index of signal Y, current segment
j	time index of signal X, reference signal

Greek letters

α	inseparability ratio = ratio of normalized difference between best matching and second-best matching reference signals, $0 \leqslant \alpha \leqslant 1$
$\alpha_{\min}$	minimum inseparability threshold
$\Delta(x_i, y_j)$	difference between x_i and y_j
η	normalized difference
τ_x	time of the real-time signal that corresponds to x_m
τ_y	time of the reference signal that corresponds to y_{j^*}

References

Chen, J.H., Liao, C.M., 2002. Dynamic process fault monitoring based on neural network and PCA. Journal of Process Control 12 (2), 277–289.

Chiang, L.H., Braatz, R.D., 2003. Process monitoring using causal map and multivariate statistics: fault detection and identification. Chemometrics and Intelligent Laboratory Systems 65, 159–178.

Colomer, J., Melendez, J., Gamero, F.I., 2002. Pattern recognition based on episodes and DTW. Application to diagnosis of a level control system, Sixteenth International Workshop on Qualitative Reasoning, Catalunya, Spain.

Honeywell, 2001. Guide to the ShadowPlant FCCU Standard Model.

Itakura, F., 1975. Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustics Speech and Signal Processing, ASSP 23 (1), 67–72.

Jain, A.K., Duin, R.P.W., Mao, J.C., 2000. Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1), 4–37.

Kano, M., Hasebe, S., Hashimoto, I., Ohno, H., 2001. A new multivariate statistical process monitoring method using principal component analysis. Computers and Chemical Engineering 25 (7–8), 1103–1113.

Kassidas, A., Taylor, P.A., MacGregor, J.F., 1998. Off-line diagnosis of deterministic faults in continuous dynamic multivariable processes using speech recognition methods. Journal of Process Control 8 (5–6), 381–393.

Krzanowski, W.J., 1979. Between-group comparison of principal components. Journal of American Statistical Association 74 (367), 703–707.

McAvoy, T.J., Ye, N., 1994. Base control for the Tennessee Eastman problem. Computers and Chemical Engineering 18 (5), 383–413.

Myers, C., Rabiner, L.R., Rosenberg, A.E., 1980. Performance tradeoffs in dynamic time warping algorithm for isolated word recognition. IEEE Transactions on Acoustics Speech and Signal Processing. ASSP 28 (6), 623–635.

Needleman, S.B., Wunsch, C.D., 1970. Journal of Molecular Biology 48, 443–453.

Nimmo, I., 1993. Start up plants safely. Chemical Engineering Progress 89 (12), 66–69.

Nomikos, P., Macgregor, J.F., 1994. Monitoring batch processes using multiway principal component Analysis. A.I.Ch.E. Journal 40 (8), 1361–1375.

Qian, M.S., 2005. Efficient methodologies for real-time state identification during process transitions, Ph.D. Thesis, National University of Singapore, Singapore.

Raich, A., Cinar, A., 1997. Diagnosis of process disturbances by statistical distance and angle measures. Computers and Chemical Engineering 21 (6), 661–673.

Rengaswamy, R., Venkatasubramanian, V., 1995. A syntactic pattern recognition approach for process monitoring and fault diagnosis. Engineering Applications for Artificial Intelligence 8 (1), 35–51.

Singhal, A., 2003. Tennessee Eastman simulation model: ⟨http://www.chemengr.ucsb.edu/~ceweb/computing/TE/tesimulation.htm⟩.

Singhal, A., Seborg, D.E., 2002. Pattern matching in historical batch data using PCA. IEEE Control Systems Magazine 22 (5), 53–63.

Srinivasan, R., Qian, M.S., 2005. Offline temporal signal comparison using singular points augmented time warping. Industrial and Engineering Chemistry Research 44 (13), 4697–4716.

Srinivasan, R., Wang, C., Ho, W.K., Lim, K.W., 2004. Dynamic PCA based methodology for clustering process states in agile chemical plants. Industrial and Engineering Chemistry Research 43 (9), 2123–2139.

Srinivasan, R., Viswanathan, P., Vedam, H., Nochur, A., 2005a. A framework for managing transitions in chemical plant. Computers and Chemical Engineering 29 (2), 305–322.

Srinivasan, R., Wang, C., Ho, W.K., Lim, K.W., 2005b. Context-based recognition of process states using neural networks. Chemical Engineering Science 60 (4), 935–949.

Srinivasan, R., Wang, C., Ho, W.K., Lim, K.W., 2005c. Neural Network Systems for Multi-dimensional Temporal Pattern Classification. Computers and Chemical Engineering 29 (5), 965–981.

Sundarraman, A., Srinivasan, R., 2003. Monitoring transitions in chemical plants using enhanced trend analysis. Computers and Chemical Engineering 27 (10), 1455–1472.

Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequence. Journal of Molecular Biology 147, 195–197.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003. A review of process fault detection and diagnosis. Part III: process history based methods. Computers and Chemical Engineering 27 (3), 327–346.

Waterman, M.S., Eggert, M., 1987. A New Algorithm for Best Subsequence Alignments with Application to tRNA-rRNA Comparisons. Journal of Molecular Biology 197, 723–728.

Williams, K., 1970. Dynamic Programming Sequential decision-making. Longman, London.