EFFICIENT METHODOLOGIES FOR REAL-TIME STATE IDENTIFICATION DURING PROCESS TRANSITIONS

QIAN MINGSHENG

NATIONAL UNIVERSITY OF SINGAPORE

2006

EFFICIENT METHODOLOGIES FOR REAL-TIME STATE IDENTIFICATION DURING PROCESS TRANSITIONS

QIAN MINGSHENG

(B. Eng., ECUST, China)

(M. Eng., ECUST, China)

A THESIS SUBMITTED

FOR THE DEGREE DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMICAL AND BIOMOLECULAR ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2006

Acknowledgements

I would like to express my deepest gratitude to my research supervisor, Dr. Rajagopalan Srinivasan for his excellent guidance and valuable ideas. His wealth of knowledge and accurate foresight have greatly impressed and enlightened me. I am indebted to him for his care and advice not only in my academic research but also in my daily life. Without him, my research would not be successful.

I am also grateful to Mr. Rangaswamy Premkumar for his understanding and giving me enough time to work on my research project when I worked in EASTMAN.

I would like to thank my lab mates in iACE lab ─ Kashyap, Anand,YewSeng and Manish for their helps to open my mind in my research.

In addition, I would like to give due acknowledgement to National University of Singapore, for granting me research scholarship and funds needed for the pursuit of my Ph.D degree. It has been a wonderful experience for me in NUS. I sincerely thank the University for this opportunity.

Finally, this thesis would not have been possible without the loving support of my family, my wife, sister, and my parents. I devote this thesis to them and hope that they will find joy in this humble achievement.

Contents

Summary

Continuous chemical plants have multiple steady state operating modes. Process monitoring, fault diagnosis and state identification during process transitions is an important task for plant operators and engineers. The ability to automatically identify process state would allow the control system to work properly. It is also important in order to ensure optimal operation, maintain quality of products and prevent accidents in processes operation. Online data from the process signal is a rich source of information and can be used for this purpose. Despite developments in process state identification from data, many important and challenging problems still persist in this area. In this thesis, new methodologies for computationally efficient process state identification have been developed.

Firstly, a new approach for temporal signal comparison has been developed. Information content is not homogenously distributed throughout a signal; rather the majority of the features of the signal are concentrated in a small number of points. In this thesis, such points, which are landmarks in the signal evolution, are termed as singular points. Process data is first segmented based on singular points. Dynamic programming and dynamic time warping (DTW) is used to find their optimal match and obtain the signal difference. Singular point augmentation can be used with traditional DTWs, the role of the latter in this case is for episode-wise comparison. In such cases, the proposed method improves the quality of signal comparison. A computationally efficient extrapolative time warping method which uses a greedy search instead of dynamic programming has also been developed in this thesis. A performance comparison of the singular point augmented time warping method with DTW reveals a substantial decrease in computational cost, which makes it amenable

for large-scale case studies. The extension of singular points based methodology for online signal comparison has also been developed and reported in this thesis.

Secondly, a new signal comparison-based approach, called dynamic locus analysis, for online state identification and fault diagnosis during process transitions has been proposed. Dynamic locus analysis is an extension of Smith and Waterman's (1981) 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.

The large-scale and complexity of modern chemical plants makes it difficult for the operator to constantly monitor all process variables. Numerous methods exist for monitoring processes; however most of them suffer from computational complexity problems when applied to large-scale processes. In this thesis, a new method called state-specific key variables selection has also been developed for large-scale processes. The state-specific key variables provide a basis for defining key variables dynamically. The state-specific key variables are selected based on properties of process signals and their features. State-specific key variables solve the problem of desynchronization across different sections and improve the sensitivity of state identification. From the operations standpoint, the monitoring load is also reduced.

All the methodologies proposed in this thesis have been tested using data from different kinds of agile operations - startup of a simulated Fluidized Catalytic Cracking

Unit, multi-mode operation in the Tennessee Eastman challenge process, a lab-scale fed-batch fermentation process, and a lab-scale distillation column. Their performance are compared with traditional methods and shown to be superior.

Nomenclature

c	Index of variable
i	Time index of signal R, current segment
j	Time index of signal T, reference signal
k, m	Index of singular points of T
l, n	Index of singular points of R
A	Sequence $A = \{a_1, a_2, a_3, ..., a_i\}$
B	Sequence $B = \{b_1, b_2, b_3, ..., b_j\}$
$D(t,r)$	Normalized DTW distance between signals T and R
$D^*(t,r)$	Minimum DTW distance between signals T and R
$D_A(i,j)$	Minimum accumulated distance from point $(1,1)$ to point (i,j).
D_S	Dissimilarity matrix of X and Y
$E(T_k^{SP}, R_l^{SP})$	Distance between $\sum_T(m,M)$ and $\sum_R(n,N)$
F	Sequence of DTW warping path $= \{c(1), c(2), ..., c(p), ...c(P)\}$
f_m	The m^{th} stage signal difference
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
$N(w)$	Normalization factor in DTW
n	Length of reference signal
l	Corresponding the start point (x_1, y_l) in reference signal which corresponding to $D_S(x_i, y_j)$
P	Path matrix
Q	Number of variables in process
R	A sampled signal with length r
R_n^{SP}	Singular points of signal $R \in \{R_1^{SP}, R_2^{SP}, ..., R_n^{SP}, ...R_N^{SP}\}\}$
S	Group separability ratio
T	A sampled signal with length t
T_m^{SP}	Singular points of signal $T \in \{T_1^{SP}, T_2^{SP}, ..., T_m^{SP}, ...T_M^{SP}\}$

w	Width of band global constraint
$w(p)$	Weight coefficient for local distance
X	Real-time signal segment $X = \{x_1, x_2, x_3, ...x_m\}$
x_H	High limit of sensor range
x_L	Low limit of sensor range
$x_i^{'}$	Normalized process variable
Y	Reference signal $Y = \{y_1, y_2, y_3, ...y_n\}$
y_{j*}	Point in reference signal that corresponds to x_m
Z	Segment of signal Y, $Z = \{y_l, y_{l+1}, ..., y_j\}$
$\Delta(x_i, y_j)$	Difference between x_i and y_j
α	Inseparability ratio = Ratio of normalized difference between best matching and second-best matching reference signals. $0 \le \alpha \le 1$
$\alpha_{\min}$	Minimum inseparability threshold
β	Duration of a singular episode
δ	Jump threshold for singular point identification
$\varepsilon\{\sum_T(m,k), \sum_R(n,l)\}$	Difference between signal segments $\sum_T(m,k)$ and $\sum_R(n,l)$
η	Normalized difference
τ	Size of neighborhood for singular point identification
τ_x	Time of the reference signal that corresponds to x_m
τ_y	Time of the reference signal that corresponds to y_{j*}
ϑ	State-differentiability
ς	Signal uniqueness
ω	Size of inspection window for singular point identification
$\kappa(T,R)$	Optimal distance between signal T and R
$\sum_R(n,l)$	Segment of signal R from R_n^{SP} to R_l^{SP}
$\sum_T(m,k)$	Segment of signal T from T_m^{SP} to T_k^{SP}

List of Figures

List of Tables

Chapter 1 Introduction

Modern chemical industries are large in scale and highly complex. Most continuous chemical process plants are operated in a multitude 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. Even when a transition is a desired change, there is often a flood of false alarms which distract the operators. This is because, at present, process automation applications like alarm management and advanced control are usually configured for a single operating state – typically a steady state mode. During transitions, operation errors are more likely to occur and equipments are likely to malfunction. Therefore, operators need more help than during steady state operation. But operations support systems have difficulties in working properly during transitions. There has been a large push consequently to create intelligent systems to manage transitions and detect faults during multiple state process operations. Early and accurate transition identification, fault detection and diagnosis can increase safety of process operation. It is also helpful in environmental protection and using resources effectively.

Online data from the process is a rich source of information and can be used for this purpose. Despite developments in state identification from online process measurements, many important and challenging problems still persist in this area. In this thesis, new methodologies for computationally efficient process state identification have been developed.

Signal comparison is important for process monitoring, fault diagnosis, and process stage identification. In Chapter 3, a new approach for signal comparison based on singular points and time warping has been presented. A robust method for uni-variate signal synchronization based on dynamic time warping (DTW) is proposed. The high computational complexity of DTW, which deters its widespread adoption, is significantly reduced by exploiting landmarks such as extreme values and sharp changes in the data, called singular points. Singular points are used to segment the process signal into regions, called episodes, with homogeneous properties. Comparison of signals is based on linking their singular points or episodes using dynamic programming. Time warping methods are used to match the corresponding episodes of the two signals. This two-step comparison approach leads to significant improvements in the speed, memory requirement, and efficiency of signal comparison. Another important advantage of the proposed approach is that since the singular points have physical meaning such as the beginning or ending of a process event, they can be directly used for state identification, monitoring, and supervision. A performance comparison of the singular points augmented time warping method with DTW reveals a substantial decrease in computational cost, which makes it amenable for large-scale case studies.

In Chapter 4, a new signal comparison-based approach, called dynamic locus analysis (DLA), for online state identification and fault diagnosis during process transitions has been proposed. Dynamic locus analysis is an extension of Smith and Waterman's (1981) 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. There are two problems, first, which part in reference signal

that corresponds to real-time signal. Second, real-time segment and corresponding part in reference signal would not match exactly due to noise and run-to-run differences. With dynamic locus analysis, all potential matching segments are compared for find the optimal results. Dynamic locus analysis is also used for segment synchronization make it robust to run-to-run differences and noise. 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.

The extension to online signal comparison has also been developed and reported in Chapter 5. During online process monitoring, there are two different stages in signal comparison – (1) Identifying the correct reference signal, and (2) Confirming that the real-time signal is similar with the previous identified reference signal. We solve the first stage using singular points, dynamic time warping, and dynamic programming. Dynamic programming is used to find the optimal linkage of the corresponding singular points between the real-time and reference signals and to calculate the extent of the real-time signal with respect to the reference. The total difference between the two signals is calculated using dynamic time warping. The total difference helps us identify the reference which is most similar to the real-time signal. A real-time extension of time warping has been used for the second stage of confirming the continued similarity with the same reference signal.

The large-scale and complexity of modern chemical plants makes it difficult for the operator to constantly monitor all process variables. Numerous methods exist for monitoring processes; however most of them suffer from computational complexity problems when applied to large-scale processes. In Chapter 6, a new approach for identifying a subset of the process variables, called key variables, which indicate the current processing state has been developed. Traditional process monitoring methods can then focus on this subset for effective monitoring. Key variables have been classified into six types and are determined using a hierarchical procedure that reflects the division of the process operation at different levels of granularity. The state-specific key variables provide a basis for defining key variables dynamically and solve the problem of desynchronization across different sections as well as improve the sensitivity of state identification.

All the above methods have been tested using three different case studies – operations stage identification during startup of ShadowPlant (a simulated fluidized catalytic cracking unit), disturbance identification in the Tennessee Eastman challenge plant, and faults identification during startup of a lab-scale distillation column. In all cases, the proposed methods correctly identified the corresponding points of the variables and found an operationally relevant signal difference.

Chapter 2 Literature Review

2.1 Signal Comparison

Modern chemical plants are large in scale and highly complex. 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 the plant operation. Potential areas of application of data-driven methods include process control, visualization of processing, operation improvement, and fault diagnosis. Data based approaches have been gaining in popularity due to significant developments in pattern classification (Webb, 2002) and statistical, information and systems theories (Chiang et al., 2001). Despite these developments in extracting information and knowledge from data, many important and challenging problems persist in knowledge extraction. In this thesis, we address one such problem – the comparison and matching of temporal signals in Chapter 3 and Chapter 5.

Dynamic Time Warping (DTW) is a popular method for signal comparison. In this thesis, we propose an extension of DTW that meets the above criteria for signal comparison.

2.1.1 Dynamic Time Warping

As described above, it is normal for two similar signals to be slightly different and not match each other perfectly. Comparison of signals with distortions is necessary for automatic word and speech recognition as well. Dynamic Time Warping is a robust method that has been widely used for matching speech patterns and calculating the difference between two signals. Two classes of DTW methods – Symmetric DTW and Asymmetric DTW – can be distinguished (Sankoff and Kruskal, 1983). A symmetric

algorithm treats the two signals equally, that is, both their time axis are mapped onto a common time axis and both patterns may be changed after the alignment. An asymmetric algorithm on the other hand, maps the time axis of the *test* signal onto the time axis of the *reference* signal. So the test signal will change to match the reference signal while the reference signal will remain unchanged. The asymmetric class is the one considered in this thesis for simplifying the comparison algorithm of DTW.

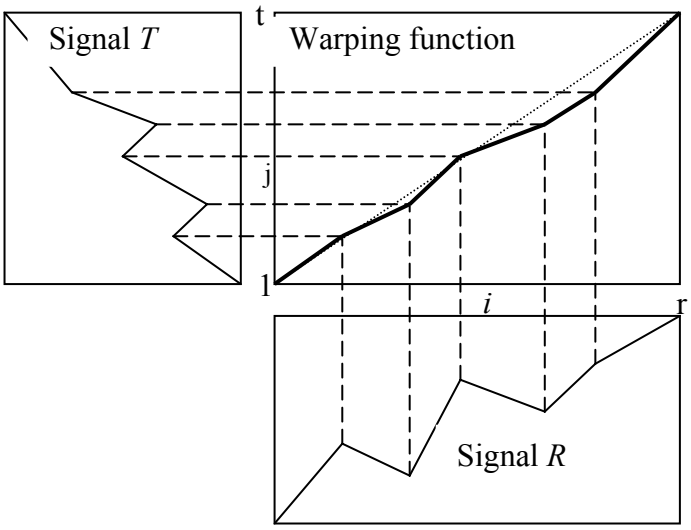


Figure 2-1: DTW of signal *T* on signal *R*

Let *T* and *R* denote two time-sampled signals of lengths *t* and *r*, and let *j* and *i* denote the time index of their trajectories, respectively. DTW finds a sequence *F** of *P* points on an *r*t* grid such that a total distance measure between the two trajectories is minimized as shown in Figure 2-1.

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

$$c(p) = [i(p)\, j(p)] \tag{2-2}$$

The minimum normalized distance $D^*(r,t)$ between the signals is found by warping their time axis and can be formulated as:

$$D^*(r,t) = \min_{F}[D(r,t)] \tag{2-3}$$

$$D(r,t) = \frac{1}{N(w)} \sum_{p=1}^{P} d[i(p), j(p)] * w(p) \tag{2-4}$$

Here, $D(r,t)$ is the normalized total distance between the two signals, $d[i(p), j(p)]$ is the local distance between the point *j(p)* of *T* and point *i(p)* of *R*, $w(p)$ 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. In this thesis, we have used $w(p) = 1$ for all cases. The optimal path F^* is found as $F^* = \underset{F}{\operatorname{argmin}}[D(r,t)]$.

Constraints are often used to define and restrict the search space and find an alignment that optimizes some criterion. They are motivated by physical considerations, to avoid excessive compression or expansion, speed up the calculation, or other problem specific limits on the alignment. As an example, endpoint constraints are commonly used in offline signal comparison and require the endpoints of S and T to match.

$$c(1) = (1,1) \ \ \& \ c(P) = (r,t) \tag{2-5}$$

Local constraints determine local features for each point. For example, the Sakoe-Chiba local constraint allows a point *(i, j)* in the grid to be reached from points *(i–1, j)*, *(i–1, j–1),* and *(i, j–1)*. The optimization problem in (1) is then transformed to the following problem, which can be solved using dynamic programming.

$$D_A(i,j) = \min \begin{Bmatrix} D_A(i-1,j) + d(i,j) \\ D_A(i-1,j-1) + 2 * d(i,j) \\ D_A(i,j-1) + d(i,j) \end{Bmatrix} \tag{2-6}$$

where $D_A(i,j)$ the minimum accumulated distance between (1,1) and the point *(i, j)*.

The Itakura local constraint defines a different set of predecessors – *(i–1, j)*, *(i–1, j–1),* and *(i–1, j–2)* and results in a local slope in [½ 2]. The optimization problem in (6) then changes to:

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

where $D_A(1,1) = d(1,1)$ and Condition (A*) indicates that the predecessor of point (*i–1, j*) is the point (*i–2, j*).

Another family of constraints – global constraints – defines the subset of the total search space for finding the optimal path. These are motivated by the fact that a wide search space is expensive in terms of both computation time and storage space. *Band global constraint* is a typical global constraint and it limits the maximum deviation of the optimal path from the linear one starting at *(1,1)* to a pre-specified amount, *w*. $w \geq |t-r|$ Global constraints are however not essential since the same objectives may be achieved through local constraints. More details of DTW can be found in Sankoff and Kruskal (1983) and Kassidas (1997).

Kassidas et al. (1998a) used DTW for synchronizing batch trajectories by combining it with multiway PCA/PLS. Kassidas et al. (1998b) reported its use for fault detection and diagnosis in continuous chemical processes and Nomikos and MacGregor (1994) for batch process monitoring. Li et al. (2004) combined DTW with

wavelet decomposition for synchronizing batch trajectories. The original signals were decomposed into approximations and details at different scales and matched at each scale separately using DTW. The matched signals were than reconstructed to obtain the synchronized signal.

In some situations DTW can fail to identify the correct correspondence between two signals. This would happen if the search range is not allowed to be sufficiently large. In situations with differences in the magnitude of the two signals, DTW would try to solve the variability in the Y-axis by warping the X-axis and thus result in inappropriate warping. The local nature of the search incorporated in DTW precludes a global perspective (See Section 3). Also, 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. The method proposed in this thesis is an alternative approach that constrains the search for the corresponding points of the two signals based on landmarks in the signal that are derived from operators' perspectives. These constraints can be used with DTW or other signal comparison approaches. We illustrate it using two variants of traditional DTW – DTW_1 based on Itakura local constraint and no global constraint, and DTW_2 based on Itakura local constraint with a band global constraint. Of the two, DTW_1 always find the minimum distance between two signals since it considers the whole search space which usually ensures the comparison is between the corresponding parts of the two

signals, but this makes the calculation slow especially for long signals. The search space considered by DTW_2 is determined by the search band B. A smaller B would require less calculation time, but may not eliminate all the differences between the two signals and result in a sub-optimal synchronization.

2.1.2 Signal Comparison based on Principal Components Analysis

Other methods for signal comparison based on Principal Components Analysis (PCA) have also been proposed in literature. In contrast to DTW which is based on the actual signal, these methods use the transformed principal components of the signals. Krzanowski (1979) proposed a PCA similarity factor that compares reduced subspaces of the original signals:

$$S_{PCA} = \frac{1}{k}\sum_{p=1}^{k}\sum_{q=1}^{k}\cos^2\theta_{pq} \tag{2-8}$$

where θ_{pq} is the angle between the p^{th} principal component of dataset S and q^{th} principal component of dataset T. Raich and Cinar (1997) used the PCA similarity factor for diagnosing process disturbances. Singhal and Seborg (2002) modified the PCA similarity factor by weighing the principal components with the square root of their corresponding eigenvalue, λ.

$$S_{PCA}^{\lambda} = \frac{\sum_{p=1}^{k}\sum_{q=1}^{k}\lambda_p^S\lambda_q^T\cos^2\theta_{pq}}{\sum_{p=1}^{k}\lambda_p^S\lambda_p^T} \tag{2-9}$$

The PCA similarity factor is only applicable for stationary signals. To extend them to non-stationary signals, Srinivasan et al. (2004) proposed a Dynamic PCA based similarity factor S_{DPCA}^{λ} that accounts for the temporal evolution of the signal. The main advantage of the PCA-based methods is their inherent ability to deal with multivariate signals and their low computational requirements. Their main shortcomings are: (1) they do not explicitly consider the synchronization problem; (2)

they are non-intuitive, especially for plant operators, since the comparison is based on a derived quantity with no physical significance; and (3) they consider the data as monolithic and arising from a single process state with specified statistical properties. This last requirement makes them unsuitable for online applications especially in multi-mode processes that can operate in multiple states. Also, for operator decision support, it is important to not only calculate the extent of similarity but also identify the *point of divergence*, i.e., the point in time from when the two signals start to deviate from one another. Since the PCA based methods consider the whole data as a single block they cannot directly detect the point of divergence.

2.2 Online Process State Identification

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 component analysis (PCA) have been widely used for process data classification, process 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. This is achieved by projecting the 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.

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 the runs built. One strong assumption of this method is that all batches have equal duration and all are synchronized. Undey and Cinar (2002) presented an adaptive hierarchical PCA for monitoring multi-stage processes. The progress of the process is modeled at each time instance by incorporating information from previous time slices.

Another family of data-driven approaches to fault diagnosis is based on signal comparison. These are based on the precept that the same types of faults or disturbances show similar features in the process signal. 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. One approach to overcome this synchronization problem is based on Dynamic Time Warping (DTW).

DTW has been used for fault detection and diagnosis in chemical processes by Kassidas (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. (2003) 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.

Trend analysis-based approaches adopt a different strategy to improve the computational performance of signal comparison. Rather than compare the raw signal, their abstraction them based on qualitative features – such as increasing trend,

decreasing trend, etc – is analyzed. Rengaswamy et al. (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 semi-quantitative features such as duration and magnitude of trends.

Long term process signal was used to identify process transition with these approaches. DTW needs the corresponding starting and ending points of the two signals to be known a priori.

2.2.1 Dynamic Programming Approaches to Discrete Sequence Comparison

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 maximally homologous (similar) subsequences among set of long discrete sequences. This 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 literature. One such is the dynamic programming approach of Smith and Waterman (1981). Let $A = \{a_1, a_2, a_3, ..., a_n\}$ and $B = \{b_1, b_2, b_3, ..., b_m\}$ 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 - 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:

$$H_{i,0} = H_{0,j} = 0, 1 \le i \le n, 1 \le j \le m \tag{2-10}$$

Other elements of H are calculated as

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

which can be rewritten in recursive form as

$$H_{i,j} = \max\{0, H_{i-1,j-1} + s(a_i, b_j), F_{i,j}, G_{i,j}\} \tag{2-11}$$

Where:

$$F_{i,j} = \max_{1 \le k \le i}\{H_{i-k,j} - w(k)\} \tag{2-12}$$

$$G_{i,j} = \max_{1 \le k \le j}\{H_{i,j-k} - w(k)\} \tag{2-13}$$

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; while $F_{i,j}$ 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 value are than 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 $s(a_i, b_j) = -1/3$ otherwise. $w_k = 1 + k/3$. The H matrix shown in Table 2-1 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 (4th and 6th 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.

Table 2-1: 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 this thesis, we extend the above algorithm for discrete sequences 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 in a long historical reference signal. The minimal difference between the real-time and reference signals reveals the process state (for eg, normal vs. abnormal, identity of transition, etc) and also an estimate of its extent of progression (from the relative position in the reference signal).

2.3 Key variables selection for Complex Chemical Process

In a complex chemical process, there are a lot of variables. The information available from each variable is different. During certain operations or for certain purposes, some variable can give much more useful information than others. These variables are called key variables for this purpose.

There have been some previous works on key variables selection for different purposes. The common approach of using the magnitude or range of variation of the variable as a measure of its importance is not a robust indicator of a variable's importance. Some variables like the temperature of the delayed-coking furnace vary over a small range during normal operation. However, these variables could be very important to the safety and efficiency of the operation and even minor changes can be detrimental. Thus, the effect of the change on the system should be used to identify key variables. Yuan and Klir (1997) presented a method for determining the key variables that contribute most to a specific partition of data. Their method is based on fuzzy c-means algorithm. The contribution of each variable to a partition is inferred from the optimal Mahalanobis distance. Key variables were selected based on the variables' affects on a specific partition of the data.

Some effort has been devoted to finding the key variables for partial process control, design and monitoring. Complex chemical process having a large number of process variables but poorly understood models can be controlled reasonably by controlling only a small subset of process variables. This is referred to as partial control. Kothare et al. (2000) gave a definition of the partial control problem. They introduced concepts such as variable dominance, and modelable responses that is also useful in selecting key variables for monitoring chemical process transition. A subset of process variables is said to be dominant for a given process if that subset is preponderant in achieving the specified process objectives. Arbel et al (1995, 1996, and 1997) gave a good example of partial control using a seventeen variable FCCU model.

Many researchers have addressed optimal sensor placement. Kretsovalis and Mah (1987) proposed a sensor-placement strategy based on the precision of the reconciled variables. Bagajewicz (1997) formulated this as a capital cost optimization problem subject to reconciliation precision bounds. This problem is defined as a mixed-integer nonlinear program (MINLP). Bhushan and Rengaswamy (2000) proposed a method for selecting the optimum number of sensors for a given process based on a process digraph. Bhushan and Rengaswamy (2002) proposed a method to design a sensor network for chemical plant based on various diagnose ability and reliability criteria. A methodology for obtaining the best sensor location irrespective of fault assumption was presented in that paper. Reliability maximization was achieved on a optimization framework for sensor location from a fault diagnosis perspective. A minimum-cost model that minimizes the cost of the fault monitoring system was also presented in that paper. They applied the sensor location procedure to the Tennessee-Eastman process.

Sadeghbeigi (2000) presented a principle for monitoring FCCU. He said that periodic material and heat balance survey on the unit was the only proper way to monitor the performance of a FCCU. All the operation of FCCU was based on three balances: material balance, heat balance, and pressure balance. Understanding of the heat balance was deemed very critical for the operator since any change to feedstock quality, operating conditions, and catalyst flow change will affect the heat balance. These principles are therefore useful for key variables selection.

Methods for reduced variable dimensional space like principal component analysis (PCA) and independent component analysis (ICA) have also been used for finding key variables (Hyvarinen, 2000). PCA aims to find uncorrelated principal components that are linear combinations of observed variables while ICA is designed to separate components that are independent and constitute the observed variables (Li and Wang, 2002). The problem with PCA and ICA based methods is that the new signal has no physical meaning. There are also difficult to use for operator monitoring. Another problem is that there are some signals that may not vary extensively but they may be important during certain operations. PCA and ICA based methods will miss such variables.

Finally, all the above methods consider the process as stationary - i.e., the same key variables are used in different process states. In this thesis, we define state-specific key variable and develop a systematic method for identifying the key variables

Chapter 3 Offline Temporal Signal Comparison Using Singular Points Augmented Time Warping

3.1 Introduction

Advances in instrumentation and data storage technologies have allowed the process industries to collect extensive operating data which can be used for extracting information about the underlying process.

One class of data-driven methods takes advantage of the notion that in many engineering problems, similar process changes – desired or undesired – usually result in similar evolution of process variables. The basic precept of these methods is that if a representative historical database of signals has been previously analyzed and suitably annotated, it can be used to identify the root cause of a change and develop an effective remedy. The online problem then is to locate an instance in the historical database that is most similar to a specific data. Pattern classification or signal comparison is a popular method for finding similar signals in historical data. The challenge in this approach arises from the fact that due to the nature of industrial processes, signals arising from two instances of the same change are not exact replicates – invariably there are deviations between the two instances. The differences could be in the length (total-time) of the two signals; duration of the constituent stages (or phases); or in the magnitudes or profiles of the variables due to run-to-run variations arising from impurities, initial conditions, seasonal effects, or operator actions. Direct comparison of two signals would therefore be incorrect since there is no guarantee that the corresponding segments of the signal are being compared. Robust yet sensitive

methods for comparing such unsynchronized signals are therefore an active area of research.

From the above review, it is apparent that an ideal signal comparison method should be:

1. **Robust:** It should be able to match corresponding parts of two signals and be robust to magnitude and duration differences, noise, and other run-to-run variations.
2. **Sensitive:** It should be sensitive to structural differences such as unmatched trends in the signals.
3. **Consistent with process state:** The correspondence between two signals should be based on the process state; that is, for signals from multi-state operations, signal segments from the same process state should be compared. This would ensure results that are compliant with operator's intuition, which is important for acceptability of the results (Jain et al., 2000).
4. **Adaptive:** The method should not require precise knowledge of the end-points of the signal. It is not easy to identify the exact start and end points of signals to be compared, especially during online signal comparison. This requirement is less critical for offline usage and will not be considered in this thesis.
5. **Computationally inexpensive:** The computational load should be modest both in terms of memory and time required.

Towards these objectives, we propose a time warping-based signal comparison approach in this thesis. The aforementioned deficiencies of DTW are overcome by augmenting the comparison with singular points.

Dynamic Time Warping (DTW) is a popular method for signal comparison. In this thesis, we propose an extension of DTW that meets the above criteria for signal

comparison. Singular points are defined in Section 3.2. Algorithms for identifying singular points are also described. For signal comparison, dynamic programming is used to find the optimal link between the singular points of two signals. This yields the corresponding segments of the two signals which can then be compared through time warping. This singular points and time warping based signal comparison methodology is proposed in Section 3.3. A detailed description of data generation is given in Section 3.4. In Section 3.5, we illustrate the proposed approach using signals from three case studies – the Tennessee Eastman process simulation, simulation of a fluidized catalytic cracking unit, and a lab-scale fermentation process.

3.2 Singular Points

Information content is not homogenously distributed throughout a signal; rather the majority of the features of the signal are concentrated in a small number of points. In this thesis, we term such points, which are landmarks in the signal evolution, as *singular points*. This philosophy can also be noted in Bakshi and Stephanopoulos' (1996) use of inflexion points to separate signal segments and Mallet and Hwang's (1992) use of singularity as the compact representation of a signal. Our singular points consist of points of discontinuities, trend changes, and extrema. The singular points of a sample signal are shown in Figure 3-1. These landmarks can be used to segment the signal into portions with homogenous properties. The segment of a signal between adjoining singular points is called a *singular episode*. An episode thus consists of regions of nearly-constant slope, small oscillations, etc.

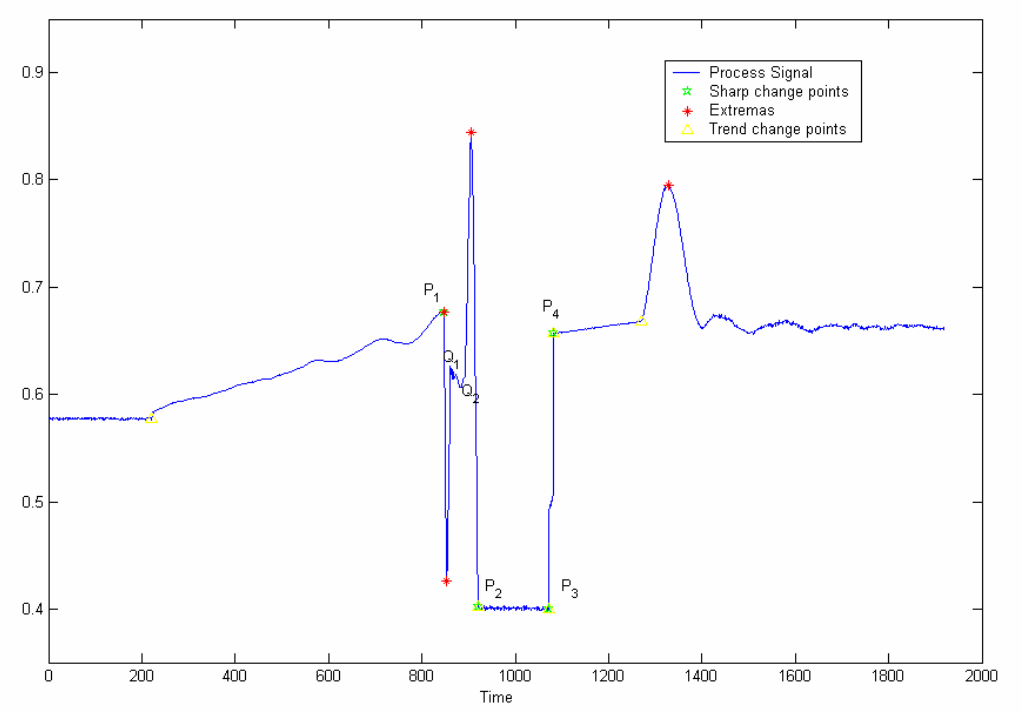


Figure **3-1:** A typical signal and its singular points

Mathematically, each singular point is a triplet $T^{SP} = \{\Gamma, \Omega, \Psi\}$ where Γ is the time of occurrence of the singular point, Ω is the magnitude of the variable at the singular point, and Ψ its type. Three types of singular points can be differentiated as (1) extrema, where the derivative of the variable changes sign (indicated by $\Psi = e$); (2) sharp changes and discontinuities, where the process variable changes by a large amount in a short period of time ($\Psi = s$); and (3) trend change points ($\Psi = t$). The reader may note that these three types broadly correspond to local extrema in the variable, its first derivative, and second derivative, respectively. In addition to these three types, for ease of explanation, the beginning and end of each signal is also nominally considered to be singular points of type $\Psi = b$ and $\Psi = c$, respectively. It should be noted that the different types of singular points are not mutually exclusive and the same point may meet more than one of the above criteria. This is illustrated in Figure 3-1 – P_1 is both an extreme point and a sharp change point; similarly, P_2 and P_4 are trend change points as well as sharp change points. In such cases, the list of all

matching types is noted. The term '*singular points of a signal'* refers to the ordered list of triplets. In the following, we describe how singular points can be identified.

3.2.1 Methods for Identifying Singular Points

Consider a signal $R = \{x_1, x_2, ..., x_i, ...x_r\}$ and ΔR be the first derivative of R.

$\Delta R = x_i - x_{i-1} \quad i = 2, 3, ..., r$

Sharp changes usually occur over a period of time. The beginning and end points of this period are marked as sharp change singular points. Sharp changes are identified based on the signal range within an *inspection window W* of size ω. If the difference between the maximum value and the minimum value within the window is larger than a pre-specified *jump-threshold* δ then the window is considered to have a sharp change episode. A period containing a sharp change is characterized by the beginning and end points of the episode. If adjoining sharp change points fall within a *neighborhood* of size τ, then they are concatenated into one sharp change episode.

Noise will show on ΔR difference. The change will take the total different between the data within the *inspection window W* which reduce the effect of random noise. *Jump-threshold* δ is also affected by the signal noise level which will also reduce the effect of noise in identifying sharp changes

Extrema points can be identified directly from sign changes in ΔR, but this would be susceptible to noise and a large number of local maxima and minima will be identified. In order to overcome this, we require additionally that the difference between adjoining extrema points is larger than $\delta / 2$. Also, extrema consistency, that is, alternating maxima and minima, is enforced among the identified extrema.

The δ chosen is based on the signal feature and noise level. In this thesis, the difference between real signal and DWT de-noise signal was used as the reference and a minimum value was also given

Trend change points are singular points where the trend of the signal undergoes a change. These are identified through statistical analysis and linear regression of the signal over a neighborhood of size τ ($\tau > \omega$). A point is considered to be a trend change point if all its left neighbors lie within three standard deviations from the regressed model, while all its right neighbors lie outside, or vice versa. These algorithms are applied to each signal to identify all the three types of singular points. The algorithm for Singular points detection is attached in Appendix A.

3.2.2 Properties of Singular Points

The following properties follow from the definition of singular points:

1. If two signals completely overlap, their singular points will be the same.
2. If two signals have the same magnitude profiles and the only difference between them is in the duration of the episodes, their singular points will have the same Ω and Ψ; the differences will only be in the Γ.
3. If two signals have the same general profile, but vary in magnitude and duration of episodes, their singular points will vary in Ω and Γ; the sequence of Ψ will be the same.
4. If two signals are completely different, their singular points will be uncorrelated.

These properties of singular points make them robust to noise. This is illustrated using an example. Table 3-1 shows the singular points of a signal when corrupted with varying levels of noise (from 0 to 10%). As can be seen, the same number of singular

points is identified in all cases. The types of the different singular points at all the noise levels are also the same. The error in Ω, that is the difference in the magnitude of the singular point in the signals with and without noise, varies from 0.4% to 2.63% as noise level increases from 1% to 10%. The error in Γ, that is the position shift of the singular points, varies from 1 to about 4 samples in a signal of length 1920 samples. These show that singular points can be robustly identified even if the underlying signal is noisy. We elaborate on robustness during signal comparison in Section 3.5.

Table 3-1: Singular points of a signal with different noise levels

Noise free signal			1% noise			3% Noise			5% Noise			10% Noise		
Γ	Ω	Ψ	Γ	Ω	Ψ	Γ	Ω	Ψ	Γ	Ω	Ψ	Γ	Ω	Ψ
221	0.5767	*t*	222	0.5809	*t*	239	0.5775	*t*	222	0.5798	*t*	224	0.5835	*t*
847	0.6772	*s*	844	0.6817	*s*	842	0.683	*s*	841	0.671	*s*	830	0.7194	*s*
847	0.6772	*e*	847	0.6765	*e*	845	0.6909	*e*	847	0.6982	*e*	843	0.6738	*e*
853	0.4259	*e*	853	0.4304	*e*	854	0.4325	*e*	854	0.4469	*e*	853	0.3889	*e*
905	0.8441	*e*	905	0.8484	*e*	905	0.8391	*e*	905	0.8366	*e*	906	0.8565	*e*
920	0.4015	*s*	920	0.3971	*s*	920	0.4063	*s*	920	0.3861	*s*	919	0.4536	*s*
920	0.4015	*t*	920	0.3971	*t*	920	0.4063	*t*	927	0.3755	*t*	927	0.3978	*t*
1072	0.3999	*s*	1072	0.4046	*s*	1067	0.3968	*s*	1066	0.405	*s*	1069	0.379	*s*
1072	0.3999	*t*	1072	0.4046	*t*	1072	0.3982	*t*	1072	0.3905	*t*	1072	0.3959	*t*
1083	0.6569	*s*	1083	0.6579	*s*	1083	0.6638	*s*	1083	0.6684	*s*	1083	0.622	*s*
1083	0.6569	*t*	1084	0.6523	*t*	1085	0.6643	*t*	1091	0.6645	*t*	1091	0.7037	*t*
1271	0.6682	*t*	1269	0.6655	*t*	1256	0.6609	*t*	1263	0.6787	*t*	1276	0.6371	*t*
1329	0.7948	*e*	1322	0.7983	*e*	1320	0.8032	*e*	1325	0.8171	*e*	1329	0.8422	*e*
Error			1.076	0.0037		1.615	0.0058		3.150	0.0128		3.769	0.0263	

3.3 Signal Synchronization and Comparison Using Singular Points

Singular points can be used to synchronize and compare signals. Consider the synchronization of a test signal *T* with a reference signal *R*. Let the singular points of the test signal *T* be $(T_1^{SP}, T_2^{SP}, ...T_m^{SP}, ..., T_M^{SP})$ $M \ll t$. Let the singular points of the reference signal *R* be $(R_1^{SP}, R_2^{SP}, ...R_n^{SP}, ..., R_N^{SP})$ $N \ll r$. The singular episode between T_m^{SP} and T_k^{SP} *(m<k)* is noted as $\sum_T(m,k)$. The segment of the signal from T_m^{SP} to the end of the signal (T_M^{SP}) is reported as $\sum_T(m,M)$. Let $\varepsilon\{\sum_T(m,k), \sum_R(n,l)\}$ be the

distance between the episodes $\sum_T(m,k)$ and $\sum_R(n,l)$ of signals T and R, respectively. The *accumulated distance* between $\sum_T(m,M)$ and $\sum_R(n,N)$ is indicated by $E(T_m^{SP}, R_n^{SP})$. From the geometry of the signal, it can be seen that, if $m < k$ *and* $n < l$, then

$$E(T_m^{SP}, R_n^{SP}) = \varepsilon\{\sum_T(m,k), \sum_R(n,l)\} + E(T_k^{SP}, R_l^{SP}) \qquad (3\text{-}1)$$

In contrast to DTW where the entire signal is monolithically compared, in the proposed method, comparison is made only between *corresponding* singular episodes. For this purpose, the corresponding singular points and episodes of the two signals have to be *linked*. As an illustration of this, for the signals in Figure 3-2, the episode A-1 of T is homologous to episode B-1 of R although the two have duration and magnitude differences. So they should be linked. Similarly episodes A-2 and B-2 in the figure should be linked.

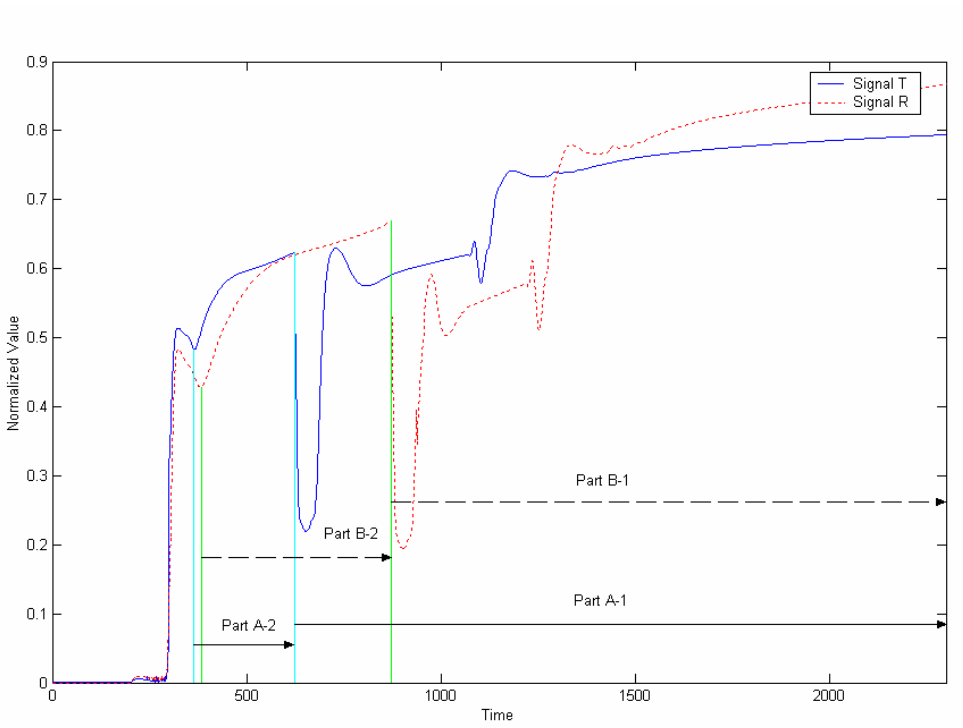


Figure 3-2: Illustration of signal comparison using singular points

In the general case, many sets of links between the singular points of the test and reference signals may be possible. Further, the corresponding singular episodes of the two signals may vary in magnitude and length but can be synchronized using time warping. Sequential decision making is therefore used to find the linkage that minimizes the total distance between the two signals. The total distance between the test signal and reference signal is calculated as the sum of the episode-wise time-warped distances.

The following constraints are applied when linking singular points:

1. For any pair of singular points (T_m^{SP}, R_n^{SP}) (T_k^{SP}, R_l^{SP}) in the two sequences, if $m < k$ then $n < l$
2. Corresponding singular points in the two series should be of the same type.
3. A singular point in one series *may* not have a corresponding singular point in the other. This could be because of run-to-run variations or the singular point may be unidentified because of noise. Such cases are considered as a link with a [Empty] singular point.

The sequential decision making problem for synchronizing the two signals is solved in a stage-wise manner using dynamic programming. The decision d_m at stage m is to find the optimal singular point of R (say R_l) that should be linked to T_m^{SP}. The signal distance for a decision at stage m is calculated as the resulting time warped distance from $\sum_T(m-1, m)$. We use r_m to represent the m^{th} stage of the process. Let f_m be the total return (or signal distance) over m stages using the optimal decision. Mathematically (Williams, 1970):

$$f_M = \min_{d_M, d_{M-1}, \ldots, d_2, d_1} \{r_M + r_{M-1} + \ldots + r_2 + r_1\} \tag{3-2}$$

To obtain a recurrence relation for f_M we make the trivial observation that the M^{th} stage return does not depend upon d_{M-1}, d_{M-2}, etc, but only on d_M, so that

$$f_M = \min_{d_M}\{r_M + \min_{d_{M-1},\ldots,d_2,d_1}(r_{M-1} + r_{M-2} + \ldots + r_2 + r_1)\} \tag{3-3}$$

which can be rewritten as

$$f_M = \min_{d_M}\{r_M + f_{M-1}\} \tag{3-4}$$

Based on the Principle of Optimality (Williams, 1970), the optimal linkage of singular points is the one that results in minimum $E(T_1^{SP}, R_l^{SP})$ where l is all the possible corresponding singular points to T_1^{SP}. In this chapter, we assume that for offline signal comparison the corresponding starting points of the two signals are known *a priori*, the optimal distance between signals T and R $\kappa(T,R)$ is therefore $E(T_1^{SP}, R_1^{SP})$. In the following, we explain how this can be implemented.

3.3.1 Algorithm for Signal Comparison Using Singular Points Augmented Time Warping

The singular points of the two signals are linked as follows:

Step 0: **Normalization**: Identify the singular points of T. Normalize signal *T* to [0 1] based on the sensor range of the signal (See Srinivasan et al. 2004).

Step 1: **Initiation**: Start the search for optimal link from the last episode of signal T $\sum_T(M-1, M)$. This is deemed as Stage M. Set *m=M*, *l=N,* and $E(T_M^{SP}, R_N^{SP}) = 0$.

Step 2: **Stage-wise Propagation** Stage m: We need to find the corresponding episode in signal R for episode $\sum_T(m-1, m)$ of signal T. Define β as the duration of

the episode $\sum_T(m-1,m)$. Find all the corresponding singular points (based on the constraints above) in signal R within the *search window* $[R_l^{SP}-2\beta \;\; R_l^{SP}-\frac{1}{2}\beta]$. Here, we have allowed a singular episode to elongate or compress by a maximum factor of 2. This factor is found to be adequate for process signals; but any other factor can also be used.

For each possible corresponding singular point in Signal R, say R_n^{SP}, calculate the local episode distance $\varepsilon\{\sum_T(m-1,m),\ \sum_R(n,l)\}$ using time warping. The possibility that the corresponding singular point in signal R is missing is also explicitly considered. In this case, time warping is used to find the matching point in R (may not be a singular point) that has the smallest distance from $\sum_T(m-1,m)$. Note that a missing singular point in signal *T* would not affect the result since the comparison will consequentially be made on a longer signal segment but still bounded by singular points at either end. The accumulated distance is obtained from (3-1) as

$$E(T_{m-1}^{SP},R_n^{SP})=\varepsilon\{\sum_T(m-1,m),\sum_R(n,l)\}+E(T_m^{SP},R_l^{SP}).$$

Resolving multiple paths: It is possible that there are multiple paths between a pair of linked singular points in T and R. In order to minimize the search computation, one can take advantage of the Principle of Optimality by which the link with the smallest accumulated distance is globally optimal and is the only one which has to be considered in further steps.

Step 3: **Termination**: Repeat Step 2 until the first singular point in T, T_1^{SP} has been linked. That is, if $m < 1$ then set $m = m - 1$ and go to Step 2.

Step 4: **Optimal link**: Among the various paths, the one with the smallest total accumulated distance is the optimal one. The minimum $E(T_1^{SP}, R_1^{SP})$ is deemed as the signal distance $\kappa(T, R)$. The optimal linkage of the singular points is found by retracing the decisions from T_1^{SP} to T_M^{SP}.

In the following, we illustrate the above algorithm.

3.3.2 Illustration of Linking Singular Points

Consider the two signals in Figure 3-3 that need to be synchronized. Their singular points are marked in the figure and need to be linked. The endpoints are known to correspond, so T_{13}^{SP} links to R_{15}^{SP} and T_1^{SP} to R_1^{SP}. Only the remaining points need to be linked. The linking process can be visualized using a tree, called the *singular points linkage tree*, an example of which is shown in Figure 3-4. In this tree, the singular points of the test signal are shown as stages in the left side. The different possible links with *R* at each stage are shown as the nodes at that level of the tree. An arc connects the linkage at a given stage to the previous decisions (or linkages). There are twelve search stages in this case as shown in Table 3-2 and depicted by the partial linkage tree in Figure 3-4.

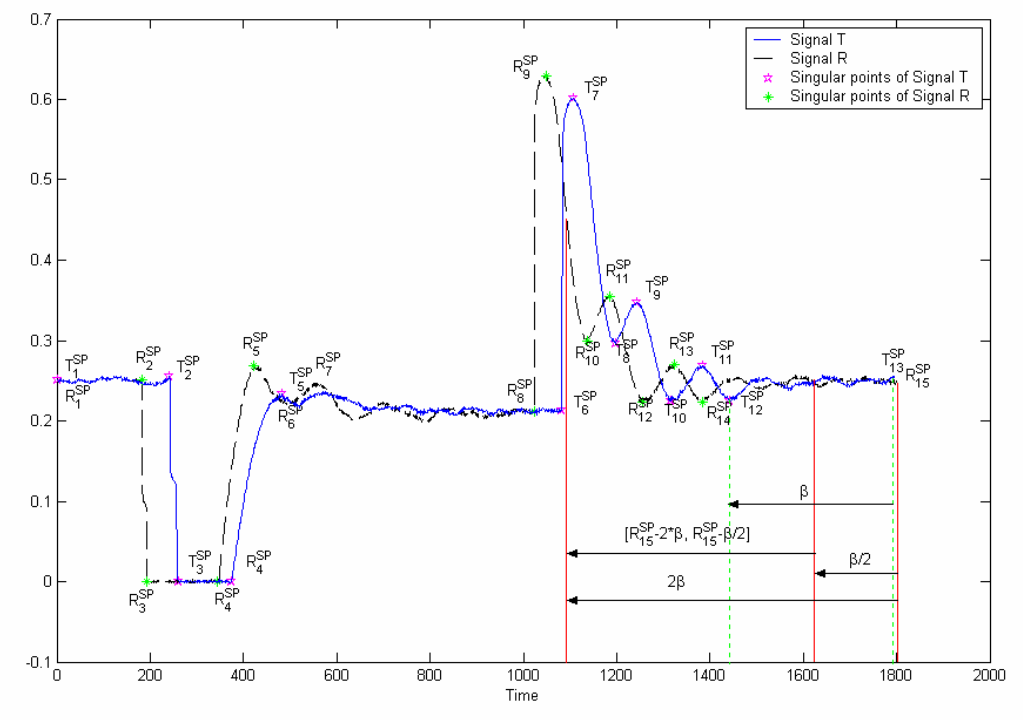


Figure 3-3: Two signals and their singular points

Table 3-2: Stage-wise linkage of singular points of *T* and *R*

Stage *m* (Singular Point in T)	**Possible links in R *(l)***												$E(T_m^{SP}, R_l^{SP})$
12	10												9.0824
	12												1.5799
	14												0.255
	EMPTY												0.2545
11	10	9											22.1186
	12	11											3.508
	14	13											0.2905
	14	EMPTY											0.2901
10	12	11	10										8.9268
	14	13	12										0.3501
	14	13	EMPTY										0.3432
9	12	11	10	9									19.7158
	14	13	12	11									0.4572
	14	13	12	EMPTY									0.4256
8	14	13	12	11	10								0.5613
	14	13	12	11	EMPTY								0.5499
7	14	13	12	11	10	9							0.9858
	14	13	12	11	10	EMPTY							0.7303
6	14	13	12	11	10	9	8						1.5472
	14	13	12	11	10	EMPTY	EMPTY						1.7527
5	14	13	12	11	10	9	8	5					4.363
	14	13	12	11	10	9	8	7					5.2568
	14	13	12	11	10	9	8	EMPTY					3.4849
4	14	13	12	11	10	9	8	5	4				4.998
	14	13	12	11	10	9	8	7	EMPTY				12.7948
3	14	13	12	11	10	9	8	5	4	2			5.9973
	14	13	12	11	10	9	8	5	4	3			5.0573
	14	13	12	11	10	9	8	5	4	EMPTY			5.0448
2	14	13	12	11	10	9	8	5	4	3	2		5.4643
	14	13	12	11	10	9	8	5	4	2	EMPTY		8.1623
1	14	13	12	11	10	9	8	5	4	3	2	1	5.8419
	14	13	12	11	10	9	8	5	4	2	EMPTY	1	8.7146

1. The search starts from the last unlinked singular point of T (T_{12}^{SP}) and proceeds to the first (T_1^{SP}). Matching begins at the last episode of the reference signal $\sum_T(12,13)$.
2. It is required to find the corresponding episode in signal R that yields the lowest error. Since the endpoint of signal R is known to be R_{15}^{SP}, only the beginning of the episode has to be identified. This can be considered to be akin to finding the singular points in signal R that can link with T_{12}^{SP}. As shown in the first row of Table 3-2 and the top of Figure 3-4, there are four possible choices – R_{10}^{SP}, R_{12}^{SP}, R_{14}^{SP}, and [Empty].
3. For each possible link, for example, T_{12}^{SP} and R_{10}^{SP}, the distance between the corresponding episodes of the two signal is calculated using DTW_1. In this case, $\varepsilon\{\sum_T(12,13),\sum_R(10,15)\} = 9.0824$. The accumulated distance $E(T_m^{SP}, R_n^{SP})$ is shown next to the arc in Figure 3-4 and in the right side of Table 3-2.
4. The above process is repeated for the preceding singular points of T, e.g. T_{11}^{SP}. At this stage, the previous linkages (say T_{12}^{SP} to R_{10}^{SP}) will constrain the choices. The previous decision is shown by the arc in Figure 3-4. The accumulated distance for the T_{11}^{SP} – R_9^{SP} link is calculated as $E(T_{11}^{SP}, R_9^{SP}) = \varepsilon\{\sum_T(11,12),\sum_R(9,10)\} + E(T_{12}^{SP}, R_{10}^{SP}) = 13.0362 + 9.0824 = 22.1186$.
5. At any stage, it is possible that a singular point in signal T is linked to the same singular point in R by multiple paths. For example, R_{12}^{SP} can be linked with T_{10}^{SP} in Stage-3 by two paths – $\{(T_{12}^{SP}, R_{14}^{SP}), (T_{11}^{SP}, [\text{Empty}]), (T_{10}^{SP}, R_{12}^{SP})\}$ and

$\{(T_{12}^{SP}, R_{14}^{SP}), (T_{11}^{SP}, R_{13}^{SP}), (T_{10}^{SP}, R_{12}^{SP})\}$. Since the link in the latter path has the smaller distance (0.3566 vs. 0.3501), it is optimal and the former link need not be considered in subsequent steps.

6. The entire singular point linkage tree is constructed by repeating this process. The detailed results for all stages are shown in Table 3-2.
7. The optimal set of assignment is the path with the smallest accumulated distance for $D(T_1^{SP}, R_1^{SP})$. A path in the tree starting from a Stage 1 node and ending in a last stage node shows the linkages for one set of assignment. Many feasible paths may exist; the one with the least accumulated distance $E(T_1^{SP}, R_1^{SP})$ is the optimal one. In this case, of the two possible paths, the path $\{(T_{12}, R_{14}), (T_{11}, R_{13}), (T_{10}, R_{12}), (T_9, R_{11}), (T_8, R_{10}), (T_7, R_9), (T_6, R_8), (T_5, R_5), (T_4, R_4), (T_3, R_3), (T_2, R_2), (T_1, R_1)\}$ with $\kappa(T, R) = E^*(T_1^{SP}, R_1^{SP}) = 5.8419$ is the optimal one. It should be noted that in this path, there are no corresponding points in Signal T for R_6 and R_7 because of run-to-run variation. This is automatically identified by the method when deriving the optimal linkage.

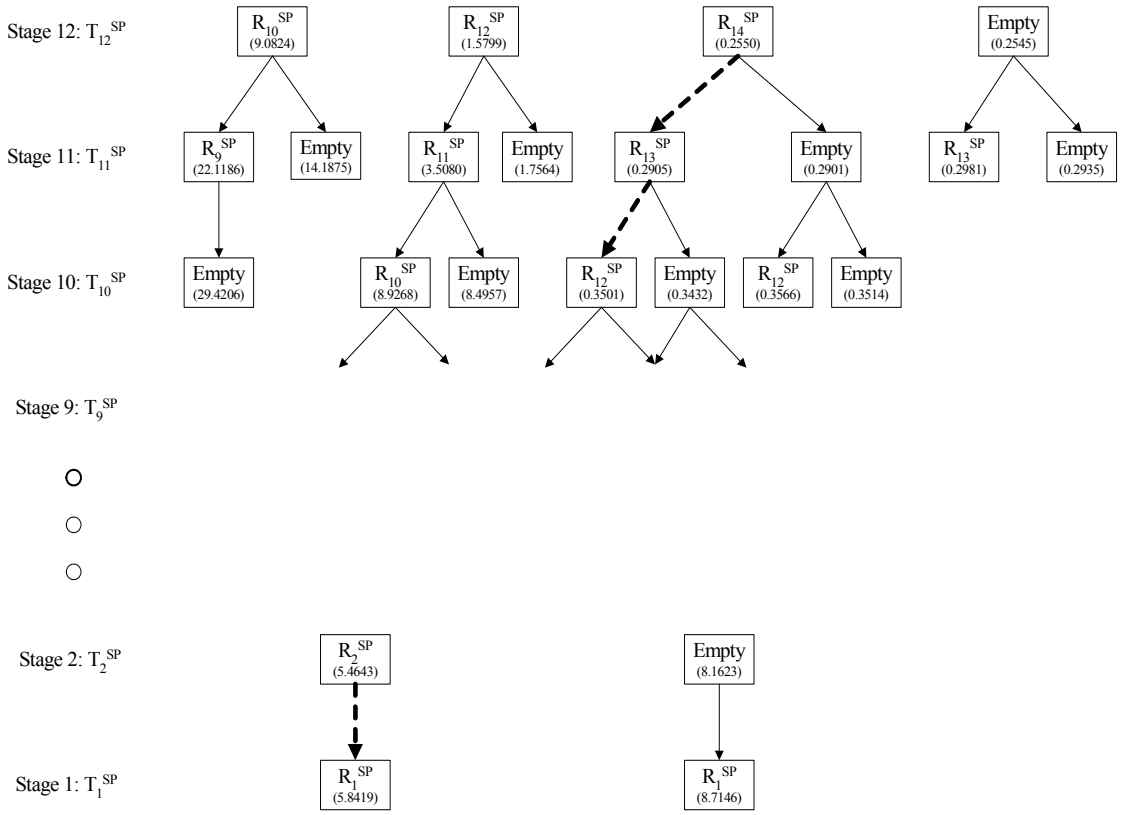


Figure 3-4: Singular points linkage tree

3.3.3 Extrapolative Time Warping: An Efficient Algorithm for Episode Comparison

The traditional DTW used above for calculating the distance between episodes provides global optimality, but at a high computational cost. The singular points method proposed here enforces the optimal linkage of the major landmarks of the two signals using dynamic programming. The global optimality of the episode-level comparison is therefore not a critical requirement since the optimal assignment of each time point within an episode has no physical significance and is rarely necessary in practical applications. The steep computational cost for episode comparison using DTW motivates an alternative algorithm for episode comparison as described next.

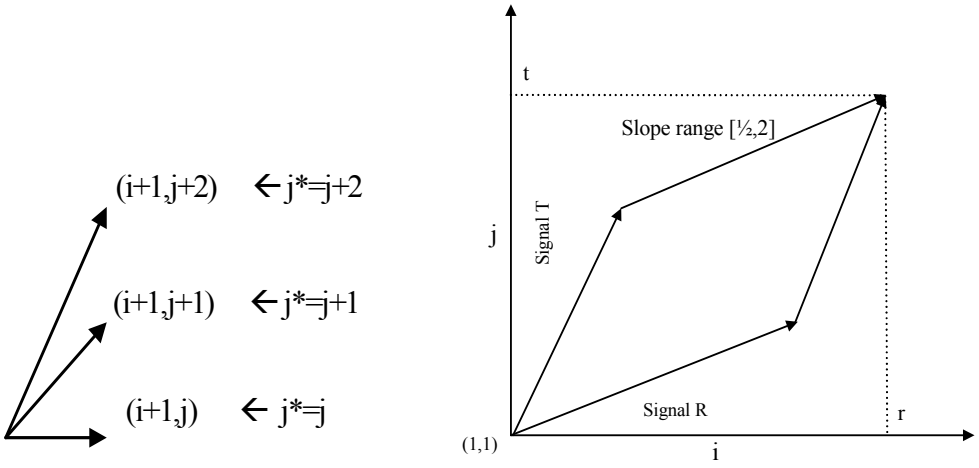


Figure 3-5: Search space for XTW and local constraints

An efficient signal comparison algorithm, called *extrapolative time warping* (XTW), based on a greedy search modification of classical DTW with Itakura local constraint is proposed here. The XTW method obviates dynamic programming for each local point by optimizing each point *locally*. In contrast to DTW, in XTW, search proceeds in the forward direction starting from the first point of the signal to the last. As shown in Figure 3-5(a), the local search is formulated as follows. Let *(i, j)* be a warping assignment time at a given step. In XTW, the optimal warping for the subsequent step, i.e., the location of j^* that corresponds to *(i+1)*, is based only on the previous decision and the current distance. Three possible successors – (*i+1, j*), (*i+1, j+1*) and (*i+1, j+2*) – are considered. The optimal search path is therefore defined as:

$$D_A(i+1,j^*) = \min \begin{Bmatrix} D_A(i,j)+d(i+1,j) & \text{or} & [\infty \text{ if Condition}(B^*)] & j^*=j \\ D_A(i,j)+d(i+1,j+1) & & & j^*=j+1 \\ D_A(i,j)+d(i+1,j+2) & & & j^*=j+2 \end{Bmatrix} \quad (3\text{-}5)$$

With initial condition $D_A(1,1) = d(1,1)$, Condition (B*) indicates that the predecessor of point $j^*=j$. Thus, for each step, the decision for the corresponding point for *i* is based only on three comparisons – to increase *j* by 0, 1 or 2. Following the Itakura local constraint, if the previous decision was to increase *j* by 2, then the

successor would not have this option (so as to maintain the local slope in the [½ 2] range) and j can increase only by 0 or 1. Similarly, if in the preceding step, if j did not increase, the successor would not have the option to remain at the same j and $j^*=j+1$ or $j^*=j+2$. Since any decision is based only on the previous decision and the current difference, dynamic programming is obviated. The search space of the XTW is the same as DTW with the Itakura local constraint and is shown in Figure 3-5(b). However, unlike DTW, in XTW once a match has been assigned, future assignments will not affect it. The greedy extrapolative search for any point decreases the computational time, but since global information is not used at each step, an optimal solution is not guaranteed by XTW. The assignment history tree, which is the origin for the large computational storage requirements of DTW, is not necessary in XTW; instead only the list of assignments needs to be maintained.

It has been shown in Section 3.5 that both XTW and DTW result in the same warping when signals differ only in synchronization. When XTW is used for comparing signals with magnitude differences, it places a larger emphasis on local magnitude similarity instead of globally optimal warping. But since in our approach, XTW is combined with the singular points linkage described above, and the latter incorporates the global information and optimization, this non-optimality of the signal comparison is not critical. Rather, it provides the computational speedup for large-scale signal comparison as illustrated next. In the following section, we use the proposed singular points augmented DTW algorithm for state identification, clustering, and transition identification in three different case studies.

3.4 ShadowPlant and Tennessee Eastman Process Description

In this thesis, all the research is based on process data. There are several simulators and lab-scale pilots are used for generating data. The main data for this

thesis is from ShadowPlant and Tennessee Eastman process. In this section, the details of process data generation are discussed.

3.4.1 ShadowPlant: A simulator of Fluidized Catalytic Cracking Unit (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, 2000). The ShadowPlant consists of five main sections as shown in Figure 3-6:

(1), Feed preheater, (2), Riser/Regenerator, (3), Main fractionator, (4), Waste heat boiler, (5) Air-preheater.

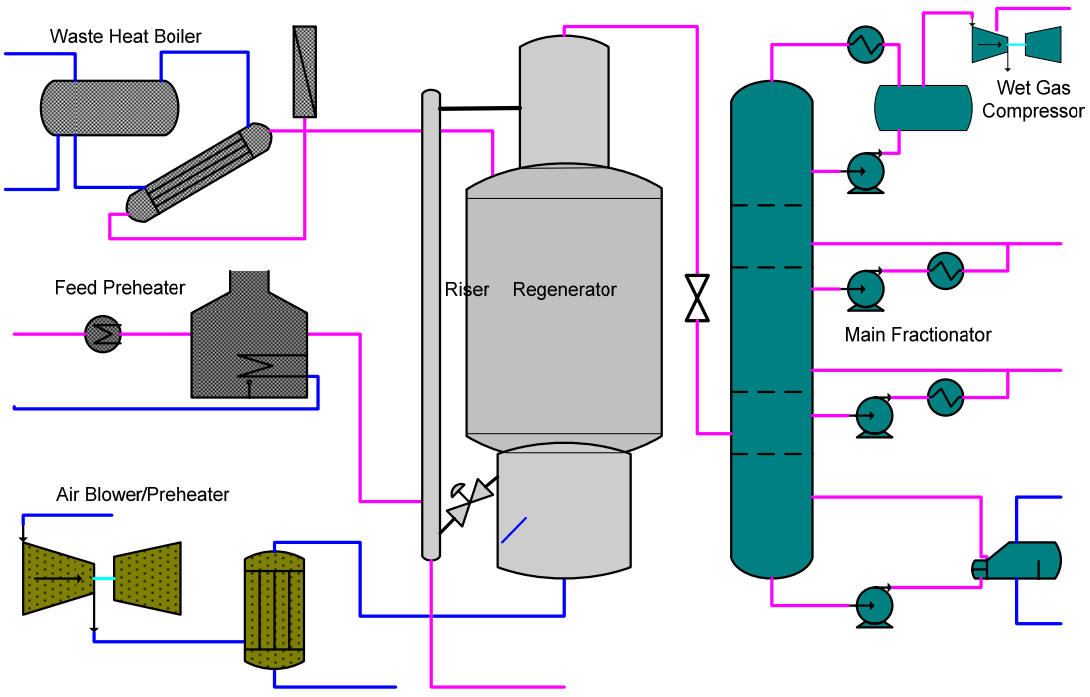


Figure 3-6: Schematic of ShadowPlant FCCU

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. 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. In Figure 3-7 and Figure 3-8, waste heat boiler and Feed preheater normal operation snapshot are shown.

Details of the unit and the startup transition are reported by Honeywell (2000), Sundarraman and Srinivasan (2003) and Srinivasan et al. (2004).

Data Generation:

Startup is the most difficult part of FCCU operation. There are seven main stages during the startup of ShadowPlant: *(1), Previous air blower startup, (2), Air blower startup, (3), Regenerator warm-up and catalyst loading, (4), Main fractionator startup, (5), Connecting reactor and main fractionator, Catalyst circulation, (6), Introduction of fresh feed, (7), Wet gas compressor startup, Increasing feed to design.*

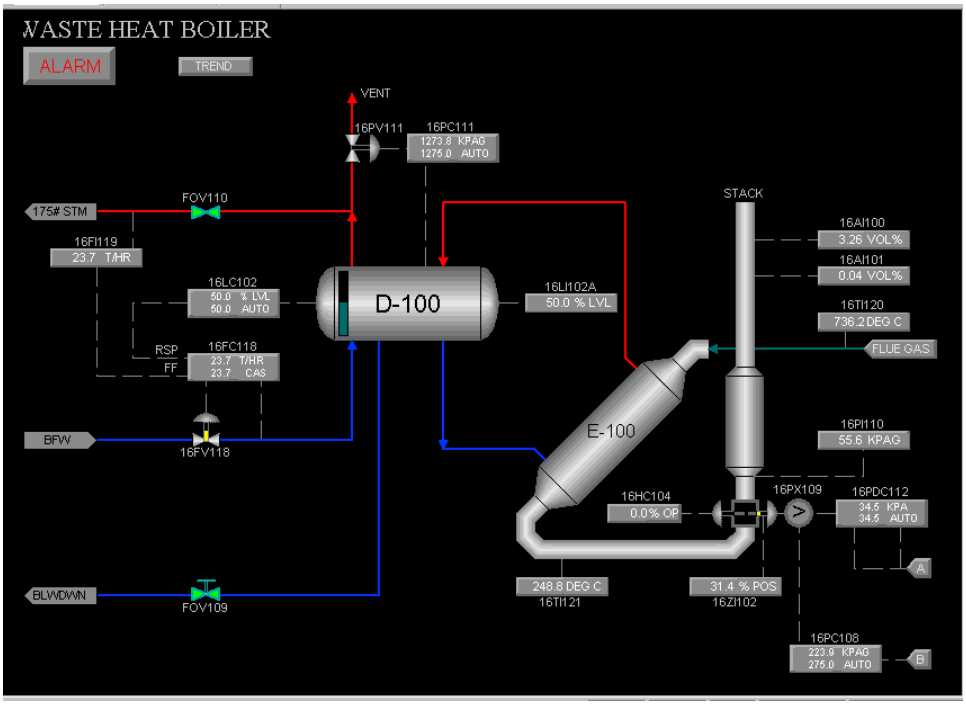


Figure 3-7: Waste heat boiler section of Shadow Plant (Normal operation)

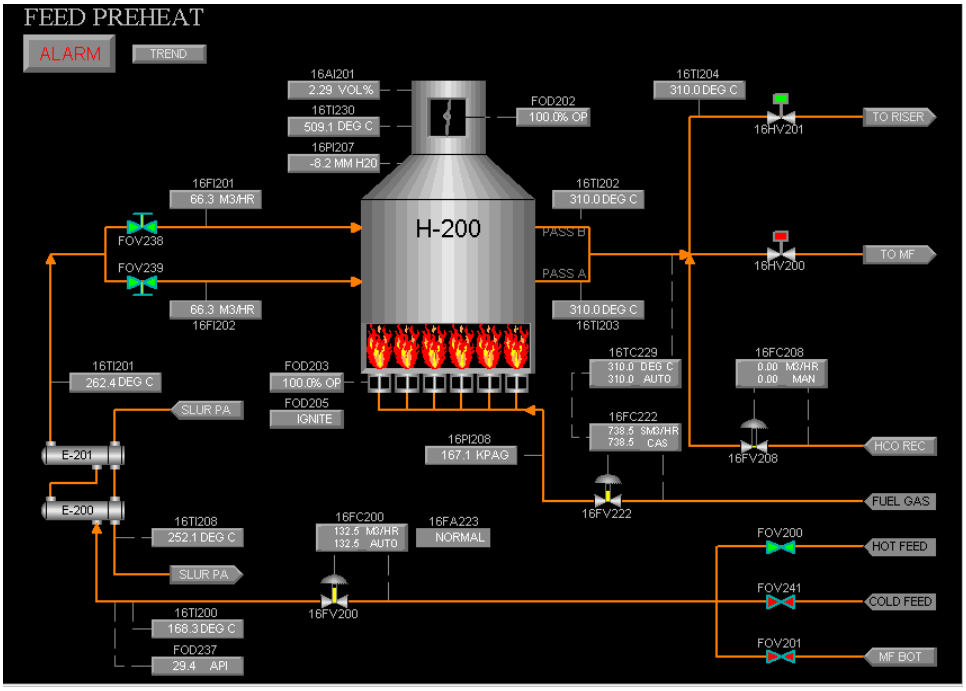


Figure 3-8: Feed preheater section of Shadow Plant (Normal operation)

The startup operation described above takes 40 to 60 hours. Several runs of the startup were performed following the standard operating procedure and data collected at 10 second intervals. 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. Several runs data (G_1 G_2 … G_8) are generated by startup ShadowPlant in this thesis. While the procedure for starting up the FCCU was the same in all the runs, operation differences in the magnitudes and duration were introduced between the runs. The average duration of the startup transition in these runs was 54 hours. Random noise was also added to the measured variables to simulate measurement noise in the real process. The entire process has 335 measured variables.

3.4.2 Tennessee Eastman Process

The Tennessee Eastman (TE) plant (Downs and Vogel, 1993) is a popular test bed for process systems applications such as plant-wide control, optimization, predictive control, faults diagnosis and signal comparison. The TE process produces two products (*G* and *H*) and a byproduct (*F*) from reactants *A, C, D,* and *E*. The reactions are given below, all of which are irreversible and exothermic and the reaction rates are a function of temperature through an Arrhenius expression.

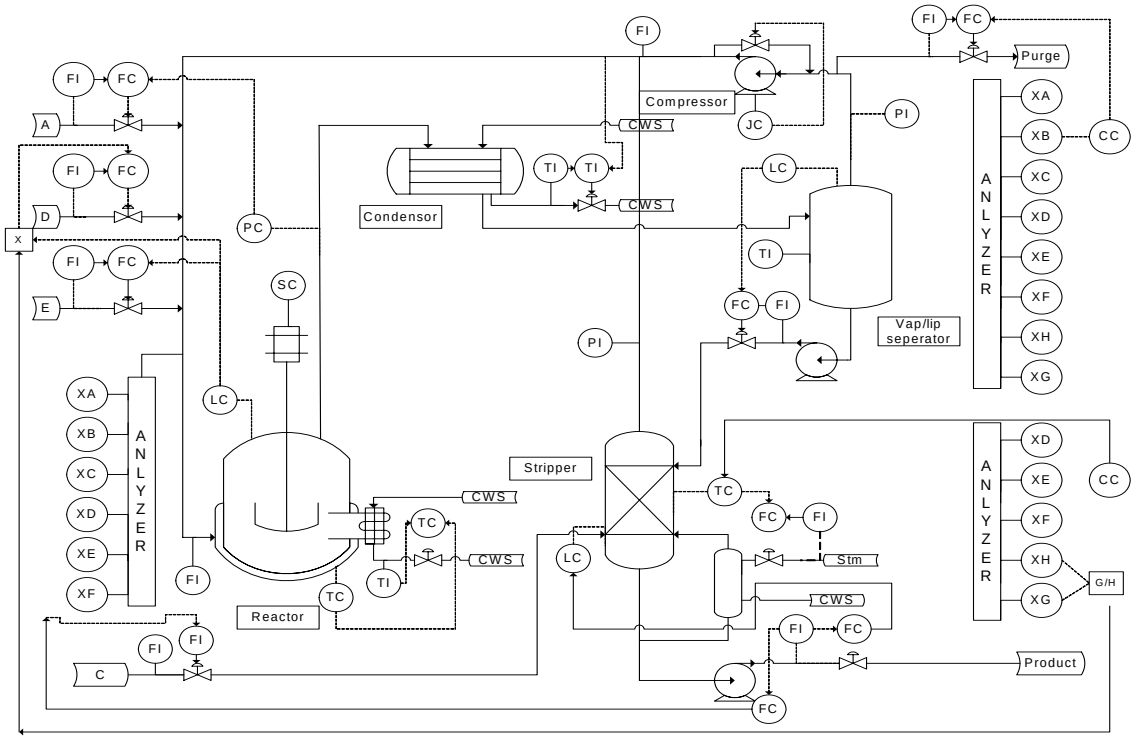


Figure 3-9: Flowsheet of Tennessee Eastman challenge process

$$A(g) + C(g) + D(g) \rightarrow G(liq)$$

$A(g) + C(g) + E(g) \rightarrow H(liq)$

$A(g) + E(g) \rightarrow F(liq)$

$3D(g) \rightarrow 2F(liq)$

The plant flowsheet is shown in Figure 3-9. The process has five unit operations: 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 process manipulated variables. The base case values for the process measurements and the manipulated variables are shown in Table 3-3 and 3-4 respectively. The closed-loop process Matlab simulator developed by Singhal (2001) based on the base control structure of McAvoy and Ye (1994) is used in this work. During the simulation, variable values are recorded every minute.

Table 3-3: Process measurements and their base value

Variable name	Variable number	Base case value	Units
A feed (stream 1)	XMEAS (1)	0.25052	kscmh
D feed (stream 2)	XMEAS (2)	3664.0	kgh^{-1}
E feed (stream 3)	XMEAS (3)	4509.3	kgh^{-1}
A and C feed (stream 4)	XMEAS (4)	9.3477	kscmh
Recycle flow (stream 8)	XMEAS (5)	26.902	kscmh
Reactor feed rate (stream 6)	XMEAS (6)	42.339	kscmh
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^3h^{-1}
Stripper level	XMEAS (15)	50.000	%
Stripper pressure	XMEAS (16)	3102.2	kPa gauge
Stripper underflow (Stream 11)	XMEAS (17)	22.949	m^3h^{-1}
Stripper temperature	XMEAS (18)	65.731	°C
Stripper steam flow	XMEAS (19)	230.31	kgh^{-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 3-4: Process manipulated variables

Variable name	Variable number	Base case value	Units
D feed flow (stream 2)	XMV (1)	63.053	kgh^{-1}
E feed flow (stream 3)	XMV (2)	53.98	kgh^{-1}
A feed flow (stream 1)	XMV (3)	24.644	kscmh
A and C feed flow (stream 4)	XMV (4)	61.302	kscmh
Compressor recycle valve	XMV (5)	22.21	%
Purge valve (stream 9)	XMV (6)	40.064	%
Separator pot liquid flow (stream 10)	XMV (7)	38.1	m^3h^{-1}
Stripper liquid product flow (stream 11)	XMV (8)	46.534	m^3h^{-1}
Stripper steam valve	XMV (9)	47.446	%
Reactor cooling water flow	XMV (10)	41.106	m^3h^{-1}
Condenser cooling water flow	XMV (11)	18.114	m^3h^{-1}
Agitator speed	XMV (12)	50	rpm

Process Disturbances Data Generation

In this thesis, data from TE plant is used for evaluating methods for signal comparison and process state identification. One application of signal comparison is in differentiating between process disturbances and between process operating states for offline analysis or online process monitoring, disturbance rejection, and other supervisory applications. The 22 PV values are usually used for such applications, as is the case here.

Next, consider intra-class differentiation that is, identifying a disturbance accurately, even if its magnitude or duration is different in different instances. This is a greater challenge for a classifier since there will be small or large differences among the time-series especially for nonlinear and noisy processes such as the TE plant. In order to generate such varying magnitude cases, we have introduced five new servo control type disturbances.

The five new disturbances XD1 to XD5 affect the A feed flowrate, reactor pressure, reactor level, reactor temperature, and compressor work. These are representative of set point changes or servo control problems in real industrial process operations. To generate different instances of the same disturbance, different start times, duration, and magnitude were used. For each disturbance, three different

parameter settings were used to make the signal different in both magnitude and duration. The disturbance XD1 is described in detail below.

XD1: Change in 'A' feed flowrate

During XD1-A, the flowrate of A feed from upstream is increased from the base case value of 0.25052 kscmh to 0.3902 kscmh in three steps, at t=180 min, t=190 min, t=200 min, as shown in Figure 3-10. In the simulation, this is brought about by quickly opening the control valve XMV(3). To recover from this disturbance, the pressure controller brings back the flow to the original base case value. The inverse disturbance of quickly decreasing the A feed flow is introduced at t=780 min. The downstream pressures (XMEAS(13) and XMEAS (16)) are also affected by these changes. . The effect on the A flow rate (XMEAS(1)) and the downstream pressures (XMEAS(13) and XMEAS (16)) is shown in Figure 3-10. Two other instances XD1-B and XD1-C with changes of magnitude 55% and 50% were also introduced.

For each disturbance XD1 to XD5, three different instances were generated by varying the time of set point change, the durations, and the magnitude. The detail of the set point changes in the different runs of XD1, XD2, XD3, XD4, and XD5 are shown in Tables 3-5 to 3-9 respectively. Figures 3-10 show the corresponding variable evolutions of XD1. Thus, the set of time series for each disturbance are different in both synchronization and magnitude.

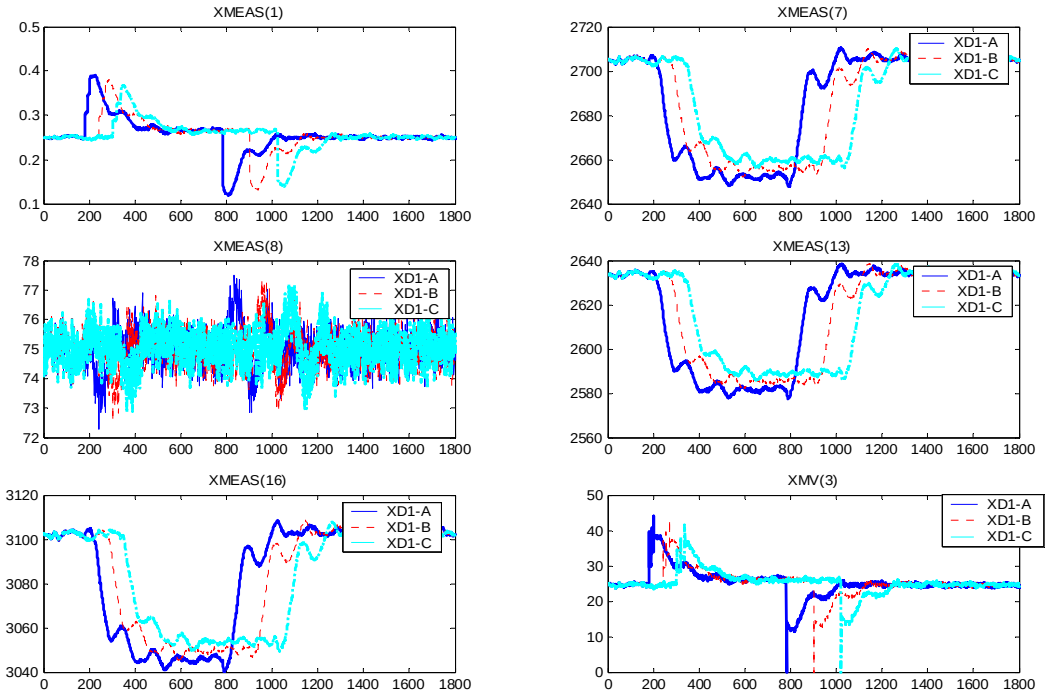


Figure 3-10: Three runs of XD1 disturbance with different magnitudes and duration

A simple signal-difference metric (ε_1) is first used to illustrate the complexity of the signal comparison. ε_1 is the average magnitude difference between the corresponding normalized signals of two multivariate time series.

$$\varepsilon(p,q) = \frac{1}{VT}\sum_{i=1}^{V}\sum_{t=1}^{T}\left|XMEAS_N^p(t,i) - XMEAS_N^q(t,i)\right| \tag{3-6}$$

Here $XMEAS_N^p(t,i)$ is the normalized measurement signal for variable *i* during disturbance *P*.

The 15 runs (XD1A to XD5C) are similar to each other and distinguishing among the classes is difficult. The ε_1 difference among the 15 runs is shown in Table 3-10. As can be seen, the simple ε_1 comparison (as well as other comparisons such as Dynamic time warping) cannot successfully classify the disturbances since the inter-class and intra-class distances are comparable. In seven of the cases, the difference between the instances of the same disturbance is larger than that with at least one of the other

disturbance. For example, the difference between XD3A and XD3B is 0.01991 while the differences between XD3A and XD5A, XD5B, XD5C are all less than 0.01719. The intra-& inter-group differences are shown in Table 3-11.

Table 3-5: Disturbance profile for XD1

	Target	Time (min)	Target	Time (min)	Target	Time (min)	Target	Time (min)
XD1-A	1.20*Base value	180	1.40*Base value	190	1.60*Base value	200	1.0*Base value	780
XD1-B	1.15*Base value	240	1.35*Base value	254	1.55*Base value	268	1.0*Base value	900
XD1-C	1.10*Base value	300	1.30*Base value	318	1.50*Base value	336	1.0*Base value	1020

Table 3-6: Disturbance profile for XD2

	Target	Time (min)	Target	Time (min)	Target	Time (min)	Target	Time (min)
XD2-A	1.03*Base value	180	1.05*Base value	190	1.07*Base value	200	1.0*Base value	1020
XD2-B	1.025*Base value	240	1.045*Base value	254	1.065*Base value	268	1.0*Base value	1080
XD2-C	1.02*Base value	300	1.04*Base value	318	1.06*Base value	336	1.0*Base value	1200

Table 3-7: Disturbance profile for XD3

	Target	Time (min)	Target	Time (min)	Target	Time (min)	Target	Time (min)
XD3-A	1.05*Base value	180	1.10*Base value	190	1.15*Base value	200	1.0*Base value	780
XD3-B	1.045*Base value	240	1.09*Base value	254	1.135*Base value	268	1.0*Base value	900
XD3-C	1.04*Base value	300	1.08*Base value	318	1.12*Base value	336	1.0*Base value	1020

Table 3-8: Disturbance profile for XD3

	Target	Time (min)	Target	Time (min)	Target	Time (min)	Target	Time (min)
XD3-A	1.05*Base value	180	1.10*Base value	190	1.15*Base value	200	1.0*Base value	780
XD3-B	1.045*Base value	240	1.09*Base value	254	1.135*Base value	268	1.0*Base value	900
XD3-C	1.04*Base value	300	1.08*Base value	318	1.12*Base value	336	1.0*Base value	1020

Table 3-9: Disturbance profile for XD5

	Target	Time (min)	Target	Time (min)	Target	Time (min)	Target	Time (min)
XD5-A	0.95*Base value	180	0.90*Base value	190	0.85*Base value	200	1.0*Base value	780
XD5-B	0.955*Base value	240	0.91*Base value	254	0.865*Base value	268	1.0*Base value	900
XD5-C	0.96*Base value	300	0.92*Base value	318	0.88*Base value	336	1.0*Base value	1020

Table 3-10: ε_1 difference between the fifteen disturbances ($x10^{-1}$)

	XD1-A	XD1-B	XD1-C	XD2-A	XD2-B	XD2-C	XD3-A	XD3-B	XD3-C	XD4-A	XD4-B	XD4-C	XD5-A	XD5-B	XD5-C
XD1-A	0	0.0686	0.074	0.2228	0.2052	0.1983	0.1735	0.1647	0.161	0.3631	0.3589	0.3443	0.0977	0.1108	0.1054
XD1-B		0	0.0629	0.2235	0.2028	0.1953	0.1735	0.1564	0.1476	0.3739	0.3483	0.3387	0.1136	0.0903	0.1023
XD1-C			0	0.2241	0.2009	0.1885	0.1847	0.1587	0.1383	0.3694	0.357	0.3278	0.1114	0.1018	0.0824
XD2-A				0	0.1446	0.166	0.2327	0.2309	0.1892	0.4558	0.4544	0.4107	0.2004	0.2015	0.1739
XD2-B					0	0.11	0.2631	0.2342	0.2413	0.4555	0.4219	0.4247	0.1914	0.1791	0.1801
XD2-C						0	0.2465	0.2424	0.2157	0.4556	0.4351	0.4001	0.1875	0.1786	0.1624
XD3-A							0	0.1991	0.1816	0.4143	0.4318	0.4113	0.1591	0.1718	0.1626
XD3-B								0	0.1736	0.4482	0.3953	0.4007	0.176	0.1415	0.1565
XD3-C									0	0.4294	0.4175	0.3697	0.1681	0.1521	0.124
XD4-A										0	0.2139	0.2909	0.3386	0.3605	0.3812
XD4-B											0	0.2003	0.3616	0.3262	0.3425
XD4-C												0	0.3589	0.337	0.3063
XD5-A													0	0.1005	0.0868
XD5-B														0	0.0864
XD5-C															0

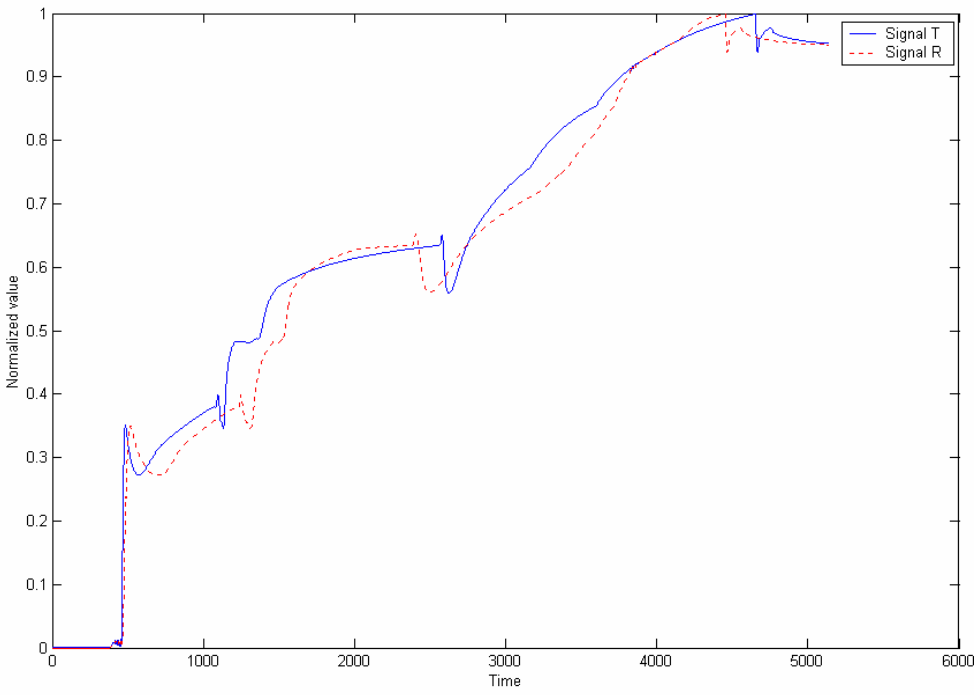
Table 3-11: Mean difference, max difference and Standard deviation of ε_1 difference between XD1 to XD5

	Mean difference					Max difference					Standard deviation of difference				
Difference	XD1	XD2	XD3	XD4	XD5	XD1	XD2	XD3	XD4	XD5	XD1	XD2	XD3	XD4	XD5
XD1	**0.0457**	0.2068	0.162	0.3535	0.1017	**0.074**	0.2241	0.1847	0.3739	0.1136	**0.0345**	0.0134	0.0141	0.015	0.0103
XD2		**0.0934**	0.2329	0.4349	0.1839	0.2241	**0.166**	0.2631	0.4558	0.2015		**0.0729**	0.0208	0.0216	0.0126
XD3			**0.1232**	0.4131	0.1569	0.1847	0.2631	0.1991	0.4482	0.176			**0.0929**	0.023	0.0162
XD4				**0.1567**	0.3459	0.3739	0.4558	0.4482	**0.2909**	0.3812				**0.1225**	0.0223
XD5					**0.0608**	0.1136	0.2015	**0.176**	0.3812	**0.1005**					**0.046**

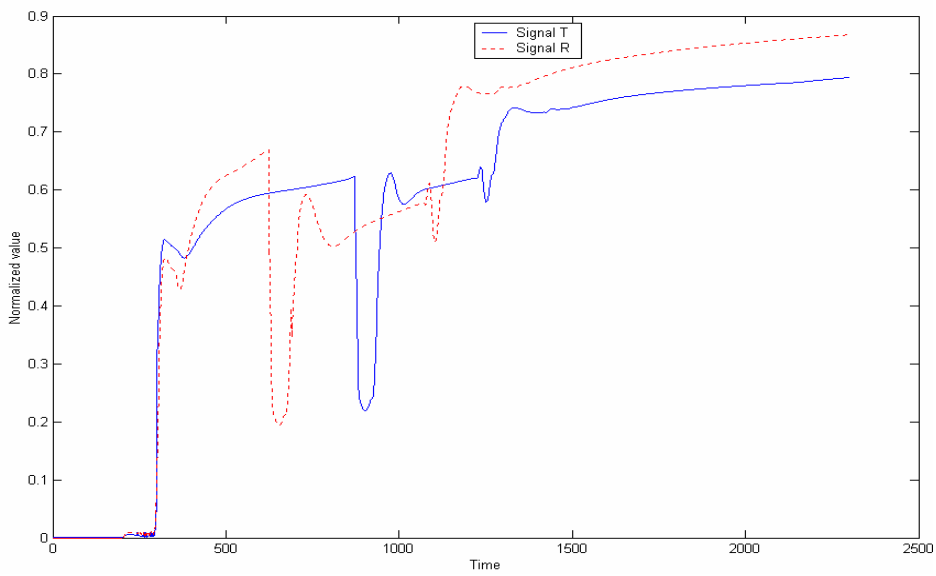
3.5 Case Studies

3.5.1 Typical signal Difference between Singular points Augmented DTW, DTW and direct comparison

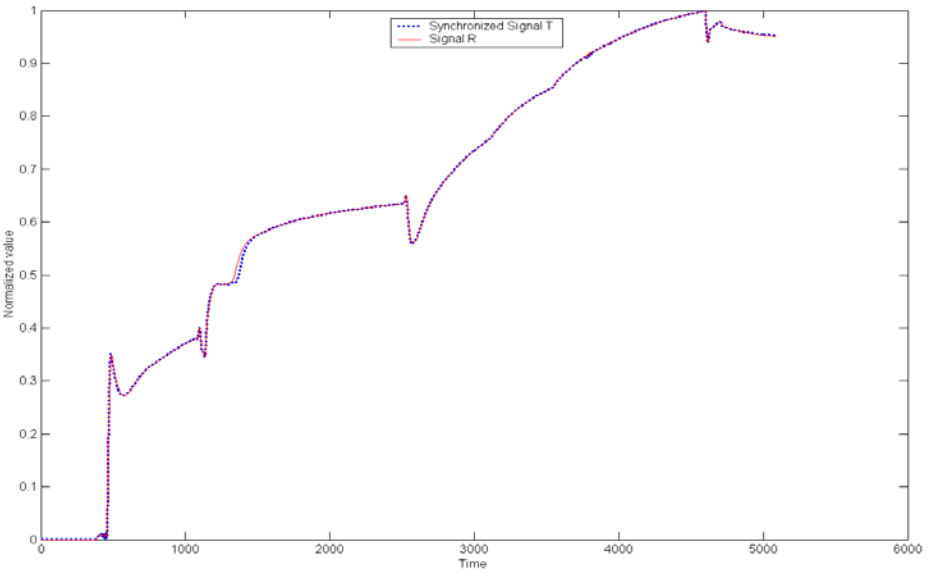
In this section, we show two examples to illustrate the efficacy of the presented approach. The difference between two signals is calculated by different methods – direct comparison, using DTW, and using singular points augmented DTW and XTW. DTW with Itakura (1975) local constraint is used in this thesis. If the two signals are exactly the same, it is obvious that there is no difference among all the methods. Figure 3-11(a) shows two signals different only in synchronization. The signals which are shown in Figure 3-11(b) are different in both synchronization and magnitude. The results for the two examples are shown in Table 3-12. The time cost of singular points augmented methods include the time for singular points detection.



(a)



(b)



(c)

Figure 3-11: Signals are different in (a), synchronization (b), both in synchronization and magnitude (c), Synchronized signal for (a) using XTW

It is clear that for cases where the only difference is in synchronization, XTW alone can work as well as the proposed method like XTW^{SP}, DTW_1, DTW_1^{SP}.

However, if two instances of a same type signal change differ in synchronization and magnitude, the proposed method using XTW^{SP} is more efficient compared with the other methods. In some cases, using XTW alone may lead to worse results than

direct comparison as illustrated in Table 3-12 where the difference from XTW alone is larger than that from direct comparison. XTW is working well for signal difference only in synchronization (See Figure 3-11(c)). For typical Itakura DTW such as DTW_1 and DTW_2, the results are better than Direct Comparison of the signal. DTW_1 always find the minimum difference between two signals since it does a full space search. Since DTW_1 is using full space search, the time cost will increase dramatically as the signal length increases. It is therefore difficult to use for complex systems with lot of signals. XTW^{SP} provides a balance between high accuracy and low computational cost.

Table 3-12: Comparison results from different methods

	Figure 3-11 (a)		Figure 3-11 (b)	
	Difference	Time Cost (min)	Difference	Time Cost (min)
Direct Comparison	0.0269	0.0007	0.0746	0.0007
XTW	8.52E-04	0.0059	0.1384	0.0033
DTW_2	0.0181	1.6543	0.0635	0.1234
DTW_1	1.78E-04	17.5381	0.0338	2.6155
XTW^{SP}	3.57E-04	0.0255	0.0360	0.0157
DTW_2^{SP}	0.0074	1.7437	0.0472	0.3383
DTW_1^{SP}	1.79E-04	3.3845	0.0355	0.7147

3.5.2 Case Study 1: Identifying Process States during Multi-mode Operation

Singular points augmented time warping can be used for identifying the state of multi-mode process operations such as a fluidized catalytic cracking.

Several runs of the startup of ShadowPlant were performed following the standard operating procedure and data for all 335 measured variables collected at 10 second intervals. Random noise was also added to the measured variables to simulate measurement noise in the real process. Two runs G_5 and G_6 are considered in detail

here. The duration of the startup transition in these runs was 23 and 25 hours, respectively. In G_6, operations such as air pre-heating and catalyst loading into the regenerator is initiated earlier; also the amount of catalyst loaded is larger. These changes lead to numerous differences between the signals from the two runs.

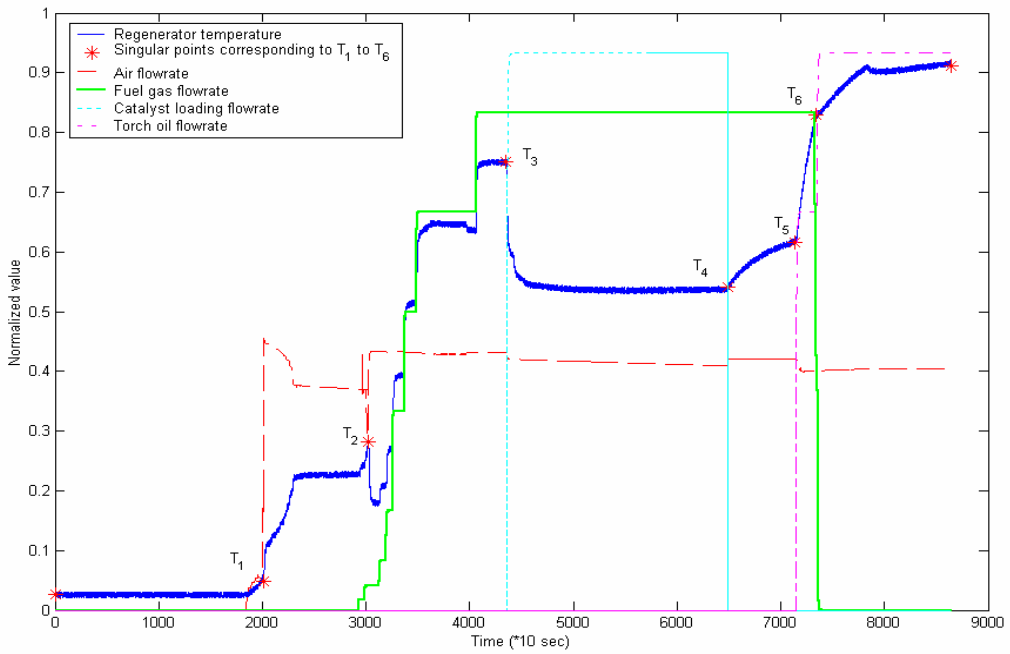


Figure 3-12: Variable profiles during the different stages of regenerator startup of G_5

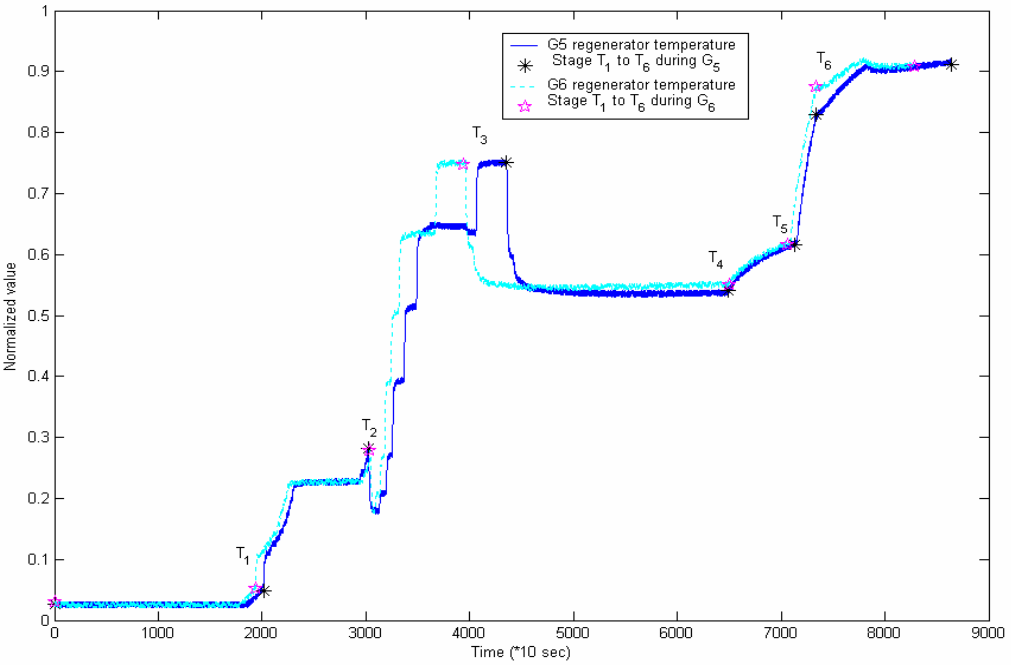


Figure 3-13: Singular points in regenerator temperature during different stages of regenerator startup

Consider the startup of the regenerator section. As shown in Table 3-13 and Figure 3-12, from the process operation and control points-of-view, there are six important stages during the startup of the regenerator. Given these stages of operation in G_5, the objective is to find the corresponding stages in G_6 by signal comparison alone without any other prior knowledge. As shown in Figure 3-13, the signals of many variables vary in both magnitude and duration between G_5 and G_6. We therefore match the signals using the singular points augmented time warping.

Table 3-13: Important process stages during startup of regenerator section of ShadowPlant

Stage	Operational description
T_1	Activate air flow to regenerator
T_2	Adjust system pressure and air flow
T_3	Load catalyst into regenerator
T_4	Stop catalyst loading
T_5	Introduce torch oil to regenerator
T_6	Turn off fuel gas for air preheater

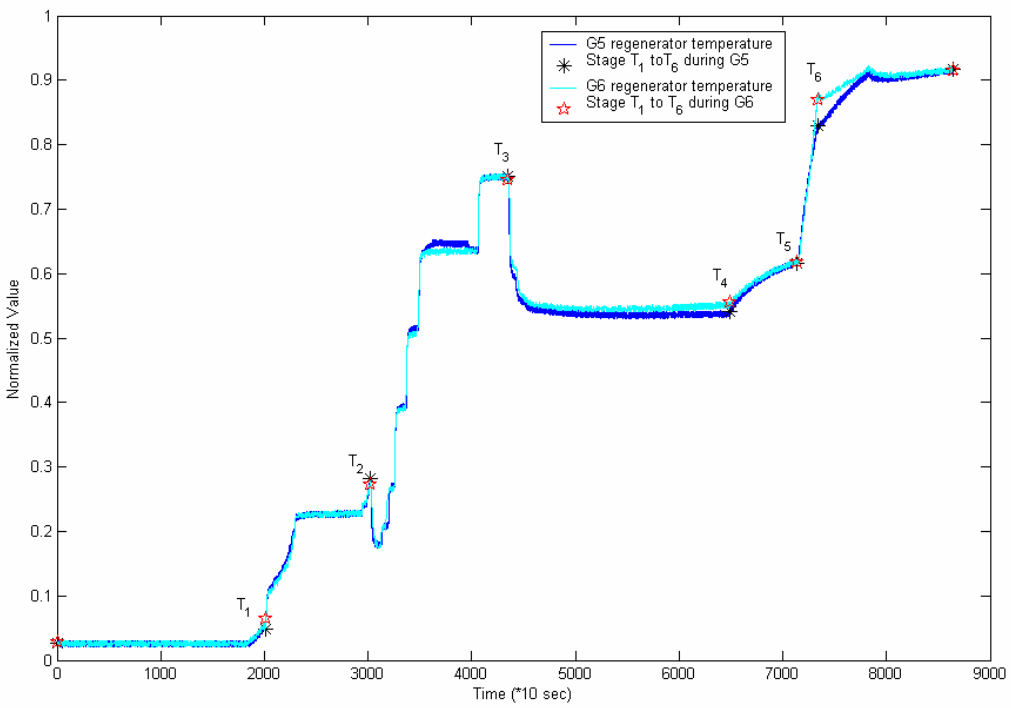


Figure 3-14: XTW^{SP} warped G_6 regenerator temperature plotted with G_5 regenerator temperature

Any of the variants of time warping described above can be used for this purpose. Here, we report results from DTW_1 and singular points augmented time warping XTW^{SP}. There is a difference of 360 samples between the two runs for this part of the startup. As an illustration of the run-to-run difference, the regenerator temperature profile and the singular points in the two runs are shown in Figure 3-13. Despite these significant differences, as shown in Table 3-14, the corresponding points of the stages are correctly identified by XTW^{SP}. A plot of the warped profile of G_6 regenerator temperature when plotted with G_5 regenerator temperature profile reveals the operational correctness of the result (See Figure 3-14).

Table 3-14: Corresponding singular points identified by signal comparison

	Start	T_1	T_2	T_3	T_4	T_5	T_6	End
G_5	1	2015	3024	4349	6492	7131	7338	8640
G_6 (Actual)	1	1937	3041	3954	6510	7074	7348	8280
G_6 (XTW^{SP})	1	1937	3041	3954	6511	7074	7339	8280
G_6 (DTW_1)	1	1937	3041	3954	6401	7067	7247	8280

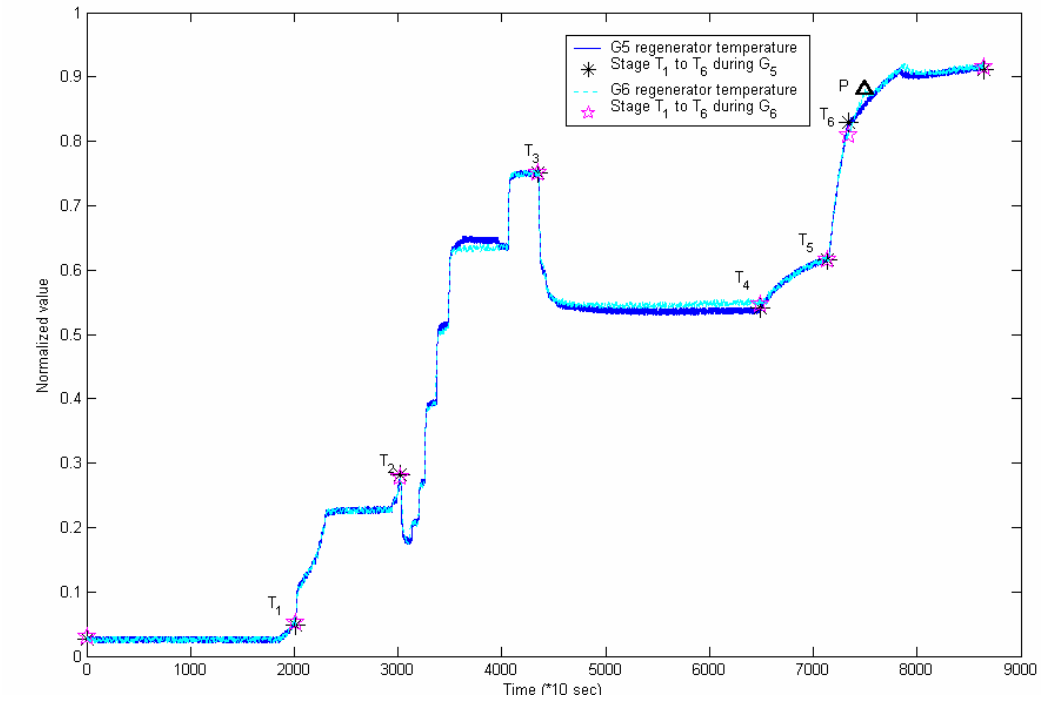


Figure 3-15: Synchronizing signals from G_5 and G_6 using DTW_1

In contrast, when DTW_1 is used to match the two signals, there are operational inconsistencies among the profiles as shown in Figure 3-15. An obvious mismatch occurs at the start of Stage 6 (at time *7348*) in run G_6 as highlighted in Figure 3-16. While the actual start of Stage 6 occurs at Point P, DTW_1 identifies a point 101 samples earlier. This is because DTW_1 finds the minimum distance between the two signals but in this case, because of the significant magnitude and duration differences, the mathematical minimum does not coincide with the corresponding stages of operation. The singular points augmented time warping is not susceptible to this failure. The benefit of the proposed approach is also clear when a computational time comparison is made – 1248.60s for DTW_1 and 9.94s for XTW^{SP} – an improvement by a factor of 125. The time cost of singular points augmented method is included the time for singular points detection. Other variables such as air flowrate into regenerator, catalyst level in regenerator, and regenerator pressure were also similarly studied. It was found that the corresponding stages of G_5 and G_6 were correctly identified in all cases.

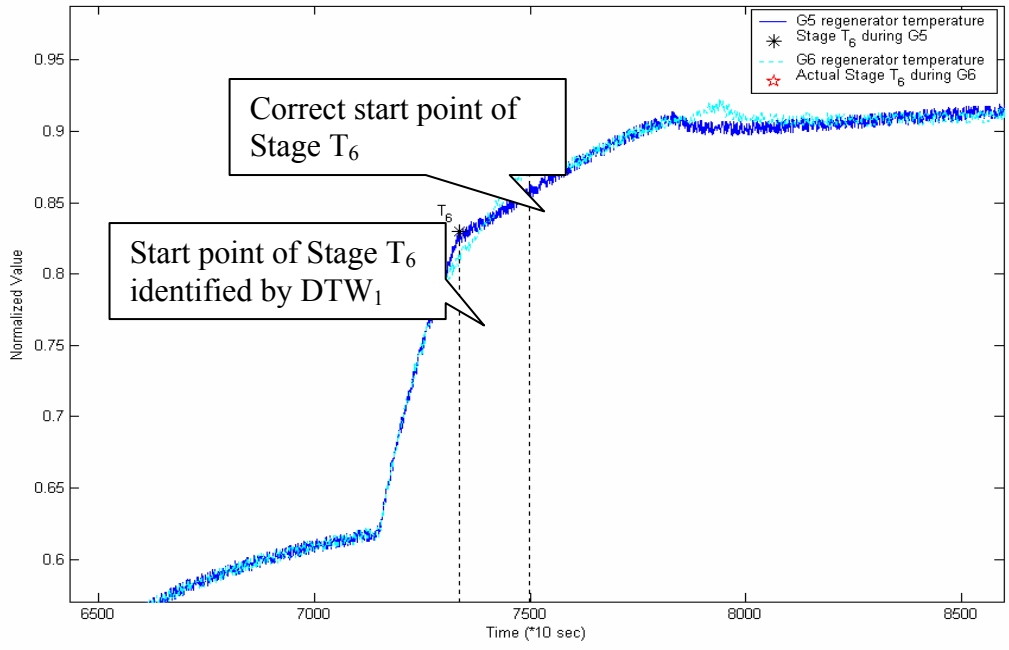


Figure 3-16: Misidentification of Stage T_6 in Case Study 1 by DTW_1

The important stages of other variables such as airflow rate, catalyst levels in regenerator, regenerator pressure were also studied with this method. By comparing the signal from G_5 and G_6, the corresponding stage in G_6 could be identified successfully with corresponding singular points. Key stages of air blower are: T_1^A, start air blower controller, T_2^A, air blower speed reached 100%, T_3^A, increase down stream pressure, T_4^A, down stream pressure stable, T_5^A, airflow affected by catalyst loading, T_6^A, airflow affected by regenerator pressure switch. Key stages of regenerator are: T_1^R, increase regenerator pressure, T_2^R, regenerator pressure stable, T_3^R, regenerator vent open, T_4^R, load catalyst to regenerator , T_5^R, catalyst level increasing from zero, T_6^R, stop catalyst loading, T_7^R, regenerator vent closed, T_8^R, regenerator pressure switch control, T_9^R, transfer catalyst to stripper, T_{10}^R, transfer catalyst finished. As shown in Table 3-15, 3-16 and Table 3-17, all the key stages of air blower (through air flowrate) and regenerator (through catalyst level, and regenerator pressure) are identified. The average difference from actual stage is only 3.2 points.

Table 3-15: Different stages of air blower identified by comparison of airflow

	Start	T_1^A	T_2^A	T_3^A	T_4^A	T_5^A	T_6^A
G_5	1	1849	2015	3008	3016	4356	7193
G_6 (Actual)	1	1771	1937	3028	3037	3958	7113
G_6 (XTW^{SP})	1	1771	1937	3025	3037	3960	7121

Table 3-16: Different stages of regenerator identified by comparison of catalyst level

	T_5^R	T_6^R	T_9^R	T_{10}^R
G_5	5026	6447	7252	7828
G_6 (Actual)	4746	6490	7221	7795
G_6 (XTW^{SP})	4738	6485	7225	7793

Table 3-17: Different stages of regenerator identified by comparison of pressure

	T_1^R	T_2^R	T_3^R	T_4^R	T_7^R	T_8^R
G_5	1992	3008	3046	4354	6483	7178
G_6 (Actual)	1926	3029	3062	3960	6493	7109
G_6 (XTW^{SP})	1914	3029	3059	3961	6496	7105

3.5.3 Case Study 2: Clustering of Process States in the Tennessee Eastman Process

Another application of singular points augmented time warping is for disturbance classification and fault diagnosis. The Tennessee Eastman (TE) plant is a popular test bed for process control, fault diagnosis and signal comparison. 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 (2001) based on the base control structure of McAvoy and Ye (1994). During the simulation, variable values are recorded every minute.

To generate signals with magnitude and duration differences, we introduced five set point changes in the process. These five new disturbance classes XD1 – XD5 are described in Section 3.4.2. The objective is to cluster the disturbances by comparing the signals. Only the 22 process measurements variables are used for signal comparison. Different methods such as DTW_1, DTW_2, and their singular points augmented versions, DTW_1^{SP} and DTW_2^{SP} as well as XTW^{SP} are compared based on the average variable-wise distance. For DTW_2 and DTW_2^{SP} the band global search width was set to *B=50,* however any value can be used. The average signal difference between the 15 groups is calculated using the various signal comparison methods and shown in Tables 3-18, 3-19, and 3-20. The table is not symmetric since the asymmetric DTW is used for this thesis. The ability of different method in differentiating between

the disturbances is shown in Table 3-18. Since the maximum intra-cluster distance (0.2008 between XD4-A and XD4-C) is greater than a large number of the inter-cluster distances, 12 out of the 20 different clusters cannot be separated by DTW_2. DTW_2^{SP} improves the separability and only four of these cannot be differentiated. DTW_1 and DTW_1^{SP} seek global optima and can differentiate among all the clusters; however this is attained at a large computational expense – a 33-fold increase in computational time. Singular points augmentation proves beneficial in this case, since DTW_1^{SP} has a 28% processing speed advantage over DTW_1. XTW^{SP} is better than DTW_2 and cannot differentiate between only one pair – XD1 and XD5. While, this performance is not as good as DTW_1, the advantage of XTW^{SP} comes to the fore when a time comparison is made. On an average, XTW^{SP} requires only 37 cpu seconds (on a Pentium 4, 2.4 GHz processor) to compare the 22 sets of signals from any two clusters in contrast to 2661 seconds for DTW_1

Table 3-18: Signal differences between process disturbances in TE process calculated using XTW^{SP} ($x10^{-1}$)

XTW^{SP}	**XD1-A**	**XD1-B**	**XD1-C**	**XD2-A**	**XD2-B**	**XD2-C**	**XD3-A**	**XD3-B**	**XD3-C**	**XD4-A**	**XD4-B**	**XD4-C**	**XD5-A**	**XD5-B**	**XD5-C**
XD1-A	0	0.0269	0.0382	0.1735	0.1730	0.1686	0.1294	0.1235	0.1156	0.3309	0.3012	0.2763	0.0772	0.0761	0.0760
XD1-B	0.0238	0	0.0214	0.1893	0.1799	0.1622	0.1375	0.1233	0.1095	0.3244	0.2978	0.2892	0.0789	0.0687	0.0698
XD1-C	0.0335	0.0249	0	0.1885	0.1794	0.1655	0.1264	0.1220	0.1070	0.3384	0.3113	0.2883	0.0831	0.0713	0.0659
XD2-A	0.1861	0.1899	0.1891	0	0.0301	0.0549	0.1935	0.1724	0.1550	0.3837	0.3490	0.3246	0.1469	0.1489	0.1430
XD2-B	0.1735	0.1752	0.1749	0.0311	0	0.0346	0.1808	0.1631	0.1515	0.3636	0.3260	0.3134	0.1514	0.1336	0.1300
XD2-C	0.1640	0.1641	0.1652	0.0518	0.0318	0	0.1998	0.1701	0.1546	0.3703	0.3718	0.3327	0.1559	0.1443	0.1255
XD3-A	0.1282	0.1418	0.1474	0.1778	0.1860	0.2035	0	0.0296	0.0586	0.3274	0.3053	0.2980	0.1033	0.1144	0.1210
XD3-B	0.1242	0.1158	0.1308	0.1595	0.1603	0.1767	0.0289	0	0.0298	0.3040	0.3091	0.2861	0.0981	0.0980	0.1037
XD3-C	0.1161	0.1105	0.1070	0.1478	0.1502	0.1567	0.0529	0.0290	0	0.3323	0.3075	0.2970	0.0903	0.0889	0.0870
XD4-A	0.3249	0.3337	0.3399	0.3638	0.3709	0.3916	0.3450	0.3301	0.3278	0	0.0389	0.0765	0.2886	0.3039	0.3062
XD4-B	0.3220	0.3229	0.3279	0.3471	0.3544	0.3598	0.3386	0.3205	0.3126	0.0437	0	0.0393	0.2984	0.2928	0.3071
XD4-C	0.2989	0.3045	0.3077	0.3257	0.3281	0.3339	0.3365	0.3109	0.2955	0.0854	0.0396	0	0.2861	0.2736	0.2816
XD5-A	0.0812	0.0824	0.0835	0.1455	0.1518	0.1499	0.1065	0.0975	0.0900	0.2885	0.2598	0.2541	0	0.0285	0.0440
XD5-B	0.0755	0.0750	0.0754	0.1397	0.1350	0.1410	0.1142	0.0992	0.0915	0.2868	0.2684	0.2478	0.0250	0	0.0243
XD5-C	0.0706	0.0715	0.0690	0.1395	0.1332	0.1296	0.1208	0.1010	0.0885	0.3178	0.2902	0.2660	0.0427	0.0255	0

Table 3-19: Signal differences between process disturbances in TE process calculated using DTW_1 and DTW_1^{SP} ($x10^{-1}$)

DTW_1^{SP}	XD1-A	XD1-B	XD1-C	XD2-A	XD2-B	XD2-C	XD3-A	XD3-B	XD3-C	XD4-A	XD4-B	XD4-C	XD5-A	XD5-B	XD5-C
XD1-A	0	0.0123	0.0198	0.1478	0.1429	0.1367	0.1014	0.0971	0.0937	0.2903	0.2759	0.2538	0.0564	0.0576	0.0588
XD1-B	0.0121	0	0.0108	0.1516	0.1376	0.1339	0.1101	0.0955	0.0884	0.2926	0.2798	0.2569	0.0613	0.0543	0.0546
XD1-C	0.0190	0.0124	0	0.1519	0.1395	0.1339	0.1041	0.1011	0.0871	0.2953	0.2925	0.2644	0.066	0.0564	0.0501
XD2-A	0.1786	0.1789	0.1820	0	0.0202	0.0363	0.1557	0.1366	0.1223	0.3200	0.3013	0.2644	0.1356	0.1354	0.1264
XD2-B	0.1643	0.1635	0.1553	0.0210	0	0.0259	0.1479	0.1357	0.1213	0.307	0.2698	0.2767	0.1435	0.121	0.1146
XD2-C	0.1516	0.1494	0.1507	0.0323	0.0204	0	0.1724	0.147	0.1232	0.3138	0.3188	0.2699	0.1401	0.1378	0.1198
XD3-A	0.1237	0.1332	0.1387	0.1736	0.1761	0.1964	0	0.0176	0.0293	0.3031	0.2837	0.2844	0.0931	0.1046	0.1142
XD3-B	0.1151	0.1108	0.1224	0.1506	0.1531	0.1696	0.0194	0	0.0191	0.2896	0.2852	0.2738	0.0853	0.084	0.0937
XD3-C	0.1080	0.1030	0.1000	0.1378	0.1416	0.1475	0.0302	0.0170	0	0.2926	0.2744	0.2649	0.0816	0.0782	0.0733
XD4-A	0.3140	0.3258	0.3309	0.3510	0.3547	0.3709	0.2900	0.2936	0.3116	0	0.0272	0.0386	0.2714	0.2896	0.294
XD4-B	0.3141	0.3014	0.3108	0.3234	0.3375	0.3467	0.2947	0.2911	0.2823	0.0253	0	0.0251	0.2792	0.2746	0.2886
XD4-C	0.2868	0.2917	0.2905	0.3083	0.3149	0.3250	0.2638	0.268	0.2683	0.0366	0.0283	0	0.268	0.2614	0.267
XD5-A	0.0669	0.0723	0.0778	0.1262	0.1359	0.1295	0.0860	0.082	0.0726	0.2613	0.2488	0.2379	0	0.0142	0.0221
XD5-B	0.0633	0.0597	0.0651	0.1190	0.1176	0.1231	0.0889	0.0768	0.0666	0.2773	0.2546	0.2428	0.0151	0	0.0122
XD5-C	0.0639	0.0591	0.0557	0.1150	0.1080	0.1151	0.0976	0.0807	0.0706	0.2829	0.266	0.2473	0.0219	0.0137	0

DTW_1	XD1-A	XD1-B	XD1-C	XD2-A	XD2-B	XD2-C	XD3-A	XD3-B	XD3-C	XD4-A	XD4-B	XD4-C	XD5-A	XD5-B	XD5-C
XD1-A	0	0.0075	0.0125	0.1399	0.1269	0.1192	0.0940	0.0827	0.0700	0.2830	0.2720	0.2537	0.0416	0.0400	0.0391
XD1-B	0.0075	0	0.0073	0.1395	0.1260	0.1188	0.0975	0.0816	0.0698	0.2859	0.2706	0.2525	0.0436	0.0369	0.0369
XD1-C	0.0123	0.0072	0	0.1414	0.1275	0.1193	0.1033	0.0872	0.0712	0.2887	0.2741	0.2531	0.0467	0.0400	0.0346
XD2-A	0.1372	0.1371	0.1394	0	0.0150	0.0247	0.0935	0.0860	0.0810	0.2880	0.2758	0.2600	0.0893	0.0919	0.0927
XD2-B	0.1248	0.1234	0.1254	0.0149	0	0.0143	0.0941	0.0855	0.0785	0.2889	0.2756	0.2588	0.0825	0.0820	0.0837
XD2-C	0.1166	0.1160	0.1172	0.0247	0.0143	0	0.0956	0.0865	0.0777	0.2906	0.2773	0.2580	0.0780	0.0773	0.0771
XD3-A	0.0910	0.0945	0.0993	0.0915	0.0928	0.0938	0	0.0137	0.0238	0.2430	0.2355	0.2240	0.0675	0.0753	0.0813
XD3-B	0.0796	0.0793	0.0839	0.0855	0.0845	0.0855	0.0136	0	0.0137	0.2457	0.2334	0.2207	0.0590	0.0615	0.0675
XD3-C	0.0676	0.0673	0.0682	0.0803	0.0786	0.0777	0.0242	0.0135	0	0.2473	0.2344	0.2181	0.0502	0.0520	0.0534
XD4-A	0.2795	0.2820	0.2850	0.2859	0.2866	0.2876	0.2409	0.2444	0.2459	0	0.0202	0.0317	0.2553	0.2647	0.2714
XD4-B	0.2675	0.2665	0.2702	0.2736	0.2735	0.2745	0.2337	0.2325	0.2344	0.0202	0	0.0179	0.2429	0.2487	0.2561
XD4-C	0.2496	0.2486	0.2490	0.2577	0.2571	0.2559	0.2223	0.2207	0.2178	0.0324	0.0181	0	0.2277	0.2337	0.2367
XD5-A	0.0406	0.0428	0.0454	0.0902	0.0833	0.0775	0.0680	0.0586	0.0498	0.2583	0.2460	0.2308	0	0.0104	0.0162
XD5-B	0.0388	0.0361	0.0390	0.0933	0.0835	0.0778	0.0773	0.0619	0.0520	0.2685	0.2520	0.2362	0.0104	0	0.0089
XD5-C	0.0378	0.0355	0.0336	0.0945	0.0854	0.0782	0.0838	0.0691	0.0534	0.2750	0.2595	0.2394	0.0162	0.0088	0

Table 3-20: Signal differences between process disturbances in TE process calculated using DTW_2 and DTW_2^{SP} ($x10^{-1}$)

DTW_2^{SP}	XD1-A	XD1-B	XD1-C	XD2-A	XD2-B	XD2-C	XD3-A	XD3-B	XD3-C	XD4-A	XD4-B	XD4-C	XD5-A	XD5-B	XD5-C
XD1-A	0	0.0136	0.0258	0.1522	0.1436	0.1366	0.1152	0.1073	0.0945	0.2969	0.2862	0.2574	0.0564	0.0578	0.0587
XD1-B	0.0125	0	0.0112	0.1478	0.1337	0.1309	0.1106	0.0967	0.0877	0.2947	0.2816	0.2588	0.0628	0.0523	0.0548
XD1-C	0.0222	0.0125	0	0.1531	0.1394	0.1306	0.1093	0.1090	0.0871	0.2888	0.2992	0.2721	0.0674	0.0597	0.0509
XD2-A	0.1953	0.1853	0.1878	0	0.0216	0.0764	0.1593	0.1383	0.1239	0.4279	0.3276	0.2698	0.1432	0.1439	0.1368
XD2-B	0.1684	0.1727	0.1576	0.0310	0	0.0332	0.1535	0.1360	0.1221	0.3397	0.2933	0.2868	0.1559	0.1317	0.1181
XD2-C	0.1556	0.1588	0.1552	0.0628	0.0425	0	0.1761	0.1568	0.1264	0.3519	0.3349	0.2838	0.1419	0.1435	0.1312
XD3-A	0.1285	0.1367	0.1396	0.1928	0.1960	0.2026	0	0.0234	0.0604	0.3032	0.3032	0.3179	0.0954	0.1107	0.1199
XD3-B	0.1274	0.1141	0.1242	0.1661	0.1752	0.1778	0.0223	0	0.0251	0.3811	0.2894	0.2895	0.0878	0.0877	0.0971
XD3-C	0.1227	0.1141	0.1036	0.1409	0.1466	0.1541	0.0635	0.0201	0	0.3007	0.2729	0.2624	0.1015	0.0839	0.0756
XD4-A	0.3197	0.3343	0.3391	0.3798	0.3864	0.4163	0.2928	0.3127	0.3308	0	0.0278	0.0611	0.2917	0.3115	0.3268
XD4-B	0.3128	0.3011	0.3137	0.3678	0.3804	0.3871	0.2965	0.2931	0.3064	0.0411	0	0.0252	0.2851	0.2875	0.2977
XD4-C	0.3014	0.2956	0.2900	0.3300	0.3440	0.3562	0.2671	0.2765	0.2741	0.0988	0.0456	0	0.2824	0.2807	0.2776
XD5-A	0.0672	0.0744	0.0801	0.1497	0.1509	0.1430	0.0860	0.0831	0.0740	0.2748	0.2640	0.2576	0	0.0186	0.0392
XD5-B	0.0659	0.0611	0.0679	0.1333	0.1368	0.1340	0.0900	0.0810	0.0668	0.2876	0.2628	0.2596	0.0153	0	0.0125
XD5-C	0.0646	0.0592	0.0574	0.1222	0.1218	0.1234	0.0981	0.0831	0.0696	0.2925	0.2758	0.2532	0.0309	0.0143	0

DTW_2	XD1-A	XD1-B	XD1-C	XD2-A	XD2-B	XD2-C	XD3-A	XD3-B	XD3-C	XD4-A	XD4-B	XD4-C	XD5-A	XD5-B	XD5-C
XD1-A	0	0.0347	0.0460	0.1700	0.1580	0.1520	0.1045	0.1090	0.1100	0.2846	0.2805	0.2701	0.0608	0.0681	0.0700
XD1-B	0.0332	0	0.0321	0.1661	0.1583	0.1504	0.1045	0.0944	0.0972	0.2879	0.2749	0.2648	0.0670	0.0566	0.0636
XD1-C	0.0435	0.0312	0	0.1659	0.1564	0.1504	0.1148	0.0966	0.0845	0.2867	0.2782	0.2623	0.0707	0.0625	0.0537
XD2-A	0.1899	0.1871	0.1893	0	0.0459	0.0965	0.1768	0.1645	0.1419	0.3762	0.3565	0.3305	0.1570	0.1525	0.1370
XD2-B	0.1744	0.1748	0.1725	0.0452	0	0.0657	0.1818	0.1620	0.1558	0.3652	0.3447	0.3170	0.1464	0.1367	0.1282
XD2-C	0.1662	0.1653	0.1643	0.0901	0.0602	0	0.1836	0.1688	0.1518	0.3695	0.3541	0.3280	0.1487	0.1380	0.1285
XD3-A	0.1291	0.1324	0.1396	0.1900	0.1955	0.1981	0	0.0984	0.1354	0.3091	0.3302	0.3395	0.0945	0.1060	0.1192
XD3-B	0.1274	0.1158	0.1210	0.1671	0.1691	0.1813	0.0924	0	0.0858	0.3332	0.2967	0.3106	0.1070	0.0869	0.0964
XD3-C	0.1237	0.1125	0.1027	0.1408	0.1590	0.1557	0.1201	0.0800	0	0.3366	0.3115	0.2803	0.1177	0.0933	0.0768
XD4-A	0.3299	0.3346	0.3371	0.4041	0.3997	0.4050	0.3209	0.3600	0.3686	0	0.1262	0.2111	0.3019	0.3227	0.3407
XD4-B	0.3228	0.3163	0.3207	0.3810	0.3700	0.3849	0.3361	0.3096	0.3395	0.1155	0	0.1150	0.3120	0.2942	0.3087
XD4-C	0.3068	0.3034	0.2991	0.3514	0.3327	0.3510	0.3451	0.3163	0.2941	0.2008	0.1056	0	0.3176	0.2946	0.2790
XD5-A	0.0648	0.0724	0.0776	0.1474	0.1412	0.1416	0.0793	0.0959	0.1082	0.2682	0.2860	0.2881	0	0.0479	0.0596
XD5-B	0.0709	0.0598	0.0660	0.1391	0.1273	0.1320	0.0863	0.0738	0.0833	0.2805	0.2611	0.2696	0.0455	0	0.0415
XD5-C	0.0721	0.0659	0.0561	0.1238	0.1155	0.1192	0.0971	0.0785	0.0662	0.2965	0.2691	0.2482	0.0543	0.0403	0

The problem of identifying a run of the process is relatively simpler. Let S be the *group separability ratio* defined as the ratio between the minimum inter-group distance and the maximum intra-group distance. The larger the *S*, the easier it is to distinguish between the groups of operation. For each case, if *S* is larger than 1, the different operations can be differentiated clearly; else the operations cannot be uniquely identified by signal comparison. The *S* for the different signal comparison methods is shown in Table 3-21. The minimum *S* value for DTW_2 is 0.4889, and DTW_2 cannot differentiate between instances of XD3 and XD5. DTW_2^{SP} does not suffer from this problem and can clearly differentiate between all the groups (minimum *S* is 1.1060). With minimum *S* values of 2.0904 and 2.2670 respectively, both DTW_1 and DTW_1^{SP} can separate all the groups clearly as can XTW. Thus, the singular points augmented time warping methods are better than the traditional ones.

Table 3-21: Group separability ratio for TE process

	DTW_1	DTW_1^{SP}	DTW_2	XTW^{SP}	DTW_2^{SP}
XD1-A	3.0860	3.3316	1.4899	2.1055	2.9099
XD1-B	4.7001	4.7661	1.7248	2.6582	4.3529
XD1-C	2.6838	2.8131	1.2200	1.8040	2.2248
XD2-A	3.2515	3.5604	1.3739	2.6961	1.9459
XD2-B	5.2574	5.2941	1.9183	4.1853	2.8659
XD2-C	3.1409	3.1708	1.2346	2.3602	1.6152
XD3-A	2.8063	2.8477	0.6606	2.0151	1.3543
XD3-B	4.2819	4.3636	0.7505	3.2902	3.4615
XD3-C	2.0904	2.2730	0.4889	1.5100	1.1060
XD4-A	7.5026	7.1393	1.3356	3.3571	2.7814
XD4-B	11.5500	8.7915	2.0688	6.5626	5.7632
XD4-C	6.8852	6.1632	1.1760	3.2417	4.1440
XD5-A	2.5710	2.5753	1.1205	1.8074	1.8252
XD5-B	3.5392	3.8239	1.1824	2.4090	2.8118
XD5-C	2.1326	2.2670	0.9016	1.4990	1.2985

The Swain-Fu distance (Kotz and Johnson, 1988) is a measure of distance between two classes and provides a metric for clustering efficiency. The Swain-Fu

distance Φ_{12} between two *n*-dimensional distributions with mean μ_i and covariance Λ_i is defined as

$$\Phi_{12} = \frac{|\mu_1 - \mu_2|}{H_1 + H_2} \tag{3-7}$$

where distance $H_i (i = 1, 2)$ is calculated as:

$$H_i = \left\{ \frac{|\mu_1 - \mu_2|^2 (n+2)}{tr\left[\Lambda_i^{-1} (\mu_1 - \mu_2)(\mu_1 - \mu_2)' \right]} \right\} \tag{3-8}$$

Table 3-22: Swain-Fu distances between disturbance classes in TE process using XTW^{SP}

	XD1	XD2	XD3	XD4	XD5
XD1	-	6.1088	3.5912	4.853	4.0327
XD2	37.4667	-	3.4793	5.4281	5.8406
XD3	13.2203	5.9372	-	6.9653	7.6293
XD4	27.5624	14.4365	8.7958	-	21.8929
XD5	20.7789	11.686	**2.5355**	4.6523	-

Table 3-23: Swain-Fu distances between disturbance classes in TE process using DTW_1^{SP} and DTW_1

	DTW_1^{SP}					DTW_1				
	XD1	XD2	XD3	XD4	XD5	XD1	XD2	XD3	XD4	XD5
XD1	-	6.2107	4.5502	7.788	5.2615	-	5.3289	2.6305	7.7558	3.7259
XD2	39.1359	-	3.4532	6.7929	6.0894	36.988	-	4.4738	8.4697	22.6327
XD3	15.8514	6.4657	-	12.6588	8.8015	13.2063	19.2704	-	8.5441	5.7449
XD4	40.8741	13.5088	30.4676	-	22.0161	63.6292	72.4435	57.6182	-	18.0194
XD5	13.0514	12.0842	**2.8562**	8.0416	-	11.75	4.4832	**2.1063**	7.4072	-

Table 3-24: Swain-Fu distances between disturbance classes in TE process using DTW_2^{SP} and DTW_2

	DTW_2^{SP}					DTW_2				
	XD1	XD2	XD3	XD4	XD5	XD1	XD2	XD3	XD4	XD5
XD1	-	3.7693	2.3385	5.0614	3.5331	-	3.7315	1.4898	3.7475	4.0377
XD2	17.4189	-	2.5111	2.8686	5.7546	29.3136	-	2.4418	3.6402	5.2075
XD3	14.9503	6.111	-	4.5207	9.3163	9.8118	4.7159	-	4.8798	4.7557
XD4	43.8458	9.2131	9.0473	-	15.5219	54.6438	11.6558	14.294	-	26.8869
XD5	8.5421	6.5947	**1.7665**	5.2571	-	6.2438	3.6404	**1.0057**	4.2501	-

The Swain-Fu distances between the five different classes among the 15 runs were calculated for the different signal comparison methods as shown in Table 3-22 to

Table 3-24. The larger the minimum distance, the clearer a method can separate each class of operation. When the minimum distance is less than 1, the five disturbances are not clearly separable. The minimum distance from XTW^{SP}, DTW_1^{SP}, and DTW_2^{SP} are 2.5355, 2.8562, and 1.7665, respectively and all the disturbance classes can be effectively differentiated. In contrast, the minimum distance from DTW_1 and DTW_2 are 2.1063 and 1.0057, respectively. The larger Swain-Fu distances of the singular points enhanced time warping methods reiterate their superiority to traditional DTW.

3.5.4 Case Study 3: Identifying Transitions during a Fed-batch Fermentation

In this section, we describe the use of the proposed method for state identification in a lab-scale fed-batch fermentation process. This process uses the yeast *Pichia pastoris* for the production of proteins such as antigens. The reader is referred to Muthuswamy and Srinivasan (2003) for more details of the process. Eight variables are being measured online in the process: Airflow, Stirrer speed, Dissolved oxygen (pO2), Cumulative base addition, Cumulative acid addition, pH, Exit O_2 concentration, and Exit CO_2 concentration. Further the carbon-dioxide evolution rate and O_2 uptake-rate are calculated by the control system based on the online measurements of O_2, CO_2. Data from seven of these variables are used for phase identification; Cumulative acid and base addition and pH are not used here. There are four important transitions in the operation of the fermentation process. T_1 occurs when dissolved oxygen level drops; T_2 and T_4 occur due to nutrient-exhaustion; T_3 occurs when the process has a critical cell mass. It should be noted that the duration of the different states are not the same across runs; neither do the state changes occur at the same times. The objective is to identify the transitions in the test run based on the knowledge of the transitions in a standard run. We have used data from Run SMB74 as the standard and from Run

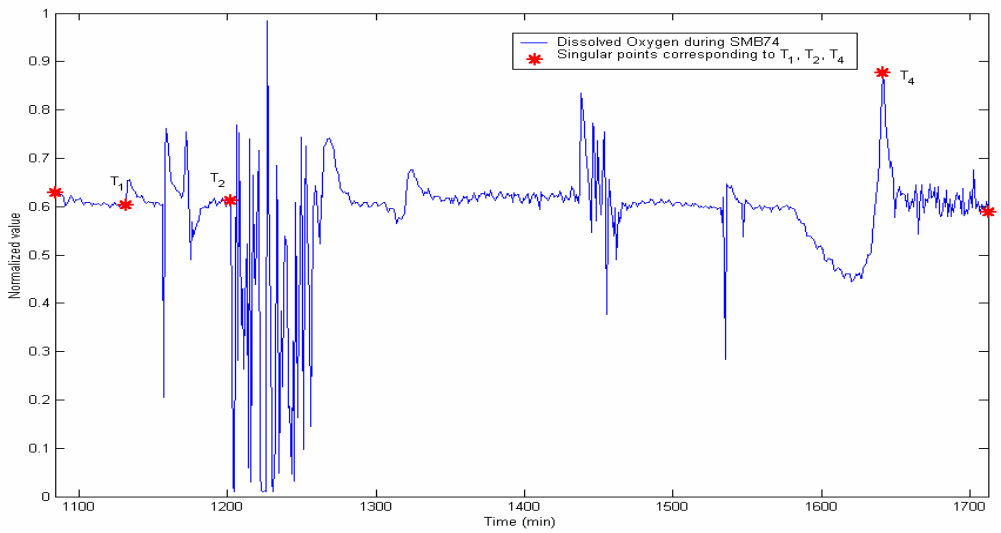
SMB78 as the test. Only T_1, T_2 and T_4 are considered here since T_3 is based on offline cell mass measurements.

For state identification, singular points were identified in all the seven variables. By analyzing the singular points and the transition times during SMB74, it is clear that dissolved oxygen is the key signal in the process and there are singular points in the data corresponding to T_1, T_2, and T_4 as shown in Figure 3-17(a). For SMB78 transition identification, in the interest of space, we report results only for the corresponding singular points although all singular points were used during signal comparison. XTW^{SP} was used for this purpose. The corresponding singular points in SMB78 are shown in Figure 3-17(b). The warped signal from SMB78 when plotted with the raw signal from SMB74 in Figure 3-17(c) shows the effectiveness of the time warping despite the large noise in the variable. As can be seen from the identified SMB78 transition timings in Table 3-25, the results are comparable to those reported by Muthuswamy and Srinivasan (2003) using a rule-based approach. The average difference of only 7 min illustrate that critical events during process operations coincide with singular points; therefore the proposed singular point augmented time warping is an efficient and intuitive data-driven approach to identify process phenomena. In the following, we illustrate the robustness of the proposed method..

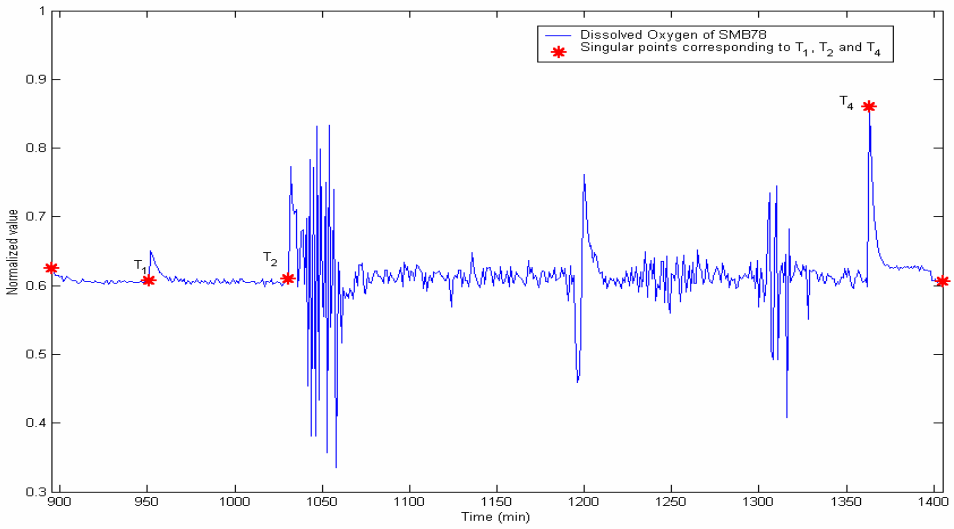
Table 3-25: Comparison between rule-based and XTW^{SP}-based transition detection

	SMB74 (min)	SMB74* (min)	SMB78(min)	SMB78* (min)
Start	1084	1084	895	895
T1	1132	1134	951	960
T2	1202	1231	1031	1042
T4	1641	1641	1363	1364
End	1712	1712	1405	1405

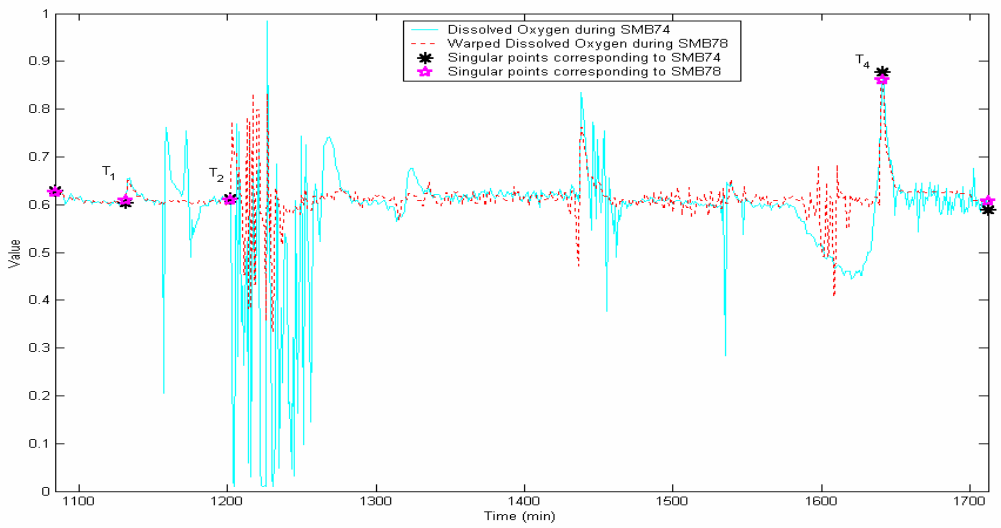
(* reported by Muthuswamy and Srinivasan, 2003)



(a)



(b)



(c)

Figure 3-17: Dissolved Oxygen profile during (a) SMB-74 and (b) SMB-78. (c) Signal Warping for transition identification based on singular points

3.5.5 Robustness to Tuning Parameters

Robust signal comparison using the singular point enhanced time warping methods depends on the robustness of (1) singular point identification, and (2) the singular point linking. The comparison of the singular episodes using time warping methods itself is deterministic and does not affect the performance. In the following, we illustrate the robustness of the former through a parametric study.

Singular point identification involves three parameters – ω, δ, and τ which affect singular point identification. While in general, different process signals may require different values for these parameters, we have found that most of the parameter settings can be used across variables and case studies. If necessary, fine-tuning can be done based on the features of a process signal. It should be noted that the comparison algorithm itself does not involve any case-specific parameters. Some basic guidelines are given below and illustrated using results from XTW^{SP} for two case studies.

The inspection window ω for finding the sharp changes points can be specified based on the variable's speed of response. If the sampling time is short compared to the time-constant of the variable, ω would have to be increased; conversely, if the sampling time is long, ω should be decreased to prevent the accumulated effect of a slow change from being flagged as a sharp change. In all the case studies in Section 4, the inspection window was set to 8 samples. A study of the sensitivity to ω was performed for Case Study 1. Consider the signal in Figure 3-15. Setting $\delta = 0.05$ and $\tau = 96$, ω was varied from 4 to 12 as shown in Table 3-26. There is no difference in the time of detection of most of the stages; even in the others the difference was found to be at most 12 samples, that is, two minutes in the entire 23 hour long startup of the process. This illustrates the robustness of the method to the settings of ω.

The neighborhood τ can be identified based on the distance between two sharp changes points that can be concatenated into one sharp change episode. It needs to be specified based on the signal characteristics, sampling time, and process knowledge. If the variable has extensive high frequency noise which should be discounted during signal comparison, τ should be increased so that adjacent sharp change episodes are concatenated. The proposed method is robust to a wide range of τ values. In Case Study 1, with $\delta = 0.05$ and $\omega = 8$, when τ is increased from 40 to 120 samples, the maximum variation in the results is 22 samples, that is 3.67 minutes (see Table 3-27), which is acceptable for this process.

Table 3-26: ShadowPlant stage identification using XTW^{SP} with different ω

ω	4	5	6	7	8	9	10	11	12	Max shift
Start	1	1	1	1	1	1	1	1	1	-
T_1	1935	1935	1935	1935	1937	1937	1931	1931	1927	10
T_2	3041	3041	3041	3041	3041	3041	3041	3041	3041	0
T_3	3960	3959	3959	3954	3954	3948	3948	3948	3948	12
T_4	6511	6511	6511	6511	6511	6511	6511	6511	6511	0
T_5	7074	7074	7074	7074	7074	7074	7074	7074	7074	0
T_6	7339	7339	7339	7339	7339	7339	7339	7339	7339	0
End	8280	8280	8280	8280	8280	8280	8280	8280	8280	-

Table 3-27: ShadowPlant stage identification using XTW^{SP} with different τ

τ	40	48	56	64	72	80	88	96	120	Max shift
Start	1	1	1	1	1	1	1	1	1	-
T_1	1937	1937	1937	1937	1937	1937	1937	1937	1937	0
T_2	3041	3041	3041	3041	3041	3041	3041	3041	3041	0
T_3	3954	3954	3954	3954	3954	3954	3954	3954	3954	0
T_4	6511	6500	6500	6500	6511	6511	6511	6511	6511	11
T_5	7073	7073	7073	7073	7074	7074	7074	7074	7073	1
T_6	7326	7324	7321	7321	7321	7317	7317	7339	7339	22
End	8280	8280	8280	8280	8280	8280	8280	8280	8280	-

Threshold is the most important parameter in singular points identification and may require tuning based on the noise level and signal profile. If the noise in the signal increases, δ should also be increased. If adjacent extrema can have a large difference in their magnitudes, the δ value should be increased. For most variables in all the case studies, we found δ =5% to be a suitable choice. The robustness of δ was also studied

in Case Study 1 by varying it from 0.02 to 0.10, with ω and τ fixed at 8 and 96 respectively, as shown in Table 3-28. A shift of about 16 samples, that is 2.66 min, was observed in two stages. For all other stages, there were no differences in the stage identification time.

Table 3-28: ShadowPlant stage identification using XTW^{SP} with different δ

δ	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	Max shift
Start	1	1	1	1	1	1	1	1	1	-
T_1	1922	1928	1937	1937	1937	1937	1935	1935	1935	15
T_2	3041	3041	3041	3041	3041	3041	3041	3041	3041	0
T_3	3944	3944	3946	3954	3960	3960	3960	3961	3961	17
T_4	6511	6511	6511	6511	6511	6511	6511	6511	6511	0
T_5	7074	7074	7074	7074	7074	7074	7074	7074	7074	0
T_6	7339	7339	7339	7339	7339	7339	7339	7339	7339	0
End	8280	8280	8280	8280	8280	8280	8280	8280	8280	-

From the process state point of view, T_1 and T_3 are denoted by sharp change points which are affected only by ω and δ. Stage T_2 is indicated by an extreme point; such singular points are only weekly affected by δ and unaffected by ω and τ. Stages T_4, T_5, and T_6 are associated with trend change points and are affected only by τ

Table 3-29: Minimum Group separability ratio in TE process for different parameter settings

			$\tau = 96$ $\delta = 0.06$		
ω	4	6	8	10	12
XD3-C	1.5901	1.5717	1.5371	1.3338	1.6175
XD5-C	1.5205	1.5582	1.5566	1.4614	1.6796
			$\omega = 8$ $\delta = 0.06$		
τ	64	80	96	120	160
XD3-C	1.5562	1.4753	1.5371	1.6314	1.5652
XD5-C	1.5164	1.5143	1.5566	1.6010	1.4568
			$\omega = 8$ $\tau = 96$		
δ	0.04	0.06	0.08	0.10	0.12
XD3-C	1.6303	1.5371	1.2967	1.1777	1.0443
XD5-C	1.6566	1.5566	1.4765	1.4413	1.4403

A similar robustness study was conducted for the other case studies as well. In the interest of space, only a summary of the results from Case Study 2 is presented here. Table 3-29 shows the two smallest group separability ratios S for different ω, τ, and δ settings. In all cases, *S* was larger than 1 and all the different operating states

can be easily distinguished. The minor changes in S for different settings of ω, τ and δ show that singular point augmented time warping is a robust method. Whenever the major singular points was detected the algorithm will works.

3.6 Discussion

Signal comparison is important for process monitoring, fault diagnosis, and process stage identification. In this chapter, we presented a new approach for signal comparison based on singular points and time warping. Process data is first segmented based on singular points. Dynamic programming and time warping is used to find their optimal match and obtain the signal difference. Singular point augmentation can be used with traditional DTWs, the role of the latter in this case is for episode-wise comparison. In such cases, the proposed method improves the quality of signal comparison. It can also be used with the computationally efficient extrapolative time warping method which uses a greedy search instead of dynamic programming. The disadvantage of XTW is that it does not guarantee a globally optimal difference between two signals, but this is not an essential requirement since the linkage of the singular points enforces the matching of the corresponding episodes. A performance comparison of the singular point augmented time warping method with DTW reveals a substantial decrease in computational cost, which makes it amenable for large-scale case studies. Its extension for online signal comparison has been explored and will be reported in a subsequent communication.

The XTW^{SP} represents an attractive tradeoff between computational complexity and mathematical optimality that is suitable for process operation applications. This was established using three different systems to solve three different operational problems: operations stage identification during startup of a simulated FCCU, disturbance identification in the Tennessee Eastman plant, and transition identification

in a fed-batch fermentation process. In all cases, the method correctly identified the corresponding points of the variables and found an operationally relevant signal difference.

The proposed method bears some similarity with the correlation optimized warping (COW) method proposed by Nielsen et al. (1998). COW aligns two signals by splitting them into a number of equal segments and segment-wise linear stretching and compression. The quality of alignment of each segment is calculated based on the correlation coefficient of the corresponding segments. The optimal combination of segment warping is the one that yields the largest value of the summed correlation coefficients and obtained using dynamic programming (Pravdova et al., 2002). In contrast to COW, where the original signal is divided into equal sections, in our work, segments are defined based on singular episodes of varying lengths. Also, we use time warping to warp the signals in contrast to the linear interpolation used in COW.

The proposed approach also shares similarities with the qualitative trends of Rengaswamy and Venkatasubramanian (1995). Qualitative trends as well as singular episodes decompose the signal into segments – the former based on shapes and the later based on singular points. Both these classifications are broadly based on first and second derivatives of the signal. The singular episodes retain the magnitude and duration information and are therefore more akin to the enhanced trends of Sundarraman and Srinivasan (2003). The proposed approach has several advantages when compared with the trend-based approaches. (1) The singular points guarantee that only the corresponding process states will be compared – an essential requirement in transition type operations. (2) It is more robust to slow drifts and high frequency changes which are usually difficult to deal with using qualitative trends. The episode-wise difference calculated by the proposed approach also has another application. It

can be used to locate the stages in the process where large differences are observed; this can be used to provide additional stage specific information, such as root cause identification for deviations in batch-type operations.

A performance comparison of the singular point augmented time warping method with DTW reveals a substantial decrease in computational cost, which makes it amenable for large-scale case studies. Its extension for online signal comparison is reported in Chapter 5.

Chapter 4 Online Fault Diagnosis and State Identification using Dynamic Locus Analysis

4.1 Introduction

Modern chemical plants are large in scale and highly complex. Due to market forces, increasingly, these plants are operated in a multitude states and frequently transition 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., 2005) the 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 (Venkatasubramanian, et al., 2003). However, most online fault detection and diagnosis methods in literature assume that the process is in a known, well characterized state (for 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 during process transitions:

1) Unknown initial state: Since agile processes are frequently in dynamic states, the diagnoser cannot assume that the process is in an *a priori* known state when diagnosis is started. It is therefore necessary to first identify the current state of the process from the online signals before evaluating if the process is in a normal condition.
2) Partial information: Since state identification 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 change 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 the 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 time-delays.
5) Low computation cost: The computational load should be modest, so that method can be deployed online even for large-scale processes with thousands of sensors.

In this Section, 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 section is organized as

follows: The dynamic locus analysis methodology for online state identification is proposed in Section 4.2. In Section 4.3, we illustrate the proposed approach using three case studies – the Tennessee Eastman challenge process, a lab-scale distillation column, and a simulation of a fluidized catalytic cracking unit. Summary and discussion from this work are presented in Section 4.4.

4.2 Dynamic Locus Analysis

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

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

For 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. Therefore, two aspects are essential in the identifying the current state of a process – (1) finding the corresponding start point in the reference signal (that is, linking l to l^*); and (2) comparing the real-time signal segment and possible segment in reference signal (that is, calculating the difference between two segments). If starts from each possible point do an independent comparison, the computation load will be unacceptable. We overcome this requirement using a concept similar to Waterman's (1987) RNA matching algorithm. The computational performance is

suitable even for real-time usage since a short real-time signal is usually adequate for state identification (See Section 4.3). In this thesis, a methodology which can identify the optimal section which has minimum warping distance with real-time signal segment was called dynamic locus analysis. It can identify the locus in reference signal with the real-time signal evolution.

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 sub-segment $\{x_1, x_2, x_3, ..., x_i\}$ in *X* and the sub-segment $\{y_l, y_{l+1}, y_{l+2}, ..., y_j\}$ in *Y*. In the general case, *l* is unknown and is determined using dynamic programming.

$$D_S(i,j) = \min_F \left\{ \sum_{d=1}^{i} \Delta(x_d, y_{j(d)}) \right\} \tag{4-1}$$

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

$$F = \{c(1), c(2), ..., c(i)\}$$

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 thesis we use the Euclidean metric

$$\Delta(x_i, y_j) = |x_i - y_j| \tag{4-2}$$

Note that $D_S(i,j)$ is not the total minimum distance between $\{x_1, x_2, x_3, ..., x_i\}$ and $\{y_1, y_2, ..., y_j\}$, 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 Waterman's (1987) dynamic programming approach and the latter through Time Warping.

In dynamic locus analysis, the optimal matching path between $\{x_1, x_2, x_3, ..., x_i\}$ 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. A warping slope between [½ 2] has been found to be adequate in terms of computational complexity while minimizing over-warping.

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*.

$$D_S(1, j) = \Delta(x_1, y_j) \quad j \in [1\ n] \tag{4-3}$$

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

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

$$D_S(i,1) = \infty \quad i \in [3\ m] \tag{4-4}$$

Step 2: Matrix Propagation

Other elements of D_S are constructed recursively using the Itakura local continuity constraint as follows:

$$D_S(i,j) = \min\left\{D_S(i-1,j-1) + \Delta(x_i, y_j), F_{i,j}, G_{i,j}\right\} \quad i \in [2\ m] \quad j \in [2\ n] \tag{4-5}$$

$$F_{i,j} = D_S(i-1, j-2) + \Delta(x_i, y_j) \tag{4-6}$$

$$G_{i,j} = D_S(i-1, j) + \Delta(x_i, y_j) \text{ or } \infty \text{ if } G^* \tag{4-7}$$

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

$$j^* = \underset{j}{\operatorname{argmin}}\left\{D_s(m,j)\right\} \quad j \in [1\ n] \tag{4-8}$$

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 these two 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 Figure 4-1a and the resulting total search space for dynamic locus analysis is shown in Figure 4-1b. It should be noted that the dynamic locus analysis search space for calculating $D_S(m, j^*)$ is larger than DTW's search space for $D(X, Z^*)$.

Step 3: Tracing the locus of *X* in *Y*

To re-create 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.

$$P(i,j)=\begin{cases}0 \ \text{ if } D_S(i,j)=G_{i,j}\\ 1 \ \text{ if } D_S(i,j)=D_S(i-1,j-1)+\Delta(x_i,y_j)\\ 2 \ \text{ if } D_S(i,j)=F_{i,j}\end{cases} \tag{4-9}$$

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 , $j^*-2*m \le l^* \le j^*-\frac{1}{2}*m$.

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

$$\eta=\frac{\sum_{d=1}^{n}\left|y_{j(d)}-x_d\right|}{n} \tag{4-10}$$

and is the DTW distance between *X* and the locus of *Y*.

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 (faults, human errors, etc). During online monitoring, the real-time signal is compared with every reference signal in the 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. 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. 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* 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. The application of these metrics is illustrated using case studies in Section 4.3. During transitions in large-scale agile processes, because of simultaneous manual or automated control actions in different sections, operations personnel need information about known the current state of each plant section. Information from the corresponding point calculated during dynamic locus analysis can be used for online state identification and characterization as illustrated in Section 4.3.3.

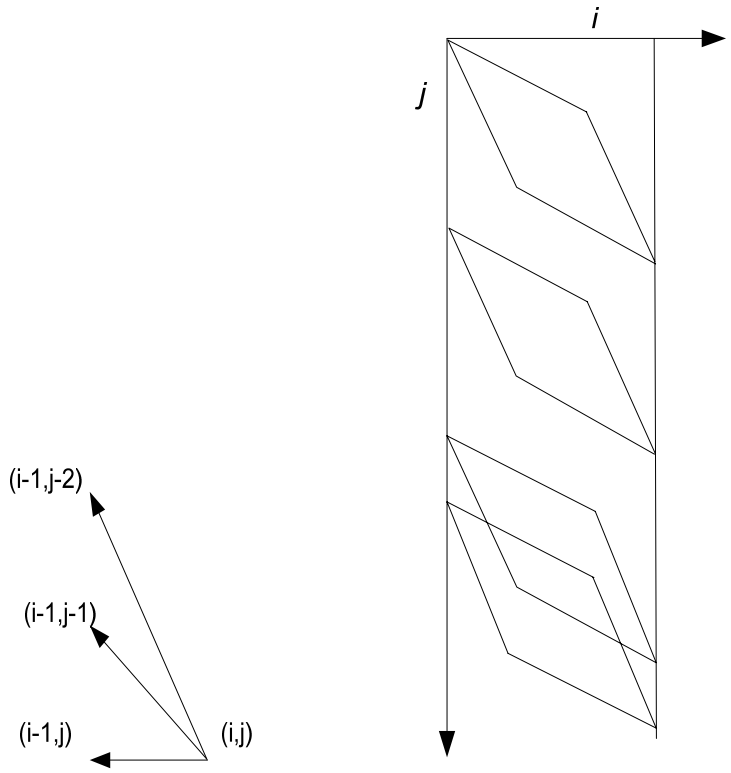


Figure 4-1: **(a):** Search space of dynamic locus analysis **(b)** Itakura local constraint

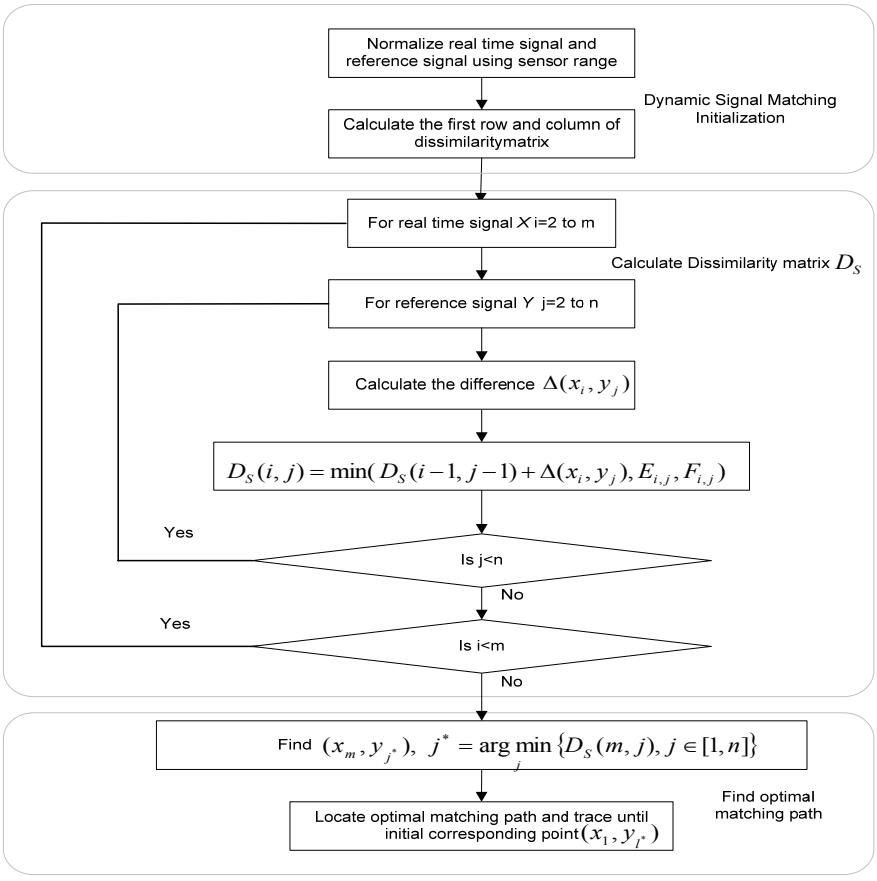


Figure 4-2: Flowchart of dynamic locus analysis

4.2.1 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 Figure 4-3.

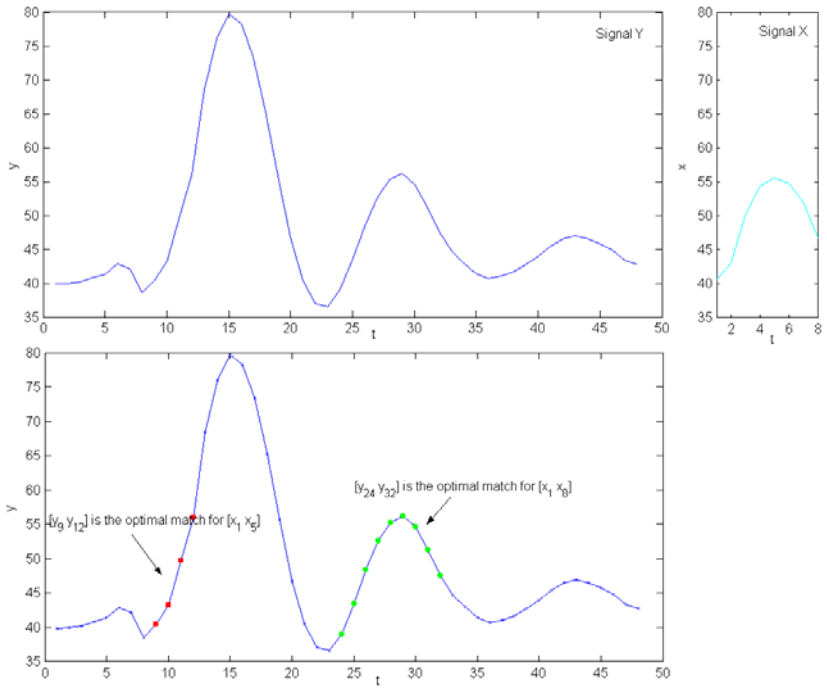


Figure 4-3: (a) Reference and real-time signal for illustrative example. (b) Corresponding points as identified by dynamic locus analysis

Step 0: Signal normalization

Normalize *X* and *Y* to [0 1] based on the range of the sensor (Srinivasan et al. 2004).

Step 1: Dissimilarity matrix initialization

The dissimilarity matrix is next calculated. The first column of D_S is calculated following (14). For example, $D_S(1,1) = |0.4070 - 0.3988| = 0.0082$. 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,

$D_S(3,1) = D_S(4,1) = \ ... \ = D_S(8,1) = \infty$ ..

Step 2: Dissimilarity Matrix Propagation

All other elements of D_S are calculated using (4-5, 4-6 and 4-7) using dynamic programming. This is similar to the Itakura local constraint in DTW. For example,

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

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 4-1 and 4-2, respectively. All other elements in the 2nd 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 4-2 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 $[y_{24}\ y_{32}]$ is the locus of X in Y. The search map is shown schematically in Figure 4-4.

The reader may note from Table 4-1 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 sub-segment $\{y_9, y_{10}, y_{11}, y_{12}\}$ better matches with $\{x_1, x_2, x_3, x_4, x_5\}$ as can be confirmed from Figure 4-3. However, since $\{x_6, x_7, x_8\}$ does not have a good match in the neighborhood of y_{12}, this sub-segment

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 then $D_S(8,32)$.

The *stability* of the locus identified by this method is also apparent from Table 4-1. 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 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 (4-2) has to be replaced by:

$$\Delta(x_i, y_j) = \sum_{c=1}^{Q} \left(w_c \left| x_{i,c} - y_{j,c} \right| \right) \tag{4-11}$$

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 thesis, since all variables are normalized based on the sensor range, 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 industrial-scale processes with a large number of variables, key variable selection or hierarchical grouping (Srinivasan, *et al,* 2005) may be needed to reduce computational cost or memory requirements.

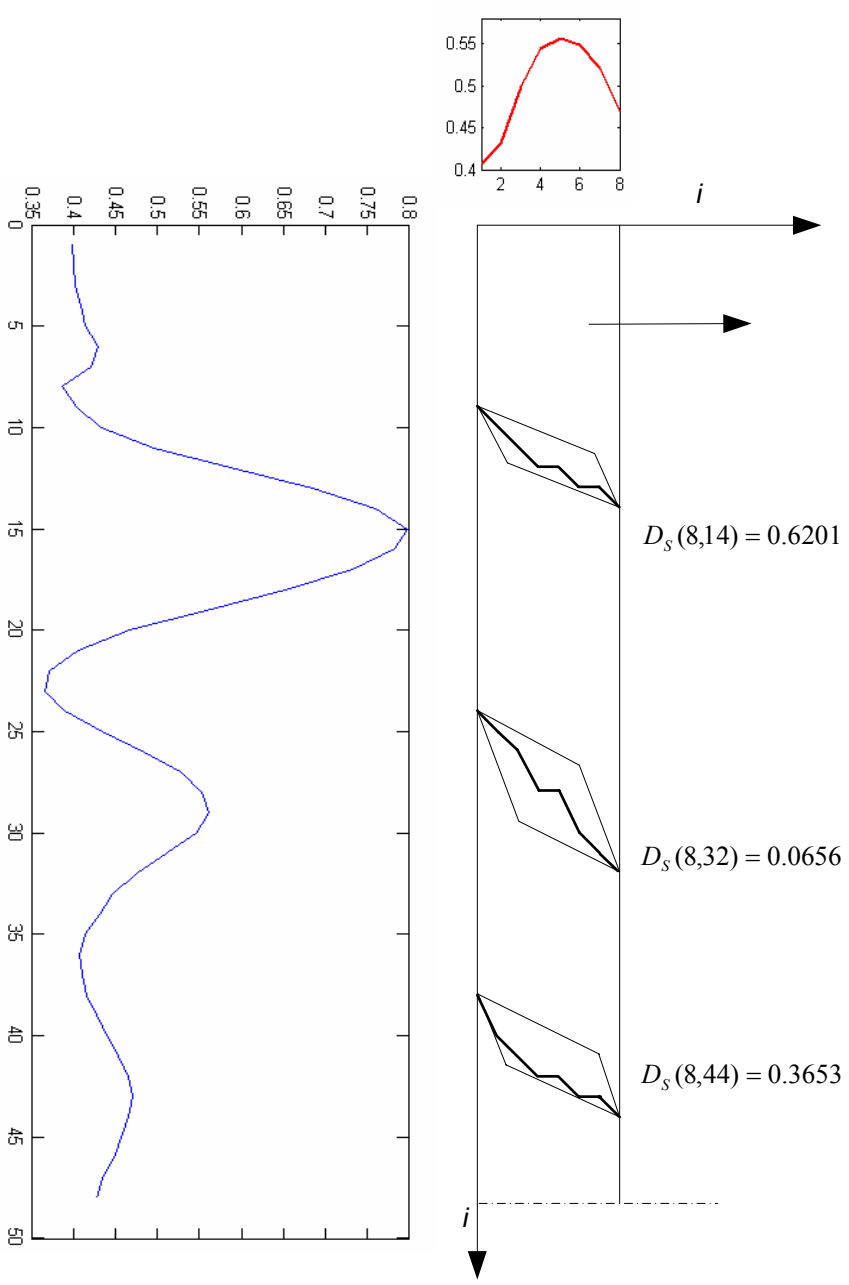


Figure 4-4: Illustrate case searching path

Table 4-1: Dissimilarity matrix of illustrate case

$D_S(i,j)$	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	0.0071	0.0382	Inf.	Inf.	Inf.	Inf.	Inf.	Inf.
2	0.3996	0.0074	0.0385	0.1385	0.2831	Inf.	Inf.	Inf.	Inf.
3	0.4015	0.0055	0.0351	0.1366	0.2793	0.4372	0.5835	Inf.	Inf.
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

Table 4-2: Parent matrix of illustrate case

$D_S(i,j)$	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

4.3 Case Studies

In this section, we evaluate the proposed dynamic locus analysis for online fault diagnosis and state identification disturbance using three case studies

4.3.1 Case Study 1: Online Disturbance Identification in Tennessee Eastman Plant

In this section, we use dynamic locus analysis to identify unknown process disturbances online and estimate their progress. Five disturbances henceforth referred to as XD1 – XD5, which affect the A feed flowrate, reactor pressure, reactor level, reactor temperature, and compressor work are considered. 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 kscmh to 0.3902 kscmh (a 60% change) in three steps starting at t=51 min as shown in Table 4-3. 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 Figure 3-11. Two other instances XD1-B and XD1-C with changes of magnitude 55% and 50% were also introduced. Similar changes were introduced to bring forth disturbance classes XD2 – XD5. The details of the TE process data generation is described in Section 3.4.2.

Table 4-3: Disturbance profiles for TE process XD1

	Target	Time (min)	Target	Time (min)	Target	Time (min)	Target	Time (min)
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

Like any process history-based method, the accuracy of the dynamic locus analysis method depends on the representative-ness 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 4.3.3. 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 reference database. Other runs were used to evaluate the method.

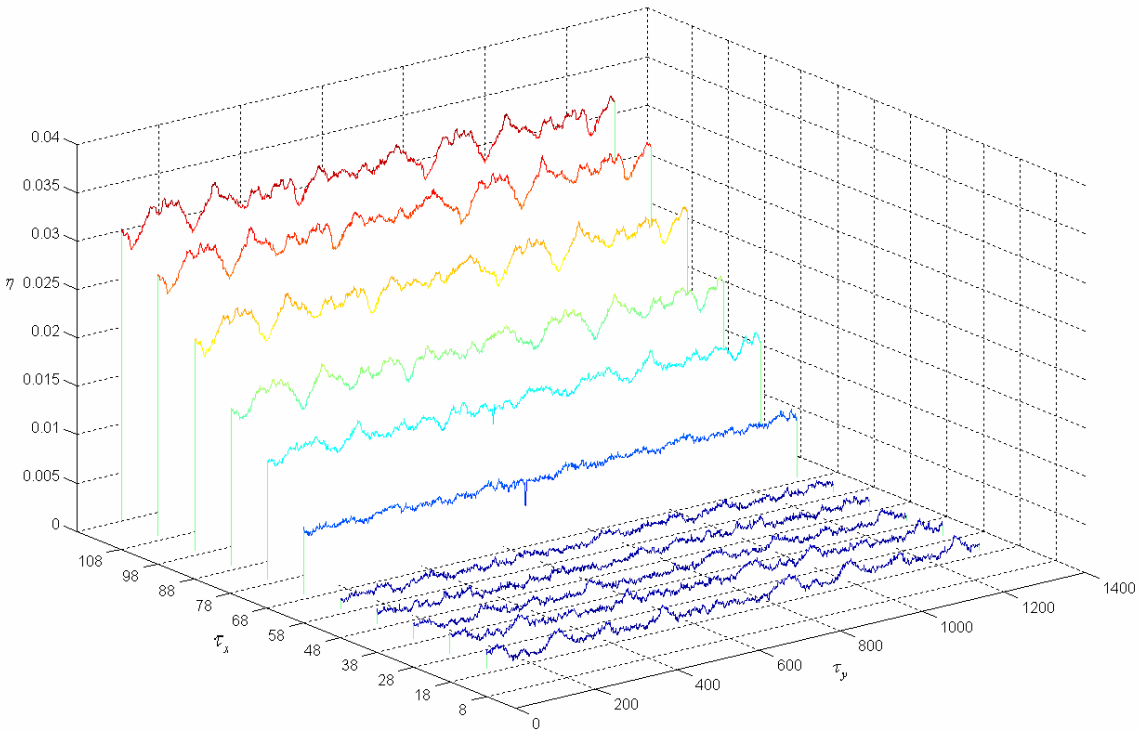


Figure 4-5: Online comparisons of real-time signal and reference signal XD0 at 10 sample snapshots from $\tau_x = 8$ to 108

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. A fault is introduced into the process at t = 51 min. Figure 4-5 shows the results from dynamic locus analysis of real-time signals from time τ_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 from the small normalized difference η. Starting at τ_x = 54 samples, the η with XD0-B increases as does the inseparability ratio α shown in Figure 4-6 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 Figure 4-6). At τ_x = 56 samples, α falls below $\alpha_{\min}$ identifying that the process is undergoing disturbance XD2 (See Figure 4-7). The location of minimum η in these plots indicate the progression of the disturbance – thus τ_x = 56 samples corresponds to τ_y = 6th sample in the XD2-B. The progression of τ_y with time is coherent only with reference XD2-B as shown in Figure 4-8; the erratic progression with all other references also bears confirmation that the disturbance in this run is XD2. Figure 4-9 (c) reveals the clear separability of the disturbance throughout the run.

To further test the accuracy of the approach, starting at τ_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 τ_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. A summary of the result for this run is reported as Run-3 in Table 4-4. A similar study was performed for nine other runs. The comparison results of real-time signal with all reference signal from τ_x = 8 to 1270 samples are shown in Figure 4-9 (a, Run-1, b, Run-2, c, Run-3, d, Run-4, e, Run-5, f, Run-6, g, Run-7, h, Run-8, i, Run-9, j, Run-10). In all cases, the proposed method correctly identified the disturbance with an average delay of 5 samples. Figure 4-9 reveals the clear separability of the disturbance throughout all runs. The computation time for the complete analysis at each time sample is 2.26 seconds (on a Pentium IV, 2.4 GHz cpu).

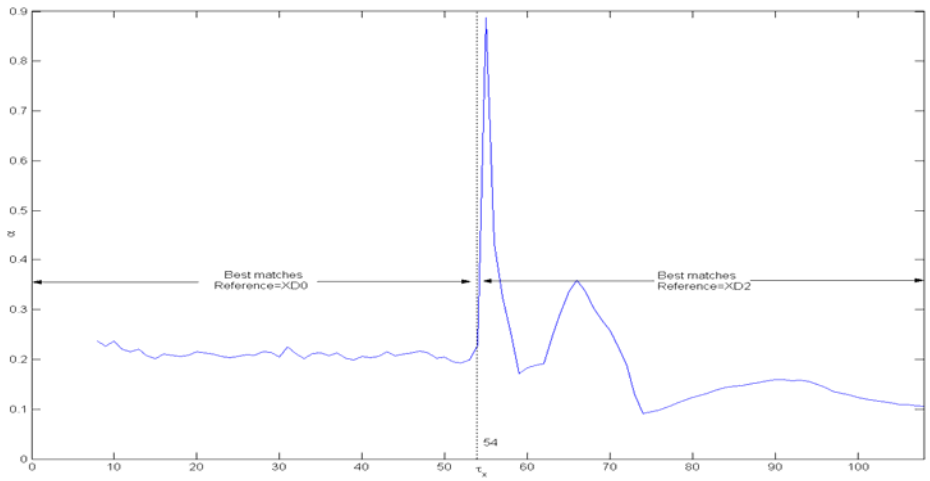


Figure 4-6: Inseparability ratio during the first 110 minutes of Run-3 of TE process

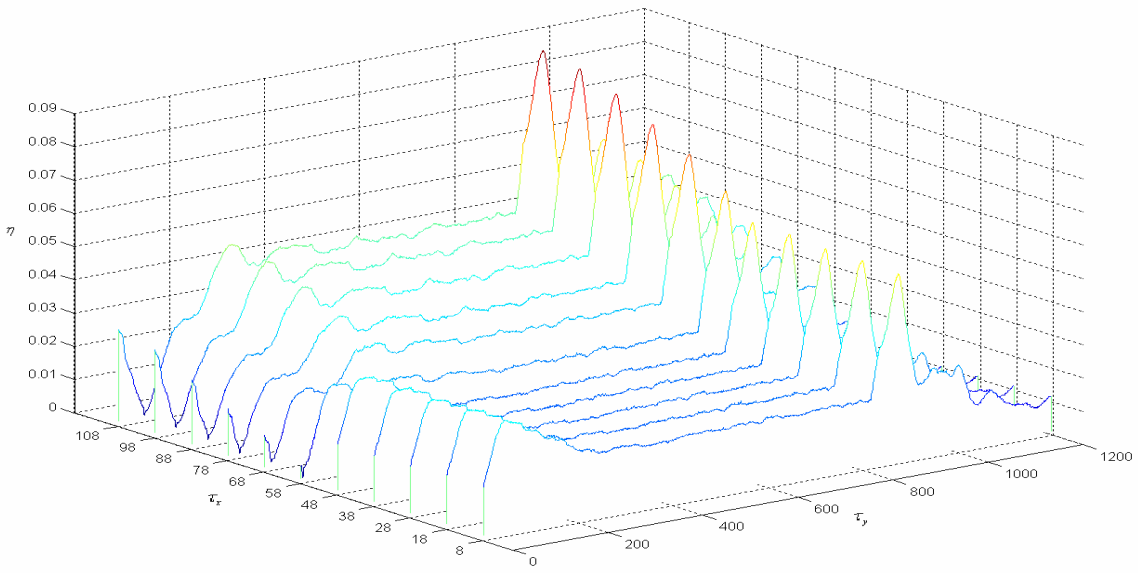


Figure 4-7: Online comparisons of real-time signal and reference signal XD2 at 10 sample snapshots from τ_x = 8 to 108 samples

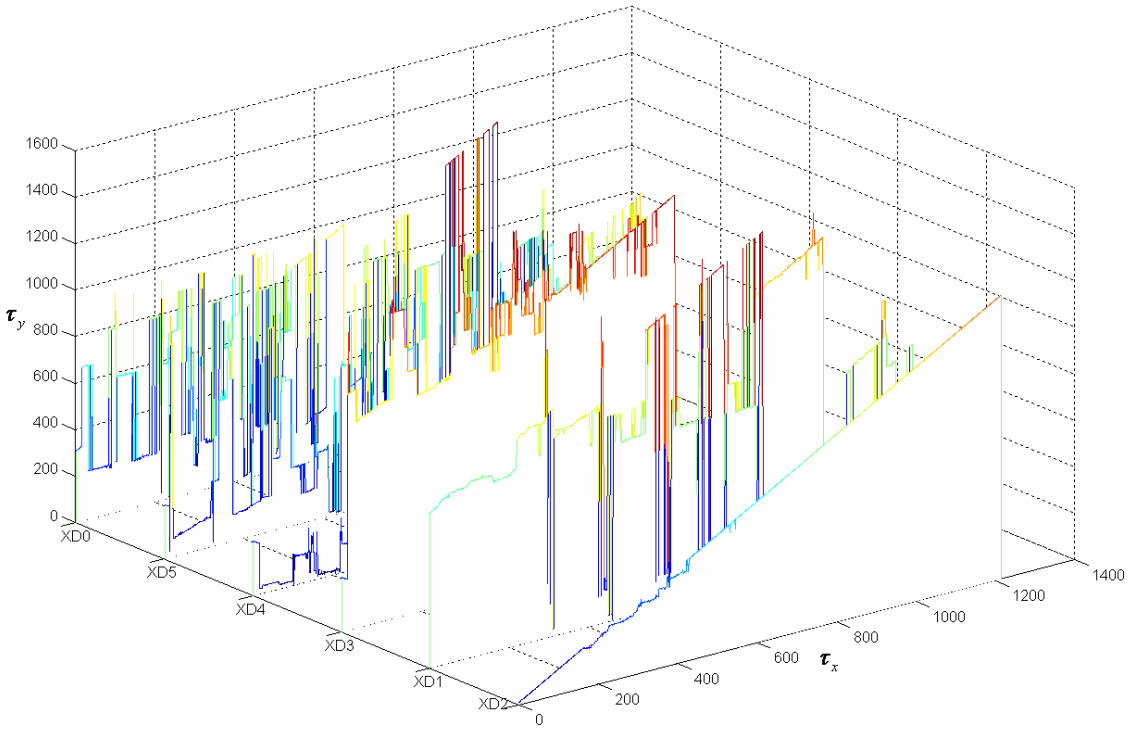
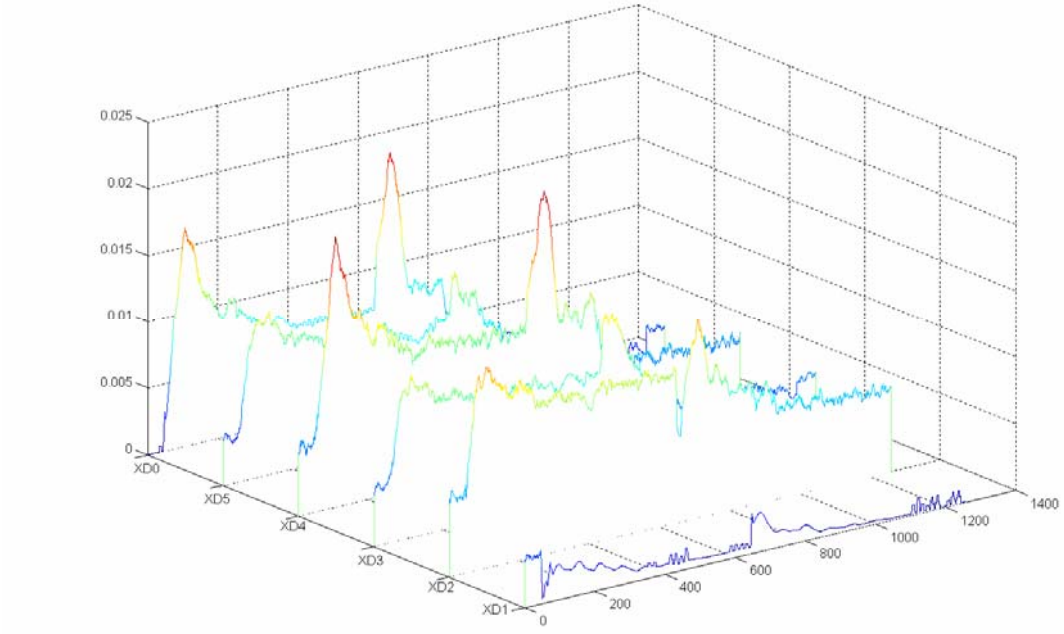
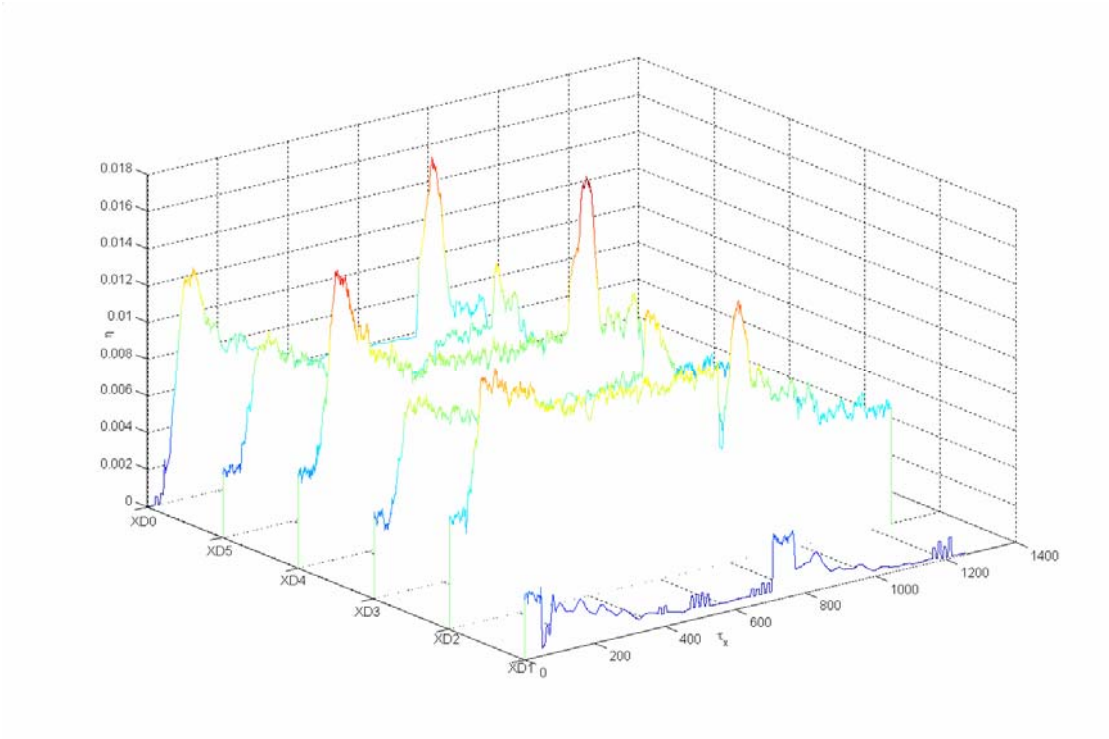


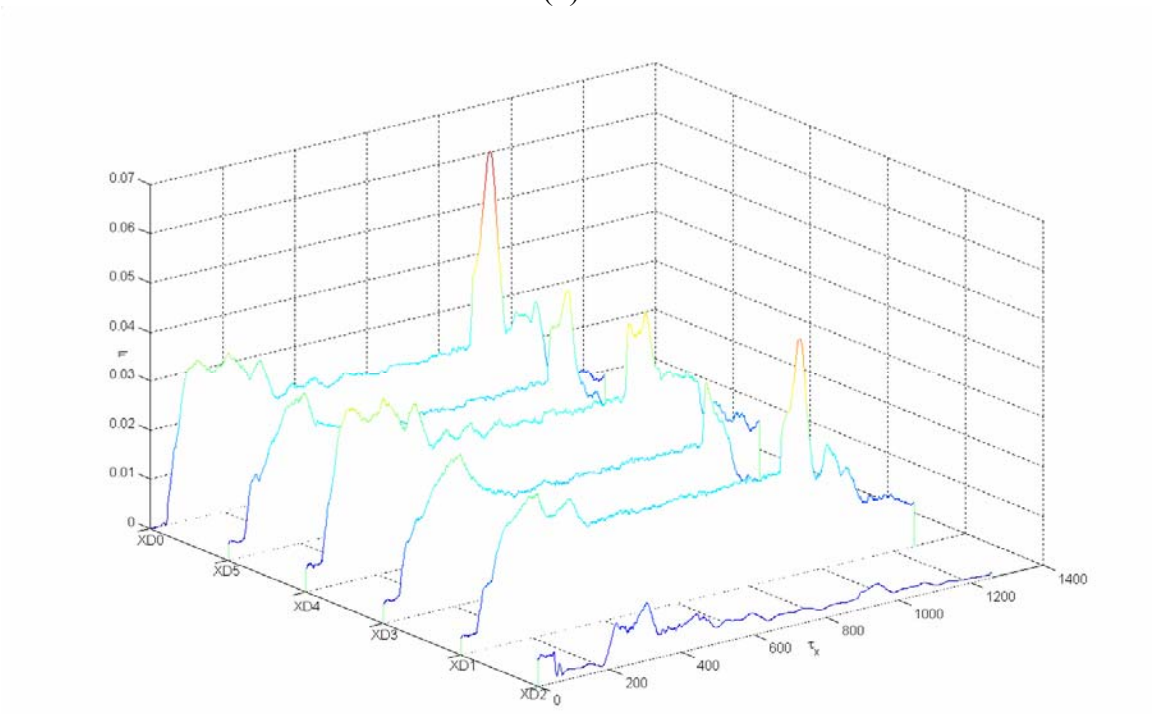
Figure 4-8: Time progression of corresponding points when real-time signal is compared with all the reference signals throughout the run.



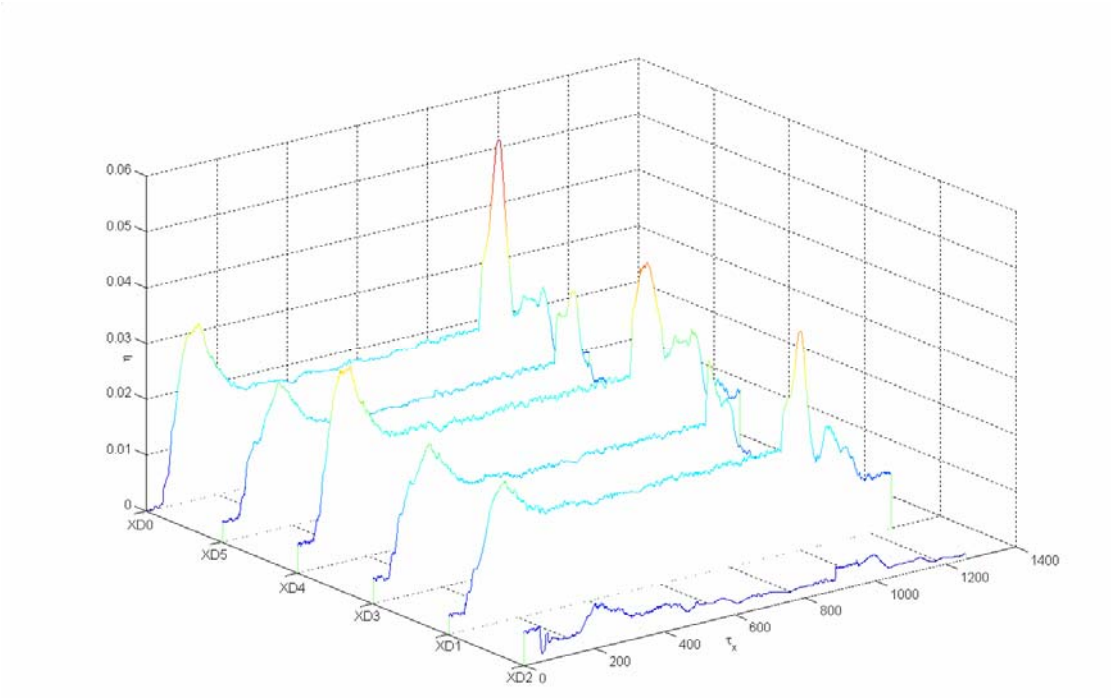
(a)


0.018
0.016
0.014
0.012
0.01
η
0.008
0.006
0.004
0.002
0
XD0
XD5
XD4
XD3
XD2
XD1
0
200
400
600
800
1000
1200
1400
τx

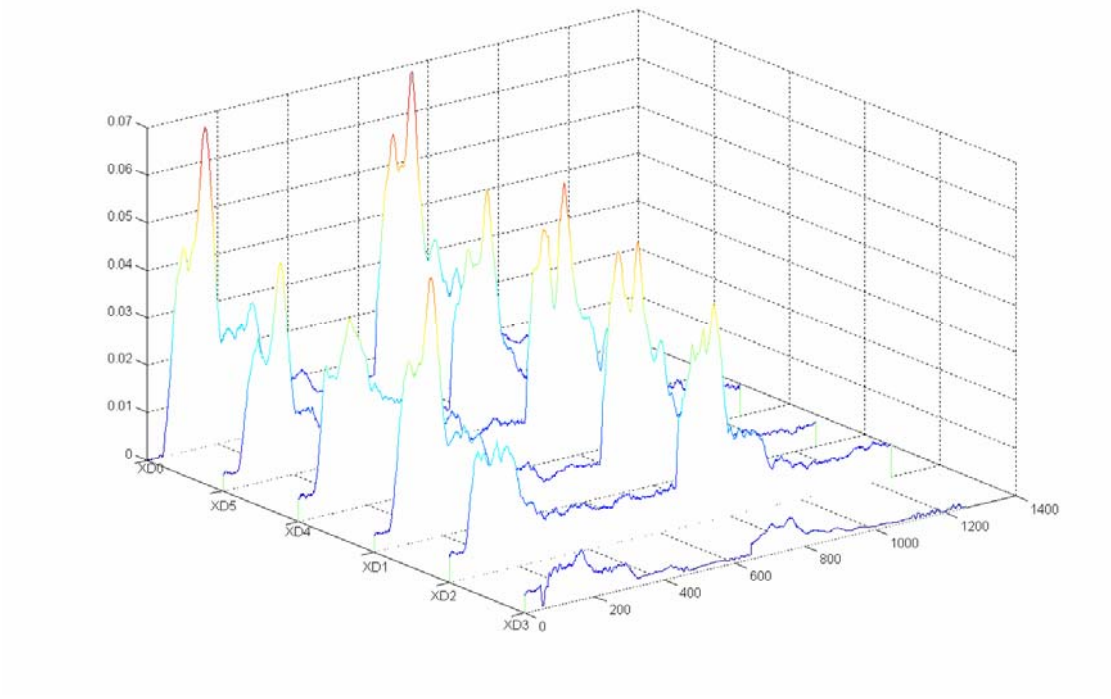
(b)


0.07
0.06
0.05
0.04
η
0.03
0.02
0.01
0
XD0
XD5
XD4
XD3
XD1
XD2
0
200
400
600
800
1000
1200
1400
τx

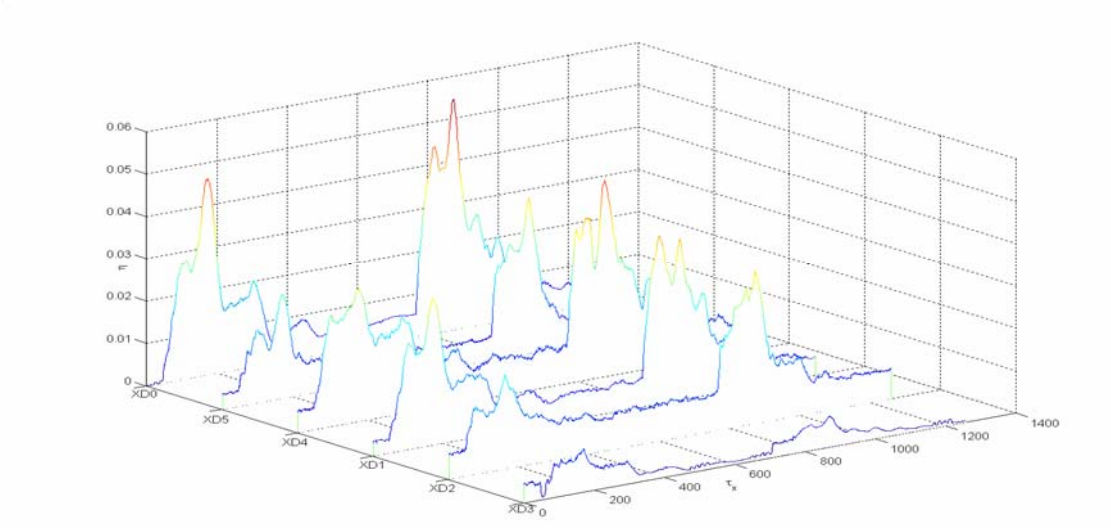
(c)


0.06
0.05
0.04
0.03
0.02
0.01
0
π
XD0
XD5
XD4
XD3
XD1
XD2
0
200
400
600
800
1000
1200
1400
τx

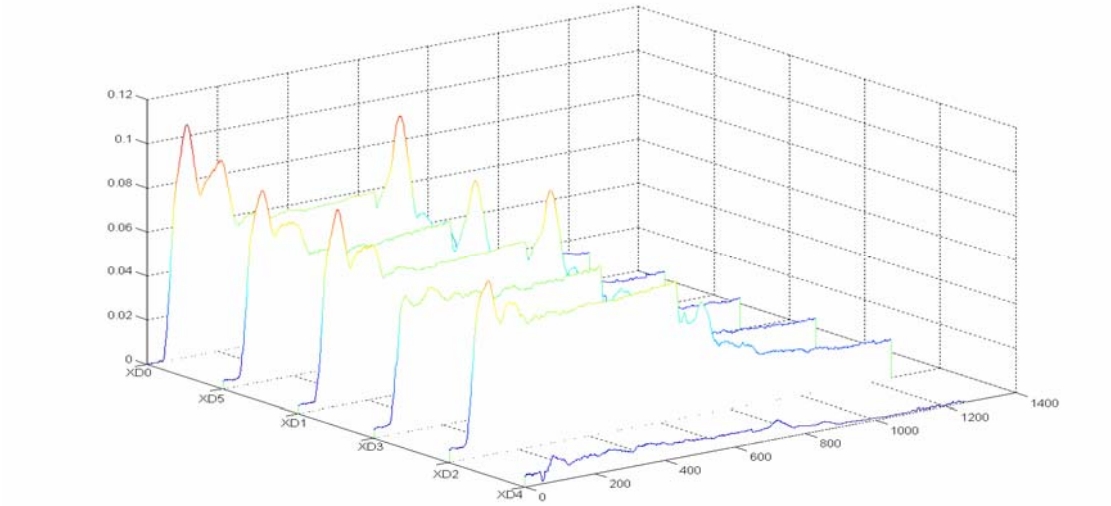
(d)


0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
XD0
XD5
XD4
XD1
XD2
XD3
0
200
400
600
800
1000
1200
1400

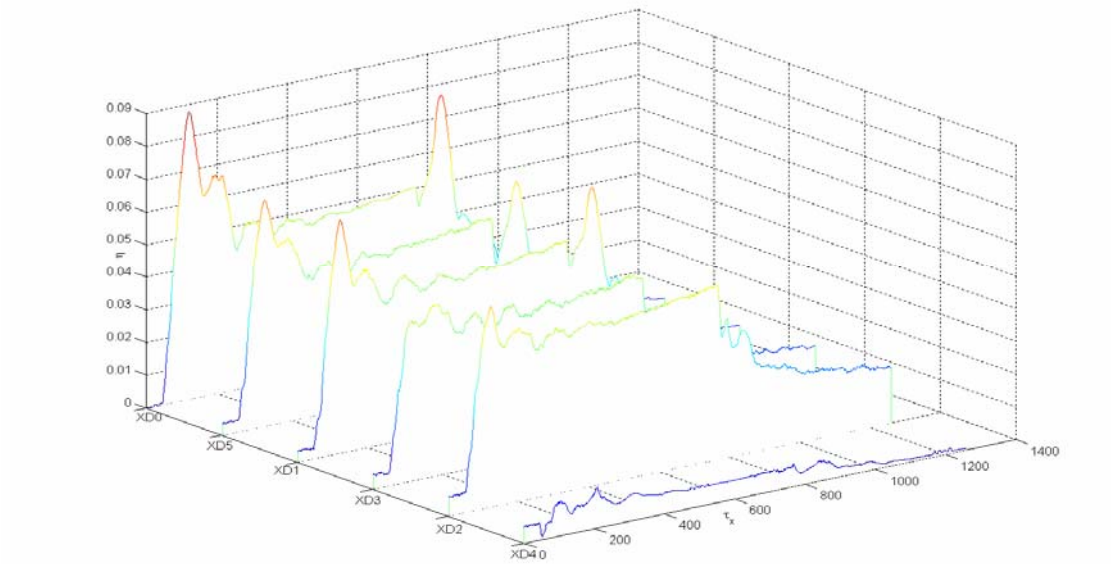
(e)


0.06
0.05
0.04
0.03
η
0.02
0.01
0
XD0
XD5
XD4
XD1
XD2
XD3
0
200
400
600
800
1000
1200
1400
τx

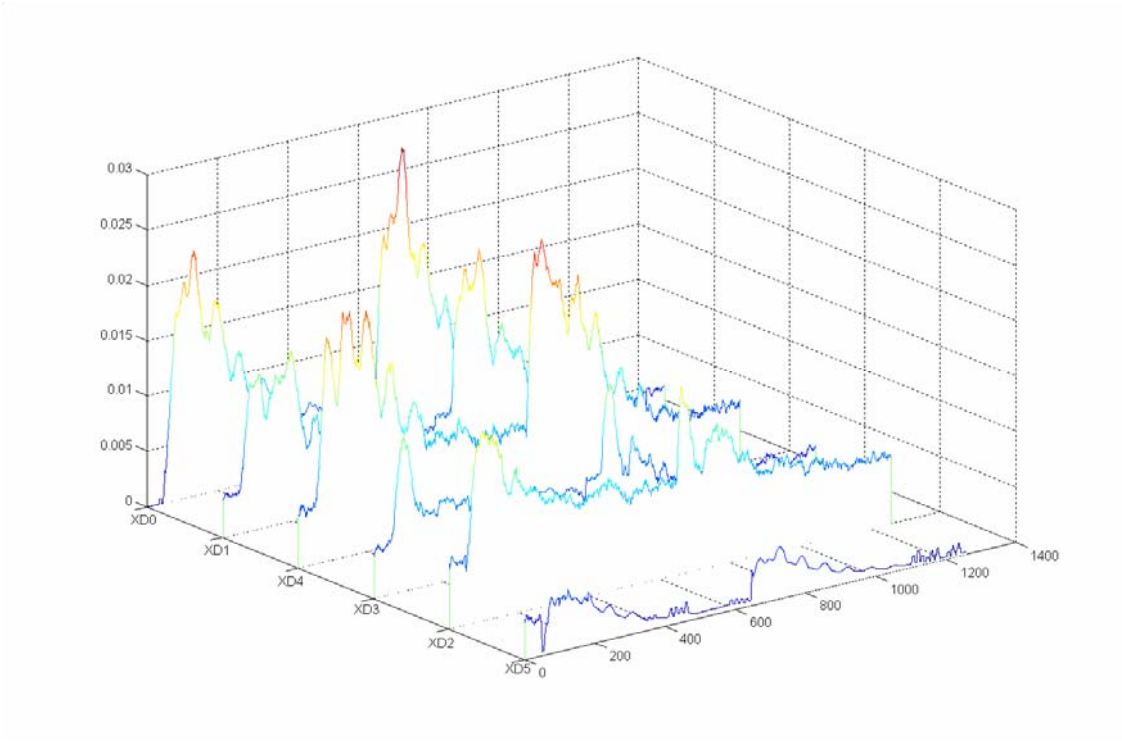
(f)


0.12
0.1
0.08
0.06
0.04
0.02
0
XD0
XD5
XD1
XD3
XD2
XD4
0
200
400
600
800
1000
1200
1400

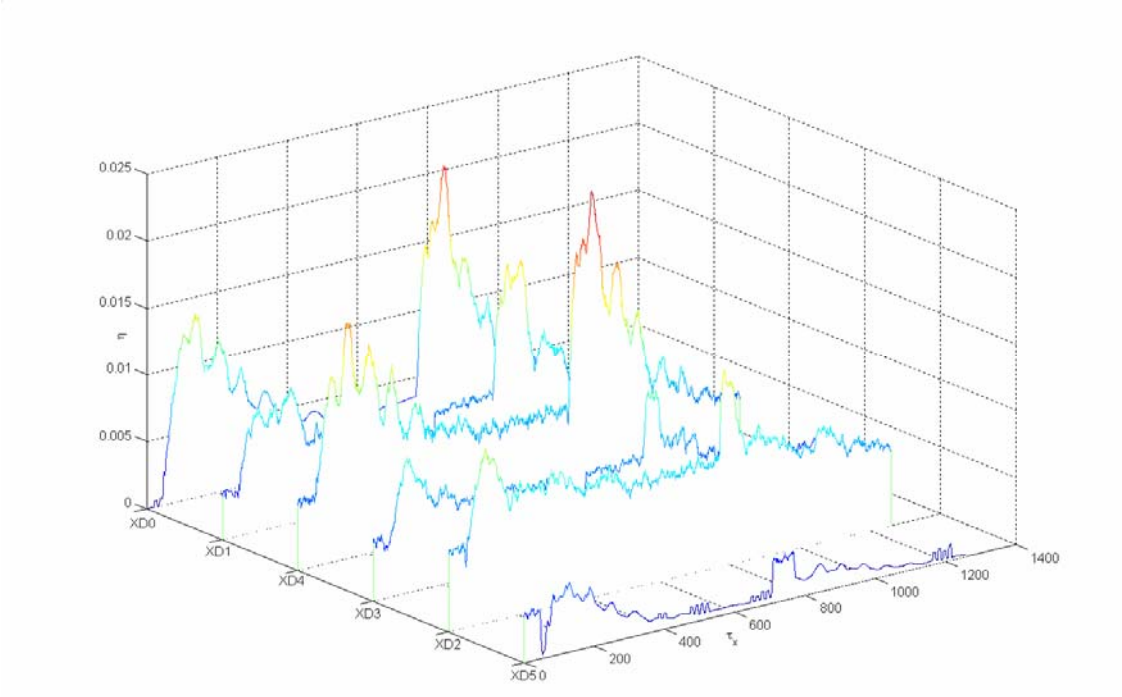
(g)


0.09
0.08
0.07
0.06
0.05
η
0.04
0.03
0.02
0.01
0
XD0
XD5
XD1
XD3
XD2
XD4
0
200
400
600
800
1000
1200
1400
τx

(h)



(i)



(j)

Figure 4-9: Run-1 to Run-10, online comparisons of real-time signal with all reference signals from τ_x = 8 to 1270 samples

Table 4-4: Online process disturbance detection in TE process

	η at time of identification						α	Best-matching reference	τ_y (sample)	τ_x (sample)	Delay (sample)	Time Cost (s)
	XD1	**XD2**	**XD3**	**XD4**	**XD5**	**XD0**						
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

4.3.2 Case Study 2: Fault Diagnosis during Startup of a Lab-scale Distillation Column

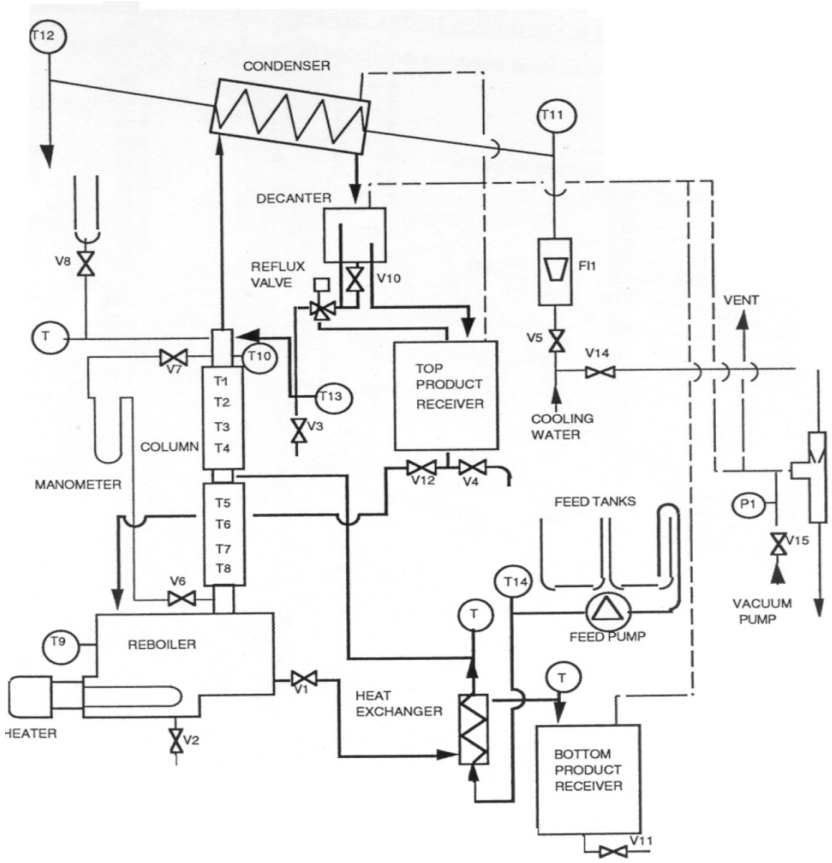


Figure 4-10: Schematic of the distillation unit set up

In this section, the proposed methodology is illustrated on a lab-scale distillation unit. The schematic of the unit is shown in Figure 4-10. The distillation column is of 2 meters 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. 19 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-second intervals. Cold startup of the distillation column with ethanol-water 30% v/v mixture is performed following the standard operating procedure shown in Table 4-5. The feed passes through a heat exchanger before being fed to the column. The startup normally

takes two hours 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 eleven runs of the process – one normal startup and the ten faults summarized in Table 4-6. For example, a higher than normal pumping rate was induced when developing the reference for DST03. This causes instability in the column and there is a drastic drop in the column temperatures as can be seen in Figure 4-11.

Table 4-5: 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

Table 4-6: 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	Low cooling water flow	Step
DST10	Feed pump malfunction	Step

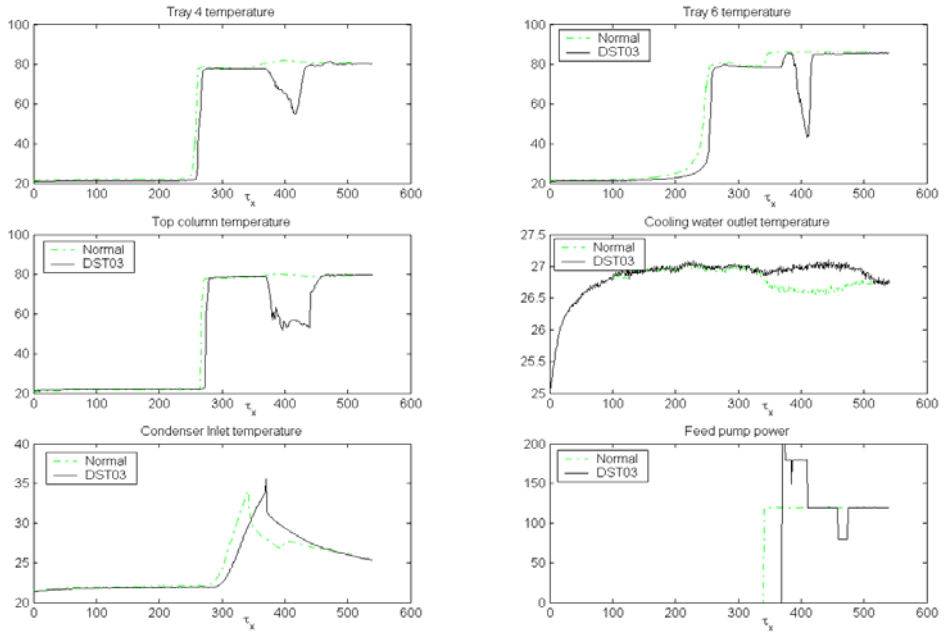
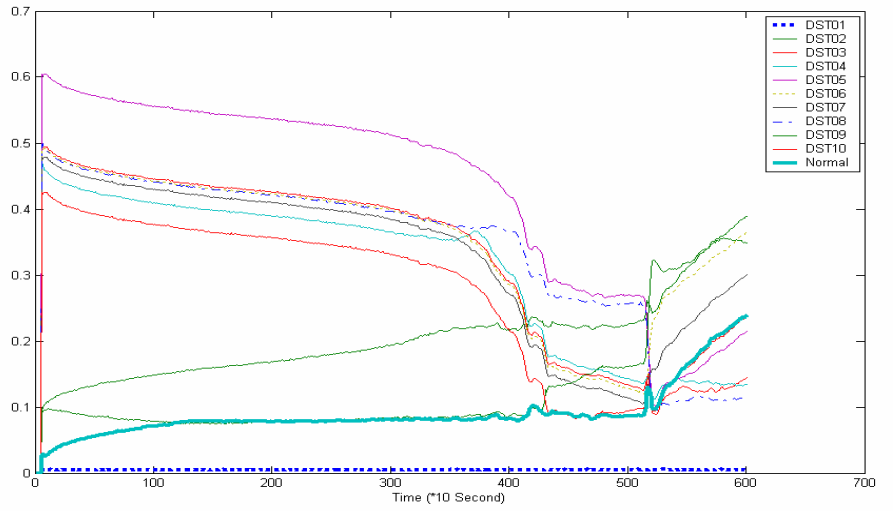
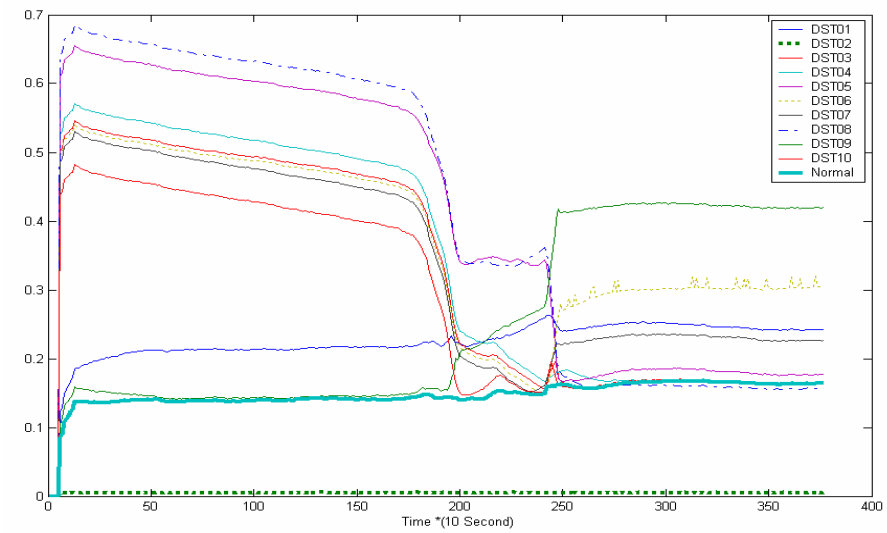


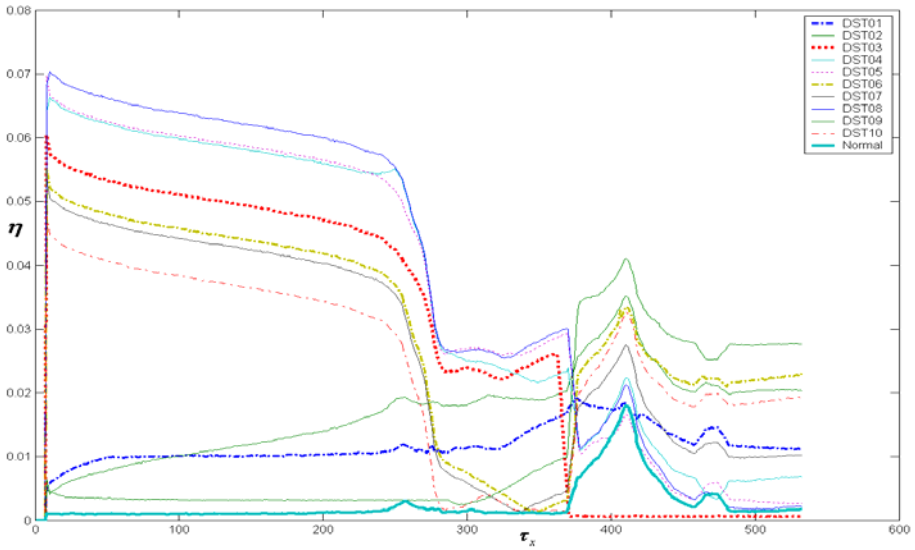
Figure 4-11: Process signals for Run-03 of lab-scale distillation column



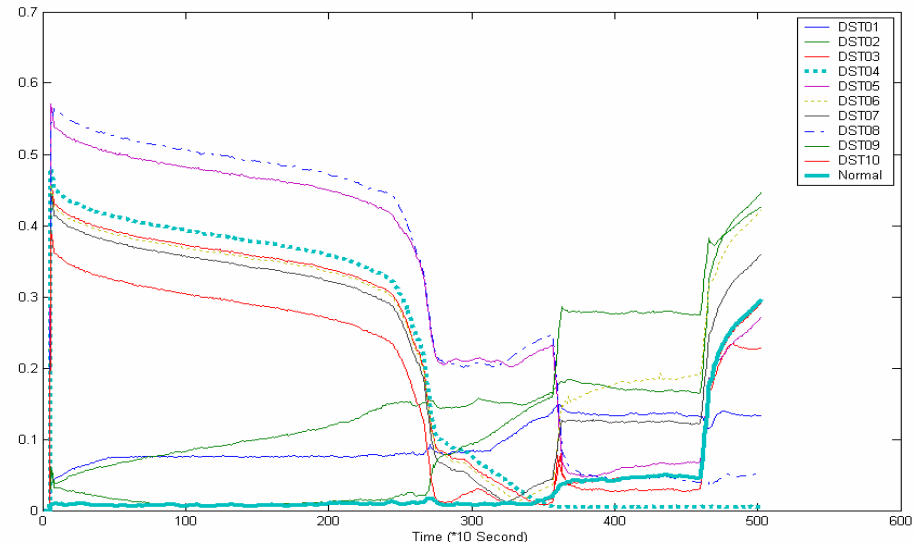
(a)



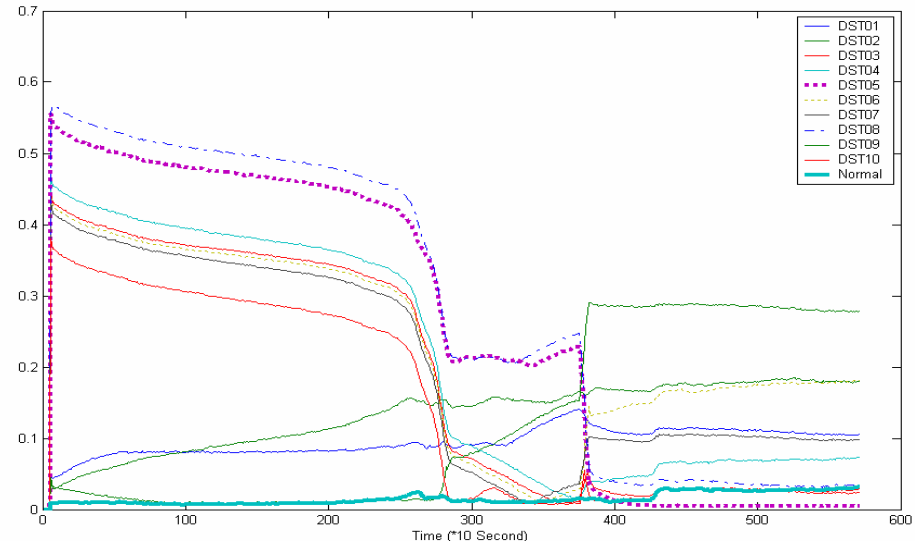
(b)


DST01
DST02
DST03
DST04
DST05
DST06
DST07
DST08
DST09
DST10
Normal
η
τ_x

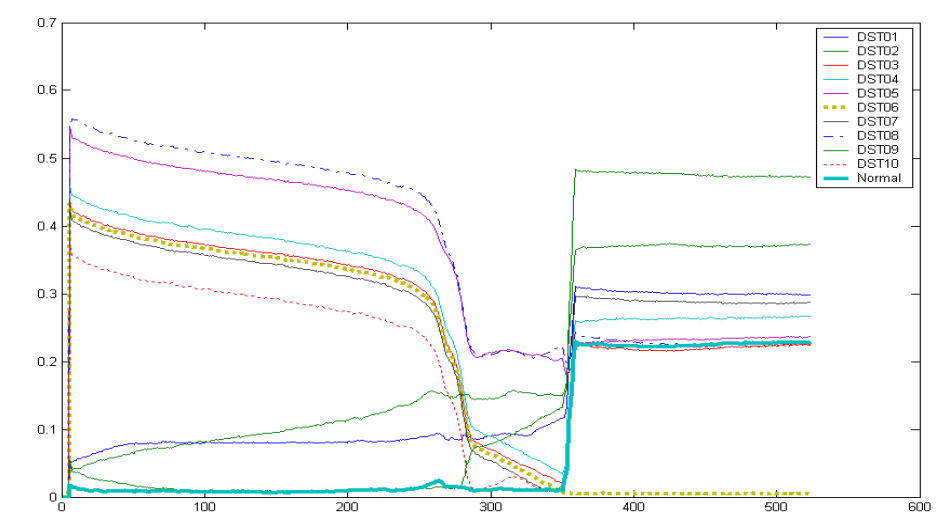
(c)


DST01
DST02
DST03
DST04
DST05
DST06
DST07
DST08
DST09
DST10
Normal
Time (*10 Second)

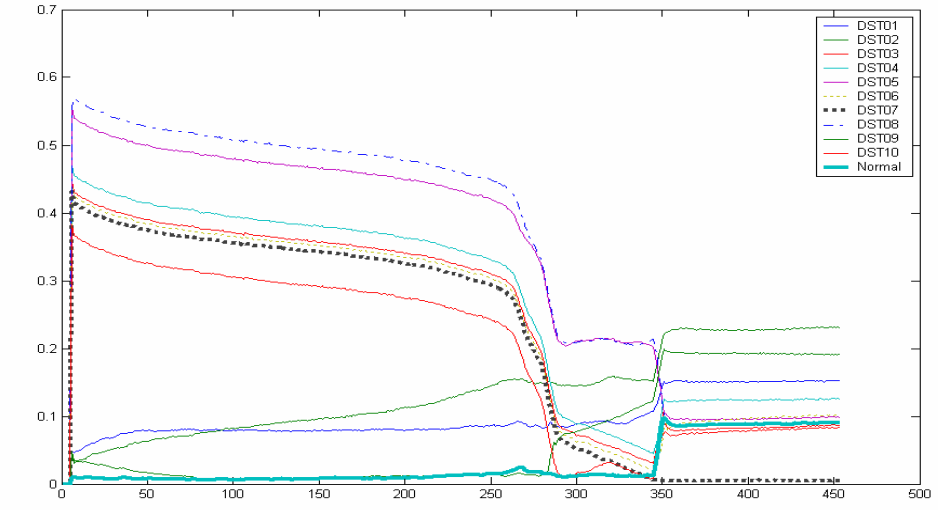
(d)


DST01
DST02
DST03
DST04
DST05
DST06
DST07
DST08
DST09
DST10
Normal
Time (*10 Second)

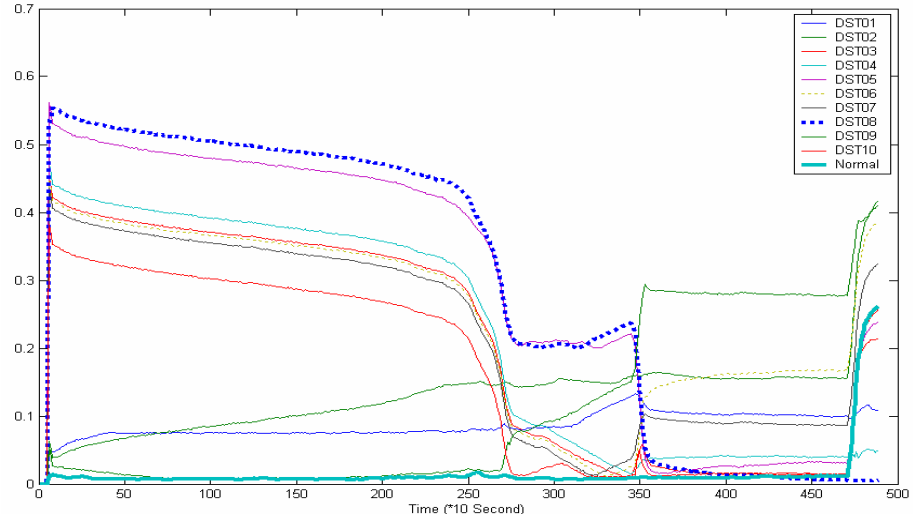
(e)


DST01
DST02
DST03
DST04
DST05
DST06
DST07
DST08
DST09
DST10
Normal

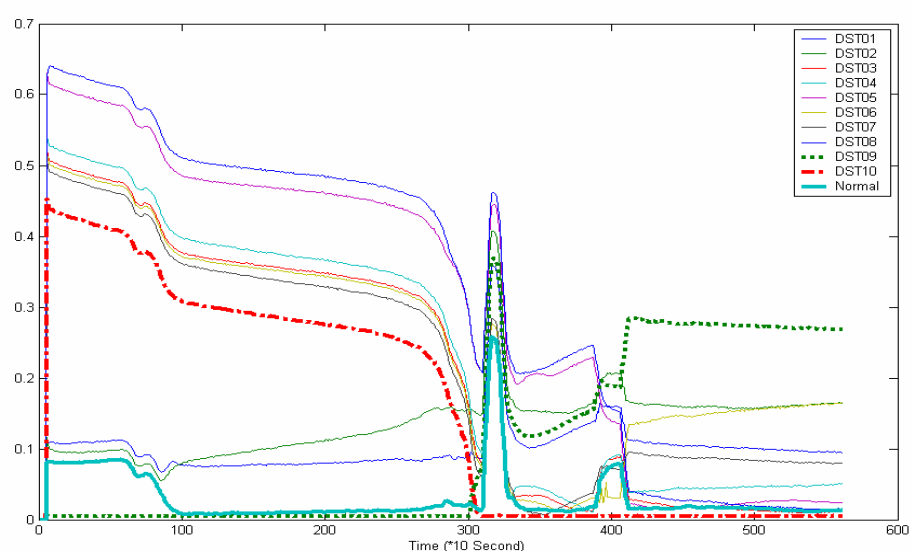
(f)


DST01
DST02
DST03
DST04
DST05
DST06
DST07
DST08
DST09
DST10
Normal

(g)


DST01
DST02
DST03
DST04
DST05
DST06
DST07
DST08
DST09
DST10
Normal
Time (*10 Second)

(h)



(i)

Figure 4-12: Normalized difference with all reference signals during Run-1 to Run-10 of lab-scale distillation column

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 is shown in Figure 4-12 (c) which shows the normalized difference between the real-time signal and the eleven references in the database throughout the startup. As can be seen there, the real-time signal is close to normal till about 3700s. The difference between real-time signal and all other references is much higher. Starting around t=3700s, 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=3710s, α falls below $\alpha_{\min}$ and identifies the fault as of class DST03. The accuracy of the identification is evident from Figure 4-13 which tracks τ_x over time and shows that the evolution of the current state continuously matches DST-03. The slight fluctuations around t= 5000s are due to measurement noise and run-to-run differences.

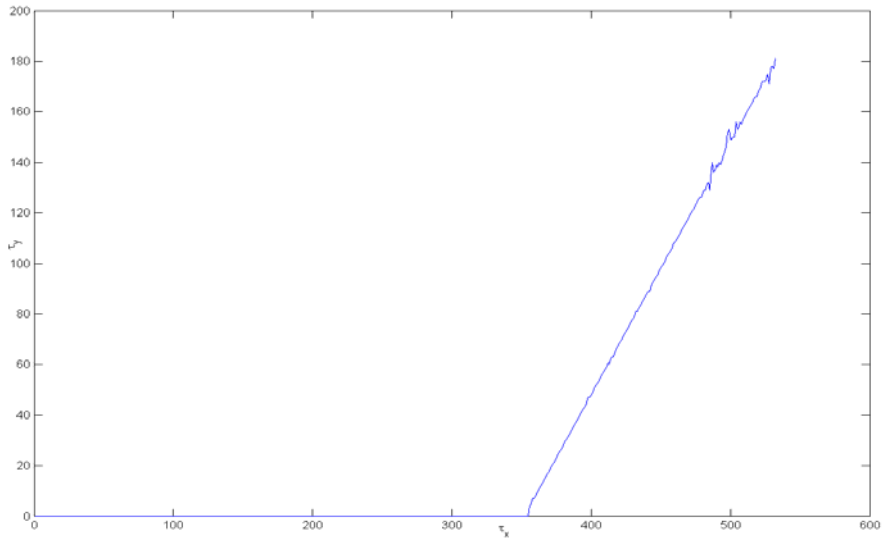


Figure 4-13: Time evolution of progression of fault between t = 0 to 550 samples during Run-3 of lab-scale distillation column

Similar tests were done for all other cases. The normalized difference between the real-time signal and the eleven references in the database throughout the startup is shown in Figure 4-12 (a, Reboiler power low, b, Reboiler power high, c, Feed pump high, d, Feed pump low, e, Tray temperature sensor fault, f, Reflux ratio high, g, reflux ratio low, h, bottom valve sticking , i, Low cooling water flow). A summary of the findings is presented in Table 4-7. All faults could be accurately identified within an average of 43 seconds of their occurrence (about 4 samples). The maximum identification delay was about 120 seconds (12 samples). The average time cost at each sample was only 0.468 second. The evolution of the faults was also identified clearly with an average incoherence of 1.55.

Table 4-7: 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)	Incoherence	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**

4.3.3 Case Study 3: Process State Identification in Simulator 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 ShadowPlant. ShadowPlant has the main function of a real FCC unit. It has more than four hundreds variable to description the process state change during process operating. We also use the data from ShadowPlant as a complex process signal for testing the method developed in this thesis. In this thesis, we focus on how to identify the real state of the process fast.

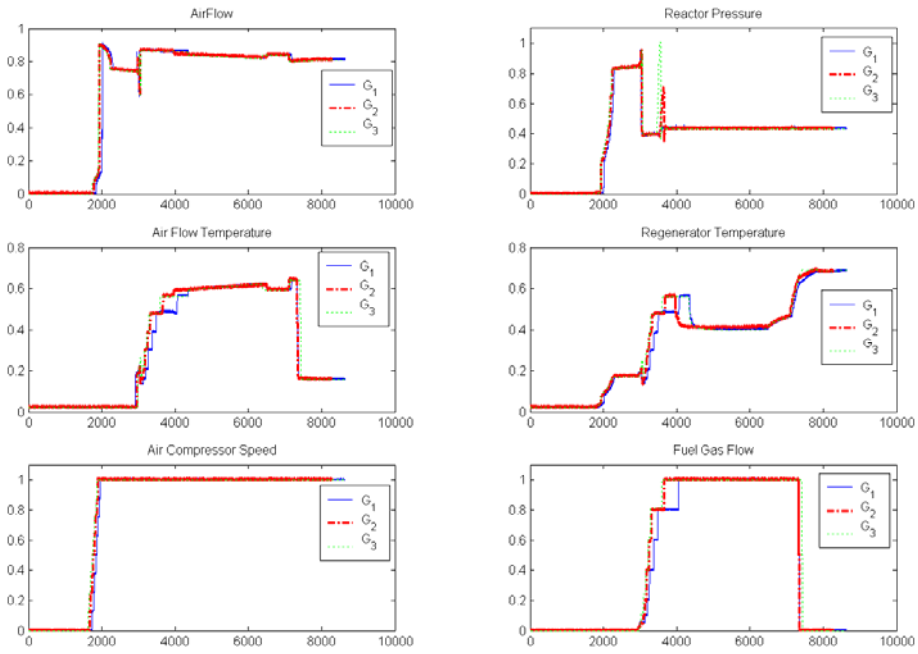


Figure 4-14: Real-time signal and reference signal of ShadowPlant

Several runs of the startup were performed following the standard operating procedure and data for all 335 measured variables collected at 10 second 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_1, G_2 and G_3 – with average startup duration of about 23 hours are considered in detail here. Data from G_3 was used as the reference signals while data from G_1 and G_2 are used for testing the dynamic locus analysis method. Using prior knowledge, the nine main sub-states listed in Table 4-8 and shown in Figure 4-14 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 sub-state. Dynamic locus analysis was used to identify the progression of the transition in real-time. Results for online sub-state identification are given in Table 4-8. In all cases, the sub-states were correctly identified within an average delay of 1.7 samples (or 17 seconds) since the inception of the sub-state. This compares favorably with other approaches such as the trend-based state identification reported by Srinivasan (2005), 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 lose of information and process operations like opening of valves and setpoint changes that are direct indicators of specific sub-states can be easily and accurately identified even in the face of noise and run-to-run variations. Despite the large-scale of the case study (335 online measurements) the average time cost for each sample was only 6.6 seconds.

Table 4-8: 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 (x10 s)	Occurrence Time (x10 s)	Detection Time (x10 s)	Detection Delay (x10 s)	Occurrence Time (x10 s)	Detection Time (x10 s)	Detection Delay (x10 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**

4.3.4 Robustness to Tuning Parameters

4.3.4.1 Effect of Parameters on Dynamic locus analysis

The performance of dynamic locus analysis is affected by two main parameters – the evaluation window *m* and the minimum inseparability threshold $\alpha_{\min}$. While in general, different process signals may require different values for these parameters, we have found that most of the parameter settings can be used across variables and case studies. *m* is around 6 to 12 which can catch the signal feature and reduce the noise effect. $\alpha_{\min}$ is around 0.50 to 0.80. In this thesis, $\alpha_{\min}$ was set as 0.70 for all the case. We test the parametric sensitivity of dynamic locus analysis in this section.

Evaluation Window: To test the effect of *m*, we repeated the above ten test runs with various *m* values as shown in Table 4-9 and Table 4-10. In all cases the proposed method identified the correct disturbance. The results shown in Table 4-9, 4-10 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 for larger values as shown in Figure 4-15, 4-16, 4-17. 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 is high. We have found that, in all cases, *m* = 8

samples provides an acceptable detection delay, low computational cost, as well as adequate coherence.

Evaluation window size is a trade-off between sensitivity and robustness. In the Lab-scale distillation column case, the incoherence is always less than 6 when $m = 8$. In ShadowPlant case study, there is an obvious drop in incoherence with $m = 20$ which may be due to slow evolution of the process.

Table 4-9: Effect of evaluation window on identification delay and time cost in TE case study

	Identification Delay (samples)						Time Cost (second)					
m	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

Table 4-10: Effect of evaluation window on identification delay and time cost in distillation column startup case study

	Identification Delay (samples)						Time Cost (second)					
m	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

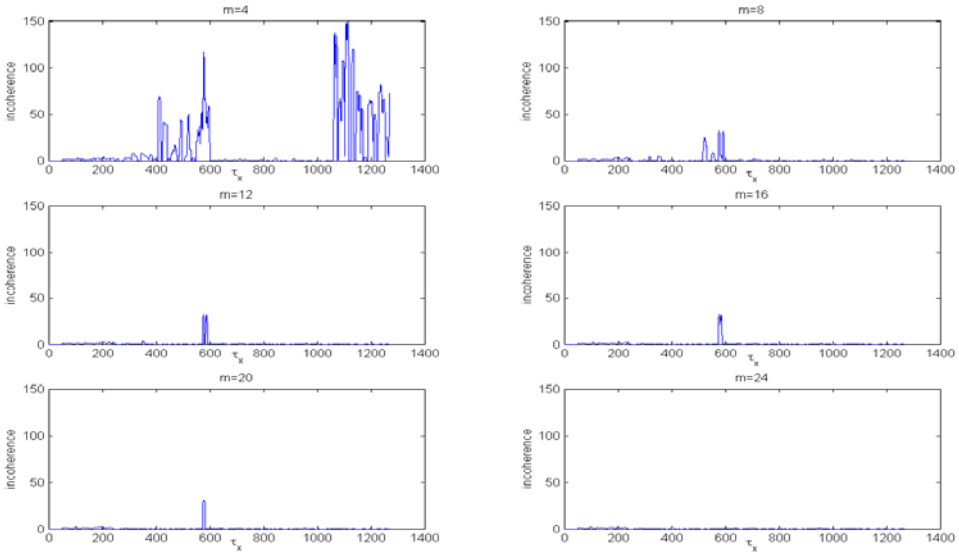


Figure 4-15: Effect of evaluation window on incoherence metric during Run-3 of TE process

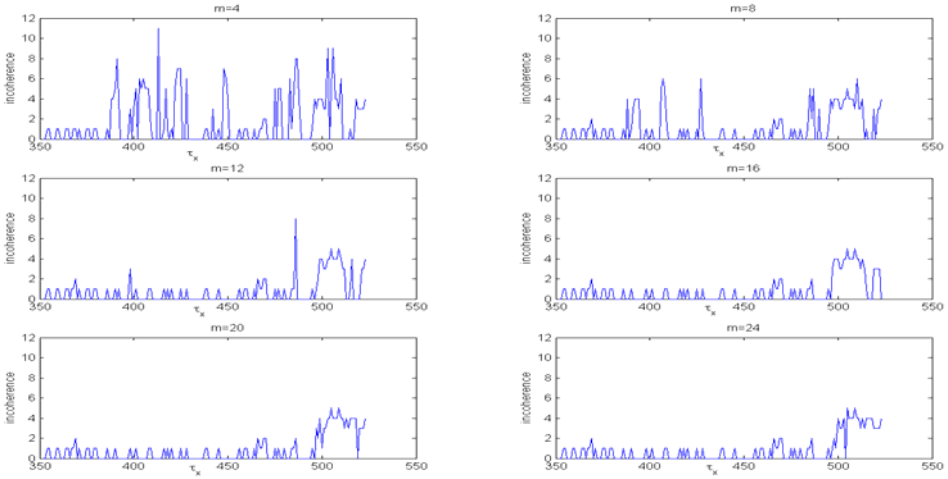


Figure 4-16: Effect of evaluation window on coherence metric during Run-06 of Lab-scale Distillation Column

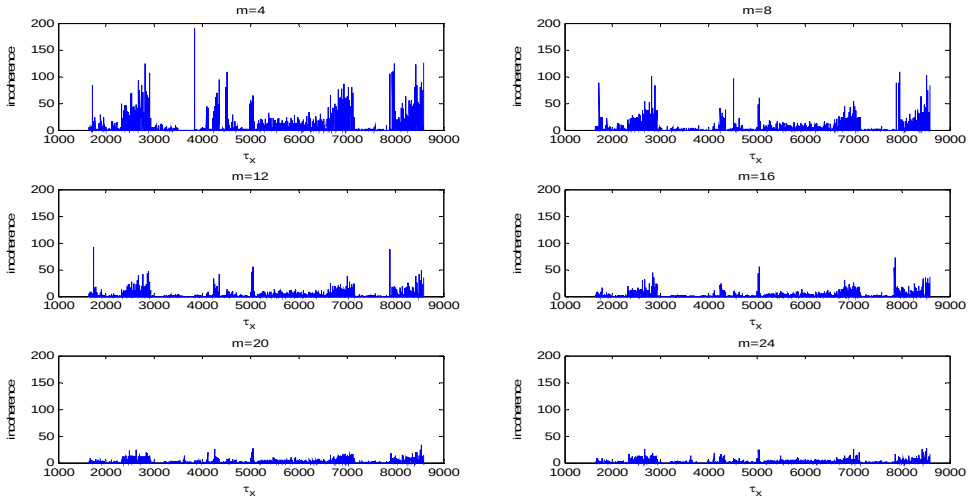


Figure 4-17: Effect of evaluation window on coherence metric of ShadowPlant

Minimum Separability Threshold: α_{min} affects the detection delay as shown in Figure 4-18. Increasing α_{min} decreases the detection delay but at the cost of accepting a smaller difference between the best matching and the next-best matching reference signals as the basis for the result. We have found α_{min} = 0.75 as a suitable threshold for all case studies.

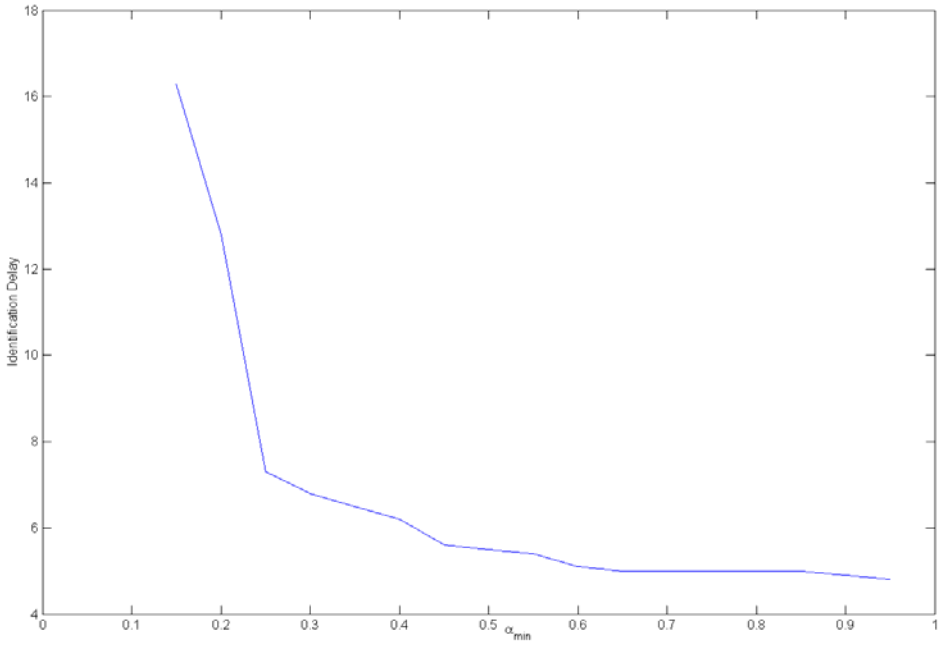


Figure 4-18: Effect of minimum inseparability threshold α_{min} on identification delay in TE process

4.3.4.2 Robustness of Noise

Dynamic locus analysis method is a noise robust method. The robustness of the proposed method to sensor noise is studied using data from TE plant. Extra measurement noise of 2%, 4%, 6%, 8% and 10% were added to the original signals. The results are shown in Table 4-11. There is some variation in the stage identification at different noise levels and the identification delay increases with noise. Even at 10% noise level, the average identification delay is still only 7.7 sample points which is only 2.7 point higher than the no noise case. Overall, the online signal comparison method is robust to noise and there is no significant affect on the result.

Table 4-11: Robustness of noise in TE disturbances identification

	Identification Delay (samples)					
Noise	0%	2%	4%	6%	8%	10%
XD1-A	5	5	5	6	6	6
XD1-C	5	5	6	6	6	6
XD2-A	5	5	5	6	6	6
XD2-C	5	6	6	6	7	12
XD3-A	5	5	5	6	6	10
XD3-C	5	5	6	6	6	8
XD4-A	5	5	6	7	7	9
XD4-C	5	5	6	6	7	8
XD5-A	5	5	5	6	6	6
XD5-C	5	5	5	6	6	6
Average	5	5.1	5.5	6.1	6.3	7.7

4.4 Discussion

Dynamic locus analysis is an extension of Smith and Waterman's (1981) 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. With dynamic locus analysis, all potential matching segments are compared for find the optimal results. 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.

In the next chapter, dynamic locus analysis is used for identifying the corresponding point between real time signal and reference signal.

Chapter 5 Online Signal Comparison Using Singular Points Augmented Time Warping

5.1 Introduction

One of the main challenges for monitoring process is to identify process signal online. Online signal identification is different from offline signal identification in that data from the process is obtained dynamically. So the state of the process has to be identified without full information. The earlier we wish to decide on the signal, the lesser the information to base it on. Each process change has its own characteristic features as revealed by the variables evolutions. Abnormal situations such as maloperation, equipment failure, etc, lead to changes in these features. In this chapter, we use signal comparison technique to identify the process state from such changes. Dynamic locus analysis is used to identify the reference signal and the base point that best fits the real-time signal. Singular point augmented DTW is then used to extend the match when new sensor samples are obtained in real-time. We can use this information to guide plant operators or engineers during abnormal situations and help them in solving process problems. The outline of this chapter is as follows: The methodology for online signal comparison using Singular points augmented Time Warping is discussed in Section 5.2. In Section 5.3, the application of the method to TE case and ShadowPlant data is discussed. Summary and conclusions from this work are presented in Section 5.4.

5.2 Online Signal Comparison Using Singular points Augmented Time Warping

The singular points based time warping approaches enforce the optimal linkage of the major landmarks of the two signals using dynamic programming. The global optimality of the episode-level comparison is therefore not a critical requirement since the optimal assignment of each time point within an episode has no physical significance and is rarely necessary in practical applications. The two-step comparison also leads to significant improvements in speed, memory requirement, and efficiency of signal comparison. Another important advantage is that since the singular points have physical meaning such as the beginning or ending of a process event, they can be directly used for state identification, monitoring, and supervision.

Singular point augmented time warping is suited for signals whose endpoints are known to correspond as these are used in the initial warping assignment (*1, 1*) and (*t, r*). This is not the case in real-time signal comparison where the online signal is available starting from an unknown state. We use dynamic locus analysis to identify the endpoints in the library signal that correspond to those of the real-time signal.

The online signal comparison problem can be stated as follows: Given a set of reference signals *K* and a real-time signal (*T*) emanating from the process operating at an unknown state, (1) identify the reference signal that best matches the current state of the process, and (2) identify the progress of the process with respect to the reference signal. The former is called the *optimal reference signal identification problem* while the latter is referred to as *real-time signal (or state) tracking*. The first step involves comparison of the real-time signal with many reference signals and is computationally more intensive than the 2nd step; hence although the first step could be repeated at every instant, it is not tenable for online application, where the requirement is that the

calculation time at every sample must be less than the sampling interval $\Delta\tau$. The proposed online signal comparison approach therefore addresses the two sub-problems separately, as described next.

5.2.1 Real- time Signal Tracking using Singular Points Augmented Time Warping

This step uses the optimal reference signal R^* calculated a priori (See Section 5.2.2) and seeks to confirm that the process continues to operate in the same state (i.e., same reference signal). It thus calls only for resynchronization based on the additional real-time sample with R^*. In the following, for ease of explanation, the time variable for the real-time signal is denoted as τ_T and that of reference signal R as τ_R. The two signals can be compared starting from the beginning (i.e, $\tau_T = 1$); alternatively a smaller evaluation window can be used. We pursue the latter option here for computational efficiency purposes.

Consider the real-time signal $T = \{t_1, t_2, t_3, ..., t_{m-1}\}$ with time index i. Our evaluation window is defined based on singular points. Let the last singular point in T be at time $\tau_T = \tau_T^{SP}$ and let the value of T at that time be $t_{\tau_T^{SP}}$. The corresponding singular point in R^* can be obtained using XTW^{SP}. The pair of corresponding singular points in T and R^* is called the *anchor* of the evaluation window. The evaluation window T_w extends from the anchor point mark to t_{m-1} at time $\tau_T = m-1$, i.e., $T_w = \{t_{\tau_T^{SP}}, t_{\tau_T^{SP}+1}, ..., t_{m-1}\}$. The *base point* at $\tau_T = m-1$ is the point in the reference signal R^* that corresponds to t_{m-1} in T. The corresponding segment of R^* which matches with T_w is denoted as R_w^*. When a new sample t_m becomes available at $\tau_T = m$, i.e., T_w is updated, the task in real-time signal tracking is to update the base point and R_w^*

through resynchronization and confirm that R_w^* continues to match T_w. In our approach this is achieved in two steps as follows:

Step A: Efficient calculation of difference between real-time signal and reference signal

Any signal comparison method can be used for calculating the difference between T_w and R_w^*. We use XTW for this purpose because of its advantageous time and space requirements. The normalized time-warped distance between T_w and R_w^* is calculated as follows:

$$\tilde{\eta}(T_w, R_w^*) = \frac{\sum_{i=\tau_T^{SP}}^{m} \left| r_{j(i)} - t_i \right|}{m - \tau_T^{SP} + 1} \tag{5-1}$$

The numerator is the XTW difference between T_w and R_w^* while the denominator is the length of the evaluation window. The difference between the latest real-time sample t_m and its warping assignment in R^*, notated as $\tilde{\Delta}(t_m, r_{\tau_{R^*}})$, is also calculated. If

$$\left(\tilde{\eta}(T_w, R_w^*) < \eta_{\max}\right) \text{ and } \left(\tilde{\Delta}(t_m, r_{\tau_{R^*}}) < 2 \times \eta_{\max}\right) \tag{5-2}$$

is satisfied, the process is considered to continue in the same stage of operation, and only the base point τ_{R^*} is updated. The first condition ensures that the broad overall trend of the reference and real-time signals is the same while the second condition allows for larger local variations in the signal due to noise or run-to-run differences. The computational efficiency of XTW comes at a price, a large $\tilde{\eta}(T_w, R_w^*)$ does not guarantee that T_w is dissimilar from R_w^* since the local greedy search in XTW can lead to an overestimate of the difference when signals become long[1]. Such

artifacts can be eliminated by a more accurate calculation using XTW^{SP}, when necessary.

Step B: Accurate calculation of difference between real-time signal and reference signal

XTW^{SP} links the singular points in the real-time and reference signals and thus calculates a accurate difference between T_w and R_w^*. So if condition (5-2) is not satisfied, a more accurate difference, notated as $\eta(T_w, R_w^*)$ and $\Delta(t_m, r_{\tau_{R^*}})$ are calculated using XTW^{SP} and a condition analogous to (5-2) is evaluated.

$$\left(\eta(T_w, R_w^*) < \eta_{\max}\right) \text{ and } \left(\Delta(t_m, r_{\tau_{R^*}}) < 2 \times \eta_{\max}\right) \tag{5-3}$$

If this is satisfied, the process is considered to continue in the same state, and the base point is updated. Otherwise a new reference signal has to be identified, starting at $\tau_T = m$ as the point of divergence τ_T^{POD} (See Section 5.2.2).

Step C: Updating the Anchor

One byproduct of performing XTW^{SP} in the previous step is that corresponding singular points between T_w and R_w^* are identified. If the process continues to be in the same state (as per (5-3)), the last corresponding singular points are henceforth used as the anchor. This results in the shortening of the evaluation window**,** so future Step A and Step B calculations become more efficient and accurate. If (5-3) is not satisfied, the process is considered to have moved to a new state and another optimal reference signal has to be identified. Algorithm for real-time signal tracking is shown in Figure 5-1.

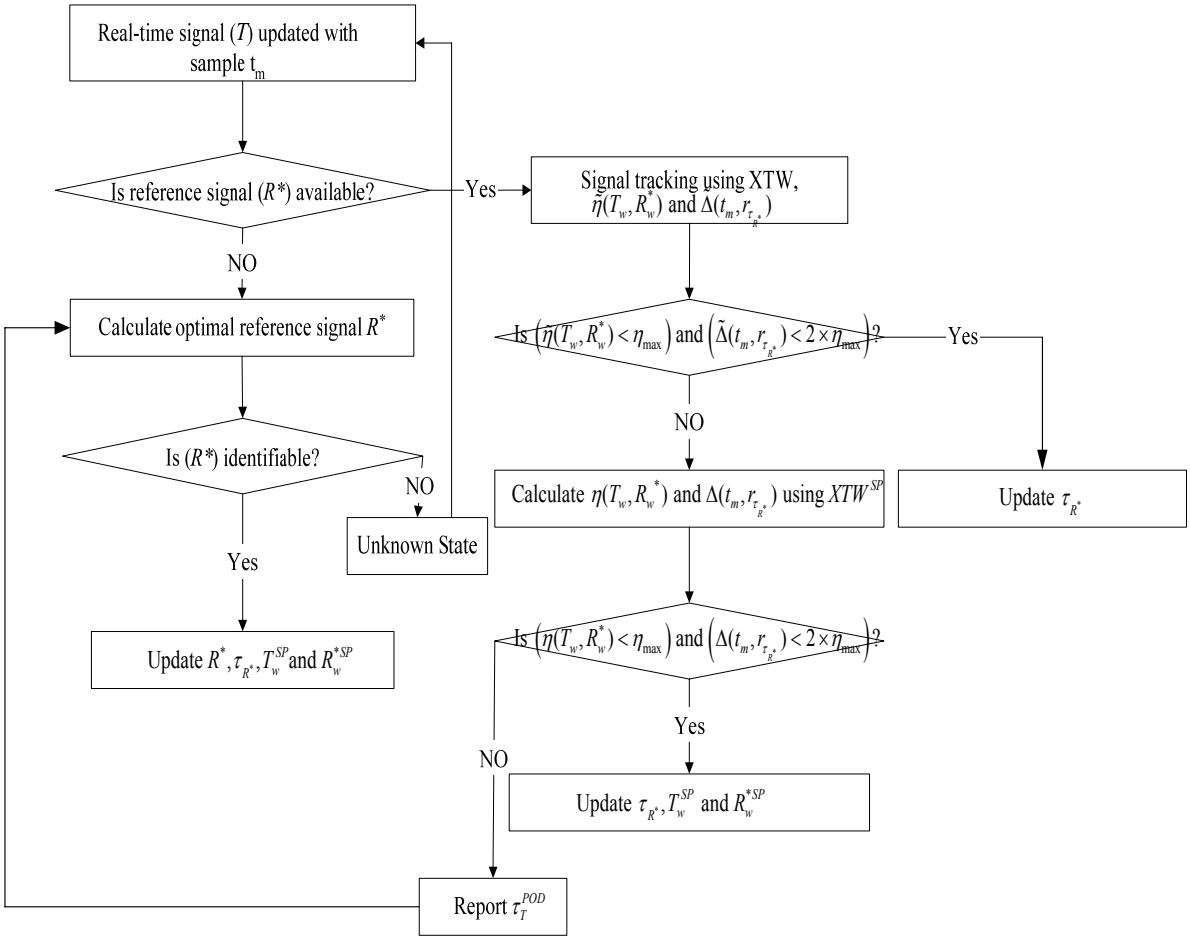


Figure 5-1: Algorithm for real-time signal tracking

5.2.2 Optimal Reference Signal Identification using Flanking Strategy

The task in this stage is to identify the reference signal R^* that best matches the state of the process at time t_m. This would be required at $\tau_T = 1$ when the reference signal is not known and when the real-time signal has diverged from the previous reference (i.e., Equation (5-3) is not satisfied). Consider a *divergent segment* of T ranging from the point of divergence to the latest sample:

$$T_D = \begin{cases} \{t_{\tau_T^{POD}}, \ldots, t_m\} & \text{if } m \geq \tau_T^{POD} + \nu \\ \{t_{m-\nu+1}, \ldots, t_m\} & \text{if } \nu \leq m < \tau_T^{POD} + \nu \end{cases} \tag{5-4}$$

Note that $\tau_T^{POD} = 1$ at $\tau_T = 1$. The process state is identified by comparing the divergent segment with all the reference signals in the library (*K*). Consider a reference signal $R = \{r_1, r_2, r_3, ..., r_n\}$ from *K* with time index *j*. Let $\eta(T_D, R)$ be the normalized difference between the T_D and *R*. The optimal reference signal R^* is defined as:

$$R^* = \arg\min_{R \in K}(\eta(T_D, R)) \tag{5-5}$$

A tradeoff exists between the speed and accuracy of reference signal identification. If the divergent segment used for identifying R^* is small, several good matches may exist in *K*. We quantify the accuracy of the located optimal reference signal in terms of the *inseparability ratio* α, defined as the ratio of the normalized difference of the best matching reference signal to that of the second-best one:

$$\alpha = \frac{\eta(T_D, R^*)}{\min_{R \in K, R \neq R^*}(\eta(T_D, R))} \tag{5-6}$$

The inseparability ratio thus reflects the uniqueness of R^*. The criterion for the optimal reference signal at $\tau_T = m$ with base point τ_{R^*} is

$$\left(\eta(T_D, R^*) < \eta_{\max}\right) \text{ and } \left(\alpha < \alpha_{\max}\right) \tag{5-7}$$

where the first condition ensures low difference between T_D and R^* while the latter ensures its uniqueness. If (5-7) is not satisfied at $\tau_T = m$, an unknown state is flagged and the calculation of R^* and τ_{R^*} repeated when the next real-time sample becomes available at $\tau_T = m + 1$.

Algorithm for optimal reference signal R^* identification is shown in Figure 5-2. Next, we describe how the difference $\eta(T_D, R)$ is calculated. Since the suitable start and endpoints of T_D in *R* are not known *a priori*, the locus of T_D in *R* has to be first calculated. This can be directly performed using DLA if the divergent segment is small

as is the case immediately after the point of divergence. We term such comparisons as *en bloc*.

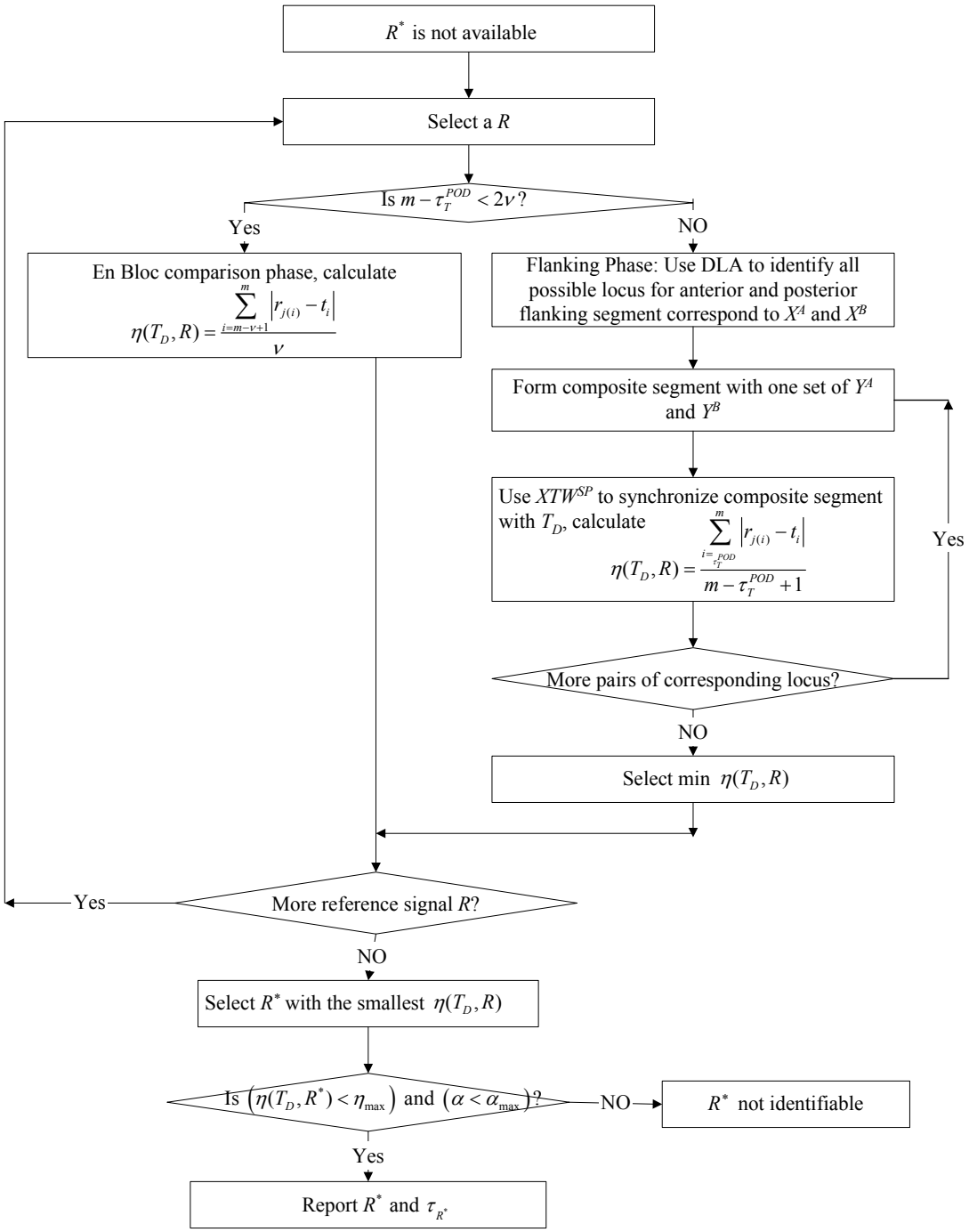


Figure 5-2: Algorithm for optimal reference signal R^* identification

En Bloc Difference Calculation

The DLA strategy is best suited for small signal lengths (See Section 4.2). When $m-\tau_T^{POD} < 2\nu$, the locus of the last ν points on the divergent segment is identified for each *R* directly using DLA. The normalized difference is calculated as:

$$\eta(T_D, R) = \frac{\sum_{i=m-\nu+1}^{m} \left| r_{j(i)} - t_i \right|}{\nu} \tag{5-8}$$

where (i, j) is the warping assignment between $\{t_{m-\nu+1}, t_{m-\nu+2,}..., t_m\}$ and its locus in R. As mentioned above, if (5-7) is not satisfied and the optimal reference signal cannot be clearly determined, comparison is repeated when an additional real-time sample becomes available. Thus the length of the divergent segment would increase with time. Once the length of the divergent segment in T_D becomes large, DLA becomes computationally expensive. We minimize the computational complexity for such cases ($m-\tau_T^{POD} \geq 2\nu$) using a flanking-based comparison strategy.

Flanking Based Difference Calculation

In the flanking-based strategy, DLA is augmented with XTW^{SP} to calculate $\eta(T_D, R)$. Short segments from the start and end of T_D, called flanking segments provide the basis for identifying the locus of T_D in *R*. The *anterior flanking segment* X^A is defined as a segment of size *v* from the start of T_D, $X^A = \{t_{\tau_T^{POD}}, t_{\tau_T^{POD}+1}, ..., t_{\tau_T^{POD}+\nu-1}\}$. Similarly, the *posterior flanking segment* is a short segment from the end of T_D, $X^B = \{t_{m-\nu+1}, t_{m-\nu+2,}..., t_m\}$. T_D can be considered to be sandwiched between the anterior and posterior flanking segments. It is obvious that any segment of the reference signal *R* that is the locii of T_D (called the *composite segment*) should also have sub-segments that have high similarity with the anterior and posterior flanking

segments. The flanking strategy for difference calculation exploits this property. The Flanking segments used for signal comparison are shown in Figure 5-3 and Figure 5-4.

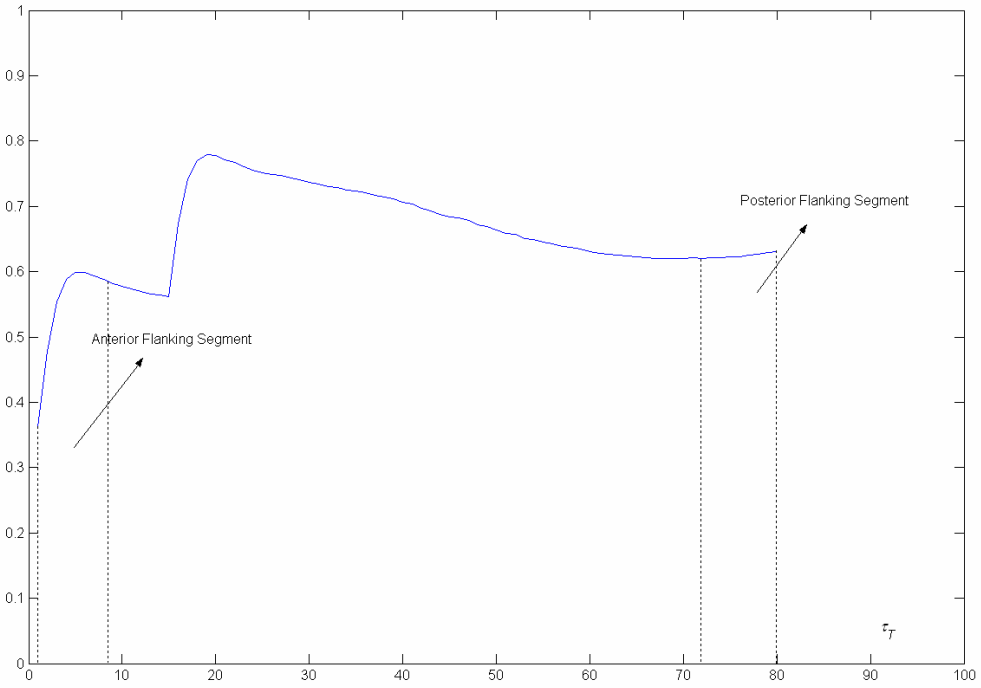


Figure 5-3: Flanking segments used for reference signal identification

DLA is used to identify all the matching segments Y^A in R such that $\eta\left(X^A, Y^A\right) < \eta_{\max}$. The same is also applied to obtain all possible reference signal segments Y^B that match the posterior flanking segment. Each pair of Y^A and Y^B defines a different composite segment in R, of which the one with the least difference with T_D is the locus. Since in general these composite segments can be long, XTW^{SP} is used to synchronize them with T_D. The difference between T_D and R is then calculated as follows:

$$\eta(T_D, R) = \frac{\sum_{i=\tau_T^{POD}}^{m} \left| r_{j(i)} - t_i \right|}{m - \tau_T^{POD} + 1} \tag{5-9}$$

A detailed illustration is given next to explain the above described online signal comparison algorithm.

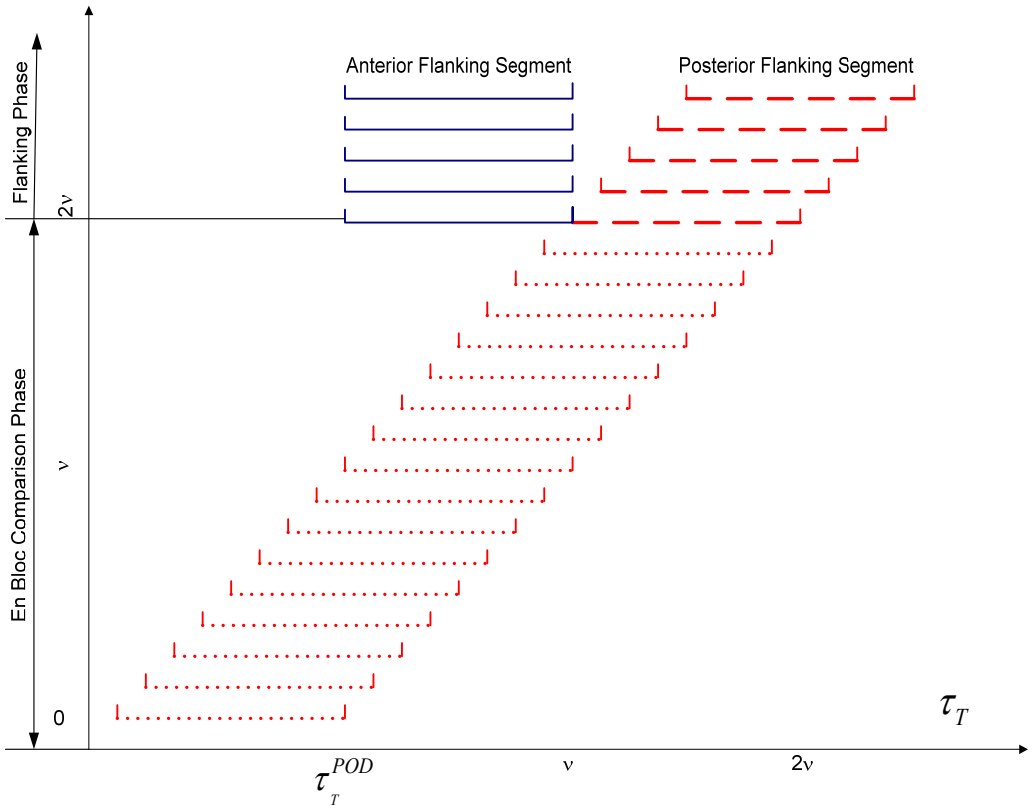


Figure 5-4: Temporal Development and Translation of Flanking segments for $\eta(T,R)$ calculation

5.2.3 Illustration: Online Computation the optimal match of the two signals

Consider the test signal T and the two reference signals R_1 and R_2 shown in Figure 5-5. Online data has been collected starting at $\tau_T = 1$. Since the optimal reference signal is unknown initially, it has to be identified first following the description in Section 5.2.2. $\tau_T^{POD} = 1$ and signal comparison starts at $\tau_T = 8\ (=\nu)$. The en bloc difference calculation strategy is first used for finding the locus in reference signals R_1 and R_2 since the signal length at $\tau_T = 8$ is less than 2ν. At $\tau_T = 8$, the DLA difference for R_1 and R_2 are as shown in Figure 5-6. $D_S(\nu, \tau_{R_1})$ has a clear minimum at $\tau_{R_1} = 199$ and $\eta(T,R_1) = 0.0151\ (< \eta_{\max} = 0.05)$. The locus for the divergent segment in R_2 is at $\tau_{R_2} = 1083$ and $\eta(T,R_2) = 0.0836 (> \eta_{\max})$. Therefore

$\alpha = 0.1810$ ($< \alpha_{\max} = 0.70$). So at $\tau_T = 8$, R_1 is confirmed to be the optimal reference signal with $\tau_{R_1} = 199$ as the base point. From the next sample, $\tau_T = 9$, real-time signal tracking as described in Section 5.2.1 is performed to confirm that *T* progresses as per reference signal R_1.

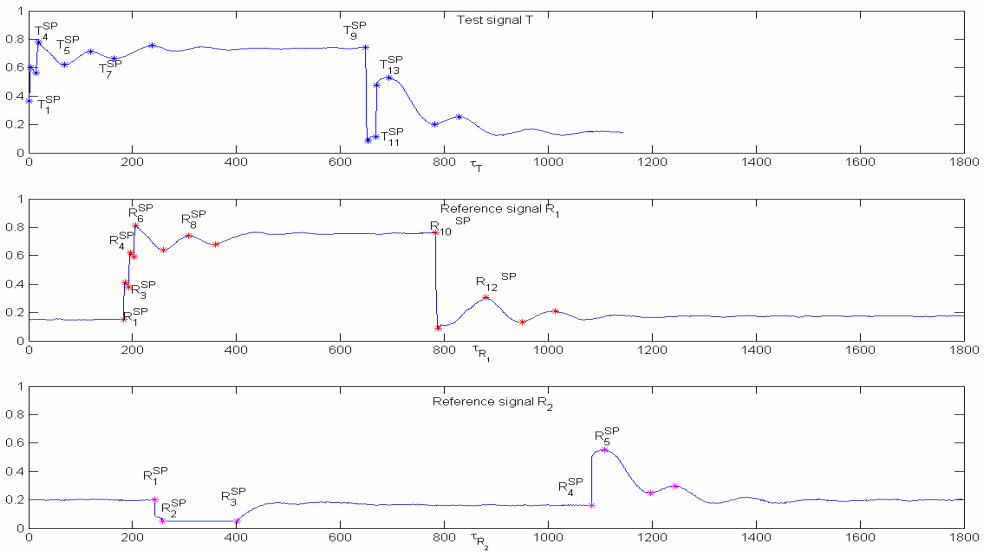


Figure 5-5: Test signal *T* and reference signals (R_1 and R_2) for illustrative example

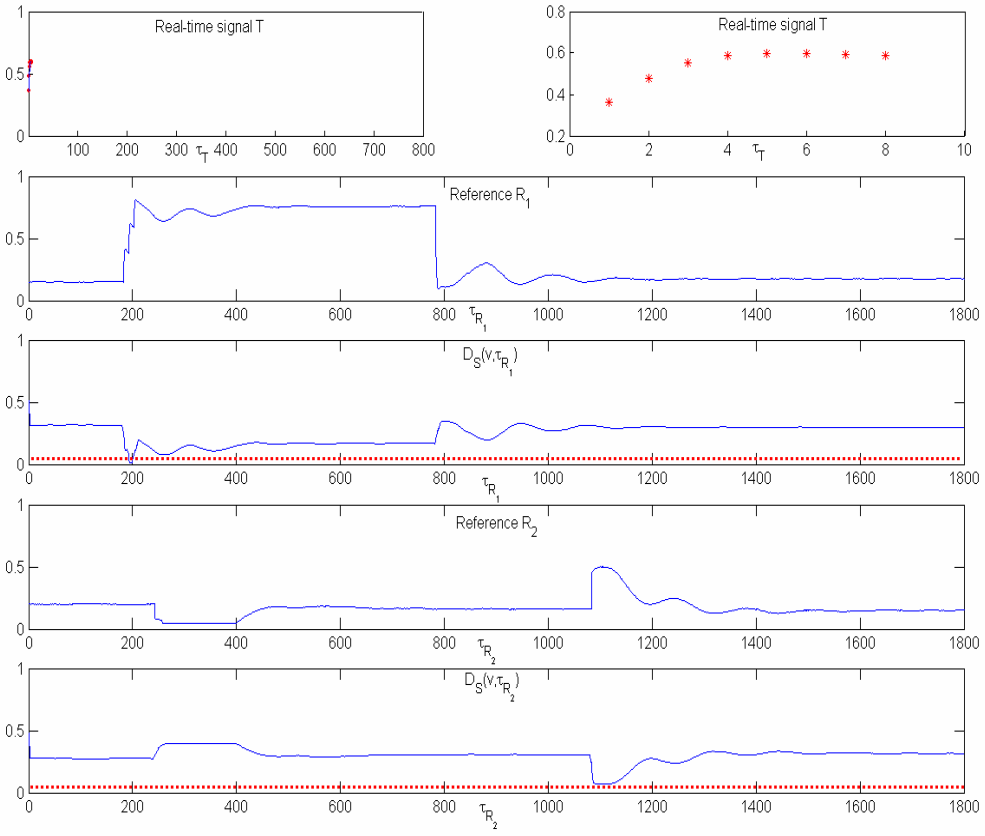


Figure 5-6**:** The comparison of real-time signal T at $\tau_T = 8$ with R_1 (shown in b) reveals a minima at $\tau_{R_1} = 199$ (shown in c). Similar comparison with R_2 depicted in (d) shows a minimum at $\tau_{R_2} = 1083$ as shown in (e)

A snapshot of the real-time signal tracking at $\tau_T = 226$ is shown in Figure 5-7. At this time, the anchor for comparison between T and R_1 is the previous corresponding singular points with $\tau_T^{SP} = 164$ and $\tau_{R_1}^{SP} = 359$. The base point is at $\tau_{R_1} = 413$. Therefore, the evaluation window $T_w = \{t_{164}, ..., t_{226}\}$ is compared with $R_w^* = \{r_{359}, ..., r_{413}\}$ using XTW. $\tilde{\eta}(T_w, R_w^*) = 0.0180$ $(< \eta_{\max})$ and $\tilde{\Delta}(t_m, r_{\tau_{R^*}}) = 0.0086 (< 2 \times \eta_{\max})$, so condition (5-2) holds and tracking can continue. Online tracking proceeds similarly till $\tau_T = 682$ when $\tilde{\eta}(T_w, R_w^*) = 0.0217$ but $\tilde{\Delta}(t_m, r_{\tau_{R^*}}) = 0.3666$ $(> 2 \times \eta_{\max})$, so equation (5-2) is violated. XTW^{SP} is then used for accurate calculation of the difference between T_w and R_w^*. $\eta(T_w, R_w^*) = 0.0213$ and $\Delta(t_m, r_{\tau_{R^*}}) = 0.3633 (> 2 \times \eta_{\max})$, so equation (5-3) is

violated too. This confirms that the real-time signal is no longer similar to R_1 and the reference signal has to be re-identified with $\tau_T^{POD} = 682$ (see Figure 5-8).

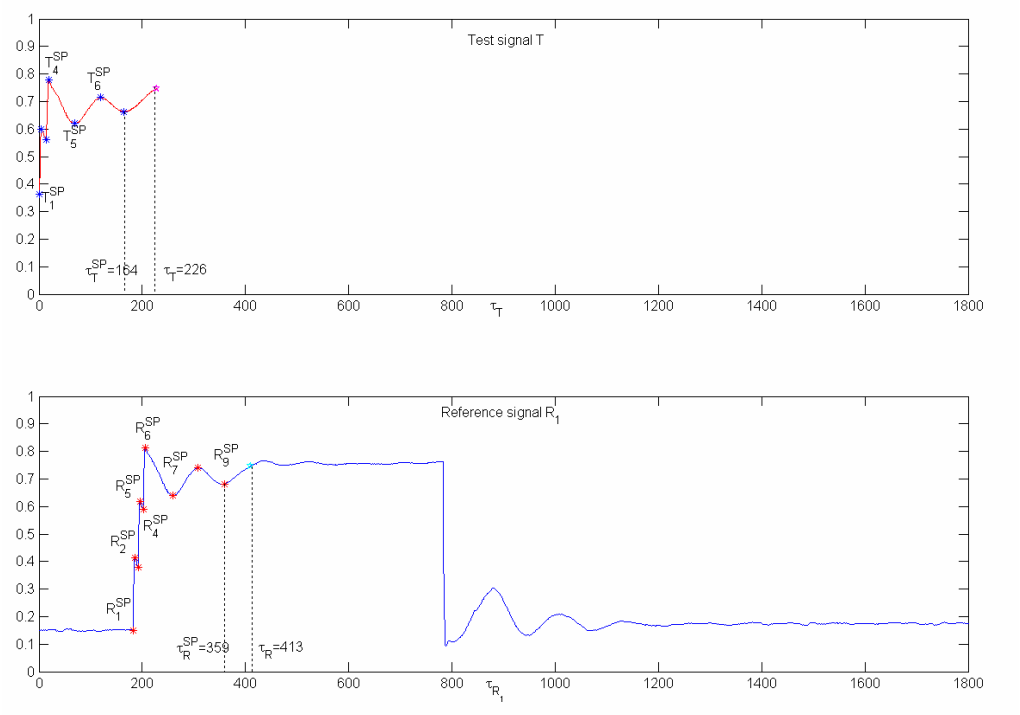


Figure 5-7**:** Snapshot ($\tau_T = 226$) the Signal comparison between$T$ and R

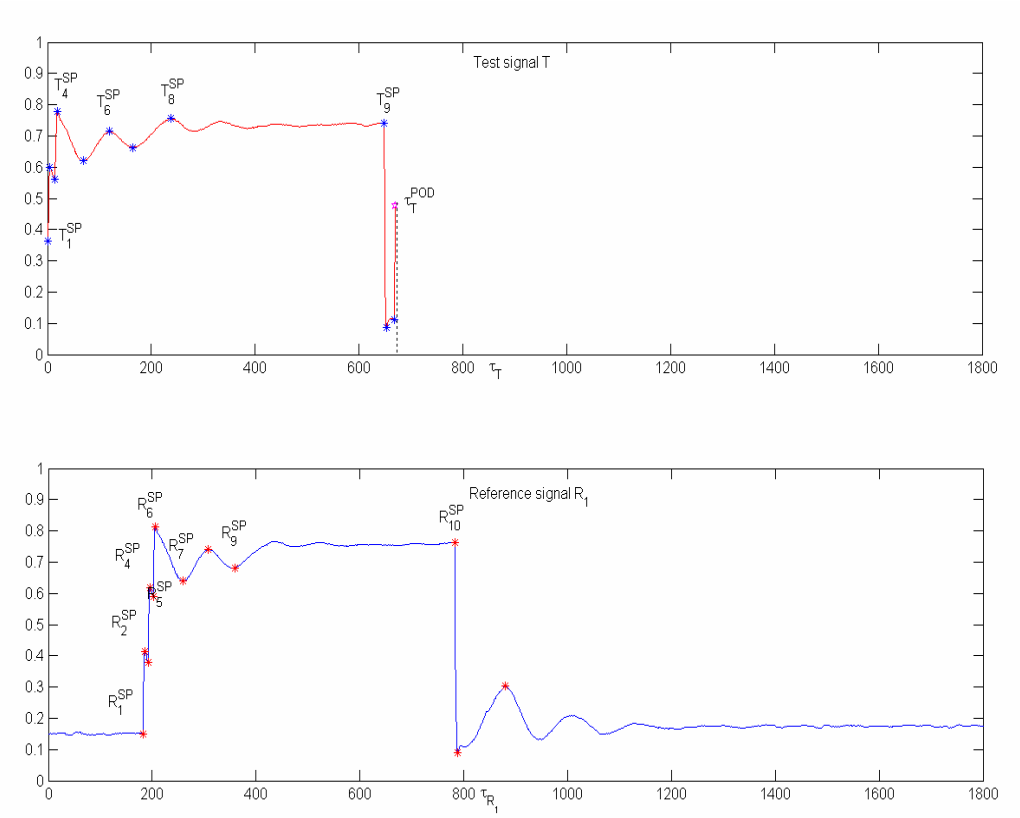


Figure 5-8**:** Snapshot ($\tau_T = 682$) the Signal comparison between$T$ and R

The divergent segment $T_D = \{t_{675}, t_{676,}..., t_{682}\}$ is used to find the new reference signal through DLA. Initially, $m - \tau_T^{POD} < 2\nu$, so en bloc difference calculation is performed. At $\tau_T = 682$, $\eta(T_D, R_2) = 0.0641$, $\alpha = 0.9596$ and equation (5-7) does not hold and the reference signal cannot be conclusively identified. Comparison is therefore repeated when subsequent samples becomes available. As shown in Figure 5-9, at $\tau_T = 685$, $\eta(T_D, R_2) = 0.0388 (< \eta_{\max})$, and $\alpha = 0.6039 (< \alpha_{\max})$. So the new reference signal is confirmed to be R_2 with $\tau_{R_2} = 1082$ as the anchor point and $\tau_{R^*} = 1087$ as base point.

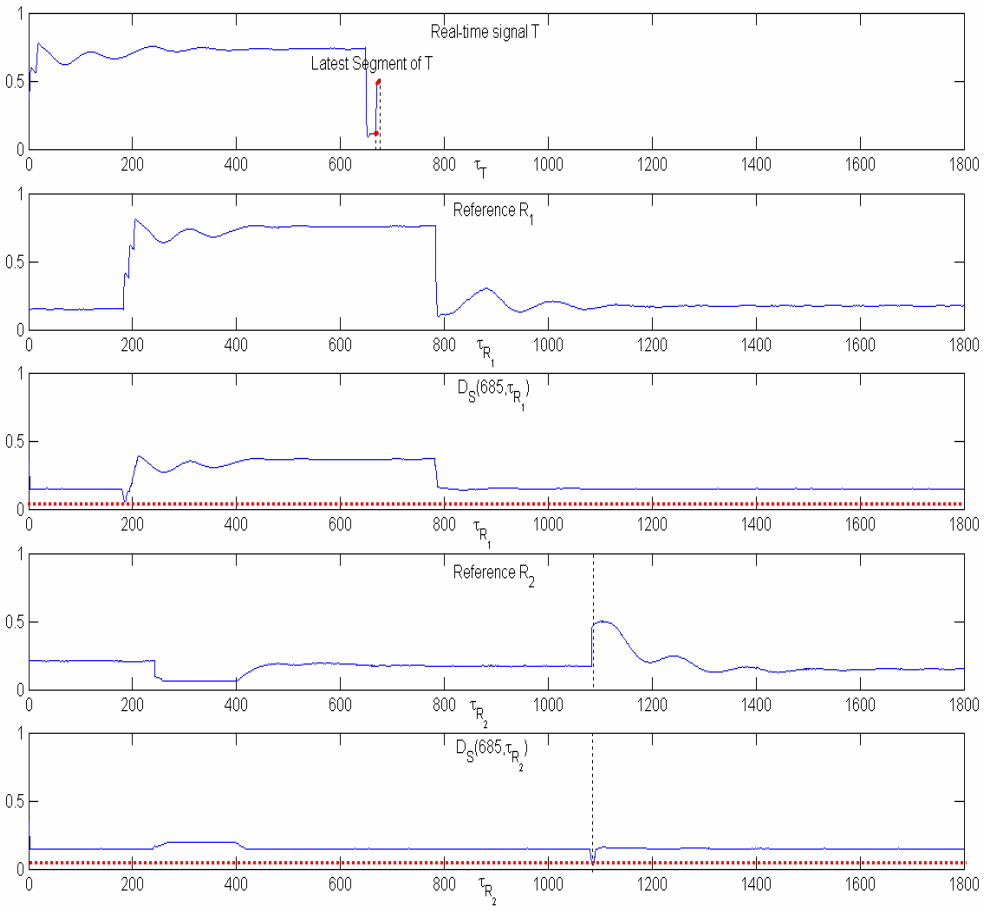


Figure 5-9**:** The comparison of real-time signal T at $\tau_T = 685$ with R_1 (shown in b) reveals minima at $\tau_{R_1} = 185$ (shown in c). Similar comparison with R_2 depicted in (d) shows a minimum at $\tau_{R_2} = 1087$ as shown in (e)

5.3 Case Studies

5.3.1 Online Process Disturbance Identification for the Tennessee Eastman Plant

The Tennessee Eastman (TE) plant is a popular testbed for process control, fault diagnosis and signal comparison as described in Sections 3 and 4. In this section, we use the above described online signal comparison method to identify process disturbances XD1 – XD5 and estimate their progress online. Figure 5-10 shows three runs where disturbance XD2 has affected the process.

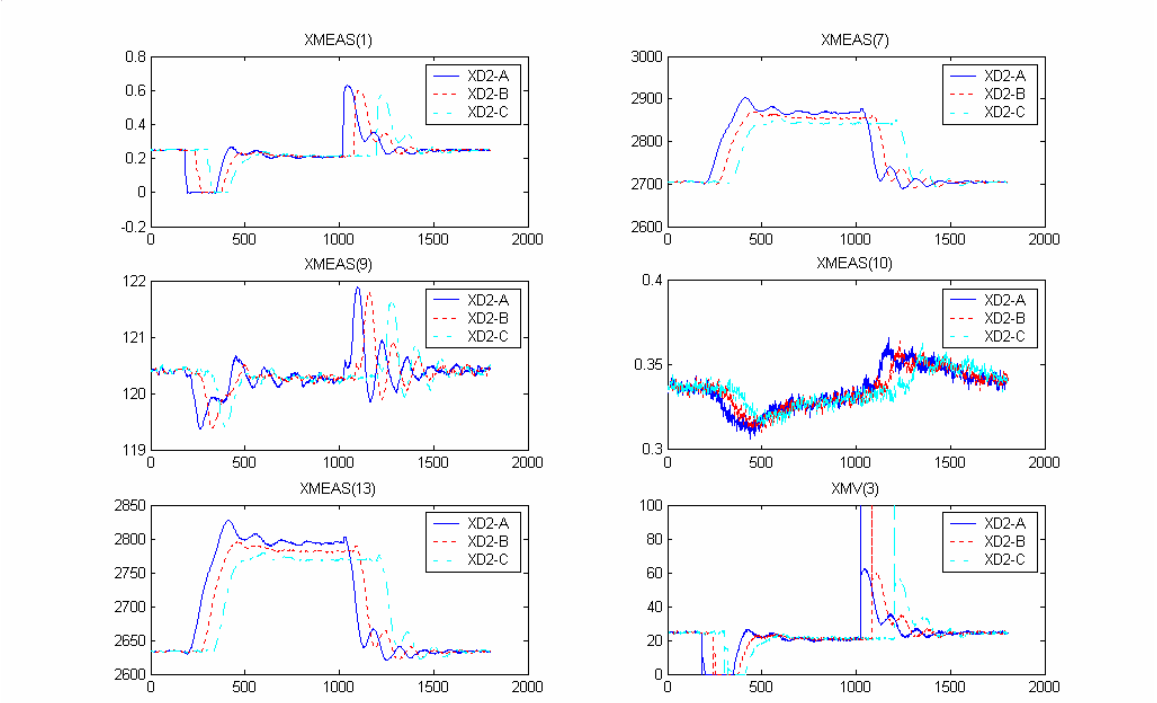


Figure 5-10: Three runs of XD2 with different magnitudes and duration

In this case, we use dynamic signal matching to identify process disturbances online. The difficulty in identifying the disturbance online can be estimated by a preliminary analysis (Section 3.4.2). Therefore, difference between the classes by direct comparison, even if the complete signal is available, is a nontrivial exercise. In this work, we consider the even more difficult task of differentiating between the disturbances as they evolve.

The proposed online signal comparison method is used for online disturbance identification as follows. Consider Run-4 where the process is in state XD2 until $\tau_T = 1030$. An unknown disturbance occurs starting at $\tau_T = 1030$ which is initially indicated when $\tilde{\eta}(T_w, R_w^*) = 0.0028$ and $\tilde{\Delta}(t_m, r_{\tau_{R^*}}) = 0.3147$, and equation (5-2) is violated. An accurate difference is calculated using XTW^{SP} and $\eta(T_w, R_w^*)$ = 0.0022 and $\Delta(t_m, r_{\tau_{R^*}}) = 0.3065$. Since this is larger than the $2 \times \eta_{\max}$ threshold even after resynchronization, as per equation (5-2) it is evident that the real-time signal does not confirm to reference signal R_2 starting from $\tau_T = 1030$ (= point of divergence τ_T^{POD}). The disturbance can be identified by calculating the new optimal reference signal R^* and the base point τ_{R^*} using the divergent segment $T_D = \{t_{m-\nu+1}, t_{m-\nu+2,}, ..., t_m\}$. In the first iteration, at time $\tau_T = 1030$, $\eta(T, R_1) = 0.0501$, $\eta(T, R_3) = 0.0511$, $\eta(T, R_4) = 0.0514$ and $\eta(T, R_5) = 0.0512$ (See Table 5-1). Since the η for all the reference signals are similar (as indicated by the inseparability ratio $\alpha = 0.9804 > \alpha_{\max}$), R^* cannot not been identified at this point and further iterations are necessary. In each subsequent iteration, as the real-time signal T evolves, the evaluation window is updated (as shown in Figure 5-4) and the analysis repeated. As the disturbance becomes more evident with time, α decreases (see Table 5-1) and at $\tau_T = 1034$, $\eta(T, R_4)$ falls below $\eta_{\max}$. The optimal reference R^* is then identified as R_4 (i.e. disturbance XD4). The base point at $\tau_{R^*} = 314$ in R_4 is found to correspond to $\tau_T = 1034$. Real-time signal tracking is then resumed for subsequent samples. The average time cost for this run was 0.0273 cpu seconds with the proposed method as against 55.2721 cpu s with DTW as summarized in Table 5-2 (depicted as the second

disturbance in Run-4).This brings out the computational advantages of the proposed strategy.

Table 5-1: TE Disturbance Identification (Run-4)

τ_T	η				α
	R_1	R_3	R_4	R_5	
1030	0.0501	0.0511	0.0514	0.0512	0.9804
1031	0.0592	0.0593	0.0589	0.0581	0.9864
1032	0.0659	0.0656	0.0654	0.0647	0.9970
1033	0.0693	0.0654	0.0572	0.0683	0.8746
1034	0.0723	0.0604	0.0408	0.0702	0.6755
1035	0.0755	0.0556	0.0240	0.0723	0.4317
1036	0.0782	0.0510	0.0072	0.0742	0.1412
1037	0.0779	0.0506	0.0070	0.0739	0.1383
1038	0.0773	0.0508	0.0068	0.0732	0.1339
1039	0.0767	0.0507	0.0065	0.0729	0.1282
1040	0.0760	0.0506	0.0064	0.0723	0.1265
1041	0.0752	0.0504	0.0061	0.0716	0.1210
1042	0.0745	0.0500	0.0061	0.0712	0.1220
1043	0.0737	0.0505	0.0060	0.0707	0.1188
1044	0.0729	0.0504	0.0060	0.0703	0.1190

A similar study was performed for fourteen other runs. A summary is presented here. The same accuracy was found in all test runs the online comparison tracking stage as well as shown in Table 5-2. In each run, two disturbances were introduced. In all cases, the proposed method correctly identified the disturbance with an average delay of 5.23 mins. The average time cost for online signal comparison is only 0.0989 cpu seconds (on a Pentium IV, 2.4 GHz cpu) in contrast to 55.3 seconds for DTW. This factor of 559 speedup in computation over DTW clearly shows the efficiency of the proposed method and illustrates its suitability for large-scale applications. For Run-9, the time cost is 0.2509 second for each update point which is obviously higher than other runs. It is because the difference between test signal and reference is larger than other cases. Once the difference larger than threshold, it needs resynchronization the real-time signal and reference signal which cost extra time than using normal XTW propagation alone.

Table 5-2: Online process disturbance detection in TE process

	Disturbance Introduction Time	Disturbance Identification Time	Best-matching reference	Identification delay (min)	Second Disturbance Introduction Time	Disturbance Identification Time	Best-matching reference	Identification delay (min)	Average Time Cost (cpu s)	Average Time Cost of DTW (cpu s)
Run-1	9	15	XD1	6	1020	1025	XD4	5	0.0501	55.3446
Run-2	9	13	XD1	4	670	675	XD4	5	0.0699	55.2385
Run-3	9	13	XD1	4	680	684	XD4	4	0.1233	55.2589
Run-4	9	17	XD2	8	1030	1034	XD4	4	0.0273	55.2721
Run-5	9	13	XD2	4	1100	1107	XD5	7	0.0734	55.3044
Run-6	9	16	XD2	7	420	427	XD5	7	0.1289	55.3314
Run-7	9	16	XD3	7	860	867	XD1	7	0.0367	55.3578
Run-8	9	13	XD3	4	970	974	XD1	4	0.0689	55.3446
Run-9	9	13	XD3	4	1100	1105	XD1	5	0.2509	55.3374
Run-10	9	14	XD4	5	820	825	XD2	5	0.0373	55.2985
Run-11	9	13	XD4	4	380	387	XD2	7	0.0535	55.3051
Run-12	9	17	XD4	8	400	407	XD2	7	0.2024	55.2517
Run-13	9	13	XD5	4	500	505	XD3	5	0.0784	55.2919
Run-14	9	13	XD5	4	600	604	XD3	4	0.0712	55.3051
Run-15	9	13	XD5	4	1080	1084	XD3	4	0.2115	55.2517
Average				5.1333				5.3333	0.0989	55.2996

5.3.2 Case Study 2: Online Fault Diagnosis during Startup of a Lab-scale Distillation Column

In this section, the proposed methodology is illustrated on a lab-scale distillation unit (Section 4.3.2)

The online signal comparison algorithm was then used for fault diagnosis and decision support during subsequent startups of the column. Consider one run (Run-3) when a fault was introduced at $\tau_T = 3590s$ when the operators introduced too large a feed pump flowrate (200 rpm) to the column. This causes instability in the column resulting in a drastic drop in the column's temperatures. Results from this run show that the real-time signal is initially close to normal. The difference between the real-time signal and all other references is much higher. $\alpha = 0.2235$ at $\tau_T = 6$ Starting from t=3700s, the difference between real-time signal and the normal reference increases indicating that there is a fault during the startup operation. The difference η between the real-time signal and DST03 is less than $\eta_{\max}$ (0.05). Also, the α falls below $\alpha_{\max}$ (0.70) and is identified as DST03.

Similar tests were done for all other cases. A summary of the findings is presented in Table 5-3. Faults in all the test runs were correctly identified with an average delay of 3.6 samples (and maximum detection delay of 11 samples for Run-03). All faults could be accurately identified within an average of 5.4 samples of their occurrence. The maximum identification delay was about 12 samples for Run-03. The average α at time of identification is 0.3828 against the $\alpha_{\max}$ threshold of 0.7 which shows the clear identification of the faults. The average computation time cost at each sample was 0.0594 second which is much less than the sampling rate of 10 seconds.

The proposed method is therefore clearly suitable for online fault diagnosis in this case as well.

5.3.2.1 Robustness and Parameter Tuning

In this section, the robustness of proposed online signal comparison method is studied. Varying amount of noises level were added to the online signal to investigate robustness to noise. The affect of the tuning parameter settings on online signal comparison was also studied. The proposed method uses $\alpha_{\max}$, $\eta_{\max}$ etc for detection. While in general, different process signals may require different values of these parameters, we have found that most of the parameter settings can be used across variables and case studies.

The robustness of the proposed method to sensor noise is reported in this part using data from the lab-scale distillation column case study. Additional measurement noises ranging from 1%, to 5% were added to all the original signals and fault diagnosis performed. Results are shown in Table 5-4. As has to be expected, there is an increase in the fault identification delay with increasing noise, however the effect is minimal with the average delay increasing from 5.3 samples to 6.6 samples. A similar study was performed for the TE case study as well (see Table 5-5) and the average identification delay increased from 5.1333 samples to 17.7333 samples at 5% noise level. This larger variation in the TE case arises from the inherent complexity and larger noise levels in the base process. Overall, the proposed method online signal comparison method is robust to noise.

The proposed method uses two tunable parameters – $\alpha_{\max}$ and $\eta_{\max}$. While in the general case, different process signals may require different values of these parameters, we have found that the same parameter settings can be used across variables and case

studies. The results of decreasing $\alpha_{\max}$ from 0.80 to 0.60 for the distillation column as well as the TE case studies show that $\alpha_{\max}$ has no significant affect on fault detection delay. A smaller $\alpha_{\max}$ would require a clearer separation between the reference signal and the optimal reference signal before a fault is confirmed. This would lead to a delay in fault identification. The average identification delay for TE case changed from 4.333 to 15.80 samples when $\alpha_{\max}$ was reduced from 0.80 to 0.60 (See Table 5-6) while for the distillation column case study the delay increased from 5.3 to 6.1 samples (See Table 5-7). The extent of robustness of $\alpha_{\max}$ is further revealed by the fact that for the distillation column case study, even setting $\alpha_{\max} = 0.30$ results in an average identification delay of only 9.1 samples. Overall these results clearly establish the robustness of the proposed method to the two tuning parameters.

A similar result was obtained for $\eta_{\max}$ as well. $\eta_{\max}$ was changed from 0.03 to 0.07 and tested in the TE process (See Table 5-8) and lab-scale distillation column data (Table 5-9). As can be seen there $\eta_{\max}$ has no significant affect on fault detection delay and the average delay remains around 5.1 sample points for the TE case and 5.5 sample points for the lab-scale distillation column.

Table 5-3: Faults diagnosis for Lab-scale Distillation column

Case	Time Fault Introduced (s)	Detection Time (sample)	Detection Delay (sample)	Identification Time (sample)	τ_y (sample)	Best matching reference	η	α	Identification Delay (sample)	Time cost (s)
Run-01	10	6	5	6	6	DST01	0.0162	0.2817	5	0.1716
Run-02	10	6	5	6	6	DST02	0.0097	0.1091	5	0.1028
Run-03	3590	370	11	371	13	DST03	0.0101	0.3456	12	0.0317
Run-04	3560	357	1	360	5	DST04	0.0217	0.6663	4	0.0342
Run-05	4250	426	1	430	6	DST05	0.0109	0.4113	5	0.0256
Run-06	3500	353	3	355	4	DST06	0.0268	0.2379	5	0.0368
Run-07	3450	346	1	347	3	DST07	0.0319	0.6940	2	0.0264
Run-08	4700	472	2	473	4	DST08	0.0100	0.2330	2	0.0256
Run-09	10	6	5	6	7	DST09	0.0103	0.1582	5	0.0556
Run-10	3000	302	2	309	5	DST10	0.0119	0.6910	9	0.0840
Average	**-**	**-**	**3.6**	**-**	**5.9**	**-**	**0.0156**	**0.3828**	**5.4**	**0.0594**

Table 5-4: Robustness of noise in TE disturbances identification

	Disturbances Introduction Time	**Identification Time (sample)**					**Identification Delay**				
		1%	2%	3%	4%	5%	1%	2%	3%	4%	5%
Run-1	181	187	192	194	194	199	6	11	13	13	18
Run-2	241	245	246	245	245	246	4	5	4	4	5
Run-3	301	305	306	313	313	313	4	5	12	12	12
Run-4	181	189	196	198	201	205	8	15	17	20	24
Run-5	241	245	244	254	256	265	4	3	13	15	24
Run-6	301	308	316	317	323	316	7	15	16	22	15
Run-7	181	188	199	201	195	203	7	18	20	14	22
Run-8	241	245	246	249	256	255	4	5	8	15	14
Run-9	301	305	317	318	326	329	4	16	17	25	28
Run-10	181	186	191	205	220	220	5	10	24	39	39
Run-11	241	245	253	253	253	253	4	12	12	12	12
Run-12	301	309	314	314	315	314	8	13	13	14	13
Run-13	181	185	188	196	193	198	4	7	15	12	17
Run-14	241	245	245	247	247	249	4	4	6	6	8
Run-15	301	305	313	315	314	316	4	12	14	13	15
Average							5.1333	10.0667	13.6000	15.7333	17.7333

Table 5-5: Robustness of noise in Lab-scale distillation column fault diagnosis

		Identification Time (sample)					Identification Delay				
	Faults Introduction Time	1%	2%	3%	4%	5%	1%	2%	3%	4%	5%
Run-1	1	6	6	6	6	6	5	5	5	5	5
Run-2	1	6	6	6	6	6	5	5	5	5	5
Run-3	359	371	371	371	371	371	12	12	12	12	12
Run-4	356	360	360	360	360	360	4	4	4	4	4
Run-5	425	430	430	430	430	438	5	5	5	5	13
Run-6	350	354	355	355	355	355	4	5	5	5	5
Run-7	345	347	347	347	347	348	2	2	2	2	3
Run-8	470	473	473	473	473	473	3	3	3	3	3
Run-9	1	6	6	6	6	6	5	5	5	5	5
Run-10	300	308	308	309	309	311	8	8	9	9	11
Average							5.3	5.4	5.5	5.5	6.6

Table 5-6: Effect of α_{min} on identification delay in TE case study

	Disturbances	**Identification Time (sample)**					**Identification Delay**				
	Introduction Time	$\alpha_{max}=0.80$	$\alpha_{max}=0.75$	$\alpha_{max}=0.70$	$\alpha_{max}=0.65$	$\alpha_{max}=0.60$	$\alpha_{max}=0.80$	$\alpha_{max}=0.75$	$\alpha_{max}=0.70$	$\alpha_{max}=0.65$	$\alpha_{max}=0.60$
Run-1	181	187	187	187	187	187	6	6	6	6	6
Run-2	241	245	245	245	245	245	4	4	4	4	4
Run-3	301	305	305	305	311	313	4	4	4	10	12
Run-4	181	189	189	189	190	191	8	8	8	9	10
Run-5	241	245	245	245	245	267	4	4	4	4	26
Run-6	301	304	305	308	313	313	3	4	7	12	12
Run-7	181	186	187	188	189	189	5	6	7	8	8
Run-8	241	245	245	245	249	249	4	4	4	8	8
Run-9	301	305	305	305	315	316	4	4	4	14	15
Run-10	181	185	185	186	186	186	4	4	5	5	5
Run-11	241	245	245	245	247	247	4	4	4	6	6
Run-12	301	305	308	309	313	327	4	7	8	12	26
Run-13	181	184	184	185	185	185	3	3	4	4	4
Run-14	241	245	245	245	245	282	4	4	4	4	41
Run-15	301	305	305	305	313	355	4	4	4	12	54
Average							4.3333	4.6667	5.1333	7.8667	15.8000

Table 5-7: Effect of α_{min} on identification delay in Lab-scale distillation column case study

		Identification Time (sample)					η				
	Faults Introduction Time	$\alpha_{max}=0.80$	$\alpha_{max}=0.75$	$\alpha_{max}=0.70$	$\alpha_{max}=0.65$	$\alpha_{max}=0.60$	$\alpha_{max}=0.80$	$\alpha_{max}=0.75$	$\alpha_{max}=0.70$	$\alpha_{max}=0.65$	$\alpha_{max}=0.60$
Run-1	1	6	6	6	6	6	5	5	5	5	5
Run-2	1	6	6	6	6	6	5	5	5	5	5
Run-3	359	371	371	371	371	371	12	12	12	12	12
Run-4	356	359	359	360	360	361	3	3	4	4	5
Run-5	425	430	430	430	432	432	5	5	5	7	7
Run-6	350	355	355	355	355	355	5	5	5	5	5
Run-7	345	347	347	347	347	348	2	2	2	2	3
Run-8	470	473	473	473	473	473	3	3	3	3	3
Run-9	1	6	6	6	6	6	5	5	5	5	5
Run-10	300	308	308	309	310	311	8	8	9	10	11
Average							5.3	5.3	5.5	5.8	6.1

Table 5-8: Effect of $\eta_{\max}$ on identification delay in TE case study

	Disturbance Introduction Time	**Identification Time (sample)**					**Identification Delay (sample)**				
		0.03	0.04	0.05	0.06	0.07	0.03	0.04	0.05	0.06	0.07
Run-1	181	187	187	187	187	187	6	6	6	6	6
Run-2	241	245	245	245	245	245	4	4	4	4	4
Run-3	301	305	305	305	305	305	4	4	4	4	4
Run-4	181	189	189	189	189	189	8	8	8	8	8
Run-5	241	245	245	245	245	245	4	4	4	4	4
Run-6	301	308	308	308	308	308	7	7	7	7	7
Run-7	181	188	188	188	188	188	7	7	7	7	7
Run-8	241	245	245	245	245	245	4	4	4	4	4
Run-9	301	305	305	305	305	305	4	4	4	4	4
Run-10	181	186	186	186	186	186	5	5	5	5	5
Run-11	241	245	245	245	245	245	4	4	4	4	4
Run-12	301	309	309	309	309	309	8	8	8	8	8
Run-13	181	185	185	185	185	185	4	4	4	4	4
Run-14	241	245	245	245	245	245	4	4	4	4	4
Run-15	301	305	305	305	305	305	4	4	4	4	4
Average							5.1333	5.1333	5.1333	5.1333	5.1333

Table 5-9: Effect of η_{max} on identification delay in Lab-scale distillation column case study

		Identification Time (sample)					**Identification Delay (sample)**				
	Fault Introduction Time	0.03	0.04	0.05	0.06	0.07	0.03	0.04	0.05	0.06	0.07
Run-1	1	6	6	6	6	6	5	5	5	5	5
Run-2	1	6	6	6	6	6	5	5	5	5	5
Run-3	359	371	371	371	371	371	12	12	12	12	12
Run-4	356	360	360	360	360	360	4	4	4	4	4
Run-5	425	430	430	430	430	430	5	5	5	5	5
Run-6	350	355	355	355	355	355	5	5	5	5	5
Run-7	345	348	347	347	347	347	3	2	2	2	2
Run-8	470	473	473	473	473	473	3	3	3	3	3
Run-9	1	6	6	6	6	6	5	5	5	5	5
Run-10	300	309	309	309	309	309	9	9	9	9	9
Average							5.6	5.5	5.5	5.5	5.5

5.4 Discussion

Online signal comparison is important for process monitoring, fault diagnosis, and process state identification. In this section, we have proposed a signal comparison based strategy for online disturbance or fault identification. Given a suitably annotated historical database of process states – normal and abnormal, the proposed method finds the best matching state at any given time by comparing the real-time sensor measurements with the signals in the database. In contrast to signal comparison strategies reported in literature, which are designed for offline signal comparison, the proposed method does not require any a priori knowledge about the online signal – specifically the beginning and end of the real-time signal do not need to coincide with those of the library signals. The endpoints of the two signals are synchronized automatically using the dynamic locus analysis methodology. DLA is inherently computationally efficient when the real-time signal is small; the flanking strategy proposed here reduces the search complexity tremendously when a long segment of the real-time signal has to be compared. The real-time signal tracking strategy based on XTW and XTW^{SP} further reduces the computational load required when the process essentially follows a previously determined reference signal. These endow the main advantage of the proposed method – is that it is significantly faster in comparison with other time warping methods. This has been illustrated clearly using two different case studies – disturbance identification in the Tennessee Eastman challenge plant and fault online diagnosis during startup of a lab-scale distillation column. As shown in Section 4, the method is also robust to noise as well as parameter settings.

Signal comparison methods use long term process signal to identified process state. With the comparison similarity between different signal groups, the similar

process state can be found. If the comparison was applied to all the variables in a complex process, the computation load will be very heavy. In next chapter, a novel approach - selecting state-specific key variables is introduced which will reduce the variables number for monitoring and comparison.

Chapter 6 Selecting State-Specific Key Variables

6.1 Introduction

A complex unit with a large number of variables may be operated in multiple states. The optimal set of sensors or key variables for monitoring a process may differ in different states of the process. It is therefore beneficial if a state-specific subset of variables is used for monitoring complex processes. The benefits of state-specific key variables are studied in this chapter. The outline of this chapter is as follow: In Section 2.3, previous work in key variables selection is reviewed. We describe some basic axioms regarding key variables and their selection in Section 6.2. Methodologies for identify different kinds of key variables are also given in this Section. We give a detailed example of its application to the ShadowPlant in Section 6.3. In Section 6.4, conclusions and discussion from this work are presented.

6.2 Basics of State-Specific Key Variables

Definition: Key variables are group of variables that can give much more information of a chemical process transition than other variables. They also have more effect on the process. Key variables will change according to the process state and purpose of use. The following are basic axioms for key variables:

1. **Key variables will change according to different observers**

The subset of the process variables which are deemed to be key variables depends on the purpose that they are to be used for. Therefore, the set of key variables used for the purpose of controlling a process unit may not be the same as the set for monitoring the same unit.

For example, during normal operation, operations personnel pay attention to process balances, reactor temperature, catalyst performance, distillation column levels, etc. However, personnel in the commercial department may be interested in variables such as the product flow, yield, quality and quantity of products. From an environmental and energy efficiency point-of-view, the key variables would relate to flue gas components, wastewater processing, and energy cost.

2. Key variables will change for different states of the unit

The set of key variables should be dynamic, that is, as the process evolves through different states of operation, the key variables would change. For each state of startup, there are some variables that are very active (varying widely) and hence important. At the same time, other variables may have to be maintained stably within a narrow range.

3. Key variables depend on features of the unit and its operation

In general, key variables can be identified based on the type of the unit operation. For example, fuel and airflow rate, and outlet temperature are key variables for furnace operation. Similarly, temperature and pressure are important in reactors. More specifically, for a Continue Catalyst Reforming (CCR) unit, the most important variables for operation are reactor pressure, temperature sulfur, heavy metal atom of the feed, and H_2/Oil rate. The actual decision on the importance of a variable depends on detailed analysis of the process and the role of the variable in the broader context of the actual processing.

4. Key variables depend on the hierarchical organization of the process

Key variables should be defined hierarchically to reflect the division of the process operation into different levels of granularity. Three layers of key variables should be distinguished. At the top-layer, process operation is comprised of macro

states. Each macro state is in turn comprised of one or more sub-states. The actual operation steps such as opening and closing valves comprise the detailed sub-states. Each sub-state is associated with a set of key variables. That is, a subset of all the process variables that can clearly distinguish one macro state from another is used as key variables at the macro-state level. Similarly, key variables that are used at the sub-state level can distinguish among the sub-states.

6.2.1 Principles for Key Variables Selection from an Operational Standpoint

The following factors should be considered when selecting key variables.

1. Comprehensibility

Every key variable should have easy to understand relationships with other variables. The ability to predict either quantitatively or at least qualitatively the effect of the key variables on other variables is important. If the variable doesn't have any obvious relation with other variables, it cannot serve as a key variable.

2. Comprehensive spatial coverage

Key variables should be selected so that they reflect and affect the status of all parts of a process. If all key variables are focused on one or a few subunits, other sections of the process will become unobservable, with adverse consequences on the safety and controllability of those sections.

3. Comprehensive temporal coverage

While selecting key variables, their response times should be considered. This is not to say that the faster a variable's response, the better it is suited to be a key variable. The group of key variable should be selected to reflect all the time scales that the process operates in.

4. Sufficiency

The set of key variables should be complete so as to represent the most important changes in the process and to be able to reflect disturbances quickly. At the same time, the number of key variable should not be large.

To derive a systematic basis for identifying key variables, we first classify the key variables.

6.2.2 Key Variable Classification

Key variables can be characterized into six types: (1) State-indication variable, (2) State-differentiation variable, (3) State-progression variable, (4) External-affect variable, (4) Active variable and (6) Important-balance variable.

1. State-indication Variable

This kind of variables is usually useful in indicating the beginning or end of the macro-state of a unit. In most chemical process, the states of some valves, flow rates, and temperatures would indicate the state of operation. In general, these variables are useless in routine operation.

2. State-differentiation Variable

State-indication variables alone cannot identify process state completely. The state-indication variables will respond when a new state starts but they don't conclusively differentiate among the states. Variables that have most differentiation value in each state are called State-differentiation variables. Their values are in different ranges during different states of the process operation.

3. State-progression Variable

For most process transitions, there are some variables that reflect the progress of the state (Davis, 1984).

4. External-affect Variable

In many modern integrated chemical plants, the state of an upstream unit would have some affect on the downstream unit's operation. It is easy to understand that operations in other subsections may affect the state of a subsection. The variables that display this kind of effect are called external affect variables.

5. Active Variable

Signals which show variation in a state should be monitored as well. These variables commonly provide detailed insight into the operation. Most operating variables fall into this category. The standard way for evaluating activity is based on the singular points analysis of the signal.

6. Important-balance Variable

For most chemical process operation, there are some important balances - mass, energy, pressure and others. The variables that affect these important balances called Important-balance variables.

The same variable can exhibit more than one characteristic, for example, the same variable can be both a state indication and a state-progression variable.

Different classes of key variables should be defined at different levels in the process hierarchy. The bottom-layer key variables are used to give the detailed operating information about current operation. For example, only active variables, external-effect variable, and balance variables need to be monitored within each detailed-state to effectively identify if the process is being operated correctly. State-indication and state differentiate variables are suited for the top-layer to find the active macro-states of the whole process. In the middle layer, key variables are used to find the sub-state of the system based on the macro-state. The key variables in the middle-layer and bottom-layer will change according to the progress of the process transition. The different variable types for each level are shown in Table 6-1.

Table 6-1: Relation between variable type and state level

Variable type	Macro-State	Sub-State	Detailed-State
State-indication Variable	√	√	×
State-differentiation Variable	√	×	×
Stag-progression Variable	×	×	√
External-affect Variable	×	×	√
Active Variable	×	×	√
Important-balance Variable	×	×	√

6.2.3 Methodology for finding each type of key variable

State-indication Variable

Since it is not easy to demarcate two adjacent states clearly, we need to use some simple operations for separation. State-indication variables are used for this purpose. Most of the time, the state-indication variable can be identified from the standard operation procedure (SOP) which specify the operations to be performed in different states. Examples include status of valve, equipment status, etc. Special valve or equipment status could also be use for this class of key variables.

State-differentiation Variable

The state-differentiation variable is determined based on signals that have different values in each state. In this chapter, we use a method motivated by the Analytical Hierarchy Process (AHP) popular in decision-making, to find the best subset of variables for separating the different states of a process transition.

First the differentiability of a single variable is measured. Consider a variable x whose signal $T = \{x_1, x_2, ..., x_i, ...x_m\}$ corresponds to all states of the process' operation. Let $T' = \{x_i, x_{i+1}, ..., x_j\}$ be the signal from a specific state M_S for which the suitability of variable x to serve as a key variable is being considered. The state-differentiability

ϑ of signal T during M_S is defined based on the average overlap of the max and min values of every non-extreme three-sample window in T' with T:

$$\vartheta = \frac{1}{j-i-1}\sum_{n=i+1}^{j-1}\frac{\text{Total number}\{\min(x_{n-1},x_n,x_{n+1}) \le T' \le \max(x_{n-1},x_n,x_{n+1})\}}{\text{Total number}\{\min(x_{n-1},x_n,x_{n+1}) \le T \le \max(x_{n-1},x_n,x_{n+1})\}} \quad (6\text{-}1)$$

$$\frac{j-i-1}{m-2} < \vartheta \le 1$$

Here the operation *Total number( )*calculates the number of times the condition in the operand is satisfied by the given signal. A small ϑ indicates that variable x cannot be used for positively identifying state M_S, while a value of $\vartheta = 1$ indicates that the variable can conclusively locate when the process is in M_S. While ϑ can indicate the state-differentiability of a single variable, it is necessary to identify the state-differentiability of a set of variables. The extension to the multivariate case where the differentiability of a set of variables is considered next.

Let the process be operated in s states. A subset q key variables from the total of Q measurements has to be selected so as to provide best overall differentiability of every state.

The Q x s dimensioned *variable differentiability matrix A* is first calculated by calculating the state-differentiability of each variable for each state by following the above procedure. Let the relative importance of differentiating any state be κ. $0 \le \kappa \le 1$ The relative importance matrix for all the states is constructed as $B = [\kappa_1 \ \kappa_2 \ ... \ \kappa_s]$. One way to estimate the relative importance of a state is based on its extent relative to other states. The correlation among the Q variables is measured using their covariance C. The following decision procedure is then used to select the q key variables using A, B, and C:

1. Calculate the overall differentiability matrix $W = A * B$ and select the variable whose coefficient in W is the largest. This y^{th} variable is deemed to be a State-differentiation key variable.

2. Each element of the variable differentiability matrix A is then updated using the covariance matrix C as follows: $A_{m,n} = A_{m,n} * (1 - C(y,m))$. Step 1 is then repeated to find the overall differentiability matrix and select the next best State-differentiation key variable. This is repeated until max (W) < a user specified threshold or q variables have been selected.

The above procedure provides a systematic way to discount variables that are correlated and is necessary since many variables offer only similar information.

State-progression Variable

These key variables are identified based on the variable's ability in indicating the progress of a state. These can be identified from the nature of the state. If a variable shows different values during the progression of a state, it is suitable for use as a state-progression variable. Consider a signal $T = \{x_1, x_2, ..., x_i, ...x_n\}$. The variable's ability to indicate the progress of a state M_S is calculated based on its uniqueness index ς within that state:

$$\varsigma = \frac{1}{n-2} \sum_{i=2}^{n-1} \frac{1}{Total\ number\{\min(x_{i-1}, x_i, x_{i+1}) \le T \le \max(x_{i-1}, x_i, x_{i+1})\}} \tag{6-14}$$

As in the case of state-differentiation variable, state-progression is calculated based on how many other points are located in the $[\min(x_{i-1}, x_i, x_{i+1}), \max(x_{i-1}, x_i, x_{i+1})]$ range during the given state. $1/n < \varsigma \le 1$. A small value of ς reveals that the variable has a poor capability to serve as a state-progression key variable while $\varsigma = 1$ indicates that every value of the variable is unique through the state; thus the variable can

exactly reveal the progression of the state. The method described above for identifying sets of variables that together have high differentiation ability can also be used here to find a set of state-progression variables.

External-affect Variable

The external-affect variable is identified based on the definition of the plant sections. It should be noted that a variable that is the input to a given section would be an output from another section. Identification of these kinds of variables has to be based on the process flow sheet and the division of the process into sections.

Active Variable

In this work, signals that are active in a state are identified from the number of singular points. A signal that has more singular points within a given period (state) is more active than another variable with fewer singular points. The algorithm for identifying singular points has been reported earlier. Other analysis techniques such as PCA and ICA may also be used for evaluating signal activity.

Important-balance Variable

Identification of such variables is based on the heat and mass balance of the process. All transient process states – normal transitions and all abnormal operations – occur when a steady-state balance is broken. Process knowledge is important to find the group of variables that affect a specific balance.

In the next Section, we describe a systematic methodology for selecting key variables in real-time.

6.2.4 Online Identification of Key Variables

The methodology for identifying key variables described above has to be applied offline. A knowledge-base of these key variables can then be used for online applications. The flow chart for online key variables selection is shown in Figure 6-1.

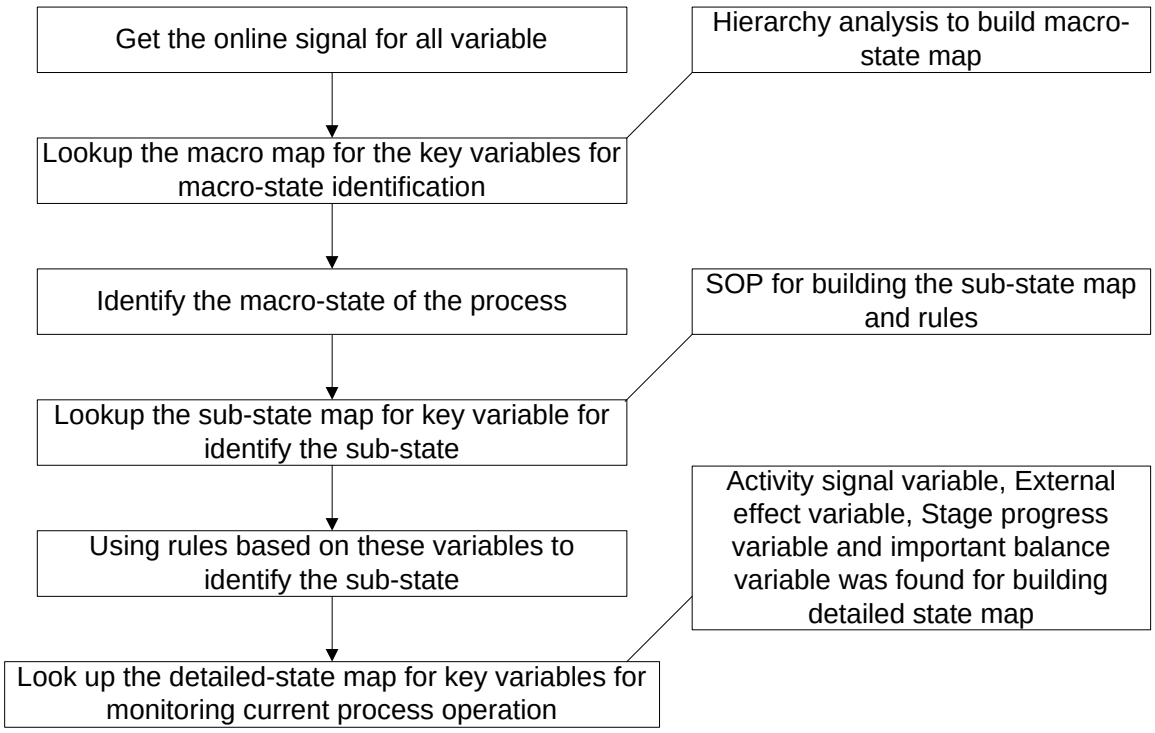


Figure 6-1: Flow chart for online key variables identification

We use different methods to find each type of key variables dynamically. The macro-state is first identified using state-indication and state-differentiation variables. These variables are used as input to a neural network which identifies the macro-state. Once the macro-state is identified, we use sub-state maps to obtain the key variables for identifying the sub-state. The sub-state map is obtained from the SOP and contains the variables that can be used for differentiating about the sub-states. In this layer, since the macro-state is already fixed, simple rules on the operation variable are adequate to distinguish between the sub-states. Given the sub-state, we lookup the detailed-map to obtain the detailed key variables in that sub-state. The detailed state is

obtained using active, external-affect, as well as important-balance variables through state-progression analysis.

A detailed illustration of the above methods for finding key variables is provided next. We also describe dynamic key variables selection and its benefits using the ShadowPlant startup case study.

6.3 Case study

State-specific key variables selecting can be used for aiding complex process operations such as the startup of a fluidized catalytic cracking. 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. A high-fidelity dynamic simulator of a FCCU, called ShadowPlant, is used here. The tags for these and other important equipment in the unit are shown in Table 6-2. Details of the unit and the startup transition are reported in Section 3.4.1.

Table 6-2: Major Process Equipment

Tag	Description
C-100	Air Blower
C-200	Wet Gas Compressor
D-100	Waster Heater Boiler Steam Drum
D-200	Main Fractionator Overhead Receiver
E-100	Waster Heat Boiler
H-100	Air Preheater
H-200	Feed Preheater
R-100	Regenerator
T-200	Main Fractionator

6.3.1 Selecting Key Variables for Monitoring ShadowPlant Startup

Since the ShadowPlant has more than four hundred variables, a hierarchical division of the process is essential. Top layer variables are used to monitor the Macro-State of ShadowPlant. The middle layer key variables are used to identify sub-state of

the ShadowPlant during the startup transition. Bottom layer variables are used to monitor the detailed operation.

There are seven main states during the startup of the ShadowPlant:

1. *Preparation for air blower for startup:* In this state, the unit is checked offline.
2. *Air blower startup:* In this state, the air blower is connected to the vent and turned on. The operator also checks the lines to the downstream units and subsequently establishes airflow to them by closing the vent.
3. *Regenerator warm-up and catalyst loading:* Fuel gas and diesel are used to increase the regenerator temperature following a specified profile. Catalyst is also loaded into the regenerator at this state.
4. *Main fractionator startup:* In this part, the main work is to preheat the feed and bring kerosene to the distillation system. Slurry boiler is started up.
5. *Connecting reactor to main fractionator and catalyst circulation:* Connecting the two main parts of the ShadowPlant together is the main purpose in this state. Cycling of the catalyst between regenerator and riser is also begun.
6. *Introducing fresh feed:* This is the key operation of the FCCU and a major milestone during the startup. The main work is to maintain a steady reaction temperature and fractionator adjustment.
7. *Wet gas compressor startup and increasing feed to design capacity:* In this period, the wet gas compressor is started up and the feed increased slowly to design capacity.

State-indication Variables:

State-indication variable are selected based on the SOP to give a clear differentiation between the above seven states. For example, FOD111.OP (Air blower speed %) is a

good indication for the air blower startup state. Other variable are also useful to confirm the state of process, for example, the regenerator temperature should register a significant increase when the airflow has been established. Similarly, FOD101.PV is a good indication variable for *Regenerator warm-up and catalyst loading* state. The lists of state-indication variables for the different states are listed below.

State	**State-Indication Variable**	**Tag**
Air blower startup	C-100 Air Blower Speed	FOD111.OP
Regenerator warm-up and catalyst loading	H-100 Regenerator Air Bypass	FOD101.PV
Main fractionator startup	Startup Fuel Gas to Main Fractionator Overhead Receiver	FOD228.OP
Connecting reactor to main fractionator and catalyst circulation	Disengager Startup Vent	16PC105.SP
Introducing fresh feed	Line up Hot Feed	FOD200.PV
Wet gas compressor startup	C-200 Wet Gas Compressor	FOD233.PV

State-differentiation Variables:

In order to find the key variables for differentiating the macro-states, the above described hierarchical method is used (see Figure 6-2). The variable differentiability for 23 selected variables is shown in Table 6-3. The coefficient for each variable in a state is based on the variable's ability to distinguish that state from other states. Thus, FOD111.OP with a variable differentiability coefficient of 1.0 is a good variable to differentiate M_S-1 whereas FOD101.PV with a value of 0.3489 has poor differentiation ability for this state. Similarly 16FC105.PV has high differentiation ability for M_S-1 and M_S-2.

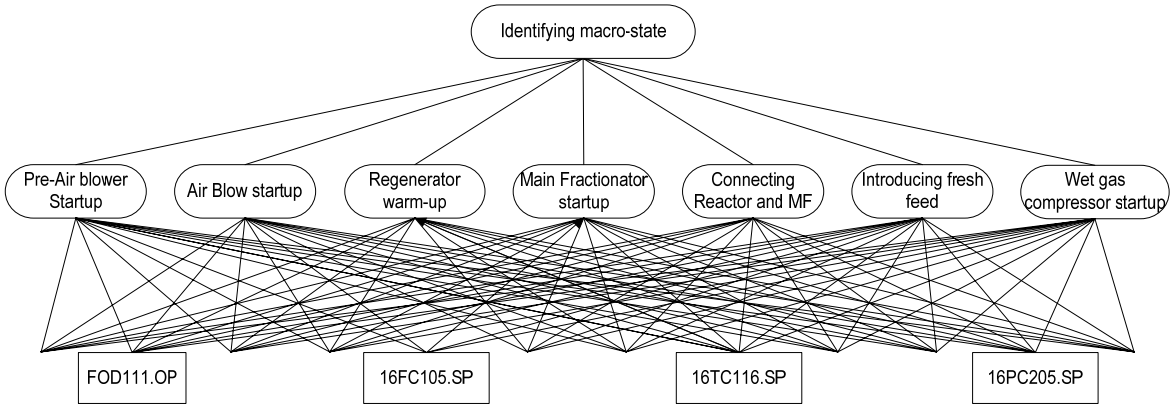


Figure 6-2: Hierarchical structure for finding the key variables for differentiating among the macro-states

The overall differentiability among the variables is used to select a small set of variables for state-differentiation. The correlation matrix among these 23 variables shown in Table 6-4 is used for this purpose. For instance, FOD102.OP has high correlation with 16FC105.PV (0.7621), and should not be included in the set. The following is the list of all the state-differentiation variables selected for the ShadowPlant startup.

16FC105.PV	C-100 Regeneration Air
16FC200.PV	Gas Oil Feed
16TC116.PV	R-100 Riser Temperature Control
16LI100.PV	R-100 Regenerator Catalyst Level
FOD111.OP	C-100 Air Blower Startup Speed
16PC205.OP	D-200 Wet Gas to C-200
FOD228.OP	Startup Fuel Gas to Main Fractionator Overhead Receiver
16LC201.PV	T-200 Bottom Level

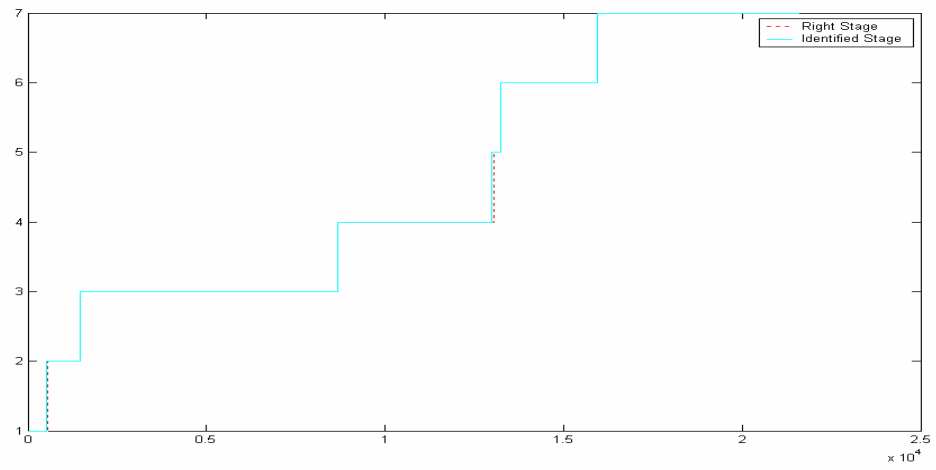


Figure 6-3: Result of ShadowPlant macro-state identification with Neural Network

A neural network was trained to differentiate among the seven states using the combined set of twelve state-indication and state-differentiation variables. It was confirmed that the macro-state of the ShadowPlant could be differentiated with an accuracy of 99.48%. Misdetection only occurs between States 4 and 5 as shown in Figure 6-3.

Table 6-3: Variable differentiability matrix for each candidate variable in case study

Index	State	M_S-1	M_S-2	M_S-3	M_S-4	M_S-5	M_S-6	M_S-7
V_1	FOD111.OP	1.0000	0.2415	0.3445	0.2059	0.0134	0.1251	0.2746
V_2	16PC111.OP	0.5861	0.4040	0.4769	0.2512	0.0163	0.1526	0.3349
V_3	16PC105.OP	0.8280	0.5669	0.3518	0.2102	0.0136	0.1277	0.2833
V_4	16FC105.PV	0.8280	0.7796	0.4308	0.2596	0.0168	0.1577	0.6509
V_5	16HC100.OP	0.0241	0.3499	0.3391	0.2026	0.0131	0.1231	0.2702
V_6	FOD101.PV	0.3489	0.6579	0.3576	0.2137	0.0139	0.1298	0.2850
V_7	16FC109.PV	0.0614	0.1146	0.3769	0.3256	0.0211	0.1978	0.2780
V_8	16FC107.PV	0.0296	0.0552	0.6574	0.2470	0.0160	0.1501	0.3294
V_9	FOD102.OP	0.3349	0.6248	0.3529	0.2144	0.0139	0.1302	0.2859
V_{10}	FOD106.OP	0.0256	0.0478	0.3415	0.2139	0.0139	0.1299	0.2852
V_{11}	16LI100.PV	0.1465	0.2734	0.7307	0.7697	0.2331	0.3130	0.6870
V_{12}	FOD228.OP	0.0266	0.0496	0.3712	0.1868	0.1260	0.1323	0.2958
V_{13}	FOD223.PV	0.0323	0.0603	0.4514	0.2698	0.0175	0.1639	0.9866
V_{14}	16LC200.PV	0.0483	0.0902	0.6751	0.1999	0.0255	0.2388	0.5242
V_{15}	16FC200.PV	0.0532	0.0993	0.7439	0.7741	0.0402	0.6054	0.8293
V_{16}	16TI203.PV	0.0540	0.1007	0.7539	0.1959	0.0245	0.2292	0.5031
V_{17}	FOD205.OP	0.0528	0.0985	0.7378	0.2313	0.0236	0.2206	0.4841
V_{18}	FOD211.OP	0.0395	0.0736	0.5515	0.3296	0.0189	0.3056	0.6707
V_{19}	16TC116.PV	0.0799	0.1491	0.6753	0.7213	0.4493	0.3186	0.7163
V_{20}	16HC103.OP	0.2047	0.3820	0.6823	0.5340	0.0346	0.3065	0.8494
V_{21}	16LC201.PV	0.0559	0.1044	0.7815	0.5288	0.0266	0.2495	0.5475
V_{22}	FOD200.PV	0.0389	0.0725	0.5430	0.3246	0.0211	0.3130	0.6891
V_{23}	16PC205.OP	0.0544	0.1015	0.7597	0.6143	0.0489	0.4575	0.8117

Table 6-4: Correlation matrix for candidate variables

COV	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}	V_{17}	V_{18}	V_{19}	V_{20}	V_{21}	V_{22}	V_{23}
V_1	1.0000	0.4510	0.4463	0.5933	0.2725	0.6544	0.1836	0.0797	0.6401	0.0483	0.3516	0.0675	0.1056	0.1796	0.1592	0.1906	0.1953	0.1435	0.1886	0.3207	0.2020	0.1407	0.1858
V_2	0.4510	1.0000	0.2602	0.6089	0.3234	0.6432	0.5102	0.3430	0.6562	0.3086	0.8756	0.1876	0.2934	0.4989	0.4424	0.5296	0.5428	0.3987	0.5241	0.7733	0.5612	0.3909	0.5162
V_3	0.4463	0.2602	1.0000	0.7353	0.4209	0.4086	0.3873	0.0689	0.3866	0.0045	0.2840	0.5608	0.7326	0.4734	0.7293	0.4428	0.4304	0.6075	0.6115	0.2163	0.4272	0.6249	0.5821
V_4	0.5933	0.6089	0.7353	1.0000	0.4707	0.7512	0.0588	0.0297	0.7621	0.0175	0.5408	0.3655	0.5633	0.4639	0.6840	0.4514	0.4607	0.4889	0.5463	0.5477	0.4670	0.4922	0.5553
V_5	0.2725	0.3234	0.4209	0.4707	1.0000	0.4983	0.0763	0.0321	0.4882	0.0195	0.2912	0.1144	0.1770	0.1907	0.2282	0.1933	0.1976	0.1787	0.2135	0.2778	0.2022	0.1781	0.2141
V_6	0.6544	0.6432	0.4086	0.7512	0.4983	1.0000	0.2806	0.1217	0.9781	0.0738	0.5373	0.1032	0.1613	0.2744	0.2433	0.2913	0.2985	0.2192	0.2882	0.4901	0.3086	0.2149	0.2839
V_7	0.1836	0.5102	0.3873	0.0588	0.0763	0.2806	1.0000	0.4488	0.2869	0.2825	0.3014	0.3983	0.3846	0.0879	0.2404	0.1483	0.1685	0.1252	0.0214	0.5839	0.1890	0.1557	0.0077
V_8	0.0797	0.3430	0.0689	0.0297	0.0321	0.1217	0.4488	1.0000	0.1241	0.6789	0.1521	0.1712	0.2678	0.4553	0.4038	0.4833	0.4953	0.3638	0.4783	0.5241	0.5122	0.3567	0.4711
V_9	0.6401	0.6562	0.3866	0.7621	0.4882	0.9781	0.2869	0.1241	1.0000	0.0755	0.5493	0.1055	0.1649	0.2805	0.2487	0.2978	0.3052	0.2241	0.2947	0.5010	0.3155	0.2198	0.2902
V_{10}	0.0483	0.3086	0.0045	0.0175	0.0195	0.0738	0.2825	0.6789	0.0755	1.0000	0.2203	0.1039	0.1624	0.2762	0.2449	0.2932	0.3005	0.2207	0.2902	0.2978	0.3107	0.2164	0.2858
V_{11}	0.3516	0.8756	0.2840	0.5408	0.2912	0.5373	0.3014	0.1521	0.5493	0.2203	1.0000	0.2778	0.3241	0.4213	0.4011	0.4287	0.4193	0.4407	0.4841	0.4824	0.4248	0.4327	0.4429
V_{12}	0.0675	0.1876	0.5608	0.3655	0.1144	0.1032	0.3983	0.1712	0.1055	0.1039	0.2778	1.0000	0.6513	0.4220	0.6023	0.4141	0.3735	0.7134	0.6962	0.0118	0.3720	0.7444	0.5095
V_{13}	0.1056	0.2934	0.7326	0.5633	0.1770	0.1613	0.3846	0.2678	0.1649	0.1624	0.3241	0.6513	1.0000	0.5880	0.8178	0.5611	0.5405	0.7359	0.7504	0.2132	0.5379	0.7506	0.7064
V_{14}	0.1796	0.4989	0.4734	0.4639	0.1907	0.2744	0.0879	0.4553	0.2805	0.2762	0.4213	0.4220	0.5880	1.0000	0.7976	0.9360	0.9193	0.7990	0.8358	0.4368	0.9161	0.7834	0.8588
V_{15}	0.1592	0.4424	0.7293	0.6840	0.2282	0.2433	0.2404	0.4038	0.2487	0.2449	0.4011	0.6023	0.8178	0.7976	1.0000	0.7937	0.8057	0.8063	0.8422	0.4332	0.8012	0.7989	0.8722
V_{16}	0.1906	0.5296	0.4428	0.4514	0.1933	0.2913	0.1483	0.4833	0.2978	0.2932	0.4287	0.4141	0.5611	0.9360	0.7937	1.0000	0.9719	0.7840	0.8342	0.4761	0.9709	0.7710	0.8854
V_{17}	0.1953	0.5428	0.4304	0.4607	0.1976	0.2985	0.1685	0.4953	0.3052	0.3005	0.4193	0.3735	0.5405	0.9193	0.8057	0.9719	1.0000	0.7345	0.7977	0.5160	0.9808	0.7201	0.8709
V_{18}	0.1435	0.3987	0.6075	0.4889	0.1787	0.2192	0.1252	0.3638	0.2241	0.2207	0.4407	0.7134	0.7359	0.7990	0.8063	0.7840	0.7345	1.0000	0.9665	0.2479	0.7293	0.9805	0.8519
V_{19}	0.1886	0.5241	0.6115	0.5463	0.2135	0.2882	0.0214	0.4783	0.2947	0.2902	0.4841	0.6962	0.7504	0.8358	0.8422	0.8342	0.7977	0.9665	1.0000	0.4207	0.8006	0.9696	0.8923
V_{20}	0.3207	0.7733	0.2163	0.5477	0.2778	0.4901	0.5839	0.5241	0.5010	0.2978	0.4824	0.0118	0.2132	0.4368	0.4332	0.4761	0.5160	0.2479	0.4207	1.0000	0.5424	0.2428	0.4715
V_{21}	0.2020	0.5612	0.4272	0.4670	0.2022	0.3086	0.1890	0.5122	0.3155	0.3107	0.4248	0.3720	0.5379	0.9161	0.8012	0.9709	0.9808	0.7293	0.8006	0.5424	1.0000	0.7169	0.8819
V_{22}	0.1407	0.3909	0.6249	0.4922	0.1781	0.2149	0.1557	0.3567	0.2198	0.2164	0.4327	0.7444	0.7506	0.7834	0.7989	0.7710	0.7201	0.9805	0.9696	0.2428	0.7169	1.0000	0.8403
V_{23}	0.1858	0.5162	0.5821	0.5553	0.2141	0.2839	0.0077	0.4711	0.2902	0.2858	0.4429	0.5095	0.7064	0.8588	0.8722	0.8854	0.8709	0.8519	0.8923	0.4715	0.8819	0.8403	1.0000

State-indication variables and State-differentiation variables are useful for process state identification, unfortunately they are not suitable for process monitoring since most of these variable only change dramatically in the beginning of the new state. For process monitoring we need the other types of key variables. The structure for monitoring ShadowPlant startup during one macro-state is shown in Figure 6-4. As shown there, from an operator standpoint, the Regenerator warm-up and catalyst loading state can be divided into four sub-states:

1. Flue gas lighting
2. Increasing temperature
3. Catalyst loading
4. Catalyst transfer

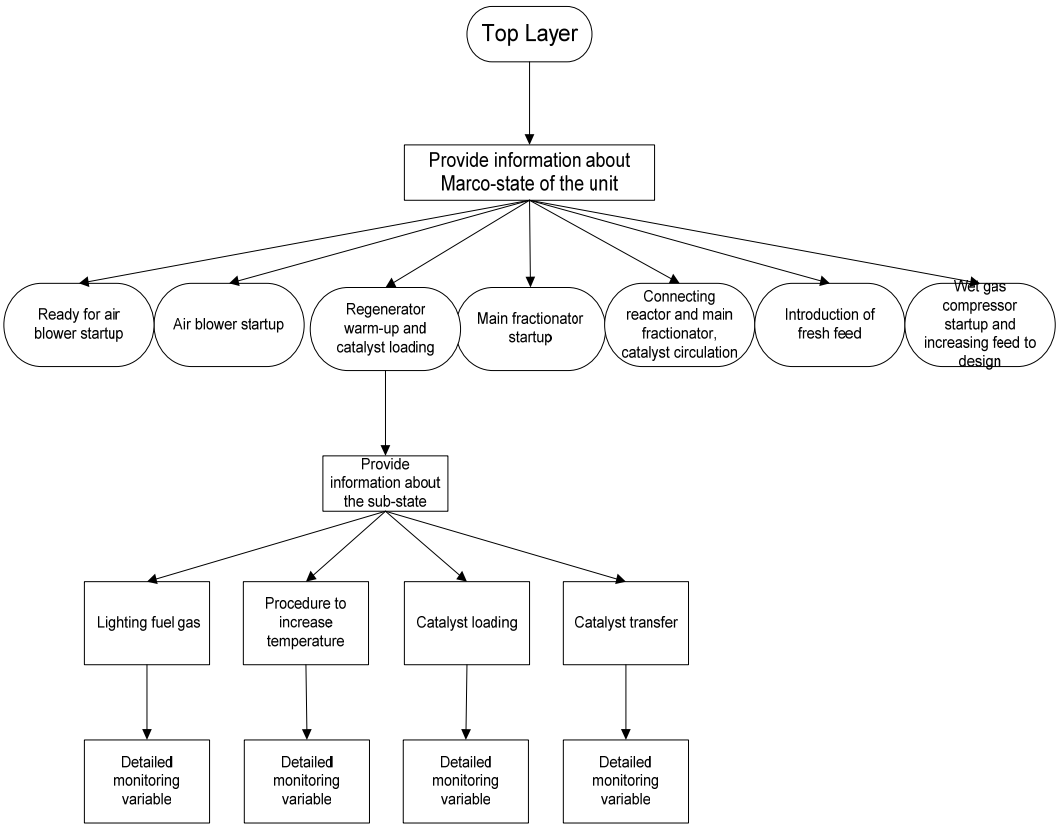
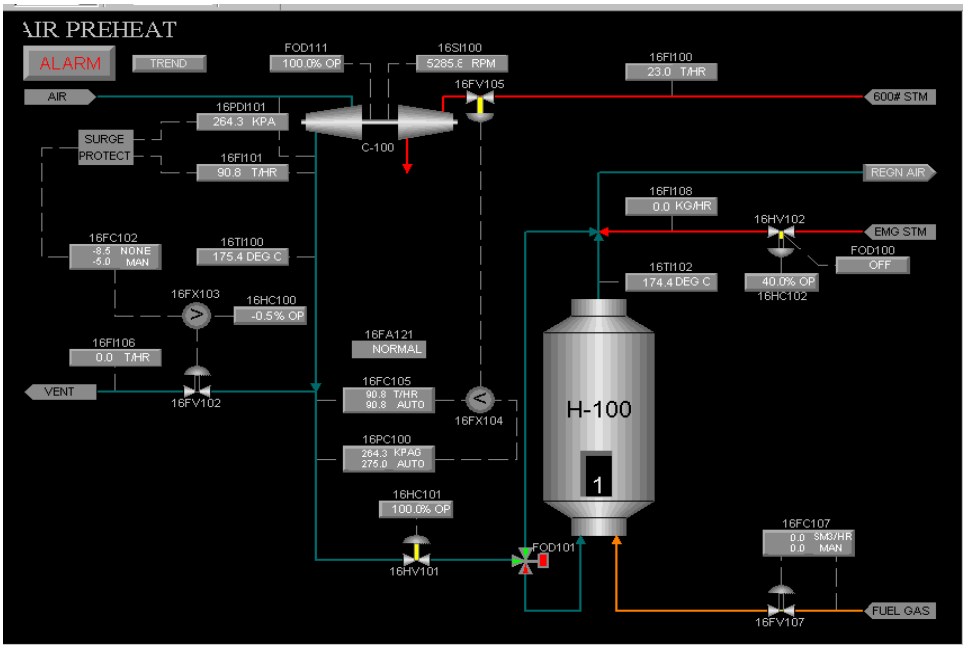
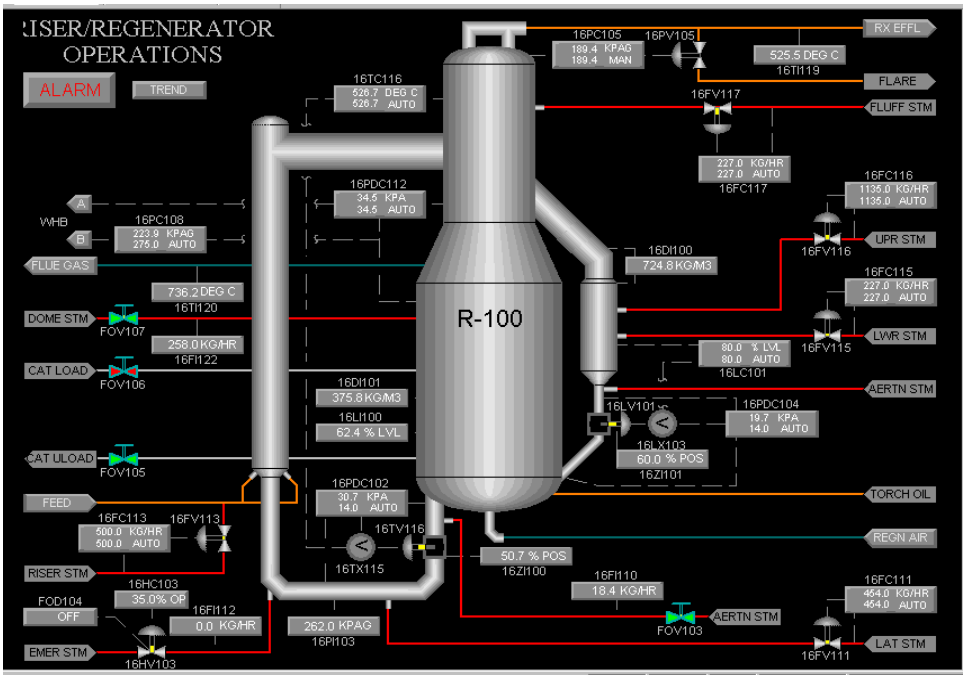


Figure 6-4: An illustration of the structure for monitoring the Regenerator warm-up state

There are four types of key variables in detailed state level – state-progression variable, external-affect variable, Active variable and Important-balance variable. In the following, we use the *Increasing temperature* sub-state to explain how these four types of key variables are identified. The preheater and Riser/Regenerator sectiions shown in Figure 6-5 are involved in this stage.



(a)



(b)

Figure 6-5: (a) Preheater of ShadowPlant, (b) Riser/Regenerator of ShadowPlant

State-progression Variables:

The state-progression coefficient ς was calculated for all the variables. The variables with high values are listed below:

Tag	State-progression Variable	ς
16TI120.PV	R-100 Flue Gas to E-100	0.9032
16TI107.PV	Regenerator Temperature	0.9032
16TI106.PV	R-100 Regenerator #1 2nd Cyclone	0.9032
16TI105.PV	R-100 Regenerator Dense Bed #1	0.7692
16TI104.PV	R-100 Regenerator Dense Bed #2	0.7692
16TI103.PV	R-100 Regenerator Dense Bed #3	0.7692
16PC108.PV	R-100 Regenerator pressure Discharge Valve	0.7158
16FC118.PV	D-100 Condensate Supply	0.6402
16LC102.PV	D-100 Steam Drum Level	0.6111
16PI110.PV	R-100 Flue Gas Stack Pressure	0.4966

Based on these, 16TI107.PV (Regenerator Temperature) is selected as the first State-progress variable. Other state-progression variables are selected based on their correlation with 16TI107.PV. 16LC102.PV (D-100 steam drum level) and 16PC108.PV (R-100 Regenerator pressure) are selected as State-progress variable for this state on this basis.

External-affect Variables:

During regenerator temperature increase sub-state, although the purpose is increasing the regenerator temperature, 16TI120.PV (R-100 Flue gas to E-100) will be directly affected by the regenerator temperature and is therefore a key input to the downstream process. Similarly, 16TI102.PV the air temperature from preheater is the main upstream factor which affects the regenerator temperature. The flowrate 16FC105.PV (air flow to regenerator) and pressure control 16PC108.OP (Flue discharge control valve opening) variables are also included as external affect variables based on the process knowledge and PFD.

16TI120.PV	R-100 Flue gas to E-100
16TI102.PV	H-100 Outlet Temperature
16FC105.PV	Air Flow to Regenerator
16PC108.OP	R-100 Regenerator Pressure Discharge Valve

Active variables:

The following variables were found to have the maximum number of singular points during the sub-state.

16FC107.SP	Fuel Gas to H-100 Preheater
16TI118.PV	R-100 Catalyst from Stripper
16TI122.PV	R-100 Catalyst to Stripper
16PDC112.PV	R-100 Regenerator/Disengager delta P
16PC105	Disengager Start-up Vent

They are therefore selected as the active-variables.

Important-balance variables:

During temperature increase sub-state, the most important balance is the energy balance around the regenerator, specifically the fuel has to be combusted instantaneously. Accumulation of fuel in the regenerator may lead an explosion. So combustion and heat balance need to be monitored carefully. The following are the important-balance variables needed for this purpose.

16FC107.PV	Fuel Gas to H-100 Preheater
16FC105.PV	C-100 Regeneration Air
16TI102.PV	H-100 Outlet Temperature
16TI105.PV	R-100 Regenerator Dense Bed #1
16FC109.PV	R-100 Torch Oil
16AI100.PV	R-100 Flue Gas Stack O2

6.3.2 Process State Identification with State-Specific Key Variables

The online selection of key variables has been used to reduce the computational complexity of the online state identification methods reported in the previous chapters. With a short segment of real-time signals from the key variables, the state of process could be identified correctly. The time cost of the two approaches – with and without state-specific key variables – is shown in Tables 6-5 and 6-6. As seen there, the time cost with state-specific key variables selection can reduce the time cost dramatically by about seven times. This is easy to understand since with the macro-state and sub-state

identification, the search range for dynamic locus analysis is narrowed. With the selection of key variables, the number of variables for state identification is between 20 to 30, which reduces the computation cost and storage space for calculation. The quality of results was also calculated using two methods.

Definition: *normalized difference* η between *X* and *Y* can be calculated as

$$\eta = \frac{\sum_{d=1}^{n} \left| y_{j(i)} - x_i \right|}{n} \tag{6-15}$$

Where x_i is real time signal, $y_{j(i)}$ is the optimal warping reference signal corresponding to x_i. In this chapter, x_i and $y_{j(i)}$ could be one value or vector, if it is a vector, the difference between the two vector following Eq. 6-4.

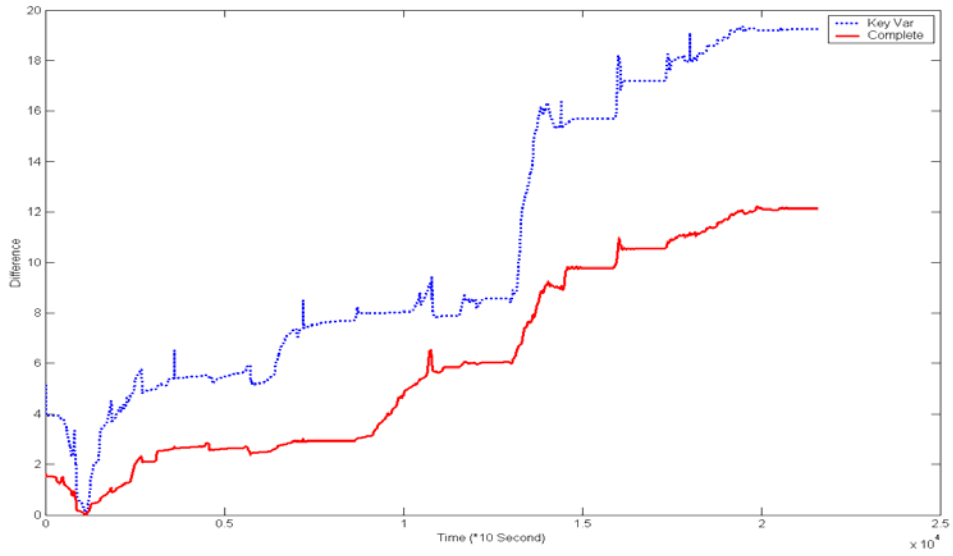
$$\Delta(x_i, y_j) = \sum_{c=1}^{Q} \Delta(x_{(i,c)}, y_{(j,c)}) \tag{6-16}$$

Where Q is dimension of vector x_i , y_j , the dimension must be same for x_i and y_j . $x_{(i,c)}$ is value of vector x_i value at *c* direction. $y_{(j,c)}$ is value of vector y_i value at *c* direction.

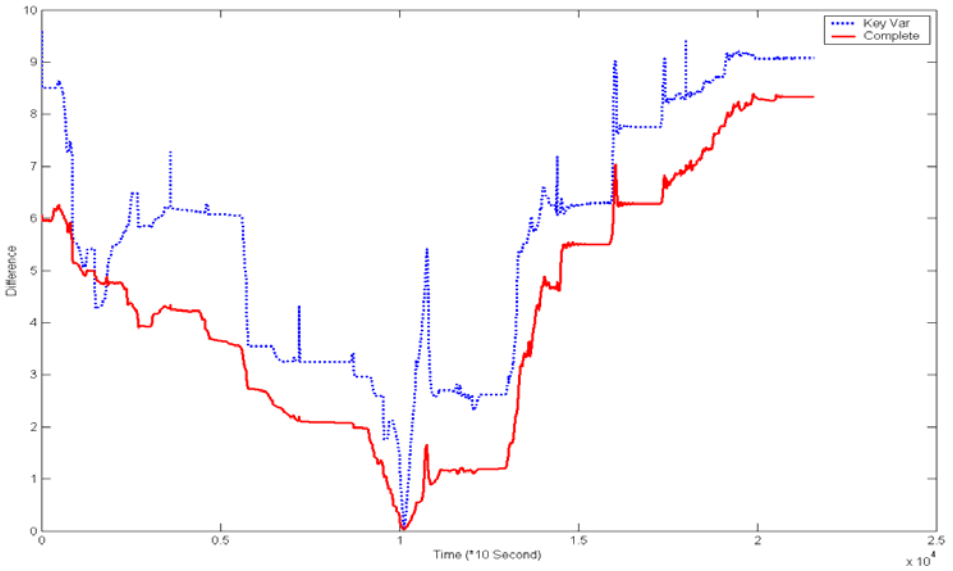
Definition: $Err_{\max}$ is maximum position shift in difference detection results over the total signal length.

If $Err_{\max}$ is large, it means that the consistency of the difference detection results is unsatisfactory. From Table 6-5, the largest $Err_{\max}$ in all segments is only 0.020%, i.e, the detected results from different length segments show good consistency. The improvement of consistency with the key variable is easy to understand. For a complex chemical process, even during process transition, there are a lot of variables that are non-varying. If all variables are used for process state identification, these variables will reduce the sensitivity. With state-specific key variables, for any state, only the selected key variables are used for monitoring, state identification, etc, which improves

sensitivity. Figure 6-6 shows the difference in results while using key variables only as versus using the complete set of variables for identifying the state during air blower startup.



(a)



(b)

Figure 6-6: Comparison between key variables and complete variable for state identification with dynamic locus analysis

Table 6-5: Process state identification performance with state-specific key variables

x_S	y_k			η			Time Cost (CPU Sec)			Sub-State	$Err_{\max}$
	L=10	L=20	L=30	L=10	L=20	L=30	L=10	L=20	L=30		
1100	1140	1137	1137	0.0031	0.0024	0.0019	0.9410	1.0310	2.704	Operation adjustment	0.014%
2100	2125	2125	2128	0.0001	0.0001	0.0002	2.0230	3.5950	5.358	Temperature increasing	0.014%
3100	3118	3120	3118	0.0002	0.0003	0.0005	2.1030	3.6840	5.288	Catalyst loading	0.009%
4100	4097	4097	4097	0.0001	0.0001	0.0001	2.4840	4.3150	6.21	Catalyst loading	0.000%
5100	5121	5121	5121	0.0001	0.0001	0.0001	2.3840	4.2260	6.069	Catalyst loading	0.000%
6100	6095	6095	6095	0.0002	0.0002	0.0002	2.3840	4.2050	6.049	Catalyst loading	0.000%
7100	7238	7233	7227	0.001	0.0010	0.001	1.3920	2.4930	3.596	Catalyst transfer	0.051%
8100	7644	7682	7689	0.0002	0.0002	0.0002	1.4720	2.5730	3.746	Catalyst transfer	0.208%
9100	9095	9099	9102	0.0026	0.0026	0.0026	0.9120	1.5520	2.244	Bring kerosene	0.032%
10100	10158	10158	10158	0.0005	0.0005	0.0005	1.4020	2.4730	3.545	Feed preheater	0.000%
11100	11073	11072	11072	0.0055	0.0044	0.004	1.1320	1.9520	2.804	Start slurry boiler	0.000%
12100	12035	12036	12036	0.0008	0.0009	0.0011	1.1220	1.9620	2.824	Start slurry boiler	0.005%
13100	13130	13130	13130	0.0008	0.0008	0.0007	0.4410	0.4410	0.541	Catalyst circulation	0.005%
14100	14261	14262	14262	0.0035	0.0035	0.0035	1.7730	3.1340	4.526	Introducing feed	0.000%
15100	15001	15001	15001	0.0021	0.0021	0.0021	1.7830	3.1440	4.487	Introducing feed	0.000%
16100	16167	16171	16175	0.0015	0.0015	0.0014	1.5120	2.6730	3.836	Wet gas compressor start up	0.037%
17100	16196	16196	16196	0.0013	0.0013	0.0013	1.8230	3.2140	4.607	Wet gas compressor start up	0.000%
18100	18131	18131	18131	0.0014	0.0014	0.0014	1.8330	3.2240	4.597	Increasing Feed	0.000%
19100	19133	19133	19133	0.0014	0.0015	0.0022	1.8501	3.3213	4.688	Increasing Feed	0.000%
Average	-	-	-	0.0014	0.0013	0.0013	1.619	2.801	4.090		0.020%

Table 6-6: Process state identification performance with all variables

x_S	y_k			η			Time Cost (CPU Sec)			Sub-State	$Err_{\max}$
	L=10	L=20	L=30	L=10	L=20	L=30	L=10	L=20	L=30		
1100	1138	1136	1136	0.0004	0.0003	0.0003	11.146	20.649	31.285	Operation adjustment	0.009%
2100	2129	2129	2129	0.0002	0.0002	0.0002	11.236	20.589	30.855	Temperature increasing	0.000%
3100	3135	3135	3135	0.0002	0.0003	0.0003	11.126	20.609	30.654	Catalyst loading	0.000%
4100	4134	4134	4134	0.0001	0.0001	0.0001	11.126	20.699	30.644	Catalyst loading	0.000%
5100	5132	5132	5132	0.0001	0.0001	0.0001	11.166	20.689	30.784	Catalyst loading	0.000%
6100	6094	6094	6094	0.0002	0.0002	0.0002	11.106	20.810	30.685	Catalyst loading	0.000%
7100	7076	7084	7084	0.0034	0.0034	0.0033	11.146	20.630	30.544	Catalyst transfer	0.037%
8100	8169	8169	8169	0.0004	0.0004	0.0004	11.166	20.580	30.504	Catalyst transfer	0.000%
9100	9087	9087	9092	0.0007	0.0007	0.0007	11.086	20.580	30.544	Bring kerosene	0.023%
10100	10141	10141	10141	0.0010	0.0010	0.0010	11.167	20.650	30.624	Feed preheater	0.000%
11100	11168	11138	11063	0.0039	0.0050	0.0049	11.146	20.740	30.504	Start slurry boiler	0.486%
12100	12033	12034	12034	0.0010	0.0010	0.0011	11.296	20.710	30.534	Start slurry boiler	0.005%
13100	13127	13127	13127	0.0014	0.0014	0.0014	11.126	20.660	30.403	Catalyst circulation	0.000%
14100	14227	14333	14332	0.0052	0.0047	0.0045	11.076	20.600	30.704	Introducing feed	0.491%
15100	15019	14996	14997	0.0011	0.0011	0.0011	11.066	20.570	30.484	Introducing feed	0.106%
16100	16129	16129	16129	0.0015	0.0017	0.0022	11.046	20.700	30.454	Wet gas compressor start up	0.000%
17100	16676	16679	16812	0.0010	0.0010	0.0010	11.096	20.630	30.584	Wet gas compressor start up	0.630%
18100	18128	18128	18128	0.0012	0.0011	0.0011	11.105	20.650	30.514	Increasing Feed	0.000%
19100	19128	19128	19128	0.0013	0.0014	0.0016	11.165	20.750	30.524	Increasing Feed	0.000%
Average	-	-	-	0.0013	0.0013	0.0013	11.1365	20.6576	30.6225		0.094%

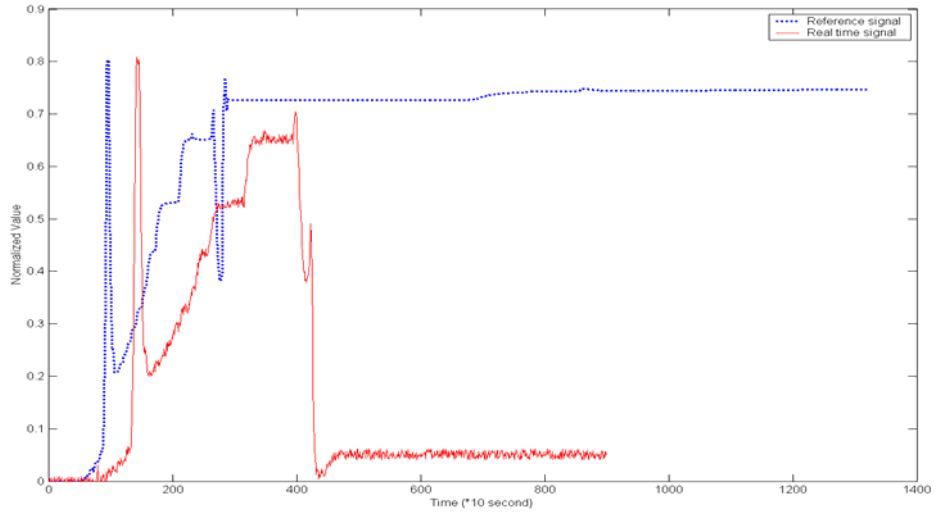
6.3.3 Fault Detection using State-Specific Key Variables

Normal startup of the ShadowPlant takes 40 to 60 h depending on the experience-level of the operator. State-specific key variables can be used for fault detection during this long startup. The overall process-monitoring problem is then decomposed into a much smaller one of monitoring a small fraction of the available variables. In this section, fault detection results from two failures are reported

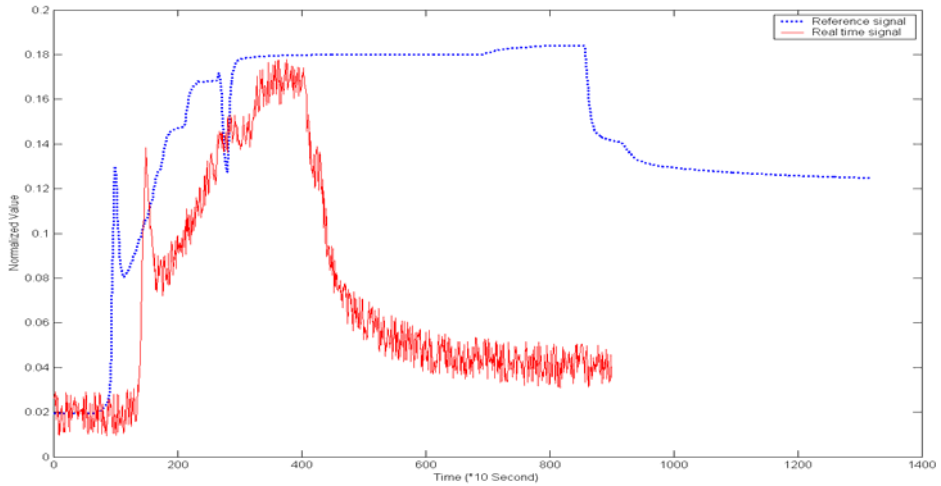
Case 1: Failure of air blower during the 'Increasing regenerator temperature' state

According to the procedure of ShadowPlant startup, the air blower is connected to the air preheater and regenerator after it reaches a steady state. At this time, the system pressure is increased by closing the vent valve. Fuel gas is introduced and lighted to heat the air which is used to warm the regenerator. At 1 hour 10 minutes (= 4200 seconds) during the startup, a failure of the air blower was simulated.

The state-specific key variables identification methodology described above has been first used to identify the key variables during the state of the process startup. The similarity between the real-time and reference signals for these key variables was estimated using the dynamic locus analysis method. The failure could be detected at 4220 second which is 20 seconds after the failure was introduced. The profiles of some key variables during this state are shown in Figure 6-7. The difference between the real-time and reference signals is shown in Figure 6-8. As can be seen there, at the occurrence of the failure, there is a distinct increase in the difference.



(a)



(b)

Figure 6-7: (a) Air blower discharge pressure, and (b) Regenerator temperature during normal and abnormal startup

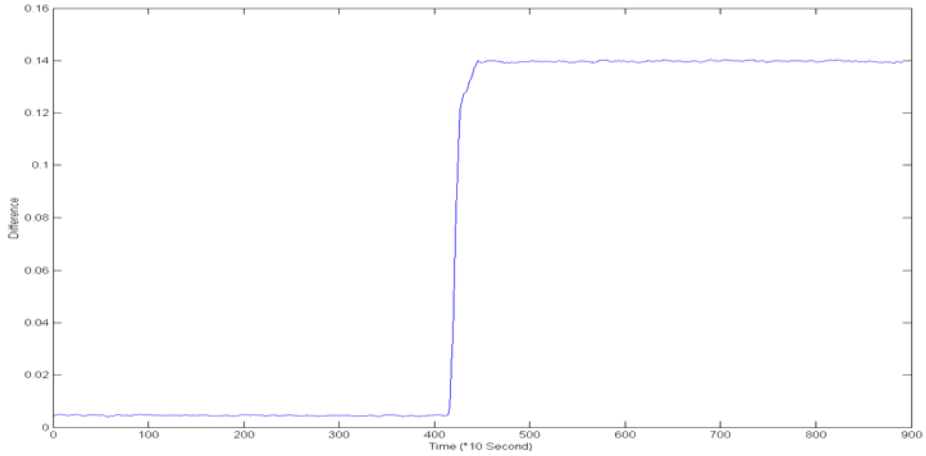
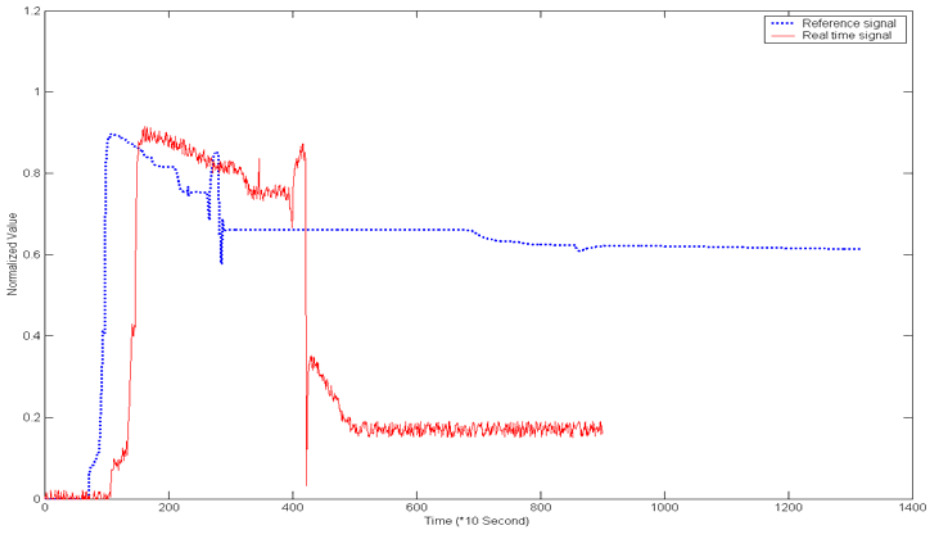


Figure 6-8: Difference between real-time and reference signals for the key variables during Case 1

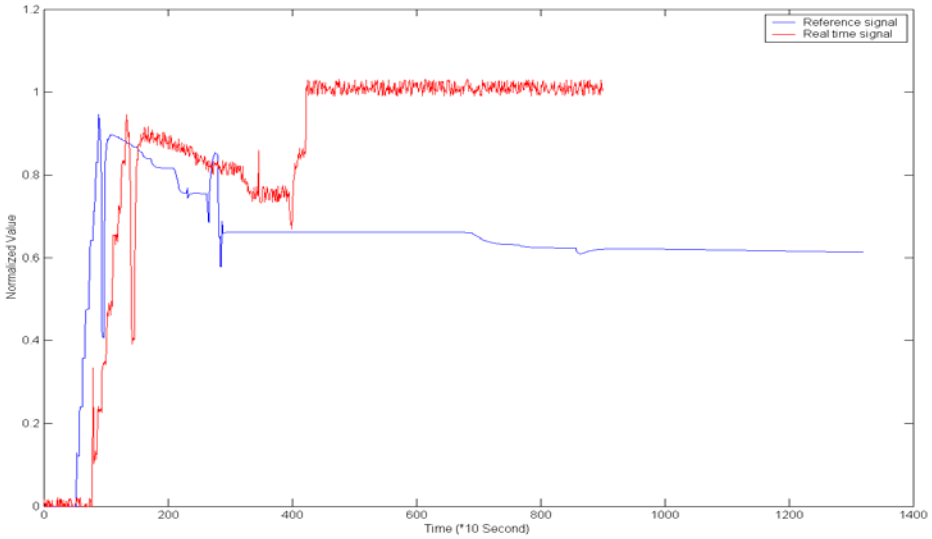
Case 2: Air blower vent valve fails open

A failure of air blower vent valve was simulated at 1 hour 10 minutes. The result of this failure is that the air will be discharged through the vent valve, which leads to malfunction of the whole system since there is no air to warm the regenerator. Some key variables of this state (air blower discharge, air blower to regenerator) respond immediately to the vent valve failure as shown in Figure 6-9.

The difference between real-time key variables and reference key variables is shown in Figure 6-10. The failure can be very sensitively detected by the key variables of this state with a detection delay of 10 seconds. In comparison, Sundarraman and Srinivasan (2003) reported a detection delay of 5 minutes and 2 minutes for the two cases respectively when using monitoring the startup using trend analysis of all the variables. Thus, using state-specific key variables results is more sensitive fault detection.



(a)



(b)

Figure 6-9: Profiles of (a) Air blower to regenerator, and (b) Air blower discharge flow during normal and abnormal runs

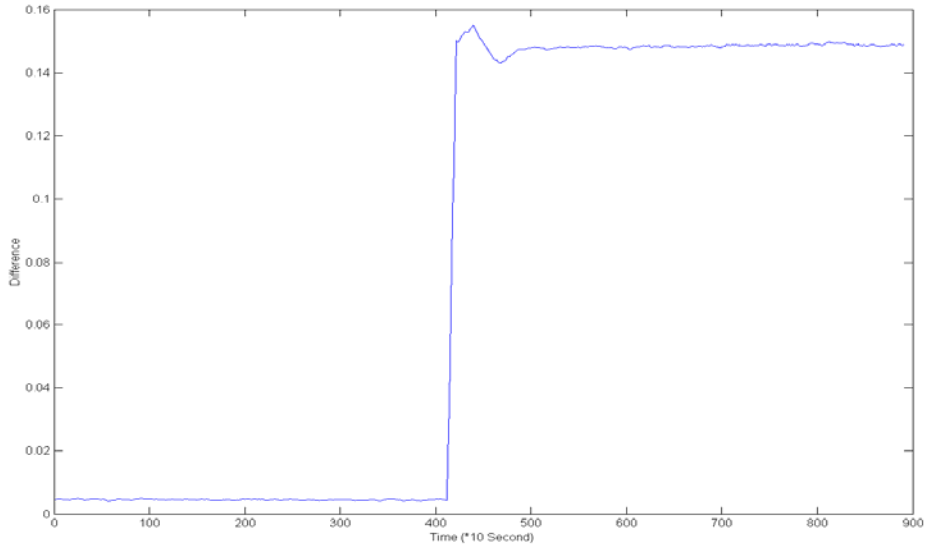


Figure 6-10: Difference between real-time signal and reference signal during Case 2

6.3.4 Synchronization with State-Specific Key Variables Approach

In chemical process state identification, process signal synchronization is one of the challenging problems. It is normal for two similar signals to be slightly different and not match each other perfectly. Synchronization between signals can be achieved through DTW or DLA as described in previous chapters. Dynamic locus analysis can be applied variable-by-variable which will increase the time cost. Alternatively, the

different variables are formed into a vector and warped by the same path. This can reduce the time cost but all variables are forced to be warped with the same path.

In real processes, different sections of a process can be operated at different paces. Within the same section, most variables would show a similar pace because of the interrelations between the variables. During ShadowPlant startup, the five different sections can be operated at different paces. For example, the regenerator/riser startup will not affect the signals in the air blower section. So any signal shift in the regenerator/riser section will not be carried over to the air blower signals. If all of the variables are warped at same pace, it will introduce desynchronization among the different sections, as shown in Figure 6-11.

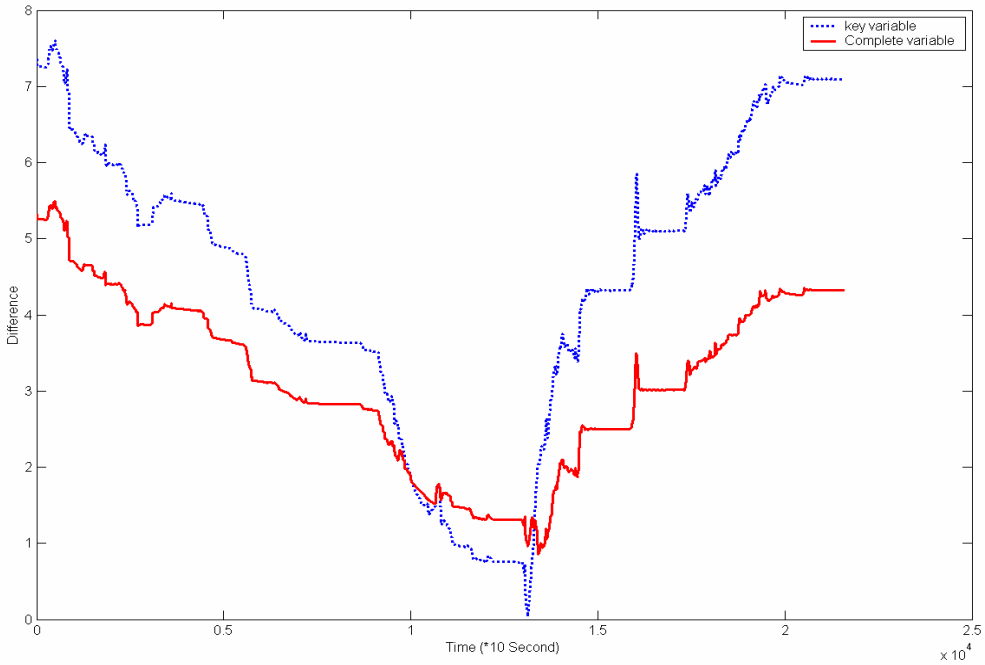


Figure 6-11: State identification using key variables and complete variables

The desynchronization among sections will increase the dynamic locus analysis difference and lead to improper location of the optimal corresponding point in reference signal when all the variables are used in the analysis. Using the key variables alone to synchronize the signals does not suffer this shortcoming as shown in Figure 12. The key variable has a unique minimum point in reference signal at t = 13128 sample, which was manually confirmed to be the correct location. In contrast when all

the variables are used, a latter time t=13413 sample in the reference signal is identified. This is because the key variables are from related sections and have a similar pace.

6.4 Discussion

A new approach for identifying a subset of the process variables, called state-specific key variables, which adequately describe the current processing status, has been developed. This is motivated by the fact that the large-scale and complexity of modern chemical plants makes it difficult both for the operator to constantly monitor all process variables as well as for automated methods which face heavy computational loads. Key variables are groups of variables that have information content with respect to specific operational problems. In this chapter, we describe the state-specific nature of key variables. Different variables maybe deemed as key variables in different states of a process transition. Even during the same state, different plant personnel – operators, safety manager, financial supervisors, etc – may use different groups of variables as key variables. Therefore, key variables selection should be dynamic.

In this chapter, we have defined the key-variable selection problem and proposed a solution methodology. Using a hierarchical structure, key variables for process monitoring are separated into macro-state layer, sub-state layer and detailed variable layer. The problem of defining appropriate key variables in each layer of the hierarchy is addressed through an offline analysis. Different kinds of detailed state key variables are identified by different methods. Some are based on signal analysis while others are based on process analysis.

Once the key variables for each state have been determined, they are stored in a knowledge base and used online for state-specific process monitoring. This state

specific monitoring problem is computationally efficient even for large-scale process and retains the intuitiveness required for operator acceptance.

The key variable selection approach enables manual and automated monitoring to be restricted to a small subset of variables without degradation in safety or quality of results. In many cases, using key variables increases the sensitivity of monitoring and state detection as has been shown with examples of dynamic locus analysis.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

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 thesis, we have proposed new signal comparison-based approaches for online state identification and fault diagnosis during process transitions.

Firstly, we developed a new approach for signal comparison called singular points augmented time warping. This method uses the major landmarks in the signal – called singular points – as the basis to restrict comparison of signals to related segments. It therefore has several advantages over traditional DTW.

1. The comparison of the signals is between the corresponding episodes of the two, it therefore yields the true distance from the operations perspective.
2. Singular points are based on operation's standpoint; so they have physical meaning and help the operator understand the real state of the process.
3. Time warping is required on shorter signal segments and optimal difference between the signals is not essential. This leads to substantial reduction in the computation time especially for long signals as is the case in many chemical engineering applications. This also makes it amenable to real-time applications.
4. This new approach integrates global information in the form of features in the signal to find the best linkage. It is thus more robust when the signals have magnitude differences and/or are substantially unsynchronized.

5. Since the proposed approach is uni-variate, it can work well even in situations where different signals of a process do not progress at the same rate as is common in real cases.

As shown in Chapter 3, the method is robust to both time shifts as well as magnitude differences between the reference and test signals. The time warping algorithm makes the comparison robust to time shifts in the test signal; the optimal linkage of corresponding singular point between test signal and reference signal makes the method robust to magnitude distortions. Even if some singular points are not detected due to process noise, the linkage algorithm ensures an optimal match. The singular point augmented methods also significantly reduce the computational load of signal comparison and are overall better than traditional time warping methods.

Secondly, we proposed an approach, called dynamic locus analysis, which is applicable in cases were the reference and test signals do not have coincident end points. Such situations occur during online signal comparison since the endpoint in the reference signals that corresponds to the real-time signal at any instant is unknown. Dynamic locus analysis is an extension of Smith and Waterman's (1981) discrete sequence comparison algorithm to continuous signals. The approach solves the online signal comparison problem considering two facets – first, the part of the reference signal that corresponds to the real-time signal is identified using a small window of data from real-time sensor measurements; second, since the real-time segment and the corresponding part in the reference signal would not match exactly due to noise and run-to-run differences, the method uses dynamic time warping approaches to account for synchronization differences. With dynamic locus analysis, all segments in the reference signal that can potential match the online signal are considered and the optimal one is identified based on minimal difference. Dynamic locus analysis can be

directly used for multivariate temporal signals and has the computational efficiency needed for real-time application. As can be seen from Chapter 4, the approach is accurate, computationally efficient and robust.

Thirdly, in this thesis, singular points augmented time warping methods have been combined with dynamic locus analysis for online signal comparison. Dynamic locus analysis is first used to identify all the segments of reference signals that adequately match the extreme segments of the real-time signal. In the second state, singular point augmented time warping is used to find the best matching reference for the complete real-time signal. The best matching is then extended in real-time as new measurements are obtained, using a computationally efficient match-point elongation algorithm. The real-time capability of the approach arises from the computational efficiency of the dynamic locus analysis and the singular point augmented time warping approaches. As shown in Chapter 5, the method is robust to noise as well as run-to-run differences.

The above methods were extensively tested using data collected from different kinds of agile operation – startup of a simulated FCCU, startup of a lab-scale distillation column, multi-mode operation of the Tennessee Eastman challenge process, and a lab-scale fed-batch fermentation process. Robustness to parameter settings and measurement noise were studied in all cases. From these studies, it was shown that most of the parameter settings can be used across variables and case studies.

Finally, in Chapter 6, a new approach for identifying a subset of the process variables, called key variables, which adequately describe the current processing state has been developed. This is motivated by the fact that the large-scale and complexity of modern chemical plants makes it difficult both for the operator to constantly monitor all process variables as well as for automated methods which face heavy

computational loads. The key variable selection approach enables manual and automated monitoring to be restricted to a small subset of variables without degradation in safety or quality of results. In many cases, uses key variables increases the sensitivity of monitoring and state detection as has been shown with examples of dynamic locus analysis.

7.2 Suggestions for Future Work

While the developments in this thesis provide a new basis for signal processing based process state identification and fault diagnosis, they can be further extended in the future along the following directions.

1. This thesis has explored the use of singular points only in the context of signal comparison. However, the concept of singular points introduced here can be used for other applications as well – examples, include data compression, operator decision support for process monitoring, fault detection, fault identification, and recovery planning – even without explicit signal comparison since the singular points and the singular episodes themselves encapsulate all the essential information from the signal.
2. The computational load and robustness of singular points identification can be improved. This particularly applies to trend change points in complex, noisy signals. It has been our observation that the algorithm for trend change point identification proposed in this thesis results in a large number of hits – many of which would not be intuitively considered as such by an experience operator.
3. The concept of singular point has been defined solely for uni-variate signals. Its extension to multivariate signals would be useful.
4. The dynamic locus analysis method as described in Chapter 4 is a multi-variate approach. But it assumes that all the signals are either synchronized to start or are

desynchronized to the same levels. This assumption allows the same time warping to be applied to all the signals. In large-scale processes, this assumption may not be valid and different variables may have different levels of desynchronization. Further work is needed to extend the dynamic locus analysis to such situations.

5. The dynamic selection of state-specific key variables could have a number of applications although it has been explored only in the context of process monitoring in this thesis. One such example is state-specific alarm management. Currently, a flood of false alarms arise especially during process transitions. The dynamic key variable selection can be used to systematically identify critical as well as nuisance alarms.
6. The key variable selection method proposed in this thesis is rudimentary and establishes only a proof-of-concept. More sophisticated methods which require less user expertise and are less data intensive should be developed.

Bibliography

Arbel, A., Z. Huang, I. H. Rinard and R. Shinnar. Dynamic and control of fluidized catalytic crackers. 1. Modeling of the current generation of FCC's, *Industrial Engineering Chemistry Research*, 34, pp 1228-1243. 1995.

Arbel, A., Z. Huang, I. H Rinard. and R. Shinnar. Dynamic and control of fluidized catalytic crackers. 2. Multiple steady states and instabilities, *Industrial Engineering Chemistry Research*, 34, pp 3014-3026. 1995.

Arbel, A., Z. Huang, I. H. Rinard and R. Shinnar. Dynamic and control of fluidized catalytic crackers. 3. Designing the control system: choice of manipulated and measured variables for partial control*, Industrial Engineering Chemistry Research*, 35, pp 2215-2233. 1995.

Bakshi, B. R. and G. Stephanopoulos. *Intelligent Systems in Process Engineering-Paradigms from Design to Operations*. San Diego: Academic Press. 1996.

Bagajewica, M. J. Design and Retrofit of Sensor Networks in Process Plants, *AIChE Journal*, 43(9), pp 2300-2306. 1997.

Basseville, M. Detecting Changes in Signals and Systems, Automatica, 24(3), pp 309-326. 1988.

Bhushan, M. and R. Rengaswamy. Design of Sensor Network Based on the Signed Directed Graph of the Process for Efficient Fault Diagnosis, *Industrial and Engineering Chemistry Research*, 39(4), pp 999-1019. 2000.

Bhushan, M. and R. Rengaswamy. Comprehensive Design of a Sensor Network for Chemical Plants Based on Various Diagnosability and Reliability Criteria. 1. Framework, *Industrial and Engineering Chemistry Research*, 41(7), pp 1826-1839. 2002.

Bhushan, M. and R. Rengaswamy. Comprehensive Design of a Sensor Network for Chemical Plants Based on Various Diagnosability and Reliability Criteria. 2. Applications, *Industrial and Engineering Chemistry Research*, 41(7), pp 1840-1860. 2002.

Bhushan, M. and R. Rengaswamy. Comprehensive Design of a Sensor Network for Chemical Plants Based on Various Diagnosability and Reliability Criteria. 1. Frame work, *Industrial and Engineering Chemistry Research*, 41(7), pp 1826-1839. 2002.

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

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

Chiang, L. H., E. L. Russell and R. D. Braatz. Fault disgnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principle component analysis, Chemometrics and Intelligent Laboratory Systems, 50, pp.243-252. 2000.

Chiang, L H., E.L. Russell, and R.D. Braatz. Fault detection and diagnosis in industrial systems, Springer. 2001.

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

Davis, R. Diagnosis Reasoning Based on Structure and Behavior, Artificial Intelligence, 24, pp 347-410. 1984.

Downs, J. J. and E. F. Vogel. A plant-wide industrial process control problem, *Computers and Chemical Engineering*, 17(3), pp.245-255. 1993.

Elman, J. L. Finding Structure in Time, *Cognitive Science*, 14, pp 179-211. 1990.

Himmelblau, D. M. *Fault Detection and Diagnosis in Chemical and Petrochemical Processes*, Elsevier Predd, Amsterdam. 1978.

Hanratty, P. J. and B. Joseph. Decision-making in Chemical Engineering and Expert Systems: Application of the Analytic Hierarchy Process to Reactor Selection, *Computers & Chemical Engineering*, 16(9), pp 849-860. 1992.

Honeywell. Guide to the Fluidized Catalytic Cracking Unit Standard Model. 2000.

Hyvarinen, A. and Oja. Independent component analysis: algorithms and applications, Neural Networks, 13, pp 411-430. 2000.

Itakura, F. Minimum Prediction Residual Principle Applied to Speech Recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-23(1), pp.67-72. 1975.

Jain, A. K., R. P. W. Duin and J. C. Mao. Statistical pattern recognition: A review, *IEEE Transactions on pattern analysis and machine intelligence*, 22(1), pp 4- 37. 2000.

Johannesmeyer, M. C., A. Singhal and D. E. Seborg. Pattern Matching in Historical Data, *AIChE Journal,* 48 (9), pp.2022-2038. 2002.

Kano, M., S. Hasebe, I. Hashimoto and H. Ohno. A new multivariate statistical process monitoring method using principal component analysis, *Computers and Chemical Engineering*, 25(7-8), pp1103-1113. 2001.

Kassidas, A. Fault detection and diagnosis in dynamic multivariable chemical processes using speech recognition methods, Ph.D Thesis, McMaster University. 1997.

Kassidas, A., J. F. MacGregor and P. A. Taylor. Synchronization of batch trajectories using dynamic time warping, *AIChE Journal,* 44 (4), pp.864-875. 1998.

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

Kothare, M. V., R. Shinnar, I. Rinard and M. Morari. On defining the partial control problem: concepts and examples, *AICHE Journal*, 46(12), pp 2456-2474. 2000.

Kotz, S. and N. L. Johnsan. *Encyclopedia of Statistical Sciences*, 9, pp135-136, John Wiley and Sons. 1988.

Kretsovalis, A. and R. S. H. Mah. Effect of Redundancy on Estimation Accuracy in Process Data Reconciliation, Chemical Engineering Science, 42(9), pp 2115-2121. 1987.

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

Li, R. F. and X. Z. Wang. Dimension reduction of process dynamic trends using independent component analysis, *Computers & Chemical Engineering*, 26(3), pp 467-473. 2002.

Li, Y., C. L. Wen, Z. Xie and X. H. Xu. Synchronization of batch trajectory based on multi-scale dynamic time warping. *International Conference on Machine Learning and Cybernetics*, 4, pp. 2403-2408. 2004.

Mallet, S. and W. L. Hwang. Singularity detection and processing with wavelets, *IEEE Transactions on Information Theory*, 38(2), pp.617-643. 1992.

Mallet, S. and S. Zhong. Characterization of signlas from multiscale edges, *IEEE Transactions on pattern analysis and machine intelligence*, 14(7), pp.710-732. 1992.

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

Muthuswamy, K. and R. Srinivasan. Phase-based supervisory control for fermentation process development, *Journal of Process Control*, 13(5), pp.367-382. 2003.

Needleman, S. B. and C. D. Wunsch. *Journal Molecular Biology,* 48, pp 443-453. 1970.

Nielsen, N. V., J. M. Carstensen and J. Smedsgarrd. Aligning of single and multiple wavelength chromatographic profile for chemometric data analysis using correlation optimized warping, *Journal of Chromatography A*, 805, pp.17-35. 1998.

Nomikos, P. and J. F. MacGregor. Monitoring batch processes using multiway principle component analysis, *AIChE Journal,* 40 (8), pp.1361-1375. 1994.

Nimmo, I. Start up Plants Safely, *Chemical Engineering Progress*, 89 (12), pp.66-69. 1993.

Pravdova, V., B. Walczak and D. L. Massart. A comparison of two algorithms for warping of analytical signals, *Analytica Chimica Acta*, 456, pp.77-92. 2002.

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

Rao, M., R. Dong and V. Mahalec. Intelligent system for safe process startup, *Engineering Application Artificial Intelligence*, 7(4), pp349-360. 1994.

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

Russell, L. H., L. H. Chiang and R. D. Braatz. Fault detection in industrial processes using canonical variate analysis and dynamic principle component analysis, *Chemometrics and Intelligent laboratory Systems*, 51, pp81-93. 2000.

Saaty, T. L. *The Analytic Hierarchy Process*, McGraw-Hill. 1980.

Sadeghbeigi, R. *Fluid Catalytic Cracking Handbook*, Gulf Publishing Company, Houston Texas. 2000.

Sankoff, D. and J. B. Kruskal. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley Publishing Company Inc. 1983.

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

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

Singhal, A. and D. E. Seborg. Matching patterns from historical data using PCA and distance similarity factors, *Proceedings of the American Control Conference*, pp. 1759-1764. 2001.

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

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

Srinivasan, R., C. Wang, W. K. Ho and K.W. Lim. Dynamic PCA based methodology for clustering process states in agile chemical plants, *Industrial and Engineering Chemistry Research*, 43 (9), p2123-2139. 2004.

Srinivasan, R., C. Wang, W. K. Ho and K.W. Lim. Context-based recognition of process states using neural networks, Chemical Engineering Science, 60(4), pp935-949. 2005.

Srinivasan, R., P. Viswanathan, H Vedam. and A. Nochur. A framework for managing transitions in chemical plants, *Computers and Chemical Engineering*, 29, p305-322. 2005.

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

Undey, C. and A. Cinar. Statistical monitoring of multistage, multiphase batch processes, *IEEE Control Systems Magazine*, 22(5), pp.40-52. 2002.

Vedam, H. and V. Venkatasubramanian. Signed Digraph Based Multiple Fault Diagnosis, *Computers & Chemical Engineering*, (21), pp S655-660. 1997.

Venkatasubramanian, V., R. Rengaswamy, S. N. Kavuri and K. Yin. A review of process fault detection and diagnosis, *Computer and Chemical Engineering*, 27(3), pp327-346. 2003.

Waterman, M. S. and M. Eggert. A New Algorithm for Best Subsequence Alignments with Application to tRNA-rRNA Comparisons, J.Mol. Biol., (197), 723-728. 1987.

Webb, A. R. Statistical pattern recognition, West Sussex: John Wiley & Sons. 2002.

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

Young, S. O., J. M. Kyung and S. Y. En. Fault diagnosis based on weighted symptom tree and pattern matching, *Ind. Eng. Chem. Re*s., 36(7), pp.2672-2678. 1997.

Yuan, Bo and G. Klir. Data-driven identification of key variables, Intelligent Hybrid Systems Fuzzy logic, Neural Networks, and Genetic Algorithms, Kluwer Academic Publishers, Boston. 1997.

Zheng, A., R. V. Mahajanam and J. M. Douglas. Hierarchical Procedure for Plantwide Control System Synthesis, *AIChE Journal*, 45(6), pp 1225-1265. 2000.

Author's Publications

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

Srinivasan R. and Qian M. Online signal comparison using singular points augmented time warping, *Industrial and Engineering Chemistry Research*, under review, 2006.

Srinivasan R. and Qian M. Online fault diagnosis and state identification during process transition using dynamic locus analysis, *Chemical Engineering Science*. under review, 2006.

Srinivasan R. and Qian M. Selecting state-specific key variables, Submitted, 2006.

Qian M. and Srinivasan R., Online Monitoring of Process States using Dynamic Programming and Dynamic Time Warping, Presented in the AIChE annual meeting, Indianapolis, Nov 3–8. 2002.

Qian M. and Srinivasan R., Online Identification of Process Transitions using Dynamic Programming and Dynamic Time Warping, Presented in the 4th Asian Control Conference, Singapore, Sep 25-27. 2002.

Qian M. and Srinivasan R., Classifying process transitions using dynamic time warping, Presented in the Regional Symposium on Chemical Engineering, Kuala Lumpur, Malaysia, Oct 28-30. 2002.

Qian M. and Srinivasan R., Indexing Process Modes and Transitions Using Singular Points, Accepted for presentation in Regional Symposium on Chemical Engineering 2001, 29-31 Oct, Bandung, Indonesia. 2001.

Appendix A:

Algorithm for Sharp changes Detection:

```
%%%% This file is using for finding the place where the signal start change quickly where it stop change quickly
%%% THD is jump-threshold; Duration is inspection window; signal is the uni-variate signal

function [A4,B4]=sharpChange(signal,THD)
Duration=8;
threshold=THD;

%%%% get the change mark for short period
excessValue = mod(length(signal),Duration);
noOfInputs = (length(signal)-excessValue)/Duration;
reshapedInput = reshape(signal(1:end-excessValue),Duration,noOfInputs);
nonFlatTrends = find((max(reshapedInput)- min(reshapedInput)) > threshold);
flatTrends = find((max(reshapedInput)- min(reshapedInput)) <= threshold);
trend(:,nonFlatTrends)=4;
trend(:,flatTrends)=1;

%%%% get the quick transition start at 8*n
Q=diff(signal);
QQ=find(abs(Q)>threshold);
if length(QQ)>0
   P=find(mod(QQ,Duration)==0);
   if P
      Change=QQ(P)/Duration+1;
      if Change<noOfInputs
         trend(:,Change)=4;
      end
   end
end

%%%% find the real sharp Change period
Position=find(trend==4);
if length(Position)==0
   startPoint=[];
   endPoint=[];
elseif length(Position)==1
   startPoint=Position(1)*Duration-(Duration-1);
   endPoint=Position(1)*Duration;
else
   left=diff(Position(1:end));
   keyPositionLeft=find(left>Duration);
   startPoint(1)=Position(1)*Duration-(Duration-1);
   if length(keyPositionLeft)>0
      startPoint(2:length(keyPositionLeft)+1)=Position(keyPositionLeft+1)*Duration-(Duration-1);
      endPoint(1:length(keyPositionLeft))=Position(keyPositionLeft)*Duration;
      endPoint(end+1)=Position(end)*Duration;
   else
      endPoint(1)=Position(end)*Duration;
   end
end

%%% % find the exact change point in inspection window
if startPoint
   for i=1:length(startPoint)
```

```
      P=startPoint(i);
      for j=1:Duration-1
         if max(signal(P-j:P-j+Duration-1))-min(signal(P-j:P-j+Duration-1))>threshold
            if abs(signal(P-j+1)-signal(P-j))>threshold/5
               startPoint(i)=startPoint(i)-j;
            end
         end
      end
      C=signal(startPoint(i):startPoint(i)+Duration);
      C1=diff(C);
      C2=find(abs(C1)>threshold/5);
      if isempty(C2)
         C2=0;
      end
      startPoint(i)=startPoint(i)+C2(1)-1;

      Q=endPoint(i);
      for j=1:Duration-1
         if max(signal(Q-Duration-1+j:Q+j))-min(signal(Q-Duration-1+j:Q+j))>threshold
            if abs(signal(Q+j)-signal(Q+j-1))>threshold/5
               endPoint(i)=endPoint(i)+j;
            end
         end
      end
      D=signal(endPoint(i)-Duration:endPoint(i));
      D1=diff(D);
      D2=find(abs(D1)>threshold/5);
      if isempty(D2)
         D2=0;
      end
      endPoint(i)=endPoint(i)-(Duration-D2(end));
   end
end
A4=startPoint;
B4=endPoint;
```

Extrema points Detection

Algorithm for Maxima Detection

```
%%%% This file is using for find the extreme points (Maxima) in a signal
%%%% There too many local extreme points in a noise signal which need remove from identification
%%% THD is threshold; signal is the uni-variate signal

function [A1,B1]=maxPoint(signal,THD)
threshold=THD;
keepMax=[];

%%%% find the max points in signal
t=1;
[a{t},b{t}]=maxFinder(signal);
positionMax=b{t};
if length(positionMax)>2
    keep=[];
    %%%% review all the local maxima points
    for i=1:length(positionMax)-1
        if isempty(keepMax)&isempty(keep)
            D1=signal(positionMax(i))-min(signal(1:positionMax(i)));
           %%%% Make sure left side has difference with the smallest point
            if D1<threshold/2
            else
                D2=signal(positionMax(i))-min(signal(positionMax(i):positionMax(i+1)));
                keep=positionMax(i);
                if D2>threshold
                    keepMax=[keepMax positionMax(i)];
                    keep=[];
                elseif signal(positionMax(i+1))>signal(keep)
                    keep=positionMax(i+1);
                end
            end
        elseif isempty(keepMax)&keep
            D3=signal(keep)-min(signal(keep:positionMax(i+1)));
            %%%% ensure the signal between two neighbor maxima points has a minima point
            if D3>threshold/2
                keepMax=[keepMax keep];
                keep=[];
```

```
        elseif signal(positionMax(i+1))>signal(keep)
            keep=positionMax(i+1);
        end
    elseif keepMax&isempty(keep)
        D4=signal(positionMax(i))-min(signal(keepMax(end):positionMax(i)));
        if D4<threshold/2
        else
            D5=signal(positionMax(i))-min(signal(positionMax(i):positionMax(i+1)));
            keep=positionMax(i);
            if D5>threshold/2
                keepMax=[keepMax positionMax(i)];
                keep=[];
            elseif signal(positionMax(i+1))>signal(keep)
                keep=positionMax(i+1);
            end
        end
    elseif keepMax&keep
        D6=signal(keep)-min(signal(keep:positionMax(i+1)));
        if D6>threshold/2
            keepMax=[keepMax keep];
            keep=[];
        elseif signal(positionMax(i+1))>signal(keep)
            keep=positionMax(i+1);
        end
    end
  end
  if keep
    D=signal(keep)-min(signal(keep:end));
    if D>threshold/2
        keepMax=[keepMax keep];
    end
  end
end
A1=keepMax';
B1=signal(A1);
```

Algorithm for Minima Detection

```
%%%% This file is using for find the extreme points (minima) in a signal
%%%% There too many extreme points in a noise signal which need remove from identification
%%% THD is threshold; signal is the uni-variate signal

function [A2,B2]=minPoint(signal,THD)
threshold=THD;

keepMin=[];
%%%% find the max points in signal
t=1;
[a{t},b{t}]=minFinder(signal);
positionMin=b{t};
if length(positionMin)>2
   keep=[];
   for i=1:length(positionMin)-1
      if isempty(keepMin)&isempty(keep)
         D1=abs(signal(positionMin(i))-max(signal(1:positionMin(i))));
        %%%% Make sure left side has difference with the minima point
         if D1<threshold/2
         else
            D2=abs(signal(positionMin(i))-max(signal(positionMin(i):positionMin(i+1))));
            keep=positionMin(i);
            if D2>threshold
               keepMin=[keepMin positionMin(i)];
               keep=[];
            elseif signal(positionMin(i+1))<signal(keep)
               keep=positionMin(i+1);
            end
         end
      elseif isempty(keepMin)&keep
         D3=abs(signal(keep)-max(signal(keep:positionMin(i+1))));
         if D3>threshold/2
            keepMin=[keepMin keep];
            keep=[];
         elseif signal(positionMin(i+1))<signal(keep)
            keep=positionMin(i+1);
         end
      elseif keepMin&isempty(keep)
```

```
        D4=abs(signal(positionMin(i))-max(signal(keepMin(end):positionMin(i))));
        if D4<threshold/2
        else
            D5=abs(signal(positionMin(i))-max(signal(positionMin(i):positionMin(i+1))));
            keep=positionMin(i);
            %%%% ensure the signal between two neighbor minima points has a maximapoints
            if D5>threshold/2
                keepMin=[keepMin positionMin(i)];
                keep=[];
            elseif signal(positionMin(i+1))<signal(keep)
                 keep=positionMin(i+1);
            end
        end
    elseif keepMin&keep
        D6=abs(signal(keep)-max(signal(keep:positionMin(i+1))));
        if D6>threshold/2
            keepMin=[keepMin keep];
            keep=[];
        elseif signal(positionMin(i+1))<signal(keep)
            keep=positionMin(i+1);
        end
    end
  end
  if keep
    D=abs(signal(keep)-max(signal(keep:end)));
    if D>threshold/2
        keepMin=[keepMin keep];
    end
  end
end
A2=keepMin';
B2=signal(A2);
```

Algorithm for Trend change points Detection

```
%%%% This file is using for finding the trend change points during process transition
%%%% Trend change point is a point where process stable trend changed
%%%% THD is threshold; distance is the stable neighborhood window, signal is the uni-variate signal


function [A5, B5]=trendChange(signal,THD)
threshold=THD;
distance=100;
Number=fix(distance/10);
limit=1.0e-2;
keep=[];
i=distance
while i<length(signal)-distance
   %%% left checking
   yy=signal(i-distance+1:i);
   [Rate,Start,R]=Regessive(yy);
   DD(1)=Start;
   DD(2)=Rate;
   Flag1=1;
   if R>(1-limit)|abs(Rate<limit)
      B=DD(1)+DD(2).*([1:distance]./distance);
      STD=sqrt(sum((yy-B').^2)/(distance-1));
      %%%% check any points out 3 standard deviation
      if isempty(find(yy<B'-3*STD|yy>B'+3*STD))&STD<limit
         Flag1=0;
      elseif max(abs(yy-B'))<threshold/5
         Flag1=0;
      end
   end
   if Flag1==0
      D1=DD(1)+DD(2).*(distance/(distance+1));
      %%%% check whether the next point out 3 standard deviation
      if signal(i+1)<D1-3*STD|signal(i+1)>D1+3*STD
         if abs(signal(i+1)-D1)>threshold/5;
            QQ=abs(diff(signal(i-Number+1:i+1)));
            QQ1=find(QQ>limit);
            QQ2=QQ1(1);
            keep=[keep i-Number+QQ2];
            i=i+distance;
         else
            Pre=signal(i)+DD(2).*([1:Number]./distance);
            PART=signal(i+1:i+Number)-signal(i);
            if PART(1:end)>limit
               keep=[keep i];
               i=i+distance;
            elseif PART(1:end)<-limit
               keep=[keep i];
               i=i+distance;
            end
         end
      end
   end

   %%% right checking
   yy=signal(i:i+distance-1);
   [Rate,Start,R]=Regessive(yy);
   DD(1)=Start;
   DD(2)=Rate;
```

```
   Flag2=1;
   if R>(1-limit)|abs(Rate)<limit
      B=DD(1)+DD(2).*([1:distance]./distance);
      STD=sqrt(sum((yy-B').^2)/(distance-1));
      if isempty(find(yy<B'-3*STD|yy>B'+3*STD))&STD<limit
         Flag2=0;
      elseif max(abs(yy-B'))<threshold/5
         Flag2=0;
      end
   end
   if Flag2==0
      D1=DD(1);
       %%%% check whether the next point out 3 standard deviation
      if signal(i-1)<D1-3*STD|signal(i-1)>D1+3*STD
         if abs(signal(i-1)-D1)>threshold/5;
            Q=abs(diff(signal(i-1:i-1+Number)));
            Q1=find(Q>limit);
            Q2=Q1(end);
            keep=[keep i+Q2-1];
            i=i+distance;
         else
            Pre=signal(i)+DD(2).*([-Number:-1]./distance);
            PART=signal(i-Number:i-1)-.Pre;
            if PART(1:end)>limit
               keep=[keep i];
               i=i+distance;
            elseif PART(1:end)<-limit
               keep=[keep i];
               i=i+distance;
            end
         end
      end
   end
   i=i+1;
end

A5=keep;
B5=signal(keep);
```