FAULT DETECTION AND DIAGNOSIS IN DYNAMIC MULTIVARIABLE CHEMICAL PROCESSES USING SPEECH RECOGNITION METHODS

By

ATHANASSIOS KASSIDAS, M. ENG.

A Thesis

Submitted to the School of Graduate Studies

In Partial Fulfillment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

FAULT DIAGNOSIS USING SPEECH RECOGNITION METHODS

Of the shop

He wrapped them carefully, neatly
in costly green silk.

Roses of ruby, lilies of pearl
violets of amethyst. As he himself judged,

as he wanted them, they look beautiful to him; not as he saw,
or studied them in nature. He will leave them in the safe,

a sample of his daring and skillful craft.
When a buyer enters the shop

he takes from the cases other wares and sells -superb jewels-
bracelets, chains, necklaces and rings.

Constantine P. Cavafy (1913)

Του μαγαζιού

Τα τύλιξε προσεκτικά, με τάξη
σε πράσινο πολύτιμο μετάξι.

Από ρουμπίνια ρόδα, από μαργαριτάρια κρίνοι,
από αμεθύστους μενεξέδες. Ως αυτός τα κρίνει,

τα θέλησε, τα βλέπει ωραία, όχι όπως στην φύση
τα είδεν ή τα σπούδασε. Μες στο ταμείον θα τ' αφήσει,

δείγμα της τολμηρής δουλειάς του και ικανής.
Στο μαγαζί σαν μπει αγοραστής κανείς

βγάζει απ' τες θήκες άλλα και πουλεί -περίφημα στολίδια-
βραχιόλια, αλυσίδες, περιδέραια, και δακτυλίδια.

Κωνσταντίνος Π. Καβάφης (1913)

Αφιερώνεται στην οικογένειά μου
Αριστομένη, Μαίρη και Ιωάννα Κασσίδα
Με όλη μου την αγάπη και το σεβασμό.

DOCTOR OF PHILOSOPHY (1997) McMASTER UNIVERSITY
(Chemical Engineering) Hamilton, Ontario

TITLE: Fault Detection and Diagnosis in Dynamic Multivariable Chemical Processes Using Speech Recognition Methods

AUTHOR: Athanassios Kassidas, M. Eng., (McMaster University)

SUPERVISORS: Professor. J. F. MacGregor and Professor P.A. Taylor

NUMBER OF PAGES: xiv, 230

ABSTRACT

Fault Detection and Diagnosis have become important topics in the process industries. The off-line diagnosis of past transient upsets can lead to important process or operation modifications that can improve the future behavior of the process. The rapid on-line diagnosis of faults is even more important since it can anticipate and minimize the impact of otherwise costly effects.

The first part of this thesis addressed the problem of fault diagnosis in multivariate, dynamic, continuous chemical processes. Two types of faults were considered: deterministic (whose root cause is a randomly occurring deterministic event) and stochastic (caused by an underlying stochastic process). A realistic simulation of a chemical plant was used as a test bed for the proposed methods. Due to the lack of accurate dynamic models for this type of process, a Pattern Recognition approach was followed. Within this framework, several methods were designed for the on-line and off-line diagnosis of both types of faults. All methods consisted of: I) a feature extraction step, where magnitude invariant features are extracted from both the reference patterns and the pattern of the new unknown fault, and II) a similarity assessment step where the distance between the new pattern and each of the reference patterns is estimated using Dynamic Time Warping.

Due to the use of magnitude invariant features and the ability of Dynamic Time Warping to synchronize similar patterns with distorted temporal correlations, the results were satisfactory in diagnosing deterministic faults. In the case of stochastic faults, the results were inconclusive. The correlation pattern between the variables was used as the feature for the diagnosis of stochastic faults. However, the slow dynamics and the effect of the recycle in the simulated chemical plant meant that unrealistically long records of data are required for an accurate estimate of this feature.

The second part of the thesis investigated the problem of fault detection in batch processes, and in particular the problem of batch trajectories of unequal duration and poor synchronization. A new method, based on Dynamic Time Warping, was proposed for the synchronization of batch trajectories of this type; the method is multivariate and requires minimal process knowledge. It was also shown how to use Dynamic Time Warping to synchronize a new batch trajectory with the reference trajectories so that batch monitoring methods based on Multivariate Statistical Methods could be used. Finally, a new on-line monitoring method was presented, based on the concept of instantaneous quadratic distance, which does not require prediction of the future behavior of the batch trajectory.

Acknowledgments

I would like to express my gratitude to my advisors, Dr. John MacGregor and Dr. Paul Taylor, for their guidance, stimulation and encouragement during these four years. In addition to being good advisors, they were also mentors and friends. I am grateful to them.

I would like also to thank Dr. T. Marlin, Dr. A. Hrymak and Dr. C. Crowe for being a valuable source of knowledge during all my years at McMaster.

The contribution from Dr. McAvoy and N. Ye of their control scheme in the Tennessee-Eastman simulation is mostly appreciated.

The financial assistance from the Department of Chemical Engineering, McMaster University, and from the McMaster Advanced Control Consortium is also appreciated. Special thanks go to Barb Owen for making all the paper work much simpler than what it could be.

This journey of mine in Canada would not have been possible without the friendship of many good people. Panos Seferlis has always been a great friend from day one. Dora Kourti was always there for advice and assistance. Bhupinder Dayal and Phil Nelson never declined an opportunity for a challenging debate. And with Alexi Gegio and Giorgo Alexantraki, we had too many beers together not to be mentioned.

Being away from home for so long was not easy. I am grateful to the families of Athena Gegiou, Pavlou and Vinas Kanaroglou, Giwrgou and Marias Papageorgiou, Elia and Eleftherias Thersidi and Dimitri and Marias Triantafullou for making this absence less painful and for offering me a much needed sense of home.

Finally, my family supported me in any possible way from thousands of miles away. They were a bright lighthouse, a point of reference invariant to the transformations of life. This thesis is dedicated to them with all my love and respect.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

This chapter introduces the concepts of Fault Detection and Diagnosis and their necessity in today's petrochemical plants. The two main approaches are presented: the mechanistic model-based and the empirical Pattern Recognition approach and their differences are discussed. Justifications are given for the Pattern Recognition approach studied in this thesis. Finally, the objectives, assumptions and research approach of this thesis are defined, for both continuous and batch processes.

1.1 Why Fault Detection and Fault Diagnosis

The need for reducing production costs has pushed today's petrochemical plants to operate close to, or even at their maximum capacities, while at the same time producing high quality and consistent products. Moreover, the incentive for optimal utilization of energy and raw materials, the wide operating windows required to achieve economically optimal operation, and the increasing environmental concerns, have resulted in highly integrated petrochemical plants and sophisticated control systems.

Under these conditions of complexity, monitoring of the plant operation and detection and diagnosis of faults are formidable tasks. In this thesis the term 'fault' is used to indicate anything that causes the plant operation to deviate from a desired operating point. Thus, the degradation of a sensor or an actuator, the failure of a pump, the poisoning of a catalyst, the sudden change in the composition of a feed stream, are all considered faults.

Obviously, many of the above faults are directly observable, in the sense that one can check the faulty sensor or the failed pump, or measure the composition of the feed stream. However, plants are usually understaffed and engineers cannot be committed to checking every sensor and actuator when a deviation from the desired operating point is detected. There exist also faults that are not directly observable (e.g., catalyst deactivation) and have to be inferred from other variables. Thus, automated Fault Detection and Fault Diagnosis schemes are of significant practical value in detecting a fault and pointing to probable sources.

Of the two tasks, Fault Detection is the first and the easier of the two. At any time, the plant operation is compared against a model which describes the normal operation. The model can be either mechanistic or statistical or a set of rules drawn from experience or a combination of the above. Significant deviations of the process measurements from the model predictions indicate that the process does not operate the way it should.

Knowing that plant operation is not normal is useful in its own right. However, the real benefits come when the source of the fault is identified and corrective actions are taken to eliminate it. This is true for any fault, but is even more important for faults that continuing to ignore them may lead to major upsets, like plant shut down, cyclic behavior that may take long to die out, or even dangerous equipment failures. Hence, Fault Diagnosis is the second step and is more complicated than Fault Detection: for detection requires a model of the normal plant operation, while diagnosis requires a model for each expected fault, as it will be discussed in the next section.

Fault Detection and Diagnosis are tasks whose major benefits are obtained when implemented on-line. However, their off-line implementation in analyzing historical data is also useful. Historical data provide information to create models for the normal and the faulty operations and they can also be used to test various proposed schemes before they are implemented on-line.

1.2 The Mechanistic Approach for Fault Detection and Diagnosis

The importance of Fault Detection and Diagnosis has lead to a large number of various approaches. Although not the only way, one could categorize them into two major classes:

- methods that utilize a mechanistic plant model; and
- methods that do not utilize a mechanistic plant model but are based on Pattern Recognition principles.

Mechanistic approaches are powerful; they utilize a process model constructed from first principles (i.e., conservation laws of mass, energy, etc.). The model consist of a deterministic and a stochastic component: the deterministic component is usually a set of nonlinear differential equations, while the stochastic component appears in the form of stochastic states, and process and measurement noise. The Kalman Filter (for linear systems) and the Extended Kalman Filter (for nonlinear systems) are the main expressions of the mechanistic approaches, where the combined model is used in conjunction with process measurements to on-line reconstruct unmeasured states and parameters. The Luenberger and the Extended Luenberger Observer are the deterministic analogs of the Kalman and the Extended Kalman Filter in cases where the stochastic component can be neglected.

Under this general class, several Fault Detection and Diagnosis schemes of various degrees of complexity have been proposed. Mehra and Peschon (1971) proposed simple statistical tests on the innovations (i.e., the difference between the actual system outputs and the expected outputs based on the model) as a Fault Detection scheme. Willsky and Jones (1976) and Willsky (1976) extended the concept of interrogating the innovations to a Fault Diagnosis scheme, by incorporating models for each suspected failure and applying a sequential likelihood test on the innovations.

The so called parity space approach is another major category in the class of mechanistic Fault Detection and Diagnosis schemes. Under this approach, the consistency (parity) of the mathematical equations is checked by using the measured process outputs. The design parameter of the method is the generation of linear equations, whose residuals should be zero, if the process measurements are taken from normal plant operation. The onset of significant residuals indicates the existence of a fault, while the direction of residuals indicates the origin of the fault (Chow and Willsky, 1984, Gertler 1988, Frank, 1990).

The Parameter Estimation (Isermann, 1984), and the Fault Detection Filter (Frank, 1990) approach are two other major mechanistic approaches. In the first approach, the physical process parameters are related to model parameters. The latter are estimated online and the key idea is that changes in the physical parameters will be reflected as changes in the estimates of the model parameters (Isermann, 1984, 1993).

In the Fault Detection Filter approach (Frank, 1990), a bank of deterministic observers is created, with each observer designed so that the residuals possess certain directional properties, indicative of a particular fault. An extension of this approach is the work of King and Gilles (1990). They consider nonlinear stochastic systems and thus they propose a bank of Extended Kalman Filters (EKF), with each filter corresponding to a specified fault. They also incorporate a Markov model technique, so that the *a priori* probabilities of the various faults are recursively updated as functions of the state variables.

Watanabe and Himmelblau (1984) used a two-step approach; in the first step, a deterministic observer is used to reconstruct the unmeasured states; in the second step, an EKF is used to reconstruct the values of stochastic model parameters, whose varying values represent faults. However, their method is specifically designed for systems that are linear in the state and input variables and nonlinear in the varying model parameters. Robertson and Lee (1993) proposed a constrained receding horizon state estimator, where the nonlinear state equations are transformed to algebraic equations via orthogonal collocation; process parameters are modeled as stochastic states and only faults that appear as changes in these parameters can be diagnosed.

The work of Fathi et al. (1993) is a combination of a mechanistic and a Pattern Recognition approach. Specifically, they use an Expert System that captures the domain knowledge and the problem solving strategy. A hierarchical structure defines a set of subsystems and a set of rules are used to postulate possible faults. The Expert System is complimented with a set of EKFs, each designed for a specified fault. Once a fault is detected and postulated, the corresponding filter is activated and its validity is checked via a Sequential Probability Ratio Test.

The above references describe just few variants of the mechanistic approach in Fault Detection and Diagnosis. However, it is not difficult to see why in chemical engineering applications, State Estimation and Fault Detection Filters via EKFs are the most popular ones. Nonlinear systems, unknown parameters, model mismatch, process and measurement noise, can only be handled in a stochastic framework, which renders impractical the deterministic observer approach. But even when EKFs are used, their success depends heavily on the quality of the mathematical model used to describe the system (King and Gilles, 1991). And for that reason, these approaches have not become widespread in chemical processes (Isermann, 1984, Frank 1990).

1.3 The Pattern Recognition Approach for Fault Detection and Diagnosis

If an accurate process model is imperative for a mechanistic approach in Fault Detection and Diagnosis, it is not a requirement when one decides to use Pattern Recognition to perform these tasks. For Pattern Recognition relies mainly on the measured information from the system; data and not models, convey the most important information.

According to the Pattern Recognition approach, historical data (i.e., patterns) from the normal operation and from past faults are collected. One can view the normal operation and each fault as separate classes in which the patterns belong to. These training data are then processed so that concise information is derived. This procedure is called feature extraction and its objective is to extract features which distinctly identify each class and distinguish it from the others. The next step is the design of a decision scheme that will use the extracted features and will try to classify the patterns in their classes in an optimal way (Tou and Gonzales, 1974). When a new unknown pattern appears, the same features are extracted and the decision scheme determines the pattern class which seems most probable to have generated the new pattern.

Supervised Pattern Recognition refers to the situation where the classes in which the training patterns belong to are known beforehand. If this information is not available, the system must find out the pattern classes present in the training data (Tou and Gonzales, 1974). This is the Unsupervised Pattern Recognition approach and is clearly more difficult than the Supervised approach.

Thus, information content of the historical data, types of features to be extracted, and design of the decision scheme, are the three main parameters that affect the performance of a Pattern Recognition approach in Fault Detection and Diagnosis, with the first parameter being the most important. If a new fault is not present in the training data set, then it is not possible to diagnose it correctly. The best that a Pattern Recognition scheme can do in such a situation is to indicate that the new fault does not resemble any of the faults existing in the database. These issues will be discussed in more detail in the next chapter.

1.4 Thesis Objectives and Outline

This thesis will study a Supervised Pattern Recognition approach to Fault Detection and Diagnosis in chemical processes. As mentioned in Section 1.2, chemical processes are integrated multivariable dynamic processes, characterized by unknown parameters, nonlinear behavior and noisy inputs. Creating reliable dynamic models and manipulating them on-line is not an easy task. For these reasons, a mechanistic approach, although powerful, will not be followed in this thesis.

On the other hand, the advent of computers has resulted in large amounts of data collected routinely from processes. Recently, efficient statistical techniques, able to utilize this wealth of information, have been implemented in monitoring of chemical plants (Kourti and MacGregor, 1995). However, most of these techniques assume conditions of steady state, and their extension is problematic when dynamic conditions apply.

This thesis is an attempt to propose new techniques in addressing the dynamic nature of Fault Diagnosis in two major classes of processes: continuous and batch processes. Tools from the area of Speech Recognition are brought in to help dealing with dynamic patterns of varying duration. Scaling procedures are proposed to remove the problem of varying magnitude of patterns. Multivariate Statistical Methods are also used to summarize process information and extract features.

In the next sections of this chapter, the objectives of the thesis are formally defined and the research approach is presented. In Chapter 2, a review of the Pattern Recognition approaches for Fault Detection and Diagnosis in chemical processes is given. This will help us understand the nature of the problem and the limitations that many of these methods face when confronted with dynamic, multivariate, magnitude and duration dependent patterns. Chapter 3 presents the theory of Dynamic Time Warping, a technique used in the recognition of isolated words. It is presented here since Dynamic Time Warping, in various versions, will be included in all methods presented in the following chapters.

Chapter 4 presents an off-line method to classify patterns of deterministic faults in continuous processes. The diagnosis of stochastic fault is studied in Chapter 5; a method suitable for both off-line and on-line implementation, is presented. The methods of both chapters address a number of requirements that any Fault Diagnosis scheme must satisfy if it is to be implemented in an industrial environment. Chapter 6 combines the methods of the previous two chapters with Principal Component Analysis, to reduce the dimension of the pattern space. The on-line implementation of the method shown in Chapter 4 is presented in Chapter 7. In Chapter 8 the problem of monitoring batch processes with unequal run lengths with be addressed, both for off-line and on-line implementation. Finally, Chapter 9 summarizes this thesis and its contributions, and proposes directions for future work.

1.5 Fault Diagnosis in Continuous Processes

Continuous and batch processes are both very important in the chemical industries. Both types of processes are dynamic, multivariable processes, but each type is characterized by special features, from a monitoring point of view, that have to be taken into consideration. This section and the next discuss these features, the requirements, assumptions and research approach for the two types of processes.

1.5.1 Requirements for a Fault Diagnosis Scheme

Let us assume that a Fault Detection scheme is in place and is signaling that the process operation is not normal. Process data are now being collected and they constitute the pattern of a fault. Let us also assume that there exist a database of patterns which represent previous known faults. The objective is to somehow assess the similarity of the new, unknown pattern with all the patterns in the database. The similarity will be computed either using the raw data or, most probably, using extracted features from both the database patterns and the new pattern. On the basis of maximum similarity, one would then postulate that the new pattern is an expression of the fault, whose pattern is most similar to the new pattern.

In order to do this assessment, the pattern classification scheme (feature extraction and similarity assessment) has to satisfy a number of requirements. These are the following:

- The classification scheme has to be independent of the magnitude of the patterns; for example, a step-like fault (bias in a sensor, step change in the feed composition) can occur with different sizes. The magnitude of the corresponding patterns will be different, but the fault is the same. However, the information content of the fault, as measured by the signal to noise ratio of the variables, does affect the classification, A fault of large magnitude has a large signal to noise ratio and therefore it will be detected with more certrainty than a fault of smaller magnitude.

- The classification scheme has to be independent of the time duration of the patterns and of the plant operating point. Continuous processes operate at various points to meet demand and quality objectives. Different production levels result in slower or faster process dynamics. A fault may occur in any of these operating points, but it must be correctly classified as the same fault.
- The classification scheme should be independent of the direction of a fault. For example, a feed composition may increase or decrease. If the new fault is a negative step in the feed composition, the classification scheme should be able to classify it as such, even if only the pattern of a positive step exists in the database.
- The onset of a fault may not be known exactly, but within a time window. The classification scheme should be robust to this uncertainty, particularly in an on-line implementation.
- The classification scheme should be able to handle a large number of noisy variables. Patterns will be multivariate and the information for a specific fault may not be in any single variable but in all of them.

These requirements have to be addressed by any Fault Diagnosis scheme if it is to be implemented in an industrial environment. In the methods proposed in this thesis, some requirements are addressed at the feature extraction step and the remaining ones at the similarity assessment step.

1.5.2 Assumptions and Research Approach

For this thesis, it will be assumed that there is a Fault Detection scheme in place which detects the onset of an abnormal plant operation. There is a considerable amount of work in monitoring of continuous chemical processes at steady-state conditions and it has been shown that Multivariate Statistical Methods are very effective in detecting deviations from the normal operation (Kresta et al., 1991). For that reason, only the problem of Fault Diagnosis will be investigated in this thesis. Thus, once a fault has been detected, it will be assumed that a multivariate time series (i.e., the test pattern) is obtained that constitutes the expression of the fault. It will also be assumed that there exists a database of reference patterns, each corresponding to a known fault.

In this thesis, faults will be classified into two major categories: deterministic and stochastic. The cause of a deterministic fault will be assumed to be a randomly occurring deterministic event, such as a step change of a feed composition. For the class of stochastic faults, it will be assumed that they are caused by an underlying stochastic process (e.g., continuous stochastic variations in a feed composition); as such, different realization of the same fault result in different patterns in the process variables. In both cases it will be assumed that the faults are not directly measured but they are observable through the deviations that they cause in the measured process variables. Finally, for deterministic faults, it will be assumed that the control system has enough degrees of freedom to drive the controlled variables back to their setpoints. For the class of stochastic faults it will be assumed that they have a persistent effect on the process; the control system will try to bring back the variables to their set points, but this is impossible unless the underlying stochastic process terminates.

Both the on-line and the off-line implementation of Fault Diagnosis will be examined in this thesis. In the case of deterministic faults, there is a major difference between the off-line and the on-line implementation. In the off-line case, there is available information about both the transient behavior and the final steady-state conditions. In the on-line case, only part of the transient behavior is available, until steady-state conditions are achieved. Therefore, any features that depend on the steady-state conditions after the fault, cannot be used for the on-line diagnosis of deterministic faults. In the case of stochastic faults, there is no difference between the on-line and off-line diagnosis since no steady-state conditions can be achieved after a stochastic fault occurs (unless the stochastic process terminates).

In all cases (deterministic/stochastic, on-line/off-line), appropriate features are extracted from the reference patterns and the test pattern; finally the similarity between the features of the test pattern with the features of each reference pattern is assessed. It will be assumed that the start and the end of the fault is given, but there will be some uncertainty (i.e., a time window) within which the initial and final points may lay. For the on-line classification problem, it will be assumed that only the start of a fault is given, again with some uncertainty.

To design and test the diagnostic method, only simulated data will be used. The Tennessee Eastman simulation will be used (Downs and Vogel, 1993) with the control scheme of McAvoy and Ye (McAvoy and Ye, 1994). Figure I.1 in Appendix shows the plant schematic and the control scheme. The problem was proposed as a test bed for studies in process control and optimization and it is based on an actual industrial process. Since its introduction (Downs and Vogel, 1993), it has attracted a number of studies in control system design (McAvoy and Ye, 1994, Ricker and Lee, 1995a, Ricker, 1996), modeling and state estimation (Ricker and Lee, 1995b) , optimization (Ricker, 1995) and Fault Detection and Diagnosis (Ku et al., 1995).

The process has five major components: a gas-phase reactor, a product condenser, a vapor-liquid separator, a recycle compressor and a product stripper (see Figure I.1 in Appendix). Since most of the streams are in gas phase, small delays exist between the different units. However, due to the recycle, the plant is characterized by long settling times; it was suggested to simulate 24-48 hours of plant operation to fully see the effect of various disturbances on the product quality as expressed by its composition.

To test the various control schemes, 20 different faults (or, equivalently, disturbances) can be simulated. Some of them are deterministic (i.e., step changes in plant inputs, faulty valves) and some are stochastic (i.e., random variations in plant inputs). Some of them are minor disturbances and the cascade structure of McAvoy and

Ye (1994) handles them without any effect on the product quality. However, some faults have a very strong effect on the plant (i.e., they cause large variations in both manipulated and controlled variables) and these are the faults that will be examined in this thesis. More details on the various case studies will be given on Chapters 4 and 5.

Another feature of the simulation is that it is possible to move the plant to various operating points. Moreover, by appropriately modifying the computer code, it is possible to introduce the same faults with varying magnitudes. Both features will be used to test the proposed methods in correctly classifying the same fault, occurring at different production levels with different magnitudes.

1.6 Fault Detection in Batch Processes

Batch processes constitute another significant class of chemical processes, particularly in the production of high added value products, such as specialty polymers, pharmaceuticals and biochemical materials. Monitoring the operation of these processes is very crucial in manufacturing consistent, good quality product. Moreover, products from batch processes are often processed in a series of steps; early detection of a fault at any of these steps will result in saving energy and plant capacity that otherwise would be wasted. If implemented on-line, there is also a chance of correcting the fault with an appropriate control strategy.

1.6.1 Problem Formulation

Monitoring of batch processes faces a number of challenges like i) the lack of fast (if any) accurate measurements of the product quality variables ii) the absence of steady state, which renders most of the standard Statistical Process Control approaches inappropriate and iii) the difficulty of developing accurate mechanistic models.

Recently, Nomikos and MacGregor (1994) have proposed a method for monitoring batch processes based on Multivariate Statistical Methods. Their method essentially builds a statistical model for the deviations from the average trajectory of readily measured process variables, based on data from good quality batches. Then, it compares the trajectory of a new batch with the average trajectory; any deviation that cannon be statistically attributed to the common process variation indicates that the new batch is different from the good quality batches. Their method can be implemented both off-line and on-line; however, the on-line implementation requires a prediction of the future behavior of the batch from the current time up to the expected end of the batch.

For the method of Nomikos and MacGregor (1994) to be used, all batches must have the same time duration and be synchronized. However, when the various steps along the batch are not automated but are left to the discretion of an operator, batches will in general have varying duration and will not be exactly synchronized. In such a case they proposed that the batches be synchronized not with respect to time, but with respect to a process variable that is strictly monotonic, is not noisy and has the same starting and ending values for all batches.

This solution assumes that such a variable exists and is easy to determine. However, there may be several variables in a batch that are not noisy. Furthermore, there may not be a single variable that is strictly monotonic throughout the whole batch trajectory and one will have to switch between the selected variables at the appropriate times. This manual synchronization is time consuming and requires a lot of ad hoc decisions.

Thus, there two objectives in this part of the thesis. The first objective is to devise an automated method that will synchronize the varying duration, good quality batches without the assumptions of the Nomikos and MacGregor synchronization method. The second objective is to propose a Fault Detection scheme for on-line implementation which will not require to forecast the future behavior of the batch.

The methods will be based only on the available process measurements. Hence, Fault Detection will be treated as a Pattern Recognition problem, where the new batch is compared to the average good quality batch and its similarity to the latter is statistically assessed. On the other hand, Fault Diagnosis is almost impossible in batch processes without the use of a mechanistic model. The reason is that the same abnormality will be expressed differently in the process measurements if it occurs at different stages along the batch. To correctly classify a fault with a Pattern Recognition approach, it would require a set of reference patterns corresponding to the same fault occurring at different times; such a rich database of patterns may not be available. Thus, the studies on batch processes will focus only on the detection of abnormal operation.

1.6.2 Research Approach

Data from an industrial emulsion polymerization batch process will be used to design the Fault Detection scheme. Figure I.4 in Appendix contains a plot of all 10 variables (scaled) from 31 good quality batches. One can see from the plots that the batches are not synchronized and do not have the same duration.

The first step will be to design a method that will synchronize them in such a way that their timing differences are reconciled. Once this is done, one can build the monitoring scheme of Nomikos and MacGregor (1994) using the synchronized trajectories. Next, the problem of synchronizing a completed new batch with the reference set trajectories will be addressed. One can then assess off-line the quality of the new batch using their monitoring scheme. Finally, a new on-line Fault Detection scheme will be proposed which uses only information up to the current time and does not require any forecasting for the future behavior of the batch.

CHAPTER 2

REVIEW OF PATTERN RECOGNITION APPROACHES

FOR FAULT DETECTION AND DIAGNOSIS

This chapter presents an overview of the three major Pattern Recognition approaches: Artificial Neural Networks, Expert Systems, and Multivariate Statistical Methods, used for Fault Detection and Diagnosis in petrochemical processes. Their relative merits and demerits are discussed in the context of the requirements for a robust scheme for Fault Diagnosis presented in the previous chapter.

2.1 Fault Diagnosis Using Artificial Neural Networks

Artificial Neural Networks (ANNs) have gained high popularity in the recent years in modeling multidimensional nonlinear input-output relations and in pattern classification (Lippmann, 1987). Their black box configuration has made them an attractive tool for many applications where limited knowledge exists about the underlying physical mechanisms.

An ANN is serially composed of a input layer, a number of hidden layers and an output layer. Each layer contains a number of nodes and the nodes of each layer are corrected with the nodes of the preceding and the subsequent layer. When used for modeling input-output relationships, the first layer of the ANN receives the values of the input variables; the predictions of the outputs are obtained from the last layer. Thus, for steady-state relationships, the number of nodes in the first and last layer is equal to the dimension of the input and output space, respectively. If the relationship is dynamic, then more nodes have to be added in the first layer to receive lagged versions of the input variables.

When used for supervised pattern classification, the first layer of the ANN receives the raw data and/or features extracted from the input patterns and contains the appropriate number of nodes; the last layer contains as many nodes as the pattern classes present in the data. A value from an output node close to one indicates that the given pattern originated from the pattern class corresponding to that particular node (Leonard and Kramer, 1991).

After the first layer, the input to each node is a weighted linear combination of the outputs of the previous layer's nodes. The output of a node is a non-linear function (the activation function) of its input. Typical activation functions are the sigmoid function and an unnormalized gaussian function, called the radial basis function.

Training the ANN is the process of adjusting the weights that connect each node with the nodes of the previous layer. The weights are adjusted so that the net outputs are as close as possible to the actual values. Depending on the activation function, this can be a linear or a nonlinear least squares problem. The number of the hidden layers, as well as the number of nodes in each hidden layer, is found experimentally depending on the accuracy of the approximation, the computation time, and the trade-off between approximation and generalization (since most probably the network will be used to predict the effect or classify input patterns, different from the ones used to train it).

In the context of Fault Diagnosis in petrochemical processes, ANNs have been used as supervised pattern classifiers. A common characteristic of most of the studies is that ANNs were trained on simulated, steady-state process data with the aim of detecting a specified number of suspected faults.

Venkatasubramanian and Chan (1989) used such an ANN and compared it against the performance of an Expert System to diagnose faults in a fluid catalytic cracking unit. The ANN was able to generalize its knowledge to diagnose combinations of static faults that were not used to train it, as well as to give an indication of the possible faults when it was fed with incomplete data. In a subsequent paper (Venkatasubramanian et al. 1990), a more complicated example was studied and it showed the ability of the ANN to correctly diagnose single or multiple static faults when fed with noisy data.

In another work, Watanabe et al. (1989) designed a Fault Diagnosis scheme for static faults that consisted of 2 ANNs in series : the first network identifies the fault, while the second identifies its severity level. Fan et al. (1993) approached the same problem (diagnose a static fault and its severity in a simulated process): they expanded the input space of an ANN by adding a number of functional units to the input layer. Although their network can correctly classify the faults and their severity, their approach is less intuitive than the 2-stage ANN.

In a more realistic application, Hoskins et al. (1991) designed an ANN to diagnose faults in a simulation of a large chemical plant with 418 inputs and 20 outputs, and they were able to correctly identify single and multiple faults using only static patterns. Along these lines is the work of Sorsa et al. (1991) but their study goes further to investigate the effect of feeding dynamic data to an ANN which was trained only on steady-state data. Their examples show that some faults can be diagnosed from the initial part of their response, while other faults can be detected correctly only after steady state is achieved. Naidu et al. (1990) used ANNs to detect sensor biases of different magnitude and onset time in a single input-single output control system. They used as an input to the network the cosine transform of the deviations of the model prediction from the process output.

In all the above applications, the sigmoid function was used as the activation function of the hidden and output layers of the network. However, despite of the preliminary encouraging results, it was soon realized that this activation function had serious undesirable properties. Kramer and Leonard (1990) illustrate these problems: i) the global character of the sigmoid function may result in incorrect classification, ii) the many local minima in the determination of the optimal weights, iii) the placement of the decision surfaces close to the edges of the classes, resulting in extrapolation errors.

In a subsequent work, Leonard and Kramer (1991) proposed that radial basis function be used instead of the sigmoid. With this activation function a novel fault will not be classified as one of the known ones. The decision surfaces are conservatively placed, which results in fewer extrapolation errors. A k-means clustering algorithm and a 2-nearest neighbor heuristic was used to find the center and the support of the function for each node; the weights are determined using linear least squares. These features were verified by the works of Sorsa and Koivo (1993) and Guglielmi et al. (1995); in both studies the radial basis function ANNs showed their advantages in diagnosing static faulty patterns over the sigmoid function networks. Similarly, Kavuri and Venkatasubramanian (1993) proposed an ellipsoidal function to be used as an activation function, together with a fuzzy clustering algorithm and heuristic rules to locate the center and axes lengths for each node.

The studies of Cooper et al. (1992) and Megan and Cooper (1995) considered dynamic data. Their objective was to adapt the proportional and integral mode of a controller (which were the outputs from the network) by examining univariate dynamic input and error patterns of fixed length (i.e., network inputs). Networks designed specifically for modeling dynamic input-output relationships are discussed in Hush and Horne (1993).

Finally, Bakshi and Stephanopoulos (1993) designed a network whose activation functions are drawn from a family of orthonormal wavelets. Although there are advantages in doing so (i.e., they are localized functions, the network can learn in increasing resolution and the weights are found by linear least squares), the implementation of the network becomes too difficult for multidimensional static patterns.

The applications of ANNs described above indicate that Fault Diagnosis via ANNs has dealt mainly with the diagnosis of static faults. On the other hand, industrial processes are dynamic systems and sometimes they need considerable amount of time to attain steady-state conditions after a fault. Incorporating dynamic information into an ANN leads to large networks whose training requires rich databases. However, only few realizations of faults usually exist and one may have to train the network using simulated dynamic data, if a reliable dynamic process model exists (Leonard and Kramer, 1991).

To avoid erroneous classification, localized functions would be needed and for the size of the required network, the determination of their center and extent would be very tedious, if not impossible. Moreover, the ANN would learn the temporal correlations and the correlations across variables of the faults used to train it; thus, it may not be robust in correctly classifying realizations of the same fault with slightly different correlations (e.g., at a different operating point).

2.2 Fault Detection and Diagnosis Using Experts Systems

Another tool that has been used for Fault Detection and Diagnosis in the petrochemical processes is the family of Expert Systems. It would be very advantageous to somehow code the knowledge-based reasoning of a process expert, combine it with a mathematical model of the system and afterwards use this device to supervise the plant operation. Knowledge-Based Systems are Expert Systems in which the diagnosis procedure uses heuristic rules drawn either from process knowledge or from simulations of a process model. On the other hand, the knowledge base of Model-Based Expert Systems is directly the process model; the pattern recognition is then applied on the residuals of the model equations. From this point of view, Model-Based Expert Systems resemble the parity space approach for Fault Diagnosis, discussed in Section 1.2. In this section, a summary of Expert Systems approaches for Fault Diagnosis from both categories is presented.

In one of the earlier works, Shum et al. (1988) designed a Knowledge-Based System for Fault Diagnosis that constructs a malfunction hierarchy: each node in the hierarchy contains qualitative knowledge of a process unit and is connected with neighboring nodes in a way similar to the way that the process units are connected. The last nodes in the hierarchy are root causes for various faults. Starting from a unit that gives a fault symptom, the reasoning process consists of traveling down the tree using the rules in each node and the symptoms, until a root cause node is reached. The Expert System is designed for a particular plant in mind, and if the plant configuration changes the system has to be redesigned.

Along similar lines is the work of Rich and Venkatasubramanian (1987). In their Knowledge-Based system, a library of process units in constructed and a set of rules, drawn from a process model, is given to each unit. When a fault symptom in any of the units appears, a possible fault is located based on the rules and the plant structure. Their method can be applied only to plants that use the units that exist in the library, otherwise a new set of rules must be written.

Kramer (1987) designed a Model-Based System where the knowledge base consists of model equations and the assumptions under which the equations are satisfied. The main idea is that if the plant measurements do not satisfy the model equations (i.e., the residuals are larger than some prespecified tolerances), then at least one of the assumptions is no longer valid, thus a fault has occurred. The belief of an assumption being violated was dependent on the magnitude of relevant residuals. With this approach, faults are not only Boolean events, but also time-evolving phenomena; e.g., the gradual deterioration of an equipment. However, it was not possible to distinguish between faults that give similar symptoms since all assumptions were considered equally important.

These issues are discussed in the work of Petti et al. (1990). They used model equations as their knowledge base and the model residuals to detect faults. Moreover, the sensitivity of each equation to the relevant assumptions was taken into account in the fault detection algorithm. In that way, they were able to distinguish between similar faults; for the case of multiple faults, their method will find all possible fault combinations that could yield a given pattern of symptoms. Chang et al. (1994) proposed a similar Model-Based Expert System, where more refined measures are constructed to distinguish between faults with similar symptoms.

All the above works consider only steady-state patterns. However, as emphasized in Chapter 1, dynamic patterns are strong indications of specific malfunctions and if examined, could provide faster and more reliable detection algorithms. Under this main idea, Konstantinov and Yoshida (1991 and 1992) tried to build a Knowledge-Based system that would consider the temporal shapes of variables. The rules do not only include values of the process variables, but also the shape of their evolution in time. A library is created with a set of primitive transient patterns whose shape is qualitatively described by the signs of the first and second derivative. During on-line implementation, the raw data are approximated with low degree polynomials, from which the signs of the first and second derivatives are estimated; a similarity index with all the library patterns is then estimated. With this approach, the Expert System was able to accurately detect in time the transition between the various stages of a biological process and appropriately modify the control strategy.

Vinson and Ungar (1995) tested two Knowledge-Based approaches for dynamic Fault Detection and Diagnosis based on qualitative simulation. The first approach uses qualitative models for each fault; plant data are examined and their behavior is qualitatively compared with each expected faulty behavior. The second approach uses qualitative and semi-quantitative reasoning (bounds on variables); the data are examined and if they do not agree with the expected normal behavior, a search is done among unit(s) in which the fault appears, to determine possible faults that result in the observed behavior. The first approach requires good qualitative knowledge of the faults, the second may be impractical in large plants due to a large number of possible faults.

Finally, the work of Cheung and Stephanopoulos (1990a and 1990b) and Bakshi and Stephanopoulos (1994a and 1994b) on Fault Detection and Diagnosis for batch processes should be mentioned. In a series of papers, the steps of an complete methodology are presented. First, a database of known quality batches is required, which shows the evolution of each measured variable. For all batches, each variable is decomposed into a series of descriptions in various time scales using the wavelet decomposition of signals. Moreover, all the descriptions are qualitatively represented by a series of segments called episodes, during which the signs of the first and the second derivative remain constant. Next, using the theory of inductive decision trees the method finds the features in the variables that will give the best classification by examining initially the qualitative and, if necessary, the quantitative representations. After this off-line analysis is done, a set of rules are drawn which constitute the knowledge base of an Expert System. This Knowledge-Based system is then used on-line to detect static or dynamic patterns that may result in a poor quality batch.

In general, reliable Knowledge-Based Expert Systems require good knowledge of the process behavior, both in normal operation and when different realizations of various faults occur. Also, the more rules are written, the more difficult it becomes to examine all the possible paths through a decision tree and to verify that all 'if-then' statements are consistent, particularly for complex multivariable plants. On the other hand, reliable Model-Based Expert Systems require accurate process models. These models may not be available, particularly when the objective is Fault Diagnosis using dynamic data.

2.3 Fault Detection and Diagnosis Using Multivariate Statistical Methods

Recently Multivariate Statistical Methods have been used for Fault Detection and Diagnosis in chemical processes (Kourti and MacGregor, 1995). Process computers now collect masses of data from a multitude of plant sensors every few minutes or seconds. Evaluating this mass of information using classical univariate methods (e.g., linear regression, univariate statistical control charts) is often inadequate because it is not only the evolution of each variable by itself that gives useful information about faults, but also the evolution of each variable relative to the other variables. This correlation structure among the variables is imposed by the physical mechanisms which govern the process operation. Multivariate Statistical Methods are designed to model this correlation structure; furthermore, they can project the information in low dimensional spaces, expressed in fictitious uncorrelated variables called principal components. The evolution of the process can then be observed in the space of the principal components.

These methods can also handle noisy measurements and missing data, two very real problems in plant operation. For all these reasons, Multivariate Statistical Methods have been successfully used in monitoring of multivariable chemical processes. In this section, some applications of the two most widely used methods: Principal Component Analysis (PCA) and Partial Least Squares (PLS) will be reviewed. The objective is to show that, although these method are very efficient in detecting plant abnormalities, they are not appropriate for diagnosis and classification of dynamic patterns.

Singular Value Decomposition in numerical analysis, and Karhunen-Loeve Expansion in Pattern Recognition (Tou and Gonzales, 1974, Wold et al., 1987) are two other names for PCA. PCA summarizes the large number of correlated measurements taken from a process at steady state, with a small number of fictitious, uncorrelated variables called principal components. The first principal component is the direction along which the measurements exhibit the greatest variability. Subsequent principal components account for the remaining variability, while also being orthogonal to the subspace defined by the previous principal components. The number of principal components can be determined by a variety of techniques like cross-validation (Wold et al., 1987), the broken stick rule (Jolliffe, 1986) and parallel analysis (Ku et al., 1995).

PLS is another widely used Multivariate Statistical Method (Geladi and Kowalski, 1986). PLS is applied when correlated quality variables are also measured together with the process variables. The objective of PLS is to build a model to predict the quality variables based on information from the process variables. PLS also finds fictitious variables (called latent variables) in the process variables; however these latent variables capture the directions in the process variables that are most predictive of the quality variables. Again, cross-validation can be used to determine the number of required latent variables (Geladi and Kowalski, 1986).

To use either PCA or PLS for process monitoring, data from normal operation are first collected. Then, PCA/PLS models are constructed from these data (the principal components and the descriptions of the data with respect to the principal components). Also, under some distributional assumptions, confidence intervals are created for these new descriptions and for various other statistics; e.g., the square of the error between the raw data and their predictions from the PCA/PLS model. In a Pattern Recognition framework, this is the feature extraction stage. When new operating data become available, the developed PCA/PLS models are applied, the statistics are computed and are compared to the corresponding confidence intervals. If the statistics are confined in their confidence intervals, there is no evidence that the process operation is not normal and vice versa.

Kresta et al. (1991), Slama (1992), Hodouin et al. (1993), and Dayal et al. (1994) presented applications of the above principles in monitoring various continuous processes: fluidized bed reactor, extractive distillation column, a fluidized catalytic cracking unit, grinding and flotation units in a mineral plant, a Kamyr digester. In all these studies, process faults expressed themselves with deviations of the statistics from their confidence intervals.

MacGregor et al. (1994) proposed the use of Multi-block PLS, a variant of the normal PLS that can handle multiple blocks in the process and quality variables matrices, in situations where the process can be naturally blocked into subsections. They proposed the use of monitoring charts for each of the subsections, as well as for the entire unit, so that faults could be located more easily. Moreover, they proposed diagnostic tools based on the underlying PLS model to pinpoint the variables that do not follow the expected correlation structure, and consequently, are at fault. With the exception of Slama (1992) and Dayal et al. (1994) where lagged version of variables were used to formulate the PCA and PLS models, all other applications mentioned above used only steady-state data to detect normal from abnormal operation.

The work of Nomikos and MacGregor (1994, 1995a and 1995b) and Kourti et al. (1995) was the first application of PCA/PLS where dynamic data were used to monitor the operation of batch processes. Their method (described in Subsection 1.6.1) assumes that all batches have the same duration; thus, time can be treated as an independent variable for the purposes of model building and monitoring. Once a faulty batch is detected, diagnostic tools are used to determine which variable at which time does not follow the expected correlation structure.

Recently, there have been attempts to apply PCA in continuous processes using dynamic data. Dunia et al. (1996) use PCA to diagnose faulty sensors. Each variable is appropriately reconstructed and by examining the associated residuals one can distinguish between a faulty sensor and an abnormal operating condition. To reduce the effect of measurement noise and process dynamics, a moving window is used to filter the residuals. However, no diagnostic tools are given for the case that a process abnormality occurs.

Raich and Cinar (1994) and Ku et al. (1995) used PCA with dynamic data to diagnose the various faults at the Tennessee-Eastman simulation. They both construct PCA models for the normal operation and for each fault. Ku et al. (1995) used lagged versions of the variables so that the PCA model accounts for dynamic relationships among the variables. Both approaches were capable of detecting and diagnosing the faults. However, faults of different magnitude/direction, occurring at different operating points were not considered.

To summarize, PCA and PLS have been proved very useful in modeling the correlation among many highly correlated variables and in detecting deviations from the normal plant operation. To detect a fault, it is sufficient to see the process moving outside of the normal operating region in the reduced space. However, to diagnose a dynamic fault, one has to compare the patterns resulting from the process excursion out of the normal operating region. This has to be a robust comparison, for almost no fault realization will be exactly similar to known past faults. Alternatively, one could use a different PCA model for each fault. However, to diagnose all the possible variations of a fault (e.g., occurring at a different operating point) a different model PCA has to be created, which means an unrealistically rich database of past faults is required. Thus, the conclusion from this discussion is that PCA/PLS are not appropriate for diagnosis of dynamic faults.

2.4 Summary and Conclusions

This chapter summarized three popular Pattern Recognition approaches for Fault Detection and Diagnosis: Artificial Neural Networks, Expert Systems and Multivariate Statistical Methods. All of them were found inefficient when faced with the requirements presented in Subsection 1.5.1 for a robust Fault Diagnosis scheme in multivariable dynamic processes. The main disadvantage of all three approaches is that they require a rich database, with every possible variation of a fault; such database is unlikely to exist. Also, training of a large Neural Network or constructing a sufficient set of rules for a complex process are both difficult tasks.

To classify a dynamic pattern in a magnitude independent way, magnitude invariant features have to be extracted. Moreover, to classify dynamic patterns that are not perfectly aligned and are characterized by similar, but possibly expanded or contracted temporal correlations, a flexible pattern matching method is required. The method should be able to appropriately translate, compress, and expand the patterns so that the magnitude invariant, similar features are matched. The next chapter present Dynamic Time Warping, a method used in the area of Speech Recognition, which is capable of performing this kind of pattern matching.

CHAPTER 3

PATTERN MATCHING VIA DYNAMIC TIME WARPING

Dynamic Time Warping is a robust method for pattern matching that has been used extensively in the area of Speech Recognition. The simplest version of Dynamic Time Warping, namely its use in the recognition of isolated words, is discussed in this chapter.

3.1 Introduction

Consider two multivariate time series, **R** and **T**, each containing a realization of a fault. Each multivariate time series will be considered to be a pattern. Pattern **R** will be assumed to be a pattern from a database of existing reference patterns; each reference pattern is the expression of a known fault. **T** is the new, or test, pattern. The objective is to somehow find which reference pattern is most similar to the test pattern in some distance sense and classify on the basis of minimum distance. This is a simple 1-nearest neighbor classifier (Tou and Gonzales, 1974).

The vectors in each pattern may contain either the raw data as recorded by the sensors or (most probably) some features extracted from the data. These features may vary from simple filtered estimates of the raw data to spectral estimates or to parameters of a specific time series model. The latter is the procedure applied in Speech Recognition; the speech signal is discretized and normalized by adjusting the maximum signal amplitude. Next, the signal is segmented to short overlapping segments, and from each one of them acoustic parameters are extracted. These may be either spectral estimates and/or coefficients of an autoregressive model of constant order. In any case, either with raw data or with some set of features, patterns **R** and **T** are viewed as two time series of respectively r and t N-dimensional vectors (frames, in the Speech Recognition nomenclature, O'Shaughnessy, 1986, Silverman and Morgan, 1990). Both **R** and **T** are stored in matrix form, where the columns represent features and/or variables and the rows represent successive vectors in time. Thus, **R** and **T** are matrices of dimension $\mathrm{r} \times \mathrm{N}$ and $\mathrm{t} \times \mathrm{N}$, respectively.

Now, the objective is to somehow estimate the similarity between patterns **R** and **T** in some distance sense. If the number of vectors in the two patterns were equal, i.e. $\mathrm{r} = \mathrm{t}$, then a logical procedure would be to estimate the quadratic distance between each vector in **R** and **T** (characterized by the same time index), and average these distances. Thus, if **R**(i,:) and **T**(i,:) are the i^{th} vectors in the patterns **R** and **T**, the local distance, *d*(i,i), between these two vectors is:

$$d(\mathrm{i},\mathrm{i}) \quad = \quad (\mathbf{R}(\mathrm{i},:) - \mathbf{T}(\mathrm{i},:))\mathbf{W}(\mathbf{R}(\mathrm{i},:) - \mathbf{T}(\mathrm{i},:))^{\mathrm{T}} \tag{3.1}$$

and thus the average distance, $D(\mathrm{t},\mathrm{r})$, between **R** and **T** would be given by[1]:

$$D(\mathrm{t},\mathrm{r}) \quad = \quad \frac{\sum_{\mathrm{i}=1}^{\mathrm{t}} d(\mathrm{i},\mathrm{i})}{\mathrm{t}} \tag{3.2}$$

When **W** is set equal to the identity matrix **I**, the Euclidean distance is obtained in Eq (3.1), while the Mahalanobis distance is obtained when **W** is equal to the inverse of the covariance matrix **S** of the features in the reference pattern vectors (O'Shaughnessy, 1986). The Mahalanobis distance has its origin in statistical decision theory, where each vector of the reference pattern **R** can be viewed as the mean of an N-dimensional multivariate normal distribution. Bayes' rule will select that reference pattern whose density was the most likely to have generated the test pattern (assuming all pattern classes

[1] $D(\mathrm{t},\mathrm{r})$ is a special case of a general form of a normalized total distance to be formally described in Section 3.3.

are equally probable). Thus, if the probability density function for a feature vector **T**(i,:), p(**T**(i,:)), is:

$$p(\mathbf{T}(i,:)) = (2\pi)^{-\frac{N}{2}} |\mathbf{S}|^{-\frac{1}{2}} \exp\left[-\left(\frac{1}{2}\right)(\mathbf{T}(i,:)-\mathbf{R}(i,:))\mathbf{S}^{-1}(\mathbf{T}(i,:)-\mathbf{R}(i,:))^{T}\right] \qquad (3.3)$$

then, for the particular ith vector, Bayes' rule will select among the reference pattern classes the one which maximizes p(**T**(i,:)), assuming equally likely reference classes. With the same argument, one can view the summation of distances in Eq (3.2) as the logarithm of multiplication between probabilities, assuming independence among successive vectors in both **R** and **T** patterns.

However, it is rarely the case that the two patterns have the same number of feature vectors. This will impose a decision: which vector of the test pattern must be compared against which vector of the reference pattern. There is also a more subtle situation: even if the two patterns contain the same number of vectors, these may not be necessarily aligned in time. Two examples will illustrate this point.

The first comes from Speech Recognition (O'Shaughnessy, 1986): consider two different utterances of the word 'sues' with the same number of frames. Most probably the difference between them will be in the duration of the sound /u/ rather than in the sounds /s/ and /z/ in the start and end of the word. A linear, frame-to-frame comparison according to Eq (3.2) will produce a large distance measure, which may result erroneously in classifying the two words as being different. The second example considers pattern classification in chemical processes. Consider two realizations of a step-type disturbance, where the step change occurs at different times from the origin. Also assume that one realization occurs at an operating point corresponding to a lower production rate. When the process operates at this operating point, it will attenuate the step change in a faster and more oscillatory way. Again, a linear, vector-to-vector summation of distances between the vectors of the two patterns will result in a large average distance and possibly in wrong classification.

Thus, what is required is a method to align similar characteristics in the two patterns. Dynamic Time Warping (DTW) is such a method. DTW uses the principle of Dynamic Programming to nonlinearly warp the two patterns in such a way that similar events are aligned and a minimum distance between them is obtained. DTW will shift some feature vectors in time, compress some and/or expand others so that a minimum distance is achieved (Nadler and Smith, 1993).

Consider again the two patterns and let j and i denote the time index of the **R** and **T** pattern, respectively. DTW will find a sequence of K points on a $t \times r$ grid:

$$\hat{\mathbf{F}} = \left\{ \mathbf{c}(1), \mathbf{c}(2), \ldots, \mathbf{c}(k), \ldots, \mathbf{c}(K) \right\} \tag{3.4}$$

where

$$\mathbf{c}(k) = (i(k), j(k)) \tag{3.5}$$

and

$$\max(r, t) \leq K \leq r + t \tag{3.6}$$

For a symmetric DTW algorithm (to be explained in the following paragraphs), this sequence can be viewed as defining a path on the $t \times r$ grid that optimally matches each vector in both patterns so that a normalized total distance between them is minimized (Sakoe and Chiba, 1978). Figure 3.1, (taken from O'Shaughnessy, 1986), illustrates the main idea behind DTW for two univariate patterns, **R** and **T**. By proceeding vector by vector through both patterns DTW finds the best vector in **R** against which to compare each vector in **T**, and vise versa (O'Shaughnessy, 1986).

As will be explained in the paragraphs to follow, there are many variants of the DTW algorithm. However, all of them can be classified either as symmetric or as asymmetric.

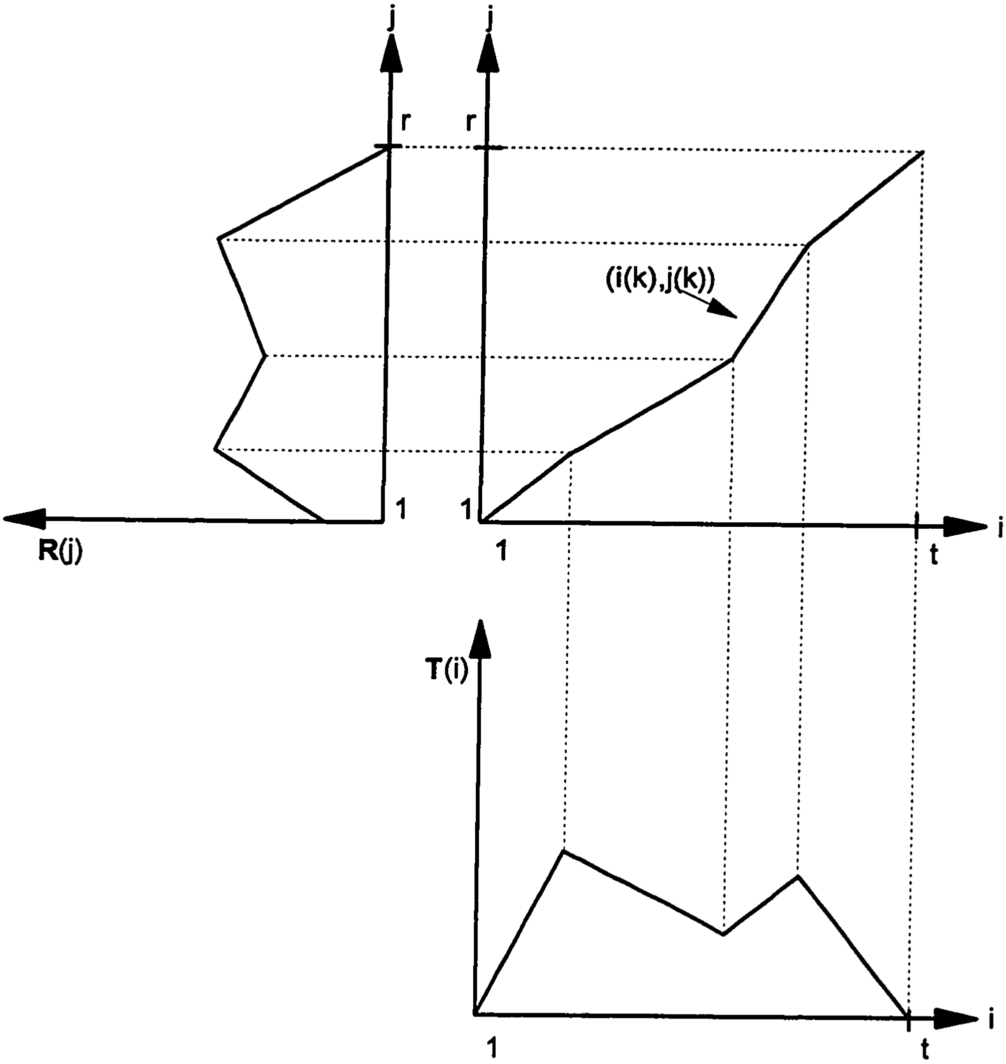


Figure 3.1: Example of nonlinear time alignment of a reference **R** and a test **T** pattern using Dynamic Time Warping.

In the symmetric versions, the time index i of the test and the time index j of the reference pattern are both mapped onto a common time index k, as Eqs (3.4) to (3.6) depict. The two patterns are considered to be equally important. The optimal path will go through each vector in both patterns. If the roles are reversed (i.e, **R** is considered as the test and **T** as the reference pattern) and their placement in the grid is interchanged (i.e., **R** is placed on the horizontal and **T** on the vertical axis), a symmetric DTW algorithm will give the same optimal path and the same total distance.

On the other hand, an asymmetric DTW algorithm will perform one of the following two tasks:

(i) it will map the time index of the reference pattern on the time index of the test pattern or vice versa, or,

(ii) it will map both time indices in a common time index, but it will tend to expand or compress one pattern relatively to the other.

For both tasks, the two patterns are not considered equivalent. Hence, if their role is interchanged, a different optimal path and a different optimal normalized total distance will be obtained. The most common asymmetric DTW algorithms map the time index of the pattern placed on the horizontal axis (i.e., **T**, in this discussion) onto the time index of the pattern placed on the vertical axis. In such a case, the common time index, k, is the time index, i, of the horizontally placed pattern, **T**, and the optimal path contains exactly t points, i.e.,

$$\hat{\mathbf{F}} = \left\{ \mathbf{c}(1), \mathbf{c}(2), \ldots, \mathbf{c}(i), \ldots, \mathbf{c}(t) \right\} \qquad (3.7)$$

where

$$\mathbf{c}(i) = (i, j(i)) \qquad (3.8)$$

The above description implies that the path will go through each vector in the **T** pattern, but it may skip vectors in the **R** pattern.

Nonetheless, both symmetric and asymmetric DTW algorithms can be cast in the same framework and a unique solution can be found using the method of Dynamic Programming. This can be done by appropriately specifying the values of specific parameters, as it will be explained in the following sections.

3.2 Local and Global Constraints

In order to find the best path through the grid of $t \times r$ points, several factors of the DTW algorithm have to be specified. These include: constraints on the endpoints of the path, local continuity constraints that define localized features of the path (i.e., slope), global constraints that define the allowable space for the path, and, finally, distance measures that will be used to define the optimization problem.

The most common, and simplest, endpoint constraints impose that the two extreme points of both patterns are matched. That implies that the first, $\mathbf{c}(1)$, and the last, $\mathbf{c}(K)$, path points are as follows:

$$\mathbf{c}(1) = (1,1) \tag{3.9a}$$

and

$$\mathbf{c}(K) = (t,r) \tag{3.9b}$$

These constraints are useful when the initial and final points in both patterns are located with certainty. However, when there is uncertainty about the location of the two extreme points, various endpoint constraints are imposed (Rabiner et al., 1978) which specify an allowable region where the first and last path point may be placed (see Figure 3.2). Their implementation depends on whether a symmetric or an asymmetric DTW algorithm is used and it will be explained in detail in Section 3.4.

The local continuity constraints reflect physical considerations (e.g., events should be compared in their natural order in time) and they also guarantee that excessive compression or expansion of the two time scales is avoided (Myers et al., 1980).

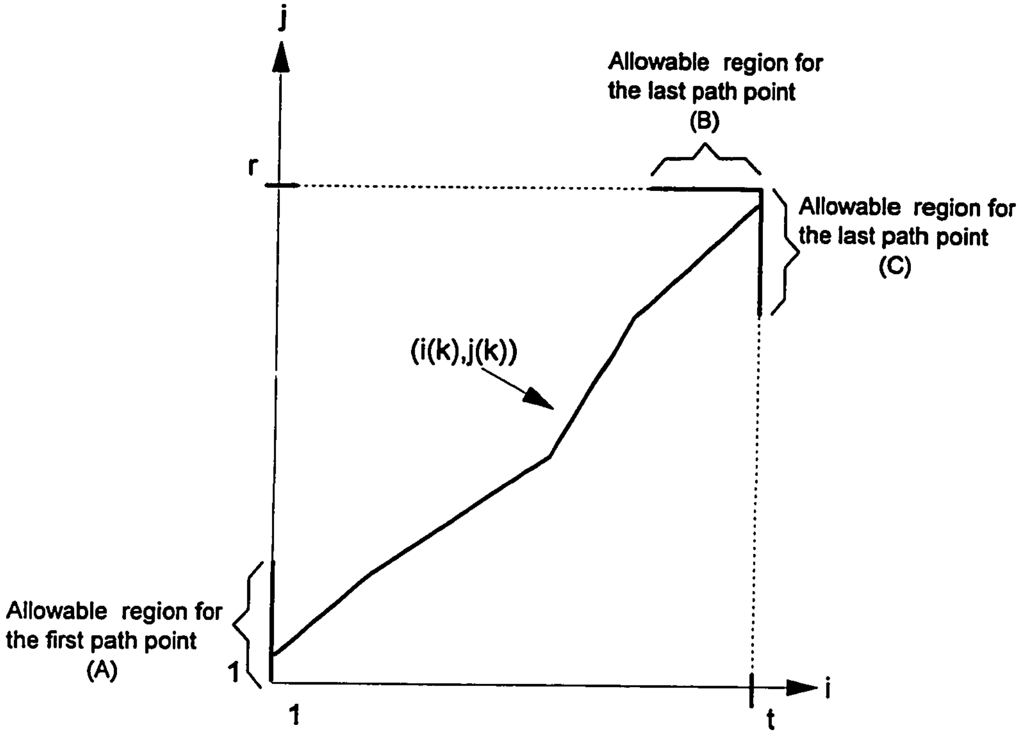


Figure 3.2: Allowable regions for the end points of the optimal path.

The first requirement is satisfied by forcing the path to be monotonous of non-negative slope. This can be expressed as:

$$i(k+1) \;\geq\; i(k) \tag{3.10a}$$

$$j(k+1) \;\geq\; j(k) \tag{3.10b}$$

The second requirement, (i.e., to avoid excessive compression or expansion of the two time scales), is achieved by not allowing the local slope of the path to exceed a specified range. This is accomplished by specifying a set of allowable predecessors for each (i, j) point in the grid: if (i, j) is the kth path point, then the previous (k-1)th path point can only be chosen from a set of specified grid points. Figure 3.3 illustrates common local continuity constraints and the corresponding slope range that they define.

In Figure 3.3(a) the Itakura local constraint is shown (Itakura, 1975). For each (i, j) point in the grid, only three predecessors are allowed: $(i-1, j)$, $(i-1, j-1)$ and $(i-1, j-2)$. Or, in other words, the only way to reach the (i, j) point is either through the $(i-1, j)$ or the $(i-1, j-1)$ or the $(i-1, j-2)$ point. The last local transition (i.e., going to the (i, j) point through the $(i-1, j-2)$ point) is characterized by a slope of 2: one horizontal and two vertical steps. Thus, a slope of 2 is the maximum slope allowed. On the other hand, two consecutive horizontal transitions are not allowed, as Figure 3.3(a) shows. That is, the local transition from point (i, j) to point $(i-1, j)$ will not be considered at all, if the optimal way to go to point $(i-1, j)$ is through the $(i-2, j)$ point. This means that whenever a horizontal local optimal transition exists (i.e., with 0 slope), it has to be followed by a transition that has slope of either one or two. This results in a minimum allowable local slope of 1/2 for the path. Hence, the Itakura local continuity constraint results in a slope range of [1/2, 2]. Moreover, it is an asymmetric constraint since horizontal local transitions are treated differently from vertical transitions; in fact, vertical transitions are not even considered.

Figures 3.3(b), 3.3(c) and 3.3(d) illustrate other types of local constraints, the so called Sakoe-Chiba constraints (Sakoe and Chiba, 1978). All of them restrain the slope of the optimal path by defining a set of allowable predecessors. The local constraint of Figure 3.3(b) is an exception to the above statement for it does not impose any restriction on the slope of the path; the path can follow horizontal or vertical local transitions with no restriction on their length. On the other hand, the local constraint shown in Figures 3.3(c) and 3.3(d) restrict the slope of the path to $[1/2, 2]$ and $[1/3, 3]$, respectively.

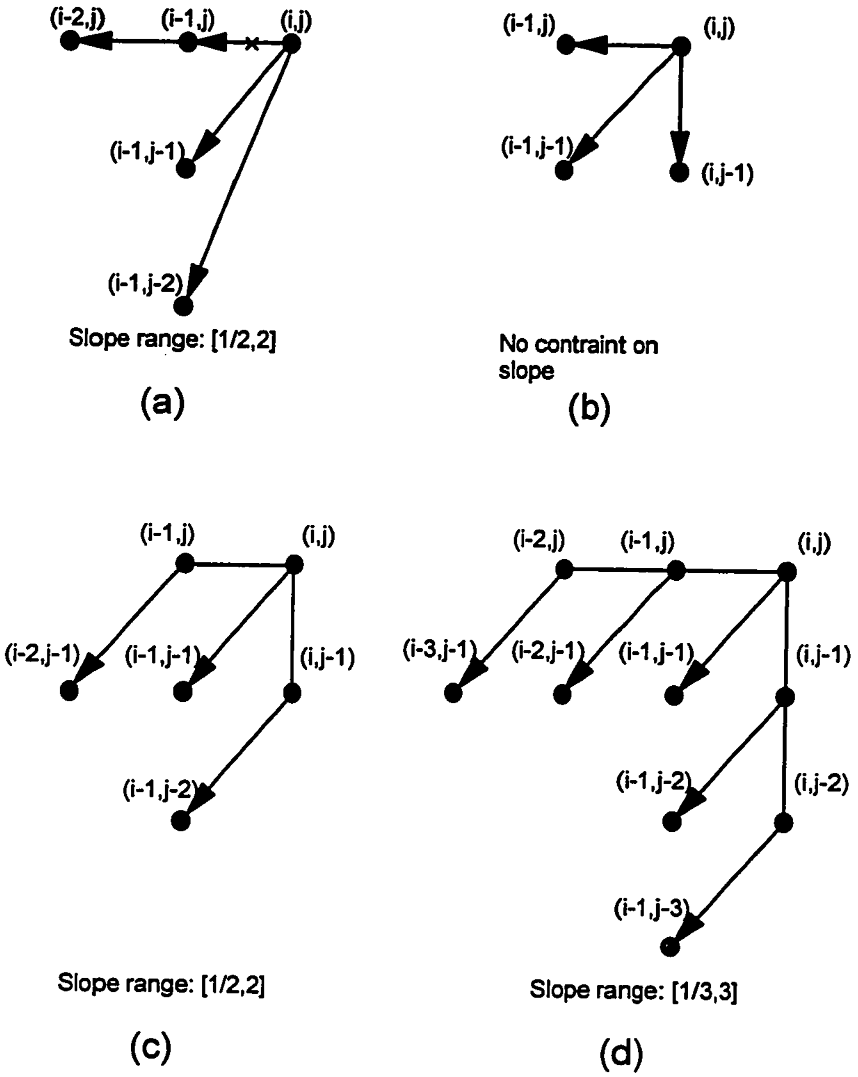


Figure 3.3: Typical local continuity constraints:
(a) Itakura local constraint, allowing slopes in $[1/2,2]$
(b) Sakoe-Chiba local constraint; no constraint on slope
(c) Sakoe-Chiba local constraint; allowing slopes in $[1/2,2]$
(d) Sakoe-Chiba local constraint; allowing slopes in $[1/3,3]$.

The way to read these local constraints can be illustrated with the following example. Consider the upper local transition of the Figure 3.3(c). It indicates that the only way to reach the (i,j) from the $(i-2,j-1)$ point is through the $(i-1,j)$ point. Moreover, all of them are symmetric since for each (i,j) point the possible predecessors are located in symmetrical local transitions about the diagonal.

One can extend these local constraints so that the desired range of slope is obtained. However, as it will be shown in the next section, this will substantially complicate the Dynamic Programming-based implementation. An easier way to impose constraints on the slope of the path is to use the local constraint shown in Figure 3.3(b), combined with a check on consecutive horizontal or vertical optimal local transitions. This modification will result in a symmetric constraint with the desired slope range. Thus, if m is the maximum number of allowable consecutive horizontal or vertical local transitions, the slope of path will be restricted between $1/(m+1)$ and $(m+1)$; i.e., a slope range of $[1/(m+1), m+1]$.

If the local constraints define a set of predecessors for each (i,j) point, the global constraints define a subset of the $t \times r$ grid as the actual search space. Most of them need not be explicitly imposed in the optimization problem. This is due to the fact that the implementation of most of the local continuity constraints automatically implies the global constraints. For example, assume that any of the local constraints of Figures 3.3(a) or 3.3(c) is used, in conjunction with the fixed-endpoint constraints of Eqs (3.9). Then the actual search space will be the area included by the lines of slope 1/2 and 2, emanating from the first $(1,1)$ and the last (t,r) path point. This is illustrated in the Figure 3.4(a): the search area is the shaded parallelogram (Itakura, 1975). In the case that the number of frames in the test pattern is twice (or half) of those in the reference pattern, the allowed search space is reduced to the diagonal line (Silverman and Morgan, 1990).

Figure 3.4(b) shows the band global constraint. This constraint does not allow the path to deviate $\pm$ M grid points from the linear path starting at the point $(1,1)$ (Sakoe and

Chiba, 1978). For a feasible search space to exist, M has to be at least equal to or greater than the absolute value of the difference between the number of feature vectors in the test and in the reference pattern, i.e., $M \geq |t - r|$. This global constraint is usually used in conjunction with the local constraint of Figure 3.3(b), that is, when no restriction is imposed on the slope of the path. The combination of the two constraints will prevent large deviations from a linear path, although this may be a indication of the dissimilarity between the two patterns. When the band constraint is present, this dissimilarity will appear as an inflated total distance between the two patterns.

Moreover, it is possible to combine different local and global constraints. For example, one can use the local constraint of Figure 3.3(c), together with the band global constraint. In such a case, the search space will be the intersection of the two shaded regions of Figures 3.4(a) and 3.4(b).

Finally, the local distance between two vectors defined in Eq (3.1) is not the only measure of dissimilarity that can be used. The type of distance used is mainly dependent on the type of features that each vector contains (O'Shaughnessy, 1986). If a quadratic distance is selected, then any arbitrary positive definite matrix **W** can be used. The Mahalanobis distance has its basis in statistical decision theory (Tou and Gonzales, 1974); however it may be difficult to obtain an accurate estimate of the covariance matrix of the features, **W**, if highly correlated features are used. The identity matrix **I** can also be used, if very little knowledge exists about the extracted features.

After introducing the main idea behind DTW, the various constraints that have to be imposed, and the selection of the local distance measure, the solution algorithm will now be presented.. In the next section, the optimization problem is defined and then its solution by the method of Dynamic Programming.

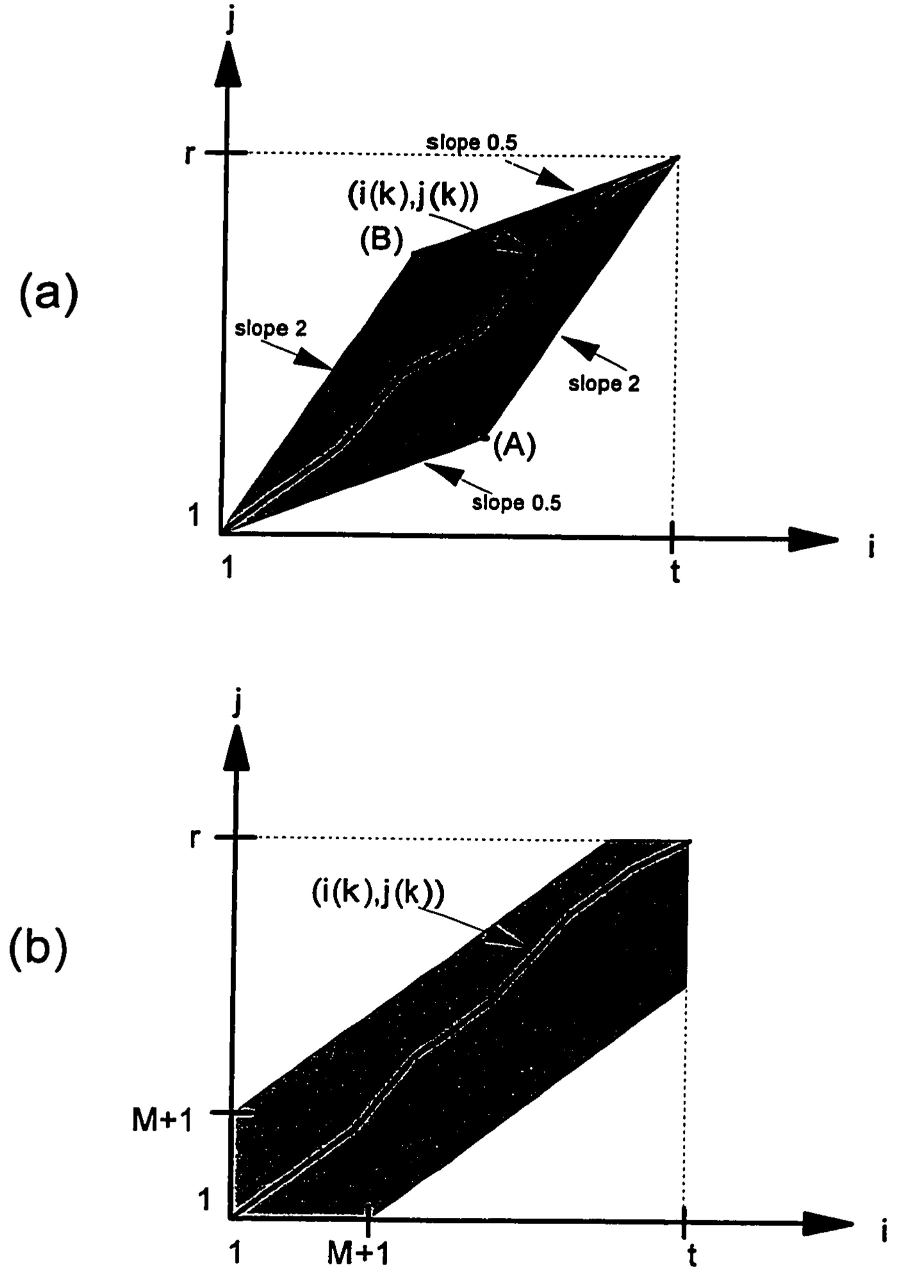


Figure 3.4: Typical global constraints:
(a) Itakura global constraint
(b) Sakoe-Chiba band constraint.

3.3 Solution Via Dynamic Programming

As mentioned above, the objective of DTW is to find the best path through a grid of t × r vector-to-vector distances such that some total distance measure between the two patterns is minimized. A general form for such a distance measure is (Sakoe and Chiba, 1978, Myers et al., 1980) :

$$D(\mathrm{t},\mathrm{r}) = \frac{\sum_{\mathrm{k}=1}^{\mathrm{K}} d(\mathrm{i}(\mathrm{k}),\mathrm{j}(\mathrm{k}))\,\mathrm{w}(\mathrm{k})}{\mathrm{N}(\mathrm{w})} \tag{3.11}$$

where: $D(\mathrm{t},\mathrm{r})$ is the normalized total distance between the patterns,

$d(\mathrm{i}(\mathrm{k}),\mathrm{j}(\mathrm{k}))$ is the local distance between the $\mathbf{T}(\mathrm{i}(\mathrm{k}),:)$ vector of the test pattern and the $\mathbf{R}(\mathrm{j}(\mathrm{k}),:)$ vector of the reference pattern,i.e., $d(\mathrm{i}(\mathrm{k}),\mathrm{j}(\mathrm{k})) = (\mathbf{T}(\mathrm{i}(\mathrm{k}),:) - \mathbf{R}(\mathrm{j}(\mathrm{k}),:))^{\mathrm{T}}\,\mathbf{W}\,(\mathbf{T}(\mathrm{i}(\mathrm{k}),:) - \mathbf{R}(\mathrm{j}(\mathrm{k}),:))$

w(k) is a nonnegative weighting function for the $d(\mathrm{i}(\mathrm{k}),\mathrm{j}(\mathrm{k}))$ local distance, and,

N(w) is a normalization factor which is a function of the weighting function w(k).

D(t,r) sums all the local distances $d(\mathrm{i}(\mathrm{k}),\mathrm{j}(\mathrm{k}))$ that lie along the path, weights them by w(k), and divides the sum by the normalization factor N(w).

Thus, the optimal path $\hat{\mathbf{F}}$ will result from the solution of the following minimization problem:

$$\hat{D}(\mathrm{t},\mathrm{r}) = \min_{\mathbf{F}} \left[\, D(\mathrm{t},\mathrm{r}) \,\right] \tag{3.12a}$$

and

$$\hat{\mathbf{F}} = \arg\min_{\mathbf{F}}[D(\mathrm{t},\mathrm{r})] \tag{3.12b}$$

where $\hat{D}(t, r)$ is the Minimum Normalized Total Distance between the two patterns.

The N(w) parameter is a scalar and serves as a normalization factor for the distance estimation. Its value will depend on the type of the weighting function w(k) that is used. Its purpose is to make the normalized total distance independent of the number of path points K and the lengths of the two patterns, t and r, so that distances from different **R-T** pairs can be compared. For example, consider the test pattern **T** and two reference patterns $\mathbf{R_1}$ and $\mathbf{R_2}$ with r_1 and r_2 feature vectors respectively; assume $r_1 > r_2$. Since $\mathbf{R_1}$ is a pattern of longer duration than $\mathbf{R_2}$, an unnormalized total distance between $\mathbf{R_1}$ and **T** will involve the summation of more local distances than between $\mathbf{R_2}$ and T. This can possibly lead to incorrect classification, even if **T** is actually more like $\mathbf{R_1}$ than $\mathbf{R_2}$. Thus, the total distance should be normalized to take into account these possible differences in the duration of the patterns.

The weighting function w(k) depends on the local continuity constraints and serves two purposes. The first is to provide more flexibility in the DTW algorithm by weighting the local distance $d(\mathrm{i(k)}, \mathrm{j(k)})$, depending on the local transition by which the $(\mathrm{i(k)}, \mathrm{j(k)})$ path point can be reached from the $(\mathrm{i(k-1)}, \mathrm{j(k-1)})$ previous path point. As Figures 3.3 show, for any (i, j) point in the grid, a set of allowable local transitions is defined by which the (i, j) point can be reached; w(k) allows some local transitions to be treated preferentially (i.e., assign small weights to them) over some others. The second purpose of w(k) is to make the normalized total distance independent of the number of the path points by imposing an appropriate value for the normalization factor N(w).

The importance of the last point can be seen in Eqs (3.11) and (3.12). The optimization problem of Eq (3.12a) uses a rational function as a criterion. In principle, it is possible to solve such optimization problems. However, Dynamic Programming cannot be used anymore, since in Dynamic Programming the global solution is obtained recursively by a series of local solutions that do not consider the best global path at all. Dynamic Programming retrieves the optimal path at the end, assuming that the optimal total distance has been found. Thus, problems like the one of Eq (3.12a), where the minimization depends simultaneously on both the total distance and the path, cannot be solved by Dynamic Programming. On the other hand, if the normalization factor N(w) is independent of the optimal path, the optimization problem reduces to:

$$\hat{D}(t,r) = \frac{1}{N(w)} \cdot \min_{F} \left[\sum_{k=1}^{K} d(i(k), j(k))\, w(k) \right] \tag{3.13}$$

and this problem lends itself to a Dynamic Programming-based solution (Myers et al., 1980).

Many different weighting functions have been proposed in the literature of DTW (Itakura, 1975, Sakoe and Chiba, 1978, Myers et al., 1980). The two most common ones are:

symmetric: $$w(k) = (\, i(k) - i(k-1)\,) + (\, j(k) - j(k-1)\,) \tag{3.14a}$$

$$\text{with } i(0) = j(0) = 0$$

asymmetric: $$w(k) = (\, i(k) - i(k-1)\,) \text{ with } i(0) = 0 \tag{3.14b}$$

The weighting function of Eq (3.14a) weights a local transition, from the (k-1)th path point to the kth path point, according to the number of horizontal and vertical steps that need to be taken for that particular local transition. Both horizontal and vertical steps are considered equivalent. Thus, it is a symmetric weighting function. On the other hand, Eq (3.14b) considers only the number of the horizontal steps required for a local transition and, for that reason it is an asymmetric weighting function. Figure 3.5 illustrates these weighting functions, when the local continuity constraint of Figure 3.3(b) is applied. The coefficients for each local transition are the result of the weighting functions of Eq (3.14a) and Eq (3.14b). Similar coefficients are obtained when different local continuity constraints are applied.

As Figure 3.5(b) shows, if the weighting function of Eq (3.14b) is used, the vertical local transitions are not weighted at all. As a result, the local distance $d(\mathrm{i(k), j(k)})$ will be omitted from the total distance, if the link to the previous path point $(\mathrm{i(k-1), j(k-1)})$ is a vertical transition. To eliminate this nonphysical occurrence, the nonzero weighting coefficients of the non-vertical arcs are averaged and equally distributed to all arcs. This is illustrated in Figure 3.6, where the local continuity constraint of Figure 3.3(c) is used. Figure 3.6(a) shows the coefficients if the symmetric weighting function of Eq (3.14a) is used. Figures 3.6(b) and 3.6(c) show the coefficients if the asymmetric weighting function of Eq (3.14b) is used, both before and after the averaging of the weighting coefficients of the non-vertical arcs.

Now the normalization factor N(w) can be defined. The normalized total distance, as defined in Eq (3.11), is an average distance between the two patterns along the optimal path. As such, it is reasonable to make it equal to the number of the local distances computed along the path (Sakoe and Chiba, 1978, Myers et al., 1980):

$$\mathrm{N(w)} = \sum_{\mathrm{k=1}}^{\mathrm{K}} \mathrm{w(k)} \qquad (3.15)$$

Hence, if the weighting functions of Eqs (3.14a) and (3.14b) are used, the corresponding normalization factors are:

$$\begin{aligned} \mathrm{N(w)} &= \sum_{\mathrm{k=1}}^{\mathrm{K}} \left[(\mathrm{i(k)-i(k-1)}) + (\mathrm{j(k)-j(k-1)})\right] && = \\ &= \mathrm{i(K)-i(0)+j(K)-j(0)} && = \mathrm{t+r} \end{aligned} \qquad (3.16a)$$

$$\mathrm{N(w)} = \sum_{\mathrm{k=1}}^{\mathrm{K}} \left[\mathrm{i(k)-i(l-1)}\right] = \mathrm{i(K)-i(0)} = \mathrm{t} \qquad (3.16b)$$

and they are both independent of the optimal path.

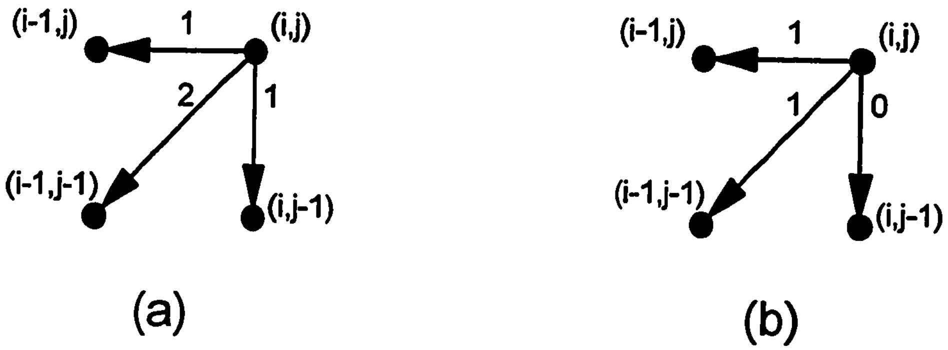


Figure 3.5: Local continuity constraint with no constraint on slope and
(a) symmetric weighting function
(b) asymmetric weighting function.

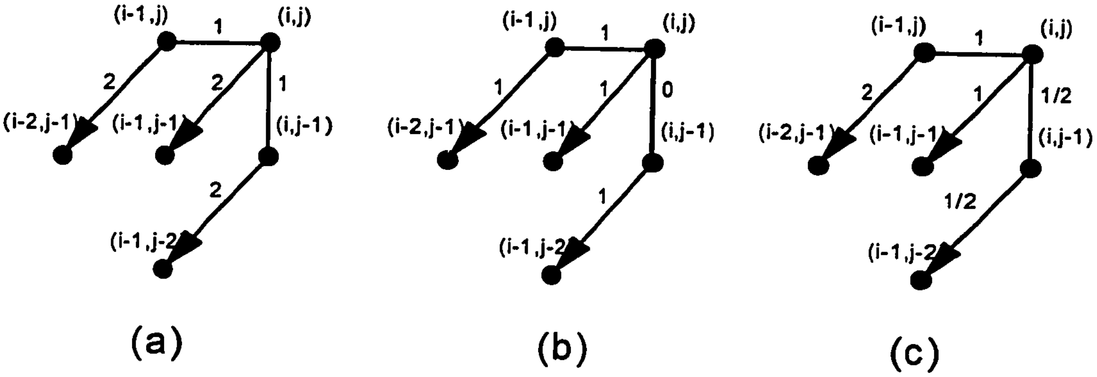


Figure 3.6: Local continuity constraint with $[1/3,3]$ range slope and
(a) symmetric weighting function
(b) asymmetric weighting function
(c) smoothed asymmetric weighting function.

It was mentioned in Section 3.1 that there are symmetric and asymmetric versions of the DTW algorithm. A symmetric DTW algorithm will result if a symmetric local continuity constraint is used together with a symmetric weighting function. Conversely, if either an asymmetric local constraint is used (e.g., Itakura local constraint) and/or an asymmetric weighting function, then the resulting DTW algorithm will be asymmetric.

The Dynamic Programming-based solution of the optimization problem shown in Eq (3.13) will now be presented. The theoretical basis of Dynamic Programming is an important property of multistage optimization problems, called the Principle of Optimality, which states that *"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision"* (Bellman and Dreyfus, 1962, Bertsekas, 1987).

For the DTW problem, the Principle of Optimality is translated into the following two rules (Myers et al., 1980, Ney, 1984, Silverman and Morgan, 1990):

Rule (I): Let $\hat{\mathbf{F}}$ be the optimal global path on the $t \times r$ grid. If $\hat{\mathbf{F}}$ goes through an (i, j) point, then the optimal path to the (i, j) point is part of $\hat{\mathbf{F}}$.

Rule (II): The optimal path to the (i, j) point depends only on previous grid points (i', j'); i.e., $i > i'$, $j > j'$.

The above rules, used in any variant of DTW, define a recursive Dynamic Programming relationship. This recursive relationship depends on the type of local continuity constraints and on the weighting function. The purpose of the following examples is to show how each of these two decision parameters, in combination with the global constraints and the endpoint constraints, affect the solution procedure. The presentation will be done from the point of view of practical implementation. Also, the relevance of each algorithm to Chemical Engineering applications will be illustrated.

3.3.1 DTW: Example 1

In this example, the simplest symmetric DTW algorithm (i.e,. without any constraint on the slope of the optimal path) will be presented. This algorithm could be used for the off-line comparison of patterns and assumes that the extreme points of the patterns are exactly known.

Assume that the fixed-endpoint constraints of Eqs (3.9) are used, together with the symmetric local continuity constraint and the weighting function of Figure 3.5(a). Also, assume that the band constraint of Figure 3.4(b) is used.

Let $\mathbf{D}_A(\mathrm{i},\mathrm{j})$ be the minimum accumulated total distance from point $(1,1)$ to point (i,j), i.e.,

$$\mathbf{D}_A(\mathrm{i},\mathrm{j}) \;=\; \min_{\mathbf{F}'}\left[\sum_{k=1}^{K'} d(\mathrm{i}(k),\mathrm{j}(k))\;\mathrm{w}(k)\right] \tag{3.17}$$

where $\mathbf{F}'$ is any path, $\hat{\mathbf{F}}'$ is the optimal path and K' are the number of path points. Thus, Eq (3.13) becomes:

$$\hat{D}(\mathrm{t},\mathrm{r}) \;=\; \frac{1}{\mathrm{t}+\mathrm{r}}\mathbf{D}_A(\mathrm{t},\mathrm{r}) \tag{3.18}$$

(as mentioned, $\mathrm{N}(\mathrm{w}) = \mathrm{t}+\mathrm{r}$ for this type of symmetric weighting function, Eq (3.16a)).

The assumed local continuity constraint implies that the (i,j) point can only be reached by either the $(\mathrm{i}-1,\mathrm{j})$, or the $(\mathrm{i}-1,\mathrm{j}-1)$ or the $(\mathrm{i},\mathrm{j}-1)$ point. However, for any of these three possible predecessor points, there is a minimum accumulated total distance. Due to Rule (I), if the (i,j) point lies on the optimal path, then the transition from the three possible predecessors has to be optimal. Also, due to Rule (II), this optimal transition will not be affected by any subsequent decision. Thus, according to the Rules

(I) and (II), the chosen local continuity constraint and the symmetric weighting function, $\mathbf{D}_{\mathrm{A}}(\mathrm{i},\mathrm{j})$ will be found by solving the following simple optimization problem:

$$\mathbf{D}_{\mathrm{A}}(\mathrm{i},\mathrm{j}) \quad = \quad \min\begin{Bmatrix}\mathbf{D}_{\mathrm{A}}(\mathrm{i}-1,\mathrm{j})+d(\mathrm{i},\mathrm{j}) \\ \mathbf{D}_{\mathrm{A}}(\mathrm{i}-1,\mathrm{j}-1)+2\,d(\mathrm{i},\mathrm{j}) \\ \mathbf{D}_{\mathrm{A}}(\mathrm{i},\mathrm{j}-1)+d(\mathrm{i},\mathrm{j})\end{Bmatrix} \tag{3.19}$$

Now, because at this point it is not known whether the (i,j) point lies on the optimal path, the decision on which of the three alternatives in Eq (3.19) was selected has to be stored. This procedure (i.e., Eq (3.19) and storage of the optimal local transition), has to be done for all the (i,j) points that lie in the allowable search area; i.e., the shaded area of Figure 3.4(b). Note however, that if the optimal path does not need to be reconstructed, these optimal local transitions do not have to be stored.

Thus, one would start from the point $(1,1)$ as[2]:

$$\mathbf{D}_{\mathrm{A}}(1,1)=2\,d(1,1) \tag{3.20}$$

and would proceed recursively via two iterations, one nested in the other, until the (t,r) grid point is reached. This constitutes the forward phase of the Dynamic Programming recursion. The outer iteration will progress on the time index, i, of the pattern placed on the horizontal axis, whereas the inner iteration will progress on the allowable range of the time index, j, of the pattern placed on the vertical axis. As Figure 3.4(b) shows, for any value of the horizontal time index, there is an allowable range (shaded region) for the vertical time index. Thus, the index of the outer loop, i, goes from 1 to t, while the range of the inner loop index, j, depends of the value of the outer iteration index. This range (for the particular global constraint of Figure 3.4(b)) is constructed as follows.

[2] The weight of 2 is according to the assumed weigthing function of Eq (3.14).

Let **l** be the lower and **u** the upper limit for the index of the inner iterations. These are both vectors of $t \times 1$ dimension. Vector **l** is constructed as follows (see Figure 3.4(b)):

$$\mathbf{l}(i) = 1 \quad , \quad 1 \leq i \leq M+1 \tag{3.21a}$$

$$\mathbf{l}(i) = i-(M+1) \quad , \quad M+2 \leq i \leq t \tag{3.21b}$$

while vector **u** is constructed as follows:

$$\mathbf{u}(i) = i+M \quad , \quad 1 \leq i \leq r-M \tag{3.22a}$$

$$\mathbf{u}(i) = r \quad , \quad r-M+1 \leq i \leq t \tag{3.22b}$$

M is a parameter that has to be chosen; it must be $M \geq |t-r|$ so that the band of width $2M$ includes the (t,r) grid point and the two constraints (i.e., the band constraint and the fixed-endpoint constraints) are compatible. In general, M should reflect the uncertainty in locating the first and the last point of the patterns. Setting M too small may result in a large distance between two similar patterns that are badly synchronized; setting M too large helps much more the comparison between dissimilar patterns than between similar patterns (Levinson et al., 1979).

The iterative procedure of Eq (3.19) finishes when the $\mathbf{D}_A(t,r)$ distance is computed and, subsequently, the minimum normalized total distance $\hat{D}(t,r)$ is computed via Eq (3.18). To reconstruct the optimal path one has to proceed in a backward manner, starting from the (t,r) point and using the stored information on the optimal decisions at the allowable (i,j) grid points. Thus, first the predecessor of the (t,r) point is located, then the predecessor of the latter is located and this is repeated until the $(1,1)$ point is reached. The following is an algorithmic summary of the solution:

Step 1: Give value for M; construct **l** and **u**; $\mathbf{D}_A(1,1) = 2\,d(1,1)$

```
Step 2:  For $i = 1,\ldots,t$
             For $j = \mathbf{l}(i),\ldots,\mathbf{u}(i)$
                 $\mathbf{D}_A(i,j) = \min\left\{\begin{array}{c} \mathbf{D}_A(i-1,j)+d(i,j) \\ \mathbf{D}_A(i-1,j-1)+2\,d(i,j) \\ \mathbf{D}_A(i,j-1)+d(i,j) \end{array}\right\}$
                 Store the optimal predecessor for the $(i,j)$ point.
             End
         End
Step 3:  Minimum Normalized Total Distance: $\hat{D}(t,r) = \frac{1}{t+r}\mathbf{D}_A(t,r)$
Step 4:  Reconstruct the optimal path, $\hat{\mathbf{F}}$, starting from the $(t,r)$ point and travel
         backwards as the optimal predecessor indices dictate until point $(1,1)$ is reached.
```

In terms of memory requirements, these are not large if only the minimum distance is sought. In that case, only two vectors of accumulated total distances have to stored. At any outer iteration, i, a vector that stores the distances $\mathbf{D}_A(i-1,j)$, $j = \mathbf{l}(i-1),\ldots,\mathbf{u}(i-1)$, is required and the $\mathbf{D}_A(i,j)$, $j = \mathbf{l}(i),\ldots,\mathbf{u}(i)$, vector of distances is computed via Eq (3.19). At the next iteration, $i+1$, the $\mathbf{D}_A(i-1,j)$ distances are not required anymore and their memory space can be used to store the $\mathbf{D}_A(i,j)$ distances. The memory space for the latter can be used to store the new $\mathbf{D}_A(i+1,j)$ distances and the whole storage-updating procedure is repeated.

Also, for some values of the outer and/or the inner iteration indices, the $\mathbf{D}_A(i,j)$ distances may not be defined. For example, when $i = 1$ and $j = 2$, applying Eq (3.19) requires the $\mathbf{D}_A(0,2)$, $\mathbf{D}_A(0,1)$ and $\mathbf{D}_A(1,1)$ distances, each associated with a possible predecessor; however $\mathbf{D}_A(0,2)$ is not defined since the $(0,2)$ grid point does not exist. In such a situation, this predecessor is not considered at all and Eq (3.19) is implemented with the remaining two possible predecessors.

If the optimal path is also sought, then for any (i, j) point in the allowable search area in the grid, an integer index (from a set of three indices, each associated with a possible predecessor) has to be stored indicating the optimal predecessor. This is done because any of the points in the allowable space can be a point of the optimal path[3]. Thus, for this example, this information has to be stored for all points that lie in the shaded region of Figure 3.4(b).

3.3.2 DTW: Example 2

This example will illustrate the implementation of an asymmetric DTW algorithm that also assumes that the extreme points are exactly known in both patterns. As in the previous example, this algorithm could also be used for off-line comparison of patterns. However it treats one pattern differently from the other and, as such, it requires a decision to be made by the user.

Assume that the fixed-endpoint constraints, the Itakura local constraint of Figure 3.3(a), and the asymmetric weighting function of Eq (3.14b) are used. This is then an asymmetric DTW algorithm that maps the horizontal time index, i, onto the vertical time index, j. Also, as mentioned in Section 3.2, these local constraints impose the shaded parallelogram of Figure 3.4(a) as the search space for the optimal path.

First, both the lower, **l**, and the upper, **b**, limit vectors (of dimension $t \times 1$) for the index of the inner iterations are constructed. Vector **l** is constructed as follows: draw two lines of slope 1/2 and 2, emanating from points $(1,1)$ and (t, r), respectively; let A be the point of intersection (see Figure 3.4(a)). The piecewise linear curve that starts at point $(1,1)$ with slope 1/2, changes slope at point A, and continues with slope 2 up to point (t, r), is the lower limit, **l**, for the inner iteration. Similarly, one can construct the upper

[3] Remember that the optimal path is not known during the forward phase, but it is found at the end, once the (t, r) point is reached.

limit vector **u**. Once **l** and **u** are constructed, the Dynamic Programming recursion starts. The algorithm of the method is as follows:

```
Step 1:  Construct l and u; $\mathbf{D_A}(1,1) = d(1,1)$
Step 2:  For $i = 1,\dots,t$
             For $j = \mathbf{l}(i),\dots,\mathbf{u}(i)$
                 $\mathbf{D_A}(i,j) = \min\begin{Bmatrix} \left[\mathbf{D_A}(i-1,j)+d(i,j)\right] \text{ or } \left[\infty \text{ if condition (A)}^{*}\right] \\ \mathbf{D_A}(i-1,j-1)+d(i,j) \\ \mathbf{D_A}(i-1,j-2)+d(i,j) \end{Bmatrix}$    (3.23)
                 *Condition (A): predecessor of point (i-1,j) is the point (i-2,j).
                 Store the optimal predecessor for the $(i,j)$ point.
             End
         End
Step 3:  Minimum Normalized Total Distance: $\hat{D}(t,r) = \frac{1}{t}\mathbf{D_A}(t,r)$
Step 4:  Reconstruct the optimal path, $\hat{\mathbf{F}}$, starting from the $(t,r)$ point and travel
         backwards as the optimal predecessor indices dictate until point $(1,1)$ is reached.
```

According to the Itakura constraint, two consecutive horizontal local transition are not allowed, and this is what Condition (A) states. Therefore, for any outer iteration, i, the optimal predecessors of the points in the previous iteration, $i-1$, have to be kept in memory. This is a difference from the first example, where the implementation of the local constraint did not require any information regarding the past optimal local transitions.

Another difference is the memory requirements to implement the distance computations. Although one can use two storage vectors for the optimal accumulated distances as in the previous example, one storage vector is also sufficient. This can be achieved if the computations in the inner iteration are performed 'down the column', i.e., for any outer iteration, the inner iteration index is decreasing instead of increasing: $\forall$ i, $j = \mathbf{u}(i),\dots,\mathbf{l}(i)$ instead of $j = \mathbf{l}(i),\dots,\mathbf{u}(i)$ (Silverman and Morgan, 1990).

Finally, in order to reconstruct the optimal path, indices have to be stored that indicate the optimal predecessor for all the (i, j) points that lie in the shaded parallelogram of Figure 3.4(a).

3.3.3 DTW: Examples 3, 4 and 5

These examples will illustrate the implementation of symmetric DTW algorithms with constraints on the slope of the optimal path. Again the fixed-endpoint constraints are used. The two patterns are considered equally important since the algorithms are symmetric. These DTW algorithms could also be used for the off-line comparison of patterns. However, the slope constraints prevent the excessive distortion of the time axes of both patterns. This will be beneficial in situations where the duration of a pattern is an important feature for its classification. In such a situation, constraining the slope of the optimal path will prevent long patterns from being mapped onto very short patterns; this will prevent wrong classifications.

Assume that the fixed-endpoint constraints, and the Sakoe-Chiba local constraint/symmetric weighting function of Figure 3.6(a) are used. These local constraints imply the same allowable search region as in the previous example; i.e., the shaded parallelogram of Figure 3.4(a). After defining the lower and the upper limit vectors for the index of the inner iterations, at any allowable point (i, j) the following simple optimization has to be performed:

$$\mathbf{D}_A(i,j) = \min\begin{Bmatrix} \mathbf{D}_A(i-2,j-1) + 2\,d(i-1,j) + d(i,j) \\ \mathbf{D}_A(i-1,j-1) + 2\,d(i,j) \\ \mathbf{D}_A(i-1,j-2) + 2\,d(i,j-1) + d(i,j) \end{Bmatrix} \tag{3.24}$$

To implement these distance computations, at any outer iteration, 3 distance vectors need to be stored: $\mathbf{D}_A(i,j)$, $\mathbf{D}_A(i-1,j)$ and $\mathbf{D}_A(i-2,j)$, $\forall$ allowable j. This local constraint requires no information on past optimal predecessors.

Now assume that the local constraint of Figure 3.3(d) is used, together with the symmetric weighting function of Eq (3.16a). The allowable search area will again be a parallelogram, but a wider one this time, with 1/3 and 3 being the slopes of its sides. The optimization problem that has to be solved at each allowable grid point is:

$$\mathbf{D}_A(i,j) \quad = \quad \min\left\{\begin{matrix} \mathbf{D}_A(i-3,j-1)+2\,d(i-2,j)+d(i-1,j)+d(i,j) \\ \mathbf{D}_A(i-2,j-1)+2\,d(i-1,j)+d(i,j) \\ \mathbf{D}_A(i-1,j-1)+2\,d(i,j) \\ \mathbf{D}_A(i-1,j-2)+2\,d(i,j-1)+d(i,j) \\ \mathbf{D}_A(i-1,j-3)+2\,d(i,j-2)+d(i,j-1)+d(i,j) \end{matrix}\right\} \tag{3.25}$$

To implement these distance computations, at any outer iteration, 4 distance vectors need to be stored: $\mathbf{D}_A(i,j)$, $\mathbf{D}_A(i-1,j)$, $\mathbf{D}_A(i-2,j)$ and $\mathbf{D}_A(i-3,j)$ for any allowable j. Again, this local constraint requires no information on past optimal predecessors.

One could generalize these local constraints so that a desired range for the slope of the optimal path is obtained. However, the selection scheme of possible predecessors would be more complicated. A simpler way to do the same thing, at the burden of a check at each point, would be the following symmetric local constraint:

$$\mathbf{D}_A(i,j) = \min\left\{\begin{matrix} \left[\mathbf{D}_A(i-1,j)+d(i,j)\right] \text{ or } \left[\infty \text{ if condition (B)}\right] \\ \mathbf{D}_A(i-1,j-1)+2d(i,j) \\ \left[\mathbf{D}_A(i,j-1)+d(i,j)\right] \text{ or } \left[\infty \text{ if condition (C)}\right] \end{matrix}\right\} \tag{3.26}$$

where: Condition (B): predecessor of point $(i-1,j)$ is the point $(i-m-1,j)$ through m consecutive horizontal moves.

Condition (C): predecessor of point $(i,j-1)$ is the point $(i,j-m-1)$ through m consecutive vertical moves.

This local constraint imposes a range of $[1/(m+1), m+1]$ for the slope of the optimal path. Also, the search area it imposes is a parallelogram with $1/(m+1)$ and $m+1$ slopes for its sides. To implement distance computations, at any outer iteration, only 2 distance vectors need to be stored: $\mathbf{D}_A(i,j)$, and $\mathbf{D}_A(i-1,j)$, $\forall$ allowable j. However, m vectors that contain the optimal predecessors of the points: $(i-1,j),\ldots,$ $(i-m,j)$ for any allowable j, have to be stored.

Finally, for all the three cases discussed at this subsection, $\mathbf{D}_A(1,1)=2d(1,1)$ and $\hat{D}(t,r)=\frac{1}{t+r}\mathbf{D}_A(t,r)$ since all of them use the symmetric weighting function of Eq (3.14a).

3.4 Relaxation of the Fixed-Endpoint Constraints

In all the examples presented so far, the fixed-endpoint constraints were used to locate the first and the last point of the optimal path. However, this may be a strong assumption, particularly when there is uncertainty in locating the boundary points of the patterns. This section describes how the fixed-endpoint constraints can be relaxed for the most common local continuity constraints. The first example illustrates how this can be done when an asymmetric DTW algorithm is used, while the second example treats the case of a symmetric DTW algorithm.

3.4.1 DTW: Example 6 - Itakura Local Constraint

It was mentioned in Example 2 that for the Itakura local constraints, the normalization factor N(w) is taken from Eq (3.16b) and is equal to t; i.e., the length of the pattern placed on the horizontal axis. The reason was that $\mathbf{D}_A(t,r)$ involved the summation of t local distances, and this was independent of the optimal path.

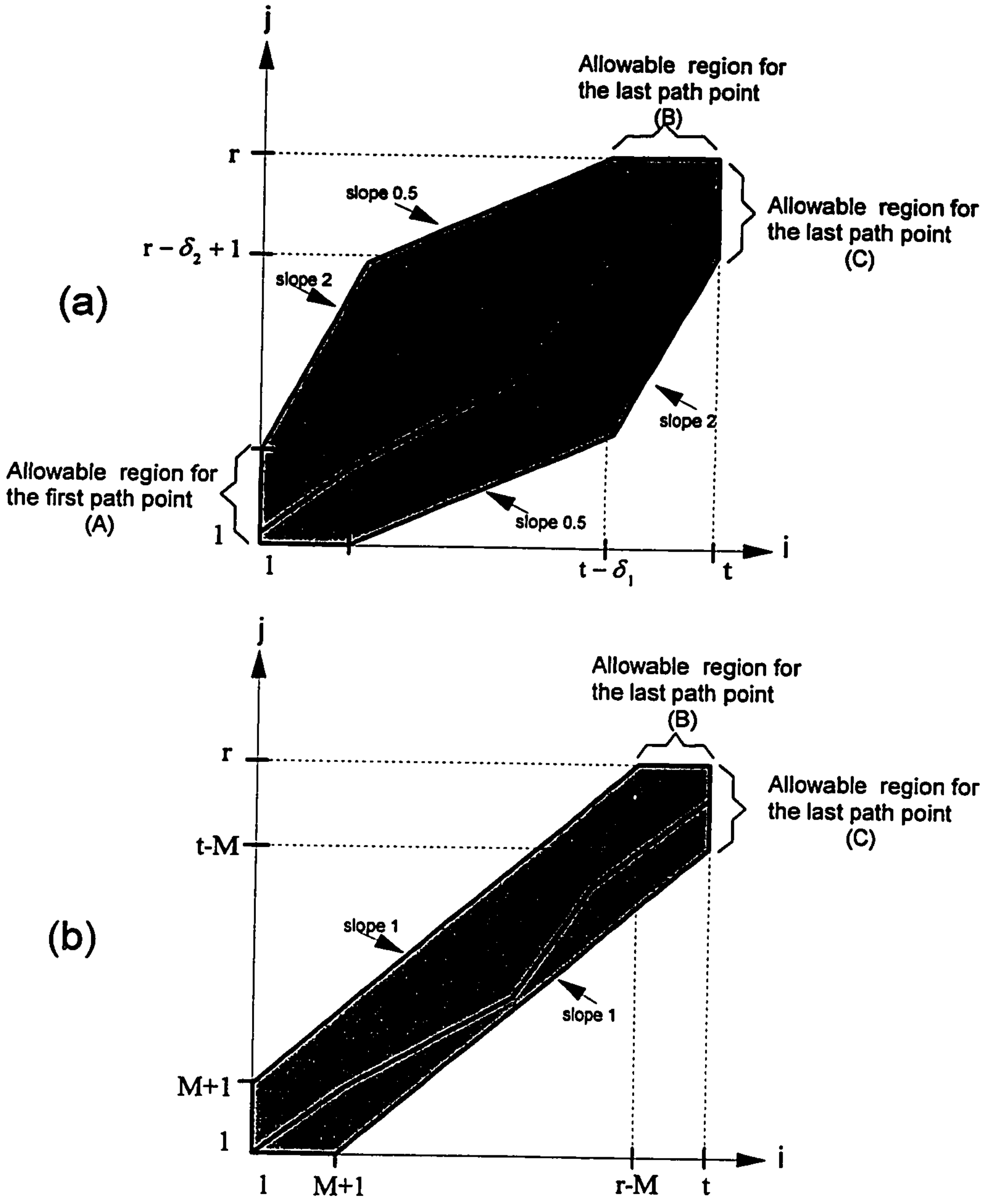

(a)
j
Allowable region for the last path point (B)
r
slope 0.5
$r - \delta_2 + 1$
Allowable region for the last path point (C)
slope 2
slope 2
Allowable region for the first path point (A)
slope 0.5
1
1
$t - \delta_1$
t
i
(b)
j
Allowable region for the last path point (B)
r
Allowable region for the last path point (C)
t-M
slope 1
slope 1
M+1
1
1
M+1
r-M
t
i

Figure 3.7: Relaxation of the fixed-endpoint constraints and modification of the search area for optimal path when:
(a) the Itakura local constraints are used
(b) the Sakoe-Chiba local constraints of Figure 3.5(a) are used.

As Figure 3.7(a) illustrates, if one could locate the first path point in Area (A) and the last point in Area (C), again t local distances would have to be summed; any two points in the two areas are separated by $t-1$ horizontal transitions. One could then allow δ_2 points: $(1,j)$, $1 \le j \le \delta_2$, among which the first path point will lie and also δ_2 points[4]: (t,j), $r - \delta_2 + 1 \le j \le r$, among which the last path point will lie.

Thus, each of the δ_2 points in Area (A) can be the start for the optimal path and as such, it will not have a predecessor. Algorithmically, this is achieved by setting the minimum accumulated total distance $\mathbf{D}_A(1,j)$ at each of these points equal to the local distance $d(1,j)$, $1 \le j \le \delta_2$. One the other side of the grid, instead of setting $\hat{D}(t,r) = \mathbf{D}_A(t,r)$ and tracking backwards along the path from point (t,r), one could locate the minimum among the distances in the points of Area (C), $\mathbf{D}_A(t,j)$, $r - \delta_2 + 1 \le j \le r$. This would then be the minimum accumulated total distance between the two patterns; the point at which this distance occurs will be the last point of the optimal path. The procedure is a generalization of the standard implementation as described in Example 2.

One could also extend the range for the last point of the path by including $\delta_1 + 1$ points in the Area (B) of Figure 3.7(a); i.e., the points (i,r), $t - \delta_1 \le i \le t$. However, the corresponding accumulated distances $\mathbf{D}_A(i,r)$ involve the summation of a variable number of local distances: each $\mathbf{D}_A(i,r)$ is obtained by summing i local distances. Thus, comparing these total distances is not consistent and some normalization has to be performed; the following is a reasonable one (Rabiner et al., 1978):

$$\mathbf{D}_{A,\mathrm{Norm}}(i,r) = \mathbf{D}_A(i,r)\frac{t}{i}, \quad t - \delta_1 \le i \le t \qquad (3.27)$$

The $\mathbf{D}_{A,\mathrm{Norm}}(i,r)$ distances can then be considered, together with the $\mathbf{D}_A(t,j)$ distances mentioned above. The minimum of them is the final result and the corresponding point is

[4] Areas (A) and (C) need not have the same number of points; here this is done for simplicity.

the last point of the optimal path. Note that this normalization is a heuristic; strictly speaking, this is a problem where the number of the local distances summed depends on the optimal path and, as such, cannot be solved by Dynamic Programming.

Finally, it is possible to disengage the check on two consecutive horizontal transitions in the boundaries of the **T** pattern. The result in the allowable search area for the optimal path is the shaded area of Figure 3.7(a), where a maximum number of δ_1 consecutive horizontal transitions[5] are allowed in the beginning and at the end of the **T** pattern. In algorithmic form, the summary of the modifications discussed above is:

```
Step 1:  Give values for $\delta_1, \delta_2$; construct $\mathbf{l}$ and $\mathbf{u}$
Step 2:  Set $\mathbf{D}_A(1,j) = d(1,j), \quad 1 \le i \le \delta_2$
Step 3:  For $i = 1,\dots,t$
             For $j = \mathbf{l}(i),\dots,\mathbf{u}(i)$
                 $\mathbf{D}_A(i,j) = \min\left\{\begin{matrix} \mathbf{D}_A(i-1,j)+d(i,j) \\ \mathbf{D}_A(i-1,j-1)+d(i,j) \\ \mathbf{D}_A(i-1,j-2)+d(i,j) \end{matrix}\right\} \quad \text{if } 1 \le i \le \delta_1 + 1\text{, or, } t-\delta_1+1 \le i \le t$

                 $\mathbf{D}_A(i,j) = \min\left\{\begin{matrix} [\mathbf{D}_A(i-1,j)+d(i,j)] \text{ or } [\infty \text{ if condition (A)}^*] \\ \mathbf{D}_A(i-1,j-1)+d(i,j) \\ \mathbf{D}_A(i-1,j-2)+d(i,j) \end{matrix}\right\}\text{, otherwise}$

                 *Condition (A): predecessor of point (i-1,j) is the point (i-2,j).
                 Store the optimal predecessor for the $(i,j)$ point.
             End
         End
Step 4:  Minimum Normalized Total Distance:
         $\hat{D}(t,r) = \frac{1}{t}\min\begin{bmatrix} \mathbf{D}_A(t,j), & r-\delta_2+1 \le j \le r \\ \mathbf{D}_A(i,r)\frac{t}{i}, & t-\delta_1 \le i \le t \end{bmatrix} \qquad (3.28)$
```

[5] Again, this number need not be equal to the width of the Area (B) of Figure 3.7(a); here this is done for simplicity.

Step 5: Reconstruct the optimal path, $\hat{\mathbf{F}}$, starting from the point in which $\hat{D}(\mathrm{t,r})$ occurs (last path point), travel backwards as the optimal predecessor indices dictate and locate the first path point.

One could ask why it was not suggested to locate the first path point at a set of points $(\mathrm{i},1)$, $1 \le \mathrm{i} \le \delta_1 + 1$. Of course, one could treat these points as having no predecessor by setting their accumulated distances equal to the local ones: $\mathbf{D}_{\mathrm{A}}(\mathrm{i},1) = d(\mathrm{i},1)$. However, the minimization problem in Eq (3.28) will not be consistent because the accumulated distances will involve the summation of an unknown number of local distances depending on the optimal path. As already mentioned, such a problem cannot be solved by Dynamic Programming. Moreover, a heuristic similar to the one above cannot be used; the normalization factor (i.e., number of local distances summed) will be unknown since the first path point is found at the end of the backward step. However, allowing δ_1 consecutive horizontal transitions in the beginning partially compensates for this prohibitive feature.

3.4.2 DTW: Example 7 - Sakoe-Chiba Local Constraints

Let the Sakoe-Chiba local constraints/symmetric weighting function of Figure 3.5(a) be used, together with the band global constraint (see Figure 3.7(a)). If the fixed-endpoint constraints are used, the $\mathbf{D}_{\mathrm{A}}(\mathrm{t,r})$ distance will involve the summation of $\mathrm{t}+\mathrm{r}$ local distances (which is the normalization factor). On the other hand, if only the first path point is fixed, then one could locate the last point, as one of the points in Areas (B) and (C). The accumulated distances $\mathbf{D}_{\mathrm{A}}(\mathrm{i,j})$ at any of these points will involve the summation of $\mathrm{i}+\mathrm{j}$ local distances; as such, a similar heuristic normalization has to be performed so that their comparison is consistent. One the other hand, if the first path point is not fixed but allowed to lie in some region close to the origin, the number of local distances involved will not be known. Thus, the first path point has to be fixed at point $(1,1)$. For this example, the algorithm is as follows:

```
Step 1:  Give value for M; construct $\mathbf{l}$ and $\mathbf{u}$; $\mathbf{D}_\mathbf{A}(1,1) = 2d(1,1)$
Step 2:  For $i = 1,\ldots,t$
             For $j = \mathbf{l}(i),\ldots,\mathbf{u}(i)$
                 $\mathbf{D}_\mathbf{A}(i,j) = \min\begin{Bmatrix} \mathbf{D}_\mathbf{A}(i-1,j) + d(i,j) \\ \mathbf{D}_\mathbf{A}(i-1,j-1) + 2\,d(i,j) \\ \mathbf{D}_\mathbf{A}(i,j-1) + d(i,j) \end{Bmatrix}$
                 Store the optimal predecessor for the $(i,j)$ point.
             End
         End
Step 3:  Minimum Normalized Total Distance: (3.29)
         $\hat{D}(t,r) = \frac{1}{t+r}\min\begin{bmatrix} \mathbf{D}_\mathbf{A}(t,j)\frac{t+r}{t+j}, & t-M \le j \le r \\ \mathbf{D}_\mathbf{A}(i,r)\frac{t+r}{i+r}, & r-M \le i \le t \end{bmatrix}$    (3.29)
Step 4:  Reconstruct the optimal path, $\hat{\mathbf{F}}$, starting from the point in which $\hat{D}(t,r)$ occurs
         (last path point), and travel backwards as the optimal predecessor indices dictate
         until point $(1,1)$ is reached.
```

Of course, one could use any of the Sakoe-Chiba constraints of Figure 3.3, together with the symmetric function. In all cases, the first path point will be fixed at $(1,1)$, and the $\hat{D}(t,r)$ minimum distance will be given by the Eq (3.29).

3.5 Suggestions

The performance of the various DTW algorithms in the recognition of isolated words was the subject of several investigations (Sakoe and Chiba, 1978, Rabiner et al., 1978, Myers et al., 1980). In all studies, several repetitions (either from the same or different speakers) were used as reference patterns for each word. The measure of performance of each algorithm was the classification error for each word. In most of the cases, symmetric algorithms with constraints on the slope of the optimal path resulted in smaller recognition errors.

In comparing pattern of faults from chemical processes one can use the appropriate DTW algorithm to accommodate the features of a particular problem. For example, in on-line Fault Diagnosis it is reasonable to assume that there will be uncertainty in locating the time origin of a fault. Therefore, the DTW algorithm presented in Subsection 3.4.1 would be the most appropriate one. A symmetric algorithm that allows for consecutive horizontal or vertical transitions in the initial part of both patterns, could also be used. On the other hand, in off-line Fault Diagnosis a symmetric algorithm with a slope constraint is suggested, since both patterns are considered equally important. The simplest algorithm of this type was described in Example 5 of Subsection 3.3.3.

In general, with the exception of the suggestions mentioned above, there are no strict guidelines on which DTW algorithm to use in which situation. This will become clear in the following chapters, where different DTW algorithms are used that take into consideration the particular characteristics of the patterns to be compared.

3.6 Extensions

The algorithms presented in the previous sections have been used in Isolated Word Recognition (IWR) (Myers et al., 1980). This is the easiest Speech Recognition task, since the boundaries of each pattern are either exactly or approximately known.

This nice feature is not present in Connected Word Recognition (CWR) and Continuous Speech Recognition (CSR) (Rabiner and Levinson, 1981, Ney, 1984, Silverman and Morgan, 1990). In CWR, the input is a sequence of words from a specified vocabulary and the recognition is performed by comparing the input signals with reference patterns representing each word in the vocabulary. Even more difficult is

CSR, where the recognition is based on subword units like syllables, phonemes (abstract linguistic units), etc. For both of these problems, extensions of DTW have been constructed but they will not be discussed here.

Finally, stochastic modelling has also been applied in all three problems of Speech Recognition: IWR, CWR and CSR. Hidden Markov Models are parametric stochastic models (in contrast to DTW, which is a deterministic non-parametric method), used to model either isolated words or subword units (Rabiner et al., 1983, Picone, 1990, Juang and Rabiner, 1991). The models are trained on replicates of the reference patterns. During recognition, the input signal is passed through each model; the recognized word is the one whose model is most likely to have generated the input signal.

DTW as applied in IWR is sufficient to perform the classification task in dynamic patterns obtained from chemical processes. There, each pattern corresponds to a fault and it is highly unlikely that two faults will occur immediately one after the other. Moreover, the small number of replicates for each fault (if any) will not allow reliable training of a Hidden Markov Model. Thus, these extensions will not be further discussed.

3.7 Summary and Conclusions

In this chapter the theory of Dynamic Time Warping has been presented as a robust pattern matching method. DTW is able to match similar but possibly unsynchronized patterns, where a linear comparison would provide erroneous results. Different algorithms of DTW have been discussed. The flexibility of the DTW to accommodate various requirements and uncertainties has been emphasized. Also, useful implementation details have been given. This chapter will provide the basis for all the DTW variants used in the rest of the thesis.

CHAPTER 4

OFF-LINE DIAGNOSIS OF DETERMINISTIC FAULTS IN CONTINUOUS DYNAMIC PROCESSES

In this chapter a method will be presented for the off-line diagnosis of deterministic faults in multivariable, dynamic, continuous processes. The method consists of a feature extraction step, where magnitude invariant features are extracted, and a similarity assessment scheme using Dynamic Time Warping. The design parameters, advantages, and limitations of the method are discussed. Case studies from the Tennessee-Eastman plant are used to illustrate its application.

4.1 Introduction

As mentioned in Chapter 1, the term 'deterministic faults' is used in this thesis to describe faults whose different realizations (i.e., either at the same operating point or at different operating point and/or of different magnitude) will produce similar patterns. The off-line diagnosis of this type of faults is the objective of this chapter. The off-line diagnosis of transient upsets can lead to important process or operation modifications which in turn can improve the future behavior of the process. Moreover, it will provide useful insight on how to build the on-line diagnostic scheme (presented in Chapter 7).

The Tennessee-Eastman simulation, described in Subsection 1.5.2, will be used throughout this chapter. The plant schematic, together with the control scheme of McAvoy and Ye (McAvoy and Ye, 1994) is shown in Figure I.1 in Appendix. The process is fed with reactants A, C, D and E; the inert B also enters the process with most of the feed streams; the products are G and H. A set of 20 programmed faults can be simulated; some of them cause major variations in the processes variables, while some are minor disturbances and are easily dealt by the control system. The major faults are either deterministic (e.g., step changes in the composition of a feed stream) or stochastic (e.g., random variation in the kinetic parameters of a reaction). Additionally, these is a set of 5 faults that are of unknown source, cause major variations in the process variables and are of periodic nature.

Of the 20 different faults, 3 were selected as cases studies for this chapter. The selected 3 faults are due to deterministic events and they all cause major variations in the process variables as the control system tries to bring the controlled variables back to their setpoints. All of them are introduced when the simulated plant is at steady state and they require about 30 hours of simulated plant operation before a new steady state is reached. The faults can be introduced with varying magnitude, direction (i.e., steps of positive and negative directions) and at different operating points.

The two operating points used in all the case studies of the thesis are characterized by different production levels and same composition for the final product. It is assumed that the same fault occurring in these operating points will result in similar correlations across variables, but with expanded or contracted temporal correlations. If operating points for different products are considered (i.e., different final product compositions), then this assumption may not be true. It will be assumed that the database contains one realization at the nominal operating point for each of the three faults. The objective will be to off-line classify realizations of the same faults, occurring at both operating points, with different magnitude, direction, and duration.

Also, 2 stochastic faults with small effect on the plant operation (they can be viewed as process noise) will be used in combination with the 3 major faults to test the robustness of the proposed method to the noise level. Finally, one of the unknown periodic upsets will be used to test the performance of the diagnosis scheme when a fault appears which does not exist in the database. Table 4.1 shows the 3 major deterministic, the 2 minor stochastic and the unknown periodic fault considered in this chapter; the notation is the one used in the original paper (Downs and Vogel, 1993).

Table 4.1: Process faults considered in Chapter 4

Fault	Process variable	Type
IDV(1)	A/C feed ratio, B composition constant (stream 4)	Step
IDV(2)	B composition, A/C ratio constant (stream 4)	Step
IDV(7)	C header pressure loss-reduced availability (stream 4)	Step
IDV(9)	D feed temperature (stream 2)	Random variation
IDV(10)	C feed temperature (stream 4)	Random variation
IDV(17)	Unknown	Unknown

Figure 4.1 shows the behavior of 4 of the simulated process variables during the IDV(1), IDV(2) and IDV(7) faults (the initial value of each variable and the scaling are chosen to facilitate the plotting). Let the corresponding patterns be $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$. These will be the known reference patterns in the database; each pattern contains 26 variables and a different number of points. The reference patterns are from faults with a magnitude of one occurring at the nominal operating point. The sampling interval, T_S, is 3 min. The faults are introduced after 3 hours of simulated plant operation (i.e., at the 61st point). A complete description of the reference patterns is given in Table 4.2, Subsection 4.4.1. Table I.1 in Appendix gives the description of all the 26 variables included in the patterns for this chapter. Figures I.2 in Appendix shows 16 of the 26 variables for all three reference patterns.

Figure 4.2 shows the same 4 variables for three unknown test patterns, $\mathbf{T}_2$, $\mathbf{T}_{12}$, and $\mathbf{T}_5$; they are all different realizations of the reference faults. $\mathbf{T}_2$ and $\mathbf{T}_5$ are realizations of faults IDV(1) and IDV(7) respectively, both occurring at the nominal operating point, with step sizes 70% and 80% respectively of the default size.

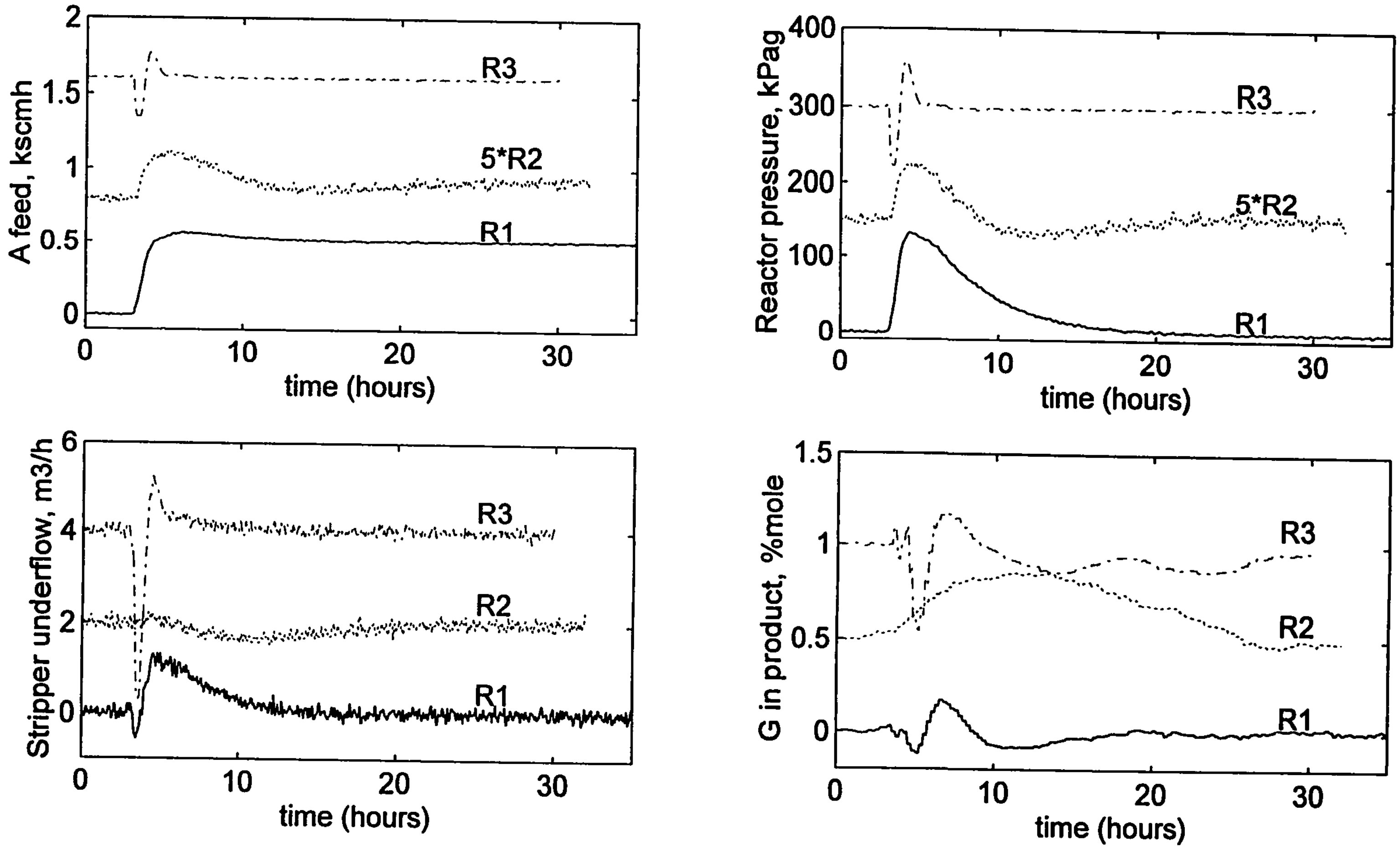


Figure 4.1: Behavior of 4 (out of 26) variables during the reference patterns, $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$; the initial value and the scaling for each variable are chosen to facilitate plotting.

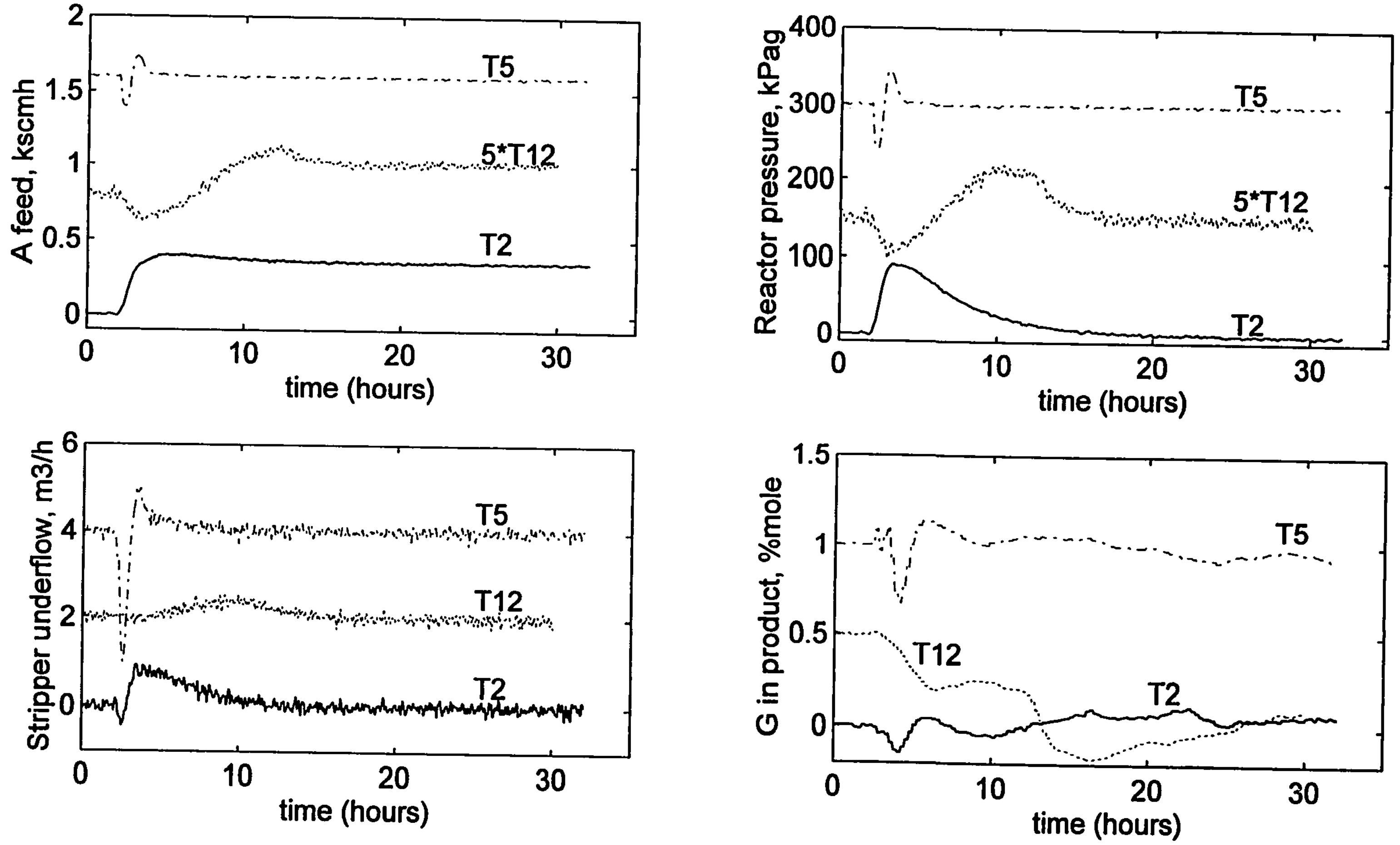


Figure 4.2: Behavior of 4 (out of 26) variables during the test patterns, T_2, T_5 and T_{12}; the initial value and the scaling for each variable are chosen to facilitate plotting.

On the other hand, $\mathbf{T}_{12}$ is the IDV(2) fault occurring at a reduced production operating point, with a negative direction, and with 80% of the default step size. Table 4.3, Subsection 4.4.1, gives the complete description of all the test patterns used in this chapter.

The 4 variables shown in Figures 4.1 and 4.2 illustrate the various problems that have been discussed in Chapter 1 regarding the problems of Fault Diagnosis in continuous processes and the requirements of a realistic diagnostic scheme. Referring to Figure 4.1: the variation of variable '*A feed*' during $\mathbf{R}_1$ is much greater than during $\mathbf{R}_2$; on the other hand, the variation of variable *'G in product'* for $\mathbf{R}_1$ and $\mathbf{R}_2$ is similar. Thus, the absolute variation of variables is not an indication of a fault. This is obvious when the faults occur with different magnitudes. Referring again to Figure 4.1: the shape of variable *'Reactor pressure'* is similar for $\mathbf{R}_1$ and $\mathbf{R}_2$; on the other hand, the variable *'G in product'*, is similar for $\mathbf{R}_1$ and $\mathbf{R}_3$. Thus, looking at individual variables makes it difficult to diagnose the faults. Finally, referring to Figures 4.1 and 4.2: it is not difficult to see that $\mathbf{T}_2$ is most similar to $\mathbf{R}_1$, and $\mathbf{T}_5$ is most similar to $\mathbf{R}_3$, by looking at the evolution of the variables. However, it is not obvious that $\mathbf{T}_{12}$ is most similar to a negative $\mathbf{R}_2$. A realistic Fault Diagnosis scheme has to be able to handle all these features; namely: large number of variables (some with contradictory information), faults with different magnitude and direction, faults occurring at any operating point.

4.2 The Problems of Scaling and Synchronization

As mentioned in the previous section, the diagnostic scheme should be able to classify the patterns independently of their magnitude. Thus, the features that will be extracted from the raw data should be stripped of magnitude information. If the similarity assessment scheme is a distance measure (e.g., via Dynamic Time Warping), step-like signals of different size will erroneously inflate the distance measure. In such a case, the raw data must be scaled to remove the difference on step size.

In Principal Component Analysis, the scaling usually is done for each variable separately by subtracting the average and dividing by the standard deviation. However, if this procedure is applied blindly, it will scale up constant, noisy variables that contain no information. Thus, it has been suggested that if the standard deviation of a variable over the data set is smaller than about four times its measurement error, it should not be included in the analysis (Wold et al., 1987). Another alternative is to scale each variable relative to the others in terms of its relative importance, providing that this process knowledge is available (Kresta et al., 1991).

However, when one deals with deterministic dynamic patterns, the aforementioned scaling procedures cannot be applied. For example, the "average" or the "standard deviation" of a step-type signal depends on how many data points are included before and after the occurrence of the step. Subtracting the estimated "average value" and/or dividing the raw data with the "standard deviation" will render the scaling procedure dependent on the duration of the pattern. This is an undesirable side effect since one of the requirements of the method is independence of the duration of the pattern.

Thus, the nonstationarity in the mean value makes inadequate any scaling procedure that involves only subtraction and division by constant factors. What is required is an operation that will remove this nonstationarity, independent of the duration of the signal. High-pass filtering is such an operation. A properly designed high-pass filter can remove low frequency components like steps or ramps from a signal. Of course, the output of the filter will exhibit a different pattern from the pattern that the raw data exhibit. Consequently, the similarity assessment scheme will work with patterns that are different from those present in the raw data. This is the major disadvantage of this scaling procedure.

After high-pass filtering is applied, then each variable can be normalized to a standard deviation of one. This will remove the effect of the magnitude of the remaining frequency components. This will also remove the effect of the engineering units used to record the variables. This estimation is of course dependent on the duration of the pattern; if a large number of data points corresponding to the steady-state conditions before and after the introduction of a fault are included in the pattern, they will scale down the standard deviation of the variable. Thus, redundant data points should not be included in the pattern. A further step is to apply low-pass filtering to remove high frequency components, present due to measurement noise.

Hence, this scaling procedure (high-pass filtering, normalization to standard deviation of one, low-pass filtering) exhibits some appealing features: it can remove the nonstationary components and remove the magnitude information of the remaining components. However, even after this scaling scheme, the patterns may have different duration and/or may not be synchronized. Moreover, due to varying magnitude of faults and different production levels, the same faults may exhibit different temporal correlations. Dynamic Time Warping (DTW) is a robust pattern matching technique that can effectively deal with all these problems, as it was described in Chapter 3. Thus, after scaling, DTW can be used to assess the similarity between the unknown test pattern and all the reference patterns and classify on the basis of the minimum distance. The next section summarizes the complete procedure (scaling and similarity assessment), presents its advantages and limitations, and discusses the design parameters and their selection.

4.3 A Complete Method

Let $\mathbf{R}_i$, $i = 1,\ldots,I$ be a the set of reference patterns, each representing a known fault. Each is a matrix of dimenion $r_i \times N$, where r_i is the number of observations, and N is the number of measured variables. Also, let $\mathbf{T}$ be an unknown test pattern, a matrix of dimension $t \times N$; the objective is to find which of the $\mathbf{R}_i$ patterns is most similar to $\mathbf{T}$. The proposed method is as follows:

Feature Extraction Steps: for each variable, in each $\mathbf{R}_i$ do the following:

Step 1: Subtract the initial level.

Step 2: Filter with high-pass filter.

Step 3: Normalize to standard deviation of one.

Step 4: Filter with low-pass filter.

Let $\mathbf{R}_{i,sc}$ be the reference patterns after the above scaling procedure.

When the test pattern $\mathbf{T}$ becomes available, apply Steps 1-4; let $\mathbf{T}_{sc}$ be the resulting scaled pattern.

Similarity Assessment Steps.

Step 5: Apply Dynamic Time Warping between $\mathbf{R}_{i,sc}$ and $\mathbf{T}_{sc}$.

Step 6: Apply Dynamic Time Warping between $-\mathbf{R}_{i,sc}$ and $\mathbf{T}_{sc}$ (i.e., the mirror images of the scaled reference patterns).

Step 7: From the $2 \cdot I$ distances obtained in Steps 5 and 6, find the minimum. The fault whose pattern results in the minimum distance is deemed to be the most likely to have generated the test pattern.

Steps 2 and 3 work towards making the diagnosis independent of the magnitude of the fault; they also handle the problem of different engineering units. Step 4 effectively downweights the noisy variables in the DTW step because their standard deviation will be significantly less than one after low-pass filtering. Since DTW is magnitude sensitive, it will concentrate on variables whose standard deviation is closest to one; i.e., the less noisy variables.

Step 1 together with the ability of DTW to match unsynrconized patterns and patterns with different temporal correlations work towards independence of the diagnosis from the production level and independence of the uncertainty in locating the edge points of the patterns. Step 6 tries to diagnose faults with a negative direction compared to the faults in the reference set. If the process was linear, this would not create any concern.

However, almost all chemical processes are nonlinear and therefore this step is questionable and its reliability depends on the degree of the process nonlinearity.

The method is multivariate in the sense that the optimal match between two patterns (as found by DTW) is the same for all variables. The method applies without any modification for any number of measured variables; the memory requirements on the DTW step are not affected by their number. Moreover, one can weight differently variables that are more important than others in the diagnosis of a fault by specifying an appropriate weight matrix, **W**, for the local distance computations.

If several realizations of each fault were available, one could construct reference distributions of the local and total distances and find appropriate confidence intervals. However, in this study it is assumed that only one realization of each fault exist in the database, and therefore the reference distribution approach is not possible. Furthermore, no assumptions are made about statistical distributions of the process variables since for the class of deterministic faults, the variables are nonstationary deterministic signals. Therefore, the distances computed in Steps 5 and 6 have no absolute meaning; they are only relative similarity measures.

The design parameters of the method are: the location of the edge points of the patterns, the type of high-pass and low-pass filters and their cut-off frequencies, the local distance, the weight matrix **W**, and finally the DTW algorithm. Locating the first and the last point for each pattern is not a big problem in an off-line analysis. One can use the Sakoe-Chiba local constraints of Figure 3.3(b) (i.e., with no constraint on the slope of the optimal path) that allow consecutive horizontal or vertical transitions to partially compensate for uncertainty in locating the two points. Even better, one can use the modifications presented in Section 3.4 to relax the fixed-endpoint constraints in the DTW algorithm. To avoid searching in unlikely regions for the optimal path and to reduce computations, the band global constraint of Figure 3.4(b) can be used.

Regarding the local distance, the simplest choice is to use the Euclidean distance (i.e., $\mathbf{W}$ being the identity matrix), unless knowledge of the importance of certain variables is available. Note that the same $\mathbf{W}$ has to be used for all $\mathbf{R}_{i,sc} - \mathbf{T}$ pairs in DTW, otherwise the comparison of distances in Step 7 will not be consistent.

The type of DTW algorithm is not very crucial. An asymmetric DTW algorithm will treat preferentially one pattern over the other; if this is not desirable, a symmetric DTW algorithm has to be used. The relaxation of the fixed-endpoint constraints is also a factor; as shown in Section 3.4, with a symmetric algorithm it is not possible to relax both of them. The studies of Sakoe and Chiba (1978), Rabiner et al. (1978) and Myers et al. (1980) showed small differences in the performance when different local constraints are used.

The most important parameter of the method is the type of the high-pass filter. High order filters are characterized by sharp frequency responses; however, they induce oscillations in the signals and, in general, they heavily modify the patterns. On the other hand, a simple first order high-pass filter does not have a sharp frequency response, but produces smoother output signals. Since the objective is only to remove the very low frequency components (e.g., steps, ramps) with the least distortion of the other components, the first order filter is better suited. Selection of the low-pass filter is not very crucial; any standard filter design can be used. The estimated power spectra of the variables can provide guidelines for the cut-off frequency of the filters.

This concludes the presentation of an off-line Pattern Recognition scheme proposed to diagnose deterministic faults in dynamic multivariable continuous processes. It tries to address the requirements for a realistic scheme presented in Subsection 1.5.1, i.e., diagnose faults independently of their magnitude, plant operating point and direction. The next section presents the case studies performed to test the proposed scheme using the Tennessee-Eastman plant simulation.

4.4 Case Studies

4.4.1 Description of the Patterns in the Reference and Test Sets

As mentioned in Section 4.1, the reference set considered in this chapter had three patterns, $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$ that correspond to three major deterministic upsets: IDV(1), IDV(2) and IDV(7). Tables 4.2 and 4.3 describe the details of the reference and the test set patterns.

Table 4.2: Patterns in the Reference Set

Pattern	Fault	Operating Point	Step size / direction	Duration / # of points	Fault occurs after / at
$\mathbf{R}_1$	IDV(1)	Nominal	+1.0	35 hrs / 701 pts	3 hrs / 61st pt
$\mathbf{R}_2$	IDV(2)	Nominal	+1.0	32 hrs / 641 pts	3 hrs / 61st pt
$\mathbf{R}_3$	IDV(7)	Nominal	+1.0	30 hrs / 601 pts	3 hrs / 61st pt

The nominal operating point is the base case of "Operating Mode 1", as described in the paper by Downs and Vogel (1993). The "Reduced Production" operating point is obtained from the nominal operating point by reducing the setpoint of the product flow rate by 15% using the control scheme of McAvoy and Ye (1994); thus, the plant operates at a production level which is 85% of the nominal production level.

The faults presented in Tables 4.2 and 4.3 are selected so that the proposed method can be tested at conditions of varying difficulty. Some test patterns are almost identical to one of the reference set patterns, while others are quite different (e.g., they occur with smaller and/or negative magnitude, at the "Reduced Production" operating point). Also, test cases $\mathbf{T}_7$ and $\mathbf{T}_8$ will investigate the role of increased noise level in the Fault Diagnosis, while case $\mathbf{T}_{16}$ investigates the performance of the method when a fault appears which is not included in the reference set.

Table 4.3: Test Set Patterns Evaluated

Pattern	Fault	Operating Point	Step size / direction	Duration / # of points	Fault occurs after / at
T_{1}	IDV(1)	Nominal	+1.0	35 hrs / 701 pts	2 hrs / 41st pt
T_{2}	IDV(1)	Nominal	+0.7	32 hrs / 641 pts	2 hrs / 41st pt
T_{3}	IDV(2)	Nominal	+0.8	32 hrs / 641 pts	2 hrs / 41st pt
T_{4}	IDV(2)	Nominal	-0.9	35 hrs / 701 pts	2 hrs / 41st pt
T_{5}	IDV(7)	Nominal	+0.8	32 hrs / 641 pts	2 hrs / 41st pt
T_{6}	IDV(7)	Nominal	-0.7	30 hrs / 601 pts	2 hrs / 41st pt

T_{7}	IDV(1)
	IDV(9)
		Nominal	+0.9
			+1.0
				35 hrs / 701 pts	2 hrs / 41st pt
					10 hrs / 201st pt

T_{8}	IDV(2)
	IDV(10)
		Nominal	+0.8
			+1.0
				35 hrs / 701 pts	2 hrs / 41st pt
					10 hrs / 201st pt

T_{9}	IDV(1)	Reduc. Prod.	+0.5	32 hrs /641 pts	2 hrs / 41st pt
T_{10}	IDV(1)	Reduc. Prod.	+0.9	35 hrs /701 pts	2 hrs / 41st pt
T_{11}	IDV(2)	Reduc. Prod.	+0.9	32 hrs /641 pts	2 hrs / 41st pt
T_{12}	IDV(2)	Reduc. Prod.	-0.8	30 hrs /601 pts	2 hrs / 41st pt
T_{13}	IDV(7)	Reduc. Prod.	+0.7	28 hrs /561 pts	2 hrs / 41st pt
T_{14}	IDV(1)	Nominal	-0.5	35 hrs /701 pts	2 hrs / 41st pt
T_{15}	IDV(2)	Nominal	-0.5	32 hrs / 641 pts	2 hrs / 61st pt
T_{16}	IDV(17)	Nominal	+1.0	25 hrs /501 pts	2 hrs / 41st pt

4.4.2 Selection of Design Parameters

The duration of the patterns is between 28 and 32 hrs; this is the suggested duration to fully see the effect of a fault in the plant (Downs and Vogel, 1993). The small differences in the duration were selected on purpose to demonstrate the ability of the method to work with patterns of unequal duration. Also, the faults in the patterns of the reference and of the test set were introduced 3 hrs and 2 hrs, respectively, after time zero. This was done to demonstrate that the method can work with unsyncronized patterns.

For the DTW algorithm, the Sakoe-Chiba symmetric local constraints with the symmetric weighting function, both shown in Figure 3.5(a), were selected; they were used in conjunction with the band global constraint of Figure 3.4(b). Also the fixed-endpoint constraints were used. Thus, the DTW algorithm used for the off-line diagnosis of deterministic faults was exactly the one described in Example 1, Section 3.3.1. The local distance was the Euclidean distance, with **W** being the identity matrix, since it is assumed that all 26 variables are equally important in the diagnosis.

The band parameter M (which defines the width of the band, $2 \cdot \mathrm{M}$) was selected so that the band constraint would be consistent with the fixed-endpoint constraints, it has to be $\mathrm{M} \geq |\mathrm{r}_\mathrm{i} - \mathrm{t}|$. Thus, for any $\mathbf{R}_\mathrm{i} - \mathrm{T}$ pair, M was set as:

$$\mathrm{M} = |\mathrm{r}_\mathrm{i} - \mathrm{t}| + 50 \tag{4.1}$$

Simple first order high-pass and low-pass filters were used. Figure 4.3 shows the power spectra of the same variables shown in Figures 4.1 and 4.2, for the reference pattern $\mathbf{R}_1$. They were computed using the `spectrum` command in Matlab, Signal Processing Toolbox (Matlab, 1988) using a window of 256 points, which implements the "Welsh method" of power spectrum estimation (Oppenheim and Schafer, 1975).

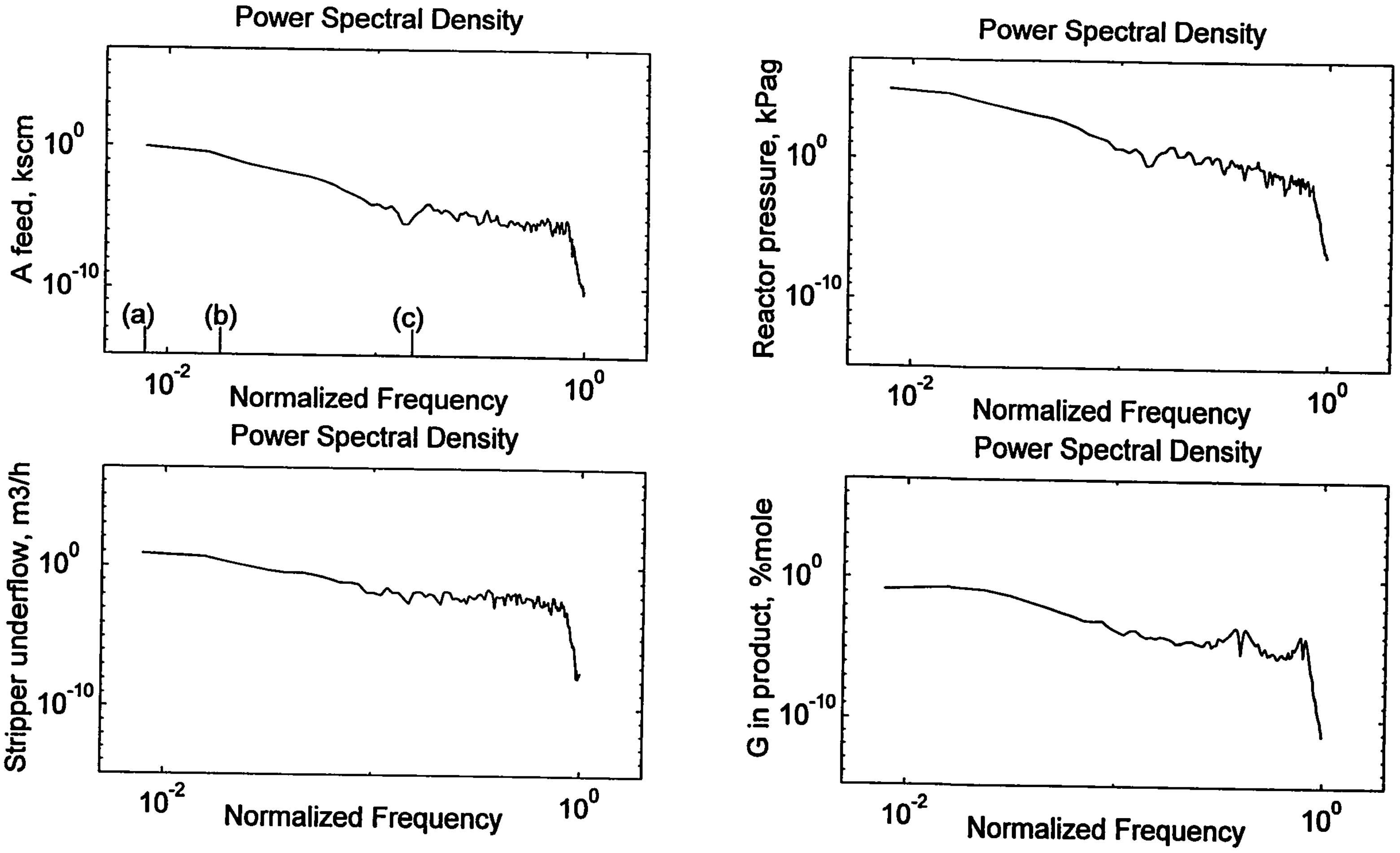


Figure 4.3: Power spectral density for the 4 variables of the pattern $\mathbf{R}_{1}$; power vs. normalized frequency; (a) normalized minimum frequency , $\hat{f}_{S} / 256$ (where $\hat{f}_{S} = 2$ is the normalized sampling frequency and 256 is the length of the window used in spectrum estimation); (b) normalized cut-off frequency for the high-pass filter, $\hat{f}_{HP} = 0.018$; (c) normalized cut-off frequency for the low-pass filter, $\hat{f}_{LP} = 0.150$.

The normalized cut-off frequencies[1] with respect to half of the sampling frequency, $f_N / 2$, for the high-pass and the low-pass filters, $\hat{f}_{HP}$ and $\hat{f}_{LP}$ respectively, were selected to be:

$$\hat{f}_{HP} = 0.018 \tag{4.1a}$$

$$\hat{f}_{LP} = 0.150 \tag{4.1b}$$

Thus, the unnormalized cut-off frequencies for the filters are:

$$f_{HP} = \hat{f}_{HP} \cdot \frac{f_S}{2} = 0.018 \cdot \frac{1}{2 \cdot 180} = 5.00 \cdot 10^{-5}\,\mathrm{Hz} \tag{4.2a}$$

$$f_{LP} = \hat{f}_{LP} \cdot \frac{f_S}{2} = 0.150 \cdot \frac{1}{2 \cdot 180} = 4.17 \cdot 10^{-4}\,\mathrm{Hz} \tag{4.2b}$$

(f_S is the sampling frequency, 1/180 Hz, since the sampling interval T_S is 180 sec).

The continuous transfer functions for the first order filters with these cut-off frequencies are:

$$G_{HP}(s) = \frac{s}{s + 2 \cdot \pi \cdot f_{HP}} \tag{4.3a}$$

$$G_{LP}(s) = \frac{1}{\frac{1}{2 \cdot \pi \cdot f_{LP}} s + 1} \tag{4.3b}$$

[1] The cut-off frequency is defined as the frequency where the amplitude ratio gets the value of $1 / \sqrt{2}$.

Discretizing the above transfer functions for the given sampling interval using the method of prewarping (Franklin and Powell, 1980) and selecting the critical frequency for each filter its cut-off frequency[2], the discrete transfer functions of the filters are:

$$H_{HP}(z^{-1}) = \frac{0.9725 - 0.9725z^{-1}}{1 - 0.9450z^{-1}} \tag{4.4a}$$

$$H_{LP}(z^{-1}) = \frac{0.1936 - 0.1936z^{-1}}{1 - 0.6128z^{-1}} \tag{4.4b}$$

The effect of the scaling procedure (i.e., Steps 1-4) in the process variables is illustrated in Figure 4.4, where the same 4 variables for all reference patterns, $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$, are shown after scaling (the real origin of all scaled variables is zero since the initial value has been subtracted in Step 1; the origins shown in Figure 4.4 are selected to facilitate plotting). Compare variable *'A feed'* in Figures 4.1 and 4.4: the step-like pattern that the variable exhibits originally during $\mathbf{R}_1$ has been filtered out by the high-pass filter; $\mathbf{R}_{1,SC}$ and $\mathbf{R}_{2,SC}$ are now similar in that variable. This is the major disadvantage of the scaling procedure: high-pass filtering removes a lot of steady-state information. One the other hand, the same variable does not appear differently in $\mathbf{R}_{3,SC}$ since no low frequency components are present. Moreover, the range of the variable now is similar in all scaled patterns, due to normalization to standard deviation of one.

4.4.3 Results

The results of applying the Steps 5, 6 and 7 are presented in Table 4.4. For each test pattern of Table 4.3, the scaling procedure is applied and then is matched via DTW with

[2] According to the method of prewarping, the continuous and the discrete filters have the same frequency response at the critical frequency (Franklin and Powell, 1980).

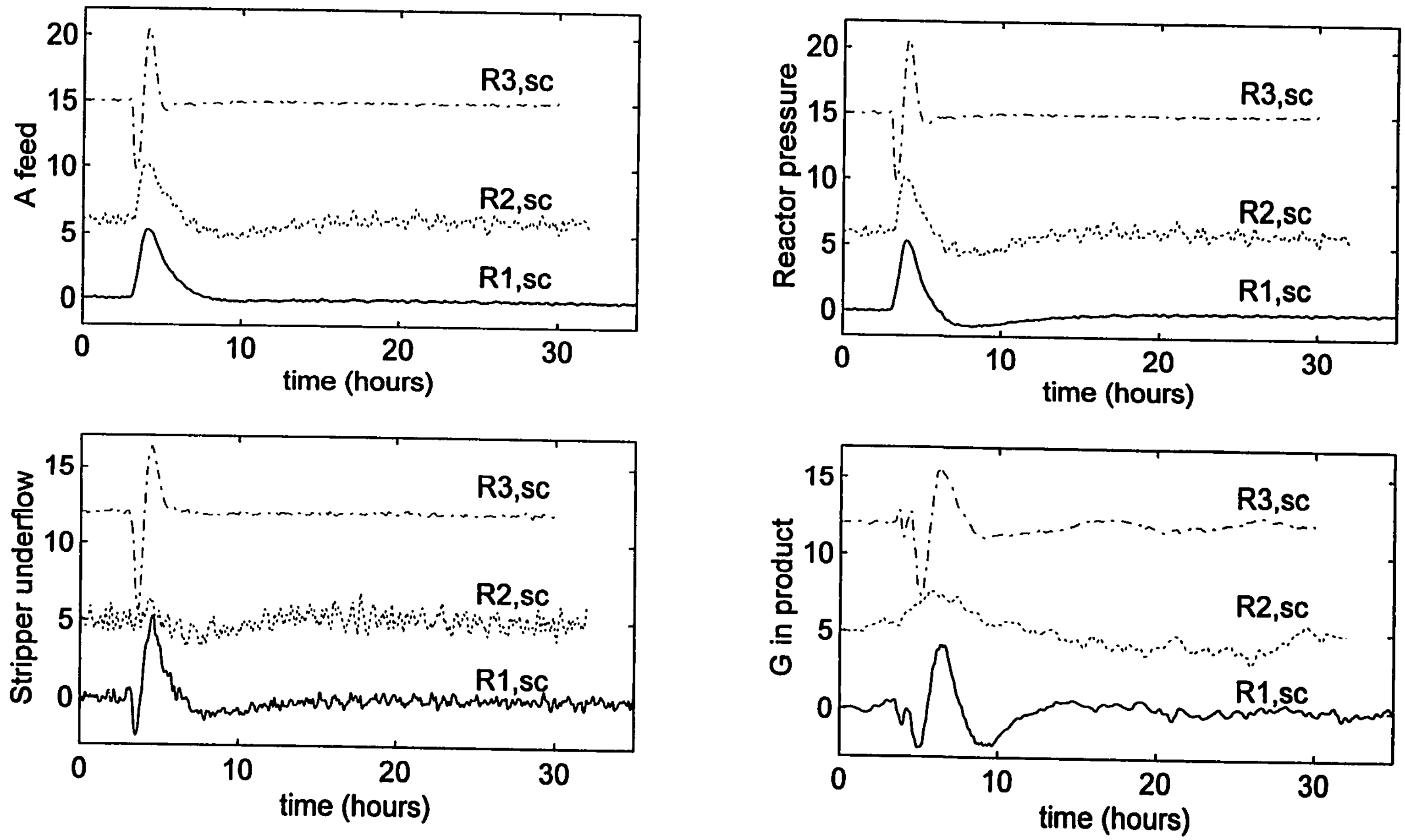


Figure 4.4: Behavior of the 4 variables during the scaled reference patterns, $\mathbf{R}_{1,sc}$, $\mathbf{R}_{2,sc}$ and $\mathbf{R}_{3,sc}$; the initial value for each variable is chosen to facilitate plotting.

all scaled reference patterns and their mirror images. Six minimum normalized total distances are then obtained for each $\mathbf{T}_{i,sc}$:

$$\hat{D}(\mathbf{T}_{i,sc},\mathbf{R}_{1,sc}),\ \hat{D}(\mathbf{T}_{i,sc},-\mathbf{R}_{1,sc})$$

$$\hat{D}(\mathbf{T}_{i,sc},\mathbf{R}_{2,sc}),\ \hat{D}(\mathbf{T}_{i,sc},-\mathbf{R}_{2,sc})$$

$$\hat{D}(\mathbf{T}_{i,sc},\mathbf{R}_{3,sc}),\ \hat{D}(\mathbf{T}_{i,sc},-\mathbf{R}_{3,sc})$$

The fault that corresponds to the reference pattern giving the minimum distance, $\hat{D}_{i,MIN}$, is selected as the most probable cause for the test pattern. Table 4.4 shows $\hat{D}_{i,MIN}$, the ratio of the other six distances over $\hat{D}_{i,MIN}$ and the decision on the most probable cause for each test pattern. The shaded cells indicate the correct diagnosis; hence, when the ratio of one appears in a shaded cell then the diagnosis is correct. Good discrimination among faults is obtained when the five of the six ratios are not close to one (i.e., $\hat{D}_{i,MIN}$ is much less than the other five distances). Figures 4.5 and 4.6 show the DTW results for two test patterns, $\mathbf{T}_2$ and $\mathbf{T}_{12}$. The plots in both figures show the band constraint, the optimal path and the minimum normalized total distance.

Comparing Tables 4.3 and 4.4, one can observe the following:

A) With the exception of pattern $\mathbf{T}_{16}$, all other diagnoses are correct; $\hat{D}_{i,MIN}$ pointed to the correct fault and direction in all cases.

B) The discrimination between the various faults is better when the test pattern is identical to one of the reference patterns and worsens as the test pattern is a fault of different magnitude and/or direction, occurring at the "Reduced Production" operating point (see results for patterns $\mathbf{T}_{1,sc}$, $\mathbf{T}_{2,sc}$, $\mathbf{T}_{9,sc}$ and $\mathbf{T}_{14,sc}$). This is expected since an identical realization of a fault will result in similar patterns and magnitudes for the process variables.

Table 4.4: Results from Similarity Assessment via DTW; Distances from each DTW match over Minimum Distance and Final Decision.

Test Pattern	$\hat{D}_{i,MIN}$	$\hat{D}(T_{i,SC}, R_{1,SC})/\hat{D}_{i,MIN}$	$\hat{D}(T_{i,SC}, -R_{1,SC})/\hat{D}_{i,MIN}$	$\hat{D}(T_{i,SC}, R_{2,SC})/\hat{D}_{i,MIN}$	$\hat{D}(T_{i,SC}, -R_{2,SC})/\hat{D}_{i,MIN}$	$\hat{D}(T_{i,SC}, R_{3,SC})/\hat{D}_{i,MIN}$	$\hat{D}(T_{i,SC}, -R_{3,SC})/\hat{D}_{i,MIN}$	Decision
$T_{1,SC}$	1.77	1.00	18.10	11.08	9.99	8.18	8.80	IDV(1)
$T_{2,SC}$	2.37	1.00	10.54	7.93	12.15	6.32	7.20	IDV(1)
$T_{3,SC}$	4.02	4.77	4.17	1.00	10.79	5.27	5.37	IDV(2)
$T_{4,SC}$	6.32	2.76	3.63	4.67	1.00	3.22	3.21	-IDV(2)
$T_{5,SC}$	0.93	15.46	17.57	24.81	26.17	1.00	20.64	IDV(7)
$T_{6,SC}$	1.26	12.77	11.81	16.72	17.01	15.69	1.00	-IDV(7)
$T_{7,SC}$	1.94	1.00	16.31	9.67	9.31	7.54	8.46	IDV(1)
$T_{8,SC}$	7.51	2.76	2.71	1.00	3.98	2.70	2.56	IDV(2)
$T_{9,SC}$	10.25	1.00	2.48	2.20	2.77	1.80	2.23	IDV(1)
$T_{10,SC}$	10.08	1.00	3.61	2.19	2.27	1.77	2.13	IDV(1)
$T_{11,SC}$	11.73	1.70	1.60	1.00	3.59	1.94	1.94	IDV(2)
$T_{12,SC}$	14.27	1.22	1.49	2.21	1.00	1.80	1.74	-IDV(2)
$T_{13,SC}$	12.30	1.74	1.66	1.79	1.95	1.00	1.59	IDV(7)
$T_{14,SC}$	4.97	6.39	1.00	2.79	4.17	3.54	3.13	-IDV(1)
$T_{15,SC}$	5.65	2.98	3.17	7.30	1.00	3.51	3.63	-IDV(2)
$T_{16,SC}$	18.26	1.00	1.15	1.15	1.00	1.13	1.17	-IDV(2)

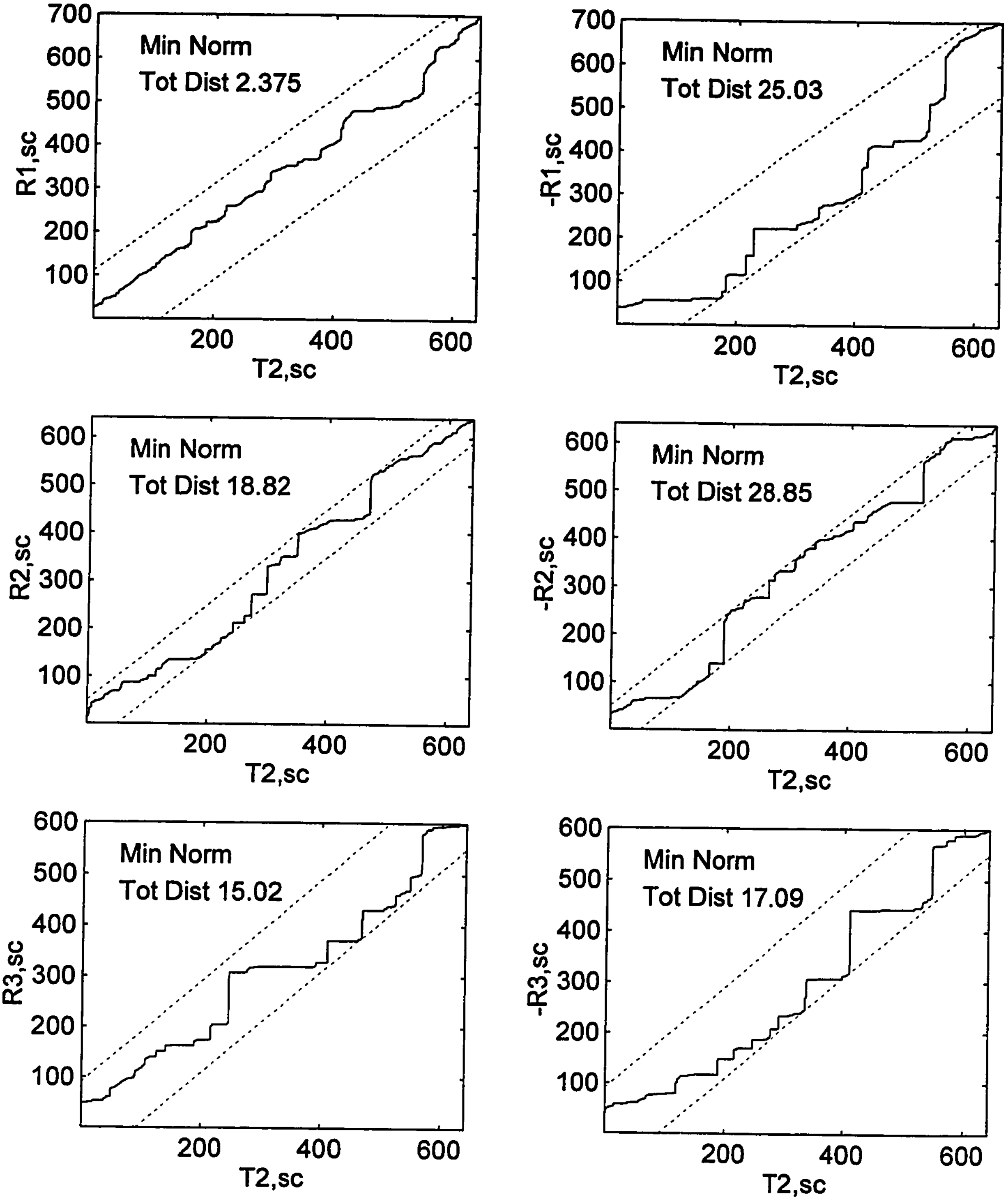


Figure 4.5: Results from DTW for test pattern $\mathbf{T}_2$; global constraints, optimal path and minimum normalized total distance between $\mathbf{T}_{2,sc}$ and $\mathbf{R}_{1,sc}$, $-\mathbf{R}_{1,sc}$, $\mathbf{R}_{2,sc}$, $-\mathbf{R}_{2,sc}$, $\mathbf{R}_{3,sc}$, $-\mathbf{R}_{3,sc}$.

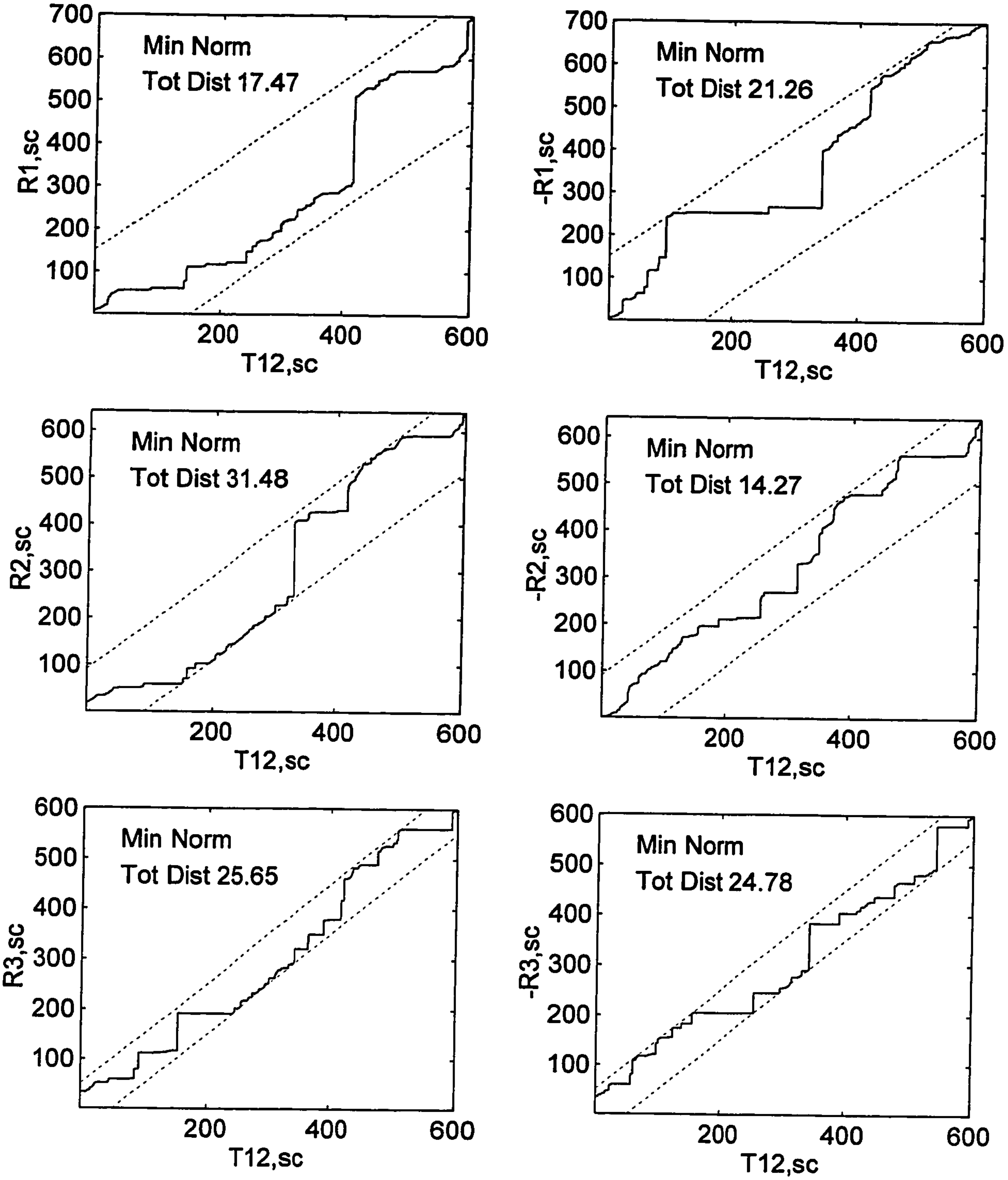


Figure 4.6: Results from DTW for test pattern $\mathbf{T}_{12}$; global constraints, optimal path and minimum normalized total distance between $\mathbf{T}_{12,sc}$ and $\mathbf{R}_{1,sc}$, $-\mathbf{R}_{1,sc}$, $\mathbf{R}_{2,sc}$, $-\mathbf{R}_{2,sc}$, $\mathbf{R}_{3,sc}$, $-\mathbf{R}_{3,sc}$.

C) The process operating point is the most important factor in the diagnosis. As the results in Table 4.4 show, when the faults occur at the "Reduced Production" operating points ($\mathbf{T}_9$ to $\mathbf{T}_{13}$) the discrimination among the correct fault and all the others is more difficult than when the faults occur at the Nominal operating point.

D) The discrimination is more difficult when the patterns are noisy. $\mathbf{R}_2$ pattern contains more noisy signals than $\mathbf{R}_1$ or $\mathbf{R}_3$, as Figure 4.1 shows. As a result, diagnosing that $\mathbf{T}_{3,SC}$ is most similar to $\mathbf{R}_{2,SC}$ is a weaker statement than diagnosing that $\mathbf{T}_{2,SC}$ is most similar to $\mathbf{R}_{1,SC}$ or that $\mathbf{T}_{5,SC}$ is most similar to $\mathbf{R}_{3,SC}$.

E) As already mentioned in Chapter 1, the signal to noise ratio for a particular pattern is a very important factor for the certainty of the classification. As expected, noisy patterns or patterns of faults of small magnitude, are classified with less certainty than patterns of faults of larger magnitude. For all the faults studied in this chapter, there will be a range of magnitudes in which the variables' variation is comparable to the common process variation; even if these faults are detected, their diagnosis will be poor.

F) Because a minimum will always exist among the six distances, a decision will be taken on which is the most probable cause of a fault. This will lead to an incorrect diagnosis for a fault that does not exist in the reference set, as the $\mathbf{T}_{16}$ pattern shows (i.e., diagnosed as -IDV(2)). This is a disadvantage of the proposed method. One the other hand, the remaining five ratios will be close to one, indicating that the other five distances are not much larger than the minimum distance. This could be a warning on the power of the diagnosis. In practice, once a check is carried out by the plant personnel and the correct cause of the fault is identified, the $\mathbf{T}_{16}$ pattern will be included in the reference set as a new pattern for future diagnoses.

4.6 Summary and Conclusions

In this chapter, a method has been proposed for the off-line diagnosis of faults in dynamic continuous multivariable processes. A Pattern Recognition approach has been followed; thus no process knowledge or process model is required. The proposed method has been designed so that it can classify process faults independently of their magnitude, direction, time origin of the fault and the production level. It consists of a feature extraction step, where the magnitude information is removed from the patterns, followed by a similarity assessment step, where Dynamic Time Warping is used for pattern comparison. The method can handle multivariate patterns of any number of variables and no assumptions about statistical distributions are required.

There are drawbacks to this procedure, however. All variables, noisy or not, are considered equally important on the decision process (i.e., they are all weighted equally); this issue will be addressed in Chapter 6. Also, the fact that the distance measures are relative and not absolute, will produce an erroneous diagnosis when a new fault appears that is not included in the database of past faults. Special precautions have to be constructed for such cases.

The most important feature of the method it that it relies on patterns and correlations that appear in the process variables. Faults that are fundamentally similar but produce different patterns in time (e.g., a step-type and a ramp-type change in the feed composition) will be treated as different faults. This can be viewed both as negative and positive feature. If only a step-type fault is available in the database, then a ramp-type fault will be considered as a different fault. However, a better diagnosis will result if both descriptions are available in the database, particularly in cases where there is a different physical cause behind each fault, e.g., a fast catalyst poisoning versus a slow catalyst fouling process. Also, similar faults that produce different patterns depending on the operating point will also be treated as different faults. In all cases, process knowledge and detailed examination of the historical data are imperative for the successful implementation of the method.

CHAPTER 5

DIAGNOSIS OF STOCHASTIC FAULTS

IN CONTINUOUS DYNAMIC PROCESSES

In this chapter a method will be presented for the off-line and on-line diagnosis of stochastic faults in multivariable, dynamic, continuous processes. The method consists of a feature extraction step where autocorrelation and crosscorrelation coefficients are the extracted features, followed by a similarity assessment scheme via Dynamic Time Warping. The Tennessee-Eastman plant is used to illustrate the advantages, disadvantages and the implementation details of the method.

5.1 Introduction

The term 'stochastic faults' is used in this thesis to indicate faults whose underlying source is a stochastic process. As such, different realizations of the same fault will result in different patterns in the process variables. Therefore, a pattern matching method based on the similarity of scaled patterns of the process variables (as the method described in the previous chapter) cannot be used. Features have to be extracted that remain constant over different realizations of the same stochastic fault and a pattern matching procedure can then be applied in this consistent feature space.

Again, the Tennessee-Eastman simulation with the control scheme of McAvoy and Ye (McAvoy and Ye, 1994) will used in this chapter (see Figure I.1 in Appendix). Two major stochastic faults can be introduced into this simulation; both of them cause major variations in the process variables. The control system tries to bring the controlled variables back to their setpoints, however no steady-state conditions can be achieved. The faults are introduced when the plant is at steady state, either at the nominal operating point or at the "Reduced Production" operating point (as they have been described in Subsection 4.4.1).

It will be assumed that one realization for each of the two faults exists in the database. The objective will be to off-line classify new realizations of the same faults. In addition to the variability caused by the random nature of a stochastic fault, the new realizations may have different magnitude and may occur at different production levels. Since the faults are stochastic, the concept of 'direction' does not apply (in contrast to a deterministic negative or positive step change in a process variable). Table 5.1 shows the faults considered in this chapter; again, the notation is the one used in the original Tennessee-Eastman paper (Downs and Vogel, 1993).

Table 5.1: Process faults considered in Chapter 5

Fault	Process variable	Type
IDV(8)	A, B, C feed composition (stream 4)	Random variation
IDV(13)	Reaction kinetics	Slow drift

Figure 5.1 shows the behavior of 4 process variables during the $\mathbf{RS}_1$, $\mathbf{RS}_2$, $\mathbf{TS}_1$ and $\mathbf{TS}_{19}$ patterns. $\mathbf{RS}_1$ and $\mathbf{RS}_2$ will be the reference patterns, while $\mathbf{TS}_1$ and $\mathbf{TS}_{19}$ are two of the test patterns (for a complete description and notation, see Tables 5.2 and 5.3, Subsection 5.5.1). $\mathbf{RS}_1$ and $\mathbf{RS}_2$ are realizations of the IDV(8) and IDV(13) faults, respectively, both occurring at the nominal operating point with the default magnitude of one. $\mathbf{TS}_1$ is also a realization of the IDV(8) fault, again occurring at the nominal operating point with magnitude of one. $\mathbf{TS}_{19}$ is a realization of the IDV(13) fault, occurring at the "Reduced Production" operating point with magnitude 80% of the default magnitude.

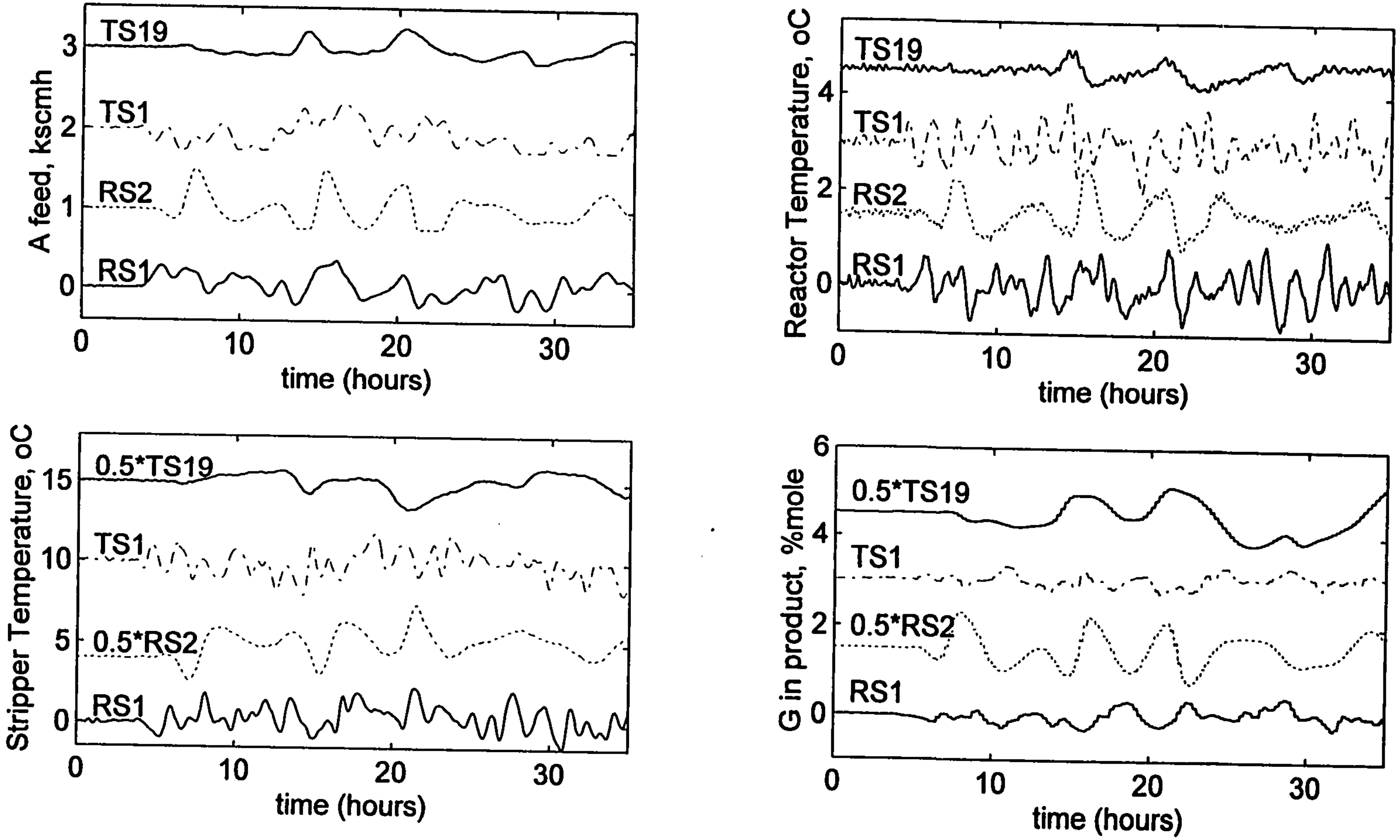


Figure 5.1: Behavior of 4 (out of 8 variables) during the reference patterns $\mathbf{RS}_1$, and $\mathbf{RS}_2$ and the test patterns $\mathbf{TS}_1$ and $\mathbf{TS}_{19}$; the initial value and the scaling for each variable are chosen to facilitate plotting.

Comparing the patterns produced by $\mathbf{RS}_1$ and $\mathbf{TS}_1$, one can see that the patterns are not similar, although both are realizations of the same fault. The same argument applies for $\mathbf{RS}_1$ and $\mathbf{TS}_{19}$ patterns. Thus, the time evolution of the variables is not a feature that can be used to classify these faults. One the other hand, patterns $\mathbf{RS}_1$ and $\mathbf{TS}_1$ are more oscillatory and contain higher frequency components than the other patterns. This could be a feature that can partially discriminate the two faults. Furthermore, the relative behavior between variables can be used as another discriminatory feature; e.g., variables *'A feed'* and *'G in product'* exhibit a strong positive correlation during $\mathbf{RS}_2$ and negative correlation during $\mathbf{RS}_1$. Thus, a feature that reflects this behavior will be useful in discriminating between the two faults.

5.2 Feature Extraction

As mentioned in the previous section, to correctly classify stochastic faults features have to be extracted that: I) reflect the process dynamics and the relative behavior of the variables; and II), remain constant with different realizations of the same fault. For stochastic processes, the autocorrelation and crosscorrelation functions of the process variables satisfy these requirements. Two different realizations of a stochastic fault should be characterized by the same correlation pattern among the variables since the physical and chemical mechanisms are the same in both cases. Thus, estimates of the autocorrelation and crosscorrelation function of the process variables can be used to represent a stochastic fault. One can then have a database of correlation patterns, each representing a past known stochastic fault. When a new unknown fault appears, its correlation pattern is estimated and compared with the database patterns. A distance-based method (either linear or based on Dynamic Time Warping (DTW)) can then be used to carry out the comparison. One can then diagnose the cause of the new fault on the basis of minimum distance.

The advantage of the above method is that the auto- and crosscorrelation functions are both scaling independent. This satisfies the requirement for independence of the Fault Diagnosis procedure on the magnitude of the fault. On the other hand, if N is the number of measured process variables, one has to estimate N autocorrelations and $\frac{(N-1)N}{2}$ crosscorrelations. This means that $\frac{(N+1)N}{2}$ correlations should be considered for completeness. Thus, there is a large increase on the dimension of the space where the subsequent pattern comparison will take place. Moreover, not all of the N variables carry useful information; some of them may not be affected by the fault, yet their autocorrelation and crosscorrelation estimates will be included in the correlation pattern. This issue will be addressed at Chapter 6.

If the auto- and crosscorrelation function is estimated for the same number of lags for all faults, then the correlation patterns will have the same number of data points and the summation of Euclidean distances, along the linear path, is a possible distance measure. However, depending on the production level, the process responds faster or slower to disturbances and consequently the correlation pattern is affected. Thus, a flexible pattern matching method like DTW is a better alternative. DTW can appropriately expand, contract or translate the correlation patterns to take account of faster or slower process dynamics. For that reason a modified DTW algorithm will be used to assess the similarity between correlation patterns. The next section presents the proposed method and its design parameters.

5.3 A Complete Method

5.3.1 The Correlation Pattern

Let **RS** be a $r \times N$ matrix which contains r data points of an N-variate stochastic time series. The correlation pattern of **RS** is defined as another matrix that has the following structure: its first N columns contain the estimates of the autocorrelation function; the following N-1 columns contain the estimates of the crosscorrelation function between the first variable and each subsequent variable; the following N-2 columns contain the estimates of the crosscorrelation function between the second variable and each subsequent variable, etc. For example, let $\mathrm{N} = 3$ and let $\left[\mathrm{x}_i \;\; \mathrm{y}_i \;\; \mathrm{z}_i\right]$, $\mathrm{i} = 1,\ldots,\mathrm{r}$ be a trivariate time series consisting of measurements of the random variables $\{x_i\}$, $\{y_i\}$, and $\{z_i\}$. The matrix **RS** is then:

$$\mathbf{RS} = \begin{bmatrix} \mathrm{x}_1 & \mathrm{y}_1 & \mathrm{z}_1 \\ \mathrm{x}_2 & \mathrm{y}_2 & \mathrm{z}_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \mathrm{x}_\mathrm{r} & \mathrm{y}_\mathrm{r} & \mathrm{z}_\mathrm{r} \end{bmatrix} \qquad (5.1)$$

The correlation pattern of **RS, Corr[RS],** is then the following $(2\mathrm{P}+1) \times 6$ matrix (where P is the number of lags):

$$\mathbf{Corr[RS]} = \begin{bmatrix} \mathrm{r_{xx}(P)} & \mathrm{r_{yy}(P)} & \mathrm{r_{zz}(P)} & \mathrm{r_{xy}(-P)} & \mathrm{r_{xz}(-P)} & \mathrm{r_{yz}(-P)} \\ \mathrm{r_{xx}(P-1)} & \mathrm{r_{yy}(P-1)} & \mathrm{r_{zz}(P-1)} & \mathrm{r_{xy}(-P+1)} & \mathrm{r_{xz}(-P+1)} & \mathrm{r_{yz}(-P+1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathrm{r_{xx}(1)} & \mathrm{r_{yy}(1)} & \mathrm{r_{zz}(1)} & \mathrm{r_{xy}(-1)} & \mathrm{r_{xz}(-1)} & \mathrm{r_{yz}(-1)} \\ 1 & 1 & 1 & \mathrm{r_{xy}(0)} & \mathrm{r_{xz}(0)} & \mathrm{r_{yz}(0)} \\ \mathrm{r_{xx}(1)} & \mathrm{r_{yy}(1)} & \mathrm{r_{zz}(1)} & \mathrm{r_{xy}(1)} & \mathrm{r_{xz}(1)} & \mathrm{r_{yz}(1)} \\ \mathrm{r_{xx}(2)} & \mathrm{r_{yy}(2)} & \mathrm{r_{zz}(2} & \mathrm{r_{xy}(2)} & \mathrm{r_{xz}(2)} & \mathrm{r_{yz}(2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathrm{r_{xx}(P)} & \mathrm{r_{yy}(P)} & \mathrm{r_{zz}(P)} & \mathrm{r_{xy}(P)} & \mathrm{r_{xz}(P)} & \mathrm{r_{yz}(P)} \end{bmatrix} \qquad (5.2)$$

where $r_{xy}(p)$ is the estimate of the correlation function between the time series $\{x_i\}$ and $\{y_i\}$ at lag p, $\rho_{xy}(p)$; i.e.,

$$\rho_{xy}(p) = \frac{E[(x_i - E(x_i))(y_{i+p} - E(y_i))]}{\left[E[(x_i - E(x_i))^2]\right]^{1/2}\left[E[(y_i - E(y_i))^2]\right]^{1/2}} \tag{5.3}$$

(with $E(\cdot)$ being the expectation operator). The estimate, $r_{xy}(p)$, of $\rho_{xy}(p)$ is given by the estimator:

$$r_{xy}(p) = \frac{\dfrac{\sum_{i=1}^{r-p}(x_i - \bar{x})(y_{i+p} - \bar{y})}{r}}{\left[\dfrac{\sum_{i=1}^{r}(x_i - \bar{x})^2}{r}\right]^{1/2}\left[\dfrac{\sum_{i=1}^{r}(y_i - \bar{y})^2}{r}\right]^{1/2}} \quad , \quad 0 \le p \le P \tag{5.4}$$

where $\bar{x} = \dfrac{\sum_{i=1}^{r} x_i}{r}$ and $\bar{y} = \dfrac{\sum_{i=1}^{r} y_i}{r}$ (i.e., the estimate for the mean). For negative lags, using the fact that $\rho_{xy}(p) = \rho_{yx}(-p)$, the crosscorrelation estimate is:

$$r_{xy}(-p) = r_{yx}(p) \quad , \quad 0 \le p \le P \tag{5.5}$$

Also, Eq (5.2) takes into consideration the facts that the autocorrelation function and its estimate are symmetric about zero, i.e., $r_{xx}(-p) = r_{xx}(p)$ and that $r_{xx}(0) = 1$.

The correlation pattern of Eq (5.2) assumes weak stationarity up to second order for all variables (i.e., their mean, variance, autocorrelation and crosscorrelation functions for any lag are independent of the absolute time). Moreover, the estimator corresponding to Eq (5.4) is a biased but consistent estimator; i.e., its bias, variance and covariance tends to zero as the number of observations tend to infinity (Jenkins and Watts, 1969).

5.3.2 The Algorithm and Design Parameters of the Method

Let $\mathbf{RS}_i$, $i = 1,...,I$ be a the set of reference patterns, each corresponding to a known stochastic fault. Each is a matrix of $r_i \times N$, where r_i is the number of observations, and N is the number of measured variables. Also, let **TS** be the pattern of an unknown stochastic fault, a matrix of dimension $t \times N$; the objective is to find which fault is most likely to have produced the pattern **TS**. The proposed method is as follows:

Feature Extraction Steps: for each variable, in each $\mathbf{RS}_i$, do the following:

Step 1: Subtract the initial level.

Step 2: Filter with high-pass filter.

Step 3: Normalize to standard deviation of one.

Step 4: Filter with low-pass filter.

Let $\mathbf{RS}_{i,sc}$ be the patterns after the above scaling procedure.

Step 5: Compute the correlation pattern for each $\mathbf{RS}_{i,sc}$, $\mathbf{Corr}[\mathbf{RS}_{i,sc}]$

When the pattern **TS** of an unknown stochastic fault is given, apply Steps 1-5; let $\mathbf{Corr}[\mathbf{TS}_{sc}]$ be the corresponding correlation pattern.

Similarity Assessment Steps.

Step 6: Apply Dynamic Time Warping between $\mathbf{Corr}[\mathbf{RS}_{i,sc}]$ and $\mathbf{Corr}[\mathbf{TS}_{sc}]$ $\forall$ i.

Step 7: From the I distances obtained in Step 6, find the minimum; classify **TS** as a realization of the fault whose correlation pattern resulted in the minimum distance.

Steps 1 and 3 are not strictly necessary since the correlation pattern is magnitude and level independent; they are included only for consistency reasons with the method described in the previous chapter. Step 2 (high-pass filtering) removes low frequency trends. This is a necessary step, dictated by the second order stationarity assumption for all variables. If high-pass filtering is not applied, slow drifts tend to dominate the correlation estimates. Step 4 (low-pass filtering) removes high frequency noise and is not very crucial. Step 6 (DTW) applies a robust comparison between the unknown and the reference correlation patterns. Different production levels result in faster or slower process dynamics (and consecutively, contracted or expanded correlation patterns), making DTW an appropriate solution.

Thus, the method tries to classify stochastic faults in a way that is independent of the magnitude of the fault and of the plant production level (as mentioned before, the concept of 'direction' does not apply in stochastic processes). However, just like in the case of deterministic faults, the magnitude of the fault affects the variables' signal to noise ratio and is an important factor for the certainty of the classification. Faults with small magnitude will be masked by the common process variation and will be poorly diagnosed (if detected at all).

The method assumes that the correlation structure does not change for realizations of the same fault at different operating points. This is a valid assumption when the process operates at different production levels, but it may not be valid for different modes of operation (e.g., different final products). In such a case, the same fault may exhibit different correlation structures, and consecutively, a correlation-based diagnostic scheme will fail to give the correct diagnosis. In such a case, realizations of the same fault at the different operating modes have to be included in the database. Finally, the on-line and off-line implementation of the method are exactly similar. The only extra consideration in the on-line case is whether a large number of data points has been collected to provide an accurate estimate of correlation pattern.

The main design parameter of the method is the type of the high-pass filter. As mentioned in the previous chapter, high order filters are characterized by sharp frequency responses; however, they induce oscillations in the signals. A simple first order high-pass filter does nor have a sharp frequency response, but produces smoother output signals. The local distance in the DTW algorithm can be the Euclidean distance (i.e., **W** being the identity matrix), unless there is prior knowledge about the importance of some auto- or crosscorrelation estimates. Again, because no assumptions are made about statistical distributions and only one realization for each stochastic fault is assumed, the final distance from DTW can only be used as a relative similarity measure. Finally, one has to decide on the DTW algorithm. This is the subject of the next section.

5.4 A Modified Dynamic Time Warping Algorithm

As mentioned in Chapter 3, one of the parameters in a DTW algorithm is the endpoint constraint, with the fixed-endpoint being the simplest constraint. However, fixing the endpoints is not a reasonable constraint when comparing two correlation patterns where one may be a contracted or expanded version of the other (due to faster or slower process dynamics). For example, the autocorrelation coefficient at lag 10 of the expanded pattern may be equivalent to the coefficient at a previous lag (e.g., 9 or 8) of the contracted pattern. DTW can handle this problem both in the interior and at the edges of the search area by allowing the end points of the path to lie in a region of the perimeter of the search area. Section 3.4 and Figures 3.7(a) and 3.7(b) show how one can relax the fixed-endpoint constraints for a symmetric and an asymmetric DTW algorithm.

Furthermore, the autocorrelation function at lag zero is by definition equal to one for all variables. Thus, forcing the optimal path to exactly match the correlations at lag zero for both patterns is a reasonable constraint. Figure 5.2 illustrates this idea; the figure shows the optimal path resulting from the comparison of two correlation patterns with the same number of lags. The star indicates the fixed-point constraint, while the open circles indicate the first and the last optimal path points, as found by the DTW algorithm. The DTW algorithm that implements such a pattern matching is as follows.

Let **Corr[RS]** and **Corr[TS]** be two correlation patterns with the structure shown in Eq (5.2), each a matrix of dimension $(2P+1)\times\frac{N(N+1)}{2}$, where P is the number of lags and N is the number of the variables. Assume that the symmetric Sakoe-Chiba local constraint with the symmetric weighting function (both shown in Figure 3.5(a)) are used, together with a constraint for consecutive horizontal or vertical local transitions. Also, the global band constraint is used.

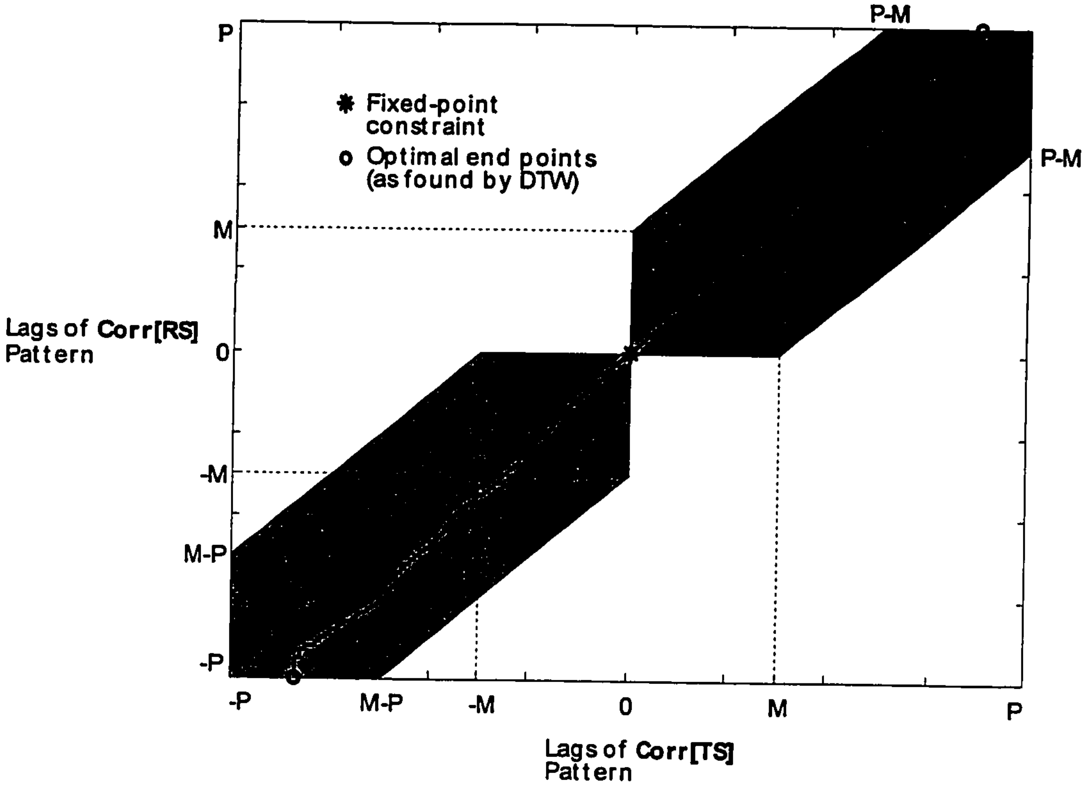


Figure 5.2: Global and endpoint constraints used in the comparison between two correlation patterns **Corr[TS]** and **Corr[RS]** via Dynamic Time Warping.

The algorithm consists of two stages: one for positive and one for negative lags. In each stage, the DTW algorithm is very similar to the one described in Subsection 3.4.2; the only difference is in the initialization step where it accounts for the fact that the correlations at zero lag are included in both stages. The steps are the following:

Stage A: Comparison for positive lags.

Step 1: Let $\mathbf{X} = \mathbf{Corr[TS]}(P+1{:}2P+1,:)$

Let $\mathbf{Y} = \mathbf{Corr[RS]}(P+1{:}2P+1,{:})$

Step 2: Give M, the parameter that defines the width $(2M+1)$ of the band within which the search for the optimal path will take place; this is the upper right shaded area of Figure 5.2.

Give m, the maximum number of consecutive horizontal or vertical local transitions.

Step 3: Let $d(\mathrm{i},\mathrm{j}) = [\mathbf{X}(\mathrm{i},{:}) - \mathbf{Y}(\mathrm{j},{:})]\,\mathbf{W}\,[\mathbf{X}(\mathrm{i},{:}) - \mathbf{Y}(\mathrm{j},{:})]^{\mathrm{T}}$ (i.e., the local distance).

Start the Dynamic Programming recursion for all allowable (i,j) points.

$\mathbf{D}_{\mathrm{A}}(1,1) = d(1,1)$

$$\mathbf{D}_{\mathrm{A}}(\mathrm{i},\mathrm{j}) = \min\left\{\begin{array}{l} [\mathbf{D}_{\mathrm{A}}(\mathrm{i}-1,\mathrm{j}) + d(\mathrm{i},\mathrm{j})] \text{ or } [\infty \text{ if condition (A)}] \\ \mathbf{D}_{\mathrm{A}}(\mathrm{i}-1,\mathrm{j}-1) + 2d(\mathrm{i},\mathrm{j}) \\ [\mathbf{D}_{\mathrm{A}}(\mathrm{i},\mathrm{j}-1) + d(\mathrm{i},\mathrm{j})] \text{ or } [\infty \text{ if condition (B)}] \end{array}\right\},$$

where: Condition (A): predecessor of point $(\mathrm{i}-1,\mathrm{j})$ is the point $(\mathrm{i}-\mathrm{m}-1,\mathrm{j})$ through m consecutive horizontal moves.
Condition (B): predecessor of point $(\mathrm{i},\mathrm{j}-1)$ is the point $(\mathrm{i},\mathrm{j}-\mathrm{m}-1)$ through m consecutive vertical moves.

Store the optimal predecessor for the (i,j) point.

Step 4: Minimum Normalized Total Distance for positive lags:

$$\hat{D}_{+} = \frac{1}{2P+1}\min\begin{bmatrix} \mathbf{D}_{\mathrm{A}}(P+1,\mathrm{j})\dfrac{2P+1}{P+\mathrm{j}}, & P+1-M \le \mathrm{j} \le P+1 \\ \mathbf{D}_{\mathrm{A}}(\mathrm{i},P+1)\dfrac{2P+1}{\mathrm{i}+P}, & P+1-M \le \mathrm{i} \le P+1 \end{bmatrix}$$

Step 5: Reconstruct the optimal path for positive lags, $\hat{\mathbf{F}}_{+}$, starting from the point in which $\hat{D}_{+}$ occurs (last path point), and travel backwards as the optimal predecessor indices dictate until point $(1,1)$ is reached.

Stage B: Comparison for negative lags.

Step 6: Let $\mathbf{X} = \mathbf{Corr[TS]}(P+1{:}{-1}{:}1,{:})$

Let **Y** = **Corr[RS]**(P + 1:-1:1,:)

Both X and Y contain at their first row the correlations at lag zero and at their last row the correlations at lag $-\mathrm{P}$.

Apply Steps 3 to 5. Find the Minimum Normalized Total Distance for negative lags, $\hat{D}_{-}$, and reconstruct the optimal path for negative lags, $\hat{\mathbf{F}}_{-}$.

Finally, combine the results from the two stages:

Minimum Normalized Total Distance: $\hat{D} = \hat{D}_{+} + \hat{D}_{-}$

Optimal Path: $\hat{\mathbf{F}} = \begin{bmatrix} \hat{\mathbf{F}}_{+} \\ \hat{\mathbf{F}}_{-} \end{bmatrix}$

The algorithm is a symmetric one; both correlation patterns are considered to be equivalent. The local constraint can be replaced by any of the other Sakoe-Chiba local constraints (Figures 3.3(b), 3.3(c), 3.3(d)). However, the symmetric weighting function has to be used in all cases. The normalization of distances at Step 4 is done on the basis of number of local distances computed for each accumulated distance (as has been described in detail in Subsection 3.4.2).

This completes the description of the method proposed to diagnose stochastic faults. Its application is illustrated in the next section with case studies from the Tennessee-Eastman plant simulation.

5.5 Case Studies

5.5.1 Description of the Patterns in the Reference and Test Sets

The reference set consisted of two patterns, **RS$_1$** and **RS$_2$** corresponding to the two major stochastic upsets: IDV(8) and IDV(13). Tables 5.2 and 5.3 describe the details of the reference and the test set patterns.

Table 5.2: Patterns in the Reference Set

Pattern	Fault	Operating Point	Step size / direction	Duration / # of points	Fault occurs after / at
RS$_1$	IDV(8)	Nominal	+1.0	35 hrs / 701 pts	3 hrs / 61st pt
RS$_2$	IDV(13)	Nominal	+1.0	30 hrs / 601 pts	3 hrs / 61st pt

The test patterns are selected to illustrate different aspects of the classification problem. Some are identical (with the exception of the seed for the random number generator) to the reference set patterns, while others are quite different (i.e., realizations of the same faults with different magnitudes at the "Reduced Production" operating point). Another issue to be examined (which is crucial for the on-line implementation of the method) is the effect of the time series length on the correlation estimates and consecutively, on the classification.

Table 5.3: Patterns in the Test Set

Pattern	Fault	Operating Point	Step size	Duration / # of points	Fault occurs after / at
TS_1	IDV(8)	Nominal	1.0	35 hrs / 701 pts	3 hrs / 61st pt
TS_2	IDV(8)	Nominal	0.5	35 hrs / 701 pts	3 hrs / 61st pt
TS_3	IDV(8)	Nominal	0.8	35 hrs / 701 pts	3 hrs / 61st pt
TS_4	IDV(8)	Nominal	0.8	25 hrs / 501 pts	3 hrs / 61st pt
TS_5	IDV(8)	Nominal	1.0	20 hrs / 401 pts	3 hrs / 61st pt
TS_6	IDV(8)	Nominal	0.7	30 hrs / 601 pts	3 hrs / 61st pt
TS_7	IDV(8)	Reduc. Prod.	1.0	35 hrs / 701 pts	3 hrs / 61st pt
TS_8	IDV(8)	Reduc. Prod.	0.5	35 hrs / 701 pts	3 hrs / 61st pt
TS_9	IDV(8)	Reduc. Prod.	0.8	35 hrs /701 pts	3 hrs / 61st pt
TS_{10}	IDV(8)	Reduc. Prod.	0.8	25 hrs /501 pts	3 hrs / 61st pt
TS_{11}	IDV(13)	Nominal	1.0	35 hrs / 701 pts	3 hrs / 61st pt
TS_{12}	IDV(13)	Nominal	0.5	35 hrs / 701 pts	3 hrs / 61st pt
TS_{13}	IDV(13)	Nominal	0.8	35 hrs / 701 pts	3 hrs / 61st pt
TS_{14}	IDV(13)	Nominal	0.8	25 hrs / 501 pts	3 hrs / 61st pt
TS_{15}	IDV(13)	Nominal	1.0	20 hrs / 401 pts	3 hrs / 61st pt
TS_{16}	IDV(13)	Nominal	0.7	30 hrs / 601 pts	3 hrs / 61st pt
TS_{17}	IDV(13)	Reduc. Prod.	1.0	35 hrs / 701 pts	3 hrs / 61st pt
TS_{18}	IDV(13)	Reduc. Prod.	0.5	35 hrs / 701 pts	3 hrs / 61st pt
TS_{19}	IDV(13)	Reduc. Prod.	0.8	35 hrs /701 pts	3 hrs / 61st pt
TS_{20}	IDV(13)	Reduc. Prod.	0.8	25 hrs /501 pts	3 hrs / 61st pt

Not all of the 26 measured variables were included in the patterns; 26 variables would result in 26 autocorrelations and 325 crosscorrelations. Although including all of them does not create computational problems, it was decided to include 8 variables out of the 26, on the basis that not all variables carry useful information. The next chapter presents a modification where all 26 variables are initially included, but their number is significantly reduced via Principal Component Analysis. The 8 variables used in the patterns in this chapter are given in Table I.2 of the Appendix; also, Figure I.3 shows their behavior during the reference patterns $\mathbf{RS}_1$ and $\mathbf{RS}_2$.

5.5.2 Selection of Design Parameters

The duration of most patterns is between 30 and 35 hrs, which is the suggested duration of the simulation in the original Tennessee-Eastman paper. Some patterns had a duration of 20 and 25 hrs; this was selected to see the effect of shorter time series on the correlation estimates. All faults were introduced after simulating 3 hrs of operation. The 60 measurements collected in these 3 hours prior to the fault were also used in the estimation of the correlation pattern (since uncertainty in the time origin of a fault is always present in practical situations). The time of introduction of the faults was not varied because exact synchronization of the patterns is not a major issue for stochastic faults when auto- and crosscorrelations are used as features.

The DTW algorithm is the one described in Section 5.4 with the parameters shown in Table 5.4.

Table 5.4: Design parameters in the DTW algorithm

Number of lags, P	80
Maximum deviation from linear path, M	30
Maximum number of consecutive horizontal or vertical moves, m	5
Weight matrix, **W**	Identity matrix, dimension 36×36

It is not easy to give definitive guidelines on how to select values for these parameters. The most important is the number of lags and this is something that can be determined based on process knowledge or by observing the correlation estimates. The recycle in the Tennessee-Eastman plant causes long dynamic behavior; thus, temporal correlations up to 4 hours (i.e., at lag 80) were included in the correlation patterns. Finally, the filters described in the previous chapter were used for the cases studies of this chapter.

Figure 5.3 shows the same 4 variables (of Figure 5.1) for the patterns $\mathbf{RS}_{1,sc}$, $\mathbf{RS}_{2,sc}$, $\mathbf{TS}_{1,sc}$ and $\mathbf{TS}_{19,sc}$ (i.e., after filtering and variance normalization has been applied). Figures 5.4 and 5.5 show part of the correlation patterns for the scaled reference patterns, $\mathbf{Corr}[\mathbf{RS}_{1,sc}]$ and $\mathbf{Corr}[\mathbf{RS}_{2,sc}]$. The correlation patterns express the fact that fault IDV(13) contains lower frequencies than IDV(8); the patterns from $\mathbf{Corr}[\mathbf{RS}_{2,sc}]$ do not die out as fast as the ones from $\mathbf{Corr}[\mathbf{RS}_{1,sc}]$. However, with the exception of this difference, most of the crosscorrelation patterns look quite similar for both faults.

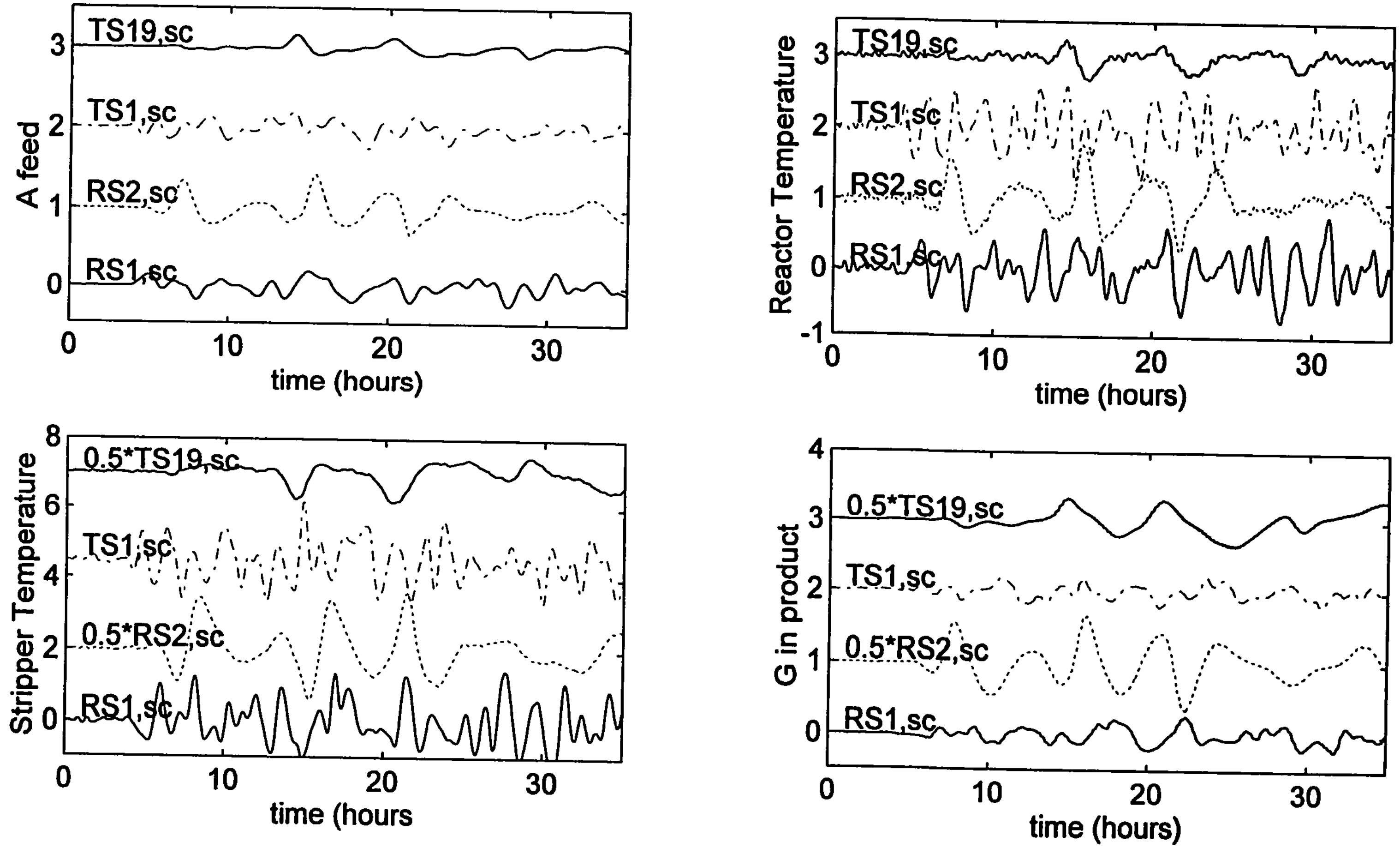


Figure 5.3: Behavior of the 4 variables during the scaled patterns, $\mathbf{RS}_{1,SC}$, $\mathbf{RS}_{2,SC}$, $\mathbf{TS}_{1,SC}$ and $\mathbf{TS}_{19,SC}$; the initial value for each variable is chosen to facilitate plotting.

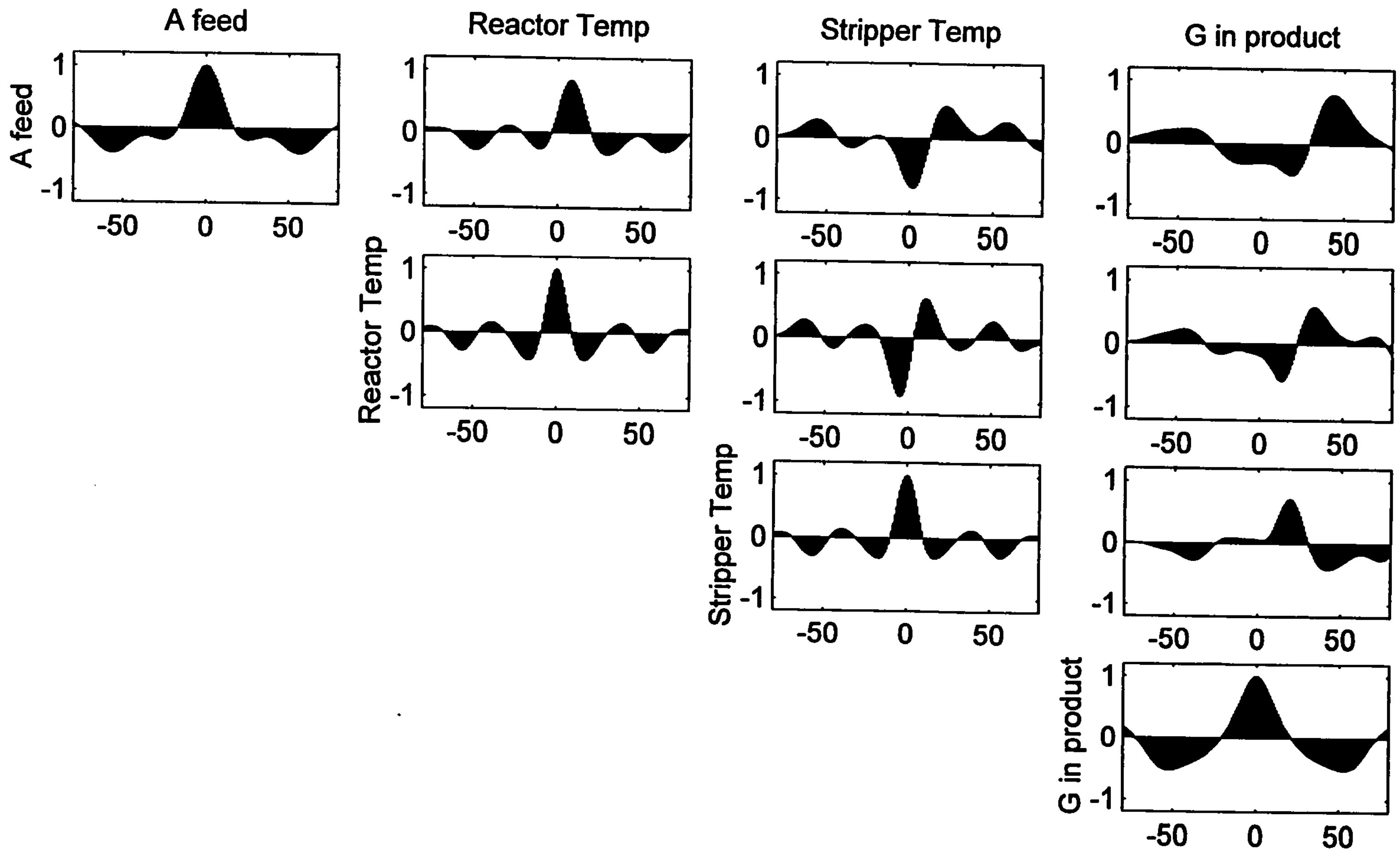


Figure 5.4: Autocorrelation and crosscorrelation estimates for the 4 variables of the scaled reference pattern $\mathbf{RS}_{1,sc}$.

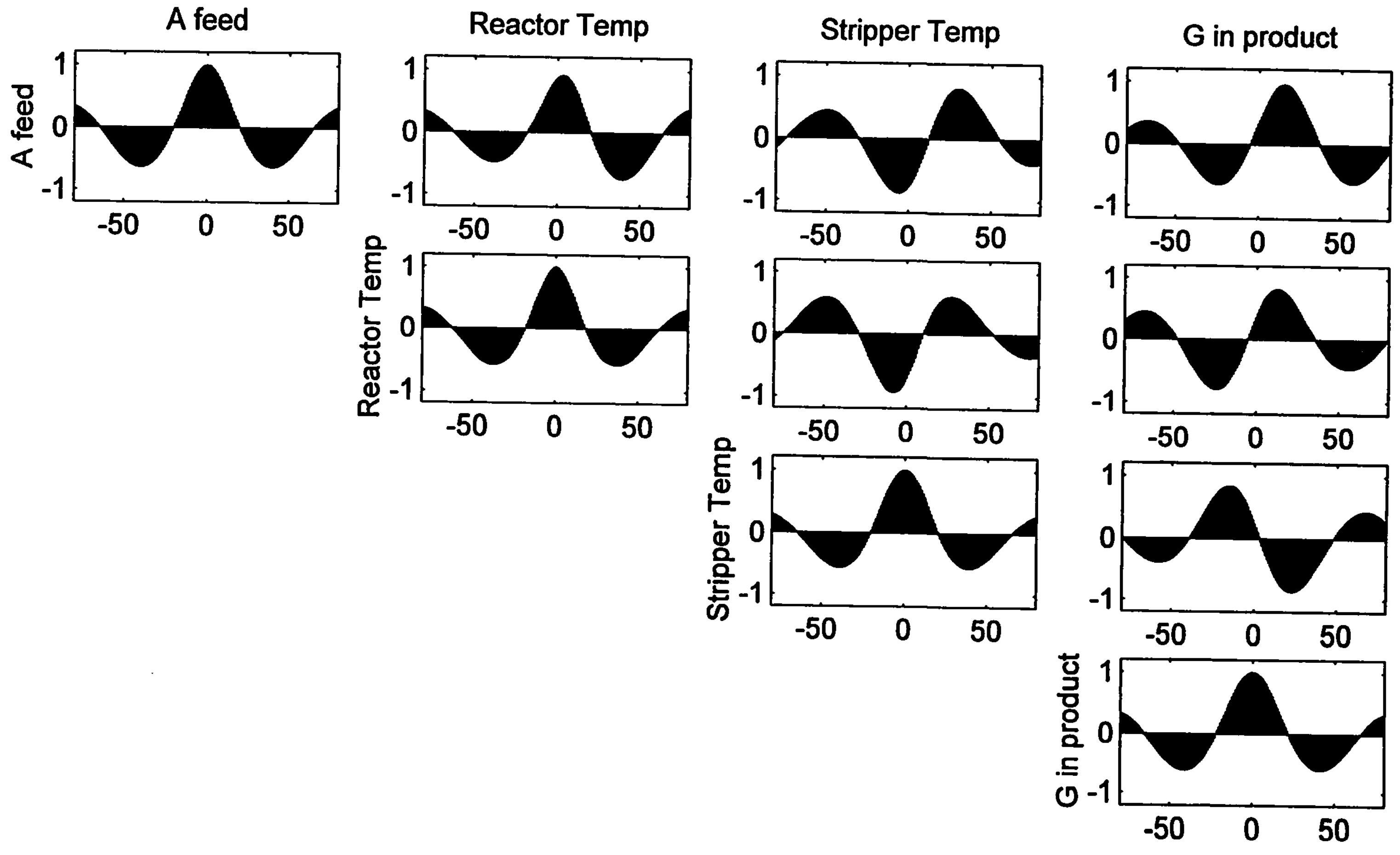


Figure 5.5: Autocorrelation and crosscorrelation estimates for the 4 variables of the scaled reference pattern $\mathbf{RS}_{2,sc}$.

5.5.3 Results

Table 5.5 shows the results from the pattern comparison via DTW. For each test pattern of Table 5.3, the Fault Diagnosis method is applied as presented in Subsection 5.3.2. Two minimum normalized total distances are obtained for each $\mathbf{Corr[TS_{i,sc}]}$:

$$\hat{D}(\mathbf{Corr[TS_{i,sc}]}, \mathbf{Corr[RS_{1,sc}]}),\ \hat{D}(\mathbf{Corr[TS_{i,sc}]}, \mathbf{Corr[RS_{2,sc}]})$$

The fault that corresponds to the reference pattern that gives the minimum of the two distances, $\hat{D}_{i,MIN}$, is selected as the most probable cause for the test pattern. Table 5.5 shows $\hat{D}_{i,MIN}$, the ratio of the two distances over $\hat{D}_{i,MIN}$ and the decision on the most probable cause for each test pattern. The shaded cells indicate the correct classification. Good discrimination between the two faults is obtained when one of the ratios is not close to one. Figures 5.6 and 5.7 show the results from applying DTW for the two correlation patterns $\mathbf{Corr[TS_{1,sc}]}$ and $\mathbf{Corr[TS_{19,sc}]}$. Both figures show the band constraint, the optimal path and the minimum normalized total distance.

Table 5.5: Results from Similarity Assessment via DTW; Distances from each DTW match over Minimum Distance and Final Decision.

Test Pattern	$\hat{D}_{i,MIN}$	$\hat{D}(\mathrm{Corr}[TS_{i,sc}], \mathrm{Corr}[RS_{1,sc}])$ over $\hat{D}_{i,MIN}$	$\hat{D}(\mathrm{Corr}[TS_{i,sc}], \mathrm{Corr}[RS_{2,sc}])$ over $\hat{D}_{i,MIN}$	Decision
TS_1	0.55	1.00	23.75	IDV(8)
TS_2	1.98	1.00	7.12	IDV(8)
TS_3	0.99	1.00	13.11	IDV(8)
TS_4	1.37	1.00	9.14	IDV(8)
TS_5	1.02	1.00	12.99	IDV(8)
TS_6	1.17	1.00	12.43	IDV(8)
TS_7	4.52	1.00	3.31	IDV(8)
TS_8	3.21	1.00	4.46	IDV(8)
TS_9	2.66	1.00	6.10	IDV(8)
TS_{10}	4.33	1.00	3.44	IDV(8)
TS_{11}	3.47	3.47	1.00	IDV(13)
TS_{12}	1.52	6.55	1.00	IDV(13)
TS_{13}	8.08	1.41	1.00	IDV(13)
TS_{14}	1.51	6.62	1.00	IDV(13)
TS_{15}	5.25	1.89	1.00	IDV(13)
TS_{16}	0.96	12.72	1.00	IDV(13)
TS_{17}	7.24	1.88	1.00	IDV(13)
TS_{18}	6.04	1.45	1.00	IDV(13)
TS_{19}	8.32	1.17	1.00	IDV(13)
TS_{20}	4.12	2.56	1.00	IDV(13)

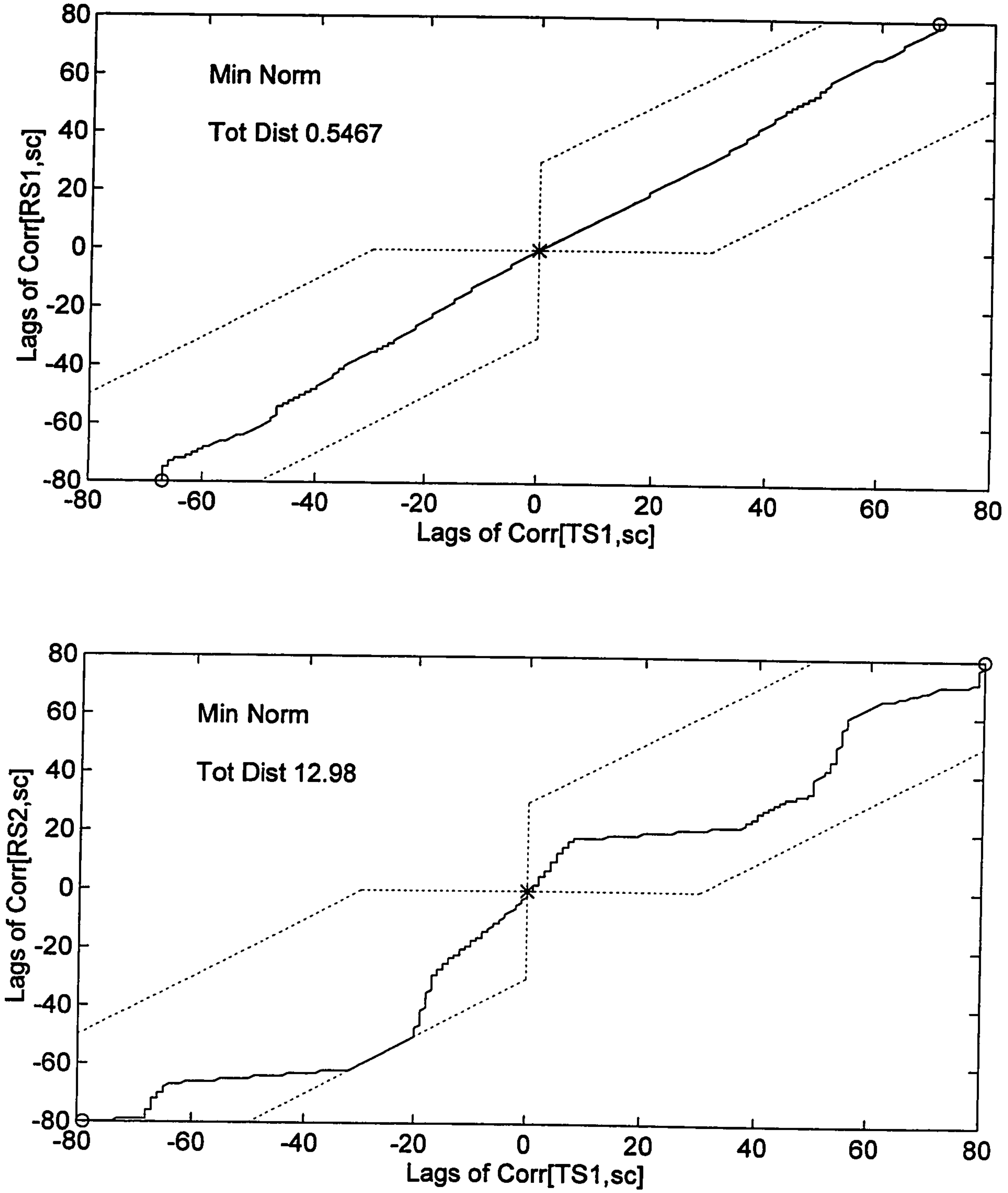
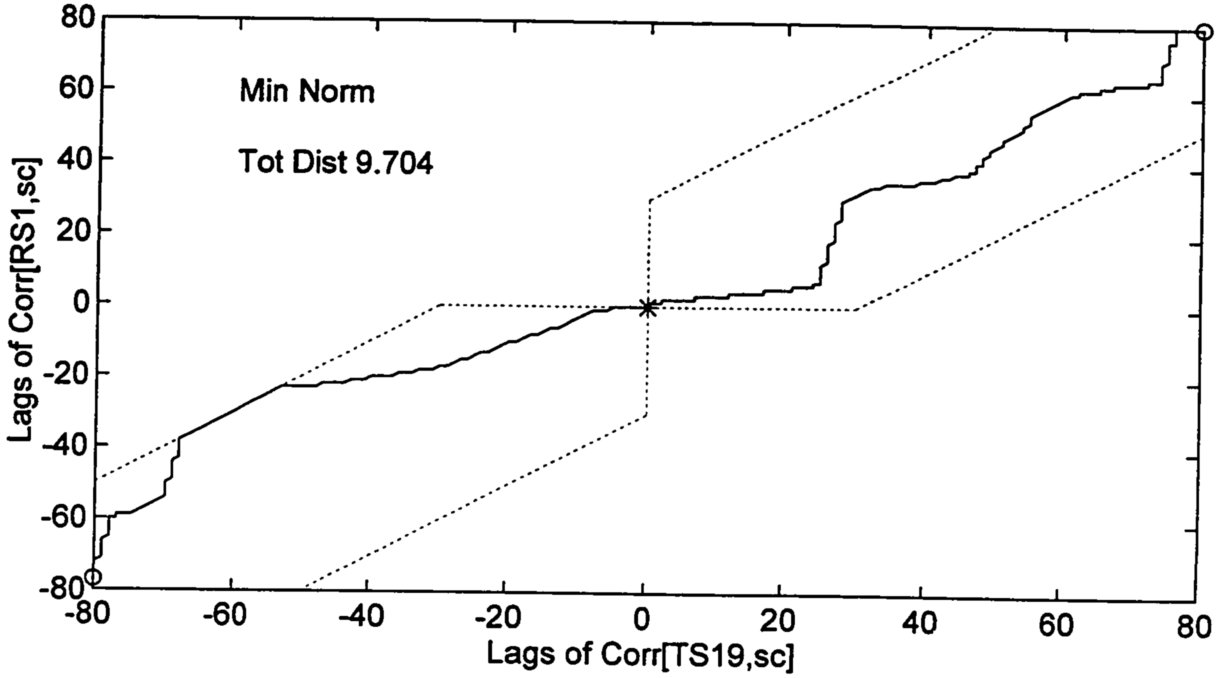


Figure 5.6: Results for test pattern **TS$_1$**; global constraints, endpoint constraints, optimal path and Minimum Normalized Total Distance between **Corr[TS$_{1,sc}$]** and **Corr[RS$_{1,sc}$]** and between **Corr[TS$_{1,sc}$]** and **Corr[RS$_{2,sc}$]**.



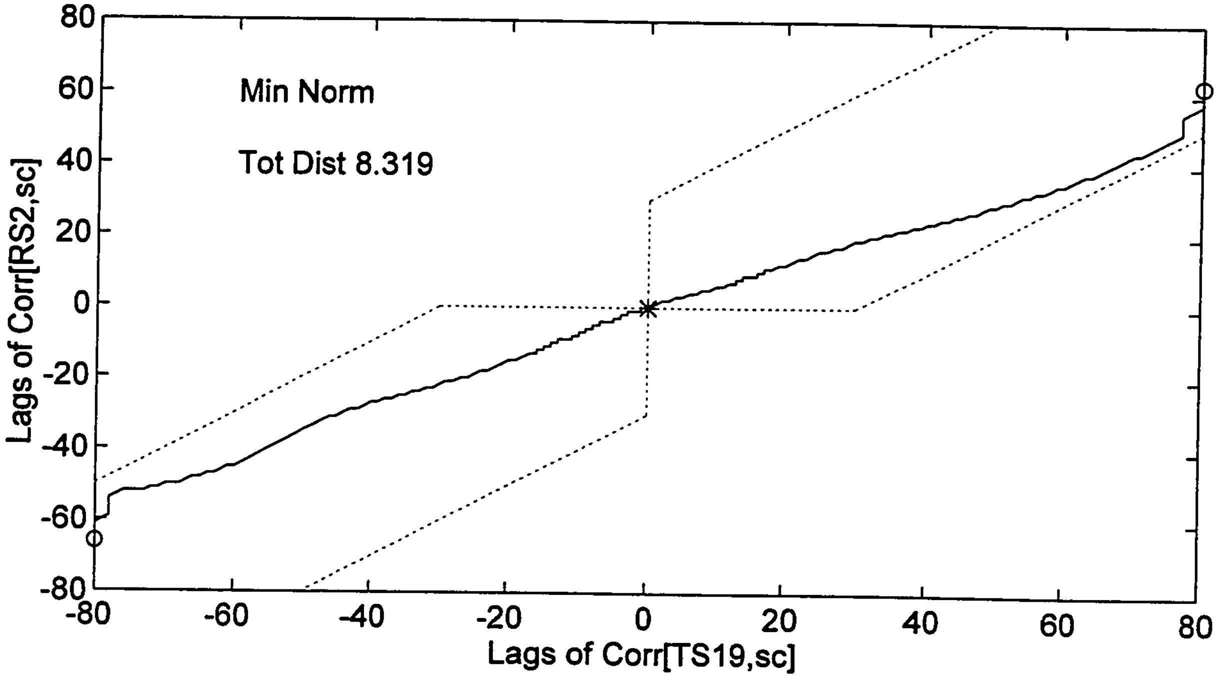


Figure 5.7: Results for test pattern **TS₁₉**; global constraints, endpoint constraints, optimal path and Minimum Normalized Total Distance between **Corr[TS$_{19,SC}$]** and **Corr[RS$_{1,SC}$]** and between **Corr[TS$_{19,SC}$]** and **Corr[RS$_{2,SC}$]**.

Examining the results of Table 5.5, one can observe the following:

A) All test patterns are diagnosed correctly.

B) For the IDV(8) fault, the results are as expected. For example, fault realizations with magnitude of one ($\mathbf{TS}_1$) are classified with more certainty than realizations with smaller fault magnitude (e.g., $\mathbf{TS}_2$, $\mathbf{TS}_3$). Similarly, faults of longer time series (e.g., $\mathbf{TS}_1$) are classified with more certainty than ones with shorter time series (e.g., $\mathbf{TS}_4$). The operating point is again the most important factor, as the results for $\mathbf{TS}_1$-$\mathbf{TS}_7$ and $\mathbf{TS}_8$-$\mathbf{TS}_{10}$ indicate.

C) However, the results for the IDV(13) fault are not as expected. Comparing the results for patterns $\mathbf{TS}_{11}$ and $\mathbf{TS}_{12}$, one can see that although $\mathbf{RS}_2$ is more similar to $\mathbf{TS}_{11}$ (default size of one) than $\mathbf{TS}_{12}$ (size of 0.5), the discrimination is better for $\mathbf{TS}_{12}$ than $\mathbf{TS}_{11}$. Also, even though $\mathbf{TS}_{14}$ is a shorter time series, it is classified with more certainty as IDV(13) than $\mathbf{TS}_{11}$, which is a longer time series. The operating point is again the most important factor for the classification of the IDV(13) fault (as the results for $\mathbf{TS}_{11}$-$\mathbf{TS}_{16}$ and $\mathbf{TS}_{17}$-$\mathbf{TS}_{20}$ indicate) but not to the same extent as is for the IDV(8) fault.

D) The stochastic nature of the faults, coupled with the nonlinear process and the effect of the recycle on the plant dynamic behavior, may explain the results of this chapter. If the fault is an uninterrupted stochastic event (e.g., the composition of the feed is randomly fluctuating), then a nonlinear plant may take a long time to express a "consistent" stochastic behavior that can be detected by observing the auto- and crosscorrelation coefficients. Even more so, when a recycle feeds back the output fluctuations to the plant input. Therefore, simulated data of 35 hours duration may not be enough for an accurate estimate of the correlation pattern.

5.6 Summary and Conclusions

In this chapter a method is proposed for the off-line and on-line diagnosis of stochastic faults in dynamic continuous multivariable processes, based on Pattern Recognition principles. No process model or other process knowledge is required. The method requires a set of reference patterns, each describing a known past fault in the process variables. After a scaling procedure, which essentially removes the low frequency trends from the signals via high-pass filtering, the autocorrelation and crosscorrelation estimates are extracted; these are the features used to classify the pattern of an unknown fault. The decision scheme is a minimum distance classifier, where Dynamic Time Warping is used for pattern matching to account for differences in the correlation patterns due to faster or slower plant dynamics.

Because the correlations are independent of the magnitude of the variables, the method can correctly diagnose faults independently of their magnitude. Also, the method classifies faults independently of the production level via the robust pattern matching that DTW offers. The method can deal with any number of variables and no assumptions on statistical distributions are required. On the other hand, all variables and consecutively their autocorrelation and crosscorrelations with other variables, are considered equally important. Correlations result in a large increase of the dimension of the space where the pattern comparison takes place and some of them may carry small amounts of information. This issue will be addressed in the next chapter.

The most important assumptions of the method are that I) a fault, whose source is a stochastic process, will result in a consistent correlation pattern in the process variables; and II) different realizations of the same stochastic fault will result in similar correlation patterns. The case studies from the Tennessee-Eastman simulation indicated that both assumptions are questionable and their validity depends on the specific fault, particularly when nonlinear processes with long dynamics are encountered. The Tennessee-Eastman simulation is characterized by these features (i.e., nonlinear processes, long dynamics due to recycle) and therefore poor diagnostic results were obtained.

CHAPTER 6

REDUCTION OF PROBLEM DIMENSION

VIA PRINCIPAL COMPONENT ANALYSIS

In this chapter the methods presented in the two previous chapters will be augmented with the addition of Principal Component Analysis (PCA) which can significantly reduce the dimension of the patterns for both deterministic and stochastic faults. In the case of deterministic faults, the use of PCA significantly improved the discriminatory power of the classifier, but inconclusive results were obtained for stochastic faults.

6.1 Introduction - Principal Component Analysis

The patterns of the deterministic faults in Chapter 4 contained 26 variables and all of them were considered equally important in the Dynamic Time Warping-based similarity assessment step. However, not all 26 variables carry useful information for Fault Diagnosis purposes; some variables may not be affected by the fault and as a result they may not exhibit any deterministic pattern. Including these noisy variables in the similarity assessment will inflate the distance found by Dynamic Time Warping and consequently reduce the discriminatory power of the classifier. Furthermore, the 26 variables are not independent but are highly correlated. Thus, one could use a smaller number of variables and still retain most of the information.

For the stochastic faults of Chapter 5, 8 certain variables out of the 26 were chosen to represent the pattern of a fault. Had all 26 variables been used to estimate the correlation patterns of faults, one would have had to estimate 356 auto- and crosscorrelations. Although this does not create any computational problems, it is a tremendous increase in the dimension of the space where the similarity assessment procedure takes place. This large number of auto- and crosscorrelations is not necessary, since these 26 variables are not independent of each other. Moreover, some of these correlations may not contain useful information, since some variables may not be affected by the fault.

Thus, what is required in both cases is a method that will reduce the dimension of the patterns in an "optimal" way. Principal Component Analysis (PCA) is such a method (Jollife, 1986, Wold, 1987). Let **X** be a $t \times N$ matrix consisting of t observations on N correlated variables. PCA finds new, fictitious, uncorrelated variables, called principal components, that summarize the information in **X**. The first principal component is the direction in the physical variables along which the data exhibit the greatest variability (as expressed by their sum of squares). Subsequent principal components explain the remaining variability, while being orthogonal to the previous principal components. One can view PCA as a sequence of two steps: the first is a rotation of the space of the physical variables to create orthogonal directions along which the variability of the process lays; the second step is a projection of the original data onto the subspace defined by the principal components. The whole procedure is beneficial in cases where the original variables are highly correlated, because a much small number of principal components is sufficient to capture most of the variability present in the original data.

In mathematical terms, matrix **X** is decomposed as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \qquad (6.1)$$

where **P** is an $N \times K$ matrix with $K < N$, and **T** a $t \times K$ matrix. The columns of **P**, called loading vectors, express the relation of the K principal components with the original variables and by construction $\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I}$. The columns of **T**, called score vectors, are the coordinates of the t N-variate data points of **X** in the subspace of the principal components. Also, $\mathbf{T}^T\mathbf{T}$ is a diagonal matrix, whose entries are the sum of squares of the original data along the directions defined by the principal components. Alternatively, the diagonal elements of $\mathbf{T}^T\mathbf{T}$ are the K largest eigenvalues of the $\mathbf{X}^T\mathbf{X}$ matrix, in descending order, and $\mathbf{P}$ contains the K associated eigenvectors of $\mathbf{X}^T\mathbf{X}$. $\mathbf{E}$ is the residual matrix; it contains the variability that cannot be explained by the principal components and in general represents process noise. From a different point of view, $\mathbf{T}\mathbf{P}^T$ is the best rank-K approximation of a rank-N matrix $\mathbf{X}$ in the sense of the Frobenious norm (Golub and Van Loan, 1989).

Because the objective of PCA is to describe the variability of $\mathbf{X}$ as measured by the sum of squares, the scaling of variables is a defining factor. Scaling each variable so that it has a zero average and a standard deviation one is the most common procedure and gives equal weight to each variable. Another common scaling procedure is to weight the variables by their relative importance, assuming that this process knowledge is available (Kresta et al., 1991, Kourti and MacGregor, 1995). To decide on the number of principal components, K, there is a number of criteria that one can use. Cross-validation is the most popular one (Kourti and MacGregor, 1995), while other criteria include the broken stick rule (Jollife, 1986) and the parallel analysis (Ku et al., 1995). According to the latter, one plots the eigenvalues of $\mathbf{X}^T\mathbf{X}$ and the eigenvalues of $\mathbf{Y}^T\mathbf{Y}$; Y has the same dimension as $\mathbf{X}$ and contains a generated data set in which all the elements are independent random deviates. One then finds the point where the two curves cross and that point defines the number of principal components to be retained.

One can also build confidence intervals for various statistics resulting from PCA, based on assumptions for multivariate normal distributions and on approximations for the distributions of quadratic forms. The most common ones are the Hotelling T^2 and the Q statistics, which are defined by the following relationships:

$$\mathrm{T}^2 = \mathbf{T}(\mathrm{i},:)\left[\frac{\mathbf{T}^{\mathrm{T}}\mathbf{T}}{\mathrm{t}-1}\right]^{-1}\mathbf{T}(\mathrm{i},:)^{\mathrm{T}}, \quad \mathrm{i}=1,\ldots,\mathrm{t} \tag{6.2}$$

$$\mathrm{Q} = \left[\mathbf{X}(\mathrm{i},:) - \mathbf{T}(\mathrm{i},:)\mathbf{P}^{\mathrm{T}}\right]\left[\mathbf{X}(\mathrm{i},:) - \mathbf{T}(\mathrm{i},:)\mathbf{P}^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \mathrm{i}=1,\ldots,\mathrm{t} \tag{6.3}$$

Confidence intervals for these statistics can be found in Kourti and MacGregor (1995) and in Nomikos and MacGregor (1995a). A large value for the T^2 statistic indicates an abnormality which causes a larger variability than the common cause variability of the normal operating data. A large Q value indicates an abnormality that changes the correlation structure present in the normal operating data. Details on multivariate monitoring of continuous processes at steady-state conditions using PCA can be found in Kresta and MacGregor (1991). The next two sections describe the incorporation of PCA into the Dynamic Time Warping methods of Chapter 4 and 5.

6.2 Diagnosis of Deterministic Faults

6.2.1 The Algorithm and Design Parameters of the Method

Let $\mathbf{R}_i$, $i=1,\ldots,I$ be a the set of reference patterns, each representing a known deterministic fault. Each is a matrix of $r_i \times N$, where r_i is the number of measurements and N is the number of measured variables. Also, let $\mathbf{T}_j$ be the pattern of an unknown deterministic fault, a matrix of dimension $t_j \times N$. The method presented in Chapter 4 is modified as follows.

Feature Extraction Steps: for each variable, in each $\mathbf{R}_i$ do the following:

Step 1: Subtract the initial level.

Step 2: Filter with high-pass filter.

Step 3: Normalize to standard deviation of one.

Step 4: Filter with low-pass filter.

Let $\mathbf{R}_{i,sc}$ be the patterns after the above scaling procedure.

Step 5: Create a matrix $\mathbf{X}$ with all the scaled patterns; i.e., $\mathbf{X} = \begin{bmatrix} \mathbf{R}_{1,sc} \\ \mathbf{R}_{2,sc} \\ \cdot \\ \cdot \\ \mathbf{R}_{I,sc} \end{bmatrix}$.

Apply PCA on $\mathbf{X}$ (i.e., $\mathbf{X} = \mathbf{TP}^{\mathrm{T}} + \mathbf{E}$); store the loading vectors in matrix $\mathbf{P}$. Also, let the corresponding points of the score vectors be the descriptions of the scaled reference patterns in the space of principal components:

$$\mathbf{R}_{1,sc}^{PCA} = \mathbf{T}(1:r_1,:),\ \mathbf{R}_{2,sc}^{PCA} = \mathbf{T}(r_1+1:r_1+r_2,:),\ldots,\mathbf{R}_{I,sc}^{PCA} = \mathbf{T}(\left[\sum_{k=1}^{I-1} r_k\right]+1:\sum_{k=1}^{I} r_k,:)$$

When the pattern of the unknown fault, $\mathbf{T}_j$, becomes available, apply Steps 1-4; let $\mathbf{T}_{j,sc}$ be the resulting scaled test pattern.

Step 6: project $\mathbf{T}_{j,sc}$ onto the subspace defined by the principal components: $\mathbf{T}_{j,sc}^{PCA} = \mathbf{T}_{j,sc}\mathbf{P}$.

Similarity Assessment Steps.

Step 7: Apply Dynamic Time Warping between $\mathbf{R}_{i,sc}^{PCA}$ and $\mathbf{T}_{j,sc}^{PCA}$ for $i = 1,\ldots,I$.

Step 8: Apply Dynamic Time Warping between $-\mathbf{R}_{i,sc}^{PCA}$ and $\mathbf{T}_{j,sc}^{PCA}$ for $i = 1,\ldots,I$.

Step 9: From the $2 \cdot I$ distances obtained in Steps 7 and 8, find the minimum; the fault whose pattern results in the minimum distance is deemed to be the most likely to have generated the test pattern.

The method extends the feature extraction stage by including PCA. After Step 5, the dimension of the scaled reference patterns will be significantly smaller than N if the patterns contain correlated variables. The matrix **P** will contain linear combinations of the scaled variables that contain most of the variation of the scaled reference patterns.

Linear combinations with small variation correspond to subsequent principal components and they will not be included in the new descriptions, $\mathbf{R}_{i,sc}^{PCA}$. However, some of these ignored principal components will still contain low variability deterministic patterns; this information is ignored by this method.

Alternatively, one could apply the canonical analysis Box and Tiao (1977) to find linear combinations of variables that are most predictable from their past history. With the Box and Tiao method the ignored components contain more random patterns, but on the other hand they may contain more variation than the ignored principal components from PCA. The Box and Tiao method applies only for multivariate autoregressive stochastic processes and it requires that the autocovariance matrix at lag zero is full rank. Both are strict assumptions (particularly the latter) for patterns of faults from chemical processes where the variables are highly correlated. For these reasons, the Box and Tiao method was not implemented in the proposed method.

The design parameters of the method are the ones of Chapter 4 (see Section 4.3) with the addition of the number of principal components to be retained. One can use any of the aforementioned criteria (i.e., cross-validation, the broken stick rule, parallel analysis) or their combination. Regarding the other design parameters (filter design, DTW variant) the comments of Section 4.3 apply here as well.

6.2.2 Case Studies, Results and Conclusions

Case studies from the Tennessee-Eastman simulation were used to evaluate the application of PCA in the Dynamic Time Warping-based method of Chapter 4. The case studies of Chapter 4 are also used in this chapter. The reference set patterns are shown in Table 4.2 and the test patterns are shown in Table 4.3. Also the same filters and the same DTW variant used in Chapter 4 were used in these case studies. The simple Euclidean distance (i.e., $\mathbf{W}$ being the identity matrix) was used as the local distance. All the 26 variables were included in the patterns.

To decide on the number of principal components the parallel analysis was used; the results are shown in Figure 6.1. Matrix Y has the same dimension as X and contains random numbers drawn independently from a normal distribution with mean zero and variance one. Based on these results, 4 principal components were retained. Figure 6.2 shows the scaled reference patterns along the 4 principal components.

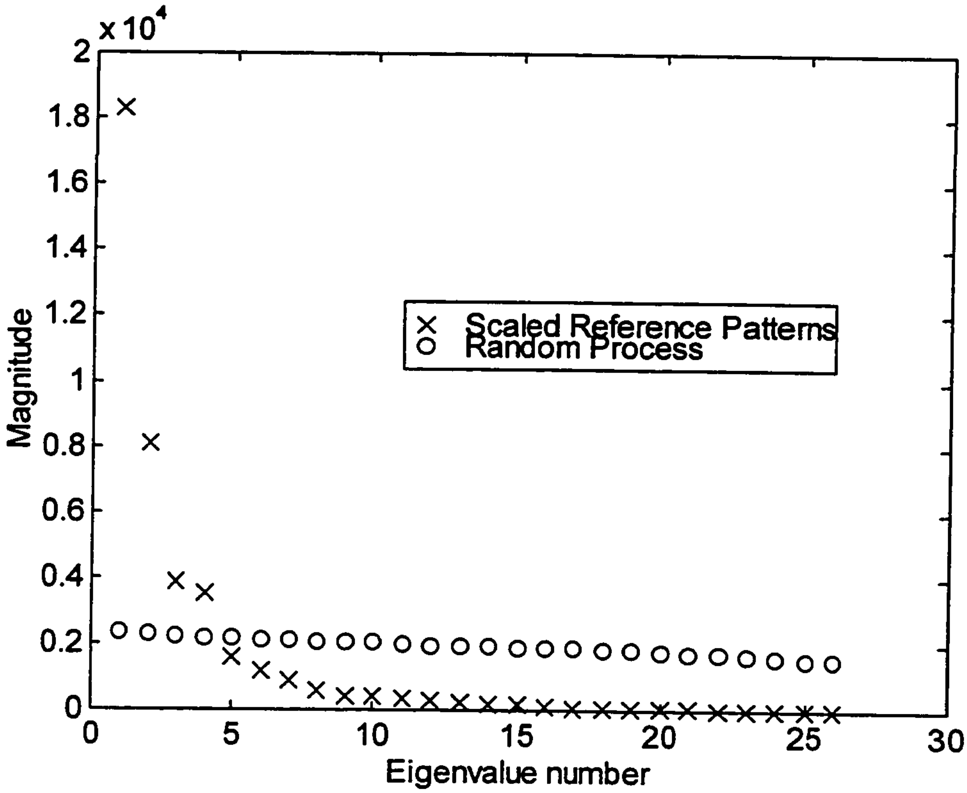


Figure 6.1: Eigenvalues of $X^T X$ and of $Y^T Y$; X contains the scaled reference patterns and Y contains an independent random process.

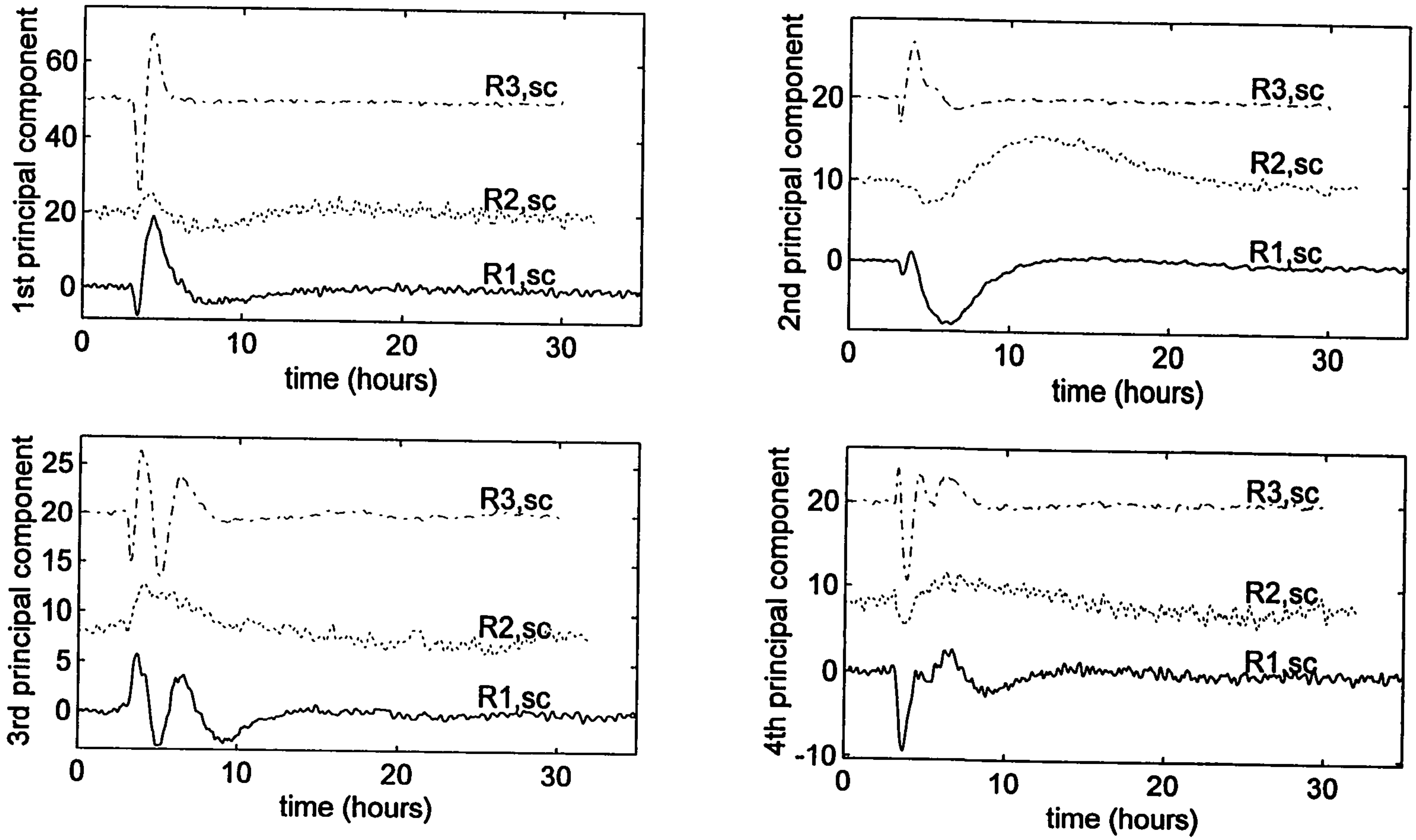


Figure 6.2: Behavior of the three scaled reference patterns, $\mathbf{R}_{1,sc}, \mathbf{R}_{2,sc}, \mathbf{R}_{3,sc}$ along the first four principal components; the initial vaules have been chosen to facilitate plotting.

Table 6.1 shows the results from the similarity assessment step. Again, for each test pattern 6 distances are obtained by comparing the test pattern to the three reference patterns and to the negative of the three reference patterns. The table shows the minimum distance, the ratio of all distances over the minimum distance and the final decision. Good discrimination is achieved when five of the six ratios are significantly greater than one. Again, the shaded cells indicate the correct diagnosis; hence, the diagnosis is correct when the ratio of one appears in a shaded cell.

Comparing results of Table 6.1 with the results of Chapter 4 in Table 4.4, one can see the following:

A) All the classifications, with the exception of pattern $\mathbf{T}_{16}$, were correct. However, in all cases the 5 ratios are greater that the corresponding ratios of Table 4.4. This indicates an increase in the discriminatory power of the classifier. For the pattern $\mathbf{T}_{16}$, all 5 ratios are close to one, indicating that all 6 distances are comparable; this could be used as a warning to indicate that $\mathbf{T}_{16}$ is actually different from all the reference patterns.

B) The improved results can be attributed to the fact that PCA has retained only that variation which is consistent among the physical variables. Inconsistent variation that inflates the distance measure has been ignored. Moreover, one can view the projection of a new pattern onto the principal component space as a very selective "filtering" operation. If the new pattern is a realization of a fault whose description exists in the database (and consequently was included in the construction of the PCA model) its projection allows the consistent variation to pass through and to be used in the DTW stage. This does not happen when the new pattern is a realization of a new fault, since that fault was not considered in the construction of the PCA model. Therefore, even before the similarity assessment step, a "filtering" takes place that favors the correct classification of faults that exist in the reference set.

Table 6.1: Results from Similarity Assessment via DTW; Distances from each DTW match over Minimum Distance and Final Decision.

Test Pattern	Minimum Distance	with $\mathbf{R}_{1,SC}^{PCA}$	with $-\mathbf{R}_{1,SC}^{PCA}$	with $\mathbf{R}_{1,SC}^{PCA}$	with $-\mathbf{R}_{2,SC}^{PCA}$	with $\mathbf{R}_{3,SC}^{PCA}$	with $-\mathbf{R}_{3,SC}^{PCA}$	Decision
$\mathbf{T}_{1,SC}^{PCA}$	0.50	1.00	50.08	26.93	23.35	20.37	21.64	IDV(1)
$\mathbf{T}_{2,SC}^{PCA}$	0.74	1.00	25.91	17.35	27.49	13.85	16.00	IDV(1)
$\mathbf{T}_{3,SC}^{PCA}$	1.12	11.63	9.33	1.00	29.74	13.58	14.08	IDV(2)
$\mathbf{T}_{4,SC}^{PCA}$	2.42	4.75	6.35	8.68	1.00	6.02	5.89	-IDV(2)
$\mathbf{T}_{5,SC}^{PCA}$	0.19	52.63	59.07	85.57	93.48	1.00	69.76	IDV(7)
$\mathbf{T}_{6,SC}^{PCA}$	0.32	33.74	32.47	48.28	47.66	41.25	1.00	-IDV(7)
$\mathbf{T}_{7,SC}^{PCA}$	0.58	1.00	42.48	21.33	20.11	17.53	19.82	IDV(1)
$\mathbf{T}_{8,SC}^{PCA}$	1.81	6.80	6.92	1.00	11.07	7.21	6.90	IDV(2)
$\mathbf{T}_{9,SC}^{PCA}$	2.38	1.00	7.04	4.56	7.55	4.33	5.05	IDV(1)
$\mathbf{T}_{10,SC}^{PCA}$	1.82	1.00	13.60	5.81	6.49	5.57	6.07	IDV(1)
$\mathbf{T}_{11,SC}^{PCA}$	3.26	3.72	2.49	1.00	7.88	4.17	4.07	IDV(2)
$\mathbf{T}_{12,SC}^{PCA}$	3.36	2.04	3.57	4.95	1.00	4.47	4.40	-IDV(2)
$\mathbf{T}_{13,SC}^{PCA}$	5.03	2.46	2.19	2.50	3.18	1.00	1.86	IDV(7)
$\mathbf{T}_{14,SC}^{PCA}$	1.84	13.81	1.00	4.34	7.47	6.64	5.58	-IDV(1)
$\mathbf{T}_{15,SC}^{PCA}$	2.06	5.30	5.88	15.24	1.00	6.87	7.06	-IDV(2)
$\mathbf{T}_{16,SC}^{PCA}$	11.23	1.18	1.37	1.22	1.00	1.39	1.38	-IDV(2)

C) The PCA model is constructed by considering only the instantaneous correlations among the variables. Shifted-in-time versions of the variables were not used, although it is typical in applications of PCA that use dynamic data. In such a case, the PCA model also captures temporal correlations. However, the temporal correlations may change at different production levels and this may result in wrong classifications. In the proposed method, the score vectors are dynamic variables and their patterns are matched via DTW. PCA is used only for feature extraction and not for classification. Section 6.4 discusses this point in more detail.

6.3 Diagnosis of Stochastic Faults

6.3.1 The Algorithm and Design Parameters of the Method

Let $\mathbf{RS}_i$, $i = 1,\ldots,I$ be a the set of reference patterns, each representing a known stochastic fault. Each pattern is a matrix of $r_i \times N$, where r_i is the number of observations and N is the number of measured variables. Also, let $\mathbf{TS}_j$ be the pattern of an unknown stochastic fault, a matrix of dimension $t_j \times N$. The method presented in Chapter 5 is modified as follows.

Feature Extraction Steps: for each variable, in each $\mathbf{RS}_i$, do the following:

Step 1: Subtract the initial level.

Step 2: Filter with high-pass filter.

Step 3: Normalize to standard deviation of one.

Step 4: Filter with low-pass filter.

Let $\mathbf{RS}_{i,SC}$ be the patterns after the above scaling procedure.

Step 5: Create a matrix $\mathbf{X}$ with all the scaled patterns; i.e., $\mathbf{X} = \begin{bmatrix} \mathbf{RS}_{1,sc} \\ \mathbf{RS}_{2,sc} \\ . \\ . \\ \mathbf{RS}_{I,sc} \end{bmatrix}$.

Apply PCA on $\mathbf{X}$ (i.e., $\mathbf{X} = \mathbf{TP}^T + \mathbf{E}$) and store the loading vectors in matrix $\mathbf{P}$. Also, let the corresponding points of the score vectors be the descriptions of the scaled reference patterns in the space of principal components:

$$\mathbf{RS}_{1,sc}^{PCA} = \mathbf{T}(1:r_1,:),\ \mathbf{RS}_{2,sc}^{PCA} = \mathbf{T}(r_1+1:r_1+r_2,:),\ldots,\mathbf{RS}_{I,sc}^{PCA} = \mathbf{T}(\left[\sum_{k=1}^{I-1} r_k\right]+1:\sum_{k=1}^{I} r_k,:)$$

Step 6: Compute the correlation pattern for each $\mathbf{RS}_{i,sc}^{PCA}$, $\mathbf{Corr}[\mathbf{RS}_{i,sc}^{PCA}]$.

When the pattern of the unknown fault, $\mathbf{TS}_j$, is obtained, apply Steps 1-4; let $\mathbf{TS}_{j,sc}$ be the scaled pattern.

Step 7: project $\mathbf{TS}_{j,sc}$ onto the subspace defined by the principal components: $\mathbf{TS}_{j,sc}^{PCA} = \mathbf{TS}_{j,sc}\mathbf{P}$; compute its correlation pattern, $\mathbf{Corr}[\mathbf{TS}_{j,sc}^{PCA}]$.

Similarity Assessment Steps.

Step 8: Apply Dynamic Time Warping between $\mathbf{Corr}[\mathbf{RS}_{i,sc}^{PCA}]$ and $\mathbf{Corr}[\mathbf{TS}_{j,sc}^{PCA}]$ for $i = 1,\ldots,I$.

Step 9: From the I distances obtained in Step 8, find the minimum; the fault whose correlation pattern results in the minimum distance is deemed to be the most likely to have generated the test pattern.

Again, PCA is used to reduce the dimension of the scaled patterns, and consequently the dimension of the correlation patterns. The only additional design parameter of the method is the number of principal components to be retained; any of the criteria mentioned before or their combination can be used. All the other design parameters are as discussed in Subsection 5.3.2.

6.3.2 Case Studies, Results and Conclusions

The case studies of Chapter 5 were used to evaluate the effect of using PCA for reduction of pattern dimension to the DTW-based method presented in Chapter 5. The reference set patterns are shown in Table 5.2 and the test patterns are shown in Table 5.3. However for the case studies of this section all the 26 variables were included in the patterns and not only the 8 variables used in Chapter 5. Also the same filters and the same DTW variant used in Chapter 5 were used in these case studies. The simple Euclidean distance (i.e., $\mathbf{W}$ being the identity matrix) was used as the local distance. Parallel analysis, as described in Subsection 6.2.2, was applied to decide on the number of principal component; it was found again that 4 principal components are significant. Table 6.2 shows the results from the similarity assessment step.

Comparing the above results with the ones of Table 5.5, one can observe the following:

A) All the classifications are correct. However, for some case studies there is large improvement in the discriminatory power (e.g., $\mathbf{TS}_{11}$) with the application of PCA. However, for some patterns the improvement is minimal (e.g., $\mathbf{TS}_2$), while for others the discrimination is actually worse (e.g., $\mathbf{TS}_6$).

B) The results of Chapter 5 indicated that the stochastic faults can produce inaccurate correlation estimates given the amount of data points collected. If this is the case, then the PCA model will also be very dependent on the amount of data collected and will not be an accurate estimate of the "true" model. These two consecutive effects (inaccurate PCA model and inaccurate correlation estimates) could be the reason for the inconsistent results of this section.

Table 6.2: Results from Similarity Assessment via DTW; Distances from each DTW match over Minimum Distance and Final Decision.

Test Pattern	$\hat{D}_{i,MIN}$	$\hat{D}(\mathrm{Corr}[\mathrm{TS}^{PCA}_{i,sc}],\mathrm{Corr}[\mathrm{RS}^{PCA}_{1,sc}])$ over $\hat{D}_{i,MIN}$	$\hat{D}(\mathrm{Corr}[\mathrm{TS}^{PCA}_{i,sc}],\mathrm{Corr}[\mathrm{RS}^{PCA}_{2,sc}])$ over $\hat{D}_{i,MIN}$	Decision
TS$_1$	0.18	1.00	19.33	IDV(8)
TS$_2$	0.44	1.00	8.58	IDV(8)
TS$_3$	0.24	1.00	14.95	IDV(8)
TS$_4$	0.34	1.00	9.79	IDV(8)
TS$_5$	0.24	1.00	16.24	IDV(8)
TS$_6$	0.31	1.00	11.05	IDV(8)
TS$_7$	0.86	1.00	3.73	IDV(8)
TS$_8$	0.66	1.00	5.16	IDV(8)
TS$_9$	0.49	1.00	8.02	IDV(8)
TS$_{10}$	0.86	1.00	4.40	IDV(8)
TS$_{11}$	0.44	9.93	1.00	IDV(13)
TS$_{12}$	0.15	23.93	1.00	IDV(13)
TS$_{13}$	1.06	4.16	1.00	IDV(13)
TS$_{14}$	0.41	8.50	1.00	IDV(13)
TS$_{15}$	0.82	4.71	1.00	IDV(13)
TS$_{16}$	0.24	18.83	1.00	IDV(13)
TS$_{17}$	0.25	14.25	1.00	IDV(13)
TS$_{18}$	1.79	1.24	1.00	IDV(13)
TS$_{19}$	0.75	3.48	1.00	IDV(13)
TS$_{20}$	1.77	1.42	1.00	IDV(13)

C) However, the reduction in the dimension of the correlation patterns should be noted. In Chapter 5, only 8 variables were included (out of the 26) in the patterns of the stochastic faults, resulting in correlation patterns of dimension 36 (8 autocorrelations and 28 crosscorrelations). In the case studies of this section, 26 variables were reduced to 4 principal components, resulting in 4 autocorrelations and 6 crosscorrelations. Moreover, no arbitrary decision had to be made on which variables to include in the patterns.

6.4 PCA as a Pattern Recognition Tool in Dynamic Signals

Recently, Ku et al. (1995) have proposed a Fault Diagnosis procedure for dynamic processes based on PCA. According to their method, one creates a separate PCA model for each reference fault. Because dynamic data exhibit temporal correlations, shifted-in-time versions of variables are used to construct the PCA model. For example, if **X** is a $t \times N$ matrix which contains t measurements on N variables scaled to an average of zero and a standard deviation of one, then one would perform PCA on the following $\mathbf{X}_{\mathbf{LAG}}$ matrix:

$$\mathbf{X}_{\mathbf{LAG}} = [\,\mathbf{X}(l+1{:}t,:) \quad \mathbf{X}(l{:}t-1,:) \quad \ldots \quad \mathbf{X}(1{:}t-l,:)\,] \tag{6.4}$$

Parallel analysis and correlation analysis are used to decide on the amount of time shift (i.e., parameter l), and on the number of principal components. For each model, confidence intervals are constructed for the T^2 and Q statistics. When the pattern of an unknown fault appears it is projected onto the subspaces defined by each PCA model. Next, the T^2 and Q statistics are obtained from each PCA model and are plotted against their corresponding confidence intervals. The PCA model whose confidence intervals include these statistics is selected as the one that best describes the new pattern. The fault that corresponds to this model is deemed as the most likely to have generated the new pattern.

Ku et al. (1995) tested their method on case studies from the Tennessee-Eastman plant. Although they were able to implement it for all 20 programmed faults, their case studies did not examine the effect of different magnitude, direction or operating point on the performance of the method. However, one can see that by including lagged variables in the PCA model a rigid model is constructed; if the same fault occurs with slightly different temporal correlation it will not be diagnosed correctly. Furthermore, because of the nonlinear nature of the plant, even a small difference in the magnitude of a fault can result in different temporal correlation.

To illustrate the above argument, the method of Ku et al. (1995) was applied for the deterministic faults studied in Chapter 4. The raw data of patterns $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$ (see Table 4.2) were used after the initial level was subtracted from each variable. This was done to account for different initial levels that the variables may start at different production levels. Also, only 10 hours of data were considered, with the fault occurring 1 hour after the origin. This was done to be in agreement with the cases studies of Ku et al. (1995). All the 26 variables were used in the patterns and the variables were shifted two sampling intervals (i.e., $l = 2$) as also done by Ku et al. (1995).

After these preprocessing steps, three different PCA models and confidence intervals for the T^2 and Q statistics were constructed. The method was tested using the patterns $\mathbf{T}_2$ and $\mathbf{T}_{10}$ (see Table 4.3). The results from the analysis are shown in Figures 6.3 and 6.4, respectively. As Figure 6.3 shows, the T^2 statistic indicates that $\mathbf{T}_2$ is most similar to $\mathbf{R}_3$, then to $\mathbf{R}_2$ and less to $\mathbf{R}_1$; the Q statistic suggests (although not very clearly) that $\mathbf{T}_2$ is most similar to $\mathbf{R}_1$ than to $\mathbf{R}_2$ or $\mathbf{R}_3$ and this is the correct diagnosis. Thus, the information from the two statistics is contradictory. In the case of pattern $\mathbf{T}_{10}$ (which is a realization of the same fault as $\mathbf{R}_1$, with 90% magnitude at a reduced production operating point) the T^2 statistic in Figure 6.4 suggests that $\mathbf{T}_{10}$ can be any of the three reference faults. However, the Q statistic suggests that $\mathbf{T}_{10}$ does not resemble any of the reference faults. Again, the two statistics give contradictory information.

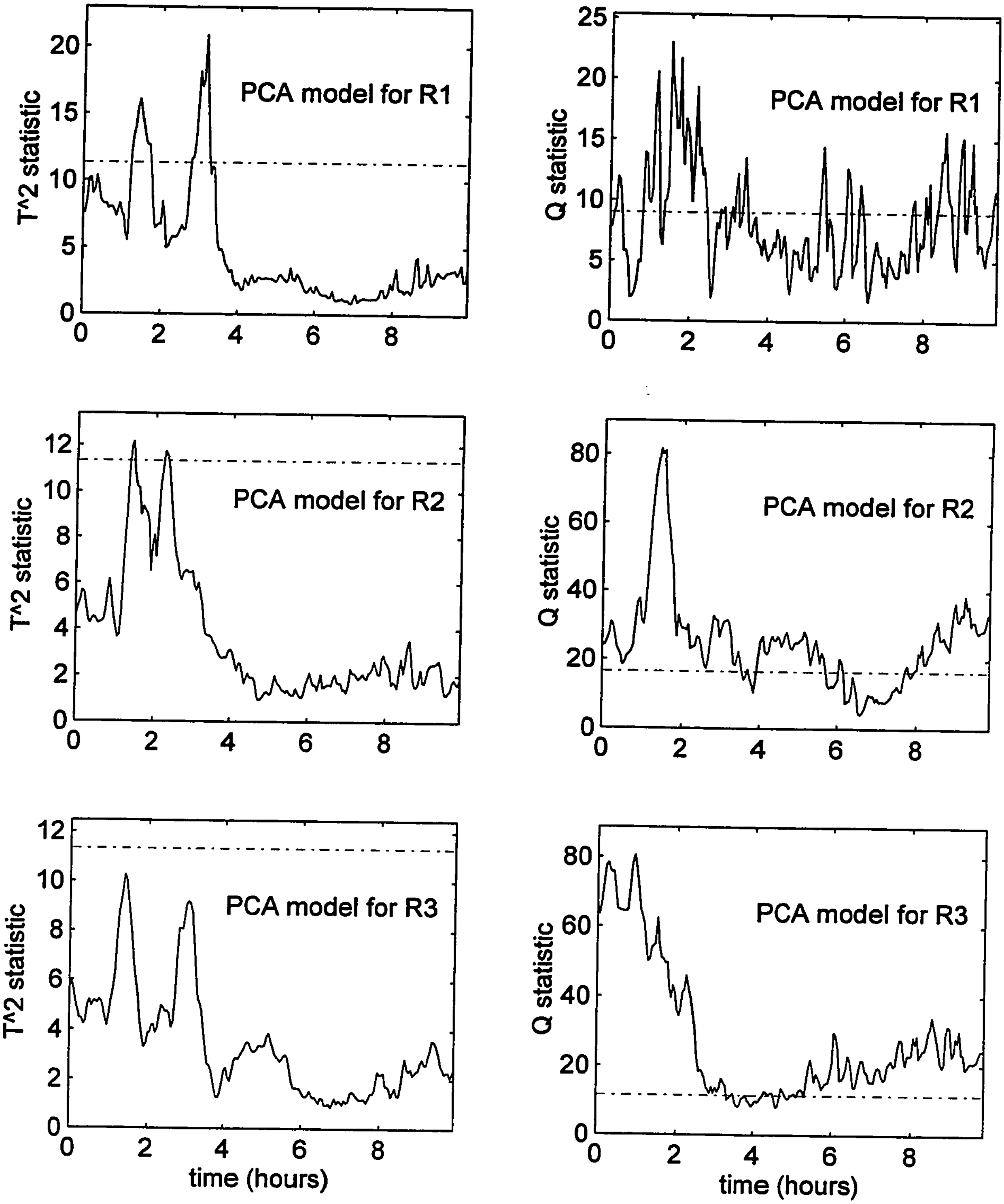


Figure 6.3: Projection of the $\mathbf{T}_2$ pattern onto the PCA models of patterns $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$; the graphs show the T^2 and Q statistics (solid lines) and their 95% confidence intervals (dashed lines).

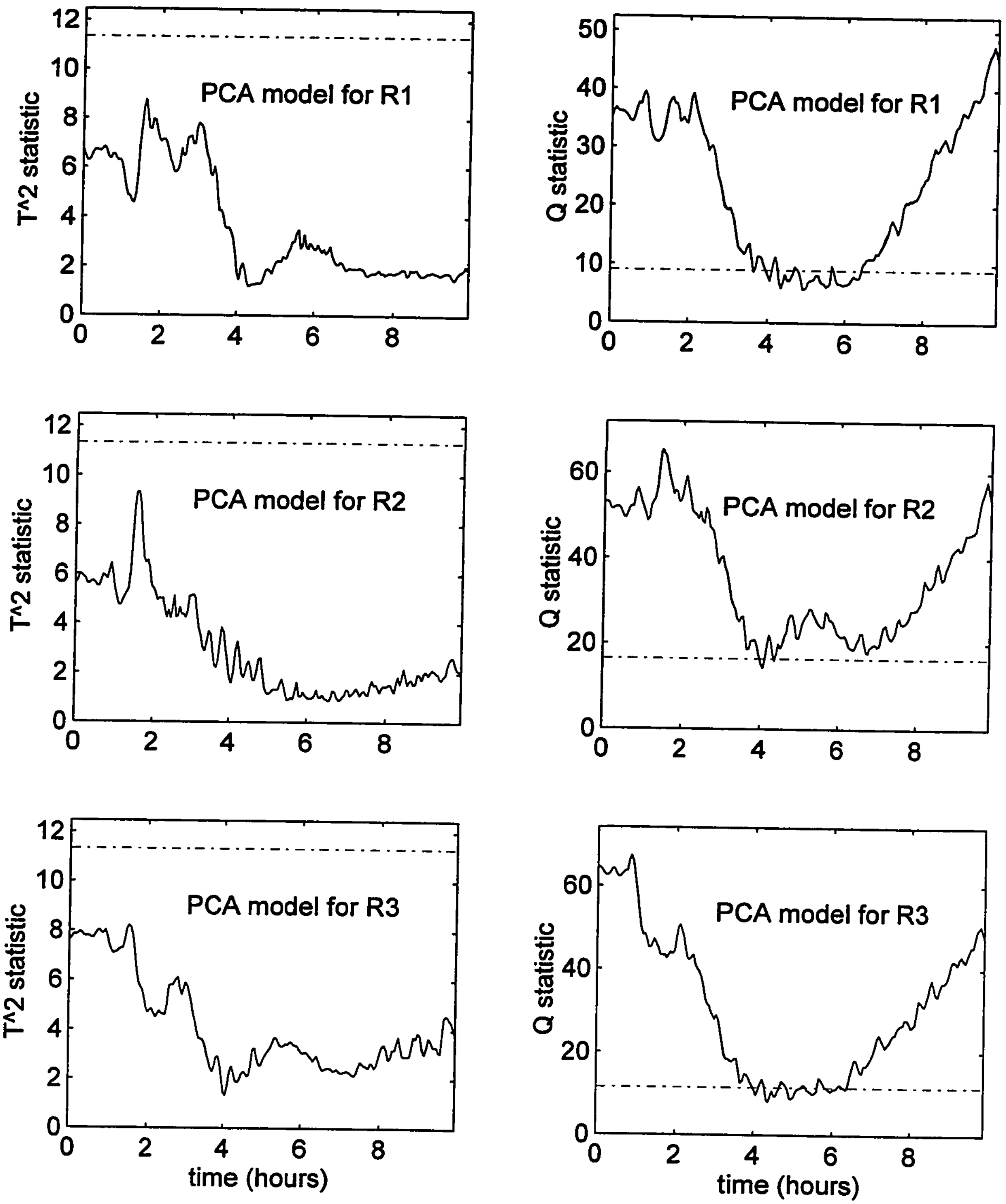


Figure 6.4: Projection of the $\mathbf{T}_{10}$ pattern onto the PCA models of patterns $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$; the graphs show the T^2 and Q statistics (solid lines) and their 95% confidence intervals (dashed lines).

These results are not surprising. In the "Reduced Production" operating point, the temporal correlations change. Any Fault Diagnosis scheme that is not flexible to account for these changes will fail to provide the correct diagnosis. As discussed in Section 2.3, to use PCA for diagnosis of dynamic patterns, a unrealistically rich database of past fault realizations would be required. One then would have to either create separate models for each realization of each fault or create one model for each fault from all realizations. On the other hand, the methods proposed in Chapter 4 and in this chapter do not require such a rich database due to the flexible pattern matching capability of Dynamic Time Warping.

6.5 Summary and Conclusions

This chapter presented a modification to the similarity assessment scheme which utilizes Principal Component Analysis. The modified methods were applied to the same case studies of Chapter 4 and 5. PCA was used as an additional feature extraction step that reduces greatly the dimensions of the patterns and extracts consistent information. In the case of deterministic faults, 26 variables were reduced to 4 principal components and a large improvement in the discriminatory power of the classifier was achieved. In the case of stochastic faults, the classification results were not always improved; a possible reason for that could be the lack of a sufficient amount of data to accurately estimate the auto- and crosscorrelations. However, the reduction in the dimension of the pattern is a major improvement due to the geometric increase of correlations with respect to the number of variables.

It has to be emphasized that PCA was used in this chapter only as a feature extraction tool and not as a diagnostic tool. There are two reasons for this: I) different realizations of the same fault will result in similar correlations across the variables, and II) different production levels will result in different temporal correlations for the same fault. For these reasons, PCA was performed by considering only the instantaneous correlations across variables (i.e., at lag zero) so that any temporal correlation is ignored. Thus, PCA will extract information which remain consistent over different realizations of the same fault and Dynamic Time Warping will account for any distorted temporal correlations. On the other hand, a PCA model which includes temporal correlations will be dependent on the operating point and this is not a desired characteristic of a robust Fault Diagnosis scheme. The results of Section 6.4 showed that a diagnostic scheme for a dynamic process based on time-shifted PCA is likely to fail to give the correct answer even for small variations of the training set faults.

CHAPTER 7

ON-LINE DIAGNOSIS OF DETERMINISTIC FAULTS IN CONTINUOUS DYNAMIC PROCESSES

In this chapter, the method presented in Chapter 4 for the off-line diagnosis of deterministic faults will be modified so that it can be applied to diagnose deterministic faults in real time. The application of the method is illustrated with case studies from the Tennessee-Eastman simulation.

7.1 Introduction

Chapter 4 presented a method to off-line diagnose deterministic faults; i.e., faults that produce similar patterns in the process variables for different realizations of the fault. The method was a sequence of two steps: the first was a feature extraction step, where high-pass filtering and variance normalization removed the magnitude information from the variables; the second step was a similarity assessment scheme using a symmetric Dynamic Time Warping (DTW) algorithm. As mentioned in Chapter 4, the high-pass filtering and variance normalization in the feature extraction step are critical because deterministic events can occur with different magnitude and a robust Fault Diagnosis scheme should classify them independently of their magnitude. DTW is used because it is a flexible pattern matching method that can deal very effectively with similar but unsynchronized patterns. However, DTW is a distance-based method and this means that an intelligent scaling of the variables is critical for the successful operation of the classifier.

Now, when one is faced with the problem of diagnosis of deterministic faults in a real time situation there are some important differences from the off-line application that have to be taken into consideration. These are the following:

I) When the test pattern of an unknown fault is evolving, one has only an initial part of the pattern; the rest of the pattern is not yet available. Moreover, there may be an uncertainty in locating the origin of the fault in the test pattern. Also, the time required for the plant to achieve steady state is unknown.

II) Because the classification has to be independent of the magnitude of the fault, the unknown test pattern has to be scaled using similar techniques to the off-line scaling. High-pass filtering is independent of the number of points and therefore does not create any concerns. However the normalization of each variable to a standard deviation of one is a problem. Even after high-pass filtering is applied the variables are not stationary, as Figure 4.4 shows. For example, by looking at Figure 4.4 one can see that the standard deviation of each variable is greater when it is estimated from the first half of the patterns that when all the data points are used. This is due to the fact that the latter part of the patterns corresponds to steady-state conditions which have small variance. Therefore, depending on how many data points are available at any time, the normalization factors will be different for each variable. Suppose that i) one uses all the data points to estimate the standard deviation and normalize the variables in the reference patterns and then ii) uses these scaling factors to normalize the variables in the evolving test pattern. If the reference and the test patterns are realizations of the same fault occurring with the same magnitude, this is a consistent scaling. If however the test pattern is a fault realization with different magnitude, the scaling factors are inconsistent; on-line pattern matching using DTW will fail due to incorrect scaling.

III) Assuming that a proper scaling has been applied, one now has to compare the test pattern with each of the reference patterns and their mirror images. Since the test pattern is still evolving, it is reasonable to compare it not with the complete reference patterns but with their initial part. If indeed the test pattern is a realization of a reference fault, then a DTW algorithm capable of locating the initial part of the appropriate reference pattern should be used.

The next section presents a DTW algorithm that can deal with the problems (I) and (III), namely the uncertainty in locating the time origin of the test pattern and the requirement of locating the initial part of the appropriate reference pattern that most resembles the test pattern. Problem (II) will be discussed in Section 7.3, where the complete method for on-line diagnosis of deterministic faults will be presented.

7.2 A DTW Algorithm for On-line Application

Let **T** be the evolving test pattern up to the current time, a matrix of dimension $t \times N$, where t is the number of data points and N the number of measured variables. Let **R** be any of the complete reference patterns, a matrix of dimension $r \times N$, with r being the number of data points. For both patterns it is assumed that an appropriate scaling procedure has been applied (to be presented in the next section).

The objective is to compare **T** with **R** using DTW in a way that accounts for the following timing discrepancies between the two patterns:

I) The time origin of the fault that generates the **T** pattern may not be known exactly. Therefore, it is possible that too many or too few data points (that correspond to the steady-state conditions before the fault) are included in **T** than are included in **R**. The DTW algorithm should locate the point in the initial part of **R** that best matches the first point of **T**.

II) Assume that **T** is being generated by the same fault that generated **R**. Since **R** is the complete description of the fault (i.e., until the new steady state is reached) and **T** is a pattern that is still evolving, **T** will best match with an initial part of **R**. The DTW algorithm should be able to find the part of **R** which mostly resembles pattern **T** at the current time.

An asymmetric DTW algorithm similar to the one presented in Example 6 of Subsection 3.4.1, can be used to account for the above timing considerations. Figure 7.1 illustrates the features of the proposed algorithm, namely the local and global constraints and the relaxed endpoint constraints. Let $d(\mathrm{i},\mathrm{j})$ be the local distance between the i^{th} and the j^{th} point of **T** and **R** respectively; also assume the identity matrix is used as the weight matrix; i.e.,

$$d(\mathrm{i},\mathrm{j}) \;=\; (\mathbf{T}(\mathrm{i},:)-\mathbf{R}(\mathrm{j},:))(\mathbf{T}(\mathrm{i},:)-\mathbf{R}(\mathrm{j},:))^{\mathrm{T}} \tag{7.1}$$

Also, let $\mathbf{D}_{\mathbf{A}}(\mathrm{i},\mathrm{j})$ be the minimum accumulated distance up to point (i,j). Then at every allowable (i,j) point the following minimization problem is solved:

$$\mathbf{D}_{\mathbf{A}}(\mathrm{i},\mathrm{j}) = \min\begin{Bmatrix} \mathbf{D}_{\mathbf{A}}(\mathrm{i}-1,\mathrm{j}) \quad + d(\mathrm{i},\mathrm{j}) \quad \text{or} \quad \infty \text{ if condition (A)} \\ \mathbf{D}_{\mathbf{A}}(\mathrm{i}-1,\mathrm{j}-1) + d(\mathrm{i},\mathrm{j}) \\ \mathbf{D}_{\mathbf{A}}(\mathrm{i}-1,\mathrm{j}-2) + d(\mathrm{i},\mathrm{j}) \\ \mathbf{D}_{\mathbf{A}}(\mathrm{i}-1,\mathrm{j}-3) + d(\mathrm{i},\mathrm{j}) \end{Bmatrix} \tag{7.2}$$

where: Condition (A): point $(\mathrm{i}-1,\mathrm{j})$ is optimally reached from point $(\mathrm{i}-3,\mathrm{j})$ via two consecutive horizontal moves.

This local constraint is an extension of the Itakura local constraint (Itakura, 1975) presented in Example 2 of Subsection 3.3.2, and extends the range for the local slope of the optimal path from $[1/2,2]$ to $[1/3,3]$.

Also, for the first δ_1+1 and last δ_1 points of the pattern **T**, the check on three consecutive horizontal transitions is disengaged, thus allowing for up to δ_1 consecutive horizontal local optimal transitions. This accounts for the possibility that too many points from either before or after the fault are included in **T**. Thus, the local constraints become:

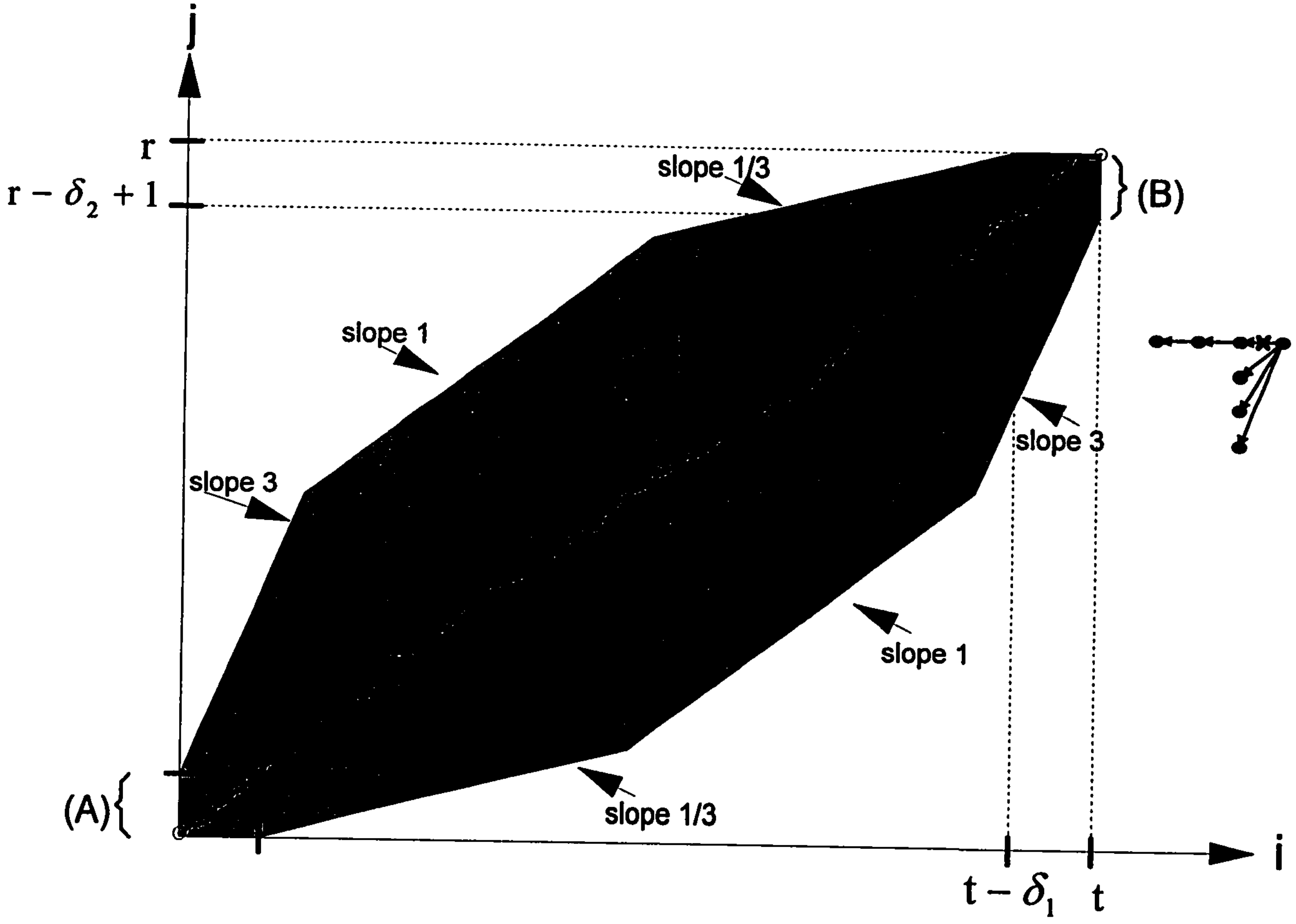


Figure 7.1: Global and local constraints of the DTW variant used for on-line diagnosis of deterministic faults. Areas (A) and (B) are the allowable ranges for the first and last optimal path points respectively. The shaded area is the allowable search area for the optimal path. The circles indicate the first and the last point of the optimal path as found by DTW.

$$\mathbf{D_A}(i,j) = \min\begin{Bmatrix} \mathbf{D_A}(i-1,j) \quad + d(i,j) \\ \mathbf{D_A}(i-1,j-1) + d(i,j) \\ \mathbf{D_A}(i-1,j-2) + d(i,j) \\ \mathbf{D_A}(i-1,j-3) + d(i,j) \end{Bmatrix} \text{ if } 1 \le i \le \delta_1 + 1\text{, or, } t - \delta_1 + 1 \le i \le t \qquad (7.3)$$

Similarly, the optimal path is allowed to start and finish at the first and last δ_2 points on the vertical axis; Areas (A) and (B) respectively in Figure 7.1. This feature accounts for the possibility that too few points from either before or after the fault may be in the **T** pattern. It also accounts for the fact that the pattern **T** may be most similar to an interior part of the pattern **R**.

The local constraints of Eqs (7.2) and (7.3) result in a search area for the optimal path that is defined by lines of slope $1/3$ and 3 emanating from the appropriate points in the grid, as Figure 7.1 illustrates. To reduce even more the search area, an extended band constraint can also be used as follows. One can draw two diagonal lines with a slope of one, one line emanating from point $(1,1)$ and the other from point (t,r); then the optimal path is not allowed to deviate by more than $\pm M$ points from the least restrictive of these diagonal lines. This extended band constraint results in a band width of $2M + |t-r| + 1$. When $t = r$, then it is reduced to the simple band constraint used in all DTW algorithms of Chapters 4, 5 and 6. The final search area is obtained as the intersection of the area defined by the lines of slope $1/3$ and 3 and the area defined by the extended band constraint. The result is the shaded area of Figure 7.1.

The above DTW algorithm is an asymmetric algorithm and as such it may skip points of pattern **R**. This is desirable in the beginning and in the end of **R**, since **T** may be similar to an interior part of **R**. On the other hand, one could argue that omitting points is not desirable if it happens in this interior part of **R** because the net result is that some points in **R** are treated preferentially over others (i.e., the ones that the algorithm skips). Thus, a symmetric DTW algorithm should be used instead of an asymmetric algorithm. However, as discussed in Example 7 of Subection 3.4.2, a symmetric DTW algorithm makes is impossible to relax the fixed initial-point constraint on the optimal path. This is clearly a major disadvantage for an on-line application and for that reason an asymmetric DTW algorithm was chosen instead of a symmetric one.

This completes the description of the DTW algorithm used in the on-line application of the proposed Fault Diagnosis scheme. The complete scheme (feature extraction and similarity assessment) is presented in the next section.

7.3 The Algorithm and Design Parameters of the Method

Let $\mathbf{R}_i$, $i = 1,\ldots,I$ be a the set of reference patterns, each representing a known fault. Each is a matrix of $r_i \times N$, where r_i is the number of raw measurements, and N is the number of measured variables. Also, let $\mathbf{T}$ be the raw measurements of an unknown test pattern up to the current time, a matrix of dimension $t \times N$; the objective is to find the $\mathbf{R}_i$ pattern whose initial part is most similar to $\mathbf{T}$. The proposed method is as follows:

Feature Extraction Steps: for each variable, in each $\mathbf{R}_i$ do the following:

Step 1: Subtract the initial level.

Step 2: Filter with high-pass filter.

Step 3: Select a set of times: t_1, t_2,...,t_L same for all variables and for all reference patterns. For each variable, in each $\mathbf{R}_i$, estimate a set of L standard deviations by using only the data from time 0 up to each of the times t_1, t_2,...,t_L.

When the current time is equal to any of the times t_1, t_2,...,t_L apply Steps 4 to 12 as follows:

Step 4: Divide each variable in each $\mathbf{R}_i$ by the its standard deviation as estimated using data from time 0 up to current time.

Step 5: Filter with low-pass filter.

Let $\mathbf{R}_{i,SC}$ be the reference patterns after the above scaling procedure.

Let **T** be the test pattern from time 0 up to current time; for each variable in **T** apply the following procedure:

Step 6: Subtract the initial level.

Step 7: Filter with high-pass filter.

Step 8: Normalize to standard deviation of one using all data in **T**.

Step 9: Filter with low-pass filter.

Let $\mathbf{T}_{\mathrm{SC}}$ be the resulting scaled test pattern.

Similarity Assessment Steps.

Step 10: Apply Dynamic Time Warping (using the algorithm described in Section 7.2) between $\mathbf{R}_{\mathrm{i,SC}}$ and $\mathbf{T}_{\mathrm{SC}}$. To speed up the computations, instead of the complete $\mathbf{R}_{\mathrm{i,SC}}$ pattern, a part of it could be used to be compared with $\mathbf{T}_{\mathrm{SC}}$. For example, at time t_1, the part of $\mathbf{R}_{\mathrm{i,SC}}$ from time 0 up to time $t_1 + \Delta t_1$ could be used; Δt_1 should be selected sufficient large to account for uncertainties in locating the time origin of the fault in both patterns.

Step 11: Apply Dynamic Time Warping between $-\mathbf{R}_{\mathrm{i,SC}}$ and $\mathbf{T}_{\mathrm{SC}}$ (i.e., the mirror images of the scaled reference patterns). The same argument used in Step 10 implies here to speed up the computations.

Step 12: From the $2 \cdot \mathrm{I}$ distances obtained in Steps 10 and 11, find the minimum; the fault whose pattern results in the minimum distance is deemed to be the most likely to have generated the test pattern.

The method is similar to the one described in Chapter 4 with only two differences: the proposed DTW algorithm and the selection of a set of times where scaling and pattern comparison takes place. The DTW algorithm used in this chapter offers the flexibility of relaxing the endpoint constraints. The need to select a set of different times where variance normalization and DTW is performed, arises from the requirement of independence of the diagnosis to the magnitude of the fault and by the nonstationary nature of the variables in the patterns.

The reference patterns describe fault realizations of a particular magnitude and duration. However, new fault realizations can occur with different magnitudes; moreover, in real time their duration is not constant but is increasing. The standard deviation estimated from data up to time t_1 will be different from the one estimated using data up to time t_2 because the variables are nonstationary, even after high-pass filtering. The proposed segmentation and scaling tries to account for this nonstationary behavior of the variables. It ensures that the estimation of the standard deviation of the variables will be based on data that occupy approximately the same initial part of the patterns. This would be exactly true if the patterns were synchronized, that is, the fault always occurs at the same time after time zero. However this will not be true in general and the method should be robust to this discrepancy.

The main design parameter of the method is the set of times t_1, t_2,...,t_L over which scaling and pattern comparison take place. One could choose them based on the dynamics of the process. Most of the variation induced by a fault occurs in the initial part of the pattern rather than in the final part (i.e., when the plant approaches a new steady state). For that reason, the difference between the two estimates of the standard deviation will be larger if t_1 and t_2 are close to the time origin of the fault than if t_1 and t_2 are close to the end of the pattern. Therefore it is suggested to space these times unevenly, placing more of them in the initial part of the pattern and fewer towards the end.

The parameters in the DTW algorithm are essentially related to the uncertainty in locating the time origin of the fault either in the reference patterns or in the test pattern. The same implies for the $\Delta t_1, \Delta t_2, ..., \Delta t_L$ parameters mentioned in Steps 10 and 11. The weight matrix, **W**, for the local distance estimation can be set equal to the identity matrix, unless there is some prior knowledge about the importance of a variable in the fault diagnosis. The type of high-pass filters to be used is another major factor. Again, first order high-pass filters are suggested since they do not induce oscillations in the patterns (see also Section 4.3 for more details).

This completes the presentation of the method proposed for on-line diagnosis of deterministic faults in continuous dynamic processes. In the next section, cases studies from the Tennessee-Eastman plant are used to illustrate its implementation.

7.4 Case Studies

7.4.1 Description of the Patterns in the Reference and Test Sets

The case studies of Chapter 4 are also used in this chapter. However, in the on-line application, the synchronization of the variables is crucial for the estimation of the standard deviation of the variables. To see the effect of having unsynchronized patterns in the on-line diagnosis, the time characteristics of the case studies of Chapter 4 were slightly modified. Table 7.1 shows the reference set used in the case studies of this chapter. The test patterns evaluated are shown in Table 7.2. From the two tables one can see that timing differences up to 1.5 hrs were introduced in the patterns. Again, faults with different magnitude and direction, occurring at different production levels are used to test the on-line method against the requirements imposed in Chapter 1 for a Fault Diagnosis scheme. More details on the faults, the operating points, and the simulation can be found in Chapter 4, Sections 4.1 and 4.4.1.

Table 7.1: Patterns in the Reference Set

Pattern	Fault	Operating Point	Step size / direction	Duration / # of points	Fault occurs after / at
R_1	IDV(1)	Nominal	+1.0	28.0 hrs / 561 pts	1.0 hrs / 21st pt
R_2	IDV(2)	Nominal	+1.0	29.0 hrs / 581 pts	2.0 hrs / 41st pt
R_3	IDV(7)	Nominal	+1.0	28.5 hrs / 571 pts	1.5 hrs / 31st pt

Table 7.2: Test Set Patterns Evaluated

Pattern	Fault	Operating Point	Step size / direction	Duration / # of points	Fault occurs after / at
T_1	IDV(1)	Nominal	+1.0	29.0 hrs / 581 pts	2.0 hrs / 41st pt
T_2	IDV(1)	Nominal	+0.7	28.0 hrs / 561 pts	1.0 hrs / 21st pt
T_3	IDV(2)	Nominal	+0.8	27.5 hrs / 551 pts	0.5 hrs / 11st pt
T_4	IDV(2)	Nominal	-0.9	28.0 hrs / 561 pts	1.0 hrs / 21st pt
T_5	IDV(7)	Nominal	+0.8	29.0 hrs / 581 pts	2.0 hrs / 41st pt
T_6	IDV(7)	Nominal	-0.7	28.0 hrs / 561 pts	1.0 hrs / 21st pt

T_7	IDV(1)
	IDV(9)
		Nominal	+0.9
			+1.0
				27.5 hrs / 551 pts	0.5 hrs / 11st pt
					8.5 hrs / 171st pt

T_8	IDV(2)
	IDV(10)
		Nominal	+0.8
			+1.0
				28.0 hrs / 561 pts	1.0 hrs / 21st pt
					9.0 hrs / 181st pt

T_9	IDV(1)	Reduc. Prod.	+0.5	28.0 hrs / 561 pts	1.0 hrs / 21st pt
T_{10}	IDV(1)	Reduc. Prod.	+0.9	28.5 hrs / 571 pts	1.5 hrs / 31st pt
T_{11}	IDV(2)	Reduc. Prod.	+0.9	28.0 hrs / 561 pts	1.0 hrs / 21st pt
T_{12}	IDV(2)	Reduc. Prod.	-0.8	28.5 hrs / 571 pts	1.5 hrs / 31st pt
T_{13}	IDV(7)	Reduc. Prod.	+0.7	28.0 hrs / 561 pts	2.0 hrs / 41st pt
T_{14}	IDV(1)	Nominal	-0.5	28.5 hrs / 571 pts	1.5 hrs / 31st pt
T_{15}	IDV(2)	Nominal	-0.5	28.0 hrs / 561 pts	1.0 hrs / 21st pt
T_{16}	IDV(17)	Nominal	+1.0	23.0 hrs / 461 pts	1.0 hrs / 21st pt

7.4.2 Selection of Design Parameters

The filters designed in Subsection 4.4.2 of Chapter 4 will also be used for the case studies of this chapter. The simple Euclidean distance (i.e., **W** being the identity matrix) will be used as the local distance. All the 26 variables shown in Table I.1 of the Appendix will be included in the patterns; they are recorded every 3 minutes.

The asymmetric DTW algorithm described in Section 7.2 will be used for the similarity assessment step. The times (after time 0) where scaling and similarity assessment are performed are selected to be: $t_1 = 2.5$ hrs, $t_2 = 5.0$ hrs, $t_3 = 7.5$ hrs, $t_4 = 10$ hrs, $t_5 = 15$ hrs, $t_6 = 22.5$ hrs for each test pattern. At the end of the test pattern (referred to as time t_7), the whole test pattern is compared with the complete reference patterns (as they are shown in Tables 7.1 and 7.2). For each of these times, values for the parameters of the DTW algorithm are given in Table 7.3.

Table 7.3: Parameters of the DTW Algorithm at Each of the Selected Times

	$\Delta t_i, i = 1,\ldots,6$ (hrs)/ no. of pts	δ_1 (no. of pts)	δ_2 (no. of pts)	M (no. of pts)
$t_1 = 2.5$ hrs (51 pts)	1.5 hrs 30 pts	20 pts	20 pts	25 pts
$t_2 = 5.0$ hrs (101 pts)	1.5 hrs 30 pts	30 pts	30 pts	35 pts
$t_3 = 7.5$ hrs (151 pts)	1.5 hrs 30 pts	40 pts	40 pts	45 pts
$t_4 = 10$ hrs (201 pts)	1.5 hrs 30 pts	50 pts	50 pts	55 pts
$t_5 = 15$ hrs (301 pts)	1.5 hrs 30 pts	60 pts	60 pts	65 pts
$t_6 = 22.5$ hrs (450 pts)	1.5 hrs 30 pts	70 pts	70 pts	75 pts
t_7 is the end time of each test pattern	-	80 pts	80 pts	85 pts

7.4.3 Results

The results of the method for the first two times (i.e., t_1 and t_2) are presented in Tables 7.4 and 7.5, respectively. At each time and for each test pattern, six minimum normalized total distances are obtained from the DTW step. The tables show the minimum of the six distances and the ratio of the other five distances to the minimum. The fault that corresponds to the reference pattern that gives the minimum distance (i.e., a ratio of one) is selected as the most probable cause of the test pattern. Again, good discrimination among faults is obtained when the five ratios are not close to one. This is done at each of the seven selected times. The shaded cells indicate the correct fault; hence, the diagnosis is correct when the ratio of one appears in a shaded cell.

Figures 7.2 and 7.3 show the DTW results for two test patterns, $\mathbf{T_2}$ and $\mathbf{T_{12}}$. Each subplot shows the 7 optimal paths (for the 7 selected times). The dotted lines indicate the upper and lower limits for the optimal path at the final time t_7. The numbers in each subplot show, for the first three of the selected times, the ratio of the 6 distances over the minimum of the six.

Examining the results presented at the two tables one can observe the following:

A) At the first time, $t_1 = 2.5$ hrs, the results of Table 7.4 indicate that six test patterns were diagnosed incorrectly. This number includes pattern $\mathbf{T_{16}}$ which, as mentioned in Chapter 4, represents a fault not present in the database of the reference patterns. The other five test patterns that are misdiagnosed (i.e., $\mathbf{T_1}$, $\mathbf{T_5}$, $\mathbf{T_{10}}$, $\mathbf{T_{13}}$ and $\mathbf{T_{14}}$) are realization of faults that exist in the database, however they are not synchronized with the reference patterns.

Table 7.4: Results from Similarity Assessment via DTW at t_1 = 2.5 hrs ; Minimum Distance, Distances from each DTW match over Minimum Distance, and Final Decision.

Test Pattern	Minimum Distance	with $\mathbf{R_1}$	with $-\mathbf{R_1}$	with $\mathbf{R_2}$	with $-\mathbf{R_2}$	with $\mathbf{R_3}$	with $-\mathbf{R_3}$	Decision
$\mathbf{T_1}$	19.32	1.91	1.39	1.91	1.50	1.51	1.00	-IDV(3)
$\mathbf{T_2}$	5.54	1.00	12.55	7.42	7.95	2.47	5.94	IDV(1)
$\mathbf{T_3}$	25.73	1.63	1.66	1.00	2.18	1.62	1.11	IDV(2)
$\mathbf{T_4}$	22.34	1.91	1.41	2.31	1.00	1.06	1.53	-IDV(2)
$\mathbf{T_5}$	15.95	3.01	1.05	2.46	1.73	2.10	1.00	-IDV(7)
$\mathbf{T_6}$	2.91	13.86	7.55	13.39	12.35	11.33	1.00	-IDV(7)
$\mathbf{T_7}$	5.17	1.00	14.04	9.56	9.80	3.06	5.44	IDV(1)
$\mathbf{T_8}$	20.93	1.22	2.36	1.00	2.42	1.28	1.27	IDV(2)
$\mathbf{T_9}$	14.19	1.00	4.48	2.42	3.49	1.25	2.54	IDV(1)
$\mathbf{T_{10}}$	14.12	2.26	2.68	2.06	3.20	1.00	2.43	IDV(3)
$\mathbf{T_{11}}$	21.97	1.28	1.93	1.00	2.13	1.33	1.13	IDV(2)
$\mathbf{T_{12}}$	21.92	1.75	1.15	1.88	1.00	1.19	1.05	-IDV(2)
$\mathbf{T_{13}}$	19.12	2.34	1.08	1.86	1.70	1.48	1.00	-IDV(7)
$\mathbf{T_{14}}$	13.15	2.92	1.98	2.76	2.84	2.81	1.00	-IDV(7)
$\mathbf{T_{15}}$	17.44	2.03	1.80	2.65	1.00	1.58	1.42	-IDV(2)
$\mathbf{T_{16}}$	22.03	1.30	1.51	1.01	1.54	1.24	1.00	-IDV(7)

Table 7.5: Results from Similarity Assessment via DTW at $t_1 = 5$ hrs; Minimum Distance, Distances from each DTW match over Minimum Distance, and Final Decision.

Test Pattern	Minimum Distance	with R_1	with $-R_1$	with R_2	with $-R_2$	with R_3	with $-R_3$	Decision
T_1	1.87	1.00	20.03	12.53	13.01	7.60	12.59	IDV(1)
T_2	1.41	1.00	32.84	17.12	25.08	12.71	16.95	IDV(1)
T_3	7.36	4.03	4.36	1.00	5.78	3.77	4.22	IDV(2)
T_4	6.62	5.07	3.13	5.61	1.00	3.98	3.59	-IDV(2)
T_5	0.62	27.20	25.54	45.07	39.16	1.00	31.88	IDV(7)
T_6	0.77	22.81	25.40	36.11	32.61	26.00	1.00	-IDV(7)
T_7	0.97	1.00	52.11	25.19	47.50	18.76	24.60	IDV(1)
T_8	6.09	3.19	6.11	1.00	6.65	4.01	4.20	IDV(2)
T_9	9.95	1.00	4.82	2.60	4.57	2.45	2.97	IDV(1)
T_{10}	8.15	1.00	5.43	3.23	4.86	2.66	3.24	IDV(1)
T_{11}	8.95	2.43	3.31	1.00	3.97	2.61	2.84	IDV(2)
T_{12}	10.34	2.99	2.13	3.49	1.00	2.36	2.18	-IDV(2)
T_{13}	13.70	1.62	1.91	2.58	2.12	1.00	2.02	IDV(7)
T_{14}	2.90	13.28	1.00	10.00	8.30	7.89	5.38	-IDV(1)
T_{15}	7.98	3.98	2.74	4.66	1.00	3.07	2.74	-IDV(2)
T_{16}	16.66	1.70	1.10	1.80	1.00	1.51	1.27	-IDV(2)

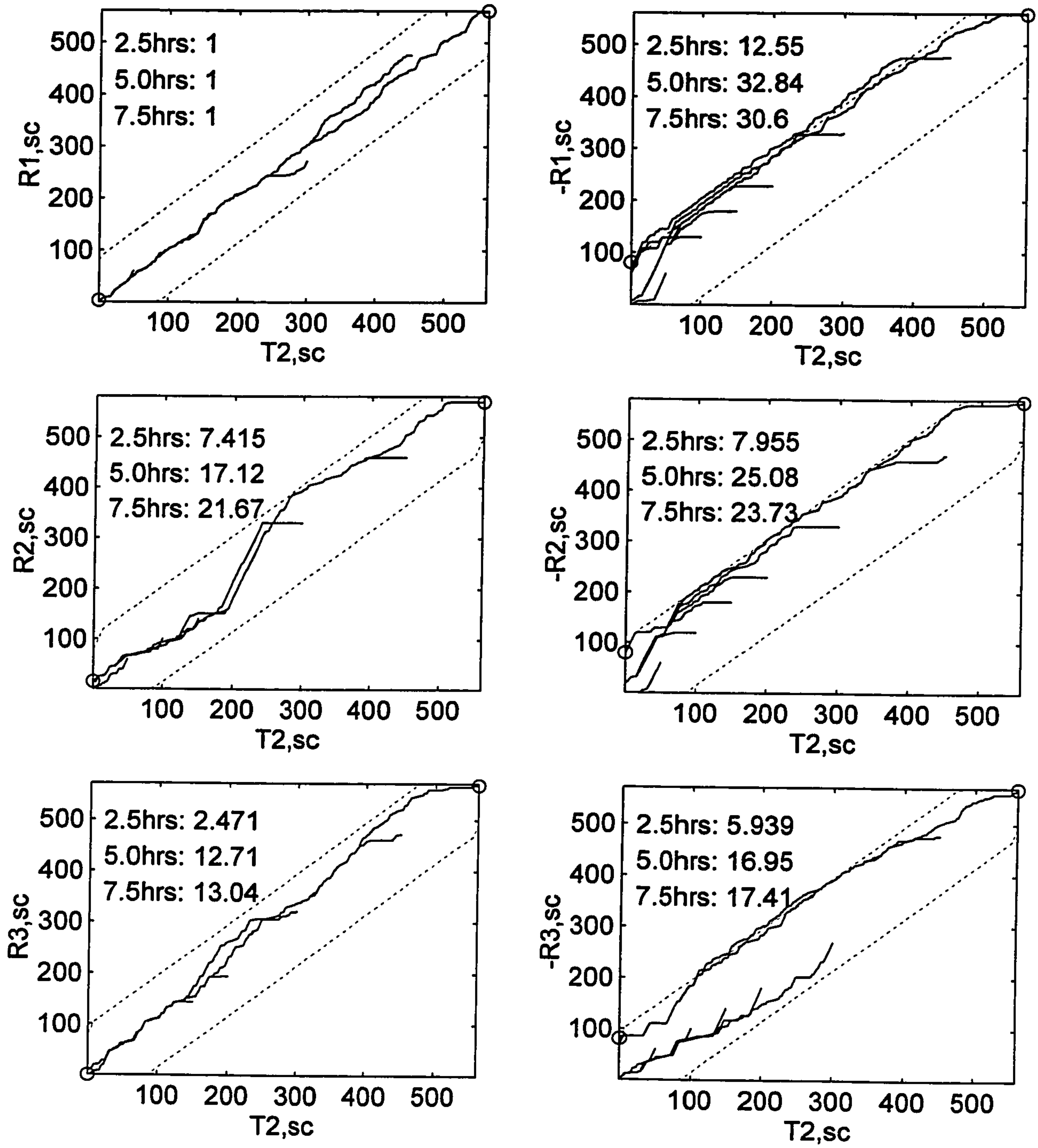


Figure 7.2: Results from the on-line diagnosis for test pattern T_2; each subplot shows the 7 optimal paths (for the 7 selected times); the dotted lines indicate the upper and lower limits for the optimal path at the final time. The numbers show, for each of the first 3 times, the ratio of the 6 distances over the minimum of the six. At each time, the reference pattern that gives a ratio of one is deemed as most similar to the test pattern.

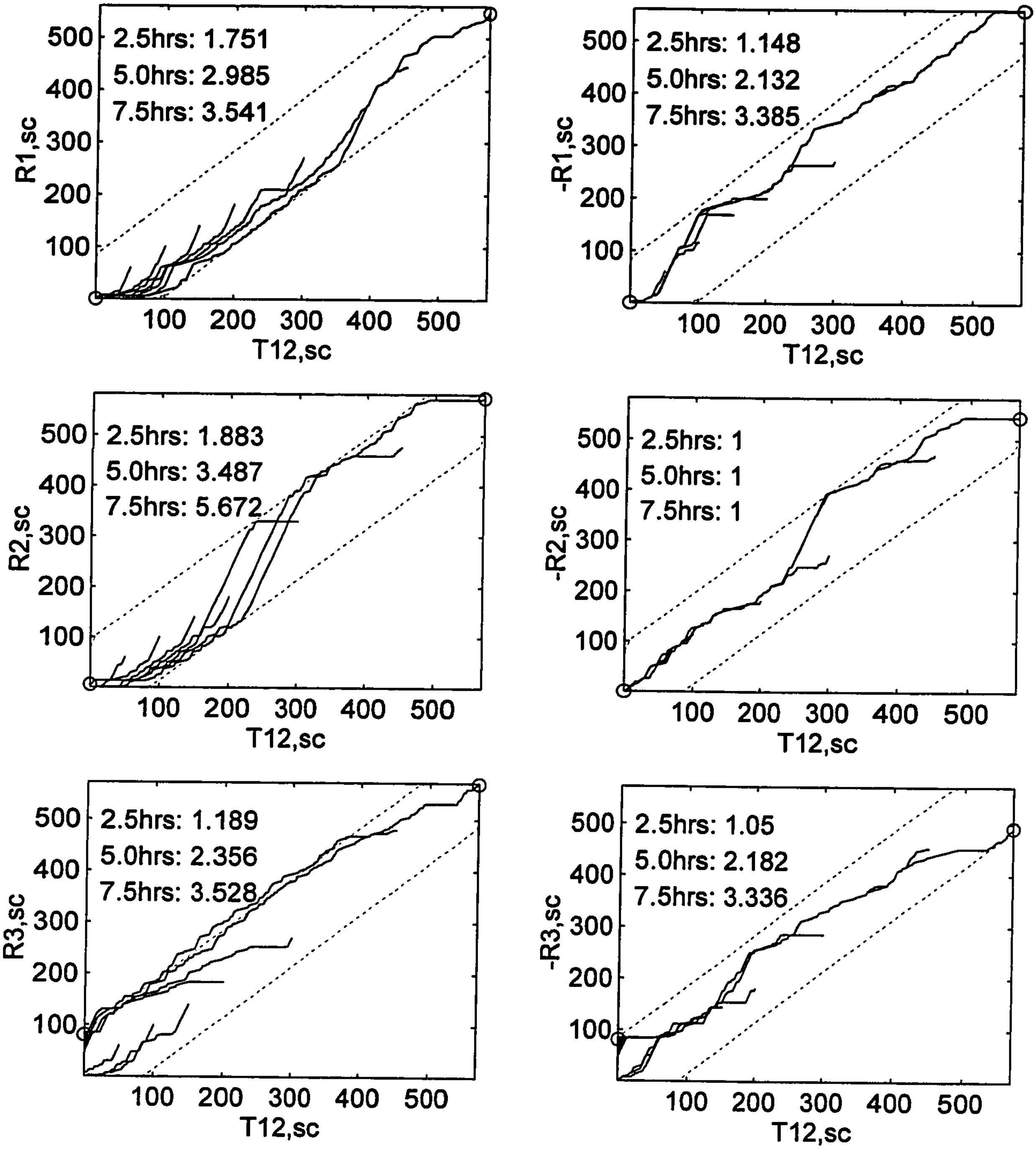


Figure 7.3: Results from the on-line diagnosis for test pattern $\mathbf{T}_{12}$; each subplot shows the 7 optimal paths (for the 7 selected times); the dotted lines indicate the upper and lower limits for the optimal path at the final time. The numbers show, for each of the first 3 times, the ratio of the 6 distances over the minimum of the six. At each time, the reference pattern that gives a ratio of one is deemed as most similar to the test pattern.

B) By the second time ($t_2 = 5$ hrs), with the exception of pattern $\mathbf{T}_{16}$ (which has no match in the reference set), all other patterns are diagnosed correctly. Also, the minimum distances found by DTW are drastically decreased (see second column in Tables 7.4 and 7.5). Similar results were obtained at all subsequent times, t_3 to t_7. As more data are collected, the time discrepancies in the synchronization of the patterns have a diminishing affect on the scaling factors. After the correct scaling is done, DTW can handle both unsynchronized patterns and differences in the temporal correlations caused by different production levels.

C) To verify that the incorrect diagnoses at time $t_1 = 2.5$ hrs were due to the discrepancies in the synchronization of the patterns, another simulation was performed where in all reference and test patterns the faults were introduced one hour after time zero. All the other parameters in the method were kept constant. The result was that all diagnoses (again with the exception of pattern $\mathbf{T}_{16}$) were correct at all times, including the first time $t_1 = 2.5$ hrs.

D) To illustrate the importance of scaling, another set of case studies was performed where a simpler scaling procedure was applied, similar to the off-line procedure of Chapter 4. In these case studies, each variable in the reference patterns was normalized using a standard deviation estimated from all the data points from time zero to the end. This was the only scaling factor that was used at all selected times, t_1 to t_7, for the reference patterns. All variables in the test patterns were then normalized by the scaling factors estimated from the reference patterns. For example, to compare $\mathbf{T}_1$ with $\mathbf{R}_1$, $\mathbf{T}_1$ was scaled with the scaling factors of $\mathbf{R}_1$; similarly, to compare $\mathbf{T}_1$ with $\mathbf{R}_2$, the scaling factors of $\mathbf{R}_2$ were used, etc. The results were good when the test pattern was a similar realization of a reference fault (i.e., same magnitude and same production level). For test patterns in which any of these two characteristics were different, the classifications were incorrect. In some cases, the test patterns were misdiagnosed at all times.

E) Finally, the arguments made in Chapter 4 regarding the effect of production level and the magnitude/direction of the fault on the classification apply in the results of this chapter as well. As expected, the more similar a test pattern is to any of the reference patterns, the more certain one can be that the diagnosis is correctly as more data points become available.

7.5 Summary and Conclusions

In this chapter a method was proposed for the on-line diagnosis patterns of deterministic faults in dynamic continuous multivariable processes. A Pattern Recognition approach was followed, where a scaling step removes the magnitude information from the variables and then a DTW step assesses the similarity between the evolving pattern of an unknown fault and the reference patterns. The comments made in Section 4.6 regarding the advantages and disadvantages of a Pattern Recognition method based on DTW apply in this chapter as well. However, there are two main differences from the method presented in Chapter 4 and both are related to the special characteristic of the on-line problem.

The first difference is that an asymmetric DTW algorithm is selected to assess the similarities between the patterns (in contrast to the symmetric algorithm of Chapter 4). The reason is the flexibility that an asymmetric DTW algorithm provides in relaxing the endpoint constraints. This is important in an on-line application where there is always uncertainty in locating the time origin of a fault. The second difference is the selection of a set of times at which scaling and pattern matching is performed. This is required because the variables in deterministic faults are nonstationary even after high-pass filtering.

The choice of the set of times at which scaling and pattern matching is performed depends on the dynamics of the process, the importance of diagnosing a fault as soon as possible, and the computational power available. In this study seven times were chosen in a period of 28 hours on average (the average duration of the patterns). The set of times is a parameter of the method that can be controlled by the user.

If the magnitude of a fault was always the same, one could scale the unknown test pattern by the scaling factors of the reference patterns; these factors would be estimated once from all data points. In this case, DTW could be used and a decision on the most likely cause for the unknown test pattern could be made at each time interval. However, it is very unlikely that faults will occur with exactly the same magnitude. The method proposed in this chapter can deal with the problem of varying fault magnitude at the expense of a deferred decision.

CHAPTER 8

MONITORING OF BATCH PROCESSES

USING SPEECH RECOGNITION METHODS

In this chapter the application of Dynamic Time Warping (DTW) to the monitoring of batch processes is presented. First, it is shown how one can use DTW to synchronize a set of good quality batches of unequal duration. Next, the combination of DTW with a monitoring method based on Multiway PCA/PLS is discussed for both on-line and off-line implementation. Finally, a new monitoring scheme based on local quadratic distances is presented. An industrial data set is used to illustrate the application and the performance of the proposed methods.

8.1 Introduction

As mentioned in Chapter 1, batch processes play an important role in the production of high added value products, such as specialty polymers, pharmaceuticals and biochemical materials. Monitoring the operation of these processes is crucial to the production of consistent, good quality product. Moreover, products from batch processes are often manufactured in a series of steps; early detection of a bad product at any of these steps will save energy, raw material and plant capacity. Early detection will also make it easier to assign a cause to the fault. If a monitoring scheme is implemented on-line, there may be a chance of correcting the fault with an appropriate control strategy.

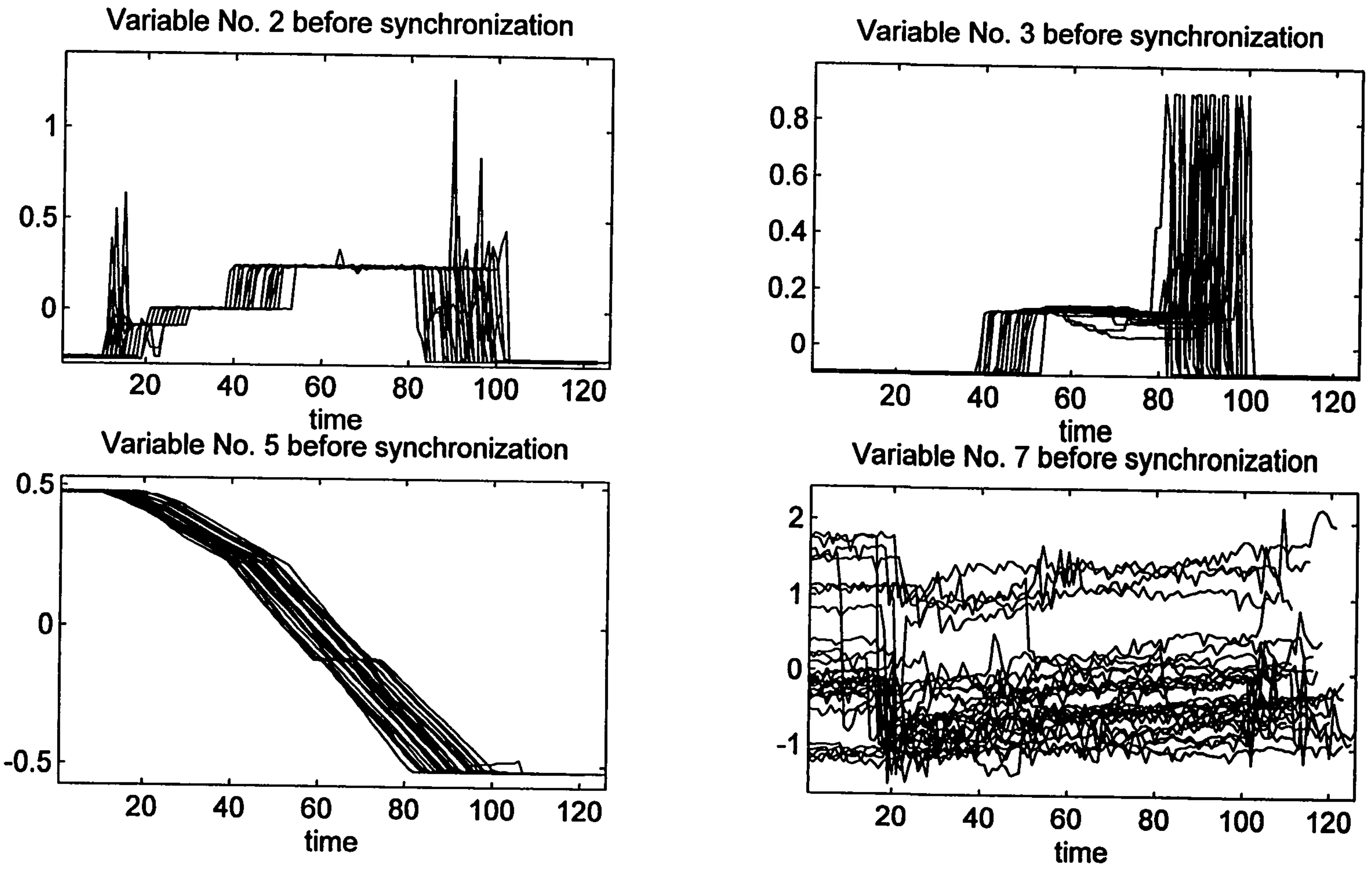
However, quality measurements in batch processes are obtained infrequently; sometimes they are obtained after the product has been shipped to the customer or after it has been forwarded to the next processing step. Fortunately, a multitude of process measurements such as temperatures, pressures, flowrates, are readily available during the progress of a batch. In view of this fact, Nomikos and MacGregor (1994, 1995a, 1995b) have proposed a method for monitoring batch processes using these readily measured process variables. Their method is based on Multiway Principal Component Analysis (MPCA), which is an extension of PCA to handle three dimensional matrices. The method essentially builds a statistical model for the deviations of the process variables about their average trajectory using data only from good quality batches. Then, it compares the variation in the variables of a new batch about the average trajectory with the MPCA model; any deviation that cannot be statistically attributed to the common process variation indicates that the new batch is different from the good quality batches. When quality measurements are available, then one can use Multiway Partial Least Squares (MPLS) to monitor the progress of the batch and predict its final quality (Nomikos and MacGregor, 1995b).

An important feature of their method, either MPCA- or MPLS-based, is that it can be implemented both off-line and on-line. However, the on-line implementation requires the prediction of the future behavior of the batch from the current time up to the expected end of the batch. Nomikos and MacGregor (1995a) discuss possible methods to carry out these predictions. A strong assumption in their method is that all batches have equal duration and are synchronized. However, the various steps comprising the batch process are not all automated; some are left to the discretion of operators. As a result, batches have different run lengths. Furthermore, even if some batches have the same duration, they may not be synchronized. In either case, one has to synchronize the batches before any analysis is performed.

To handle the problem of synchronization, Nomikos and MacGregor (1994) propose the use of an indicator variable to synchronize the batches. According to their proposal, the batches are plotted not with respect to time, but with respect to another variable that must be strictly monotonic, has the same starting and ending values for all batches and is not noisy. Then, a constant increment is selected and one progresses along the indicator variable. Synchronization is performed by retaining the points in the batch trajectories that are characterized by the same values of the indicator variable.

The indicator variable approach to synchronize batches assumes that that such a variable exists and process knowledge can be used to determine it. However, there may be several variables in a batch that are not noisy. It also relies on a single variable to perform an important operation, and as result it is not robust to missing values (a common problem in industry) of the indicator variable. Furthermore, there may not be a single variable that is strictly monotonic throughout the whole batch trajectory and one will have to switch between the selected variables at appropriate times. In summary, using an indicator variable is time consuming, requires ad hoc decisions and may not be feasible for many batch processes.

The previous chapters of this thesis have emphasized the capability of Dynamic Time Warping (DTW) to handle unsynchronized patterns by nonlinearly warping their time axes so that similar features within the patterns are matched. Thus, DTW can provide a more flexible and automatic solution to the problem of synchronization of batch trajectories. This is the subject of this last chapter of the thesis. First an iterative method based on DTW will be presented for the synchronization of batch trajectories. The method is multivariate in the sense that it does not rely on a single variable. Then, it is shown how to combine DTW with the monitoring scheme of Nomikos and MacGregor (1994) for both the off-line and the on-line implementation of batch monitoring. Finally, a novel monitoring scheme is presented based on the concept of instantaneous distance of a new batch from the average trajectory. The on-line implementation of this new monitoring scheme uses only information up to the current time and does not require any forecasting for the future behavior of the batch.



Variable No. 2 before synchronization

Variable No. 3 before synchronization

Variable No. 5 before synchronization

Variable No. 7 before synchronization

Figure 8.1: Behavior of 4 (out of 10) variables during the 31 good quality batches before synchronization; the variables have been divided with their average range.

Data from an industrial emulsion polymerization batch process will be used to illustrate the above points. Figure 8.1 shows 4 variables (out of 10) for a set of 31 good quality batches. Figure I.4 in the Appendix contains a plot of all the 10 variables used in this study. Both figures show the variables after they have been divided by their range; the latter has been estimated by averaging the ranges from all 31 batches (for each variable).

The variables shown in Figure 8.1 illustrate a number of issues relating to batch process data. The most important is that the batches are not synchronized and do not have the same duration. Variable No. 5, with the exception of the starting and ending part, is smooth and strictly monotonic; thus, it could be used as an indicator variable to synchronize the batches, as Nomikos and MacGregor (1994) proposed. However, it could not be used to synchronize the batches in the beginning and in the end, since this variable remains constant at these times. Variables No. 2 and 3 are piecewise constant with occasional step changes in their level. As such, they do not contain enough information to make them useful as indicator variables, but the times where their values step from one level to the next could be used to test the quality of the synchronization. Finally, Variable No. 7 is a noisy variable, and therefore one would not use it as an indicator variable.

8.2 Synchronization of Good Quality Batches

8.2.1 Dynamic Time Warping Algorithm and Synchronization Procedure

Let $\mathbf{B}_{i,SC}, i = 1,\ldots,I$, be a set of I trajectories of good quality batches. Each $\mathbf{B}_{i,SC}$ is a matrix of $b_i \times N$, where N is the number of measured process variables and b_i is the number of data points, ordered from time zero (first row in the matrix) to the end of the batch (last row). Also, assume that some appropriate scaling for the variables has been applied. Finally, assume that a reference batch trajectory, $\mathbf{B}_{REF,SC}$, has been somehow defined; this is a matrix of $b_{REF} \times N$. The issues of scaling and definition of reference trajectory will be presented in the following subsection. Now the objective is to synchronize the scaled batch trajectories, $\mathbf{B}_{i,SC}, i = 1,\ldots,I$, with the scaled reference trajectory, $\mathbf{B}_{REF,SC}$.

As discussed previously in this thesis, DTW works with pairs of patterns. Thus, one needs to separately synchronize each batch trajectory with the reference trajectory. The main question is what kind of DTW algorithm should be used; more specifically, whether it should be a symmetric or an asymmetric algorithm.

As emphasized in Chapter 3, when two patterns (e.g., $\mathbf{B}_{i,SC}$ and $\mathbf{B}_{REF,SC}$) are compared via a symmetric DTW algorithm, both of them are considered equivalent. The optimal path will go through all points in both patterns. The result is the mapping of the time axes of both patterns onto a common time axis. After DTW is performed, the synchronized patterns have equal duration, which is greater than the duration of the patterns before synchronization (i.e., greater than b_i and b_{REF}). This common duration is determined by the DTW algorithm and cannot be specified a priori. Furthermore, it is different for each $\mathbf{B}_{i,SC}$ that is synchronized with $\mathbf{B}_{REF,SC}$. Therefore, if a symmetric DTW is used to synchronize each $\mathbf{B}_{i,SC}$ with $\mathbf{B}_{REF,SC}$, the result will be a set of expanded patterns with unequal duration; each $\mathbf{B}_{i,SC}$ will be individually synchronized with $\mathbf{B}_{REF,SC}$ but not with each other. The final situation will be identical to the initial situation: having a set of batch trajectories with unequal duration.

On the other hand, the most common asymmetric DTW algorithms treat one pattern preferentially. The optimal path goes through all points in one of the patterns (which can be viewed as the defining pattern) and can skip points of the other. The result is the mapping of the time axis of the defining pattern onto the time axis of the other. After DTW is performed, the synchronized patterns have equal duration, equal to the duration of the defining pattern. For the current problem, one would use $\mathbf{B}_{REF,SC}$ as the defining pattern and map its time axis onto the time axis of each $\mathbf{B}_{i,SC}$. The end result will be a set of synchronized patterns with equal duration, b_{REF}, all of them synchronized with $\mathbf{B}_{REF,SC}$ and synchronized with each other.

At first sight this appears to be a reasonable solution. Unfortunately, the synchronized trajectories may not contain all the data points of the original $\mathbf{B}_{i,SC}$ because the optimal path may have skipped selected points in them. This is an undesirable side effect because features that appear in some $\mathbf{B}_{i,SC}$ and do not appear in $\mathbf{B}_{REF,SC}$ (e.g., a spike) will be left out. In effect, a subtle filtering is performed that removes inconsistent features. If a MPCA/MPLS model is constructed from the 'filtered' trajectories, it will be biased towards false alarms since it will not consider inconsistent features that may be present in a new batch trajectory.

In summary, symmetric DTW algorithms include all points in the original trajectories but result in expanded trajectories of various lengths. Asymmetric DTW algorithms may eliminate points but will produce synchronized trajectories of equal length. The following method (symmetric DTW algorithm combined with an asymmetric synchronization procedure) purposes to achieve a compromise between the two extremes.

Step A: *Symmetric DTW Algorithm*

For each $\mathbf{B}_{i,SC}$, apply DTW between $\mathbf{B}_{i,SC}$ and $\mathbf{B}_{REF,SC}$ using the following constraints:

i) fixed-endpoint constraints (see Section 3.2)

ii) band global constraint (see Section 3.2)

iii) local constraint:

$$\mathbf{D}_A(i,j) = \min\begin{Bmatrix} \mathbf{D}_A(i-1,j) & + d(i,j) \\ \mathbf{D}_A(i-1,j-1) & + d(i,j) \\ \mathbf{D}_A(i,j-1) & + d(i,j) \end{Bmatrix},\ \mathbf{D}_A(1,1) = d(1,1) \qquad (8.1)$$

At the end, reconstruct the optimal path.

Step B: *Asymmetric Synchronization*

When more than one points of $\mathbf{B}_{i,SC}$ are aligned with one point of $\mathbf{B}_{REF,SC}$ do as follows:

i) take the average of these points of $\mathbf{B}_{i,SC}$

ii) align this average point with the particular point of $\mathbf{B}_{REF,SC}$.

After synchronization, $\mathbf{B}_{i,SC}$ contains as many data points as $\mathbf{B}_{REF,SC}$; i.e., b_{REF}.

The second step *Asymmetric Synchronization* can be best illustrated by means of an example. Assume that $\mathbf{B}_{i,SC}$ is placed on the horizontal and $\mathbf{B}_{REF,SC}$ on the vertical axis. This arrangement does not affect the DTW algorithm since it is symmetric. Also assume that after DTW, the following three points are included in the optimal path: $(i-1,j)$, (i,j) and $(i+1,j)$. According to them, the $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ points of $\mathbf{B}_{i,SC}$ are all aligned with the j^{th} point of $\mathbf{B}_{REF,SC}$. The proposed synchronization takes the average of the three: $\frac{\mathbf{B}_{i,SC}(i-1,:)+\mathbf{B}_{i,SC}(i,:)+\mathbf{B}_{i,SC}(i+1,:)}{3}$, and aligns this average with $\mathbf{B}_{REF,SC}(j,:)$.

The proposed DTW algorithm is a symmetric algorithm and as such the optimal path passes though all the points in both patterns. On the other hand, the local constraint favors diagonal over horizontal or vertical local transitions. The local constraint is a modification of the one used in Subsection 3.3.1, Example 1; i.e.,

$$\mathbf{D}_A(i,j) \quad = \quad \min\begin{Bmatrix} \mathbf{D}_A(i-1,j) & + \; d(i,j) \\ \mathbf{D}_A(i-1,j-1) & + 2d(i,j) \\ \mathbf{D}_A(i,j-1) & + \; d(i,j) \end{Bmatrix} \tag{8.2}$$

The local constraint in Eq (8.2) gives a weight of 2 to the local distance $d(i,j)$ for a diagonal local transition (i.e., from $(i-1,j-1)$ to (i,j) point). This weight was the result of a symmetric weighting function; its purpose was to provide independence of the final distance to the number of points in the optimal path (as described in Section 3.3).

However in this problem only the optimal path is of interest and not in the final distance found by DTW. If the smaller weight of 1 is used instead (i.e., as in Eq (8.1)) , diagonal local transitions are preferred over horizontal or vertical ones; and it is the horizontal and vertical transitions that distort the time axes of $\mathbf{B}_{i,SC}$ and $\mathbf{B}_{REF,SC}$. Thus, the constraints of Eq (8.1) result in smaller distortions of the time axes of both $\mathbf{B}_{i,SC}$ and $\mathbf{B}_{REF,SC}$ and consequently in less averaging in Step B.

Even if Eq (8.1) is used (i.e., favoring diagonal over horizontal or vertical local transitions), the resulting DTW algorithm of Step A is still a symmetric algorithm. The following Step B is an asymmetric operation that synchronizes all $\mathbf{B}_{i,SC}$ in a way that all have the same duration b_{REF}. However, all points of $\mathbf{B}_{i,SC}$ (even if some of them have been averaged) are included in the synchronized trajectory. This is the difference between the proposed method and any other asymmetric DTW algorithm. An asymmetric algorithm will completely ignore points in $\mathbf{B}_{i,SC}$, while the proposed method will average selected points. One can use the synchronized trajectories from the proposed method to build a MPCA/MPLS model; the model will be still biased towards false alarms (due to averaging of inconsistent features). However, it will be less prone to false alarms than a model which was based on synchronized batches from an asymmetric DTW algorithm.

This completes the description of a method that takes a set of scaled batch trajectories of unequal duration and synchronizes them with a reference trajectory in such a way that: I) all synchronized trajectories have the duration of the reference trajectory and II) all data points in the original scaled trajectories are included although some will be averaged. In this discussion, it has been assumed that the raw trajectories have been scaled appropriately and that also a reference trajectory exist. The next subsection presents these issues in detail, together with the complete method for synchronization of batch trajectories.

8.2.2 An Iterative Method for Synchronization of Batch Trajectories

As described in the previous chapters, DTW is a distance-based method and as such is sensitive to the scaling of variables. In the case of batch processes an intelligent scaling should accomplish two objectives. The first is to remove the effect of the various engineering units used to record the variables. This is easily achieved by dividing each variable by its standard deviation or its range. The second and most important objective is to give more weight to variables that are consistent from batch to batch. The synchronization of batch trajectories should rely more on these variables (e.g., Variables No. 2,3 and 5 in Figure 8.1) and less on noisy variables (e.g., Variable No. 7 in Figure 8.1). This relative importance of variables is expressed through the weight matrix, **W** used in the local distance computation in DTW; i.e.,

$$d(\mathrm{i},\mathrm{j}) = \left[\mathbf{B}_{\mathrm{i,SC}}(\mathrm{i},:) - \mathbf{B}_{\mathrm{REF,SC}}(\mathrm{j},:)\right]\mathbf{W}\left[\mathbf{B}_{\mathrm{i,SC}}(\mathrm{i},:) - \mathbf{B}_{\mathrm{REF,SC}}(\mathrm{j},:)\right]^{\mathrm{T}} \tag{8.3}$$

One choice would be to arbitrarily assign a weight to each variable; however this would require process knowledge and a number of ad hoc decisions. A more appealing choice would be to devise a procedure that will automatically detect and increase the weight of consistent variables and decrease the weight of the rest. For each variable, the sum of the squared deviation from the average trajectory over all batches can be used as an indicator of consistency over different realizations.

Regarding the reference trajectory, a reasonable choice would be to set it equal to the average trajectory. However, at the start of the synchronization procedure is not possible to average the batch trajectories since each one of them has a different duration. Thus, one trajectory from the set could be used as the reference trajectory. One could then synchronize all other trajectories to this particular one using the DTW/synchronization method of the previous subsection. After synchronization all trajectories will have the same duration and so an average trajectory can be defined. The whole procedure can then be repeated and in the next iteration the average trajectory can be used as the reference one.

These are essentially the main steps of the iterative procedure proposed for the synchronization of unequal batch trajectories, which is now being presented in detail. Let $\mathbf{B}_i, i = 1,\ldots,I$, be a reference set of trajectories which contain the raw measurements from I good quality batches; each is a matrix of $b_i \times N$, where N the number of measured process variables and b_i is the number of data points. The steps of the method are as follows:

Step A: Scaling

For each variable, find its average range (i.e., the difference between the maximum and the minimum value) by averaging the range from each batch.

Store these values because they will be used in the off-line and on-line monitoring of a new batch (see next section).

Divide each variable in all batches with its average range.

Let $\mathbf{B}_{i,SC}, i = 1,\ldots,I$, be the resulting scaled batch trajectories.

Step B: Synchronization

Step 0: Select one of the trajectories, $\mathbf{B}_{k,SC}$, as the reference trajectory: $\mathbf{B}_{REF,SC} = \mathbf{B}_{k,SC}$.

Consequently: $b_{REF} = b_k$.

Set **W** (the weight matrix in the DTW algorithm) equal to the identity matrix.

Execute the following steps for a specified maximum number of iterations.

Step 1: Apply the DTW/synchronization method between $\mathbf{B}_{i,SC}, i = 1,\ldots,I$, and $\mathbf{B}_{REF,SC}$ as described in the previous subsection.

Let $\tilde{\mathbf{B}}_{i,SC}, i = 1,\ldots,I$ be the synchronized trajectories, with b_{REF} now being their common duration.

Step 2: Compute the average trajectory, $\overline{\mathbf{B}}_{SC}$; i.e., $\overline{\mathbf{B}}_{SC} = \frac{\sum_{i=1}^{I} \widetilde{\mathbf{B}}_{i,SC}}{I}$.

Step 3: For each variable, compute the squared deviations from the average trajectory. The inverse of this value will be the weight of the particular variable for the next iteration; the (j, j) element of the diagonal matrix $\mathbf{W}$ will be:

$$\mathbf{W}(j,j) = \left[\sum_{i=1}^{I} \sum_{k=1}^{b_{REF}} [\widetilde{\mathbf{B}}_{i,SC}(k,j) - \overline{\mathbf{B}}_{SC}(k,j)]^2 \right]^{-1}.$$

Normalize $\mathbf{W}$ so that the sum of the weights is equal to the number of variables; i.e., replace $\mathbf{W}$ with $\mathbf{W}\frac{N}{\sum_{j=1}^{N} \mathbf{W}(j,j)}$.

Step 4: For the first 3 iterations, keep the same reference trajectory: $\mathbf{B}_{REF,SC} = \mathbf{B}_{k,SC}$. For subsequent iterations, set the reference equal to the average trajectory: $\mathbf{B}_{REF,SC} = \overline{\mathbf{B}}_{SC}$.

The scaling step assumes that each variable starts and ends at the same value in all trajectories (up to some process noise). This is a reasonable assumption since the same product is manufactured in all batches. As mentioned before, dividing each variable by its average range removes the effect of the various engineering units used to record the variables. Note that for each variable, the same scaling factor is used in all batches. When the trajectories are plotted after scaling, their relative position (which indicates that they are not synchronized) remains the same as before scaling. This would not be true, if each variable in each batch had been divided by its range as estimated from that particular batch.

The synchronization step assumes initially that all variables are equally important in the synchronization by setting the weight matrix $\mathbf{W}$ equal to the identity matrix. After the first iteration, the weight of each variable depends on the magnitude of its deviation from its average trajectory. Variables that do not deviate much from their average trajectory are weighted more in the DTW/synchronization process than others with larger deviations. Also, note that one of the original scaled trajectories is used as the reference trajectory for the first 3 iterations; the average trajectory is used only after the third iteration. Initially all variables are weighted equally and it takes two or three iterations for the weight matrix to start converging towards its final value. Since noisy variables are also weighted equally to consistent ones, the synchronization of the trajectories is poor and the average trajectory from these iterations can differ significantly from the one obtained at subsequent iterations. For that reason, the same trajectory is kept as the reference trajectory for the initial iterations and the average trajectory is used at subsequent iterations.

Note that the length of the synchronized trajectories at the end of the iterative procedure will be the length of the trajectory initially used as the reference trajectory. Alternatively, one could estimate the average duration from the initial trajectories and the trajectory whose duration is closest to the average duration could be used as the average trajectory for the first three iterations. By doing that, the duration of the synchronized trajectories at the end will be the average duration of the available realizations. The choice of the initial reference trajectory is a matter of user preference.

Finally, the maximum number of iterations is another parameter of the method set by the user. One could also monitor the change of the weight matrix **W** from one iteration to the next and use it an indicator for convergence.

This concludes the description of the proposed method for the synchronization of unequal batch trajectories. In the next subsection the method is illustrated through its application on a set of industrial data.

8.2.3 Results and Discussion

Figure 8.1 shows 4 variables (out of 10) for 31 trajectories, $\mathbf{B}_i$, $i = 1,\ldots,31$, from an industrial emulsion polymerization process. Their duration varies from 106 to 126 data points and the average duration is 115. There exist three trajectories with that duration and one of them, $\mathbf{B}_{21}$, was chosen to be the average trajectory for the first three iterations. For the DTW/synchronization procedure, the band global constraint was used with maximum allowable deviation $M = 35$ (from the linear path emanating from point $(1,1)$). The iterative procedure was executed for 10 iterations. The band global constraint was never active at any iteration and for any $\mathbf{B}_{i,SC} - \mathbf{B}_{REF,SC}$ pair; it just served the purpose of speeding the computations.

The results after the final (10th) iteration are shown in Figures 8.2 and 8.3. Figure 8.2 shows the variables after the trajectories have been synchronized and Figure 8.3 shows how the weights of the 4 variables change with respect to the iterations. As Figure 8.2 illustrates, the variables are now synchronized. This is more apparent by looking at the times of the step changes in Variables No. 2 and 3 and the spike in Variable No. 3. Due to the averaging of selected points by the asymmetric synchronization procedure, some of the spikes in Variable No. 2 (see Figure 8.1) have been filtered; however they are not completely removed as Figure 8.2 shows.

It was mentioned in Section 8.1 that the Variable No. 5 is a smooth variable and it could be used as an indicator variable to synchronize the trajectories as Nomikos and MacGregor (1994) proposed. The proposed iterative procedure verified this argument (as Figure 8.3 shows) since the weight of Variable No. 5 accounts for about 85% of the total weight (the indicator variable solution essentially gives 100% of the total weight to this variable). However, Variable No. 5 cannot be used as an indicator variable at the beginning and at the end of the batches since it remains constant over these time intervals.

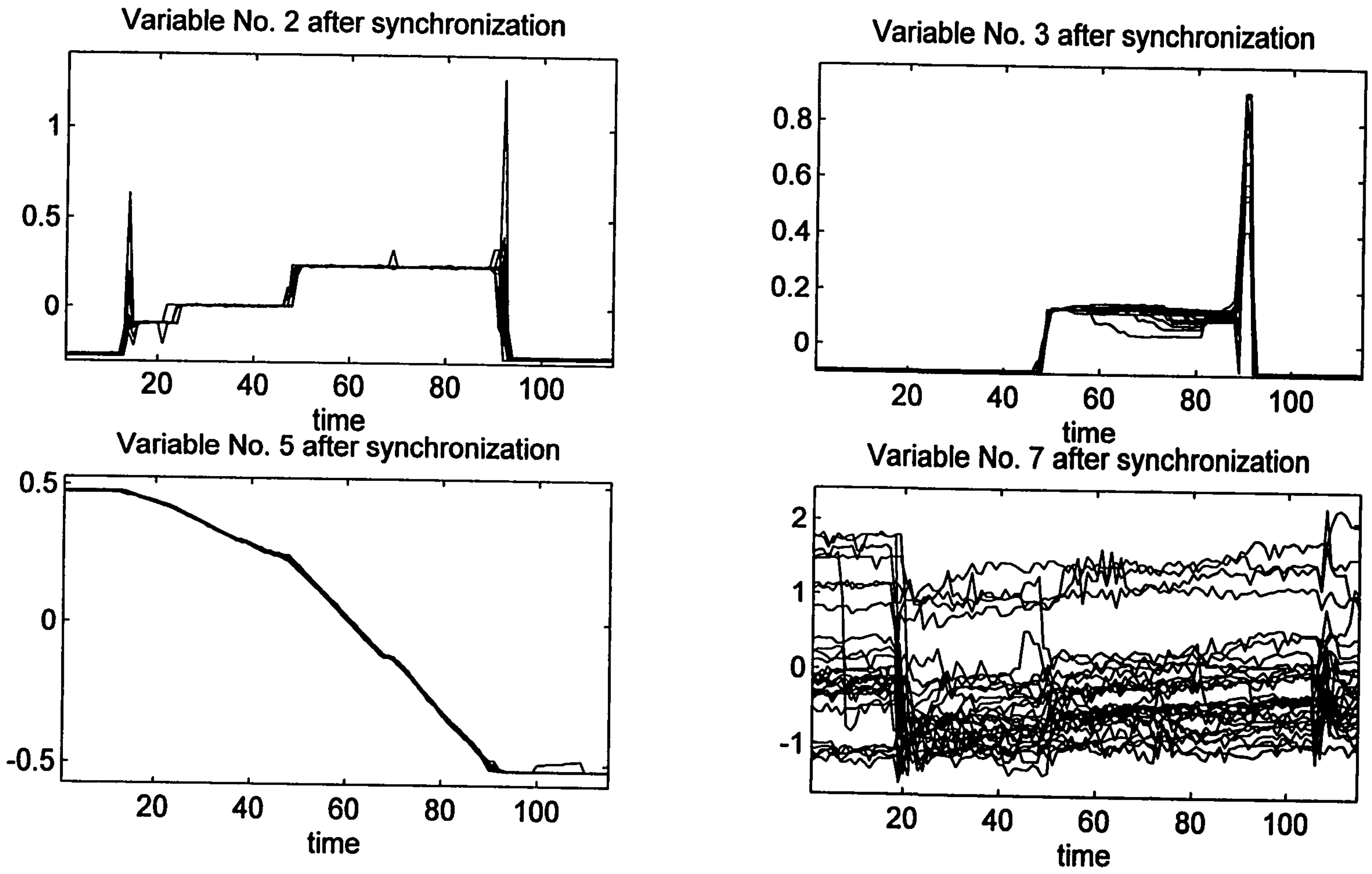


Figure 8.2: Behavior of 4 (out of 10 variables) for all 31 batches after synchronization.

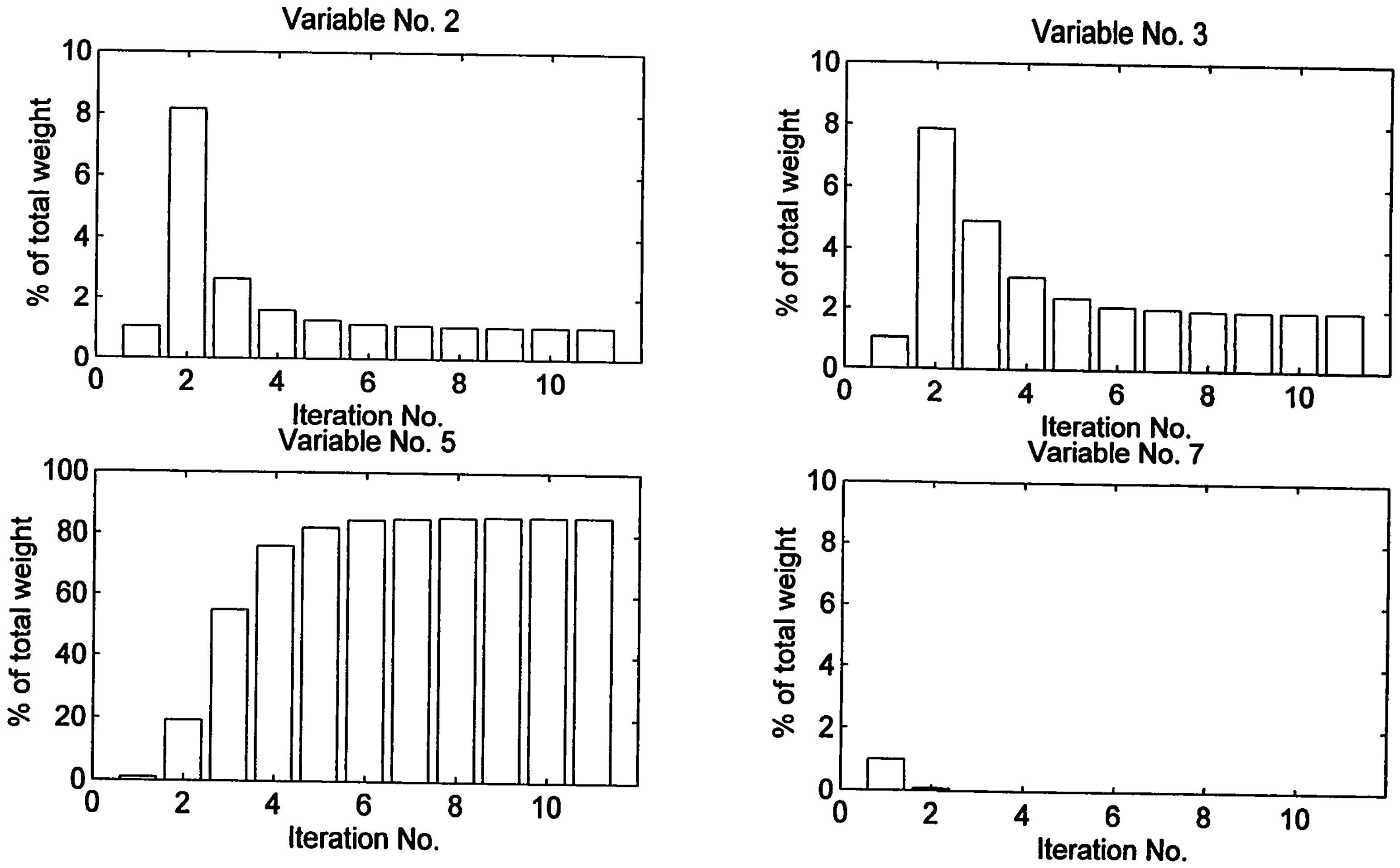


Figure 8.3: % of total weight versus iteration number for the 4 variables shown in Figures 8.1 and 8.2.

The proposed method does not suffer from this limitation and it will synchronize the trajectories, as long as there are some other variables that exhibit any variation over these intervals. This can be seen from the fact that 15% of the total weight has been distributed to the other variables. Finally, Variable No. 7 is a noisy variable (as Figure 8.1 shows) and one would not base any synchronization on that variable. The iterative procedure quickly responded to that and gave small weight to Variable No. 7 after the first iteration (see Figure 8.3).

The iterative procedure could also be used to pinpoint the most appropriate variable to be used as an indicator variable if one wants to use this simpler method for synchronization without relying on expert process knowledge. There may be situations where several variables are smooth and monotonic; thus they could all be candidates for the role of the indicator variable. The proposed method could assist in choosing the most appropriate one by selecting the variable that gets the largest weight in matrix **W**.

Due to the nonlinear warping of the trajectories imposed by DTW and the asymmetric averaging operation, it was not possible to construct a proof for the convergence of the proposed iterative procedure. Therefore, possible failure modes should be investigated. One possible failure allows one variable to take almost all the weight, even though its deviation from the average trajectory is only slightly smaller than some other variable. This 'positive feedback' within the iterative procedure is clearly an undesirable feature since the small differences in the deviations from the average trajectory that determine the indicator variable could be just a feature of the particular data set. In this case, the 'positive feedback' effect would mean that the indicator variable may change from one data set to the other.

To investigate the 'positive feedback' failure mode the following case study was performed. Variable No. 5 was spliced into two parts and two artificial variables were created. The first contains the initial part of Variable No. 5 up to the point that reaches the value of zero; then, it is padded with zeros up to the end of the trajectory.

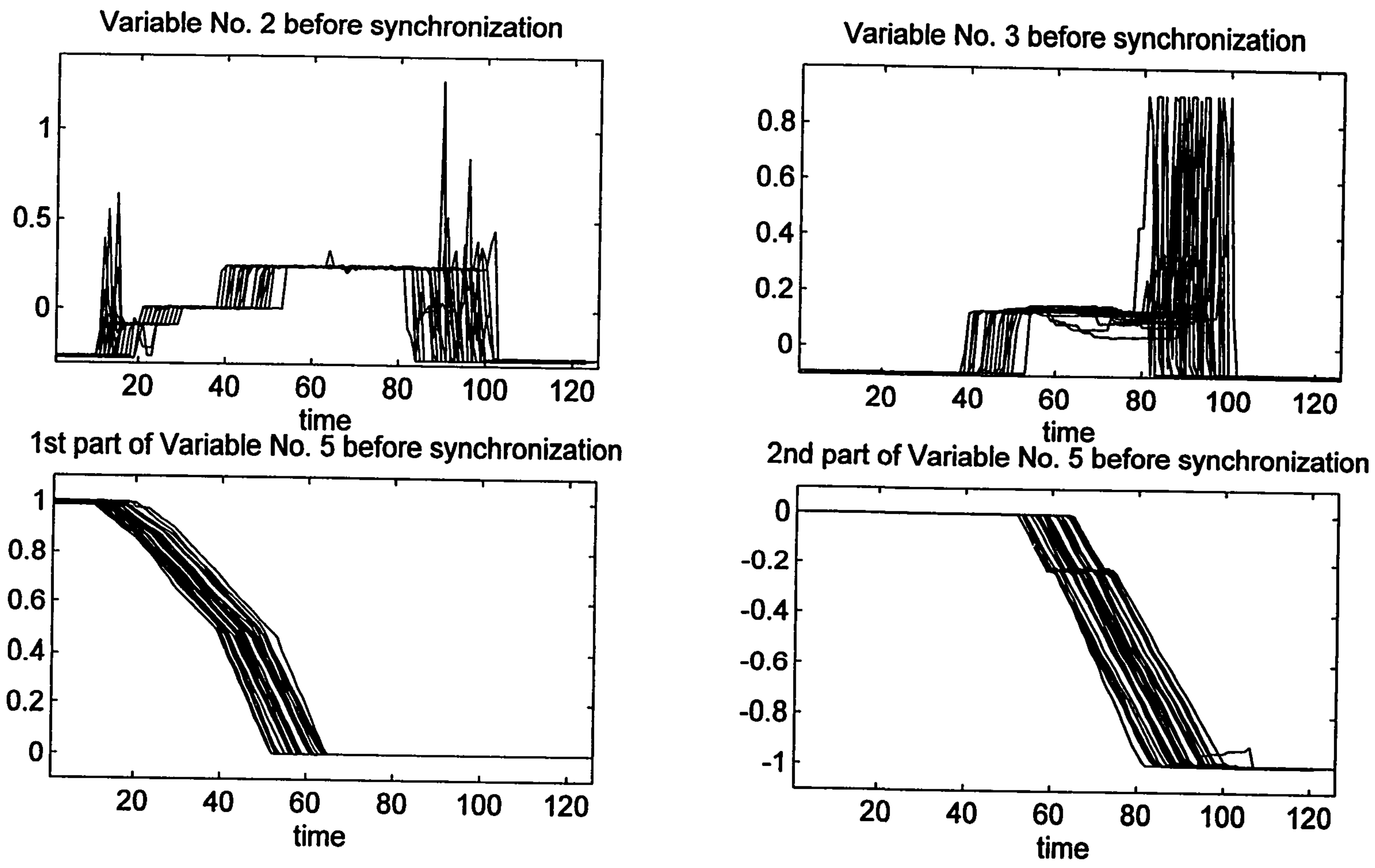


Figure 8.4: Behavior of 4 (out of 11 variables) during the 31 good quality batches before synchronization; Variable No. 5 has been split into two variables.

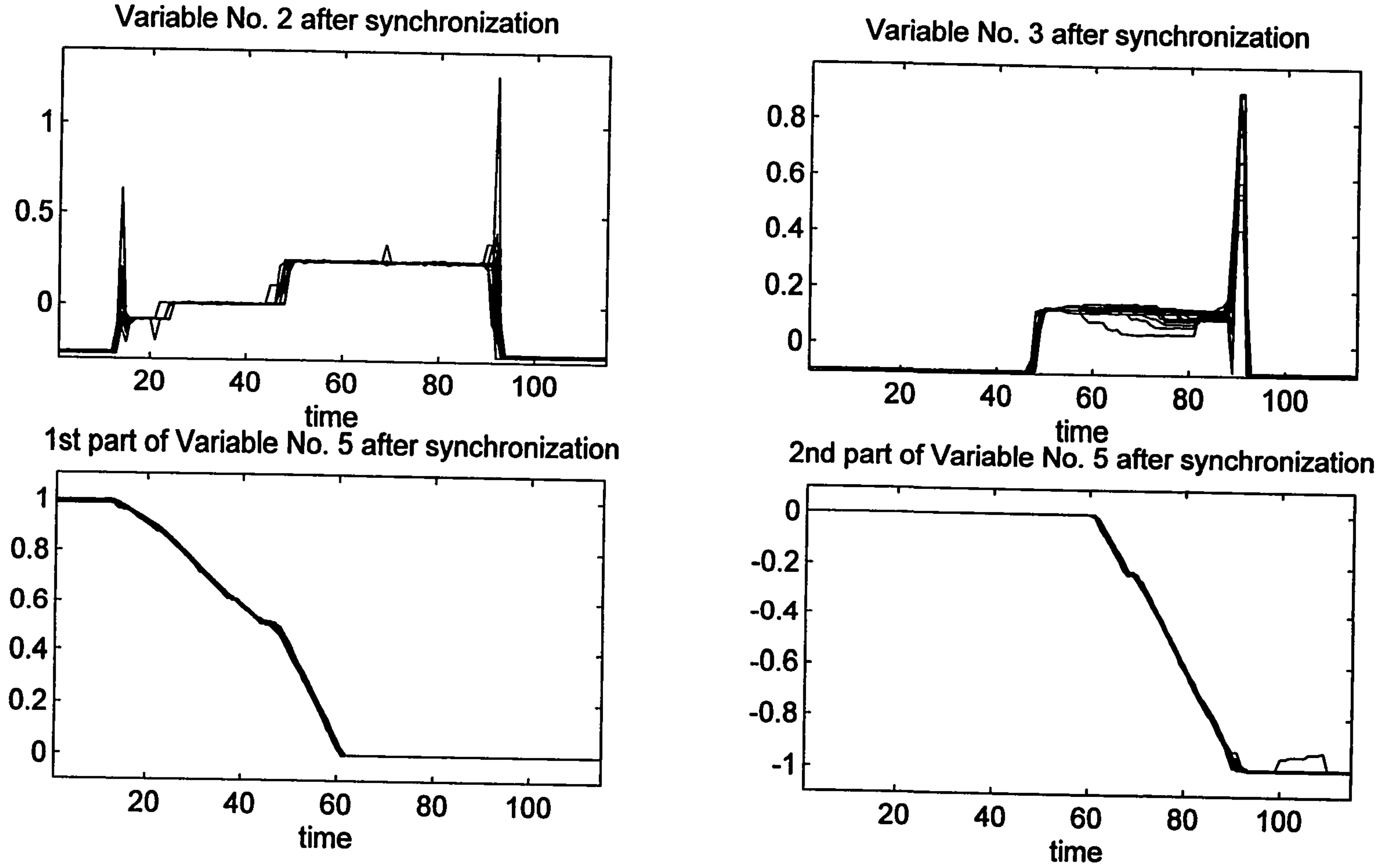


Figure 8.5: Behavior of the 4 variables shown in Figure 8.4 after synchronization.

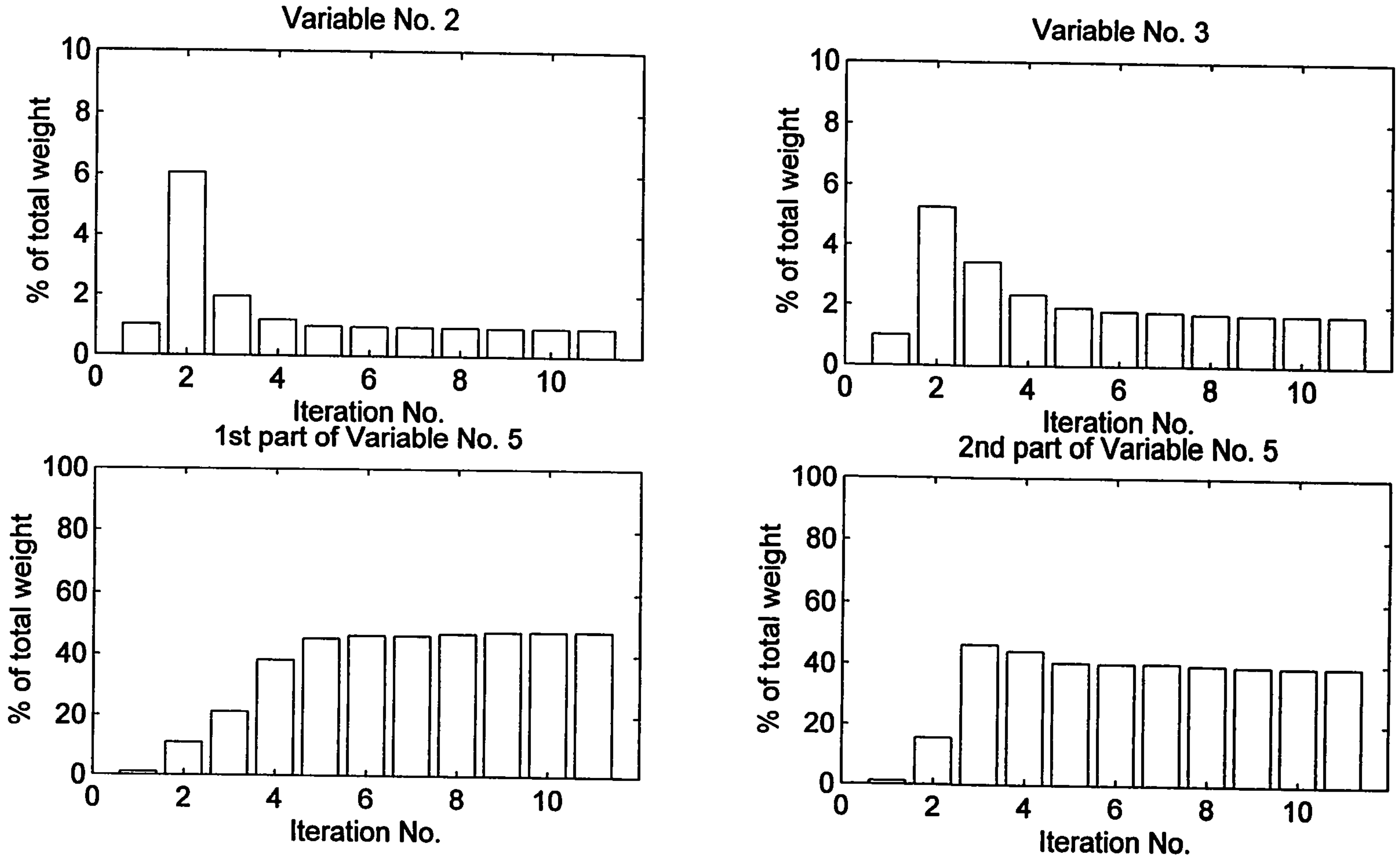


Figure 8.6: % of total weight versus iteration number for the 4 variables shown in Figures 8.4 and 8.5.

Similarly, the second artificial variable contains initially a number of zeros, followed by the second part of Variable No. 5. The two variables are shown in Figure 8.4.

Thus, for this case study the batch trajectories contained 11 variables: 9 original plus the two artificial ones. The original Variable No. 5 was not included in the data set. Next, the synchronization method was applied with the same parameters described before and the results are presented in Figures 8.5 and 8.6. As Figure 8.5 shows, the synchronization of the variables is quite good and similar to the one obtained before (i.e., Figure 8.2). The variables' weights are shown in Figure 8.6. Interestingly, about 85% of the total weight is now distributed between the two artificial variables and not to just one of them.

Although this case study does not constitute a proof, it shows that the danger of 'positive feedback' is not likely to exist in industrial data. For a variable to take all the weight, the iterative procedure would have to remove all of that variable's variation about the reference trajectory. However, at each iteration it is the original scaled trajectories that are synchronized with the reference trajectory and not the synchronized trajectories from the previous iteration. Process noise and measurement noise are always present in industrial data, and this random variation cannot be completely eliminated with any warping of the time axes of the patterns.

8.3 Batch Monitoring using MPCA/MPLS

8.3.1 The Off-Line Implementation

Once timing differences have been removed by the synchronization procedure, the synchronized batch trajectories can now be used to build a MPCA/MPLS model for process monitoring as proposed by Nomikos and MacGregor (1994, 1995b). However, one important feature has been removed from the raw data (i.e., the timing differences) and it could be the case that this feature is indeed affecting the final product quality. To account for this possibility, the amount of time distortion (which was exerted upon each trajectory) should be included in the MPLS model. These time distortions can be treated as an additional variable in the initial condition matrix of the MPLS model (see Kourti et al., 1995).

Now, assume that the complete trajectory of a new batch, $\mathbf{B}_{\mathrm{NEW}}$, (a matrix of $b_{\mathrm{NEW}} \times N$) is available. The objective is to use the MPCA/MPLS-based monitoring scheme to assess the product quality of the new batch. Most probably the duration of the new batch, b_{NEW}, will not be equal to the duration of the synchronized batches, b_{REF}, that were used to construct the monitoring model. Even if $b_{\mathrm{NEW}} = b_{\mathrm{REF}}$, some stages of the new batch may not be synchronized with the corresponding stages of the reference trajectory. In either case, the new trajectory has to be synchronized before the monitoring scheme is applied. The following method purposes to accomplish this task.

Step A: *Scaling*

Divide each variable in the new batch with the average range estimated from the trajectories of the reference set.

Let $\mathbf{B}_{\mathrm{NEW,SC}}$ be the resulting scaled new trajectory.

Step B: *Synchronization*

Step 1: Let $\mathbf{B}_{\mathrm{REF,SC}}$ and $\mathbf{W}$ be the reference trajectory and the weight matrix used in the last iteration of the synchronization procedure.

Step 2: Apply the DTW/synchronization method presented in Subsection 8.2.1 to synchronize the new batch trajectory with the reference set trajectories.
Let $\widetilde{\mathbf{B}}_{\mathrm{NEW,SC}}$ be the synchronized new batch.

The new trajectory $\widetilde{\mathbf{B}}_{\mathrm{NEW,SC}}$ is synchronized with the reference trajectories, its duration is b_{REF}, and the MPCA/MPLS-based batch monitoring scheme can now be applied. Note that some points in $\widetilde{\mathbf{B}}_{\mathrm{NEW,SC}}$ will be averages of selected points of $\mathbf{B}_{\mathrm{NEW,SC}}$ as a result of the asymmetric synchronization procedure (described in Subsection 8.2.1).

Since the averaging operation smoothes spurious features, $\widetilde{\mathbf{B}}_{\mathrm{NEW,SC}}$ is biased towards the null hypothesis, i.e., the new batch being of good quality. This is a compromise that one has to accept if wants to use the same MPCA/MPLS model to monitor each new batch.

8.3.2 The On-Line Implementation

The on-line implementation of the MPCA/MPLS-based monitoring scheme is similar to the off-line implementation with one important difference: in the on-line case, the prediction of the future behavior of the batch trajectory up to its expected end is required. Nomikos and MacGregor (1995a) discuss possible methods to carry out these predictions. However, they assume that the new batch is synchronized with the reference set batches. In real time, this assumption means that the progress of the new batch up to the current time (i.e., t_{CUR}) is equivalent to the progress of the reference set batches up to time t_{CUR}. Therefore, one has to predict the behavior of the batch from time t_{CUR} and onward up to its end; the end time for the new batch is assumed to be the common duration of the reference set batches.

This assumption may not be always true in an industrial batch process since some stages of the process may not be automated. Let $\mathbf{B}_{\mathrm{NEW}}$ be the raw measurements of the evolving new batch, a matrix of $\mathrm{t} \times \mathrm{N}$, with t being the number of data points from time zero up to the current time. To monitor on-line its progress, one would have to answer the following question: which point r of the reference trajectory best represents the progress of the new batch up to the current time? DTW can provide an answer to this question as follows:

Step A: *Scaling*

Divide each variable in the new batch with the average range estimated from the trajectories of the reference set.

Let $\mathbf{B}_{\mathrm{NEW,SC}}$ be the resulting scaled new trajectory.

Step B: *Synchronization*

Step 1: Let $\mathbf{B}_{REF,SC}$ and $\mathbf{W}$ be the reference trajectory and the weight matrix used in the last iteration of the synchronization procedure.

Step 2: Apply the DTW symmetric algorithm presented in Subsection 8.2.1. However, since only the first t data points of the new batch are available, one would have a set of accumulated distances $\mathbf{D}_A(t, j), j = \mathbf{l}(t), \ldots, \mathbf{u}(t)$; $\mathbf{l}(t)$ and $\mathbf{u}(t)$ are the lower and upper bound imposed by the band constraint on the index of the inner iteration.

Let r be the point in the $\mathbf{D}_A(t, j), j = \mathbf{l}(t), \ldots, \mathbf{u}(t)$ vector where the minimum occurs; i.e., $r = \arg\min_j [\mathbf{D}_A(t, j)]$.

Step 3: Synchronize the t points of the new batch to the first r points of the reference trajectory using the asymmetric synchronization method presented in Subsection 8.2.1. After synchronization, the new trajectory, $\tilde{\mathbf{B}}_{NEW,SC}$, will have r points.

Step 4: Predict the progress of the new batch from point $(r+1)$ up to the final point of the reference trajectory b_{REF}. Now the MPCA/MPLS-based monitoring scheme can be applied on-line as described by Nomikos and MacGregor (1994, 1995b).

The above method has to be repeated as soon as another measurement from the new batch is available. Again, $\tilde{\mathbf{B}}_{NEW,SC}$ is biased towards the null hypothesis (i.e., the new batch being of good quality) because of the averaging of selected points in $\mathbf{B}_{NEW,SC}$. Again, it is a necessary compromise that has to be made so that the monitoring model constructed from the reference set batches is used at each time step.

8.4 On-Line Batch Monitoring Using a Distance-Based Method

Even when combined with DTW, the on-line monitoring scheme of Nomikos and MacGregor (1994) still requires the prediction of a batch trajectory up to its end. A monitoring scheme will be presented in this section that is based on the concept of the instantaneous distance of a new batch trajectory from the average trajectory and, as such, no predictions are required. On the other hand, it is essentially a univariate scheme, because at each point in time it only considers the sum of the weighted squared deviation of each variable from its average trajectory; any change in the correlation among the variables is not penalized. The details are presented in the following subsections along with selected results.

8.4.1 Selection of Distance and Construction of the Reference Distribution

After synchronization of the set of reference batches, one has a set of trajectories of equal duration and from them the average trajectory can be calculated. At each time interval, one can compute a distance between the average trajectory and any of the synchronized trajectories. Because of process disturbances and measurement noise, this quadratic distance is a random variable. If one could construct its probability distribution, then it would be possible to construct confidence intervals and use them to monitor the progress of a new batch. This is the main idea of the proposed monitoring scheme. Therefore, in order to implement it one has to answer two questions that are closely related: i) what quadratic distance to use and ii) how to construct its probability distribution.

The simplest option for the distance would be to use the same weighted quadratic distance used in the local distance computations of DTW; i.e., Eq (8.3). Because this type of distance uses a diagonal weight matrix, it penalizes only deviations of each variable from its average trajectory; changes in the correlations among the variables would not be considered. However, process faults are often revealed by the changes in the correlations among variables; each process variable may be well within its individual in-control limits, but still the process may be at fault. Therefore, a quadratic distance with a diagonal weight matrix may perform poorly as a fault detector.

On the other hand, one could use Hotelling's T^2 statistic (Kourti and MacGregor, 1995):

$$T^2 = \left[\mathbf{B}_{NEW,SC}(i,:) - \overline{\mathbf{B}}_{SC}(j,:)\right]\mathbf{S}_j^{-1}\left[\mathbf{B}_{NEW,SC}(i,:) - \overline{\mathbf{B}}_{SC}(j,:)\right]^T \qquad (8.4)$$

where $\overline{\mathbf{B}}_{SC}$ is the average trajectory, obtained from the synchronized trajectories and $\mathbf{S}_j$ is the estimated covariance matrix for the j^th^ data point of $\overline{\mathbf{B}}_{SC}$, estimated from the I observations of the synchronized trajectories; i.e., $\tilde{\mathbf{B}}_{i,SC}(j,:), i = 1,\ldots,I$. The problem with this statistic is that the $\mathbf{S}_j, j = 1,\ldots,b_{REF}$ matrices are very ill-conditioned because of highly correlated variables. By inverting these matrices, the very small eigenvalues of $\mathbf{S}_j$ (that generally represent process noise) dominate the value of T^2. In practice, small variations due to process noise in $\mathbf{B}_{NEW,SC}(i,:)$ will cause very large values of the T^2 statistic.

To circumvent this problem one could use Principal Component Analysis (PCA). After the synchronization, there are I realizations at each of the b_{REF} points in time: $\tilde{\mathbf{B}}_{i,SC}(j,:), i = 1,\ldots,I$. This would result in a set of b_{REF} PCA models. At each point in time, DTW could be used initially to synchronize the new batch against the average trajectory. Next, PCA could be used to assess the similarity of the current measurement of the new batch with the appropriate point (found by DTW) of the average trajectory.

Although the PCA solution seems the most promising of the three (diagonal weight matrix, Hotelling's T^2 statistic, PCA) the first alternative was chosen. The main purpose was to expose how one would implement such a scheme and not to study the effect of different distances on batch monitoring. Using the diagonal weight matrix is the simplest choice and the closest one to DTW. Furthermore, if one wants to use PCA for batch monitoring, the method of Nomikos and MacGregor (1994) combined with DTW (as discussed in Subsection 8.3.2) would be an easier alternative than a set of b_{REF} PCA models.

Therefore, the simple diagonal weight matrix is chosen. However, there is still another issue to be considered. As the synchronization results of the industrial example in Subsection 8.2.3 showed, after 10 iterations the 85% of the total weight in matrix **W** was allocated for Variable No. 5. Since this matrix **W** will be used to on-line synchronize the new batch with the average batch, Variable No. 5 will again control the synchronization. Most of the information that Variable No. 5 was carrying for purposes of fault detection, is lost after the synchronization. For that reason, it was decided to completely ignore Variable No. 5 in the fault detection step. The weight matrix to be used for fault detection, $\mathbf{W}_{FD}$, is obtained from **W** by setting the $\mathbf{W}(5,5)$ element to zero and normalizing the rest of the weights to 9 (since 9 is the number of the remaining variables).

After the selection of the type of distance used for monitoring, the problem of constructing the reference distribution must be addressed. If either Hotelling's T^2 statistic or PCA is used, one can construct approximate theoretical confidence intervals assuming multivariate normal distributions (for the T^2 statistic) and invoking the Central Limit Theorem (for PCA). Alternatively, if one does not want to make any assumptions about statistical distributions, computer-based methods can be used like bootstrap or jackknife. These methods require no theoretical calculations and can be used for statistics of any degree of complexity (Efron and Tibshirani, 1993). The bootstrap is a generalization of the jackknife and is more reliable for complex statistics. Thus, the method of bootstrap was chosen to generate the reference distribution of the weighted instantaneous distances.

According to the bootstrap method, one draws a number of random samples with replacement, called bootstrap samples, from the original sample. In this case, the original sample is the set of the I synchronized batches. One then would draw randomly and with replacement from this original sample and create another sample of the same size (i.e., of I batches). This new set of batches would be a bootstrap sample. Next, the average trajectory is computed from the bootstrap sample and for each of its members, its weighed Euclidean distance from the average trajectory is computed. Thus, for each member batch of the bootstrap sample, a time series of b_{REF} instantaneous distances is obtained. This procedure is repeated for a large number of bootstrap samples B; in this case, it was chosen to use $B = 500$. Note that for any bootstrap sample and for any point in time, the same $\mathbf{W}_{FD}$ weight matrix is used in the distance computations.

Once all B bootstrap samples have been generated, then for each of the b_{REF} points in time, one has a set of $I \cdot B$ (in this case $31 \cdot 500 = 15{,}000$) instantaneous weighted distances. The 5% of the largest distances (i.e., $0.05 \cdot 15{,}000 = 750$ in this case) are excluded and the largest distance of the remaining ones is used as the upper bound of the 95% confidence interval for the instantaneous distance. The result of this final step is a time series of b_{REF} distances, $\mathbf{d}_{FD,95\%}$, that represent the upper 95% confidence interval for the weighted instantaneous distance of a scaled batch trajectory from the scaled average trajectory. For this data set, the result is shown in Figure 8.7.

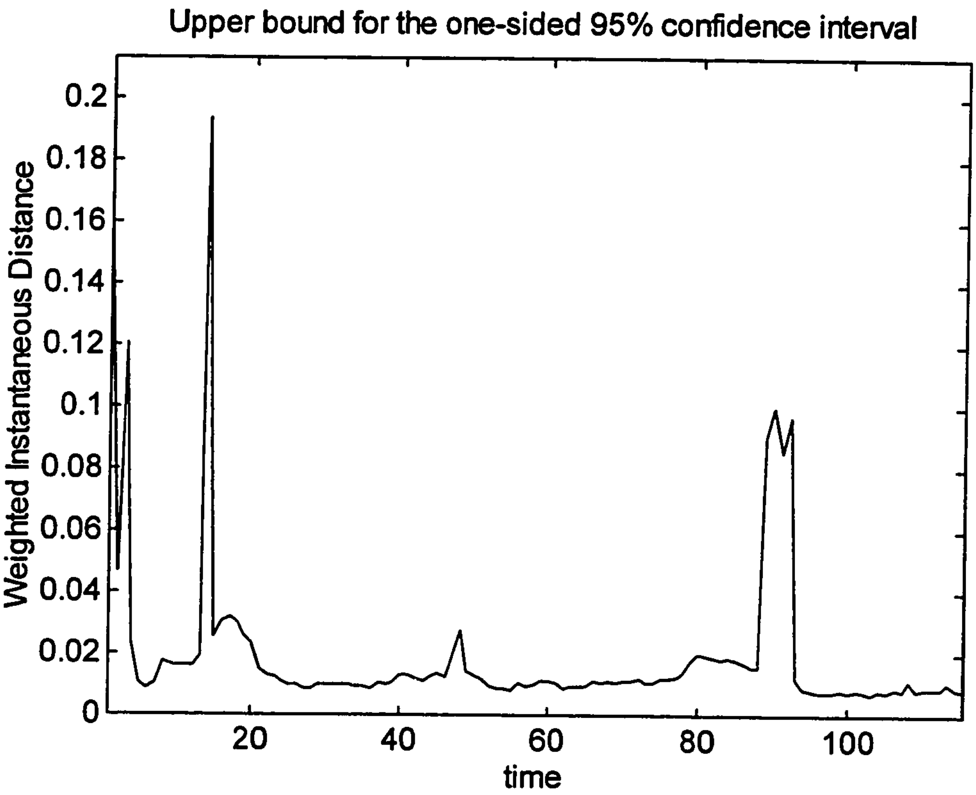


Figure 8.7: Upper bound of the 95% one-sided confidence interval for the instantaneous weighted distance; $B = 500$ bootstrap samples were used.

The two sharp peaks at approximately time 15 and time 95 result from the spikes that some of the variables exhibit at these times (see Figure 8.2). Similarly, the confidence intervals are wider at the beginning due to large variations of the reference set trajectories (see Figures I.4(a) and I.4(b) in the Appendix).

This completes the discussion on choosing an instantaneous distance to be used for fault detection and on constructing its confidence interval. However, before fault detection is applied, the new batch has to be synchronized on-line with the average trajectory. This is a problem for which DTW can provide an answer. The next subsection presents the DWT-based on-line batch monitoring scheme.

8.4.2 An On-line Batch Monitoring Method Using Dynamic Time Warping

The DTW variant chosen for on-line synchronization of the new batch against the average trajectory is very similar to the one presented for the on-line diagnosis of deterministic faults in Chapter 7. It is an asymmetric algorithm that maps the time axis of the new scaled trajectory (up to the current time) to the time axis of the average scaled trajectory. As such, the optimal path will pass through all the points of the new trajectory, but it may skip points of the average trajectory. Also, the relaxed end-points constraints are used as presented in Example 6 of Subsection 3.4.1.

To implement the algorithm, the time axis of the new trajectory is placed on the horizontal axis and the time axis of the average trajectory is placed on the vertical axis. The scaled average trajectory $\overline{\mathbf{B}}_{SC}$ and the weight matrix $\mathbf{W}$ (to be used in the DTW local distance computation) are obtained from the final iteration of the synchronization procedure presented in Subsection 8.2.2.

Let $\mathbf{B}_{NEW}(1,:)$ be the first measurement of the new trajectory; it is a row vector of dimension $1 \times N$ with N being the number of measured variables. As a first step, $\mathbf{B}_{NEW}(1,:)$ is scaled by dividing each variable by its range (which was estimated from the reference set batches). Let $\mathbf{B}_{\mathrm{NEW,SC}}(1,:)$ be the new scaled vector. Then, a set of local weighted distances is evaluated between $\mathbf{B}_{\mathrm{NEW,SC}}(1,:)$ and the first δ_2 points of $\overline{\mathbf{B}}_{\mathrm{SC}}$; i.e.,

$$d(1,\mathrm{j}) = \left[\mathbf{B}_{\mathrm{NEW,SC}}(1,:) - \overline{\mathbf{B}}_{\mathrm{SC}}(\mathrm{j},:)\right] \mathbf{W} \left[\mathbf{B}_{\mathrm{NEW,SC}}(1,:) - \overline{\mathbf{B}}_{\mathrm{SC}}(\mathrm{j},:)\right]^{\mathrm{T}} \text{ , } \mathrm{j} = 1,\ldots,\delta_2 \qquad (8.5)$$

The minimum accumulated total distances $\mathbf{D}_{\mathrm{A}}(1,\mathrm{j}), \mathrm{j} = 1,\ldots,\delta_2$ are set equal to these local distances; i.e.,

$$\mathbf{D}_{\mathrm{A}}(1,\mathrm{j}) = d(1,\mathrm{j}) \text{ , } \mathrm{j} = 1,\ldots,\delta_2 \qquad (8.6)$$

This allows the first point of the optimal path to be any of the points $(1,\mathrm{j}), \mathrm{j} = 1,\ldots,\delta_2$.

Now, let r_1 be the point where the minimum value of the $\mathbf{D}_{\mathrm{A}}(1,\mathrm{j})$ values occurs:

$$\mathrm{r}_1 = \arg\min_{\mathrm{j}}\left[\mathbf{D}_{\mathrm{A}}(1,\mathrm{j})\right] \text{ , } \mathrm{j} = 1,\ldots,\delta_2 \qquad (8.7)$$

Therefore, the $\mathrm{r}_1^{\mathrm{th}}$ point of the average trajectory, $\overline{\mathbf{B}}_{\mathrm{SC}}(\mathrm{r}_1,:)$, is deemed to be the most similar to the first point of the scaled new batch, $\mathbf{B}_{\mathrm{NEW,SC}}(1,:)$. To assess if indeed $\mathbf{B}_{\mathrm{NEW,SC}}(1,:)$ is a measurement from a good quality batch, its weighted distance from $\overline{\mathbf{B}}_{\mathrm{SC}}(\mathrm{r}_1,:)$ is computed using the $\mathbf{W}_{\mathrm{FD}}$ weight matrix; i.e.,

$$d_{\mathrm{FD}}(1,\mathrm{r}_1) = \left[\mathbf{B}_{\mathrm{NEW,SC}}(1,:) - \overline{\mathbf{B}}_{\mathrm{SC}}(\mathrm{r}_1,:)\right] \mathbf{W}_{\mathrm{FD}} \left[\mathbf{B}_{\mathrm{NEW,SC}}(1,:) - \overline{\mathbf{B}}_{\mathrm{SC}}(\mathrm{r}_1,:)\right]^{\mathrm{T}} \qquad (8.8)$$

and this value is compared with the $\mathrm{r}_1^{\mathrm{th}}$ point of the 95% confidence interval vector $\mathbf{d}_{\mathrm{FD,95\%}}$. If the value of $d_{\mathrm{FD}}(1,\mathrm{r}_1)$ is less than $\mathbf{d}_{\mathrm{FD,95\%}}(\mathrm{r}_1)$, then the first point of the new batch is deemed as originating from a good quality batch. Conversely, if $d_{\mathrm{FD}}(1,\mathrm{r}_1)$ is larger than $\mathbf{d}_{\mathrm{FD,95\%}}(\mathrm{r}_1)$, it is suspected that the new batch is of poor quality.

This procedure is repeated at each subsequent measurement. Let $\mathbf{B}_{\mathrm{NEW,SC}}(\mathrm{t},:)$ be the current scaled measurement of the new batch. The total accumulated distances between the new batch and the average batch are updated by the following DTW recursive relation:

$$\text{for } \mathrm{t} \in [\delta_1 + 2, \delta_3] : \mathbf{D}_{\mathbf{A}}(\mathrm{t},\mathrm{j}) = \min \begin{Bmatrix} \mathbf{D}_{\mathbf{A}}(\mathrm{t}-1,\mathrm{j}) + d(\mathrm{t},\mathrm{j}) \quad \text{or } \infty \text{ if Cond. (A)} \\ \mathbf{D}_{\mathbf{A}}(\mathrm{t}-1,\mathrm{j}-1) + d(\mathrm{t},\mathrm{j}) \\ \mathbf{D}_{\mathbf{A}}(\mathrm{t}-1,\mathrm{j}-2) + d(\mathrm{t},\mathrm{j}) \\ \mathbf{D}_{\mathbf{A}}(\mathrm{t}-1,\mathrm{j}-3) + d(\mathrm{t},\mathrm{j}) \end{Bmatrix}, \mathrm{j} = \mathbf{l}(\mathrm{t}), \ldots, \mathbf{u}(\mathrm{t}) \qquad (8.9a)$$

where: Condition (A):point $(\mathrm{t}-1,\mathrm{j})$ is optimally reached from point $(\mathrm{t}-3,\mathrm{j})$ via two consecutive horizontal moves.

$$\text{and for } \mathrm{t} \le \delta_1 + 1 \text{ or } \mathrm{t} \ge \delta_3 + 1: \quad \mathbf{D}_{\mathbf{A}}(\mathrm{t},\mathrm{j}) = \min \begin{Bmatrix} \mathbf{D}_{\mathbf{A}}(\mathrm{t}-1,\mathrm{j}) + d(\mathrm{t},\mathrm{j}) \\ \mathbf{D}_{\mathbf{A}}(\mathrm{t}-1,\mathrm{j}-1) + d(\mathrm{t},\mathrm{j}) \\ \mathbf{D}_{\mathbf{A}}(\mathrm{t}-1,\mathrm{j}-2) + d(\mathrm{t},\mathrm{j}) \\ \mathbf{D}_{\mathbf{A}}(\mathrm{t}-1,\mathrm{j}-3) + d(\mathrm{t},\mathrm{j}) \end{Bmatrix}, \mathrm{j} = \mathbf{l}(\mathrm{t}), \ldots, \mathbf{u}(\mathrm{t}) \qquad (8.9b)$$

where: $d(\mathrm{t},\mathrm{j}) = \left[\mathbf{B}_{\mathrm{NEW,SC}}(\mathrm{t},:) - \overline{\mathbf{B}}_{\mathrm{SC}}(\mathrm{j},:)\right] \mathbf{W} \left[\mathbf{B}_{\mathrm{NEW,SC}}(\mathrm{t},:) - \overline{\mathbf{B}}_{\mathrm{SC}}(\mathrm{j},:)\right]^{\mathrm{T}}$.

As Eq (8.9b) indicates, the check on consecutive horizontal moves is disengaged for the first $(\delta_1 + 1)$ points and after the δ_3 point of the new batch. This is done to account for the possibility that the new batch may contain more points from the initial and final stages than the reference set batches. The converse possibility (i.e., the new batch to contain less points from its initial and final stages than the reference set batches) is accounted by the relaxed endpoint constraints.

Again, let r_t be the point where the minimum value of the $\mathbf{D}_{\mathbf{A}}(\mathrm{t},\mathrm{j})$ values occurs:

$$\mathrm{r}_\mathrm{t} = \arg\min_{\mathrm{j}} \left[\mathbf{D}_{\mathbf{A}}(\mathrm{t},\mathrm{j})\right], \ \mathrm{j} = \mathbf{l}(\mathrm{t}), \ldots, \mathbf{u}(\mathrm{t}) \qquad (8.10)$$

Finally, the $d_{\mathrm{FD}}(\mathrm{t},\mathrm{r}_\mathrm{t})$ instantaneous weighted distance is evaluated:

$$d_{\mathrm{FD}}(t,r_t) = \left[\mathbf{B}_{\mathrm{NEW,SC}}(t,:) - \overline{\mathbf{B}}_{\mathrm{SC}}(r_t,:)\right]\mathbf{W}_{\mathrm{FD}}\left[\mathbf{B}_{\mathrm{NEW,SC}}(t,:) - \overline{\mathbf{B}}_{\mathrm{SC}}(r_t,:)\right]^{\mathrm{T}} \quad (8.11)$$

and it is compared with the r_t^{th} point of the 95% confidence interval vector $\mathbf{d}_{\mathrm{FD,95\%}}$.

When the final scaled measurement of the new batch becomes available, $\mathbf{B}_{\mathrm{NEW,SC}}(b_{\mathrm{NEW}},:)$, the minimum value of the $\mathbf{D}_{\mathrm{A}}(b_{\mathrm{NEW}},j)$ distances is located:

$$r_{b_{\mathrm{NEW}}} = \arg\min_{j}\left[\mathbf{D}_{\mathrm{A}}(b_{\mathrm{NEW}},j)\right],\ j = \mathbf{l}(b_{\mathrm{NEW}}),\ldots,\mathbf{u}(b_{\mathrm{NEW}}) \quad (8.12)$$

The point $(b_{\mathrm{NEW}}, r_{b_{\mathrm{NEW}}})$ is the final point of the optimal path. One can then travel backwards through the $b_{\mathrm{NEW}} \times b_{\mathrm{REF}}$ grid of (i,j) points as the indices of the optimal predecessors indicate and reconstruct the optimal path $\hat{\mathbf{F}}$. Therefore, constructing the optimal path is essentially an off-line operation, since it requires the final point of the new batch. Moreover, the set of points:

$$\hat{\mathbf{F}}_{\mathrm{ON\text{-}LINE}} = \left\{\ (1,r_1),(2,r_2),\ldots,(t,r_t),\ldots,(b_{\mathrm{NEW}},r_{b_{\mathrm{NEW}}})\ \right\} \quad (8.13)$$

can be viewed as the on-line approximation of the optimal path.

The two paths, although practically very similar (as the results of the next subsection will illustrate), in principle they could be quite different. As a matter of fact, the only point that is guaranteed to be the same in the two paths is the last one $(b_{\mathrm{NEW}}, r_{b_{\mathrm{NEW}}})$ since it is found by the same minimization; i.e., Eq (8.12). Any of the other points in $\hat{\mathbf{F}}_{\mathrm{ON\text{-}LINE}}$ is found by a minimization scheme that does not necessarily obey the DTW constraints; i.e., local continuity constraints and (even) monotonicity. For example, if $(t-1, r_{t-1})$ and (t, r_t) are two consecutive points of $\hat{\mathbf{F}}_{\mathrm{ON\text{-}LINE}}$, there is no guarantee that $0 \le r_t - r_{t-1} \le 3$. The point (t,r_t) is found by a minimization scheme, (i.e., Eq (8.10)), which does not consider where the point $(t-1,r_{t-1})$ lies on the grid. On the other hand, any two consecutive points of the true optimal path $\hat{\mathbf{F}}$ will obey these constraints, since they are imposed by the local continuity constraints of Eqs (8.9a) and (8.9b).

Practically, however, the two paths are almost identical. In any well-behaved batch process there exists variables that are smooth and they clearly indicate the evolution of the process in time. These variables are heavily weighted in the computation of the total accumulated distances $\mathbf{D}_A(t,j), j = \mathbf{l}(t),\ldots,\mathbf{u}(t)$ which capture the similarity of the evolving new batch with the average trajectory. Making a decision about the optimal path based on these distances, without the final point being known yet, is still a reasonable decision. The same practical approximation is also used in digital communication systems where the objective is to estimate the most probable state sequence of a discrete-time finite-state Markov process (Forney, 1973).

As a final implementation detail, the vectors **l** and **u** (i.e., the lower and upper limits for the index j of the interior iteration in the DTW algorithm) have to be specified. In all DTW applications discussed in the previous chapters, these limits were constructed in the beginning. However, this is not possible in the case of on-line batch monitoring simply because the duration of the new batch is not known beforehand. Therefore, the limits have to be created as the new batch evolves. As Eq (8.5) shows, for the first point of the new batch these limits are:

$$\mathbf{u}(1) = \delta_2,\ \mathbf{l}(1) = 1 \tag{8.14}$$

For all subsequent points the limits are defined as follows: let r_{t-1} be the point where the minimum of the $\mathbf{D}_A(t-1,j)$ distances occurs; i.e.,

$$r_{t-1} = \arg\min_j\left[\mathbf{D}_A(t-1,j)\right],\ j = \mathbf{l}(t-1),\ldots,\mathbf{u}(t-1) \tag{8.15}$$

Then, the upper and lower bounds for the next point of the new batch will be:

$$\mathbf{u}(t) = \min(r_{t-1} + \delta_2, b_{REF}),\ \mathbf{l}(t) = \max(r_{t-1} - \delta_2, 1) \tag{8.16}$$

This completes the presentation of the DTW-based monitoring method. It requires information only up to the current time without any predictions of the future behavior of the batch. Furthermore, every point in the new trajectory is tested against the average trajectory and no average of the points is used. On the negative side, only deviations from the average trajectory are penalized; any change in the correlations among the variables is not considered. This limitation can be removed by using a different PCA model at each time interval.

Another important characteristic of the method is that the decision on whether the new batch is good or bad is based only on the current measurement. Although all the previous measurements are used to synchronize the new batch with the average trajectory (through the total distances $\mathbf{D}_A(i,j)$), the fault detection uses only the information from the most current measurement. This is both a positive and a negative feature. It is positive because it allows for fast detection of a fault, since the previous good quality points do not affect the current decision. On the other hand, it does not utilize all the accumulated information to assess the quality of the batch up to the current time.

8.4.3 Case Studies and Results

The monitoring scheme presented in the previous subsection was applied on the reference set batches. However, since all 31 batches gave good quality product, a faulty batch was not available to test the proposed method. Therefore an artificial batch was constructed from Batch No. 19 by increasing the value of Variable No. 3 by 0.1 from time 50 to time 65. This was the only difference between this artificial new batch, $\mathbf{B}_{NEW,SC}$, and $\mathbf{B}_{19,SC}$. Figure 8.8 shows Variable No. 3 and three other variables for the new batch and the average batch.

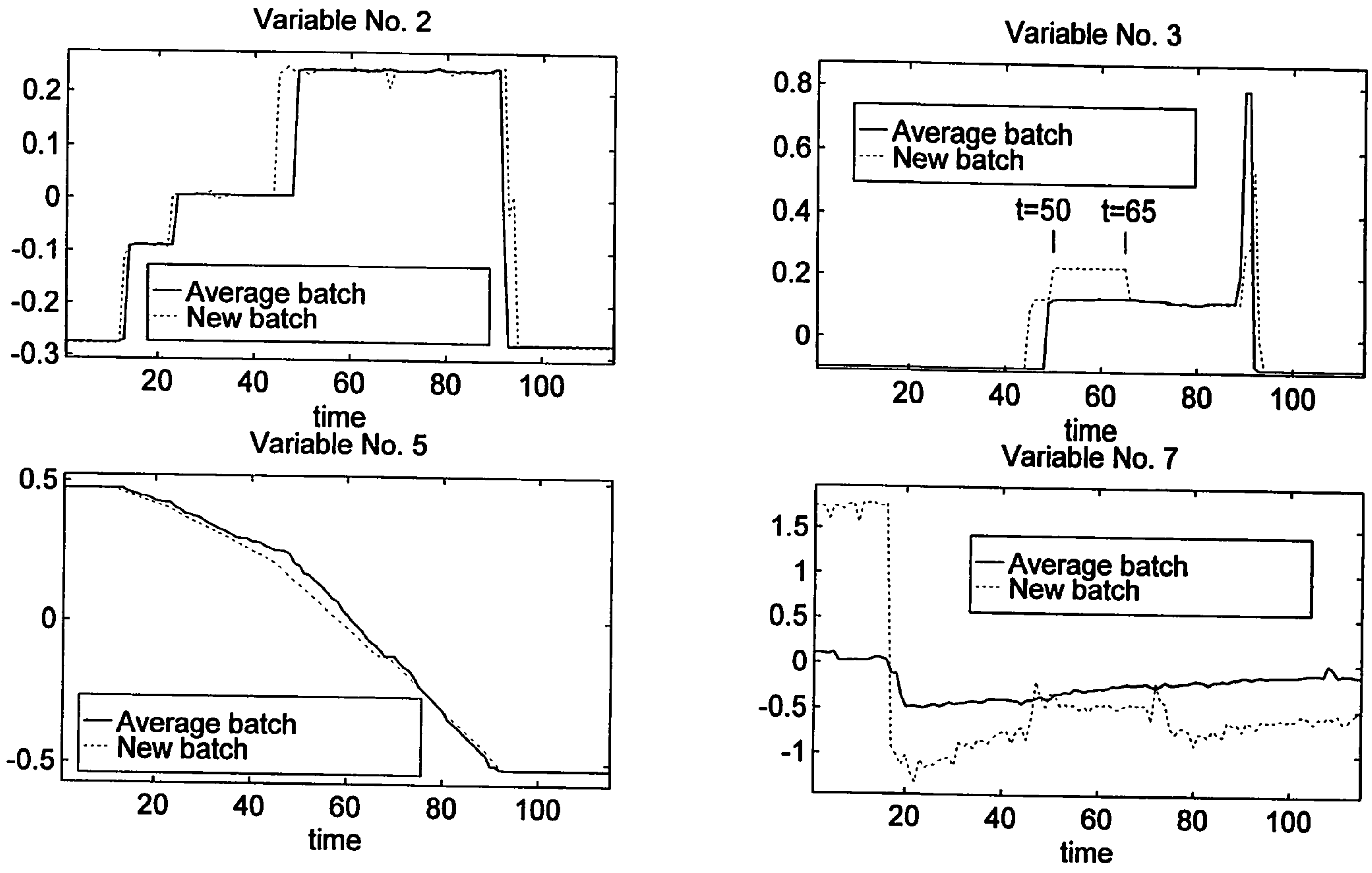


Figure 8.8: Behavior of 4 (out of 10) variables during the average batch and during a new batch. The average batch is the one obtained after 10 iterations. The new batch was obtained from Batch No. 19 by increasing the value of Variable No. 3 by 0.1 from time 50 to time 65.

The results from the monitoring method are shown in Figures 8.9 and 8.10, for Batch No. 23, $\mathbf{B}_{23,SC}$ and for the new batch $\mathbf{B}_{NEW,SC}$, respectively. The duration of $\mathbf{B}_{23,SC}$, $\mathbf{B}_{NEW,SC}$ and $\overline{\mathbf{B}}_{SC}$ was 118, 115 and 115 respectively. In all case studies the parameters in the DTW algorithm were given the following values:

$$\delta_1 = 30,\ \delta_2 = 20,\ \delta_3 = 70$$

The upper graph in Figure 8.9 shows the on-line approximation of the optimal path (dotted curve), the true optimal path obtained at the end (solid curve) and the on-line bounds for the inner iteration of the DTW algorithm (dashed curves). One can see that the on-line approximation of the optimal path agrees quite well with the true optimal path. The deviations between the two paths occur approximately between time 12 and 16 and between time 100 and 110. At any time in between these two points the two paths coincide. This can be attributed to the fact that between time 15 and 100 Variable No. 5 (which control the synchronization) is evolving. The accumulated distances capture the progress of this variable and as a result the on-line approximation of the optimal path is exact.

However, before time 12 and after time 100, Variable No. 5 remains constant. The synchronization of the new batch has to be done by considering other variables which are weighted much less (since Variable No. 5 gets approximately 85% of the total weight). Random noise in some of these variables results in small differences among many $\mathbf{D}_A(i,j)$ values that could all be candidates for points of the optimal path. These random differences then affect which point will be selected as approximation of the optimal path.

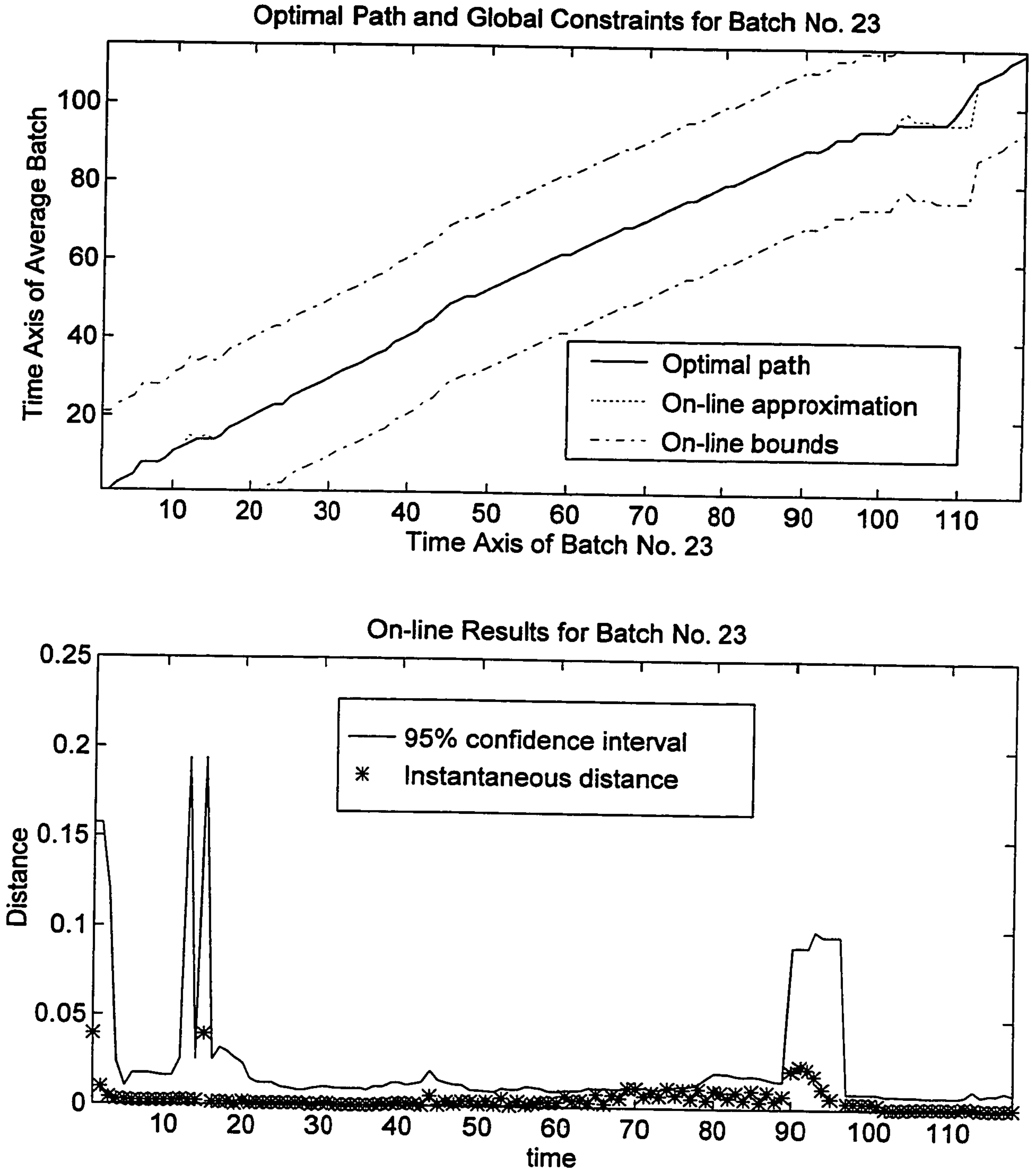


Figure 8.9: Optimal path (obtained off-line), on-line approximation of the optimal path and global constraints from monitoring Batch No. 23 (top graph). Also, local distances and their 95% confidence intervals from on-line monitoring for Batch No. 23 (bottom graph).

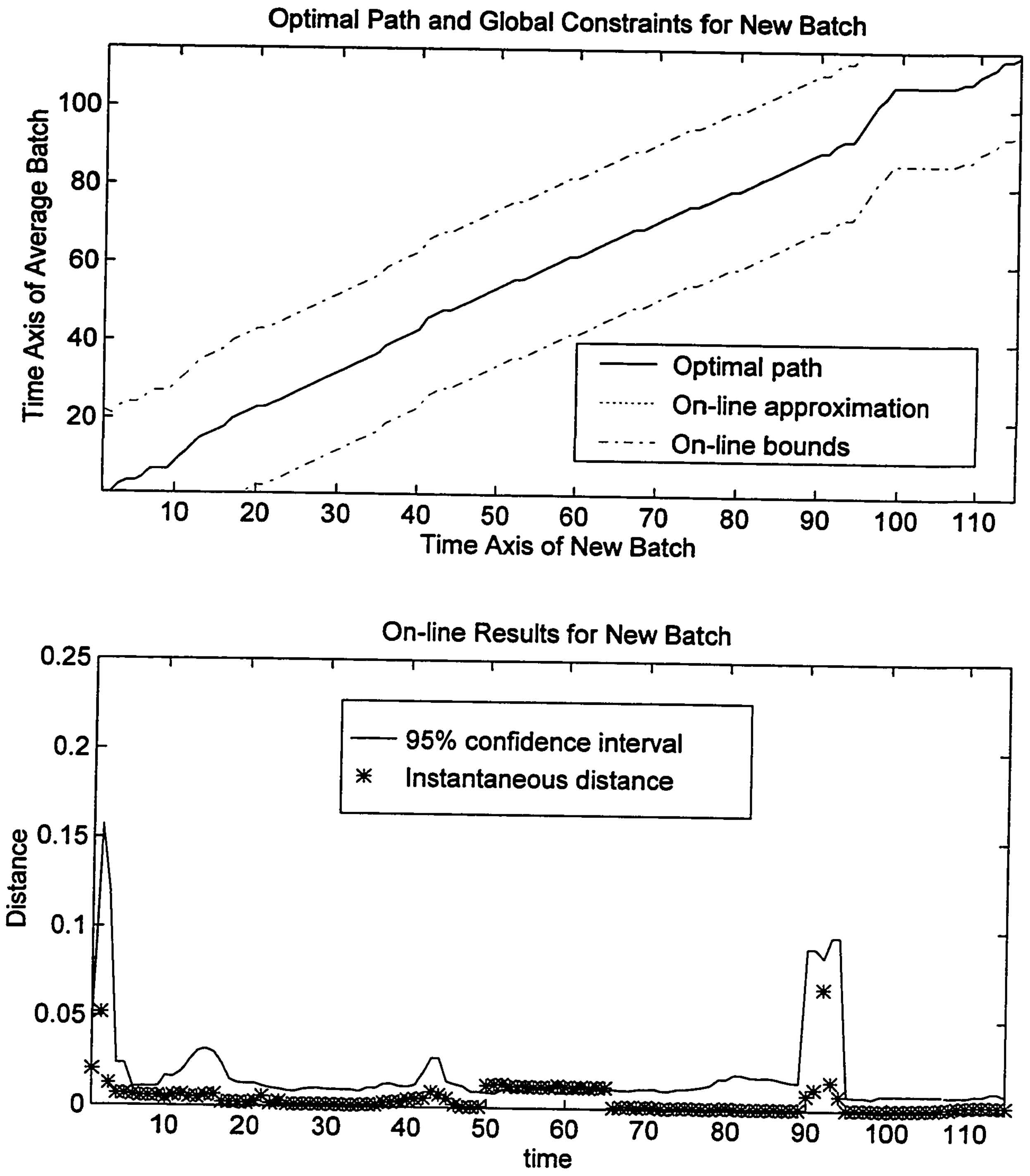


Figure 8.10: Optimal path (obtained off-line), on-line approximation of the optimal path and global constraints from monitoring the new batch (top graph); the on-line approximation agrees exactly with the optimal path. Also, local distances and their 95% confidence intervals from on-line monitoring for the new batch (bottom graph).

The problem could have been avoided if the total weight was distributed more evenly in more that one variable. It is for that reason that it is suggested to constrain the percentage of the total weight that a single variable can take during the synchronization procedure. In this particular case study, these two time periods correspond to idle conditions which do not affect the product quality.

The bottom graph in Figure 8.9 shows the results from monitoring Batch No. 23 on-line. As the graph shows, some distances between time 65 and 80 are close to their confidence intervals, but they never violate them. Based on this graph, the assessment would be that the is no adequate evidence to doubt that the batch is of good quality. This of course is the correct diagnosis, since Batch No. 23 was a good quality batch.

Figure 8.10 shows the results for the new batch. As the top graph in Figure 8.10, the on-line approximation of the optimal path agrees with the true optimal path at all times. The monitoring results are shown in the bottom graph. Between time 50 and 65, all distances are marginally larger than their confidence intervals. This persistent feature is an evidence that the new batch at these times exhibits a deviation from the average trajectory that cannot be attributed to random events. Therefore, there is a consistent fault that causes larger than normal variations in the new batch and one has some evidence to doubt the hypothesis that the batch is operating in a normal manner.

8.5 Chapter Summary

This chapter presented the application of Dynamic Time Warping in monitoring of batch processes. Batch processes have to go through a sequence of stages, not all of which are automated. As a result, batches are not synchronized and have different durations. However the synchronization of the trajectories to a common length is a necessary condition for the application of any monitoring scheme. To solve the problem of batch synchronization an iterative method was proposed based on DTW. The method is multivariate since it does not rely on a single variable to perform the synchronization in contrast to the indicator variable method. Moreover, the method pinpoints the most consistent variable, i.e., the variable with the smallest deviation about its average trajectory. This variable could be used as the indicator variable at subsequent studies if one wants to use this simpler approach.

Once the batch trajectories are synchronized, one can then build the MPCA/MPLS batch monitoring model. The second part of the chapter proposes an asymmetric DTW/synchronization method that can be used in conjunction with a MPCA/MPLS monitoring scheme. Details are given for both the off-line and the on-line implementation.

When used on-line, the MPCA/MPLS-based monitoring method requires the prediction of the future behavior of the new batch up to its end. The last part of this chapter presents a new method for on-line batch monitoring which does not require predictions. The method is based on the concept of the instantaneous distance of a new batch trajectory from the average trajectory. For the purposes of illustration, a simple weighted quadratic distance is chosen that does not penalize changes in the correlation among the variables. This is a major limitation of the method, since process faults often express themselves more clearly as changes in the correlation structure. The method essentially performs a hypothesis test at each time step by assessing the similarity of the current measurement to one point of the average trajectory. Previous data points to do not affect the assessment of whether or not the current measurement originates from a good quality batch. Therefore, the response of the method to process faults is immediate. Other features of the method include the use of bootstrap to construct the reference distribution of distances and again the use of DTW to synchronize the new trajectory with the average trajectory.

CHAPTER 9

SUMMARY, CONTRIBUTIONS AND EXTENSIONS

9.1 Thesis Summary

This thesis addressed several issues in the general area of Fault Detection and Diagnosis in industrial chemical processes. In the first part of the thesis, the problem of Fault Diagnosis in continuous petrochemical processes was investigated, while the second part studied the problem of Fault Detection in batch processes. The simulation of the Tennessee-Eastman plant with the control system of McAvoy and Ye (1994) (for continuous processes) and data from an industrial emulsion polymerization process (for batch processes) were used to illustrate each problem and evaluate the proposed solutions.

In the study of continuous processes, two types of faults were investigated: deterministic and stochastic. The cause of a deterministic fault was assumed to be a randomly occurring deterministic event, such as a step change in a feed composition. The underlying cause of a stochastic fault was assumed to be a random process (e.g., continuous random variations in a feed composition) and as such, different realizations of the same fault result in different patterns in the process variables. In both cases it was assumed that the faults are not directly measured but are observable through the deviations of the process variables from their steady-state values.

In this work, the Fault Diagnosis problem was approached from the point of view of Supervised Pattern Recognition. It was assumed that a reference set of past realizations of known faults was available and that each of these realizations produced a dynamic pattern in the process variables. Appropriate features were extracted from these patterns; the objective was to extract information that would identify each fault and distinguish it from the rest. In order to diagnose an unknown fault, these features were extracted from the process variables and the similarity (in the sense of a distance measure) was assessed against the features of each of the reference patterns. The known fault whose reference pattern has the minimum distance was deemed to be the most likely to have generated the dynamic pattern in the process variables.

Therefore, the investigation of the Fault Diagnosis problem in continuous processes (which is presented in the first part of the thesis) is an attempt to answer the following questions: I) what type of features have to be extracted from the raw measurements and II) how to assess the similarity between the patterns using the extracted features. These two questions had to be considered in conjunction with important constraining factors imposed by the type of faults and by the nature of continuous chemical processes.

The first of these factors is the differences between deterministic and stochastic faults. As mentioned above, deterministic faults are caused by randomly occurring deterministic events. As a result of this, different realizations of the same fault will produce similar patterns in the process variables. Therefore, the actual patterns that the process variables exhibit could be used to differentiate among several faults of this type. On the other hand, stochastic faults are caused by underlying random processes; consequently, the pattern of the process variables is different for each realization of the same fault. Therefore, the process variables cannot be used directly to diagnose stochastic faults; some other features are required. These features must be consistent over different realization of the same fault, while at the same time, being a distinctive signature of each fault.

Another important factor, for both deterministic and stochastic faults, is the magnitude of the fault. Any reference pattern is an expression of a fault realization of a certain magnitude. However, faults can occur with different magnitudes and a robust diagnostic scheme should be able to classify them correctly. Finally, the direction of a deterministic fault is another factor that has to be considered. A step up in a feed composition has to be followed by a step down at a later time. In general, the reference set may not contain the patterns of a fault occurring in both directions. The diagnostic scheme should be able to diagnose a fault of one direction using the information from a fault realization of the opposite direction.

In addition to all these fault characteristics that have to be addressed by a Fault Diagnosis scheme, the nature of chemical processes imposes some additional constraints. Chemical plants operate over a wide range of conditions in order to meet various demands and quality specifications. A fault can occur at any operating point, yet it has to be diagnosed correctly even if the reference set contains a fault realization at a different operating point. This is a strong requirement for a diagnostic scheme based on Pattern Recognition which does not use a mechanistic model of the process. For this thesis, a simpler Fault Diagnosis problem is addressed by considering operating points that are characterized only by different production levels and not by different product qualities. This implies that the correlation structure among the variables remains roughly the same for fault realizations at different operating points. However, the temporal correlations of the variables do change; different flowrates result in different dead times and time constants, thus affecting the speed of the dynamic response. The diagnostic scheme should be flexible enough to correctly classify a fault even when it exhibits faster or slower temporal correlations.

Besides Fault Diagnosis in continuous processes, similar problems are encountered in Speech Recognition and particularly in Isolated Word Recognition. The same word can be uttered with different duration and intensity, in different environments, and by different speakers; yet the Speech Recognition system should be able to classify it correctly independently of these variations. A major part of the Speech Recognition research has concentrated on the type of features to be extracted from speech signals; these are nonstationary high frequency signals and different from the outputs produced by a chemical process. However, even when the correct features are extracted, the problem of a flexible pattern matching scheme still remains. Dynamic Time Warping (DTW) is a flexible pattern matching method which works with pairs of patterns and is able to translate, compress, and expand the patterns so that similar features are matched. Moreover, DTW is flexible enough to accommodate the special characteristics of a particular pattern comparison. The most important of them is the uncertainty in locating the time origin and the end of a speech signal (or for this thesis, the origin and end of a fault).

Therefore, DTW appeared to be a promising solution to the similarity assessment problem between faults since it can perform pattern matching that is robust to I) the plant's production level and II) possible uncertainties in locating the fault's time origin. However, the problem of feature extraction still remained: what features should be extracted for each kind of fault so that the diagnosis is independent to the magnitude and the direction of the fault ? This question is answered in Chapters 4, 6 and 7 for the class of deterministic faults and in Chapters 5 and 6 for the class of stochastic faults.

Chapter 4 presents a complete method for the off-line diagnosis for deterministic faults. In the feature extraction step, a scaling procedure is applied to the raw measurements: their initial values are subtracted, they are then filtered by a first order high-pass filter, next they are normalized to standard deviation of one and finally they are filtered with a first order low-pass filter. This is done for all the reference patterns and for the new unknown pattern. The proposed scaling procedure tries to make the diagnosis independent of the magnitude of the fault and of the plant production level by removing the level and the magnitude information from the raw measurements. High-pass filtering removes low frequency components (e.g., steps) since small differences in the magnitude of these components result in large changes in the similarity metric. After scaling, the similarity of the new scaled pattern with each of the reference scaled patterns and their mirror images is evaluated using DTW. The fault whose scaled pattern results in the minimum distance is deemed to be the most likely to have generated the new pattern. The results from the case studies were promising: all diagnoses were correct, they were in agreement with intuitive expectations and showed that the proposed method could be a practical tool for the problem of deterministic Fault Diagnosis in continuous processes.

In Chapter 5 a method is proposed for the diagnosis of stochastic faults in continuous processes. The method can be used for both off-line and on-line applications. In the feature extraction step, a high-pass filtering operation is used to remove the low frequency components and then autocorrelation and crosscorrelation coefficients are estimated. These features reflect both the process dynamics and the relative behavior of the variables. Most importantly, they remain constant over different realizations of the same fault. Moreover, they are magnitude independent and this satisfies the important requirement of a magnitude independent fault diagnosis. In the decision step, a DTW variant is used to assess the similarity between two correlation patterns. The use of DTW is dictated because changes in the plant production level change the temporal behavior of the correlation patterns. Another advantage of the proposed method is that it can be used in on-line applications without any modifications. On the other hand, the number of correlation coefficients grows very fast with the number of variables and the result is a much higher dimensional feature space. The results from the case studies were inconclusive. Although all the diagnoses were correct, the results did not agree with intuitive expectations. For example, faults with different magnitudes were classified with more certainty than faults identical to the reference set faults. A possible cause for these results could be the slow process dynamics (induced by the recycle streams). This meant that more time was required for the process to exhibit a consistent correlation pattern than what was considered to be reasonable for timely fault diagnosis.

Chapter 6 addresses the problem of large dimensionality in the feature space by proposing the use of Principal Component Analysis (PCA) as a means for feature extraction. In the case of deterministic patterns, one PCA model was constructed from all reference patterns after they had been filtered and normalized. The principal components for each pattern are now its description in a lower dimension feature space. The new unknown pattern is similarly scaled, then passed through the PCA model and its lower dimension description is obtained. Finally its principal components are compared using DTW with the ones of the reference patterns and their mirror images. The improvement over the results obtained in Chapter 4 was significant. This could be attributed to the fact the PCA retains only variation that is consistent among the measurements.

In the case of stochastic patterns, a similar extension based on PCA was proposed. Again, one PCA model was constructed from all the reference patterns, after they had been filtered. For each stochastic pattern, the correlation pattern of the principal components was obtained. When the pattern of a new unknown fault appears, it is similarly scaled, then is passed through the PCA model and the correlation pattern of its principal component is computed. Finally, DTW is used to assess the similarity between correlation patterns. The results were similar to the ones obtained in Chapter 5 (i.e., they contradicted intuitive expectations regarding the diagnosis of faults that are different realizations of the reference set faults). Nonetheless, the reduction in the dimension of the correlation patterns was significant.

The last part of Chapter 6 discusses the inadequacy of PCA when used for classification of dynamic signals. In this thesis, PCA was only used as a means of feature extraction, but not for pattern classification. The latter task was given to DTW which is a flexible method for comparison of patterns. DTW can deal with unsynchronized patterns and distorted temporal correlations caused by different production levels and changing fault magnitudes. On the other hand, PCA is a rigid classifier for dynamic patterns and as shown by the results, is very likely to fail when faced with different production levels and fault magnitudes.

Chapter 7 studies the on-line diagnosis of deterministic faults in continuous processes. The method is very similar to the off-line method of Chapter 4; i.e., it is a sequence of a filtering and a normalization step, followed by a similarity assessment step using DTW. However, instead of making a decision at each time interval about the diagnosis of a new pattern, a set of times is selected where similarity assessment via DTW takes place. If a fault always occurred with the same magnitude, one could use the scaling factors from the reference patterns to normalize the standard deviation of the variables of a new fault. In such a case, a similarity assessment step based on DTW could provide a diagnosis at each time interval. However, faults can occur with different magnitudes and so the normalization of their patterns using the reference scaling factors will result in erroneous values for the normalized patterns. DTW will fail since it is a distance-based method and is therefore very sensitive to scaling. The method proposed in Chapter 7 addresses this problem by applying DTW only after enough information has been gathered to estimate the scaling factors of the new pattern. The results were promising and agreed with intuition.

Chapter 8 studied the problem of fault detection in batch processes. Batch processes play an important role in the production of high added value products, such as specialty polymers, pharmaceuticals and biochemical materials. Therefore there exists a large economic incentive to manufacture consistent, good quality product. Nomikos and MacGregor (1994) have proposed a method based on Multiway Principal Component Analysis (MPCA) for monitoring batch processes using the readily measured process variables. One requirement for the successful implementation of their method is that the batch trajectories have the same duration and are synchronized. However, batch processes are often a sequence of separate stages and some of them may not be automated, but left to discretion of operators. The result is a set of trajectories that have unequal durations, are not synchronized, yet they represent batch runs that produced good quality product.

Nomikos and MacGregor addressed this problem by finding an indicator variable that can uniquely reflect the progress of the batch process and then synchronizing the trajectories with respect to this variable. However this solution assumes the existence of such a variable and the expert process knowledge to identify it among the many measured process variables. The first part of Chapter 8 proposes a new method for trajectory synchronization based on DTW. The method is multivariate since it does not rely on a single variable for the synchronization. Moreover, through an iterative procedure the method identifies the most consistent variables; i.e., variables with small squared deviations about their average trajectory. The variable with the smallest deviation could then be used as the indicator variable at subsequent studies, if one wants to use this simpler approach. Once the trajectories are synchronized, the MPCA/MPLS-based model for batch monitoring can be constructed. The method was implemented on an industrial data set and the results illustrated its practical usefulness.

To assess the quality of a new batch while the batch run is still in progress, one has to synchronize the new batch trajectory with the reference trajectories in real time. This is discussed in the second part of Chapter 8, where it is shown how one can use DTW to perform this on-line batch monitoring.

Finally, the last part of Chapter 8 presents a new method for on-line batch monitoring which does not require any predictions for the future behavior of the batch. The method is based on the concept of instantaneous weighted quadratic distance between a new batch trajectory and the average trajectory. Because the method uses instantaneous distances, it does not require any predictions and its response to process fault is immediate. The bootstrap method is used to generate the reference distributions of distances; therefore no assumption about the probability distributions is required. On the other hand, the quadratic distance does not penalize changes in the correlation among the variables. This is a major limitation of the method, since process faults often express themselves as changes in the correlation structure. A possible solution to this problem could be the use of different PCA models at each time interval.

9.2 Thesis Contributions

A new method was proposed for the off-line diagnosis of deterministic faults in continuous, dynamic, multivariable chemical processes. The method is designed to diagnose faults independently of their magnitude, direction, plant production level, and uncertainty in the time origin of a fault. The method is based on Pattern Recognition principles; it consists of a scaling procedure where magnitude invariant features are extracted, followed by a similarity assessment step where Dynamic Time Warping is used as a flexible pattern comparison scheme. Also, an on-line implementation of the method was proposed which takes into consideration the difficulties associated with scaling the evolving pattern of a fault in real time.

In addition to the diagnosis of deterministic faults, the problem of diagnosing stochastic faults was also investigated. A new method was proposed for stochastic fault diagnosis which can be implemented both on-line and off-line. It is also based on Pattern Recognition principles and tries to diagnose faults independently of their magnitude and of the plant production level. The method uses the correlation pattern of a stochastic fault as the feature and then a specially designed Dynamic Time Warping algorithm is used to assess the similarity between correlation patterns.

The methods for the off-line diagnosis of both deterministic and stochastic faults were extended by including Principal Component Analysis as an additional step in the feature extraction stage. In the case of deterministic faults, significant improvement was observed in the discriminatory power of the classifier. The result were inconclusive for stochastic faults; however, the large reduction in the dimension of the feature space was a major benefit. The limitations that Principal Component Analysis faces when is used for the classification of dynamic patterns were also illustrated.

In the study of batch processes, a new multivariate method was proposed to equalize the duration and synchronize the events in batch trajectories. The method is an iterative procedure where a Dynamic Time Warping algorithm and a scaling scheme are used to produce a set of synchronized trajectories. These trajectories could then be used to construct the batch monitoring scheme of Nomikos and MacGregor (1994). Also, it was shown how to use Dynamic Time Warping to synchronize a new trajectory with the reference trajectory for both off-line analysis and for real time applications. Finally, a new fault detection method was proposed, based on the concept of the instantaneous distance of a new trajectory from the average trajectory, which does not require any predictions about the future behavior of the new batch.

9.3 Recommendations for Future Work

The results obtained from the diagnosis of stochastic faults in continuous processes were inconclusive. Some further investigation could be carried out to determine the time duration of data collection that is required for an accurate correlation pattern estimation for processes with significant recycle flowrates. This also begs the question whether there is any other feature that can be used to reliably classify stochastic patterns without requiring an unrealistically large duration of the fault.

In all the methods presented in the thesis, the extracted features were common for all patterns in their respective classes. For example, all patterns of deterministic faults were similarly normalized and filtered; also in Chapter 6, the same Principal Component Analysis model was used for all of the faults. A more powerful alternative would to extract features from each pattern individually; e.g., construct a different PCA model for each pattern. Next, a reference distribution of distances could be constructed for each fault, assuming that a sufficient number of fault realizations existed. The pattern of an unknown fault would then be compared with each fault in its respective feature space and the final distance measure would be assessed against the respective reference distribution. This would be a more powerful fault diagnosis scheme since each fault defines its own features. However, a large number of fault realizations would be required, but this may be possible given that continuous processes operate over long periods of time.

Finally, the method presented in this thesis would be best utilized within the framework of an Expert System. In industrial processes, fault diagnosis is a more complicated task than pattern classification. It requires the examination of a series of important issues that have to be addressed before fault diagnosis is performed, e.g., sensor validation, data reconciliation, fault detection, determination of whether a fault would be treated as deterministic or stochastic. This thesis tried to solve only a part of this large problem. The combination of effective solutions for each particular problem is required and this can be best handled in an Expert System framework.

In the area of on-line monitoring of batch processes, instead on using the simple weighted Euclidean distance for fault detection, one could use different PCA models at each time step. This modification will allow for the detection of faults that express themselves as changes in the correlation structure. However, such a method still does not consider the time history of the batch since each PCA model is constructed from data taken at a single time step. One could include the behavior of the batch over time by including data from a small number of past time steps. At each time, Dynamic Time Warping could be used to locate the points in time (of the reference trajectories) whose data can be used to construct the PCA models. By doing this one takes advantage of: I) the synchronization capability of DTW, II) the local time history of the batch, and III) the capability of PCA to detect changes in the correlation structure and to locate which variables are responsible for these changes. Also, this method would not require any averaging of the points in the new batch. Of course, more computational power would be required to implement such a method in real time; however, this tends to be less of an issue with the development of faster and cheaper computers.

BIBLIOGRAPHY

Bakshi, B.R., and G. Stephanopoulos, “Wave-Net: a Multiresolution, Hierarchical Neural Network with Localized Learning”, *AIChE Journal*, **39**(1), 57-81 (1993).

Bakshi, B.R., and G. Stephanopoulos, “Representation of Process Trends-III. Multi-Scale Extraction of Trends from Process Data”, *Computers and Chemical Engineering*, **18**(4), 267-302 (1994a).

Bakshi, B.R., and G. Stephanopoulos, “Representation of Process Trends-IV. Induction of Real-Time Patterns from Operating Data for Diagnosis and Supervisory Control”, *Computers and Chemical Engineering*, **18**(4), 303-332 (1994b).

Bellman, R.E., and S.E. Dreyfus, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1962.

Bertsekas, D.P., *Dynamic Programming*, Prentice-Hall, Inc., Englewood Cliffs, NJ,1987.

Box, G.E., and G.C. Tiao, “A Canonical Analysis of Multiple Time Series”, *Biometrika*, **64**(2), 355-365 (1977).

Chang, I.C., C.C. Yu, and C.T. Liou, “Model-Based Approach for Fault Diagnosis. 1. Principles of Deep Model Algorithm”, *Industrial and Engineering Chemistry and Research*, **33**, 1542-1555 (1994).

Cheung, J.T., and G. Stephanopoulos, “Representation of Process Trends. Part I. Formal Representation Framework”, *Computers and Chemical Engineering*, **14**, 495-510 (1990a).

Cheung, J.T., and G. Stephanopoulos, “Representation of Process Trends. Part II. The Problem of Scale and Qualitative Scaling”, *Computers and Chemical Engineering*, **14**, 511-540 (1990b).

Chow, E.Y., and A.S. Willsky, "Analytical Redundancy and the Design of Robust Failure Detection Systems", *IEEE Transactions on Automatic Control*, **AC-29**, 603-614 (1984).

Cooper, D.J., L. Megan, and R.F. Hinde, "Comparing Two Neural Networks for Pattern Based Adaptive Process Control", *AIChE Journal*, **38**(1), 41-55 (1992).

Dayal, B.S., J.F. MacGregor, P.A. Taylor, R. Kildaw, and S. Marcikic, "Application of Feedforward Neural Networks and Partial Least Squares Regression for Modelling Kappa Number in a Continuous Kamyr Digester", *Pulp & Paper Canada*, **95**(1), 26-32 (1994).

Downs, J.J., and E.F. Vogel, "A Plant-Wide Industrial Process Control Problem", *Computers and Chemical Engineering*, **17**(3), 245-255 (1993).

Dunia, R., S.J. Qin, T.F. Edgar, and T.J. McAvoy, "Use of Principal Component Analysis for Sensor Fault Identification", *Computers and Chemical Engineering*, **20**, Supplement A, S713-S718 (1996).

Efron, B., and R.J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall, Inc., 1993.

Fan, J.Y., M. Nikolaou, and R.E. White, "An Approach to Fault Diagnosis of Chemical Processes via Neural Networks", *AIChE Journal*, **39**(1) 82-88 (1993).

Fathi, Z., W.F. Ramirez, and J. Korbicz, "Analytical and Knowledge-Based Redundancy for Fault Diagnosis in Process Plants", *AIChE Journal*, **39**(1), 42-56 (1993).

Forney Jr., G.D., "The Viterbi Algorithm", *Proceeedings of the IEEE*, **61**(3), 268-278 (1973).

Frank, P.M, "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy - A Survey and Some New Results", *Automatica*, **26**(3), 459-474 (1990).

Franklin, G.F., and J.D. Powell, *Digital Control of Dynamic Systems*, Addison-Wesley Publishing Company, Inc., 1980.

Geladi, P., and B.R. Kowalski, "Partial Least Squares", *Analytica Chimica Acta*, **185**, 1-17 (1986).

Gertler, J.J., "Survey of Model-Based Failure Detection and Isolation in Complex Plants", *IEEE Control Systems Magazine*, 3-11 (1988).

Golub, G.H., and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1989.

Guglielmi, G., T. Parisini, and G. Rossi, "Keynote Paper: Fault Diagnosis and Neural Networks: A Power Plant Application", *Control Engineering Practice*, **3**(5), 601-620 (1995).

Hodouin, D., J.F. MacGregor, M. Hou, and M. Franklin, "Multivariate Statistical Analysis of Mineral Processing Plant Data", *CIM Bulletin*, **86**, 975, 23-34 (1993).

Hoskins, J.C., K.M. Kaliyur, and D.M. Himmelblau, "Fault Diagnosis in Complex Chemical Plants Using Artificial Neural Networks", *AIChE Journal*, **37**(1), 137-141 (1991).

Hush, D.R., and B.G. Horne, "Progress in Supervised Neural Networks", *IEEE Signal Processing Magazine*, **10**, 8-39 (1993).

Isermann, R., "Process Fault Detection Based on Modeling and Estimation Methods - A Survey", *Automatica*, **20**(4), 387-404 (1984).

Isermann, R., "Fault Diagnosis of Machines via Parameter Estimation and Knowledge Processing - Tutorial Paper", *Automatica*, **29**(4), 815-835 (1993).

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

Jenkins, G. M., and D. G. Watts, *Spectral Analysis and its Applications,* Holden-Day, Inc., San Fransisco, 1968.

Jolliffe I.T., *Principal Component Analysis*, Springer-Verlag, New York, 1986.

Juang, B.H., and L.R. Rabiner, "Hidden Markov Models for Speech Recognition", *Technometrics*, **33**(3), 251-272 (1991).

Kavuri, S.N., and V. Venkatasubramanian, "Using Fuzzy Clustering With Ellipsoidal Units in Neural Networks for Robust Classification", *Computers and Chemical Engineering*, **17**(8) 765-784 (1993).

King, R., and E.D. Gilles, "Multiple Filter Methods for Detection of Hazardous States in an Industrial Plant", *AIChE Journal*, **36**(11), 1697-1706 (1990).

Konstantinov, K.B., and T. Yoshida, "A Knowledge-Based Pattern Recognition Approach for Real-Time Diagnosis and Control of Fermentation Processes as Variable Structure Plants", *IEEE Transactions on Systems, Man, and Cybernetics*, **21**(4), 908-914 (1991).

Konstantinov, K.B., and T. Yoshida, "Real-Time Qualitative Analysis of the Temporal Shapes of (Bio)process Variables", *AIChE Journal*, **38**(11), 1703-1715 (1992).

Kourti, T., and J.F. MacGregor, "Process Analysis, monitoring and diagnosis, using multivariate projection methods", *Chemometrics and Intelligent Laboratory Systems*, **28**, 3-21 (1995).

Kourti, T., P. Nomikos and J.F. MacGregor, "Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS", *Journal of Process Control*, **5**(4), 277-284 (1995).

Kramer, M.A., "Malfunction Diagnosis Using Quantitative Models with Non-Boolean Reasoning in Expert Systems", *AIChE Journal*, **33**(1), 130-140 (1987).

Kramer, M.A., and J.A. Leonard, "Diagnosis Using Backpropagation Neural Networks - Analysis and Criticism", *Computers and Chemical Engineering*, **14**(2), 1323-1338 (1990).

Kresta, J.V., J.F. MacGregor, and T.E. Marlin, "Multivariate Statistical Monitoring of Process Operating Performance", *Canadian Journal of Chemical Engineering*, **69**, 35-47 (1991).

Ku, W., R.H. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis", *Chemometrics and Intelligent Laboratory Systems*, **30**, 179-196 (1995).

Leonard, J.A., and M.A. Kramer, "Radial Basis Function Networks for Classifying Process Faults", *IEEE Control Systems Magazine*, **31**, 31-38 (1991).

Levinson, S.E., L.R. Rabiner, A.E. Rosenberg, and J.G. Wilpon, "Interactive Clustering Techniques for Selecting Speaker Independent Reference Templates for Isolated Word Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-27**(2), 134-141 (1979).

Lippmann, R.P., "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, **4**, 4-22 (1987).

MacGregor, J. F., C. Jaeckle, C. Kiparissides, and M. Koutoudi, "Monitoring and Diagnosis of Process Operating Performance by Multi-block PLS Methods with an Application to Low Density Polyethylene Production", *AIChE Journal*, **40**(5), 826-838 (1994).

McAvoy, T.J., and N. Ye, "Base Control for the Tennessee Eastman Problem", *Computers and Chemical Engineering*, **18**(5), 383-413 (1994).

Megan, L., and D.J. Cooper, "A Neural Network Strategy for Disturbance Pattern Classification and Adaptive Multivariable Control", *Computers and Chemical Engineering*, **19**(2), 171-186 (1995).

Mehra, R.K., and J. Peschon, "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems", *Automatica*, **7**, 637-640 (1971).

Myers, C., L.R. Rabiner, and A.E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing,* **ASSP-28**(6), 623-635 (1980).

Nadler, M., and E.P. Smith, *Pattern Recognition Engineering*, John Wiley & Sons, 1993.

Naidu, S.R, E. Zafiriou, and T.J. McAvoy, "Use of Neural Network for Sensor Failure Detection in a Control System", *IEEE Control Systems Magazine*, **10**, 49-55 (1990).

Ney, H., "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing,* **ASSP-32**(2), 263-271 (1984).

Nomikos, P., and J.F. MacGregor, "Monitoring Batch Processes Using Multiway Principal Component Analysis", *AIChE Journal*, **40**, 1361-1375 (1994).

Nomikos, P., and J.F. MacGregor, "Multivariate SPC Charts for Monitoring Batch Processes", *Technometrics*, **37**(1), 41-59 (1995a).

Nomikos, P., and J.F. MacGregor, "Multi-way Partial Least Squares in Monitoring Batch Processes", *Chemometrics and Intelligent Laboratory Systems*, **30**, 97-108 (1995b).

Oppenheim, A.V., and R.W. Schafer, *Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, NJ,1975.

O'Shaughnessy, D., "Speaker Recognition", *IEEE ASSP Magazine*, **3**, 4-17 (1986).

Petti, T.F., J. Klein, and P.S. Dhurjati, "Diagnostic Model Processor: Using Deep Knowledge for Process Fault Diagnosis", *AIChE Journal*, **36**(4), 565-575 (1990).

Picone, J., "Continuous Speech Recognition Using Hidden Markov Models", *IEEE ASSP Magazine*, **7**, 26-41 (1990).

Rabiner, L.R., and S.E. Levinson, "Isolated and Connected Word Recognition - Theory and Selected Applications", *IEEE Transactions on Communications*, **COM-29**(5), 621-659 (1981).

Rabiner, L.R., S.E. Levinson, and M.M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition", *The Bell System Technical Journal*, **62**(4), 1075-1105 (1983).

Rabiner, L.R., A.E. Rosenberg, and S.E. Levinson, "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-26**(6), 575-582 (1978).

Raich, A., and A. Cinar, "Statistical Process Monitoring and Disturbance Isolation in Multivariate Continuous Processes", *Proceedings of ADCHEM '94*, 452-475, Kyoto, 1994.

Rich, S.H., and V. Venkatasubramanian, "Model-Based Reasoning in Diagnostic Expert Systems for Chemical Process Plants", *Computers and Chemical Engineering*, **11**(2), 111-122 (1987).

Ricker, N.L., "Optimal Steady-State Operation of the Tennessee Eastman Challenge Process", *Computers and Chemical Engineering*, **19**(9), 949-959 (1995).

Ricker, N.L., "Decentralized Control of the Tennessee Eastman Challenge Process", *Journal of Process Control*, **6**(4), 205-221 (1996).

Ricker, N.L., and J.H. Lee, "Nonlinear Model Predictive Control of the Tennessee Eastman Challenge Process", *Computers and Chemical Engineering*, **19**(9), 961-981 (1995a).

Ricker, N.L., and J.H. Lee, "Nonlinear Modeling and State Estimation for the Tennessee Eastman Challenge Process", *Computers and Chemical Engineering*, **19**(9), 983-1005 (1995b).

Robertson, D., and J.H. Lee, "Integrated State Estimation, Fault Detection and Diagnosis for Nonlinear Systems", *Proceedings of the ACC*, 389-392, San Francisco, 1993.

Sakoe, H., and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", *IEEE Transactions on Acoustics, Speech and Signal Processing,* **ASSP-26**(1), 43-49 (1978).

Shum, S.K., J.F. Davis, W.F. Punch III, and B. Chandrasekaran, "An Expert System Approach to Malfunction Diagnosis in Chemical Plants", *Computers and Chemical Engineering*, **12**(1), 27-36 (1988).

Silverman, H.F., and D.P. Morgan, "The Application of Dynamic Programming to Connected Speech Recognition", *IEEE ASSP Magazine*, **7**, 7-25 (1990).

Slama, C., *Multivariate Statistical Analysis of Data from an Industrial Fluidized Catalytic Cracking Process Using PCA and PLS*, M. Eng. Thesis, McMaster University, Canada,1992.

Sorsa, T., H. Koivisto, and H.N. Koivo, "Neural Networks in Process Fault Diagnosis", *IEEE Transactions on Systems, Man, and Cybernetics*, **21**(4), 815-825 (1991).

Sorsa, T., and H.N. Koivo, "Application of Artificial Neural Networks in Process Fault Diagnosis", *Automatica*, **29**(4), 843-849 (1993).

The Mathworks, Inc., *Signal Processing Toolbox for Use with Matlab™ User's Guide*, MA, 1988.

Tou, J.T., and R.C. Gonzalez, *Pattern Recognition Principles*, Addison-Welsley, Massachusetts, 1974.

Venkatasubramanian, V., and K. Chan, "A Neural Network Methodology for Process Fault Diagnosis", *AIChE Journal*, **35**(12) 1993-2002 (1989).

Venkatasubramanian, V., R. Vidyanathan, and Y. Yamamoto, "Process Fault Detection and Diagnosis Using Neural Networks - I. Steady-State Processes", *Computers and Chemical Engineering*, **14**(7), 699-712 (1990).

Vinson, J.M., and L.H. Ungar, "Dynamic Process Monitoring and Fault Diagnosis With Qualitative Models", *IEEE Transactions on Systems, Man and Cybernetics*, **25**(1), 181-189 (1995).

Wold, S., K. Esbensen, and P. Geladi, "Principal Component Analysis", *Chemometrics and Intelligent Laboratory Systems*, **2**, 37-52 (1987).

Watanabe, K., and D.M. Himmelblau, "Incipient Fault Diagnosis of Nonlinear Processes with Multiple Causes of Faults", *Chemical Engineering Science*, **39**(3), 491-508 (1984).

Watanabe, K., I. Matsuura, M. Abe, M. Kubota, and D.M. Himmelblau, "Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks", *AIChE Journal*, **35**(11), 1803-1812 (1989).

Willsky, A., "A Survey of Design Methods for Failure Detection in Dynamic Systems", *Automatica*, **12**, 601-611 (1976).

Willsky, A.S., and H.L. Jones, "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems", *IEEE Transactions on Automatic Control*, **AC-21**, 108-112 (1976).

NOTATION

normal symbols	scalars
bold small symbols	row vectors or column vectors
bold capital symbols	matrices
$\mathbf{A}(\mathrm{i},\mathrm{j})$	the (i,j) element of matrix **A**
$\mathbf{A}(:,\mathrm{j})$	the j^{th} column of matrix **A** as a column vector
$\mathbf{b} = \mathbf{A}(\mathrm{N}:-1:1,\mathrm{j})$	**b** is a column vector that contains the j^{th} column of matrix **A** in reverse order; i.e., the first element of **b** is the last element of $\mathbf{A}(:,\mathrm{j})$, the second element of **b** is the one-before-the-last element of $\mathbf{A}(:,\mathrm{j})$ etc.
$\mathbf{A}^{\mathrm{T}}$	the transpose of matrix **A**
$\mathbf{A}^{-1}, \lvert\mathbf{A}\rvert$	the inverse and the determinant of a square matrix **A**
$\lvert\mathrm{a}\rvert$	absolute value of scalar a
R, RS	any reference pattern for deterministic and stochastic faults, matrices of dimension $\mathrm{r} \times \mathrm{N}$
N	number of features and/or variables for each pattern
r	length of **R** and **RS** patterns
T, TS	any test pattern for deterministic and stochastic faults, matrices of dimension $\mathrm{t} \times \mathrm{N}$ (in Chapter 5)
t	length of **T** and **TS** patterns

$\mathbf{R}_i, \mathbf{RS}_i$ i^{th} reference pattern for deterministic and stochastic faults; matrices of dimension $r_i \times N$

r_i length of $\mathbf{R}_i$ and $\mathbf{RS}_i$ patterns

$\mathbf{T}_i, \mathbf{TS}_i$ i^{th} test pattern for deterministic and stochastic faults; matrices of dimension $t_i \times N$

t_i length of $\mathbf{T}_i$ and $\mathbf{TS}_i$ patterns

$\mathbf{R}_{i,sc}, \mathbf{RS}_{i,sc}$ i^{th} reference pattern for deterministic and stochastic faults after scaling; matrices of dimension $r_i \times N$

$\mathbf{T}_{i,sc}, \mathbf{TS}_{i,sc}$ i^{th} test pattern for deterministic and stochastic faults after scaling; matrices of dimension $t_i \times N$

Corr[RS] the correlation pattern of the **RS** pattern of a stochastic fault; matrix of dimension $(2P+1) \times \frac{N(N+1)}{2}$

P number of lags for correlation estimates

$d(i, j)$ local distance (in any DTW algorithm) between the $\mathbf{T}(i,:)$ and $\mathbf{R}(j,:)$ vectors of the **T** and **R** patterns respectively (**T** is placed on the horizontal axis and **R** is placed on the vertical axis): $d(i,j) = (\mathbf{T}(i,:) - \mathbf{R}(j,:))\,\mathbf{W}(\mathbf{T}(i,:) - \mathbf{R}(j,:))^T$

W weight matrix used in $d(i, j)$

$\hat{\mathbf{F}}$ optimal path; sequence of K points on a $t \times r$ grid, $\hat{\mathbf{F}} = \{\, \mathbf{c}(1), \mathbf{c}(2), \ldots, \mathbf{c}(k), \ldots, \mathbf{c}(K) \,\}$

K number of points of the optimal path

$\mathbf{c}(k)$ the k^{th} point of the optimal path; $\mathbf{c}(k) = (i(k), j(k))$

m maximum number of allowable consequtive horizontal or vertical optimal transitions

M maximum deviation from the linear path emanating from point $(1,1)$

$w(k)$ weighting function for the $d(i(k), j(k))$ local distance

$N(w)$ normalization factor; $N(w) = \sum_{k=1}^{K} w(k)$

$D(t,r)$ normalized total distance between **R** and **T**:

$$D(t,r) = \frac{\sum_{k=1}^{K} d(i(k), j(k)) \cdot w(k)}{N(w)}$$

$\hat{D}(t,r)$ minimum normalized total distance: $\hat{D}(t,r) = \min_{F}\left[D(t,r)\right]$

$\mathbf{D}_A(i,j)$ minimum accumulated total distance between patterns **R** and **T** from point $(1,1)$ to point (i,j)

$\mathbf{u}, \mathbf{l}$ upper and lower limit for the index of the inner iteration in the DTW algorthm; vectors of $t \times 1$.

δ_2 number of points $(1,j)$, $1 \le j \le \delta_2$, and (t,j), $r - \delta_2 + 1 \le j \le r$, among which the first and last points of the optimal path will lie

δ_1 maximum number of consequtive horizontal transitions allowed in the begining and at the end of the **T** pattern

$\mathbf{X} = \mathbf{T}\mathbf{P}^{T} + \mathbf{E}$ PCA decomposition for matrix **X**

T^2, Q T^2 and Q statistics used in PCA models

$\mathbf{B}_i$ the i^{th} trajectory from a reference set of I trajectories of good quality batches; matrix of dimension $b_i \times N$

N number of measured process variables

b_{i} number of observations for the ith batch trajectory

$\mathbf{B}_{\mathrm{i,SC}}$ the ith scaled trajectory

$\widetilde{\mathbf{B}}_{\mathrm{i,SC}}$ the ith scaled trajectory after synchronization

$\mathbf{B}_{\mathrm{REF,SC}}$ scaled reference batch trajectory; matrix of dimension $\mathrm{b}_{\mathrm{REF}} \times \mathrm{N}$

$\mathrm{b}_{\mathrm{REF}}$ number of observations for the reference batch trajectory

$d(\mathrm{i,j})$ local distance (used in DTW algorithm) between the ith vector of $\mathbf{B}_{\mathrm{i,SC}}$ and the jth vector of $\mathbf{B}_{\mathrm{REF,SC}}$; i.e.,

$$d(\mathrm{i,j}) = \left[\mathbf{B}_{\mathrm{i,SC}}(\mathrm{i},:) - \mathbf{B}_{\mathrm{REF,SC}}(\mathrm{j},:)\right]\mathbf{W}\left[\mathbf{B}_{\mathrm{i,SC}}(\mathrm{i},:) - \mathbf{B}_{\mathrm{REF,SC}}(\mathrm{j},:)\right]^{\mathrm{T}}$$

$\overline{\mathbf{B}}_{\mathrm{SC}}$ the average scaled trajectory; i.e., $\overline{\mathbf{B}}_{\mathrm{SC}} = \dfrac{\sum_{\mathrm{i=1}}^{\mathrm{I}} \widetilde{\mathbf{B}}_{\mathrm{i,SC}}}{\mathrm{I}}$

$\mathbf{B}_{\mathrm{NEW}}$ the new batch trajectory; matrix of dimension $\mathrm{b}_{\mathrm{NEW}} \times \mathrm{N}$

$\mathrm{b}_{\mathrm{NEW}}$ number of observations in the new batch trajectory

$\mathbf{B}_{\mathrm{NEW,SC}}$ the new batch trajectory after scaling

$\widetilde{\mathbf{B}}_{\mathrm{NEW,SC}}$ the new batch trajectory after scaling and synchronization

r_{t} the point where the minimum value of the $\mathbf{D}_{\mathrm{A}}(\mathrm{t,j})$ values occurs; i.e., $\mathrm{r}_{\mathrm{t}} = \arg\min_{\mathrm{j}}\left[\mathbf{D}_{\mathrm{A}}(\mathrm{t,j})\right]$, $\mathrm{j} = \mathbf{l}(\mathrm{t}),\ldots,\mathbf{u}(\mathrm{t})$

$d_{\mathrm{FD}}(\mathrm{t,r_t})$ the instantaneous weighted distance (used for fault detection) between an observation vector in the new batch trajectory and an observation vector in the reference batch trajectory, i.e.;

$$d_{FD}(t,r_t) = \left[\mathbf{B}_{NEW,SC}(t,:) - \overline{\mathbf{B}}_{SC}(r_t,:)\right]\mathbf{W}_{FD}\left[\mathbf{B}_{NEW,SC}(t,:) - \overline{\mathbf{B}}_{SC}(r_t,:)\right]^T$$

$\mathbf{W}_{FD}$ the weight matrix of the quadratic diastance used for fault detection

$\mathbf{d}_{FD,95\%}$ the upper 95% confidence interval for the weighted instantaneous distance of a scaled batch trajectory from the scaled average trajectory

$\hat{\mathbf{F}}_{ON-LINE}$ the on-line approximation of the optimal path; i.e.,

$$\hat{\mathbf{F}}_{ON-LINE} = \left\{ (1,r_1),(2,r_2),\ldots,(t,r_t),\ldots,(b_{NEW},r_{b_{NEW}}) \right\}$$

APPENDIX

Table I.1: the 26 variables used in the patterns of Chapter 4

Var. No.	Variable Name	Units	Original Var. No.*
1	A feed	kscmh	XMEAS(1)
2	D feed	kg/h	XMEAS(2)
3	E feed	kg/h	XMEAS(3)
4	A+C feed	kscmh	XMEAS(4)
5	Recycle flow	kscmh	XMEAS(5)
6	Reactor feed rate	kscmh	XMEAS(6)
7	Reactor pressure	kPag	XMEAS(7)
8	Reactor temperature	°C	XMEAS(9)
9	Purge rate	kscmh	XMEAS(10)
10	Product separator underflow	m^3/h	XMEAS(14)
11	Stripper underflow	m^3/h	XMEAS(17)
12	Stripper temperature	°C	XMEAS(18)
13	Stripper steam flow	kg/h	XMEAS(19)
14	Compressor work	kW	XMEAS(20)
15	Separator cooling water outlet temperature	°C	XMEAS(22)
16	B in purge gas	%mole	XMEAS(30)
17	G in product	%mole	XMEAS(40)
18	H in product	%mole	XMEAS(41)
19	D feed flow	% opening	XMV(1)
20	E feed flow	% opening	XMV(2)
21	A feed flow	% opening	XMV(3)
22	A and C feed flow	% opening	XMV(4)
23	Compressor recycle valve	% opening	XMV(5)
24	Stripper liquid product flow	% opening	XMV(8)
25	Reactor cooling water flow	% opening	XMV(10)
26	Condenser cooling water flow	% opening	XMV(11)

Table I.2: the 8 variables used in the patterns of Chapter 5

Var. No.	Variable Name	Units	Original Var. No.*
1	A feed	kscmh	XMEAS(1)
2	D feed	kg/h	XMEAS(2)
3	A+C feed	kscmh	XMEAS(4)
4	Reactor temperature	°C	XMEAS(9)
5	Purge rate	kscmh	XMEAS(10)
6	Stripper temperature	°C	XMEAS(18)
7	G in product	%mole	XMEAS(40)
8	H in product	%mole	XMEAS(41)

* This is the notation for the variables used in the original Tennessee-Eastman paper (Downs and Vogel, 1993). XMEAS is the vector that stores the process variables and XMV is the vector that stores valve positions for the manipulated variables.

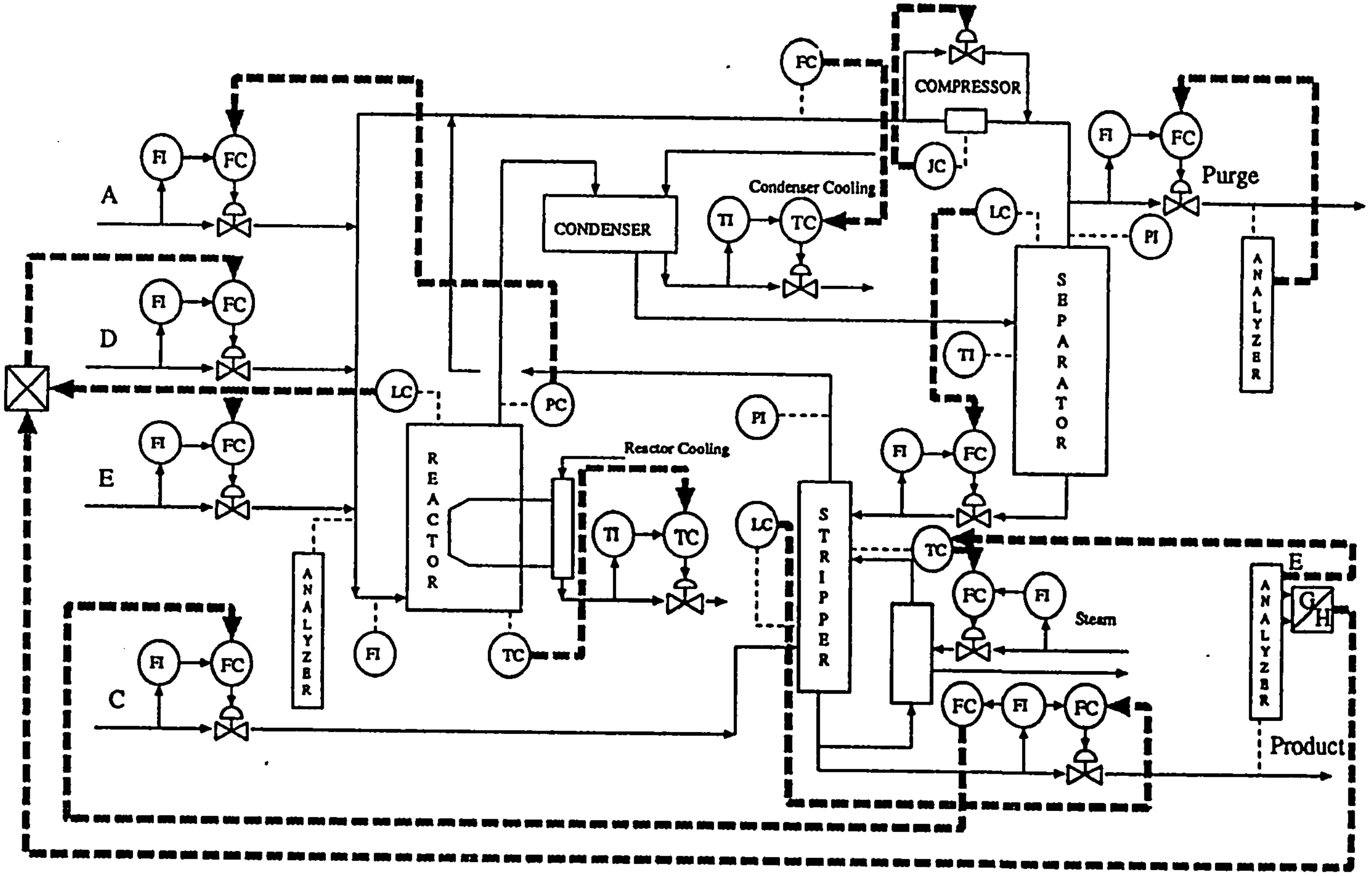


Figure I.1: Tennessee-Eastman plant schematic with the control scheme of McAvoy and Ye; taken from McAvoy and Ye, 1994.

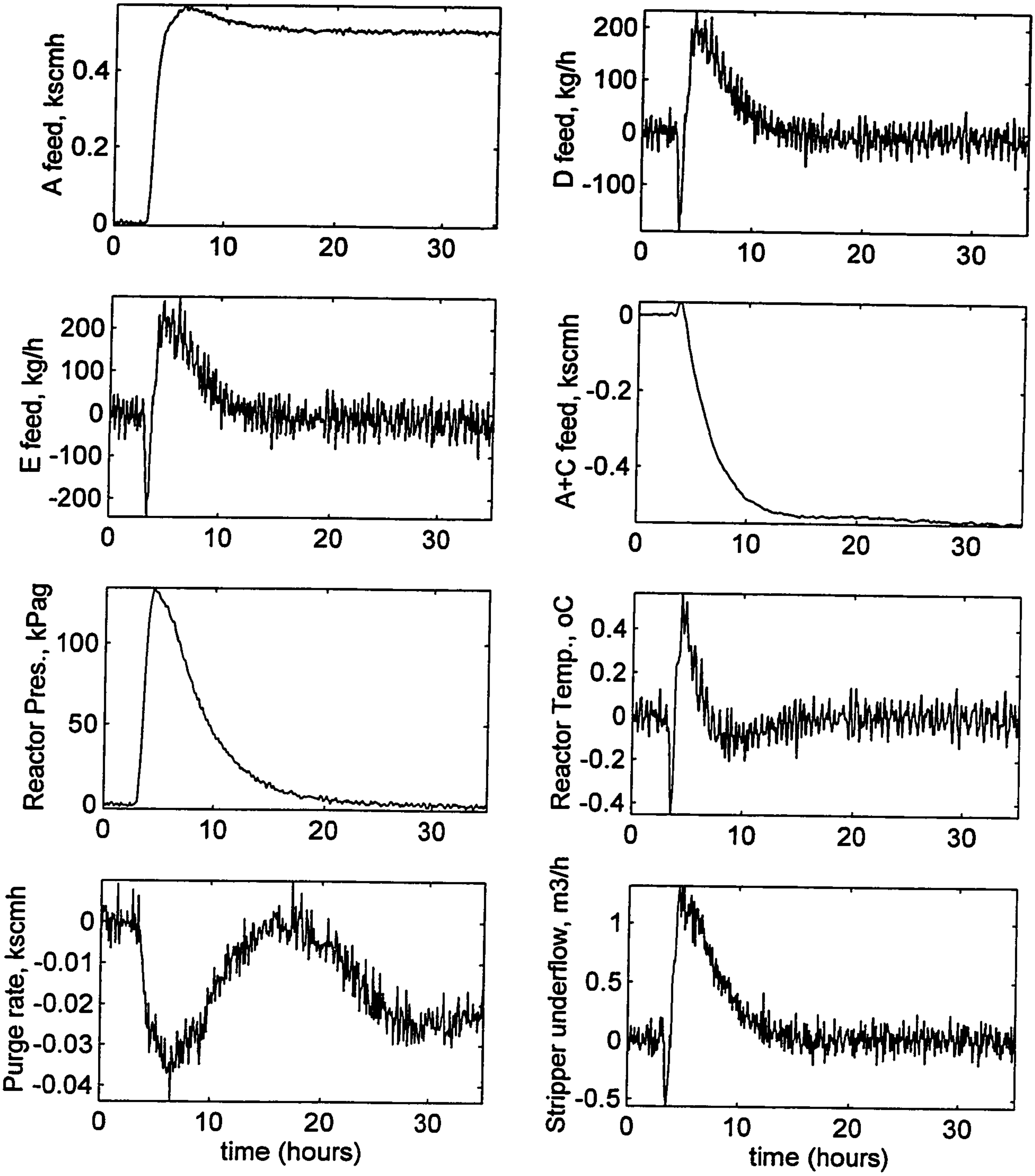


Figure I.2a [Part I]: Behavior of 8 (out of 26) variables during the reference pattern $\mathbf{R}_1$; the initial value for each variable has been subtracted.

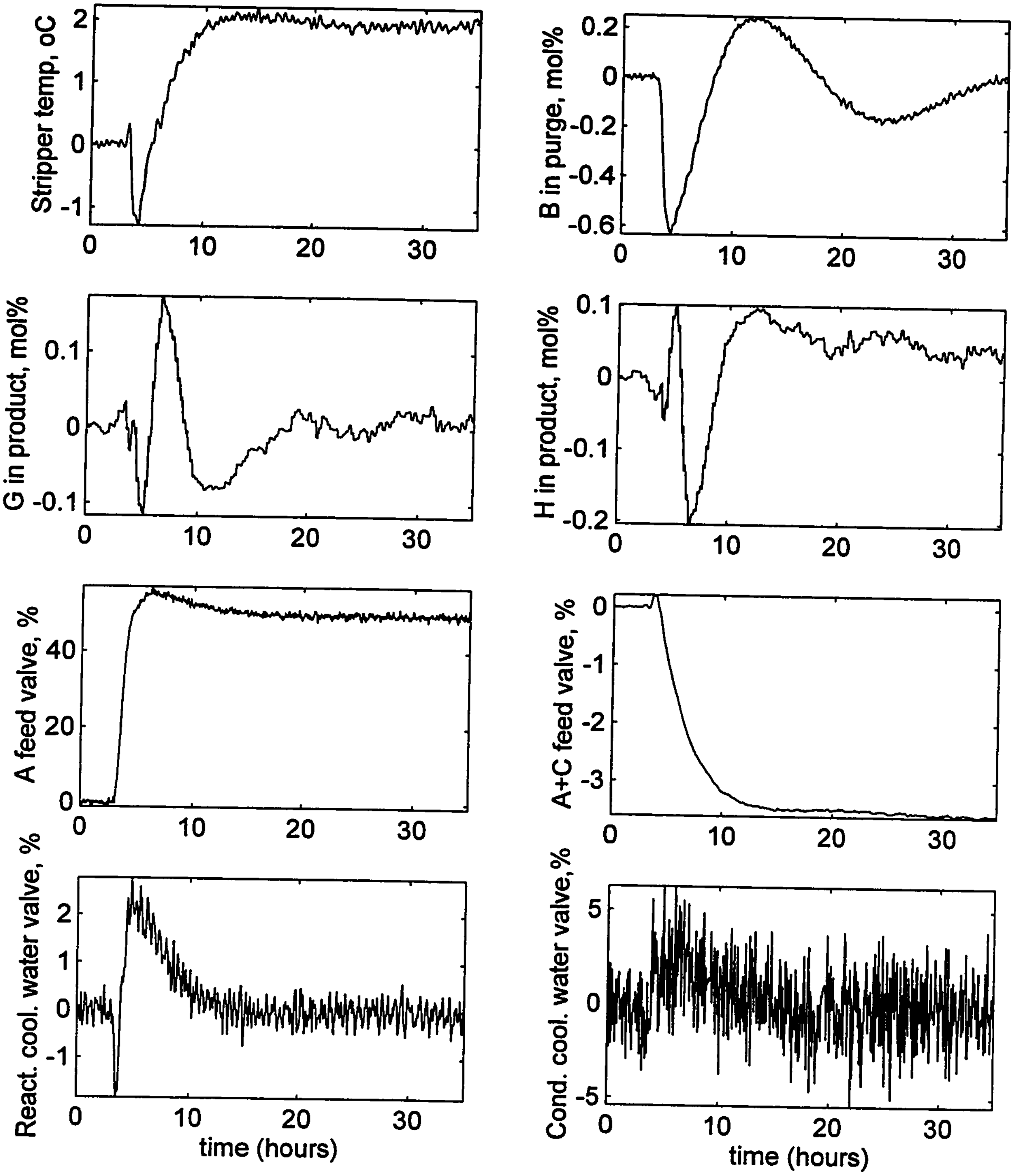


Figure I.2a [Part II]: Behavior of 8 (out of 26) variables during the reference pattern $\mathbf{R}_1$; the initial value for each variable has been subtracted.

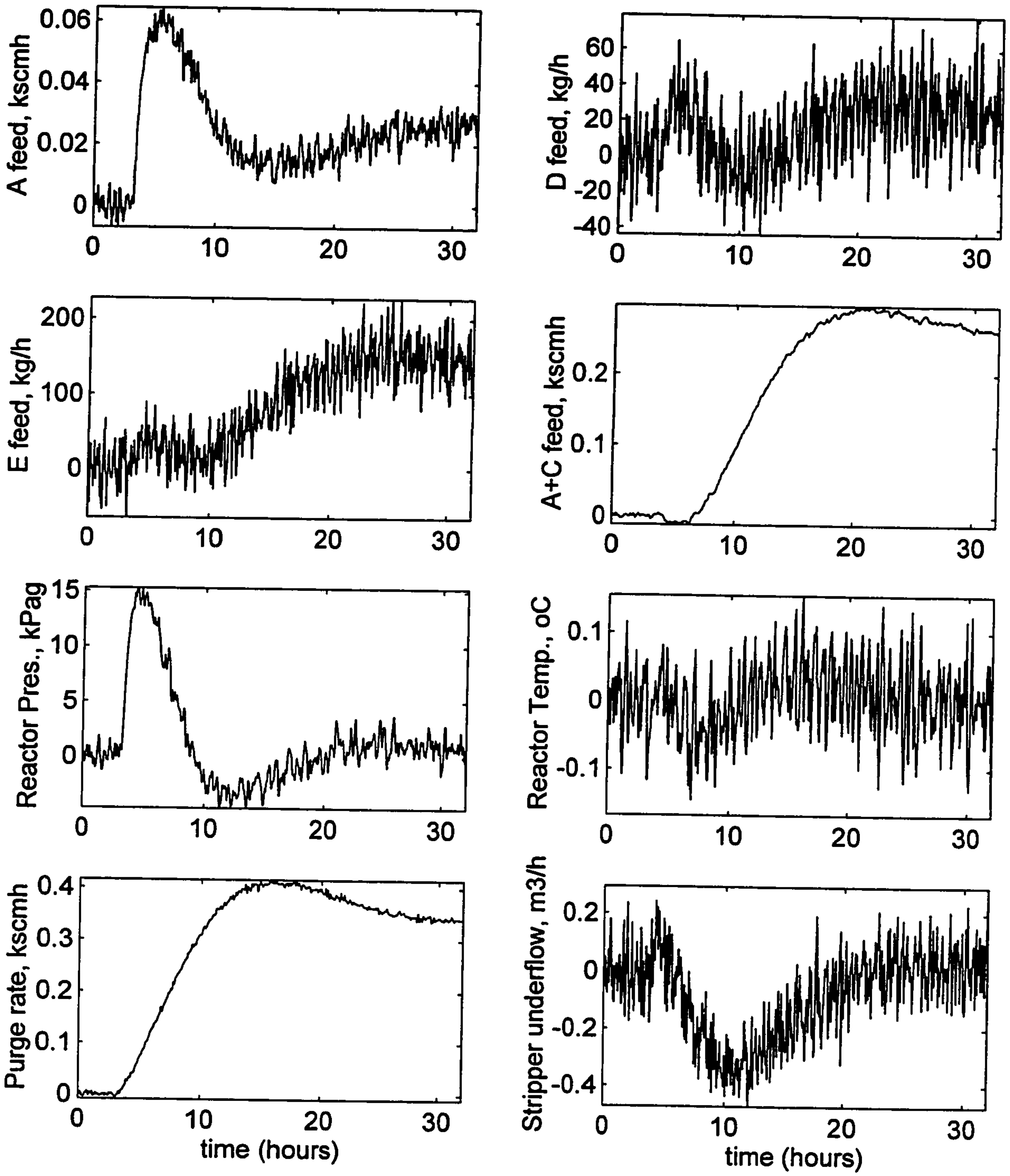


Figure I.2b [Part I]: Behavior of 8 (out of 26) variables during the reference pattern $\mathbf{R}_2$; the initial value for each variable has been subtracted.

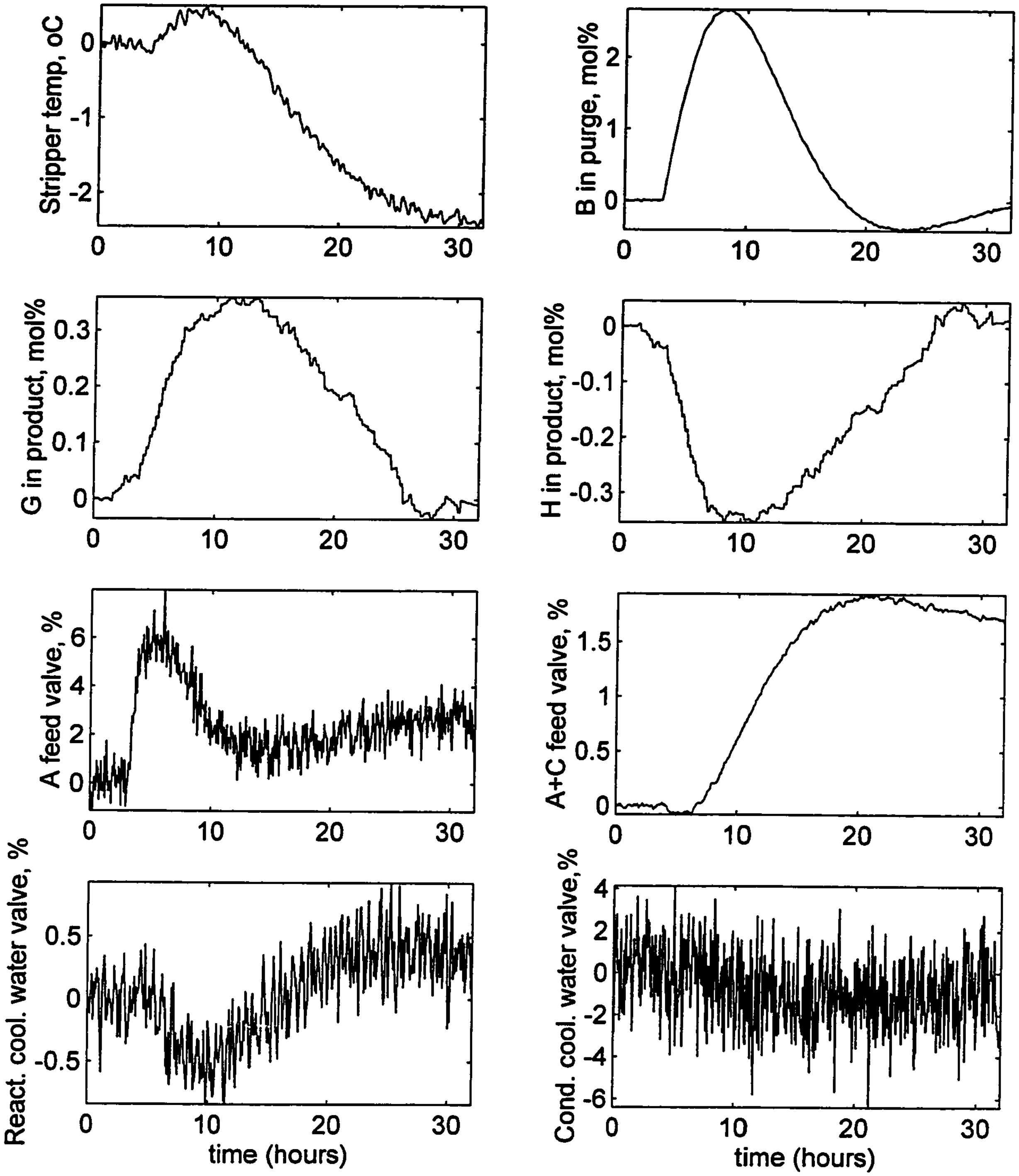


Figure I.2b [Part II]: Behavior of 8 (out of 26) variables during the reference pattern $\mathbf{R}_2$; the initial value for each variable has been subtracted.

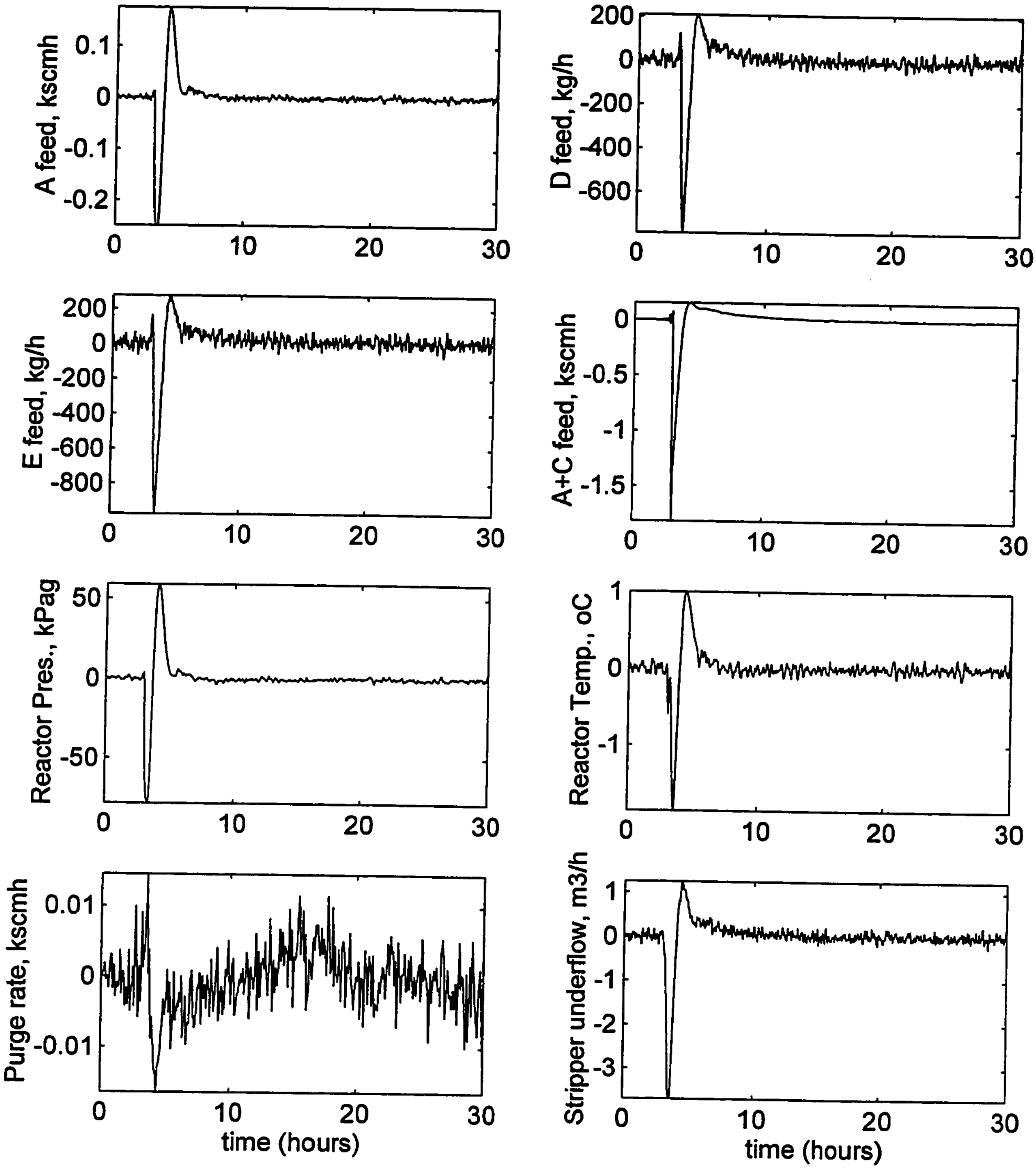


Figure I.2c [Part I]: Behavior of 8 (out of 26) variables during the reference pattern $\mathbf{R}_3$; the initial value for each variable has been subtracted.

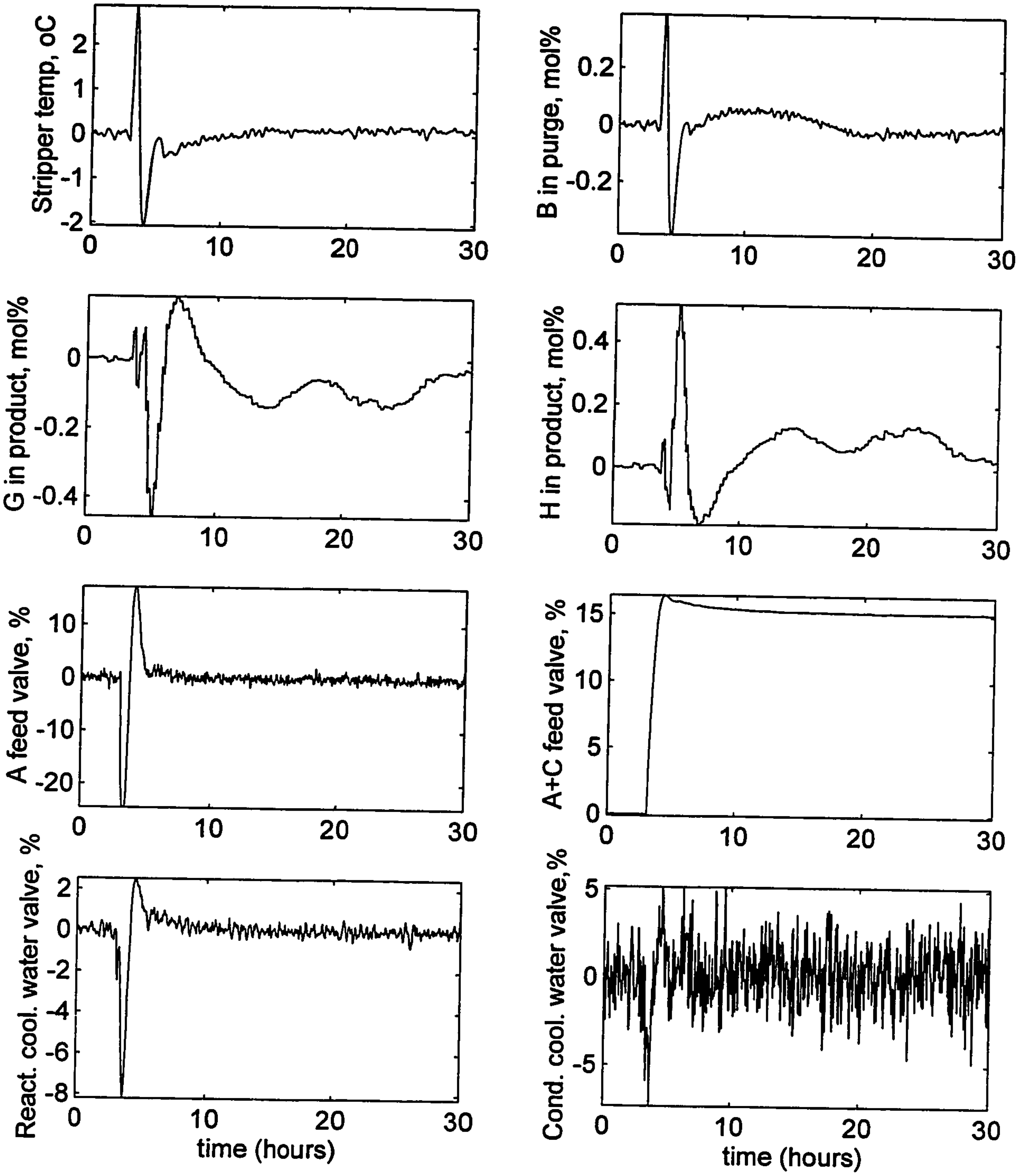


Figure I.2c [Part II]: Behavior of 8 (out of 26) variables during the reference pattern $\mathbf{R}_3$; the initial value for each variable has been subtracted.

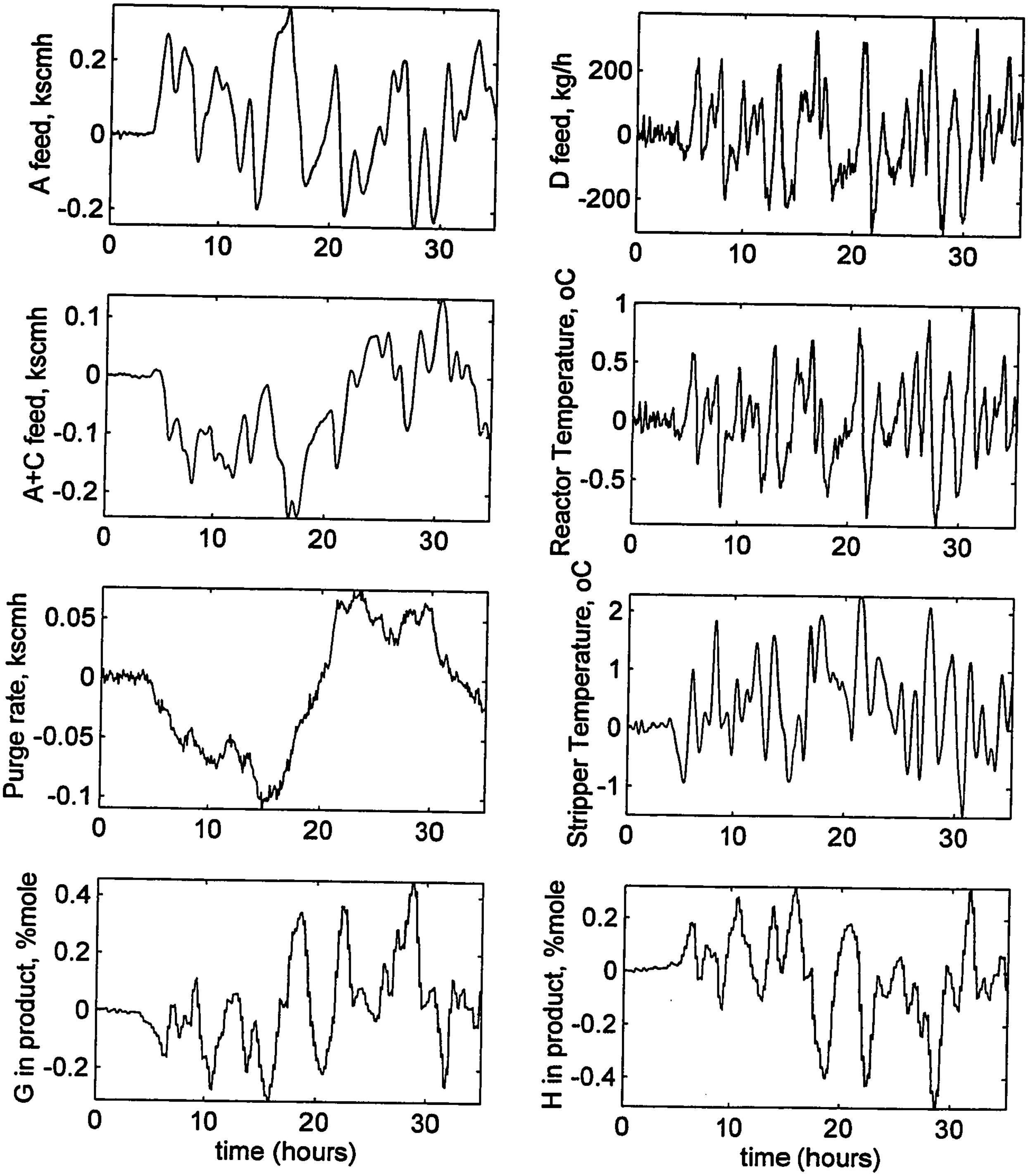


Figure I.3a: Behavior of the 8 variables during the reference pattern $\mathbf{RS}_1$; the initial value for each variable has been subtracted.

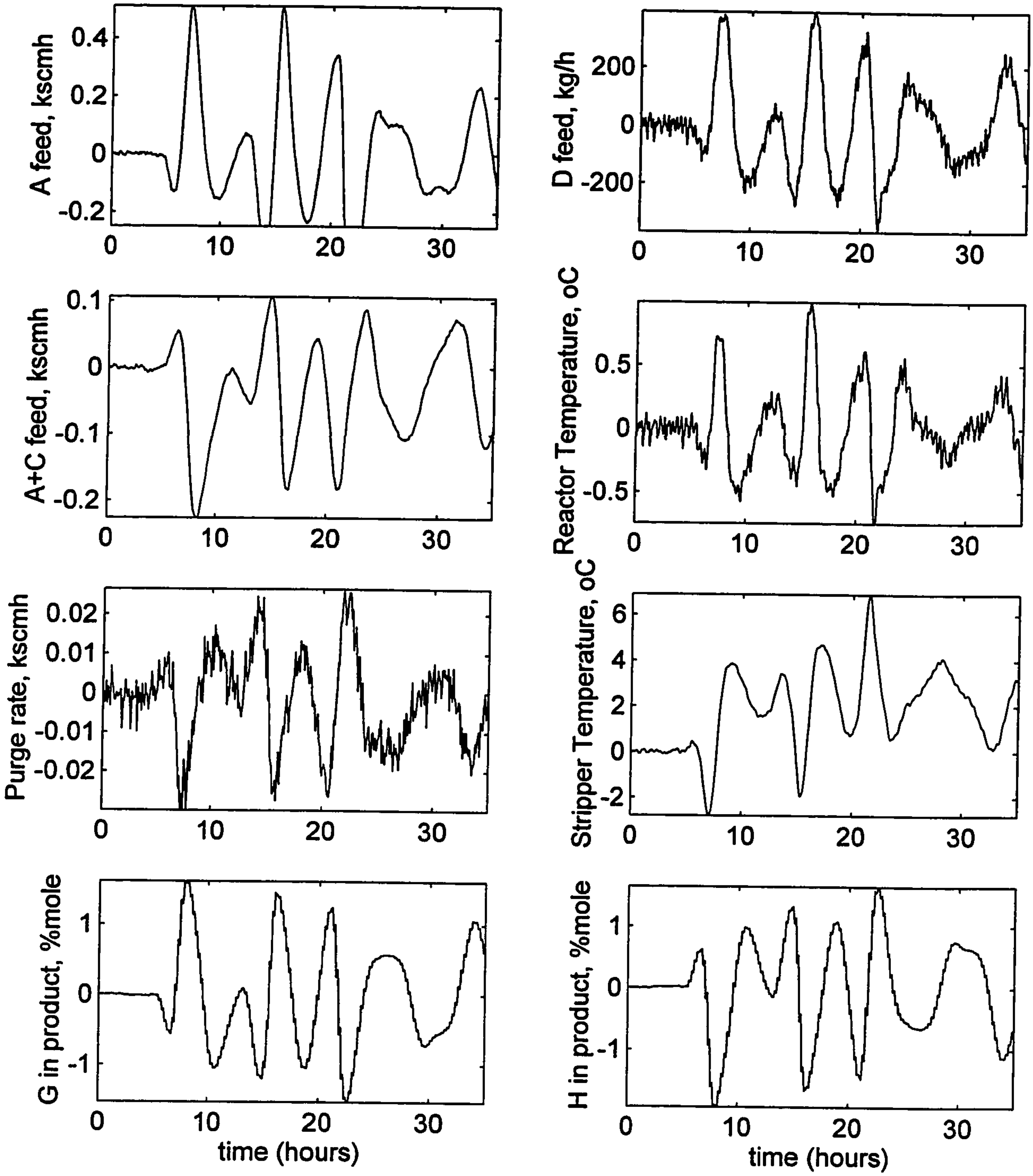


Figure I.3b: Behavior of the 8 variables during the reference pattern $\mathbf{RS_2}$; the initial value for each variable has been subtracted.

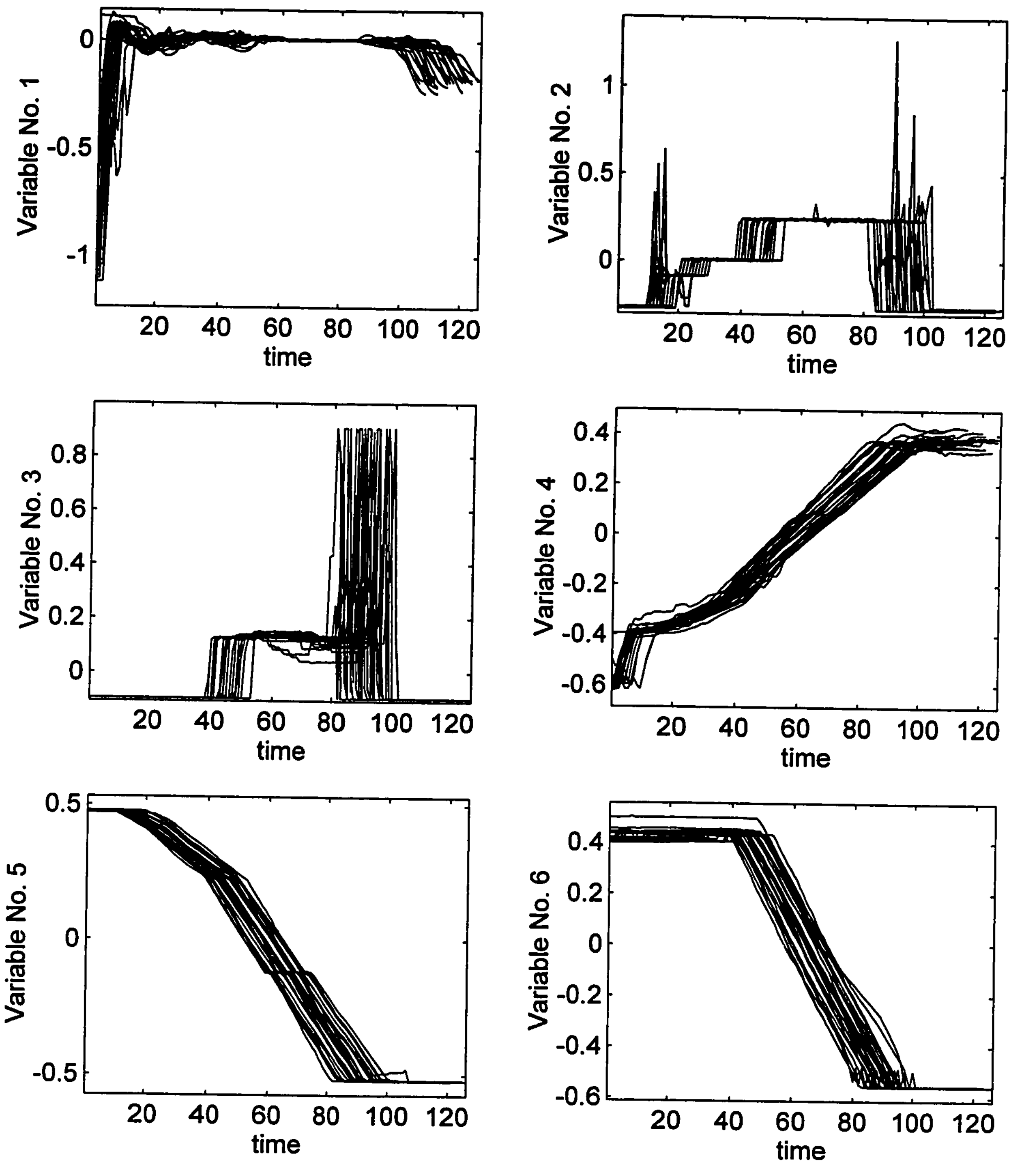


Figure I.4a: Behavior of the Variables No. 1 to 6 during the 31 good quality batches before synchronization; the variables have been divided with their average range.

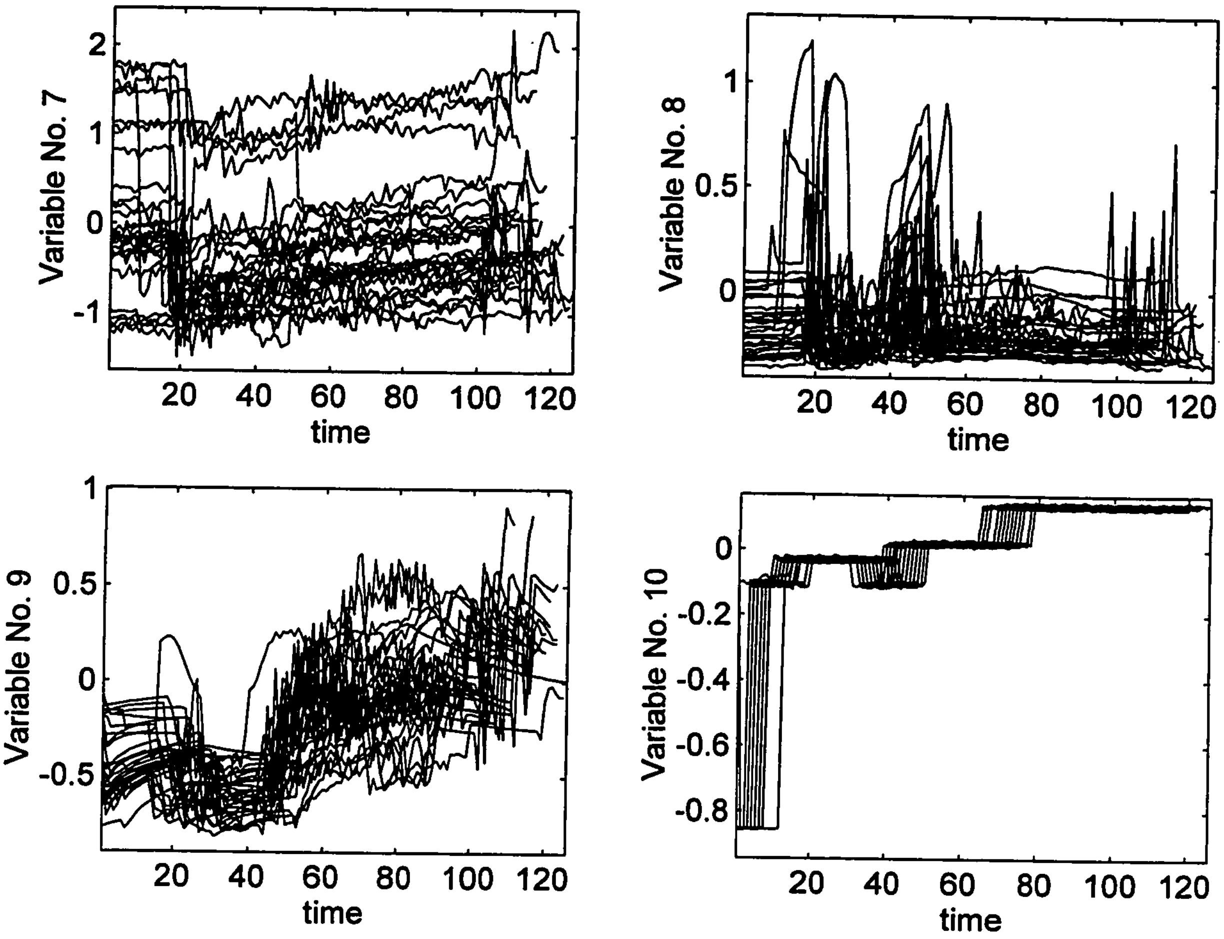
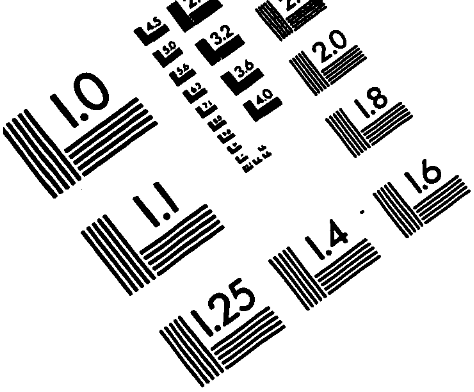
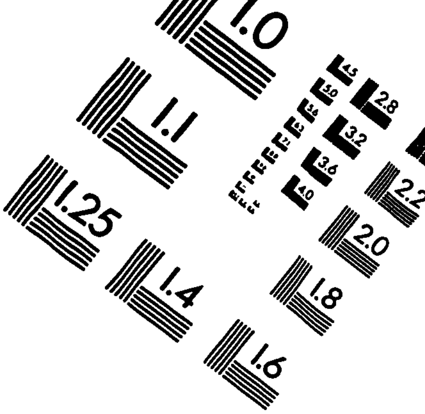
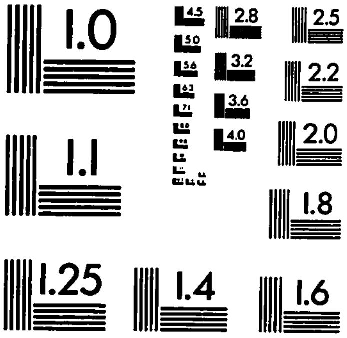
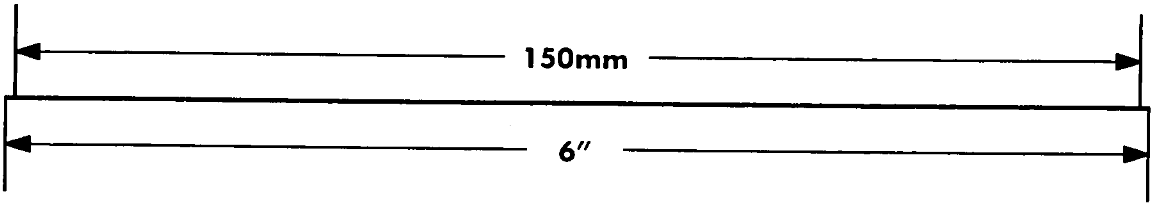
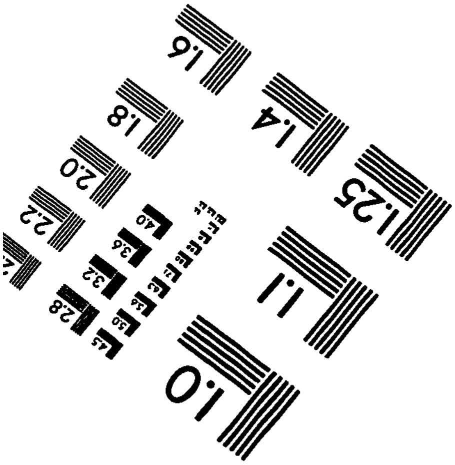
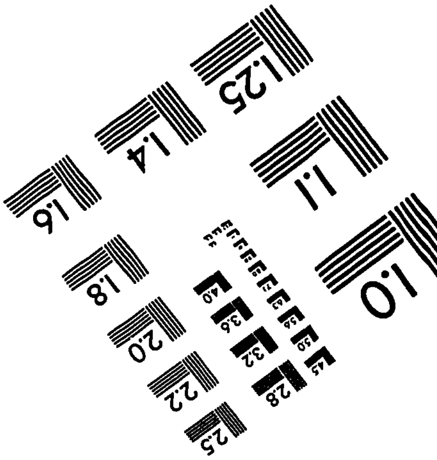


Figure I.4b: Behavior of the Variables No. 7 to 10 during the 31 good quality batches before synchronization; the variables have been divided with their average range.



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989