Universitat de Girona

PATTERN RECOGNITION BASED ON QUALITATIVE REPRESENTATION OF SIGNALS. APPLICATION TO SITUATION ASSESSMENT OF DYNAMIC SYTEMS

Fco. Ignacio GAMERO ARGÜELLO

Dipòsit legal: GI. 1247-2012
http://hdl.handle.net/10803/83670

Universitat de Girona

PhD Thesis

Pattern Recognition Based on Qualitative Representation of Signals. Application to Situation Assessment of Dynamic Systems

Fco. Ignacio Gamero Argüello

2012

Thesis Advisor:
Dr. Joan Colomer Llinàs

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy at the University of Girona. Doctoral Programme in Technology.

A Eva y a esos locos bajitos,
Eloy y Sofía

In the algebra, there is so complete a subjection to certain rules and formulas, that there results an art full of confusion and obscurity calculated to embarrass, instead of a science fitted to cultivate the mind.

Discourse on the method of rightly conducting the reason, and seeking truth in the sciences.

René Descartes (1596-1650)

Wit consists in knowing the resemblance of things which differ, and the difference of things which are alike.

- Germany (pt. III, ch. VIII)

Anne Louise Germaine De Stael-Holstein
French writer (1766 - 1817)

Acknowledgements

(In order of appearance)

I would like to use these lines to express my most sincere thanks to Josep Lluís de la Rosa, who encouraged me to begin my doctoral studies and gave me the opportunity to stay at the UdG as an associate professor.

I would especially like to thank Joan Colomer for his help. He is, probably, the most patient supervisor in the world.

I also thank Joaquim Meléndez for his support, ideas and opportunities given.

I owe a lot more to all three. Their door was always open to listen to me, they have trusted me and helped me at different stages. Thank you for holding my hand.

I would like to thank the laboratory of Automatic Control and Computers for Industry at the University of Lille, for the use of the plant pilot.

Thanks also to those researchers who have always responded to my e-mails. Among them, Eamon Keogh and Sylvie Charbonnier, who provided me with the qualitative representation from her algorithm.

No less important are the people closest to me. Regards and thank you very much for your friendship.

Finally I want to thank Eva for her unconditional support and immense patience. Thanks Eva for your love and affection that gave me that final prodding.

I do not want to forget anyone; if I have, it is a terrible oversight.

Agradecimientos

(En orden de aparición)

Quisiera utilizar estas líneas para expresar mi más sincero agradecimiento a Josep Lluís de la Rosa, quien me animó a empezar esta tesis y me dio la oportunidad de entrar en la UdG como profesor asociado.

Quisiera agradecer especialmente a Joan Colomer su ayuda. Él es, posiblemente, el director de tesis más paciente del mundo.

También quiero agradecer a Joaquim Meléndez su apoyo, sus ideas y las oportunidades.

A los tres, sin embargo, les debo mucho más. Han tenido siempre su puerta abierta para escucharme, han confiado en mí y me han ayudado en diferentes aspectos. Gracias por esa mano tendida.

Un recuerdo de agradecimiento también a aquellos investigadores que siempre han respondido a mis e-mails. Entre ellos Eamon Keogh y Sylvie Charbonnier, quien me facilitó las representaciones cualitativas obtenidas mediante su algoritmo.

Quisiera también agradecer al laboratorio *Automatic Control and Computers for Industry* de la Universidad de Lille por el uso de la planta piloto.

No menos importantes son las personas más cercanas a mí. Un saludo y muchas gracias por vuestra amistad.

Por último, quiero agradecer a Eva su apoyo incondicional y su inmensa paciencia. Gracias Eva por tu amor y el afecto que me dio ese empujón final.

No me gustaría olvidarme de nadie, si lo hice fue un terrible descuido.

Abstract

Supervision has grown as an active research topic over the last twenty years. It involves process monitoring followed by situation assessment and action proposals in order to assure process operating under specifications. The main focus is centred on situation assessment to decide on the adequacy of process behaviour with respect to specifications. It consists of evaluating the state of the process and its evolution. Fault detection, isolation and identification are the typical steps involved in this procedure.

Multiple approaches have been proposed for each of these objectives, but it is not always possible, or it is unfeasible, to have a mathematical (functional or structural) model to represent the system operation. Therefore, other types of approaches must be considered in order to identify the different situations that the system may experience at a given time, these situations can be either normal operations or failures.

Classification methods are typically proposed as strategies for diagnosis. Here, identification of the functional states is reduced to recognising the current shapes of variables as well-known states, commonly taking advantage of a process expert or past experiences. On the other hand, there are some drawbacks such as the great deal of process data available, the lack of interpretability of raw data and the need to associate them with a reduced number of predefined system states (normal operation, fault *1*, … fault *n*). This is because human knowledge is related to concepts and symbols whereas process acquisition systems provide monitoring systems with numerical data. Consequently, these type of knowledge-based decision systems are usually forced to work in a higher level of abstraction using symbolic variables instead of raw data coming from sensors. To solve this, qualitative representations are proposed to represent signal trends.

This doctoral dissertation deals with the study of classification methods when performing qualitative trends analysis. Thus, a tool specifically designed to obtain qualitative trends based on episodes from a measured variable is presented and some techniques are proposed to deal with these problems. The main contribution of this thesis is a new algorithm that returns a normalised index related to the degree of similarity between sequences of episodes. The aim is to obtain qualitative trends and their classification by means of the extracted knowledge from past experiences. Performance of the proposed techniques is validated using different scenarios: a laboratory plant, an industrial plant and an electrical system problem known as voltage sag.

Contents

List of figures

List of tables

Nomenclature

Acronyms

AI	Artificial Intelligence
APCA	Adaptive Piecewise Constant Approximation
ASM	Abnormal Situation Management
CBR	Case Based Reasoning
DFT	Discrete Fourier Transform
DP	Dynamic Programming
DTW	Dynamic Time Warping
EPW	Extreme Points Warping
EpDTW	DTW for episodes
ES	Expert System
FDD	Fault detection and diagnosis
IDDTW	Iterative Deepening Dynamic Time Warping algorithm
IFAC	International Federation of Automatic Control
LCS or LCSS	Longest Common Subsequence
MPCA	Multiway Principal Component Analysis
MPLS	Multiway Projection to Latent Structures
MSPC	Multivariate Statistical Process Control
NLPCA	Nonlinear Principal Component Analysis
PAA	Piecewise Aggregate Approximation
PCA	Principal Component Analysis
PDTW	Piecewise Dynamic Time Warping
PLA	Piecewise Linear Approximation
QPT	Qualitative Process Theory
QSI	Qualitative Similarity Index
QSSI	Qualitative Sequence Similarity Index
QTA	Qualitative Trend Analysis
SAX	Symbolic Aggregate approximation
SDL	Shape Definition Language
SDTW	Segmented Dynamic Time Warping
SPC	Statistical Process Control
SPE	Squared Prediction Error
STW	Segment-wise Time Warping

Notation

$\boldsymbol{A} = \langle a_1, a_2 ... a_m \rangle$	Sequence of symbols or strings of length *m*
dist	Distance function
distS	Distance between two signatures
$\mathbb{E}=\langle \mathbb{E}_1, \mathbb{E}_2,.., \mathbb{E}_M \rangle$	Sequence of episodes
$\mathbb{E}_N$	An episode *N* within a sequence of episodes
F	Characteristic function

i,j	Indices for any time series
(i,j)	Position of a cell in a matrix: i=row, j=column
M	Alignment matrix
Md	Direction matrix
sim()	Similarity function
SPE_i	SPE calculated for the measurement i
$\boldsymbol{x}$	Any finite time series
$\boldsymbol{x} = [x_1,x_2,.., x_i,..,x_m]$	Time series of length m with equi-spaced sampling interval.
$\boldsymbol{x} =[(t_1,x_1),\ldots,(t_m,x_m)]$	Time series as a sequence of pairs timestamp and value
$\mathbf{W} =[w_1, w_2,.., w_k,.., w_m]$	Pathway of length m
$w_k =(i_k,j_k)$	The k^{th} pair of time indexes i_k and j_k in the pathway **W**
$\mathcal{Z}$	Alphabet of symbols
$\gamma()$	Cumulative score function
$\gamma(i,j)$	Cumulative value in the cell indexed by i,j
$\delta(a_i, b_j)$	Score given for matching a_i, with b_j

Chapter 1.

Introduction

1.1. Motivation

The growing necessity for improvements in product quality, process productivity, production time reduction and environmental and safety requirements have motivated the need for automation in the monitoring processes. High process efficiency is expected and therefore unnecessary stops should be eliminated, the occurrence of abnormal situations predicted and acted upon quickly and effectively if those situations are unavoidable.

On the other hand, the considerable increase of process measures and their complexity, makes it necessary to assist the plant operator in monitoring and the decision-making in order to manage abnormal situations. The ASM® Consortium (ASM, n.d.) defines an abnormal situation as:

- A disturbance, or series of disturbances, in a process that causes plant operations to deviate from the normal operating state.
- The nature of the abnormal situation may be of minimal or catastrophic consequence. It is the job of the operations team to identify the cause of the situation and execute compensatory or corrective actions in a timely and efficient manner.
- A disturbance may cause a reduction in production; in more serious cases it may endanger human life.
- Abnormal situations extend, develop, and change over time in the dynamic process control environments increasing the complexity of the intervention requirements.

Fault diagnosis is the first step in Abnormal Situation Management (ASM). The aim of ASM is timely detection, diagnosis and correction of abnormal conditions. The main objective is to avoid plant shutdowns. Furthermore, early diagnosis can reduce the loss of productivity during an abnormal event if it is performed when the plant is still operating within a controllable region (Fig. 1.1, Mylaraswamy, 1996). Specifically, diagnosis should assess the current process state using online process measurements while the process is operational. Therefore, identifying that the process is normal is as important as identifying that there is a problem with it. Diagnosis deals with process measurements and process knowledge. The latter encompasses a variety of knowledge which includes known facts and process behaviour, an estimate of the current state of the process and, maybe, future behaviour. To sum up, diagnosis can be defined as a transformation of the process measurements using process knowledge to identify different process states.

Reconfiguration follows diagnosis of faults and assessment of situations. It consists of proposing changes in the plant using the diagnosis result in order to keep the process under specifications when a faulty situation (non-catastrophic) is detected and diagnosed. Simple actions could be set points adjustment or tuning parameters in controllers. However, other actions related to maintenance or planning (control or system) could be proposed according to a predefined strategy (i.e. repair manuals, maintenance schedulers, planners and agendas).

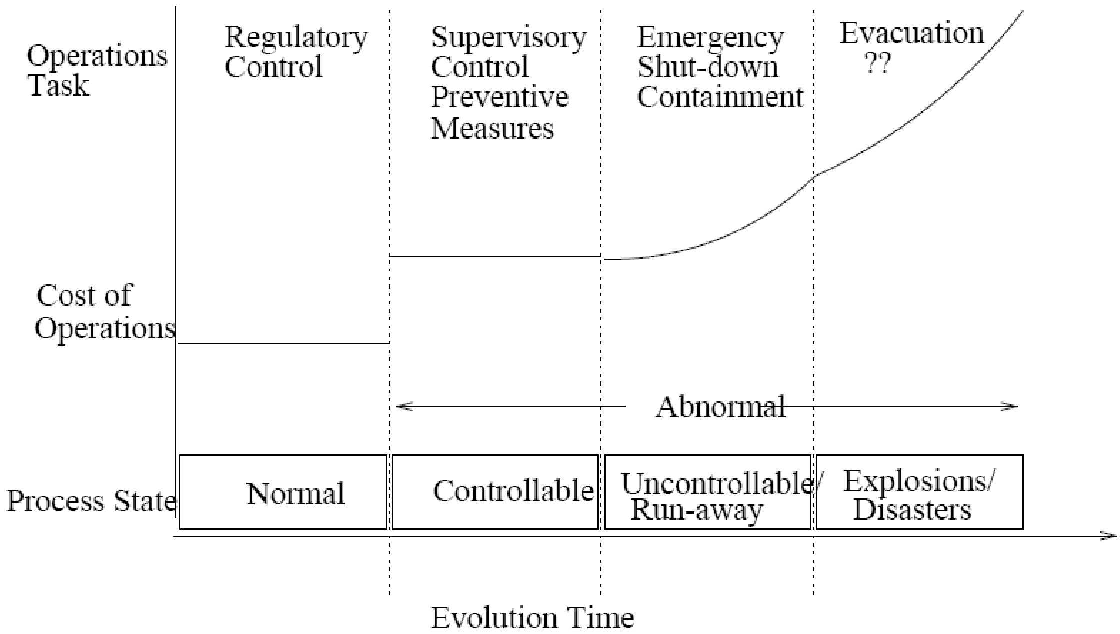


Fig. 1.1 Impact of faults in process operations (Mylaraswamy, 1996)

The importance of fault diagnosis when the plant is in a controllable state is completely out of doubt, not only from a safety point of view, but also in terms of economic costs. A study obtained by the ASM Consortium estimate the cost of lost production due to abnormal situations is at least $10 billion annually in the U.S. petrochemical industry due to unexpected events. For this, there is an immediate need to design better and more efficient intelligent real-time operator support system. Such a system can perform automated control actions as well as to give support to operators for decision making. Nevertheless, the implementation of such systems in real plants presents a variety of difficulties: complexity of modern plants, lack of useful process models, different sources of knowledge, efforts to maintain the system and operator's adaptation.

1.2. Objectives

Operators have an important role in the adequate functioning of processes, but the complexity, sophisticated control strategies, highly integrated plants and millions of pieces of collected data is an obstacle too large to overcome without support. A supervision project involves process monitoring followed by situation assessment or fault diagnosis and action proposals to assure process operating conditions. The main focus is centred on situation assessment to decide on the adequacy of process behaviour with respect to specifications.

Thus, when the monitoring task detects a deviation in process behaviour, the fault or situation must be identified. For this, process knowledge has to be obtained, which can be taken from analytical models of the process variables, but it is difficult to obtain accurate models and they are sometimes too complex to deal with. Another way of obtaining knowledge is by using previously processed data and applying Artificial Intelligence (AI) knowledge-based techniques in order to assess situations.

Knowledge-based detection can respond to a description of situations from the signals or their abstractions. Therefore, in many cases it is reasonable to create a 'mental model' of plant operations and use methods to handle this knowledge, such as a human expert operator. Plant operators typically monitor the process status for signs of normality or abnormality in the measured process signals. Their knowledge is based on the ability to recognise patterns that may indicate abnormal operation. In this case, the problem in the situation assessment process lies in the computerised recognition of patterns. Then, a correct representation and classification of these patterns allows the identification of certain kinds of situations. When some of these process statuses are observable, they are called symptoms. Symptoms are, therefore, observable behaviour in the process when an undesirable event occurs in the system and they can be classified through an experience-based diagnosis. A good diagnostic method should be able to detect these symptoms and compare them to the normal state and flag them as abnormal. However, different patterns belonging to the same kind of situation can have a different duration or magnitude. This fact, obvious to a human observer, can greatly complicate the task for a computer. For this reason, it is necessary to have tools to handle the process signals.

Knowledge-based systems are usually forced to work at a higher level of abstraction using symbolic variables instead of raw data from sensors to deal with human knowledge and reasoning. Qualitative trend representations based on episodes are a tool for abstracting a quantitative monitored signal in a qualitative form using symbols (or episodes). Therefore, episodes applied to situation assessment are, without a doubt, an important support for situation assessment because the study of trends allows symptoms (or patterns) in the process to be classified.

The objective of this thesis is two-fold. First of all, although of secondary importance, it aims to improve the performance of existing qualitative episode-based trend analysis, to make it more widely applicable depending on the different needs of the process plants. This method will be applied to process variables in order to obtain qualitative patterns that can be compared to others. It is thus possible to assess different process statuses.

The second, and main, aim of this thesis is to obtain a general similarity measure for qualitative trends signals. Since episodes are an asynchronous and qualitative representation, it is essential to have an algorithm that, when dealing with this type of representation, can be applied to situation assessment in dynamic systems using pattern matching. Thus the main subject of this thesis is the introduction of a new method that quantifies similarity between qualitative patterns of signals by means of different alignment algorithms.

1.3. Outline

The chapters of this thesis are divided into those dedicated to the state of the art and basic concepts behind the thesis, and those dedicated to presenting the contributions. The chapter organisation is intended to provide a linear reading of the work developed. The following paragraphs provide a more detailed overview of the thesis:

Chapter 2 introduces the terminology used in the field of fault detection and diagnosis. An overview of various diagnostic methods from different perspectives is also provided. The general formulation used in subsequent chapters and a discussion about problems in fault detection and diagnosis are also presented. More emphasis has been placed on knowledge-based representations and the task of diagnosis as a classification problem, which is the challenging problem that has motivated this thesis.

Chapter 3 provides the reader with some definitions of qualitative trends based on episodes and their use for process monitoring and fault detection and diagnosis. Several qualitative representations are presented and the main formalism used in this work is described. Other studies about symbolic representations are also outlined.

Chapter 4 is dedicated to presenting the architecture and development of a software tool for qualitative trend extraction. This software, called Qualtras, facilitates the abstraction of the most significant characteristics of the signals by representing any process signal by means of episodes.

In chapter 5, terminology and problems related to the retrieval of similar time sequences are enumerated. Next, an overview of the most common methods for sequence alignment and similarity is given and the comparison of patterns as a classification method for diagnosis is introduced.

Chapter 6 proposes the use of the Dynamic Time Warping (DTW) algorithm to reduce the effects of time misalignments. However, due to the high computational cost, two different solutions are developed. First, the application of representations based on episodes is proposed. Secondly, a modification of DTW in order to be applied online is explained. Both approaches are illustrated with practical examples.

Chapter 7 covers the design of a new algorithm (QSSI) to deal with the comparison of qualitative sequences based on episodes. The algorithm returns a normalised index related to the degree of similarity between qualitative trends signals. This is the most innovative aspect of the work since QSSI addresses the main drawbacks of DTW when it performs situation assessment by comparing qualitative trends. Two application examples based on the QSSI similarity principle are presented.

Chapter 8 offers an illustrative comparison between the three methods developed in this thesis and different similarity measures which deal with numerical series. The results obtained suggest studying the use of qualitative methods as an alternative to numerical similarity techniques.

Finally some conclusions from this work are drawn up in Chapter 9, where a perspective of future research opportunities will also be given.

The thesis finishes with three annexes and the bibliography.

Chapter 2.

Fault Detection and Diagnosis

2.1. Introduction

Nowadays, the interest for supervision is increasing due to the growing demands for quality, safety, reliability, availability and cost efficiency in industrial processes. As systems grows in size and complexity, the possibility of misbehaviour increases. Thus, the call for fault tolerant systems is gaining more and more importance. Fault tolerance could be achieved either by passive or active techniques (Frank and Koppen-Seliger, 1997):

- The *passive approach* makes use of robust control techniques to ensure that the closed-loop system becomes insensitive with respect to faults. This solution allows small faults be tolerated without control system reconfiguration.
- The *active approach* provides fault accommodation, i.e., the reconfiguration of the control system when a fault has occurred. Reconfiguration can be thought at various degrees, i.e. set point changes, parameters re-tuning or structural changes. The aim of this approach is to avoid a fast degradation of the whole system due to this fault. The majority of actual solutions involve human decision.

This chapter introduces the terminology used in the field of supervision and fault detection and diagnosis (FDD). It also overviews different diagnostic methodologies from different perspectives and remarks on the importance of human knowledge of process behaviour and how it can be used to implement supervisory structures.

2.2. Terminology

The terminology in the field of supervision, fault detection and diagnosis is not consistent in the literature. Consequently, the SAFEPROCESS Technical Committee (International Federation of Automatic Control) tried to find commonly accepted definitions. Some of these preliminary proposals are collected in Isermann and Ballé, 1997, which is generally in accordance with the terminology used in this text.

About states and signals:

- Fault: Unpermitted deviation of at least one characteristic property or variable of a system from its acceptable/usual/standard condition.
- Failure: A permanent interruption of a system's ability to perform a required function under specified operating conditions.
- Malfunction: Irregularity in fulfilment of a systems desired function.
- *Error*: Deviation between a measured or computed value of an output variable and the specified or theoretically correct value.
- *Disturbance*: An unknown (and uncontrolled) input acting on a system.
- *Perturbation*: An input acting on a system which results in a temporary departure from steady state.
- *Residual:* Fault information carrying signals, based on deviation between measurements and model based computation.
- *Symptom*: A change of an observable quantity from normal behaviour.

About functions:

- *Fault detection*: Determination of faults present in a system and time of detection.
- *Fault isolation:* Determination of kind, location and time of detection of a fault. Follows fault detection.
- *Fault identification:* Determination of the size and time-variant behaviour of a fault. Follows fault isolation.
- *Fault diagnosis*: Determination of the kind, size, location and time of detection of a fault. Follows fault detection. Includes fault isolation and identification.

- *Monitoring:* A continuous real-time task of determining the condition of a physical system, by recording information, recognising and indicating anomalies in the behaviour.
- *Supervision*: Monitoring of a physical system and taking appropriate action to maintain the operation in the case of faults.
- *Protection:* Means by which a potentially dangerous behaviour of the system is suppressed if possible or, means by which the consequences of a dangerous behaviour are avoided.

About models:

- *Quantitative model*: Use of static and dynamic relations among system variables and parameters in order to describe a system's behaviour in quantitative mathematical terms (also called analytical or *numerical model).*
- *Qualitative model :* use of static and dynamic relations among system variables and parameters expressed in symbolic terms in order to describe systems behaviour in qualitative terms such as causalities of if-then rules.
- *Diagnostic model*: A set of static and dynamic relations which link specific input variables -the symptoms- to specific output variables- the faults.
- *Analytical redundancy*: Use of two or more, but not necessarily identical ways, to determine a variable where one way uses a mathematical process model in analytical form.

About system properties and its measures:

- *Reliability:* Ability of a system to perform a required function under stated conditions, within a given scope, during a given period of time. It can be expressed by the Mean Time Between Failure (MTBF). It is the mean value of time passed between two consecutive failures
- *Safety*: Ability of a system not to cause a danger for persons or equipment or environment.
- Other terms such as, a*vailability or dependability*, are less frequent terminology, referring to probability of satisfactory operation of systems through time. They are not used in this text.

Although all of these topics exist in the bibliography and correspond to different stages in the study of faults of plants, the majority of works in the domain are centred on: fault detection, fault diagnosis, monitoring and supervision.

The scope of supervision is not only to detect malfunctions and faults, but also to propose actions against these situations. Therefore, basic tasks associated to a supervisory system have a correspondence with fault diagnosis, (Gentil, 1996), and other fault related tasks. Once faults are detected and localised, actions can be proposed or ordered to assure global performances. See Fig. 2.1 for the relationship between tasks and terminology.

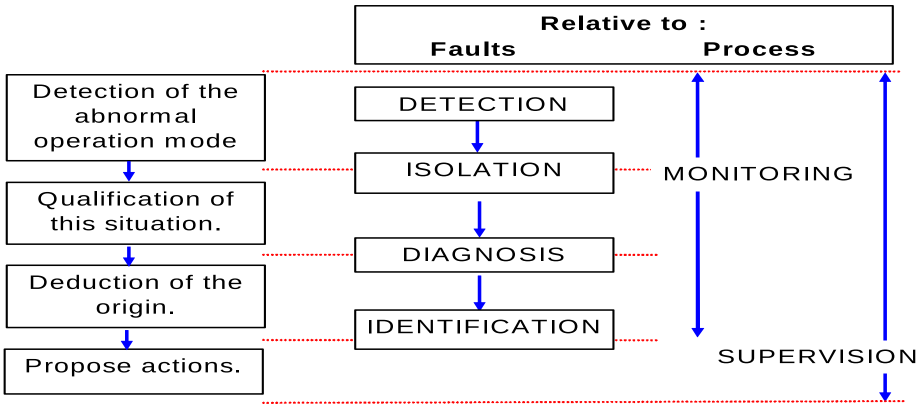


Fig. 2.1 Supervision tasks.

Nowadays, commercial industrial applications cover simple monitoring tasks that consist in data management (storing, visualisation and representation) and alarm generation. This is the case of extended SCADA packages. More advanced systems can diagnose and propose actions, but final decision about alarm certainty or action validity is restricted to human operators.

Several survey papers over the last three decades have summarized much of the research into the fundamental processes of automated FDD. The first major survey was written by Willsky, 1976. Other key survey papers were presented by Isermann, 1984, Gertler, 1988, Frank, 1990, Isermann and Ballé, 1997, Frank and Koppen-Seliger, 1997. The developments in fault-detection methods are also summarized in the books by Himmelblau, 1978, Pau, 1981, Patton et al., 1989, Mangoubi, 1998, Gertler, 1998, Chen and Patton, 1999, Patton et al., 2003 and Isermann, 2006. In (Venkatasubramanian et al., 2003a; Venkatasubramanian et al., 2003b; Venkatasubramanian et al., 2003c) was published a three part series reviewing process fault detection and diagnosis, and Katipamula and Brambley, 2005 review different methods orienting the paper to building systems.

2.2.1. Types of faults

The types of faults depend basically on their location within the system, the number of components that can be affected and their temporal evolution.

Concerning the effects of the faults, they are classified in *additive faults* (those which correspond to sensor and actuator faults) and *multiplicative faults* (or parametric):

- *Additive process faults*: These are unknown inputs acting on the plant, which are normally zero and which, when present, cause a change in the plant outputs independent of the known inputs.
- *Multiplicative process faults*: These are changes (abrupt or gradual) in some plant parameters. They cause changes in the plant outputs which depend also on the magnitude of the known inputs. Such faults best describe the deterioration of the plant equipments, such as contamination, clogging, or the partial or total loos of the power.

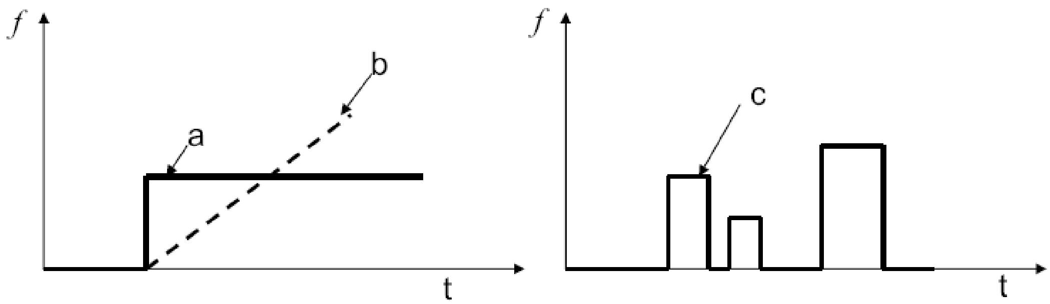


Fig. 2.2 Time-dependency of faults: (a) abrupt; (b) incipient; (c) intermittent

Fault location can be distinguished in:

- *Sensor faults*: These are discrepancies between the measured and the actual values of the individual plant variables.
- *Actuator faults*: These are discrepancies between the input command of an actuator and its actual output.
- *Plant faults*: such faults change the dynamical properties of the system, either by a disturbance entering the process or structural malfunctions.

Regarding the time dependency of faults, they can be distinguished in Fig. 2.2:

- *Abrupt faults*: These are faults that appear "abruptly" in a time instant. For example in a break down of a power supply.
- *Incipient faults*: These are faults that increase steadily and that are brought about by wear.
- *Intermittent faults*: These are faults that do not appear continuously. For example an intermittent electrical contact.

2.2.2. Structures and Methodologies.

Supervision is essentially the set of techniques used with the goal of assuring the integrity of a system. The definition given above assigns to supervision the role of detecting (to recognise and to indicate) in real time abnormal behaviour of a process taken benefit of all information available about the process (measures, models, history, experience and so on).

According to these goals, the main part of supervision of a complex system is focused on to *detect* and *isolate* occurring faults and *provide* information about their size and source (Zhang et al., 2002). The most important and difficult task is centred on fault detection and diagnosis where difficulty increases with the real time constraints and complexity of systems (non-linear, coupled dynamics, time dependencies, etc.). The classical procedure of a fault diagnosis system is depicted in Fig. 2.3. This is achieved in three basic steps, residual generation, evaluation and analysis, not always clearly separable.

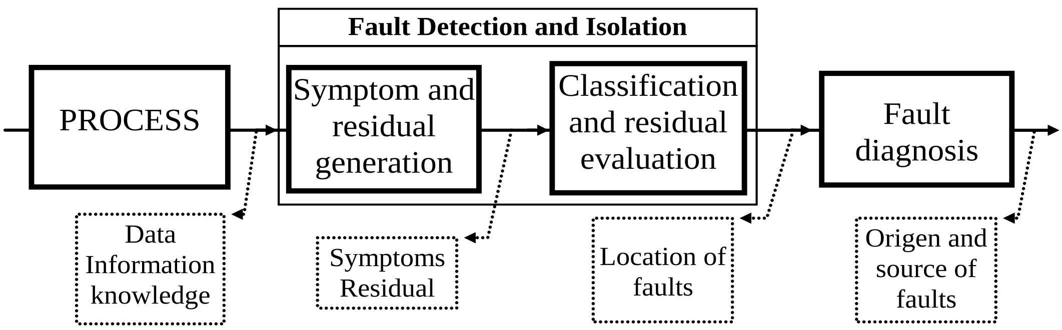


Fig. 2.3 Schematic representation of the procedure of fault diagnosis.

The methodology used in fault detection is clearly dependent on the process and the sort of available information. A distribution of fault detection methods depending on applications is summarised in Isermann and Ballé, 1997. Existing approaches range from analytical methods to artificial intelligence and statistical approaches.

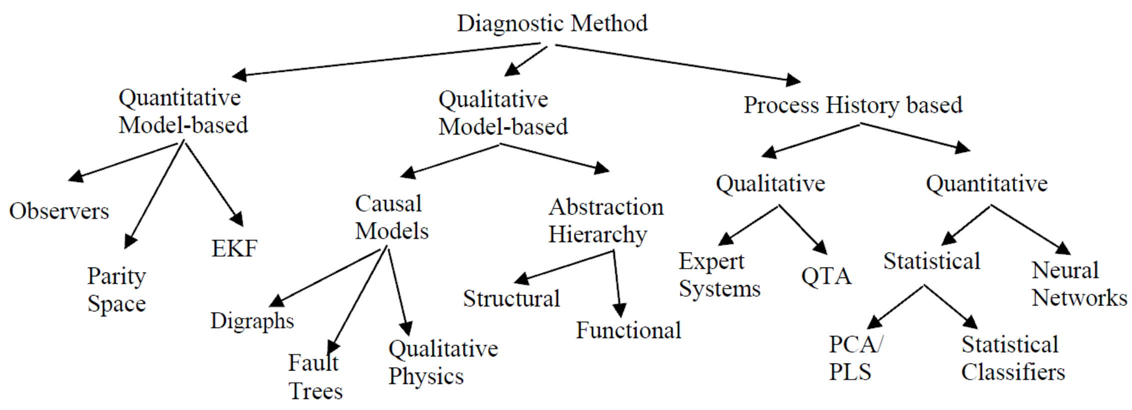


Fig. 2.4 Classification of diagnostic algorithms (Venkatasubramanian et al., 2003a)

Dash and Venkatasubramanian, 2000 and Venkatasubramanian et al., 2003a classify the diagnostic systems (Fig. 2.4) based on the a priori knowledge used, that is, the set of failures and the relationship between the observations (symptoms) and the failures. Thus, the methodology used in fault detection and diagnosis is clearly dependent on the process and the sort of available information. From a modeling perspective, there are methods that require accurate process models, semi-quantitative models, or qualitative models. Most of these methods are based on comparing measures and simulations for obtaining a residual. At the other part of the spectrum, there are methods that do not assume any form of model information and rely only on historic process data. The knowledge-based model offers an alternative to that situation in which an accurate model is difficult to be obtained.

In any case, model based fault diagnosis of dynamic processes is a very active area of research. Approaches coming from two different research communities (FDI community in the Automatic Control area and DX community in the Artificial Intelligence area) are proposing different methodologies that share many concepts and tools. However, in practice there are difficulties in sharing results coming from both communities due to the different kind of formalisms and backgrounds. Puig et al., 2002 try to connect both communities in order to share tools and combine methodologies.

Following a classification close to that of Fig. 2.4, this work classifies the methods within two categories: model-based techniques and non-model based techniques, although some authors can include the techniques stated below in different groups. Model-based techniques can use quantitative or qualitative models. Quantitative models are sets of quantitative mathematical relationships based on the underlying physics of the processes. Qualitative models are models consisting of qualitative relationships derived from knowledge of the underlying physics. In contrast the non-model based techniques are based solely on process history. The main examples of techniques for both categories are described in the following sections. The aim is to provide the general concepts being a good entry point to this field and to place the thesis within the broad scope of the fault diagnosis problem.

2.3. Model-Based Techniques

The model-based diagnosis (MBD) approach rests on the use of an explicit model of the system to be diagnosed (Frank et al., 2000; Isermann, 2005, Angeli, 2008). The occurrence of a fault is captured by discrepancies between the observed behaviour and the behaviour that is predicted by the model. A definitive advantage of this approach is that it only requires knowledge of normal system operations, following a consistency-based reasoning method.

Most of model-based fault detection and diagnosis methods rely on the concept of analytical redundancy (Kinnaert, 2003; Kleer and Kurien, 2003, Ding, 2008). In contrast to physical redundancy, when measurements from parallel sensors are compared to each other, now sensory measurements are compared to analytically computed values of the respective variable. Such computations use present and/or previous measurements of other variables, and the mathematical model of the process describing their nominal relationship to the measured variable. The idea can be extended to the comparison of two quantities generated analytically, obtained from different sets of variables. In either case, the resulting differences, called residuals or analytical redundancy relations are indicative of the presence of faults in the system.

The generation of residuals needs to be followed by residual evaluation, in order to arrive at detection and isolation decisions. Because of the presence of noise and model errors, the residual are never zero, even if there is no fault. Therefore the detection decision requires testing the residuals against thresholds, obtained empirically or by theoretical considerations.

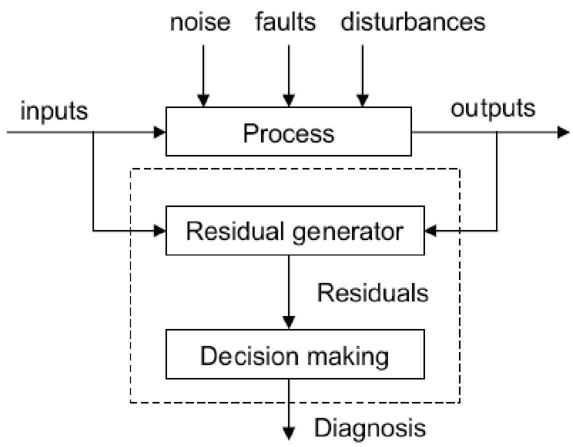


Fig. 2.5 Conceptual structure of model-based fault diagnosis

2.3.1. Principles of model-based diagnosis

Fig. 2.5 depicts the conceptual structure of model-based fault diagnosis system, which concerns two stages: residual generation and decision making. This structure of two stages was proposed by (Chow and Willsky, 1980) and has been accepted by the majority of the scientific community of fault diagnosis. The two stages can be described as follows:

Residual generator

Its aim is to generate a residual or a fault indicator signal, using the inputs/outputs information from the monitored process. This auxiliary signal is created in order to reflect the fault appearance in the process. Residual may be equal to zero or close to zero in fault free situation and different from zero under fault situation. Above means that residual do not depend on the inputs and outputs, in ideal conditions. The algorithm used for generate residual is called residuals generator.

Decision making

Once residuals are obtained they are examined in a module of change detection with the purpose of deciding if there exists a fault (or not). The decision making procedure can be performed by a simple test of thresholds on the instantaneous values or a moving average of the residuals, or even statistical theory can be used. Threshold logic, statistical decision theory, pattern recognition, fuzzy decision making or neural networks are actual methods used to decide whether and where a fault has occurred.

2.3.2. Fault diagnosis techniques based on analytical models

Different kinds of process models and methods can be used to generate residuals of state or output variables. The more used techniques for residual generation by means of analytical models are: observers (Chen and Patton, 1999; Mendonça et al., 2009), parity equations (Gertler, 1991, Gertler, 1998), parameters estimation (Isermann, 1997) and structural analysis (Staroswiecki et al., 2000).

Observer based methods

It is one of the most known residual based techniques. The basic idea of the observer or filter-based approaches is to estimate the states or outputs of the system from the measurements by using either Luenberger observers in a deterministic setting (Beard, 1971; Frank, 1996) or Kalman filters in a stochastic case (Willsky, 1976; Basseville, 1988). Then, the error between real data and estimated data or a function of them is used as residual. The flexibility in selecting observer gains has been studied (Frank and Ding, 1997). The freedom in the design of the observer can be utilized to enhance the residuals for isolation. The dynamics of the response can be controlled, within certain limits, by placing the poles of the observer.

The extension of existing results of this linear technique to the non-linear case is not an easy task. Alcorta García and Frank, 1997 present a survey of the principal observer-based approaches to FD for deterministic non-linear dynamic systems. Some approaches can solve non-linear problems but expressed in special forms (Zhang et al., 2002). Because of this, a complete solution to the fault detection and isolation problem is still unsolved.

Parity (consistency) relations

Parity equations are rearranged and usually transformed variants of the input-output or state-space models of the plant (Gertler, 1991; Gertler and Singer, 1990). The essence is to check the parity (consistency) of the plant models with sensor outputs (measurements) and known process inputs. The idea of this approach is to rearrange the model structure so as to get the best fault isolation. Parity relations concepts were introduced by (Chow and Willsky, 1984). Further developments have been made by (Gertler et al., 1990; Gertler et al., 1995; Staroswiecki and Comtet-Varga, 2001) among others.

There is a fundamental equivalence between parity relations and observer based methods. Both techniques produce identical residuals if the generators have been designed for the same specification (Frank, 1990; Gertler, 1991; Ding and Jeinsch, 1999).

Parameter estimation

The parameter estimation approach detects faults by the estimation of parameters within a dynamic system, where the faults are assumed to be reflected by these features (Isermann, 1984; Isermann, 1997, Patton et al., 1999). The system parameters can be classified as physical and abstract parameters, which directly and indirectly represent the status of a real system, respectively. As, in most practical cases, these parameters are not obtained, parameter estimation methods are applied by measuring the input and output signals, provided that the first principle (physical principle based) model is well known (Isermann 2005). However, due to the difficulty of constructing an accurate model for a complex non-linear system, the application of this method is often restricted to simple linear systems.

A relationship, though weaker, has been found between parity relations and parameter estimation as well (Delmaire et al., 1994; Gertler, 1995; Gertler, 2000).

Structural analysis

The structural analysis is the study of the system properties, which are independent of the actual values of the parameters. Only links between the variables and parameters are represented in this analysis. These links result from the operating model and are called relations or constraints. They are independent of the operating model and are thus independent of the form under which this operating model is expressed (qualitative or quantitative data, analytical or non-analytical relations). The links are represented by a graph, on which a structural analysis is performed (Izadi-Zamanabadi and Staroswiecki, 2000; Blanke et al., 2006). The main advantages of the structural analysis approach are: it determines the part(s) of the system on which some Analytical Redundant Relation (ARR) (Puig et al., 2004) can be generated, and it is used to obtain the calculation sequences of the ARR.

2.3.3. Qualitative model based methods

In these techniques the knowledge is obtained from the structure and the behaviour of the process as a set of relations that describe the interactions between various process variables. The goal is to dispose of a rough model to be used as a model-based approach. Contrary to the analytical model, the qualitative models can be incomplete or contain uncertainties.

In this section, some causal model-based methods will be briefly discussed. Cause-effect arguments are a basic component of human reasoning about system behaviour, and a causal model reflects the causal relationships between process variables.

Signed Directed Graph (SDG)

The most widely used form of causal knowledge is the Signed Directed Graph (SDG). The process variables are represented as graph nodes and causal relations by directed arcs. Nodes have qualitative states, then, the state of the system is described qualitatively by a pattern. The cause-effect graph is a subgraph of the signed digraph consisting of valid nodes (any variable which is first affected by the root cause) and consistent branches (a consistent path for the propagation of the influence of its initial node to its terminal node).

A problem with the SDG process models is that they only describe local, direct causalities between variables. To overcome this problem, Oyeleye, 1989 introduces the Extended SDG, which analyzes the loops in the SDG and insert additional non-physical arcs into the graphs.

In Wilcox and Himmelblau, 1994 the Possible Cause-Effect Graph (PCEG) was presented as a generalization of the SDG. There are two concepts involved in representing the process state relative to the PCEG: the representation of the complete state using a pattern, and the representation of incomplete knowledge of the process state using a constraint.

Li and Wang, 2001 presents a methodology for qualitative modelling and simulation of the temporal behaviour using a fuzzy clustered digraph. The qualitative information is represented by several classes, obtained as clusters using PCA to categorically characterize dynamic trends of individual variables. The quantitative information is introduced by the utilization of fuzzy c-means clustering approach for automatic fuzzy grouping of the data points in the PCx plots. The study is focused on simulation, rather than on fault diagnosis, and as a data based method it needs extensive training.

Recent works considered the use of wavelets as signal preprocessors in order to perform SDG in processes with load-fluctuations (Tsuge et al., 2000).

Fault Tree Analysis (FTA)

FTA is an analysis technique for safety and reliability aspects that uses a graphical representation to model causal chains leading to failures. The fault tree is a logic tree that begins with the top event (incident) and continues by deductive reasoning through all the intermediate events to primary events and initiating events. The tree usually has layers of nodes. They provide a computational means for combining logic to analyze systems faults. At each node different logic operations like AND and OR are performed for propagation. The main difference between a SDG and a fault tree representation is in the primary unit that makes them. In a fault tree, the primary unit is an event, while in a SDG, the primary unit is a process variable.

The fault tree is constructed by asking questions such as what could cause a top level event. In answering this question, one generates other events connected by logic nodes. The tree is expanded in this manner till one encounters events (primary events) which need not be developed further (Lapp and Powers, 1977). Once the fault tree is constructed, the next step in the analysis is the evaluation of the fault tree. Fault trees are usually generated manually. Considerable knowledge, system insight and overview are necessary to consider various failure modes and their consequences at a time. Because of this, the effort can be diminished by the automation of Fault Tree generation (Liggesmeyer and Rothfelder, 1998; Mäckel and Rothfelder, 2001).

Qualitative physics

Qualitative physics is an area of AI concerned with modeling a physical system in order to simulate it or solve particular problems regarding the system (Ramil and Smith, 2002). Three examples of qualitative models without using graphical representation but focused on representing the dynamics of systems using equations are:

- The Qualitative Simulation (QSIM) method by Kuipers, 1986 which represents qualitative behavior using qualitative differential equations. Qualitative modeling involves specifying a constraint model of the physical process in terms of qualitative versions of mathematical relationships such as addition, multiplication and differentiation. QSIM representation and simulation algorithm allows to reason mathematically about the description of the process.
- Qualitative Process Theory (QPT) construes physical systems as consisting of entities whose changes are caused by physical processes (Forbus, 1996). The domain is described by a collection of objects and each of these objects

completely defines a qualitative state. The qualitative state is defined by a set of parameters which take on values in a quantity of space.

- Compositional modelling is a strategy for organising and reasoning about models of physical phenomenon. It uses explicit modelling assumptions to decompose domain knowledge and semi-independent model fragments, each describing various aspects of objects and processes (Falkenhainer and Forbus, 1991).

2.4. Non-model-Based Techniques

Model-based fault detection requires process variables (measures) to compare real process response and model response. This comparison is performed under the assumption that the same input is provided for both systems. Therefore, process measures and actions must also be supplied as input to the model. Other methods can be applied if only process outputs are available.

In non model-based techniques, past experimental records are analysed in order to detect irregularities which would link the observed data (the symptoms) with the final conclusions (the diagnosis). The non-model techniques can be divided into two categories: signal-based approaches and knowledge-based approaches, although sometimes the dividing line between the categories is not clearly defined.

2.4.1. Signal-based approaches

Signal-based methods are focused on analysing signal features. Proper signals, or symptoms, are extracted from the system, which carry significant information about the fault of interest. The symptoms are either directly, or after proper modifications, used for fault diagnosis. Typical symptoms are: the magnitudes of the time functions of measured signals, limit values, trends, statistical moments of amplitude distribution or envelope, spectral power densities or frequency spectral lines, correlations coefficients, covariance and so on.

Knowledge of the system is assumed to consist of learning associations between process measures and operating conditions. In this sense, they could be considered as knowledge-based methods. There are numerous approaches of signal based methods; some of them are as follows:

- Physical redundancy: multiple sensors measure the same physical variable finding any discrepancy among them.
- Statistical techniques: mean, variance, control limits, entropy, etc. are estimated.
- Frequency analysis: Some plant measurements have a typical frequency spectrum under normal operating condition; any deviation from this is an indication of abnormality.
- Limit checking: plant measurements are compared to fixed thresholds.
- Probabilistic: Bayes decision.

The use of statistical techniques for FD is based on viewing diagnosis in terms of quality control. Statistical methods aspire to give early detection. The calculations provide an early warning indicator of effects on the process, allowing for correction at the best possible moment. For example, an instrument is used to take measurements on a certain quality variable. If the measurement is within the control limits, it is assumed that the instrument is working normally. If the measurement falls outside the control limits, the instrument is statistically out of control. Statistical Process Control (SPC) has been widely used to analyse a process, or its outputs, so as to take appropriate actions to ensure stable levels of quality within the process. If the majority of the processes are multivariate in nature, then, a Multivariate SPC (MSPC) has to be used.

A statistical technique that has a wide area of applications is Principal Component Analysis (PCA). It involves a mathematical procedure that transforms a number of correlated variables into a smaller number of uncorrelated variables called principal components. So, PCA permits a reduction in the dimensionality of data while retaining the useful information, computing a compact and optimal description of the data set. PCA has been extensively used for fault detection, fault diagnosis, process monitoring and sensor faults. The generation of principal components and their use is described elsewhere (Jolliffe, 1986; Efthimiadu et al., 1995; Martin and Morris, 1996).

Basically, the application of PCA to FD consists in the calculus of the Squared Prediction Error (SPE) of residual space for the *i*th sample of process variables. The process is considered normal if the value SPE_i is between the control limits. Another index often used for FD is the Hotelling T^2 test. It is an overall measure of variability. For new data, the score space is monitored by computing the T^2 value as the sum of the squares of the selected scores scaled by the respective eigenvalue computed from data representing normal operation. Other monitoring charts can be displayed (e.g., plots of one principal component against another). From the analysis of different monitoring charts a fault can be diagnosed.

Several applications performing PCA can be found in the literature. Kourti and MacGregor, 1996 examine the application of some traditional statistical process control methods, including PCA, for process monitoring in a variety of industrial processes. Dunia and Qin, 1998 develop an approach for process and sensor fault identification and reconstruction based on principal component analysis. Singhal and Seborg, 2001 and Ge and Song, 2008 use a PCA similarity factor to develop both methodologies for pattern matching in multi-variate time-series. Yoon and MacGregor, 2004 present a multiscale PCA algorithm and procedures for both fault detection and isolation. Furthermore also the combination of PCA with other techniques has been successful. Maurya et al., 2003 present a PCA-QTA technique for fault diagnosis. Lu et al., 2003 develop a wavelet-based time-frequency approach to improve PCA-based methods by extending the time-domain process features into time-frequency information. In (Fourie and de Vaal, 2000) multiscale wavelet decomposition is first performed on process data. Then, linear PCA and Nonlinear PCA are performed separately and an ANN is trained using the linear PCA transformed data set as the input layer and the non-linear principal scores as the output layer.

2.4.2. Knowledge-based approaches

In the case of noticeable modelling uncertainty, a more suitable strategy is that of using knowledge-based methods. Instead of output signals, any kind of symptoms can be used and robustness can be attained by using only those symptoms which are not strongly dependent upon the systems uncertainty. In this case, knowledge which is often incomplete and cannot be represented by analytical models has to be processed.

These methods are a field in continuous evolution, where AI techniques have an important role. There is not a unified theory to be applied to these methods and, in fact, knowledge-based methods can be applied to all three phases of fault diagnosis; namely residual generation, residual evaluation and fault analysis, although the phases in this case are not always as clearly separable as in the case of the analytical approach.

Knowledge based methods correspond mainly to classification methods and pattern recognition approaches. These techniques can use the history knowledge either qualitatively or quantitatively based on the nature of subsequent analysis. Expert Systems, Artificial Neural Networks and Qualitative Trend Analysis are the main examples of techniques of this group.

Symptoms based methods

They consist in organizing the expert knowledge that is used to link observations with solutions. Then, these techniques deal with process variables to identify fault symptoms in them, where a symptom is a subjective evidence that indicate the existence of a fault. Symptoms-based methods allow a forward reasoning from fault to possible faults. These methods can be performed with different techniques: Expert Systems, Fuzzy logic (Isaza et al., 2009), Case Based Reasoning (CBR). In such implementations, major difficulties exist in the knowledge acquisition task and knowledge representation.

Expert Systems are also known as rule based systems. They use sets of rules to match observed symptoms to causes. Typically, rule based (RB) methods are made up of an antecedent part (series of events) and a consequence part, which maps these events to a known fault or some other event, which forms the antecedent of some other rule and so on.

An improvement respect to these methods corresponds to the use of fuzzy logic in the RB methods. In the case of Fuzzy Logic Systems (Tarifa and Scenna, 1997) the rules are put in a fuzzy way. A rule-based system can also be viewed as a lookup table, which contains explicit mapping of known symptoms to root causes.

In (Kempowsky et al., 2006; Kempowsky et al., 2005) a tool (SALSA) based on fuzzy methodology of conceptual clustering and classification is used to perform situation assessment. First, an offline learning stage allows obtaining behavior patterns from training data. So SALSA identifies several classes representing operations in the system that they need to be validated. Once the behaviour pattern is elaborated, online data is analyzed to recognize what is the current functional state of the process assigning it to the class with the highest adequacy. If a new situation cannot be classified to any existing class, the approach can generate new classes from the unrecognised patterns.

Qualitative trend-based methods

This approach transforms data from the process variables in description of its trends in an explicit and meaningful form. Qualitative abstraction of variables represents valuable information that can be used to examine the current status of the plant. The procedure for Qualitative Trend Analysis (QTA) has three main components:

1. The language used to represent trends.
2. The method for identification of trends in variable readings.
3. Interpretation of those trends in terms of fault scenarios.

The qualitative representation of trends has fundamental elements called primitives. Groups of primitives form episodes and episodes combine to form a trend.

A method used for primitive identification can be based on first and second derivatives of the process trend calculated using the finite difference method. Another method is the use of an ANN.

Observed symptoms or trends, possibly in combination with other information, trigger the memory of similar situations in the past. The primitives thus identified are used in a knowledge base (KB) to perform fault diagnosis.

More details about this method are given in Chapter 3.

Neural network based methods

The Artificial Neural Network (ANN) based methods have received considerable attention over the last few years. ANNs are inspired by biological neural systems. They consist of a network of many simple units (neurons) where each neuron can compute an output that is a (generally nonlinear) function of the weighted sum of inputs. The weights are parameters of each connection in the net. Using a set of training data, the weights of all connections are adjusted so that the output matches the correct classification as closely as possible. After proper training, a good net may perform very well in classifying unknown input data. The training is quite time-consuming, and may require a large set of training data.

In the petrochemical industry ANNs have been used as supervised pattern classifiers. They are trained on historical or simulated steady state process data with the aim of detecting a specified number of suspected faults.

The first reports (Hoskins and Himmelblau, 1988; Venkatasubramanian and Chan, 1989; Watanabe et al., 1989; Venkatasubramanian et al., 1990) show the application of Backpropagation networks (BPNs) using sigmoidal functions in the first layer. In more recent studies, Radial Basis Function networks (RBFNs) are preferred because they provide more reliable generalisation and fewer extrapolation errors (Gomm et al., 1998; Yu et al., 1999).

Many successful studies have been reported on integrating wavelet transformations with neural networks (Zhao et al., 1998). Wavelets functions are more localized and pick up edge effects.

Ruiz et al., 2001 presented a fault detection and diagnosis scheme that incorporates the advantages of neural networks but as a supplement to a fuzzy system in a block oriented configuration. The neural networks require no explicit coding of knowledge and the fuzzy system provides insight into the problem-solving process.

Self Organizing Maps (SOMs), which are trained unsupervised, are not always able to classify data correctly. However, their ability to classify data autonomously is very interesting and useful when real industrial processes are considered (Koivo, 1994).

Regarding the special case of faults in sensors, auto-associative neural networks have been showing good results. Their application is based on the Nonlinear Principal Component Analysis (NLPCA) technique (Kramer, 1992). Furthermore, a robust auto-associative neural network can be used to gross error detection, identification and removal into a singles step. Dong and McAvoy, 1996 suggest a method that uses principal curves and three-layered neural networks. Mo et al., 1998 suggest NLPCA that is based upon functional-link auto-associative neural network (the input layer is expanded by using the concept of functional link).

Recent works enhance the use of the ANN framework for FD by big improvements in the following issues: speed of training, introduction of time explicitly into the classifier design and on-line updation using a mirror-like process model (Rengaswamy and Venkatasubramanian, 2000).

2.5. Motivation to incorporate expert knowledge in diagnosis systems

The complexity of designing diagnosis approaches has been introduced in the previous sections. The need to incorporate expert knowledge in these designs is present in all tasks involving supervisory systems. Fault detection can be performed by using analytical models, but these models are not always available and the final decision concerning the residual generated is always submitted to a human expert. On the other hand, the use of knowledge-based representations is needed by numeric to qualitative interfaces in order to reason about process variables. The important role of human knowledge in process supervision is to concentrate attention on the techniques that permit the use of expert knowledge to automate those tasks.

Nowadays, the main contribution for designing supervisory systems comes from the experience of operators and expert engineers. In fact, they are also present in the majority of applications for final decision making. Sometimes, the description and translation into computers of this expert knowledge, related to process variables, becomes very difficult or impossible due to the different nature of human descriptions and data obtained from process.

Usually, experts describe situations or estimations of those situations, while data is instantaneous samples of measures. In the procedure of matching the evolution of process variables and these situations, humans use an imprecise description of magnitudes. An example can clarify these difficulties: the following sentence, "when *temperature* in the reactor *increases*, *open* the input *valve slowly*", describes an action (*open valve slowly)* to be performed when a process variable (*temperature*) experiments certain behaviour (*increasing*). This expert description is easily interpreted by humans, but difficult to interface with numerical magnitudes coming from the process (*temperature*) or actuators to perform this action (*open valve*). They are imprecise descriptions of numerical magnitudes available in the process. This imprecise description must be processed before being used in the control structure. The representation of this kind of information and the ability to deal with the relationship between imprecise variables is in the scope of qualitative methods. The use of qualitative techniques implies a description of process variables given by short sets of labels or symbols (*low*, *normal*, *high*). The number to symbol translation must be a reliable task despite the great deal of numerical data and the imprecision, uncertainty or incompleteness of measured signals. At the same time the result must be useful for the supervision tools.

An additional inconvenience of using expert knowledge for process supervision refers to temporal references. Usually it describes process behaviour in an uncertain period of time, or changes in the evolution of process variables without dating these events. In the example cited in the previous paragraph, the label *increasing* is related to a characteristic of a process variable during an imprecise period of time. This consideration must be taken into account when building numeric to qualitative interfaces in order to benefit from these kinds of descriptions of variables evolution. This also applies to more general descriptions of process behaviour such as *transient* or *steady state*, for instance. In this case, both possibilities are exclusive, but real transition between both states is gradual. Then difficulties exist in determining the limits between both, because it includes all the process variables analysed. Thus, time is a fundamental variable in studying dynamic systems but, at the same time, its qualification is difficult. Control systems normally use a synchronous representation of time, with a constant sampling rate but the interesting events are not periodic. Thus, supervisory systems need asynchronous representations of time.

Knowledge-based methods try to use expert knowledge of processes directly. These methods are more process dependent, since they use specific abstraction tools (abstractors) that must be designed to carry as much information as possible about the faults to be detected. Abstractors can be used as analysis tools to obtain significant information from process signals at several abstraction levels, which is from the measured signal, to more elaborated information about it, in the sense of its compactness and representativeness. This thesis proposes QTA to represent qualitative trends of signals (tendencies, oscillation degrees, alarms, degree of transient state, etc.). One of these techniques is the representation of signals by means of episodes. An episode can integrate numeric and symbolic data at several levels of abstraction. The next chapter describes this type of representation in more detail.

The task of diagnosis can be viewed as a classification problem or a pattern recognition task, and thus the diagnostic system is also referred to as a diagnostic classifier. Whenever an abnormality occurs in a process, a general diagnostic classifier would come up with a set of hypotheses or faults that explains the abnormality. The completeness of a diagnostic classifier would require the actual faults to be a subset of the proposed fault set. The resolution of a diagnostic classifier would require the fault set to be as minimal as possible. Thus, there is a trade-off between completeness and resolution.

Furthermore, classification through the comparison and matching of temporal signals (process measurements) when performing fault diagnosis can also be affected by the time misalignments within the measurements. For instance, in symptom-based methods, such as case-base reasoning, a representative historical database of signals (cases) that have been analysed previously and suitably annotated, is used to identify the root cause of a change (fault) and develop an effective remedy (diagnosis). A hard problem in this method is to locate an instance (case) in the historical database (case-base) that is the most similar to specific data. Pattern classification or signal comparison is a popular method for finding similar signals in historical data. The challenge in this approach results from the fact that, because of the nature of industrial processes, signals that result from two instances of the same change are not exact replicates invariably; there are deviations between the two instances. The differences could be in the length (total time) of the two signals or in the magnitudes or profiles of the variables. Therefore, the direct comparison of two signals would be incorrect, because there is no guarantee that the corresponding segments of the signals are being compared. Consequently, robust yet sensitive methods for comparing unsynchronised signals are an active area of research.

Part of this thesis has focused on the task of developing a new tool to perform qualitative trends from process variables. Thus, the main body of effort has been aimed at providing a useful tool for calculating a similarity measure between different trends. These representations are composed by qualitative and quantitative information in an asynchronous time domain. A set of case examples has been developed to test both techniques.

2.6. Conclusions

The aim of this chapter is twofold: first to review some of the definitions and methods used in the field of FDD, and second, to place this work within the broad scope of fault diagnosis.

The methods have been classified into two categories: model-based methods and non model-based methods. The use of one or the other category is only subject to the knowledge of process behaviour or faults.

Although quantitative model-based methods are most accurate and reliable they have two major shortcomings: the complex technology or natural process is generally for nonlinear time-varying systems, which makes it particularly difficult to detect structural changes in the system and to obtain adequate models for this purpose. Secondly, the model available is often assumed to represent normal operating conditions, and the impact of a departure from these conditions on the model outputs is difficult to predict. Consequently, these methods are difficult to apply to a dynamic process submitted to repeated changes in the operation mode.

On the other hand, when analytical models are hardly available, knowledge-based models are a realistic alternative, allowing one to exploit as much knowledge of the process as is available, but, for instance, expert systems need expert knowledge of the process.

The main difficulty with knowledge-based methods is the translation of the numeric values (data coming from the process) to qualitative data (symbols) that can be used with these techniques. This problem has motivated the first part of this thesis, therefore the next chapter is devoted to presenting some definitions and the main formalisms in the qualitative representations and their use for process monitoring and fault detection and diagnosis.

Chapter 3.

Qualitative Representations of Signals

3.1. Introduction

Over recent years most industrial plants have been rebuilt or restructured, and the process control equipment modernised to include a wide range and number of sensors. Process data is usually stored in databases to be analysed by experts, but as the amount of data stored increases, the task of extracting useful information becomes more complex and difficult. Also, the growing complexity of control systems makes it more and more difficult to make decisions. However, the large volumes of information collected at industrial plants by modern control systems have to be used to improve efficiency and productivity, to avoid unscheduled shutdowns and abnormal situations. All this information must be represented with the aim of studying it and concluding new knowledge or making decisions about a problem. A simple way of knowledge representation are rules (if, then). It is also possible to represent it using trees, diagraphs or case-bases. Experts usually use qualitative language to describe system behaviour, for instance: *high*, *medium*, *low*. There is an important part of the AI community that studies how to represent qualitative information and to deal with it in a way similar to natural language and thought processes.

The study of qualitative sequences has been often reported by researchers as Qualitative Trend Analysis (QTA) (Venkatasubramanian et al., 2003c) or Qualitative Shape Analysis (Rengaswamy et al., 2001). In contrast to qualitative methods using relationships derived from knowledge of the underlying physics (Manders et al., 1999; Kuipers, 2001), QTA is a non model-based technique that uses historical data from previous experience of the process.

The variation of a process variable concerning time is called the trend of that variable. A process trend has an intuitive meaning about how process behaviour changes over time. A qualitative trend is obtained from one or more signals (or functions of them), adequately processed. The objective is to represent the signal using qualitative symbols or shapes, also called primitives by some authors (Rengaswamy and Venkatasubramanian, 1995), that have a meaning within the context of the process.

Qualitative trends based on episodes and their analysis is a knowledge-based method for extracting features, that allows the diagnosis of situations if the episodes are processed by others tools, for instance, using fuzzy logic, such as reference (Dash et al., 2003) to pattern recognition. In this case, the dimensionality of variables is not altered but the complexity of the system states is reduced allowing only a finite set of qualitative descriptions. A benefit of this representation is that it allows the extraction of meaningful information, that is, the interpretation by an expert operator who would usually suffer from an excess of process data.

In this thesis the method of representing knowledge is to build qualitative trends based on episodes which represent dynamic behaviours in the process. Thus, in this chapter, several formalisms and approaches concerning qualitative representations are described.

3.2. Qualitative representations based on episodes

Signal representation by means of episodes provides a good tool for situation assessment. On the one hand, uncertainty, incompleteness and heterogeneity of process data make qualitative reasoning a useful tool. On the other hand, reasoning not only with instantaneous information, but using the historic behaviour of the process is necessary. Moreover, since a great deal of process data is available for the supervisory systems, to abstract and use only the most significant information is required. The representation of signals by means of episodes provides an adequate response to these necessities.

Episodes are a portion of qualitative information that has a specific meaning for a specific representation. In a qualitative context based on episodes, a trend is defined as the sequence of qualitative episodes over time. The general concept of episode in the field of qualitative reasoning was introduced by Williams, 1986, who defined an **episode** as a set of two elements: a time interval, named **temporal extent**, and a **qualitative context**, giving the temporal extension significance.

This definition allows an episode to be defined as explicitly as the qualitative context. The description of variable behaviour over time is referred to as a **history** or **trend**. A qualitative trend is a contiguous, non-overlapping sequence of episodes; then, trends use a qualitative representation of time because they break down time into a set of regions of interest. To achieve the desired descriptions, every episode must achieve the largest continuous interval of time in which the variable maintains a single qualitative value.

The motivation to use representation by episodes comes from the needs of supervisory systems (Williams, 1986,Konstantinov and Yoshida, 1992,Gentil, 1996). These needs can be summarised as:

- *Need for qualitative reasoning*: uncertainty, incompleteness and heterogeneity of process measurements make the qualitative representation of signals a good tool for supervisory tasks. Moreover, the heterogeneity of the knowledge that has to be used in supervisory tasks makes qualitative reasoning a necessary tool in many cases.
- *Need for temporal reasoning*: in the field of supervision, reasoning, not only about instantaneous information, but also about the historic behaviour (trajectory) of processes is necessary. For this reason, reasoning about time is essential. In process control, time is usually sampled in a uniform way; but the important events for supervision (e.g., faults) happen in an asynchronous way. Therefore, asynchronous representations will be more useful than synchronous ones.
- *Need for a compact knowledge representation*: since a great deal of process data is available for supervisory systems, it is necessary to abstract and use only the most significant information. This implies a qualification of the signal behaviour. The abstraction process could be done in a temporal way (qualifying time) or by qualifying a signal characteristic.
- *Need for natural knowledge representation*: knowledge about processes often comes from human operators and engineers' experience. To make the use of this kind of knowledge easier, signals must be represented in a natural way.

Clearly, the problem of trend identification is a difficult task. The disparity in rates of fault evolution highlights the need for automated identification to be extremely robust while using minimal or no a priori information about behaviour or process data. The presence of sensor noise further complicates the task. So, formalisms have to take into account some important issues (Dash et al., 2004):

- *Time-scale of identification*: Process behaviours changes depending on events that occur in the process. The changes in the behaviours can be slow or quick depending of the disturbances, faults or noise present in the signal process. So, time-scale trends have to be adapted over the time.
- *Noise*: In a real process the noise in the process signals disturbances the information about process behaviours, but sometimes filters can eliminate important information in the trend.
- *Scale variant nature*: The representation of qualitative trends depends on the scale which they are observed. If the process signals have important variations in the values ranges over time, it will be necessary to adjust parameters in the representation, such as size of samples window for the detection of these changes in the trend. Also, it is possible to obtain several trends in different scales of observation.
- *Simplicity*: Trends have to be easy to understand in order to extract information and reason about it. Also the computational complexity of the algorithm should not be prohibitive to restrict its usefulness.

Cheung and Stephanopoulos, 1990 exposed the conditions that have to be assumed for a formal representation of process trends:

- *Existence*: a trend is a physical entity with real values. A trend is an integral description of the temporal description of the temporal transformation of process behaviour as detected by sensor measurements.
- *Stability and localisation of its characteristics*: a trend must be stable to small perturbations and its geometrical features can be detected and localised in the time scale.
- *Uniqueness in scaling*: it is possible that different scales of observation exist for data observation and therefore for the trend, but the trend of a variable is unique for a given scale in a given time interval.
- *Functional characteristics*: the trend must represent the physical transformation of process behaviour over time. In the formal representation geometrically distinguished time points are defined: they represent a transition point for a change in level (for instance a change from negative to zero), change in direction (for instance a change from increasing to decreasing at a maximum point) or a change in the curvature.

Several qualitative representations based on episodes have been developed in order to give general representations for any process signals. In the following subsections, some of these formalisms and approaches are described.

3.2.1. Triangular and trapezoidal representation of process trends

A general formalism for the qualitative representation of trends is developed in Cheung and Stephanopoulos, 1990; introducing the concept of trend as a continuous sequence of qualitative states. The qualitative state *QS* of a variable *x* at $t \in [a,b]$ is defined as the triplet of qualitative values characterised by the signs of the first and the second derivative.

$$QS(x,t)=\begin{cases} undefined & \text{if } x \text{ is discontinuous at } t \\ \langle [x(t)],[dx(t)],[ddx(t)] \rangle & \text{otherwise} \end{cases} \quad [3.1]$$

where:

$$[x(t)] = \begin{cases} + & if \quad x(t) > 0 \\ 0 & if \quad x(t) = 0 \\ - & if \quad x(t) < 0 \end{cases}$$
$$[dx(t)] = \begin{cases} + & if \quad x'(t) > 0 \\ 0 & if \quad x'(t) = 0 \\ - & if \quad x'(t) < 0 \end{cases} \quad [3.2]$$
$$[ddx(t)] = \begin{cases} + & if \quad x''(t) > 0 \\ 0 & if \quad x''(t) = 0 \\ - & if \quad x''(t) < 0 \end{cases}$$

Thus, an episode of the variable *x* is defined as any temporal interval $(t_i,t_j) \subset [a,b]$ such that *QS(x,t)* is constant $\forall t \in (t_i,t_j)$. So an episode is the pair

$$< t\text{-}extent, QS(x,t_i,t_j) >$$

defined as the temporal extent of the episode $t\text{-}extent=(t_i,t_j) \mid t_i < t_j$ and the qualitative state of *x* over (t_i,t_j), $QS(x,t_i,t_j)=QS(x,t) \quad \forall t \in (t_i,t_j)$.

Each episode then represents an interval of uniform behaviour where all the qualitative properties are constant.

This approach has a practical extension in the triangular and trapezoidal representations. A triangular episode is the triangular region constructed by the intersecting lines of the initial slope, the final slope and the average slope through the boundary points (Fig. 3.1). For any time interval (t_i, t_j) a triangular episode is defined by the set:

$$< [ddx], (t_i, t_j), <x(t_i), x'(t_i)>, <x(t_j), x'(t_j)>>$$

such that the qualitative value of [ddx] is constant at (t_i, t_j)

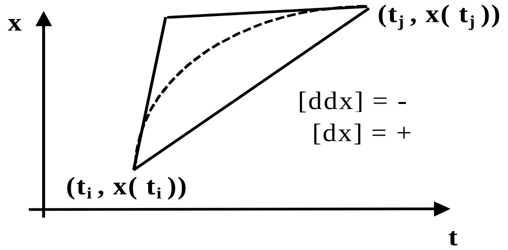


Fig. 3.1 A triangular episode.

Triangular episodes can be considered as geometric primitives used for modelling episodes, and consequently trends. Every trend can be represented by a series of episodes and the most concise representation results when all the episodes are maximal. There are seven basic types of episodes (Fig. 3.2) defined by the qualitative values alone; these seven types are defined under the constancy of [dx] and [ddx].

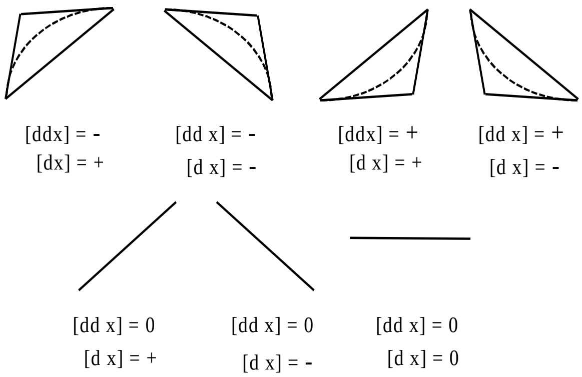


Fig. 3.2 Set of triangular episodes.

The triangular representation allows explicit information of significant information of a trend to be given. Qualitative features like '*increasing*', '*spikes*', '*oscillations*', etc., can be directly expressed by the type of matched triangular shape. At the same time quantitative information, like coordinates of extrema and inflexion points, are also available.

The trapezoidal representation is based on trapezoidal episodes. A trapezoidal episode (Fig. 3.3, Fig. 3.4) is an episode where qualitative context is defined by the constancy of the qualitative value *[ddx]*. In addition, it includes the value and the slopes at the boundary points and the convexity point of the episode. The boundary points of episodes are the second order zero crossings and the convexity points are defined as the point at which the slope is equal to the slope of the line joining the boundary points of the episode.

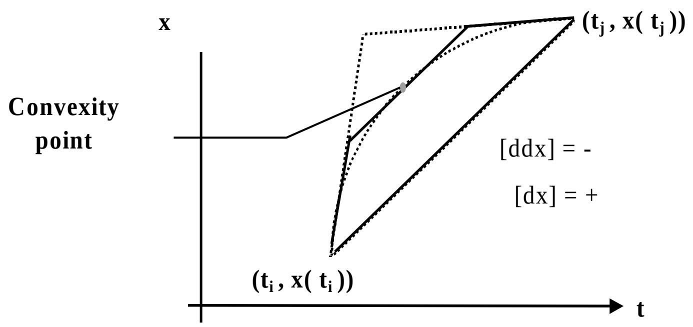


Fig. 3.3 A trapezoidal episode.

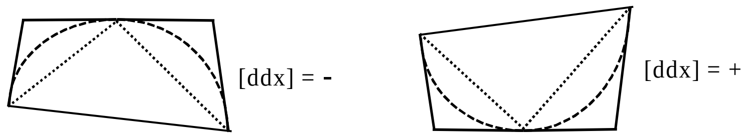


Fig. 3.4 Types of trapezoidal episodes.

Trapezoidal episodes can be built using the triangular representation by grouping consecutive triangular episodes, or they can also be built from another trapezoidal representation, grouping trapezoidal episodes and obtaining representations in different temporal scales. This method is called qualitative scaling; it is a local scaling and brings out the local character of trends with minimum distortion. A similar trapezoidal representation is also given by Ayrolles, 1996.

The trend-based window approach suggested by Cheung and Stephanopoulos, 1990 allows the trends to be viewed from different scales. In each window, the trend segment is represented by an initial slope, a final slope, and a line segment connecting the two critical points. A series of such segments represents a process trend.

This representation was later used by Bakshi and Stephanopoulos (Bakshi et al., 1994; Bakshi and Stephanopoulos, 1994a; Bakshi and Stephanopoulos, 1994b) to develop a methodology to extract temporal characteristics from trends, based on multiscale analysis using wavelets. The methodology consists of three steps: First, a wavelet signal decomposition that acts as a noise removing filter; second, the triangular representation of smoothened process signals and finally, a search algorithm that makes use of decision trees and Shannon's entropy comparisons for the identification of certain classes of process outcomes. In Stephanopoulos et al., 1997 this methodology has been implemented as a part of a broader system aimed at fermentation database mining, diagnosis and control.

The same representation is used in Wong et al., 2001 as a basis for classification of tendencies. First, signals are filtered by means of wavelets and next the triangular episodes are obtained. Next, logical fuzzy is used to convert the quantitative values of magnitude and duration related to the episodes in another symbolic representation including 'small, medium, large' symbols representing these quantitative values. As a result the 7 initial symbols become 57. Then classification of sequences is carried out by means of HMM (Hidden Markov Models), that they convert the input in a numeric probability, and compared with another method that uses a back-propagation neural network (BPNN). The methods are applied to a simulation of a tank reactor.

More recently, Villez et al., 2008 used the cubic spline wavelet to obtain a filtered signal in different scales. The resultant signal is qualified according to its first derivative and two letters are used to represent upward and downward behaviour, although they do not include temporal information. The approach is illustrated by a diagnosis example where the resulting qualitative representation is looked up in a dictionary of trends. Finally, in Villez et al., 2009 the seven primitives are used in order to represent different faulty profiles. They conclude that qualitative analysis may be a good task for the automation of fault detection and diagnosis tasks.

3.2.2. Qualitative Temporal Shape Analysis

Konstantinov and Yoshida, 1992 propose a representation of signals based on an expandable set of profiles. Each profile (*qshape*) is characterised by two symbolic strings (*SD1* and *SD2*) representing the evolution of first and second derivative signs of the signal.

$$
\begin{aligned}
&SD1\ x\ t\ = \mathrm{sd1} = \ +,-,\dots \\
&SD2\ x\ t\ = \mathrm{sd2} = \ +,-,\dots \\
&t \in\ t_1, t_2
\end{aligned}
\qquad [3.3]
$$

Then, the qualitative shape of a signal is represented by the combination of these strings:

$$
\begin{aligned}
&\mathrm{qshape}\left[x\ t\ \right] = \left[SD1\ x\ t\ , SD2\ x\ t\ \right] = \left[\ +,-,\dots\ ;\ \ +,-,\dots\ \right] \\
&t \in\ t_1, t_2
\end{aligned}
\qquad [3.4]
$$

Hence, two temporal shapes are considered qualitatively equivalent if their *qshapes* coincide. The analysing procedure extracts in real time *SD1* and *SD2* over a predefined time interval [t_1,t_2] and compare them with those of an expandable shape library that stores all interesting shapes (Fig. 3.5).

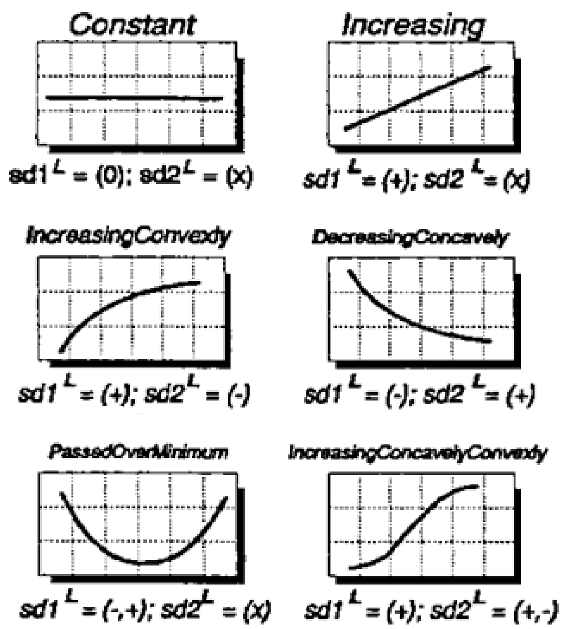


Fig. 3.5 Some elements of the expandable set, represented by sd1 and sd2.

The time-scale of analysis is fixed a priori which limits its generic applicability and it is possible that the representation of trends can be multiple. Also, with increasingly complex shapes, the library can become quite big and the simple reasoning based on the extent of derivatives sign matching may not be suitable.

3.2.3. Trends Description Language (TDL)

Janusz and Venkatasubramanian, 1991 propose a qualitative description (TDL) of signals consisting of primitives, episodes, trends and profiles. Primitives are based on the sign of first and second derivatives (positive, zero or negative). Then, nine basic types, represented in Fig. 3.6, make up the set of primitives.

The first step in obtaining the signal profile is the identification of each signal sample with a primitive (Fig. 3.7b). This step is solved as a pattern recognition problem by means of neural networks. Then, consecutive samples with the same primitive are grouped to build episodes (Fig. 3.7c). The trend of a signal consists of a series of episodes, and finally the profile is obtained by adding quantitative information (Table 3.1). In this case, the proposed quantitative information is the signal value at the boundary points and the length of the episode, which can be considered the minimum to identify subtle distinctions between trends.

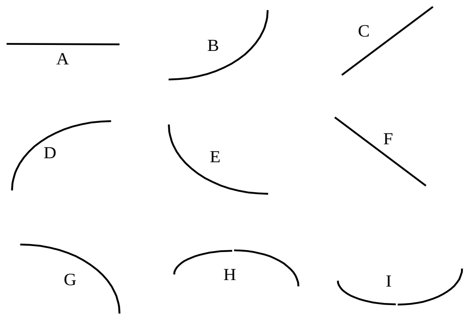


Fig. 3.6 Set of primitives.

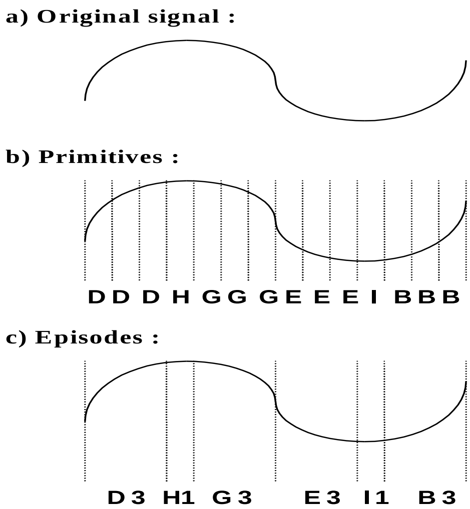


Fig. 3.7 Episodes (primitive & length) recognition.

Table 3.1 Signal Trend (series of episodes) and Profile (episodes & signal value at the boundary points) corresponding to Fig. 3.7.

Signal Trend:	(D3)(H1)(G3)(E3)(I1)(B3)
Profile:	(6.1 D3 7.3)(7.3 H1 7.3)(7.3 G3 6.4)(6.4 E3 5.5)(5.5 I1 5.5)(5.5 B3 7.0)

Later, Rengaswamy, 1995; Rengaswamy and Venkatasubramanian, 1995 extended this method using syntactic pattern recognition involving an error correcting code (ECC) acting as a postprocessor to rectify inconsistencies due to noise and discontinuities. The primitives were identified from the sensor data using a neural network. Each data set in a given time window would be classified to one of the primitives.

Vedam and Venkatasubramanian, 1997 improved the framework proposed in Rengaswamy and Venkatasubramanian, 1995 with an adaptive algorithm for the identification of trends based on wavelets. The algorithm can adjust the time window in the input of neural nets to adapt the identification to the distinct dynamics of the process. The primitives identified are used as input to a knowledge base in order to perform fault diagnosis. This system, called W-ASTRA is demonstrated on a fluidised catalytic cracking unit. Later Vedam et al., 1998 proposed a B-Spline based compression method to identify piecewise linears for automatic trend extraction wherein the window is adaptive to the sensor trend and avoids the use of neural networks.

Another trend detection approach based on wavelets is suggested in Flehmig et al., 1998 to localize intervals in which the measured process signal is well approximated by a polynomial. The method is illustrated using measurements derived from a industrial desalination plant.

An improvement based in Rengaswamy and Venkatasubramanian, 1995 is developed by Rengaswamy et al., 2001 for the automated detecting and diagnosing of different kinds of oscillations in control loops. Initially they consider just a restricted set of primitives: increasing, decreasing, steady. To identify the primitives, a time window is chosen to perform a local structure identification procedure using a feed-forward neural network. Then a global time-scale identification procedure looks for significant changes in the qualitative shapes. With the identification of global time-scales, the algorithm can distinguish between four types of shapes: namely, increasing, decreasing, oscillating, and steady.

Using the primitive-based language in (Rengaswamy and Venkatasubramanian, 1995), Dash et al., 2004 proposed a novel approach to automate the identification of process trends based on an interval-halving procedure. The idea is to fit polynomials to the data and to identify qualitative primitives. For this purpose first data length is recursively halved until a unimodal region (quadratic) with acceptable error is found. The second part comprises of assigning primitives to the piecewise polynomials and the labelling is based on the sign of the first and second derivatives. Some guidelines for tuning the parameters used in the framework also are presented. The same technique is used in Dash et al., 2003 where a fuzzy logic-based multivariate framework is presented for inference the trend information with the behavior or state process. The application of this approach was illustrated in the fault diagnosis of an exothermic reactor case study.

Afterwards, a similar work was published in Maurya et al., 2007 applied to the fault diagnosis of the Tennessee Eastman process. In this work they defined some similarity measures and confidence index (C.I.) used later in Maurya et al., 2010. Here, an online implementation of the interval-halving algorithm is developed. The key feature of the algorithm is the use of an adaptive window size. They discuss a framework for online fault diagnosis where the presence of a non-A primitive (steady behaviour) and the departure from the NOR (normal operating region) indicates the presence of a fault. After detection, they estimate the time at which the fault occurred in order to extract an appropriate portion from the sequence of primitives to compute the similarity between the extracted trend and the trends from a database.

Based on the 7 primitives, Sundarraman and Srinivasan, 2003 decided to use only first-order trends, since they are simpler and more robust to noise. The authors describe a process variable as an ordered collection of enhanced atoms. An enhanced atom consists of a first-order shape, the time duration for which that shape is manifested, and the variable magnitudes at the beginning and end of the shape. This trend, including quantitative information, is called an enhanced trend. Then, they defined a distance measure as the maximum value obtained from matching degree to shape, magnitude and duration for two trends.

3.2.4. A general formalism

The majority of the representations described in the previous section are based on the evaluation of the sign of the first and the second derivatives. The proposal described in Colomer, 1998 and Meléndez and Colomer, 2001 is the extension of this formalism to both qualitative and numerical context in order to be more general. It means that a general formalism must be able to build episodes according to any feature extracted from variables (e.g., the level of noise, a threshold overtake or some deviation with respect to a normal value). In fact, some analytical methods of fault detection or diagnosis use other information apart from the first or the second derivative of signals (Basseville, 1988).

Since this formalism is used as a basis of the toolbox to represent qualitative trends in this work, the main definitions are identified below.

The initial representation of time and signals

The starting point for qualitative representation must be the measured signal. Supposing that the signal is obtained at discrete time instants $[t_0,t_1,...,t_i]$, the time and the measured signal values can be represented in a natural way as follows:

$[t_0,t_1,...,t_i]$: time

$[x(t_0),x(t_1),...,x(t_i)]$: signal

$t_i < t_j \Leftrightarrow i<j$

The final objective, that is the qualitative representation, can be obtained from this initial quantitative representation.

Characteristic Function

Characteristic function is the basis of the qualitative representation presented. The value of a characteristic function F at a time instant t_i can be any function of the signal at t_i and at the previous time instants, the previous characteristic function values and the time:

$$F(t_i) = f(x(t_0), x(t_1), ..., x(t_i), F(t_0), F(t_1), ..., F(t_{i-1}), t_0, t_1, ..., t_i) \quad [3.5]$$

Some examples of characteristic functions can be

- the signal:

  $F(t_i) = x(t_i)$

- its first derivative (numerical approximation):

  $F(t_i) = (x(t_i)-x(t_{i-1}))/(t_i - t_{i-1})$

- some transforms (FFT, Wavelet transform...),
- its level of noise, etc

This characteristic function must be chosen according to the interesting features of the signal (and process) and to the supervisory system goals (changes to be detected). When more than one characteristic function is needed to describe all the interesting variable characteristics, the previous notation is expanded to:

$$F(t_i) = \langle F_1(t_i), F_2(t_i), ..., F_{nf}(t_i) \rangle \quad [3.6]$$

with n_f the set of characteristic functions.

Characteristic Interval, Characteristic State, Qualitative State

The qualitative description of a signal needs the definition of its qualitative state. In order to provide it with significance, this definition must be made according to the concept of characteristic function as defined above.

The simplest way to obtain qualitative states from characteristic functions is the qualification of its range of values. In this way, the range of values of each characteristic function is divided into a set of contiguous and non-overlapping intervals, called **characteristic intervals**. Thus, the **characteristic state** of a characteristic function indicates that the value of the characteristic function belongs to a characteristic interval.

Let n_j be the number of characteristic intervals for a characteristic function F_j :

$[I_j^1, I_j^2, \ldots I_j^{nj}]$: set of characteristic intervals of the characteristic function F_j.

$[S_j^1, S_j^2, \ldots S_j^{nj}]$: set of symbols associated with characteristic intervals I_j^i $\forall$i.

$QF_j(t_i)$: Characteristic state of F_j at t_i.

Then, for any k:

$$QF_j(t_i) = S_j^k \Leftrightarrow F_j(t_i) \in I_j^k \quad [3.7]$$

and for a time interval:

$QF_j(t_i, t_j)$: Characteristic state of F_j at $[t_i, t_j)$

$$QF_j(t_i, t_j) = S_j^{\,k} \Leftrightarrow F_j(t) \in I_j^{\,k} \quad \forall t \in \left[t_{i,} t_j\right) \qquad [3.8]$$

The range of values of each characteristic function can be qualified in several ways, depending on the process behaviour and the supervisory system characteristics. In Fig. 3.8 a), b) c) and d) some ways of choosing the characteristic intervals are shown. There are infinite possibilities, but a large amount of characteristic states also means a poor abstraction capability from the point of view of representation compactness.

Finally, the **qualitative state** of a variable, which qualitatively represents all its interesting characteristics (given by the characteristic functions and the characteristic intervals), is defined as the set of characteristic states for each characteristic function:

$$QS(x, t_i) = < QF_1(t_i), QF_2(t_i), \dots QF_{nf}(t_i) > \qquad [3.9]$$

and for a time interval:

$$QS(x, t_i, t_j) = < QF_1(t_i, t_j), QF_2(t_i, t_j), \dots QF_{nf}(t_i, t_j) > \qquad [3.10]$$

This definition supplies a generalisation of the concept of qualitative state.

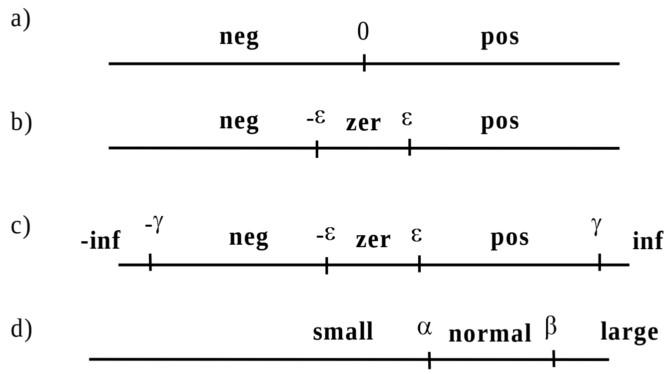


Fig. 3.8 Some possibilities (a), b), c) and d)) in the choice of characteristic intervals.

Characteristic Instants

One of the most important goals of the representations of signals in episodes is the qualification of time, dividing it into significant intervals. That is, simplifying the signal from the temporal point of view. This simplification is done by the temporal extension of the episodes, which has to be determined from the changes in the qualitative state.

Consider a variable $x(t)$, a **characteristic instant** is defined as a time instant where there is a change in the qualitative state of $x(t)$. Formally, t_i is a characteristic instant of $x(t) \Leftrightarrow QS(x,t_i) \neq QS(x,t_{i-1})$

Episodes, Fundamental and Auxiliary characteristics

An episode $\mathbb{E}_k$ is defined as a set of numerical and qualitative values, including the qualitative state QS and the left and right characteristic instants (t_l, t_r):

$$\mathbb{E}_k = \; < t_l^{\,k}, QS^k, t_r^{\,k} > \tag{3.11}$$

The qualitative state and the left and right characteristic instants are the necessary data that describe the temporal extension and the qualitative context of episodes; so, it can be called the **fundamental characteristics** of an episode. It is noted that the width between the pair of characteristic instants t_l, and t_r is not necessarily the same for each episode.

Also, there are other qualitative or quantitative data that can be interesting from the point of view of process supervision but not necessary to qualify the time (to determine the temporal extension of episodes). These characteristics can be called **auxiliary characteristics** to distinguish them from the fundamental characteristics. An example of auxiliary characteristic is the ‘convexity point’ in the trapezoidal representation.

Therefore, an episode $\mathbb{E}_k$ will be a set of qualitative and quantitative data as follows:

$$\mathbb{E}_k = \left\langle t_l^{\,k}, QS^k, \text{ auxiliary characteristics }, t_r^{\,k} \right\rangle \tag{3.12}$$

with the conditions:

$$\begin{cases} t_r^{\,k} = t_l^{\,k+1} \\ QS^k = QS\ \ \mathrm{x}, t_l^{\,k}, t_r^{\,k} \quad \text{unique over } \left[t_l^{\,k}, t_r^{\,k} \right. \\ QS^k \neq QS^{k+1} \end{cases} \qquad [3.13]$$

This last condition can be expressed as:

$$QF_j^{\,k} \neq QF_j^{\,k+1} \text{ for any } F_j \qquad [3.14]$$

In this thesis, an episode $\mathbb{E}_k$ follows the expression:

$$\mathbb{E}_k = \left\langle QS^k, d^k, \text{ auxiliary characteristics } \right\rangle \qquad [3.15]$$

$$d^k = t_r^{\,k} - t_l^{\,k}. \qquad [3.16]$$

Finally, the qualitative representation of a variable will be composed by a sequence of episodes:

$$\boldsymbol{\mathbb{E}} = \left\langle \mathbb{E}_1,\ \mathbb{E}_2,\ \mathbb{E}_3 \ldots\ \mathbb{E}_N \right\rangle \qquad [3.17]$$

which qualitatively describe a variable over a period of time limited by the left characteristic instant of the first episode and the right characteristic instant of the last one.

3.2.5. Example of representation based on the general formalism

According to this formalism, a new representation to describe signal trends depending on the second derivative as characteristic function is proposed. Derivatives are computed by means of a band-limited FIR differentiator (Colomer and Meléndez, 2001) in order to avoid noise amplification. Then, the range of values of the second derivative is divided in five disjoint intervals by means of two significant values ε and γ (Fig. 3.9).

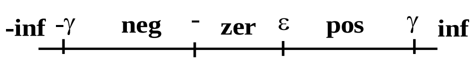


Fig. 3.9 Characteristic intervals.

Consequently, 5 qualitative states (Table 3.2) compose the fundamental set of episodes, representing the qualitative values of the characteristic function.

Table 3.2 Set of qualitative states

$$QS\,(x,t_i) = QF\,(t_i) = \begin{cases} -inf & if\ F(t_i) \in (-inf, -\gamma] \\ neg & if\ F(t_i) \in (-\gamma, -\varepsilon) \\ zer & if\ F(t_i) \in [-\varepsilon, \varepsilon] \\ pos & if\ F(t_i) \in (\varepsilon, \gamma) \\ inf & if\ F(t_i) \in [\gamma, inf) \end{cases}$$

In order to obtain a more significant representation, the qualified first derivative (Fig. 3.10) at the characteristics instants (tl and tr, beginning and end of each episode) is used as a qualitative auxiliary characteristic. Then, a set of 13 types of episodes is obtained (Fig. 3.11). A major benefit of this set of episodes for supervisory tasks is that discontinuities and stability periods (usual in fault situations and in normal situations respectively) are explicitly represented by means of 5 types of episodes (⏋⎾—⏌⎿).

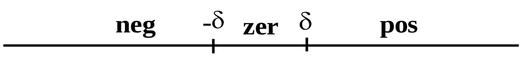


Fig. 3.10 Qualitative values for first derivative.

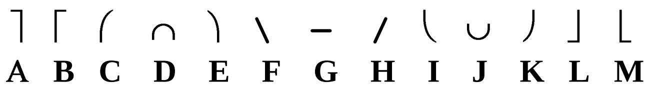


Fig. 3.11 Set of episodes.

As example the Fig. 3.12 shows the qualitative representation of a signal obtained step by step.

An approach based on this representation is used in Ramil and Smith, 2002 to extract qualitative information from different systems. Then they compare this information against the theoretical results from various software evolution models.

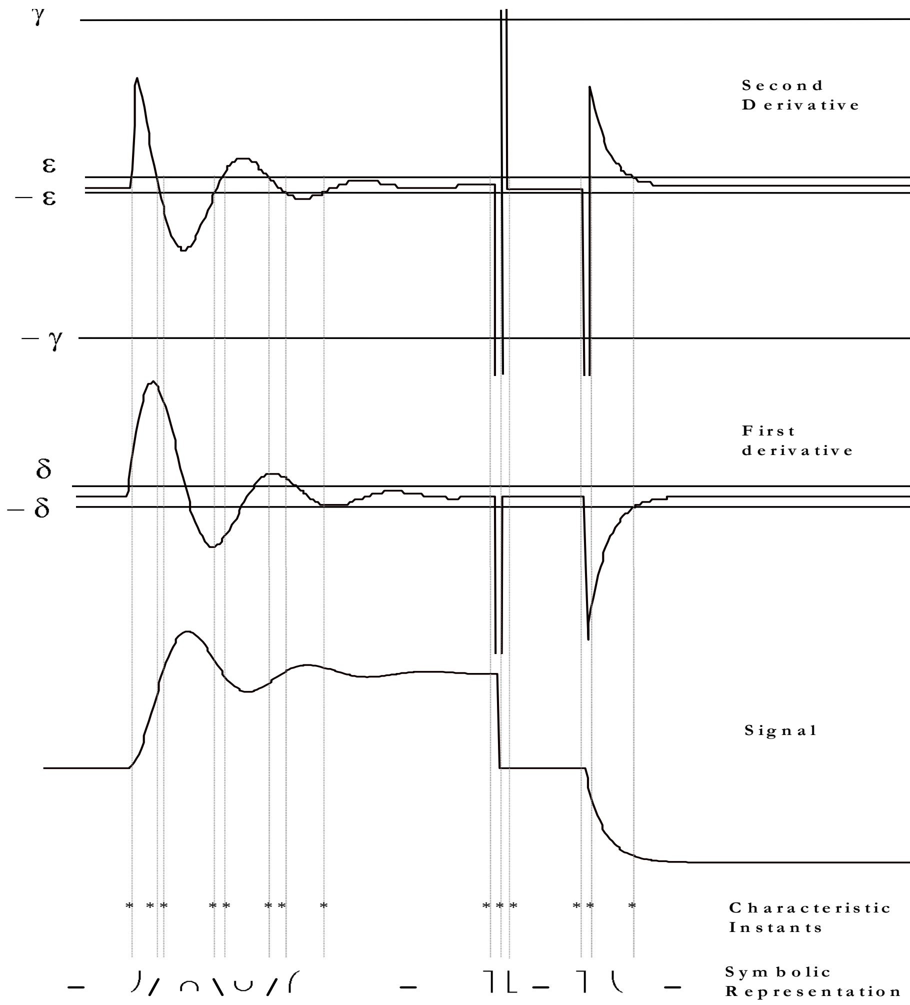


Fig. 3.12 Example of a qualitative representation

3.3. Other representations

Many other studies about temporal series have developed techniques for dimensionality reduction that can be interpreted as qualitative representations. These techniques provide results similar to those described in the previous section but they cannot be considered as based on episodes according to the definition introduced by Williams, 1986. The next section is dedicated to temporal representation by obtaining a linear approximation of raw data as the most common technique. Finally, a method known as SAX is explained.

3.3.1. Linear segments

One of the areas of interest in data mining is the representation of time series, so many high level representations have been proposed (Faloutsos et al., 1994; Geurts, 2001;

Keogh et al., 2001a). Fig. 3.13 illustrates the most commonly used representations, each can be visualised as an attempt to approximate the signal by a linear combination of basis functions.

On the other hand, linear approximation can represent the mental models given by human operators. Thus, this type of representation could be qualified and used for the same purposes as previous methods.

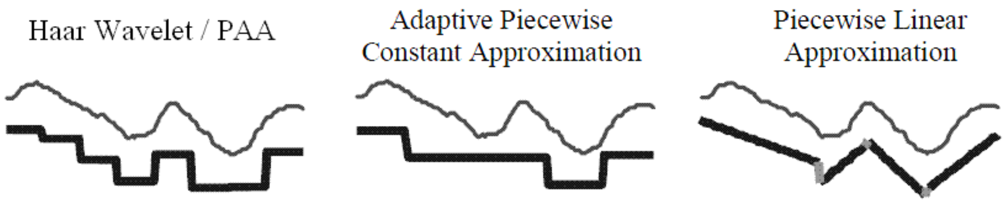


Fig. 3.13 The most common representations for time series data (Lin et al., 2003).

Haar wavelets

Haar wavelets are the fastest to calculate and the easiest to implement among wavelet family. The Haar transform can be seen as a series of averaging and differencing operations on a discrete time function. For an input represented by a list of 2^n numbers, the signal is averaged, pairwise, to get the new lower-resolution representation of data with values. This process is repeated recursively, pairing up the averages to provide the next scale, finally resulting in $2^n - 1$ differences and one final average. So the first wavelet bases, representing different resolution levels, can be combined to produce an approximation of the original sequence. More information can be found in (Struzik and Siebes, 1999; Chan et al., 2003).

Piecewise Aggregate Approximation (PAA)

Piecewise Aggregate Approximation (PAA) approximates a time series by dividing it into *k* segments of equal length, and then uses the mean value of each segment to form a feature vector to represent the original sequence (Keogh et al., 2001b; Yi and Faloutsos, 2000). The PAA representation can be identical to the Haar wavelet representation as is demonstrated in Keogh and Pazzani, 2000b. PAA is used in a symbolic approach called SAX (see section 3.3.3).

Adaptive Piecewise Constant Approximation (APCA)

Keogh et al., 2001a extend the idea of PAA to Adaptive Piecewise Constant Approximation (APCA). APCA is similar to PAA, except that it allows arbitrary length segments. Thus, there may be several short segments representing the regions with great fluctuations, while there may be fewer long segments representing the flat and featureless regions. This significant feature enables APCA to have a smaller reconstruction error than that of PAA.

Piecewise Linear Approximation

Intuitively, piecewise linear approximation (PLA) refers to the approximation of a time series T, of length n with K straight lines, where $K<<n$. The algorithms which return a piecewise linear representation from time series are called segmentation algorithms. There are different algorithms for determining the approximating line, the most common use techniques such as linear interpolation and linear regression (Chen et al., 2007; Keogh et al., 2003).

Other works obtain similar results without reference the PLA method. The authors in (Galati and Simaan, 2006) present an automatic decomposition approach called ALESDA, a least squares error algorithm supplemented with a combinatorial search algorithm capable of finding an intuitive decomposition in the form of simple primitives such as ramps, steps, and impulses.

Yamashita, 2006 suggests a very simple methodology for diagnosing valve stiction. The method determines typical patterns from valve-input and valve-output in the control loop. It consists in the approximation of time trends of signals by means of three simple symbols: *increasing* (*I*), *decreasing* (*D*) and *steady* (*S*). The identification of proper symbols is based on the calculation of the derivative of the signal and its normalization with respect to mean and standard deviation. The identified series of symbols are combined to form a time series of movements which is the basis for calculating a stiction index.

Charbonnier and Gentil, 2007 splits the data into linear segments and classifies the latest segments into seven shapes: *Steady*, *Increasing*, *Decreasing*, *Positive* or *Negative Step*, *Increasing/Decreasing* or *Decreasing/Increasing Transient*. After that, they transform the obtained shapes into 3 types of episodes defined as {[*steady*, *increasing*, *decreasing*], duration, extreme values}. The approach is used to recognise specific situations for Intensive Care Unit patient monitoring by means of simple rules.

Later, the approach was improved and presented as the online auto-tuning trend extraction method in Charbonnier and Damour, 2008. Basically, the methodology consists of three successive steps. First, a segmentation algorithm splits the data into successive line segments. Siegel's repeated median filter is used as a linear approximation method (Fried et al., 2006). Then the difference between the linear approximation calculated by the segmentation algorithm, $y(k)$, and the measured signal, $y_m(k)$, is calculated. This variable is called the residual and it will be used to estimate the signal variability. In the second step the segments are transformed into episodes. The primitives are either increasing or decreasing, depending on the sign. Otherwise, it is steady. The last step is to aggregate, if possible, the current episode to the previous one to form the longest possible episode.

3.3.2. Shape Definition Language (SDL)

Agrawal et al., 1995 present a Shape Definition Language (SDL) for retrieving objects contained in histories based on shapes. The method consists of the conversion of the original data into a string of symbols, where a symbol is a label describing classes of transitions. The idea is to divide the range of the possible variations between adjacent values in a collection of disjoint ranges, and to assign a label for each one of them. So the behaviour of a series may be described taking into account the transitions between consecutive values. Fig. 3.14 shows an example of a translation using the set of symbols (*Down, down, stable, zero, up, Up*). Every string of symbols may describe an infinite number of curves. The language also consists of different operator, allowing some operations between the obtained profiles and their comparison based on matching their labels.

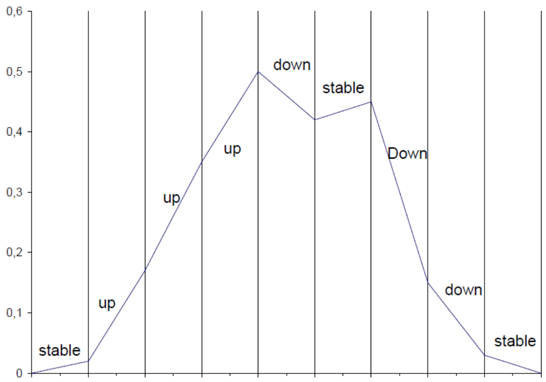


Fig. 3.14 Time sequence and assigned labels.

3.3.3. Symbolic Aggregate approXimation (SAX)

A symbolic representation approach called SAX (Symbolic Aggregate approXimation) is presented in Lin et al., 2003. First, the normalised time series of length *n* is transformed into the Piecewise Aggregate Approximation (PAA) representation and then the PAA representation is symbolised into a discrete string (Fig. 3.15) or word of longitude *w*. Each PAA coefficient has an associated symbol determined by breakpoints. The breakpoints produce *a* equal-sized areas under Gaussian curve and the symbols are assigned from down to up. Note that the breakpoints are similar to characteristic intervals and so the Gaussian distribution can be another type of characteristic function.

The parameters *w* controlling the numbers of elements and *a* controlling the granularity of each element have to be chosen empirically. This may present a drawback because if the reduction scale (the ratio *n/w*) is larger, the chance of information loss increases. Otherwise, if *n/w* is small, PAA becomes meaningless.

Since the appearance of SAX several authors have adopted the method in their work. Some examples are given below.

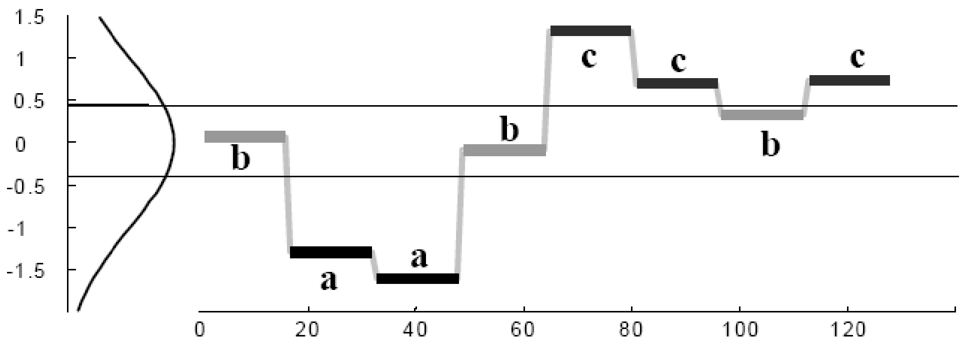


Fig. 3.15 A normalised time series is discretised into the symbolic string baabccbc (Lin et al., 2003).

The authors in Tanaka et al., 2005 transform time series data into a symbol sequence by SAX algorithm. Previously, for multi-dimensional time series data, PCA has been chosen as reduction method to obtain 1-dimensional time series data since they need to maintain only the characteristic patterns that appear frequently in the original multi-dimensional time series data. Finally, an algorithm using the concept of Minimum Description Length (MDL) extracts the most frequently occurring pattern (motif) (Lin et al., 2002). The concept of MDL is to select the best model which compresses data well. The experiments show this approach to be useful in extracting motifs, although if the contribution ratio of time series for first principal component is low the motifs extracted may not be representative.

In Hung and Anh, 2007 a SAX-based pattern matching is supplemented by a post-processing step which applies PLA to compare the patterns in a more direct manner.

SAX was also adopted in Minnen et al., 2007 where discretisation and random projection are used to reduce the comparison space size and the number of candidate motif subsequences respectively. Time series motifs are approximately repeated subsequences in a longer time series data (Lin et al., 2002, Vahdatpour et al., 2009).

3.4. Conclusions

In this chapter several formalisms and approaches about qualitative representations have been described.

Using variables as qualitative trends based on episodes reduces the complexity of the system state allowing only a finite set of qualitative descriptions. A benefit of this representation is that it allows the extraction of meaningful information, that is, the interpretation by an expert operator who would usually suffer from an excess of process data. So, qualitative process trends providing an intuitive meaning about process behaviour can be used for monitoring, or could be supplied to other fault detection and diagnosis tools, expert systems or classification methods.

Finally, it is possible to extract qualitative representations from many other studies about dimensionality reduction for temporal series. The last section is dedicated to these techniques.

The general formalism described by Meléndez and Colomer, 2001 suggests the possibility of representing the signals by using any interesting characteristic. This formalism allows representations of signals based not only on signal dynamics but also on any user-defined characteristics according to signal behaviour and supervision objectives. Moreover, in order to improve the representation usefulness, some additional characteristics of episodes can be used. These characteristics are called *auxiliary characteristics* and bring about a more significant set of episodes.

This approach will be the basis of the tool presented in the next chapter. Although some other approaches could be used, this is the most general one. In this way, results obtained by using it should be easily extrapolated.

Chapter 4.

Qualtras: a generalised tool to generate online qualitative episodes

4.1. Introduction

In the previous chapter, a general formalism to extract qualitative representations was described. This formalism will be the basis of the representation based on episodes from now on. So, its implementation has been carried out to develop a tool capable of generating episodes online for different variables. The next section summarises the CHEM Project, within which the toolbox Qualtras was developed, and afterwards its characteristics are described.

4.2. Origin and Motivation

Qualtras (‘‘Qualitative Trend Analysis Software’’) was developed as a result of our participation in the project GROWTH CHEM ‘Advanced Decision Support System for Chemical/Petrochemical Manufacturing Processes’ (Cauvin and Celse, 2004). The CHEM Project resulted from an initiative by the IMS consortium (Intelligent Manufacturing Systems) and brought about an international collaboration in order to improve the supervision of complex plants.

The aim of this project was to build flexible software consisting of advanced and specialised toolboxes in order to improve safety, product quality and operation quality as well as to reduce the economic losses from faulty states in the refining, chemical and petrochemical processes. The result was the integration, in a modular fashion, of a number of software tools based on different techniques developed by European research and academic institutions, for process monitoring, fault detection and diagnosis.

The objective of one of the work packages was to conduct situation assessment via process trend analysis, in view of imprecise or non-existent models. This was performed by extracting qualitative and semi-qualitative information from process measurements and by building several layers of meaning and representation of process measurements. Within this context we developed the toolbox ‘Qualitative representation of process trends variables’, later named as Qualtras. This tool allows the obtaining of an asynchronous qualitative and quantitative representation (episodes) from input signals based on its most significant features.

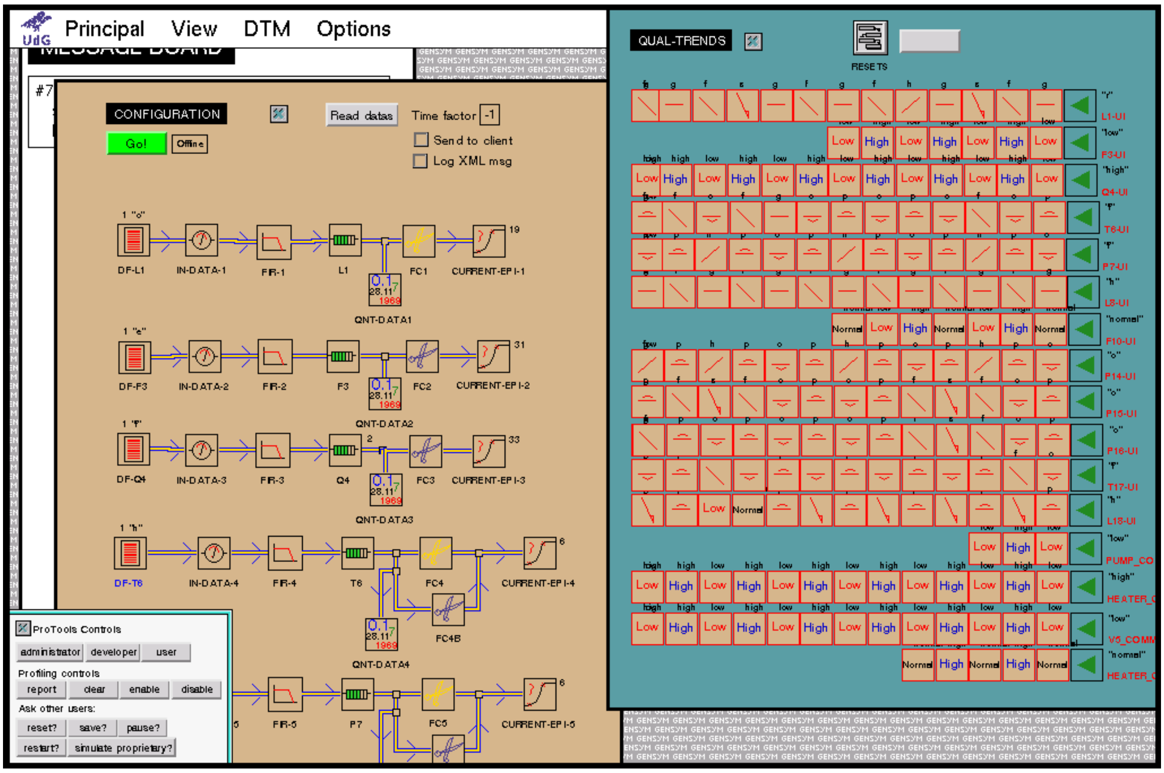


Fig. 4.1 A general view of the toolbox Qualtras implemented in G2.

The aim of such a type of description is to offer an understandable representation to the operator or to be used as a numeric to qualitative interface for supervisory applications. A set of desirable characteristics for such a tool have been identified as follows:

- Real-time operation.
- Modular design to enable different performance.

- Open architecture to enable integration of new or external modules.
- Ability to include rules (it can work alone or to feed other tools).
- Universal communication language among software applications.

4.3. The architecture of Qualtras

Qualtras facilitates the abstraction of the most significant characteristics of the signals, representing any process signal by means of episodes. It allows a compact representation to be obtained and shows the qualitative process perception that expert operators could have (e.g., ‘‘the pressure dropped abruptly’’). That is, it provides an intuitive representation which is easier to interpret. So the user can extract valuable information from the output data, or can give information to a fault detection expert system. Qualtras was delivered as a software library in the real-time expert system shell G2 (Gensym Corporation) and can be used to build customised applications in an easy way (Fig. 4.1). G2 allows object-oriented and graphical programming of real-time application. Blocks managing information are represented by objects. Operation schema can easily be modified by adding a block and connecting it to other blocks.

This toolbox can be used in any operation mode during the plant operation stage, both online and offline. A representation of a variable can be obtained offline (from a recorded signal), or online, generating the episodes at the characteristic instants (when episodes are finished, not at any sample time).

Qualtras was initially based on Meléndez and Colomer, 2001, but it provides greater flexibility since it allows choice from among a set of basic functions in order to detect changes between episodes. These functions even could be considered simultaneously when necessary to offer a richer representation; for example, to describe interesting behaviour of a signal with very different dynamics (e.g., spikes and slow slopes). The episodes generated by Qualtras can be used in diverse applications. An illustrative example will be presented in section 4.4. Moreover, it can be used as a pre-processing tool for other decision support tools. For instance, in Kempowsky et al., 2005 the episodes provide information to a qualitative situation assessment software tool named SALSA and some tests in a fluidised bed gasifier illustrate the approach.

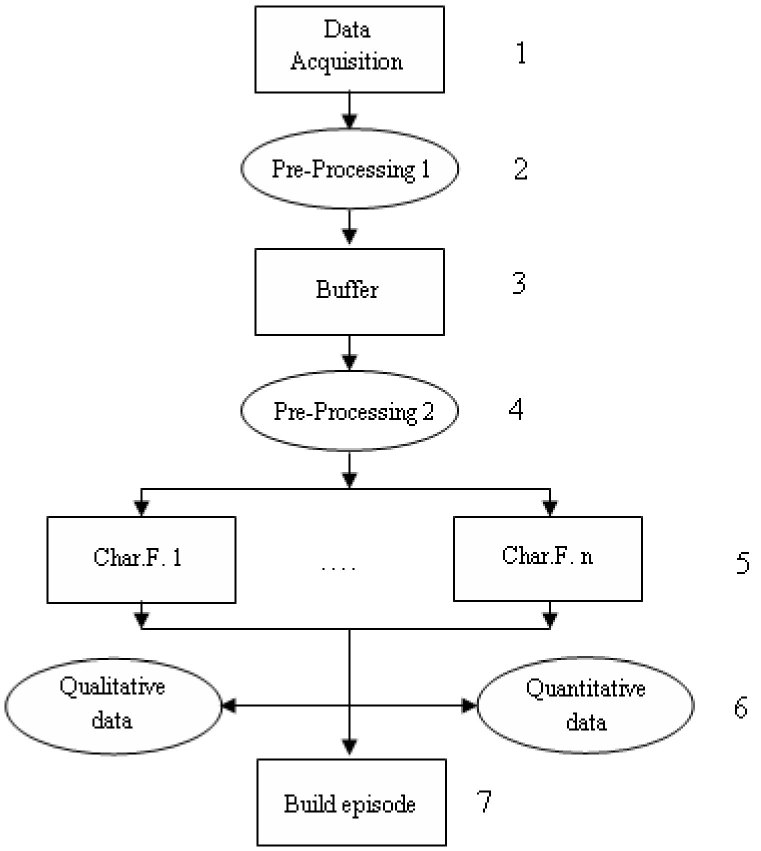


Fig. 4.2 The architecture of Qualtras.

The architecture of Qualtras is designed as several interconnected modules to create modular application configurations. Each module provides a dialogue box that enables the user to configure parameters. The processing of each variable forms a sequence that should not be less than 4 modules and up to 9. Fig. 4.2 shows a scheme of these modules. In the figure, rounded modules represent optional stages, while squared modules should be considered mandatory. Also, Pre-processing and Characteristic Function modules could contain interchangeable blocks, allowing the operator to use the most appropriate as the case may require, including a combination. The details of the tasks performed by each of the modules are discussed below:

1. Data acquisition: this module acquires online data from the plant and it also supports the function of historical data retrieval. Simulation in real time is allowed when data comes from files or a data base.

2. Pre-processing 1: this module pre-processes the measured signal. Typically any filter could be part of this module. By default Qualtras uses a Bandlimited FIR-derivative filter (Colomer and Meléndez, 2001) to compute derivatives. The window size depends on the noise in the signal. Long sliding windows produce greater filtering but also greater delays.

3. Buffering: the data buffer (adjustable size) stores internal data for the operation of the toolbox.

4. Pre-processing 2: this second pre-processing module enables operations that need an array of data as input. The benefits of some techniques are evident. The wavelet transform (Mallat and Zhong, 1992) can detect and characterise singularities by decomposing signals into different scale components. Another operation allows the use of Piecewise Aggregate Approximation (PAA) to reduce the length of waveforms without losing information. PAA (Keogh et al., 2001b) is a dimensionality reduction technique that approximates a time series in a set of M equal-length segments. The average value of the samples in each segment is used as a representative of them. Although other techniques could be implemented here, the cubic spline wavelet (Bakshi and Stephanopoulos, 1994a) and PAA were chosen initially.

These first 4 modules constitute the first step in the processing of data. The idea is simple: to acquire, filter and store process data. The following modules are:

5. Characteristic function: according to the general formalism described in Section 3.2.4, a characteristic function F (Meléndez and Colomer, 2001), of a signal, $x(t)$, is formally defined as any function of the variable at a time instant t_i, and at the previous time instants, the previous characteristic function values and the corresponding time instants (eq. [3.5]). It is used to determine the Characteristic State (eq. [3.8]) indicating that the value of the characteristic function belongs to a characteristic interval. So, the range of values of each characteristic function is divided into a set of contiguous and non-overlapped intervals.

Finally, the qualitative state (eq. [3.10]) is defined as the set of characteristic states for each characteristic function. In Qualtras the user should define the alphabet or symbols to represent each qualitative state.

The toolbox implements some characteristic functions (first derivative, second derivative, signal, linear regression) or can implement other qualitative representations as characteristic functions, like SAX (Lin et al., 2007), since this representation can be adopted as a new sequence of episodes only by converting it to asynchronous. Therefore, an adequate function has to be chosen according to requirements.

If more than one characteristic function is used, the user must define the episodes corresponding to the combination of qualitative states. The main idea is that the tool can be configured in a way that only the interesting characteristics of the signal behaviour are used to obtain its description.

For example, in some cases, there are measurements that are increasing or decreasing slightly. If the measurement keeps decreasing/increasing very slowly then it is possible that the first-derivative will not be capable of detecting it, so it is important to know if the value is still around the set point or not. In such cases it is also necessary to detect the magnitude. This is the case in the configuration in Fig. 4.3 where the input variable is filtered (FIR-12) and qualified using two characteristic functions: FC12 and FC12B. These functions return qualitative values from the first derivative and the magnitude respectively. Since two characteristic functions are used together, the tool recognises this situation and it constructs a table where combinations of the two qualitative values are summarised as a final episode type (Table 4.1). The user selects the final episode type by choosing between a set of available symbols (Fig. 4.4) and he can even build a new one. Finally, note that some quantitative data is also captured by means of a quantitative data block (QNT-DATA12).

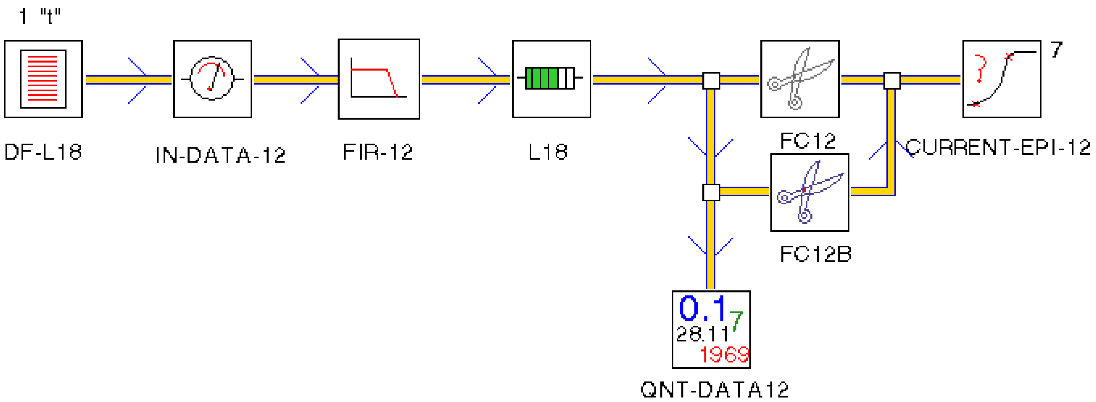


Fig. 4.3 Configuration using first derivative and magnitude for extracting episodes.

6. Qualitative and quantitative data: optionally, there is other qualitative or quantitative information that could also be interesting from the point of view of process supervision but is not significant in determining the temporal extension of episodes, i.e.: mean, slope, maximum, minimum…. Thus, this information will be embedded into the episode as auxiliary characteristics associated with this signal.

The purpose of these last two modules is to obtain the qualitative state of a process variable and the associated auxiliary characteristics. At this moment the signal has been qualified, it constitutes the primary segmentation of the variable but it is not an episode yet.

Table 4.1 Configuration table of L1.

	Combinations of episodes		
	Signal	First-derivative	Types
1	Low	Very low	31 (Low)
2	Normal	Very low	20 (s)
3	Low	Low	16 (o)
4	Normal	Low	6 (f)
5	Low	Zero	32 (Normal)
6	Normal	Zero	7 (g)
7	Low	High	17 (p)
8	Normal	High	8 (h)
9	Low	Very high	33 (High)
10	Normal	Very high	19 (r)

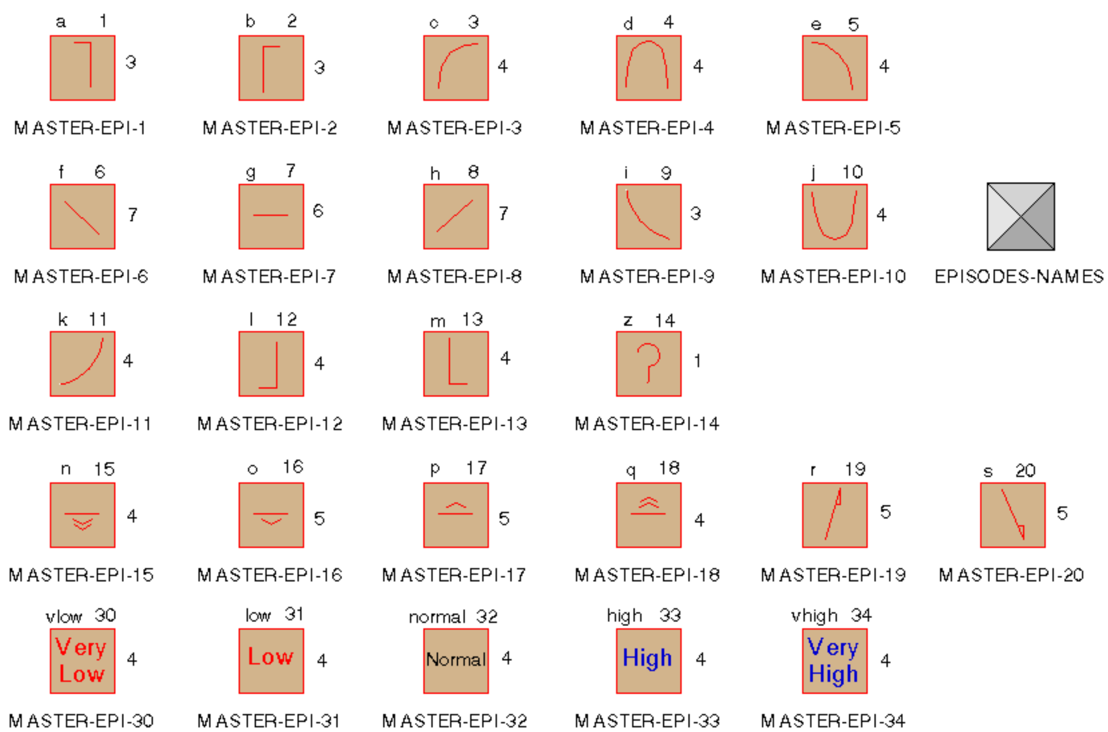


Fig. 4.4 Collection of episodes available in Qualtras.

7. Building episodes: this element contains information about the current episode (qualitative value, temporal extension and auxiliary data) and the history of the episodes. It also adds the temporal extension and timestamp to finished episodes. To know if an episode is finished, this block starts a procedure that tries to associate the last episode with the former one to form the longest possible episode. The procedure also aggregates episodes of very short longitude with the contiguous ones. These episodes usually correspond to transitory or noisy episodes. The minimum duration can be readjusted individually for each type of episode in order to be considered a true qualitative state. This operation provides high robustness regarding noise, but has the drawback of introducing a delay in communication when it works in online mode.

Finally, it visualises the episode and stores it as historical data in csv format (comma-separated values) and furthermore, this block can initialise the communication online of the resulting episodes. The successful communication of Qualtras with other tools in online mode can be achieved by structuring the output data. Qualtras uses XML architecture for communication between applications. XML (Extensible Markup Language) is a standard concerned with the description and structuring of data so that applications can share data using common vocabularies (Bierman, 2000).

4.4. Application example: fault prediction in a blast furnace

The applicability of this tool is demonstrated in a pre-fault detection approach in a blast furnace (Gamero et al., 2006). The example discusses the analysis of differential pressure signals in a blast furnace stack by using PCA (Principal Component Analysis) and episodes generated by Qualtras. The objective is to predict aerodynamic instability in a blast furnace with sufficient warning to enable the blast volume to be reduced in order to minimise that instability. The main characteristic of this process is the lack of a valid physical model in order to predict instabilities. So, the expert knowledge and signal-based methods are used to provide an efficient approach to slip prediction.

This example provides a good opportunity to show the operation of Qualtras when working separately, as well as being used as a complement for other strategies in process supervision. The approach described in this section was developed within the CHEM project and tested in a real blast furnace operated by Corus (Fig. 4.5).

The process description will be given next and afterwards some PCA concepts are briefly described before showing the test results.

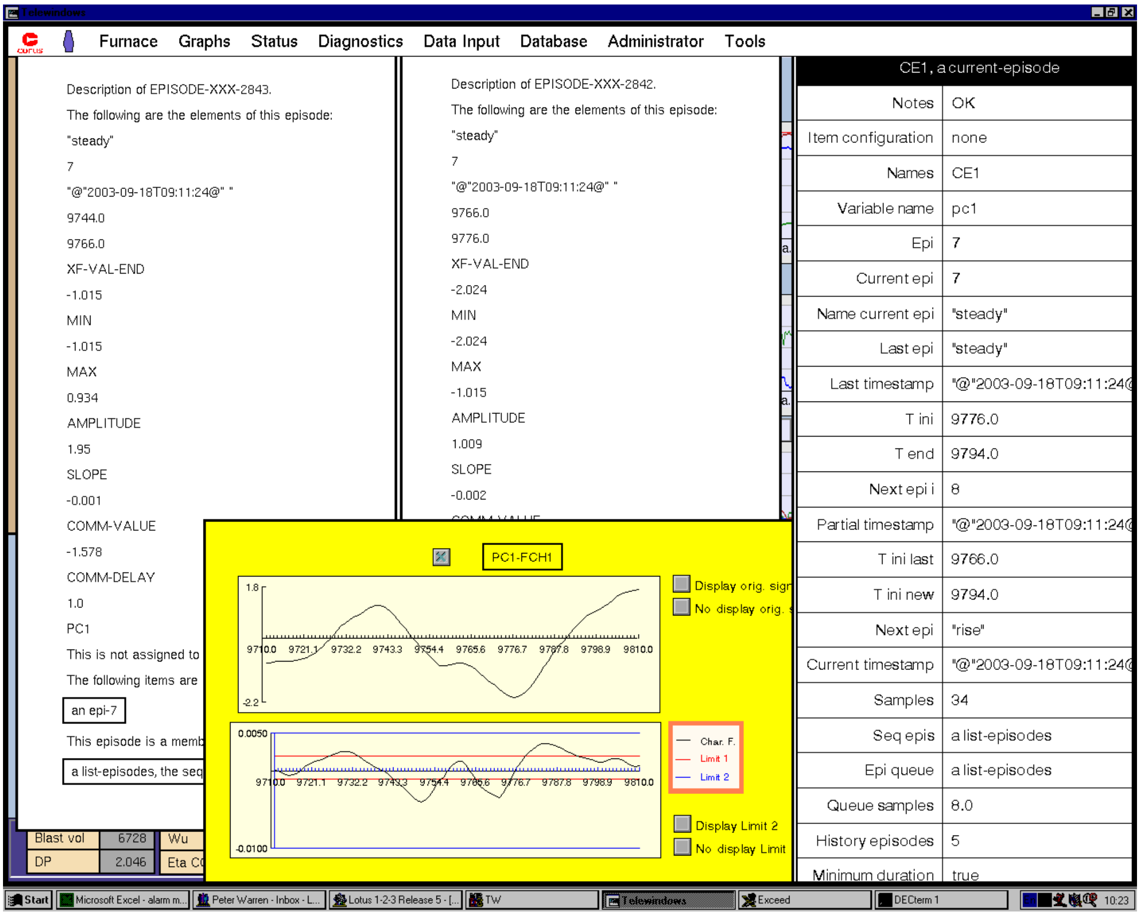


Fig. 4.5 A capture from Qualtras integrated in the monitoring application in Corus.

4.4.1. Process description

In the blast furnace process, iron oxide is reduced to metallic iron, which is then melted prior to tapping at 1500 deg C. Liquid slag is also removed, the slag being formed from the non-ferrous components of the iron ore (predominantly lime, silica, magnesia and alumina).

The blast furnace itself is a water-cooled vessel, of circular cross-section, about 30m tall. Layers of coke and prepared iron ore (burden) are alternately charged into the top in a controlled manner. Air at approximately 1100 deg C (hot blast) is blown into the bottom of the furnace through copper water-cooled tuyeres. The air reacts with the coke, and supplementary injected oil, to form a reducing gas of CO and H2. The burden takes about 7 hours to reduce and melt as it descends the furnace stack; the residence time of the ascending reduction gases is in the order of seconds. The furnace is kept full by charging a fresh batch of burden as soon as the level at the top drops below a preset level. Burden descent is usually steady at about 5 metres/hour. The liquid iron and slag collect in the furnace hearth, built from carbon bricks, and are periodically removed by opening a clay-lined taphole.

Redcar blast furnace is the largest in the UK, producing 65000 tonnes of liquid iron per week. Each hour requires the charging of 5 to 6 batches of prepared ore, each of 110 tonnes, and the same number of batches of coke, each of 22 to 23 tonnes. Heat is removed at the wall through large water-cooled panels, called staves, installed over 13 rows. These are made from copper in rows 7 to 10, the region of highest heat load, and cast iron above and below these levels.

The Redcar plant has 4 columns of pressure tappings. The pressure is measured at each stave row between row 6 and row 13 in each quadrant. Fig. 4.6 shows the furnace cross section. Under normal state, the pressure should reduce gradually between the bottom (row 6) and the top (row 11) of the section of furnace stack under examination. It has been noticed that trends in differential pressure (pressure drop) between the lower and middle part of the stack can be a good indication that the furnace process will become unstable. The differential pressures found most responsive are the pressure measured at row 6 minus the pressure measured in row 9, for each quadrant individually.

An ‘unstable’ event is when the material in the furnace stack suddenly ‘slips’ rather than descends steadily. This is often preceded by the ascending gas channelling through zones of lower resistance rather than being evenly distributed across the furnace. Such an event can be predicted by a sudden change in differential pressure values over the 4 columns of pressure tappings. In Fig. 4.7, differential pressure trends are recorded at a frequency of one sample/min. over a 10-hour period. In this figure, the occurrence of slips is marked by the dotted lines at the right of each of the two highlighted periods. It can be seen that the differential pressures reduce significantly prior to these events.

A physical model is not available to predict slips so it is important the expert knowledge that should be considered. Therefore, other methods based on signals in the data rather than models could be used.

The methods described have been tested in the blast furnace to predict slips over the 22 days where significant instability was experienced over a 2 year period.

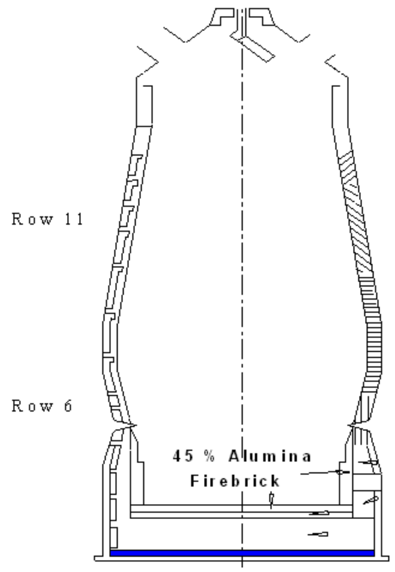


Fig. 4.6 Transverse section of blast furnace

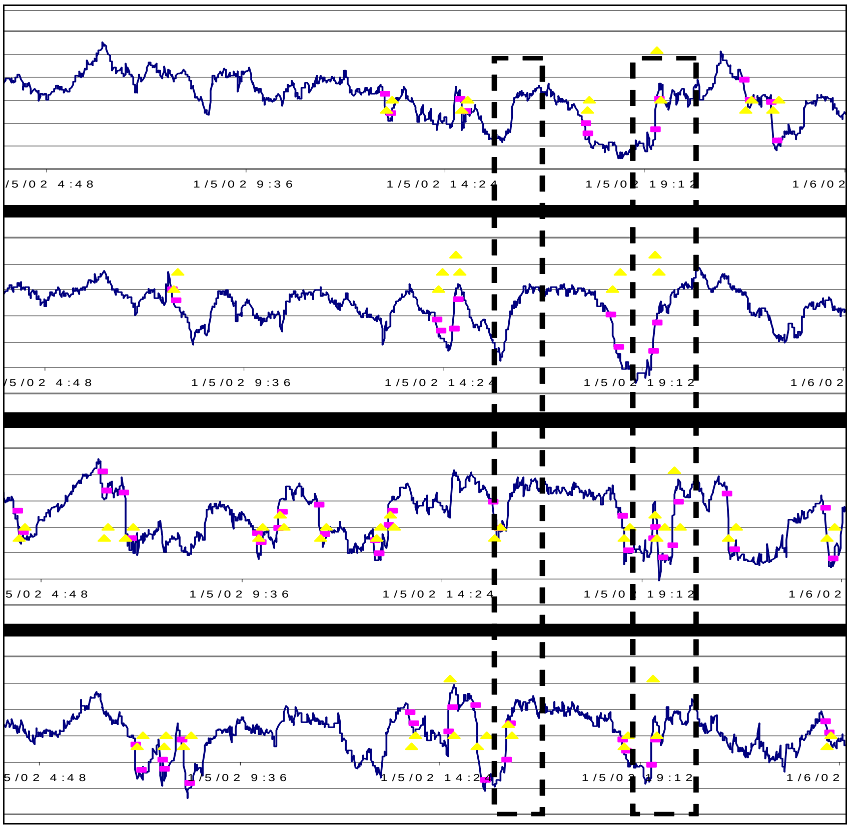


Fig. 4.7 Fault determined by pressure differentials

4.4.2. PCA

Principal Component Analysis (PCA) is a multivariate technique that transforms a number of correlated variables into a smaller number of uncorrelated variables called principal components. PCA has been extensively used for fault detection, fault diagnosis, process monitoring and sensor faults. However, in this application the emphasis is placed on predicting the fault so that action can be taken to reduce its effect. Several applications performing PCA can be found in the literature. (Yoon and MacGregor, 2004; Singhal and Seborg, 2001; Maurya et al., 2003).

Thus, the aim of PCA is to reduce the dimensionality of data without loss of useful information. This is achieved by redefining variables to generate new axes for the system in an optimum manner to maximise the variation described by each new axis. The first axis describes the greatest variance in the data, the second axis the next greatest variance in the data with subsequent axes describing the remaining variance. This generates the principal components ($PC_1,\ldots,PC_m$) which are a linear combination of the original normalised ones;

$$\begin{aligned} \mathrm{PC}_1 &= \lambda_{1,1}x_1 + \lambda_{1,2}x_2 \;\ldots\ldots\; + \lambda_{1,n}x_n \\ &\ldots \\ \mathrm{PC}_m &= \lambda_{n,1}x_1 + \lambda_{n,2}x_2 \;\ldots\ldots\; + \lambda_{n,n}x_n \end{aligned} \qquad [4.1]$$

where $m<n$, being n is the number of original variables, x the original normalised variable, λ the eigenvector which transforms the original normalised variables onto the principal component axis and PC the principal component. The generation of principal components and their use is described elsewhere (Jolliffe, 1986; Efthimiadu et al., 1995; Martin and Morris, 1996).

Since only a few principal components are usually required to describe the variation of interest within the data, it is possible to use these to formulate a MSPC chart. This in practice means that instead of monitoring all of the relevant variables, which would prove difficult and could be misleading, only a small number of principal components need monitoring to detect when a process is going out of control, even when only one of the original variables has caused the process change. It is this aspect of PCA that makes it suitable for applying to blast furnace instability detection. In practice, it has been found sufficient to monitor only the first two principal components which represent 68% of the variance in the data. An adaptive PCA algorithm was developed for on-line blast furnace application to account for normal process variations.

PCA is applied by means of the Computas iMSPC toolbox © ("intelligent Multivariate Statistical Process Control")[1], developed in G2. This toolbox contains all standard functions from MSPC theory required to implement multi-variate models. The toolbox has a comprehensive user interface to implement and modify models, and to display their outputs. The MSPC modules are embedded in a framework for on-line Performance and Quality Management (PQM) that emulates sophisticated expert reasoning on the results of the MSPC calculations. The aim is to enable early detection and prevention of performance and quality problems.

The multi-variate model is calculated using a standard desktop statistical analysis. It is important to select periods of data representing unstable periods to ensure that the model does not simply represent process 'noise'.

4.4.3. Prediction instabilities on the blast furnace

Several tests were performed in order to predict blast furnace aerodynamic instability with sufficient warning to enable the blast volume to be reduced in order to minimize that instability in the blast furnace. The effect is usually seen as one or more of a sudden slip of the burden, detected by the rapid descent rate of the level measuring rods; a temporary increase in heat loss to the furnace wall (heat flux); or gas channelling. All are caused by disruption of the normal gas flow through the burden.

Next the results obtained in three approaches are presented. The first two apply individually the two methods described previously. The third one combines PCA and QTA, so that Qualtras is applied on the principal components rather than on the differential pressure signals.

By using episodes

This approach uses differential pressure trends in order to extract episodes. The necessity of obtaining an effective qualitative representation from the first instants of the episode without waiting until the end of the episode for its identification, leads to the choice of the first derivative as used feature in order to obtain the episodes representation. This simple representation, however, offers enough information to characterize the signals of the analyzed process. Due to the use of only the first derivative at the beginning of each episode, only five basic types are obtained. However, since the previous state to a slip is often a fall in pressure followed by a

[1] Copyright © Computas AS

steady trend at low level, a differentiation has been introduced among the episodes of stability according to the value of the signal. So the type episode called G was subdivided in High (Gh), Medium (Gm) and Low (Gl) as presented in Fig. 4.8. This representation was adopted to represent symptoms of blast furnace instability by means of differential pressure trends as a first approach. An example of this representation for two variables is shown in Fig. 4.9.

The first test on the blast furnace was composed of 12 data sets representing several levels of instability (Table 4.2). This group of data sets is representative of the most important instabilities that could appear in the blast furnace with acquired experience over a 2 years period. The trend prior to a slip is given by the step from a stability episode (Gh, Gm, Gl) to a stability episode Gl by means of episodes indicating fall (A, F). The notation used for each quadrant in Table 4.2 is as follows:

S: A sequence of episodes exists with instability risk.

F: A significant falling episode exists.

Nr: The time between sequence recognition (instability prediction) and the slip in minutes.

-: There isn't a sequence suggesting risk.

Table 4.2 Results from each quadrant using episodes.

	Depth of slip	Q1	Q2	Q3	Q4
Data set 1	2.0m	S 81	S F 41	S F 34	S F 60
Data set 2	2.0m	S F 43	S 19	S 54	S F 21
Data set 3	1.5m	S F 67	-	-	S 979
Data set 4	1.5m	S F 40	S F 5	-	S F 51
Data set 5	1.5m (partial)	S F 65	S F 24	S F 27	-
Data set 6	1.5m (partial)	S 32	S F 32	S 41	S 41
Data set 7	1.5m	S F 62	-	-	S F 69
Data set 8	1.5m (partial)	-	S 38	S F 34	-
Data set 9	1.0m	S 32	S F 45	S F 12	S F 28
Data set 10	1.0m	S F 90	-	S F 34	S 31
Data set 11	No slips	-	-	-	-
Data set 12	No slips	-	S	-	-

A significant falling episode means that other quantitative values such as amplitude or slope could be considered. This property facilitates the exclusion of false alarms produced by similar sequences and validates a sequence which precedes an unstable event. Now, note from analysing the results in Table 4.2 that a valid sequence always exists in at least in one quadrant. Nevertheless, when a valid sequence occurs in only one quadrant false alarms arise. This fact was confirmed during the on-line operation where some false alarms arose but they were discarded by an expert operator. In order to significantly decrease the number of false alarms, the pre-fault stage will be indicated by a valid sequence of episodes in two or more quadrants and the time after sequence recognition it must be kept in mind. That is, a validity interval should be defined for significant sequences. A value of approximately 60 minutes is enough defined by an expert operator. Consequently, some slips will not then be predicted especially partial slips or small slips.

The results obtained are encouraging and validate the episode based approach, although they suggest the need to use more process signals in order to discriminate false alarms and detect small slips. To this end, PCA allows an increase in the number of signals considered whilst reducing the dimensionality of data.

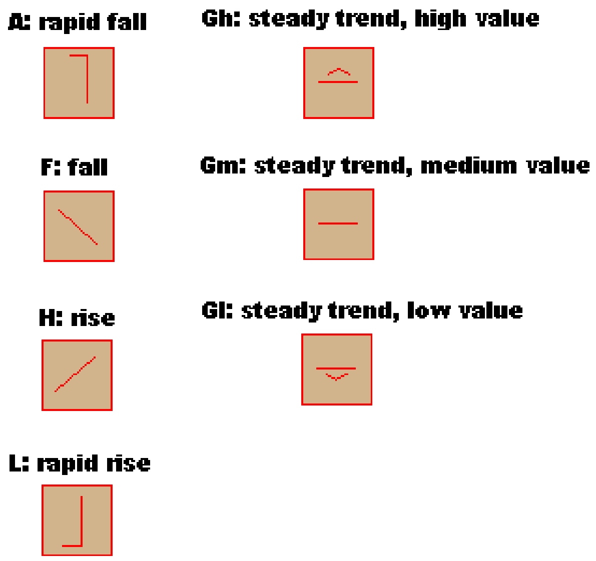
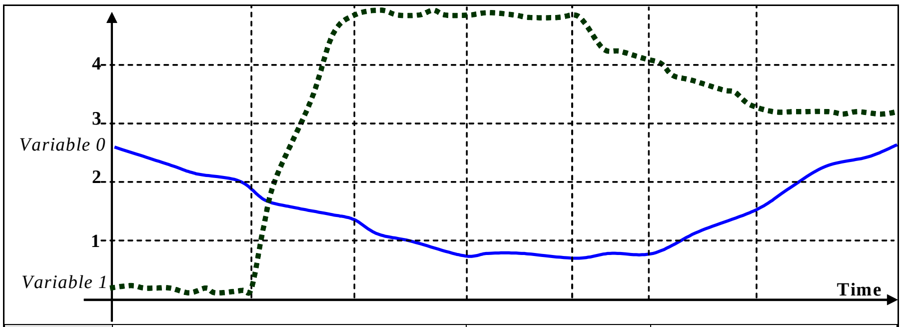


Fig. 4.8 Set of episodes used in first approach



Variable 0	**F-1-10-0,8**										**Gl-11-5-0,8**					**H-16-7-2,7**						
Variable 1	**Gl-1-4-0,1**				**L-5-3-5,0**			**Gh-8-6-4,9**						**F-14-5-3,2**					**Gm-19-4-3,1**			
Time-line	**1**	2	**3**	4	**5**	6	7	**8**	9	10	**11**	12	13	**14**	15	**16**	17	18	**19**	20	21	22

Fig. 4.9 Episode representation of process variables

By using PCA

A seven variable PCA model was selected from the three different models developed at Corus. The model is run every minute. It uses differential pressures in the furnace stack and off gas analysis. A number of contribution analysis schemes have been developed, based on the first two principal components, to identify which parameters have undergone the greatest change when bi-variate plots exceed warning and action limits for a set number of samples.

This model has been run on line at Redcar Blast Furnace with messages being sent to the test queue.

It was found that when the action limit was exceeded for several minutes, and at least one differential pressure was identified as having changed significantly, process stability deteriorated. When only the warning limit was exceeded, many false alarms were generated.

The technique is also used for data compression: A stability index is calculated as the square root of $PC1^2 + PC2^2$, then normalized on a scale of 0 to 10. This is trended on the user interface as a summary indicator of process stability; the higher the value the more stable the process.

A summary of results comparing PCA model and the approach described in next section is shown in Table 4.3. This table indicates that only 8/19 major events were predicted, but three of these ones were predicted exclusively by iMSPC.

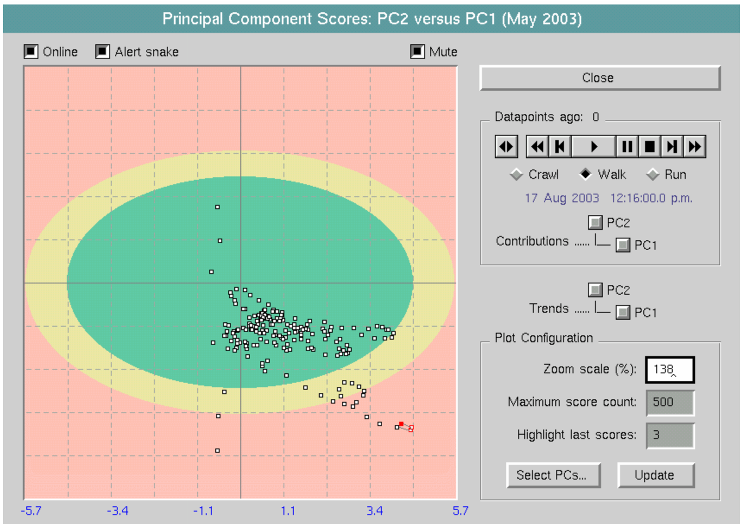


Fig. 4.10 Bi-variate score plot. Example event detected by iMSPC exceeding action limits.

By using PCA and episodes

Many events were missed using only iMSPC. So a technique was developed where two of the principal components generated by iMSPC are fed to Qualtras (Fig. 4.11) for the generation of episodes. It is now necessary to adapt Qualtras to the new trends, so a greater segmentation between episodes is introduced basing on their different range of values. Fig. 4.12 shows the new set of episodes using PCA.

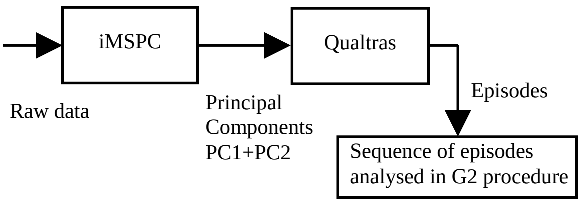


Fig. 4.11 Symbolic representation of the hybrid approach.

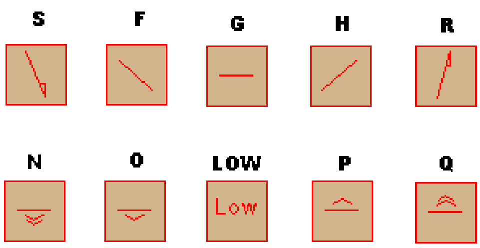


Fig. 4.12 Set of episodes used in third approach

The episodes generated were analysed and a common pattern was identified as pre-faulty state based on the sequence of episodes, their slopes and amplitudes. These were converted to rules and have been programmed into a G2 procedure, which is called whenever the episode type is updated (Fig. 4.13). Thus a current episode of type 'low' or 'O' in PC2 initiates rule analysis which looks for any valid sequence using the previous three episodes. For each previous episode only certain types are permitted but the initial type should be an episode indicating falling ('F', 'O', 'S'). In this way, according to the valid types in Fig. 4.13 a valid pre-fault sequence could be formed from 2 to 4 episodes (for example 'S-LOW' and 'F-S-G-LOW'). Also, some quantitative characteristics in PC1 as the amplitude of all episodes and the slope of initial episode are considered in order to validate the rule. This has runs on line.

Further analysis has improved on this rule by including as additional condition that the heat flux must be above a threshold or rising to enable the alarm. This filtered the false alarms generated from the off line and on line testing.

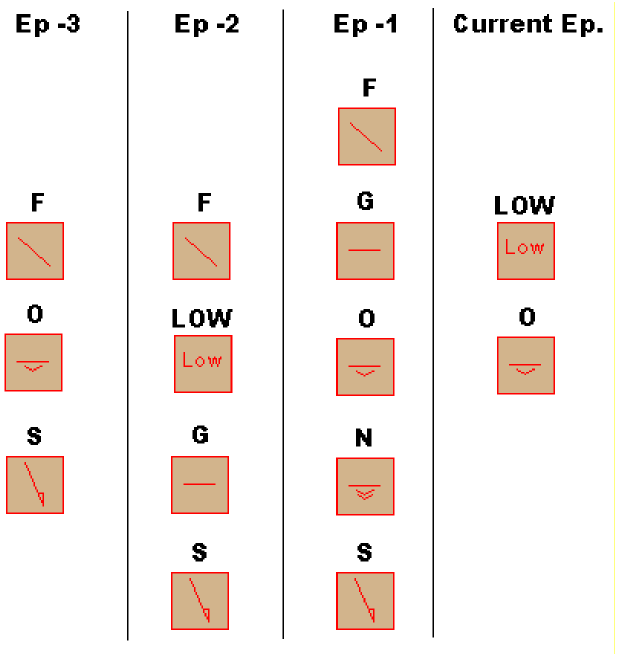


Fig. 4.13 Valid sequences in pre-fault mode

Results

Table 4.3 summarizes testing of the above combinations of toolboxes on 22 days where some instability was experienced over the 2 years. An event is classified as having been predicted if one toolbox, or combination of toolboxes, predicted it more than 10 minutes before it occurred. Event types are defined as major event (slip >= 1m and/or excessive heat flux) and minor event (smaller slip and/or significant rise in heat flux).

Of the nineteen major events, fifteen were predicted using the combination of toolboxes (episodes generated from PC1 and PC2). The average warning was 38 minutes with a standard deviation of 15 and range of 13 to 65 minutes. Of these fifteen, five were also predicted using iMSPC alone. Considering both toolboxes only one major event was not predicted.

In the other way only 4/10 of the minor events were predicted. However, it is unlikely that action would have been taken in most cases of minor events, so a missed minor event is not seen as important.

Sometimes alarms were also generated during an event (high heat flux) but are still relevant. It is proposed that the advice will be released to operators following testing with the additional heat flux condition to ensure that false alarms are not generated.

Resuming the results obtained by this approach, almost the 95% of the major events are predicted in front of the 57% obtained with Qualtras alone or even worst results using iMSPC. However, combination of both methods demonstrates a higher accuracy in predictions with fewer false alarms.

Table 4.3 Analysis of off line predictions made by iMSPC and Qualtras.

Event type	Major	Minor
Number of events	19	10
Predicted by iMSPC only	8	1
Pred. by Qualtras on PC1/PC2	15	4
Not predicted	1	6

4.5. Conclusions

The interpretation of time series in terms which a human interpreter arrives at almost unconsciously is very useful when one reasons about processes. Qualtras facilitates the abstraction of the most significant characteristics of the signals by representing any process signal by means of episodes. That is, it provides an intuitive representation as well as reducing the complexity of the system state by allowing only a finite set of qualitative descriptions. Each episode is composed of a time interval, a symbol describing its behaviour and a set of quantitative and/or qualitative data with additional information.

The implemented set of basic functions to detect changes between episodes requires the specific tuning of parameters and thresholds to achieve desired performance levels. These requirements require a priori knowledge of the process. However, the open architecture of Qualtras allows the addition of functions designed specifically for a particular process or new techniques without, or almost without, settings. Then, if only useful characteristics are used to construct episodes, the representation obtained will be, at the same time, the most simplified and the most significant from the supervisory system point of view.

The output of the tool could be used for monitoring or could be supplied to other fault detection and diagnosis tools, expert systems or classification methods. The illustrative example has shown how our qualitative trend representation tool allows it to complement other applications and improve diagnosis strategies. Individually, each one of the techniques is shown to be useful in developing a decision support system applicable to an industrial process. However, by combining both techniques the results are better since the combined approach was able to predict almost all of the major unstable events. So, Qualtras has been validated in the extraction of qualitative representations for the different approaches developed in this work.

Moreover, the study of process dynamics implies dealing with the evolution of variables, and the dynamics of process variables can be representative of specific states. In these cases, the episode-based representation of a process trend can be treated like proven cases, considering the pattern of several process variables instead of certain episodes as was seen in the rule-based ES of the example. Thus, the task of diagnosis can be viewed as a classification problem or a pattern recognition task.

Classification through the comparison and matching of temporal sequences is an active area of research in the study of time series. Also, signal comparison can be affected by several problems as differences in the length (total time) of the two signals, or in the magnitudes, or time misalignments. Consequently, some issues related to the similarity of time sequences are enumerated in the next chapter.

Chapter 5.

Time Series Similarity

5.1. Introduction

Process measurements are the fundamental element for carrying out supervisory systems tasks, namely, fault detections and diagnosis. A suitable and organised way to register process measurements is a time series or time sequence. Then, it is possible to associate some of these time series with certain process states or symptoms, that is, to categorise time series into a finite number of classes extracting particular features that distinguish one signal from the others. Later, symptoms can be classified by an experience-based diagnosis.

A good diagnostic method should be able to detect these symptoms and compare them to the normal state and flag them as abnormal. Unfortunately, time series inherently contain inaccurate information since measurements are obtained and transferred with imperfect instruments. Using this data without any pre-processing may affect the execution of supervision.

The pervasiveness and importance of time series data has motivated a lot of research focused on problems such as clustering, similarity search, classification and prediction, etc., (Fu, 2011; Ding et al., 2008; Nong, 2003; Moller-Levet et al., 2003b). The problem of comparing patterns in an efficient and precise way in series of data is an important problem in a wide variety of applications. Multiple approaches have been developed from such different perspectives, including by merging techniques, which makes it difficult to classify them in a linear order.

Although in this thesis, the objective is to deal with qualitative representations based on episodes, the aim of this chapter is to present some approaches proposing a distance/similarity that allows the comparison between series, with special attention to those algorithms that will form the basis of the contributions presented in the next chapter.

In addition, due to their affinity with a strictly qualitative representation, many methods for performing similarity retrieval in the domain of strings over finite alphabets are addressed.

This review does not include techniques from the field of machine learning such as hidden Markov models or neural networks. Some work related to these techniques can be found in Wong et al., 2001and Srinivasan et al., 2005 respectively. Also not included are techniques from multivariate statistics like the principal components (PCA) of the sequences. The reader can find an example of the PCA similarity factor proposed in Krzanowski, 1979 and modified in Singhal and Seborg, 2002a, Singhal and Seborg, 2002b and Singhal and Seborg, 2006.

The chapter has been organised as follows: section 5.2 presents the terminology and problems related to the comparison of time sequences used in the literature. Regarding similarity approaches, it is common to distinguish between those based on signature, those that propose a data reduction and works based on the temporal alignment. Section 5.3 provides an introduction to these methods. Section 5.4 is dedicated to DTW, the most used algorithm that deals with temporal alignment. Later in Section 5.5 other related algorithms are presented. In Section 5.6, the most widely used algorithms for pairwise sequence alignment are summarised. Section 5.7 will be concerned with some methods for pattern recognition based on qualitative trends. Finally, some concluding remarks are given.

5.2. Terminology and similarity measures

This section contains the definitions for some commonly used keywords in similarity-based retrieval of time series. The central concept of “time sequence” will be illustrated and structural similarity presented.

5.2.1. The time series representation

A time series $\boldsymbol{x}$ (or time sequence) of length m is often defined as an ordered collection of values of a variable taken in increasing periods of time. The instants of time at which the measurements are taken are known as time points, and the length between time points is called the sampling interval. This length can vary if the sequence was sampled at different timestamps (eq. [5.1]), or be regular if the sampling interval is constant (eq. [5.2]). Formally:

$$\boldsymbol{x} = [(t_1,x_1),(t_2,x_2),\ldots,(t_i,x_i),\ldots,(t_m,x_m)] \qquad [5.1]$$

$$\boldsymbol{x} = [x_1,x_2,\ldots, x_i,\ldots,x_m] \qquad [5.2]$$

In this thesis the time series will always be referenced as the one shown in eq. [5.2].

The variable comes from a variety of different domains, from engineering (Notohardjono and Ermer, 1986) to scientific research (Tilman and Wedin, 1991), finance (Boschen and Weise, 2003) and medicine (Guo et al., 2003; Yum and Kim, 2003).

Antunes and Oliveira, 2001 distinguish four main groups for time series representation: time-domain continuous, transformation-based, discretisation-based and generative models.

In time-domain continuous representation the simplest approach is to represent a time series using the original elements, ordered by their instant of occurrence without any pre-processing (e.g., $\boldsymbol{y} = [y_1, y_2\ldots y_n]$, where each y_i represents a value at a point in time). Other alternatives are transformations related to the length of the series. If the series are too long to be manageable, they can be shortened using, for example, piecewise linear functions. There are several approaches to segmenting the time series in order to reduce dimensionality.

In transformation-based representations the idea is to transform the initial sequence from time to another domain, and then use a point in the new domain to represent each original series. For example, the use of Discrete Fourier Transformation (DFT) to transform the sequence in a point in the frequency domain choosing the k first frequencies and then representing each sequence as a point in the k-dimensional space (Agrawal et al., 1993). Other examples of transformations could be the Discrete Wavelet Transformation (Popivanov and Miller, 2002) and the Single Value Decomposition (Korn et al., 1997).

Discretisation-based representations translate the initial time series with real elements to a discretised sequence. An example of this translation in the Transitional State Discrimination (TSD) approach (Moller-Levet et al., 2003a), where the time series are discretised according to the difference in values between successive time points.

In generative model representation the idea is to obtain a model that can be viewed as a generator for the time series obtained. For example, Hidden Markov Models (HMM) (Ji et al., 2003) or Auto Regressive models (AR) (Ramoni et al., 2002).

This dissertation deals with time-domain representations in order to discretise them and represent time series as sequences of qualitative or semi-qualitative symbols.

5.2.2. Similarity queries and indexing

After representing the time series appropriately, a similarity measure is needed as a measure of their likeness. Most of the work done so far in time series similarity comes from the problem of similarity queries in the field of temporal databases. Similarity queries can be classified into two categories: whole sequence matching (Agrawal et al., 1993; Shatkay and Zdonic, 1996; Bozkaya et al., 1997) or subsequence matching (Faloutsos et al., 1994; Moon et al., 2002; {Siu, 2003 #189}). Whole matching refers to the comparison of two complete sequences, while subsequence matching, as its name implies, is the comparison of a small sequence with small sequences parts in a complete sequence.

Most existing methods for evaluating similarity queries exhaustively compare the query sequence with each sequence in the database. Thus, for large databases, exhaustive search techniques become too expensive due to the volume of data to be processed for each query. To solve this problem many existing indexing methods are used to process queries efficiently (Chakrabarti and Mehrotra, 1999; Gaede and Günther, 1988). However, certain properties of time sequences make the standard methods unsuitable and experience has shown that the performance of spatial access methods degrades considerably for dimensionalities at about 15. In this sense, when the count of features is great, the meaningful analysis of data is almost impossible. This is known as the dimensionality curse (Bellman, 1961).

The general solution to this problem is dimensionality reduction (Carreira-Perpiñán, 2001; Keogh and Pazzani, 2000b), to extract the signature of low dimensionality from the original sequences, in a manner which, to some extent, preserves the distances between them, and then perform the indexing and searching in the signature space. Thus in a certain number of occasions it can be useful or even necessary to first reduce the dimensionality of the data to a manageable size, keeping as much of the original information as possible, and then feed the reduced-dimension data into the system (Fig. 5.1). Dimensionality reduction can also be seen as a feature extraction or coding procedure, or in general, as a representation in a different coordinate system in which most of the data variance is preserved in a few dimensions. The distance between the new series representations and the original series must be preserved. An index with the subset of values extracted from the original data is built. This index provides an efficient comparison of time series.

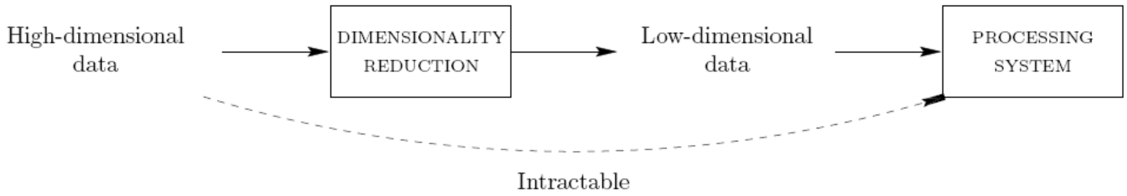


Fig. 5.1 The dimensionality reduction problem shown as a pre-processing stage.

Although the importance of using indexing methods has been explained above, dimensionality reduction may in itself speed up the sequential scan, and some methods rely only on this, without using any index structure.

5.2.3. Distance or similarity measures

The most common usage of similarity measures refers to distances in metric space with a global nonnegative distance function. Given a data space $\mathcal{D}$ defined on a time series and any two data $\boldsymbol{x}$, $\boldsymbol{y} \in \mathcal{D}$, a distance function on $\mathcal{D}$ is defined as *dist*: $\mathcal{D} \times \mathcal{D} \rightarrow \mathcal{R}^+$. Where $\mathcal{R}^+$ is the positive real data.

This distance or dissimilarity includes the similarity between two objects, because an interpretation of the result of the function is not yet specified. These two concepts of similarity and dissimilarity measures are antithetical and can be easily transformed. For example,

$$sim(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{dist(\boldsymbol{x}, \boldsymbol{y})}{dist_{\max}} \qquad [5.3]$$

Where $dist_{\max} = dist(\boldsymbol{x},\boldsymbol{y})$ if the objects are most dissimilar.

Typical interpretations of similarity measures are normalised similarity ($sim(\boldsymbol{x},\boldsymbol{y})$) or dissimilarity measures ($dist(\boldsymbol{x},\boldsymbol{y})$). For example, the value of these metrics can be between 0 and 1, where $sim(\boldsymbol{x}, \boldsymbol{y}) = 0$ if the objects are least similar and $sim(\boldsymbol{x}, \boldsymbol{y}) = 1$ if the objects are most similar or identical. In the same sense *dist(**x**, **y**)* = 0 if the objects are most similar and *dist(**x**, **y**)* = 1 if the objects are most dissimilar.

For two time series ***x*** and ***y*** of length *n*, some examples of frequent distances are:

***Discrete metric*:**

$$dist(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{n} \delta(x_i, y_i) \quad | \quad \delta(x_i, y_i) = \begin{cases} 0 & if \quad x_i = y_i \\ 1 & otherwise \end{cases} \qquad [5.4]$$

Minkowski or* L*p* *distance: many similarity measures are based on the L*p* distance, also often called the Minkowski distance.

$$dist(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{n} \left| x_i - y_i \right|^{P} \right)^{1/P} \qquad p \geq 1 \qquad [5.5]$$

Manhattan distance: for special cases, if *p* = 1 this yields the L_1, Manhattan or city block distance.

$$dist(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{n} \left| x_i - y_i \right| \qquad [5.6]$$

Euclidean distance: for *p* = 2, *dist* is identical to the Euclidean metric L_2. Most of the research work adopts Euclidean distance as the metric for sequence similarity.

$$dist(\boldsymbol{x},\boldsymbol{y}) = \sqrt{\sum_{i=1}^{n} \; x_i - y_i \;^{2}} \qquad [5.7]$$

Chebyshev distance: is the maximum distance in any dimension, and the upper bound for Minkowski distances of growing order *p*.

$$dist(\boldsymbol{x},\boldsymbol{y}) = \max_{i=1}^{n} \left(\left| x_i - y_i \right| \right) \qquad [5.8]$$

Hamming distance: is the number of corresponding bit positions or letters that differ.

$$dist(\boldsymbol{x},\boldsymbol{y}) = i \quad | \quad 1 < i < n,\ \ x_i \neq y_i \qquad [5.9]$$

The distance function directly affects the matching quality of the retrieved results, such as the accuracy of classification and clustering. This function is application and data dependent, and needs to be carefully designed to meet application requirements.

5.2.4. Motivation for non-metric distance functions

Mathematically, a distance ordinarily indicates a function which satisfies the metric axioms:

- Zero property: *dist*(*x*,*y*) = 0 if and only if *x* = *y*
- Nonnegative property: *dist*(*x*,*y*)>=0 for all *x* and *y*
- Symmetry: *dist*(*x*,*y*) = *dist*(*y*,*x*) for all *x* and *y*
- Triangle inequality: *dist*(*x*,*y*) <= *dist*(*x*,*z*) + dist(*z*,*y*) for all *x*,*y*,*z*

It is generally desirable to use a distance function which satisfies the axioms above, since several of the proposed indexing or search techniques require the triangular inequality to hold. However, even assuming that a metric is used, Weber et al., 1998 has shown that the performance of any indexing scheme degrades to that of a sequential scan when there are more than a few dimensions. Otherwise, when the distance is not a metric or that the number of dimensions is too large, bounding techniques are used.

Moreover, distance functions that are robust to extremely noisy data will typically violate the triangular inequality. Some of these functions achieve this by not considering the most dissimilar parts of the objects. So, they represent an accurate model of the human perception, since when comparing any kind of data we mostly focus on the portions that are similar and we are willing to pay less attention to regions of great dissimilarity. In fact, the human appreciation of similarity does not appear to satisfy the triangle inequality, a result shown by Tversky and Gati, 1982.

Thus, Vlachos et al., 2004 identifies some desirable issues which a distance function should address:

Different Sampling Rates or different speeds. Time series are not guaranteed to be the outcome of sampling at fixed time intervals, for example, the sensors collecting the data may fail for some period of time, leading to inconsistent sampling rates. Also, two time series moving in exactly the same way could do it at different speeds.

Outliers. Noise might be introduced due to an anomaly in the sensor collecting the data or can be attributed to human 'failure'.

Different lengths. Normally metric functions deal with time series of equal length, but often their lengths are different.

Efficiency. The similarity model has to be sufficiently complex to express the user's notion of similarity, yet simple enough to allow efficient computation of the similarity.

There are numerous non-metric measures for distance or similarity between time series, such as, for example, Dynamic Time Warping or Longest Common Subsequence described below. In such cases, there is empirical evidence and rational use of these functions representing the similarity between two sequences.

5.3. Similarity methods

Among the measures for similarity of time series data reported in the literature, it is common to distinguish those based on signature, those that propose a data reduction and works based on the temporal alignment. This section reviews these issues.

5.3.1. Methods based on signature extraction.

In many cases, calculating the distance between two objects can be costly due to the dimensionality curse problem. To avoid this problem a variety of methods where the time-series are represented as points in a low dimensional feature space have been presented. The idea is to extract a new object, a signature, much smaller in size than original series but representing their fundamental characteristics. So the search may be speeded up considerably by the use of signatures, where the signatures have the following properties:

- Calculating the distance *distS* between two signatures is less costly than calculating the distance *dist* between two original objects (sequences).
- The distance between two signatures never overestimates the true distance between the original objects. This means that, in the best case, the distance function *distS* provide the same value than *dist*, and in other cases the distance *distS* will be a lower bound between the original series.

The main advantage of signature extraction, however, is not this speedup, since the query time complexity is still linear in the database size. The real power of this approach comes with the ability to index the signatures.

Many signature types have been proposed for similarity based retrieval of time series. One of the older techniques of dimensionality reduction is transform the series from the time domain to frequency domain by means of a transform function, based on the Euclidean distance preservation stated in the Parseval's theorem and the results of (Oppenheim and Schafer, 1975). The indexation of the first coefficients of the Discrete Fourier Transform (DFT) was the method presented in (Agrawal et al., 1993) and (Rafiei and Mendelzon, 1997). They use the Euclidean distance to measure the similarity of the time-series represented by the first few coefficients of their Fourier transformation. The index was constructed with a R*-tree, (Beckmann et al., 1990). Some works extend this technique to subsequence matching as (Faloutsos et al., 1994). Another transformation from the Discrete Wavelet Transform (DWT) family, the Haar transform, was proposed in (Chan and Fu, 1999) instead of DFT. There is no advantage of this approach over DFT as was established in (Wu et al., 2000). These, and other signature types, are discussed in more detail in Hetland, 2001.

5.3.2. Methods based on data reduction

Another large group of work is characterised by the fact that it does not handle all the original data of each series collectively. Instead, the dimension is reduced by selecting a subset of the original data or by groups. Some of them apply transformations again but on the subset of the resulting clusters.

Keogh and Pazzani, 1998 introduced a time series representation consisting of piecewise linear segments to represent shapes and a weight vector that contains the relative importance of each individual linear segment. The total weight associated with a sequence of a given length is constant, regardless of how many segments are used to represent it. In this paper, for a time series A, sampled at k points, its segmented version containing K linear segments, is denoted as $\mathbf{A}$, where $\mathbf{A}$ is a 5-tuple of vectors of length K.

$$\mathbf{A} \equiv \{AXL, AXR, AYL, AYR, AW\}$$

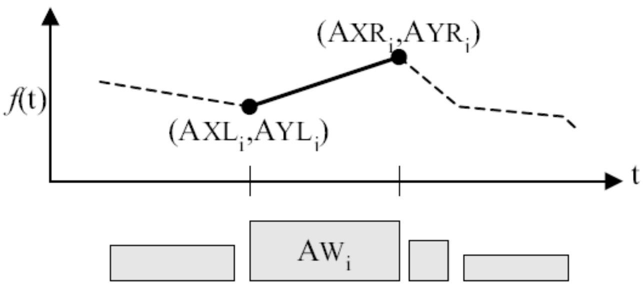


Fig. 5.2 A time series represented by a sequence of straight segments.

The i^{th} segment of sequence A is represented by the line between (A_{XLi} A_{YLi}) and (A_{XRi} A_{YRi}), and A_{Wi}, which represents the segment weight. Fig. 5.2 illustrates this notation. These vectors are used to calculate a new distance measure defined as:

$$dist(A,B) = \sum_{i=1}^{k} A_{Wi} \cdot B_{Wi} \cdot \left| (A_{YLi} - B_{YLi}) - (A_{YRi} - B_{YRi}) \right| \qquad [5.10]$$

An approach independently introduced by Keogh and Pazzani, 2000b and Yi and Faloutsos, 2000 uses an approximation to the original series using *k* segments of equal length (PAA) and uses the average value of each segment as a coordinate of a *k*-dimensional signature vector. This approach allows the building of an index in linear time. While Keogh and Pazzani, 2000b uses the Euclidean distance as the basis for the new distance defined over the index space, Yi and Faloutsos, 2000 shows that this signature can be used with arbitrary Lp distances.

Later, Keogh et al., 2001a proposed an improved version using Adaptive Piecewise Constant Approximation (APCA). This is similar to the PAA, except that the segments need not be of equal length. Two distance measures are developed for the APCA, one which is guaranteed to underestimate the Euclidean distance, and one which can be calculated more efficiently, but which may generate some false dismissals. It is also shown that this technique can handle arbitrary L_p norms. The empirical data suggests that the APCA outperforms both methods based on the discrete Fourier transform, and methods based on the discrete wavelet transform with a speedup of one to two orders of magnitude.

Perng et al., 2000 proposed the Landmark Model for pattern querying in time series databases instead of working directly with raw data. In its most general form, the model allows any point of great importance to be identified as a landmark. The specific form used in the paper defines an n^{th} order landmark of a one-dimensional function to be a point where the function's n^{th} derivative is zero. Thus, first-order landmarks are extrema, second-order landmarks are inflection points, and so forth. Using increasing types of landmarks will produce a more accurate representation. But fewer landmarks result in a smaller index tree. The most suitable selection of landmarks and the number of them used are domain-driven by the data. Since landmarks are sequential, it can reduce an indexing problem into a string-indexing problem. This model does not rely on Euclidean distance; rather it identifies features of landmarks that are invariant under several transformations. While the work does allow a flexible query language, the feature-extracting step requires the careful choice of several parameters.

Megalooikonomou et al., 2005 proposes a new method for representing time series data, the Multiresolution Vector Quantized (MVQ) approximation, along with a new distance function. The method partitions each time series into equi-length segments and represents each segment with the most similar key subsequence from a codebook. The codebook is generated earlier during a training phase. By counting the appearance frequency of each codeword in each time series a new representation is obtained. The new representation of a time series is a vector showing the appearance frequency of every codeword. Then they choose the Histogram Model as the distance measure.

Another technique for reducing dimension is introduced in Fu et al., 2008. The time series pattern matching is based on Perceptually Important Point (PIP) identification and applied to the technical analysis of stock data in a financial domain, where frequently used stock patterns are typically characterised by a few salient points. These points are perceptually important in the human identification process and should also be taken into accounts in the pattern matching process. The proposed scheme follows this idea by locating those PIPs in the time series P in accordance with the query pattern Q.

5.3.3. Methods based on temporal alignment

The aim of sequence comparison is to quantify the degree of similarity or, equivalently, the distance between the sequences. Time series are often affected by problems (Keogh, 1997) when comparing them: noise, offset translation, amplitude or longitude scaling and time misalignments. Some of these problems can be removed by some form of transformation. However, the time misalignment problem needs to be addressed in a different way.

Time misalignment is the unmatching of two time series due to a distortion (expansion or compression) in the time axis of one or both time series. For example, Fig. 5.3 depicts two similar time series with the same mean and variance; note that while the sequences have an overall similar shape, they are not aligned in the time axis. The intuitive feature alignment is also depicted. Thus, because of time misalignment, the direct comparison of two signals would be incorrect, since there is no guarantee that the corresponding segments of the signals are being compared.

Moreover, sequence alignment is a widely used method to deal with sequence similarity. So, an alignment may be constructed as an intermediate step or as a goal in itself. An alignment of two sequences can be measured by defining an alignment cost function. Typically the cost function determines the cost or penalty for mismatched, deleted and inserted sequence elements in the alignment. An alignment with the minimal cost with respect to a given cost function is the optimal alignment. Determination of optimal and near-optimal alignments of sequences is an important research tool as sequences with low alignment costs have been shown to be frequently related.

As our contributions are focused on similarity-based temporal alignments, an in-depth analysis of related methods will be discussed below. Sections 5.4 and 5.5 address those techniques dealing with temporal alignments, which establish correspondences both in time and in space between multiple sequences. Namely, for each value x at time t in sequence X, find its corresponding value y at time t' in the other sequence, where $t'=t\pm w$ and w is the temporal displacement.

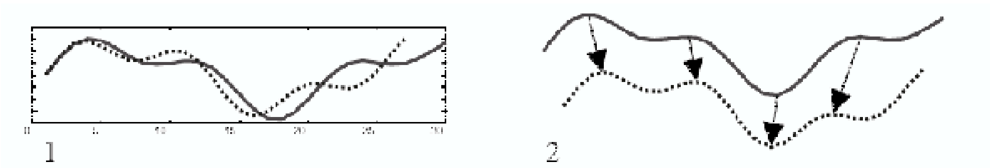


Fig. 5.3 a) Two similar time series with the same mean and variance, b) the intuitive feature alignment.

5.4. Dynamic Time Warping (DTW)

Most of the algorithms that operate with data time series use Euclidean distance or some variation of it. However, these distances are very sensitive to small distortions in the time axis (time misalignment). A method that tries to solve this inconvenience is Dynamic Time Warping (DTW) (Sankoff and Kruskal, 1983), which uses dynamic programming (Bellman and Stuart, 1962; Sakoe and Chiba, 1978; Silverman and Morgan, 1990) to align time series with a given template so that the total distance measure is minimised.

This old technique is the most used among elastic measures and the main reference for new similarity methods based on temporal alignment. The original algorithm forms the basis for the contributions developed in Chapter 6. Thus, this section is dedicated exclusively to this technique.

DTW finds the optimal alignment between two time series in such a way that they may be warped non-linearly by stretching or shrinking them along their time axis. This warping between two time series can then be used to find similar regions between the two time series or to determine the similarity between two time series. Distance is calculated as an accumulation of local distances between corresponding elements in the time series being compared. DTW has been widely used in word recognition to compensate for temporal distortions related to different speeds of speech.

An example of how one time series is warped to another is shown in Fig. 5.4. Note that while the sequences have an overall similar shape, they are not aligned in the time axis. Euclidean distance, which assumes the i^{th} point on one sequence is aligned with i^{th} point on the other (a), will produce a pessimistic dissimilarity measure. The nonlinear alignment (b) provided by DTW allows a more sophisticated distance measure to be calculated. The superiority of DTW over the Euclidean distance metric has been demonstrated by many authors (Aach and Church, 2001; Bar-Joseph et al., 2002).

The complete problem formulation of DTW is illustrated in Appendix A.

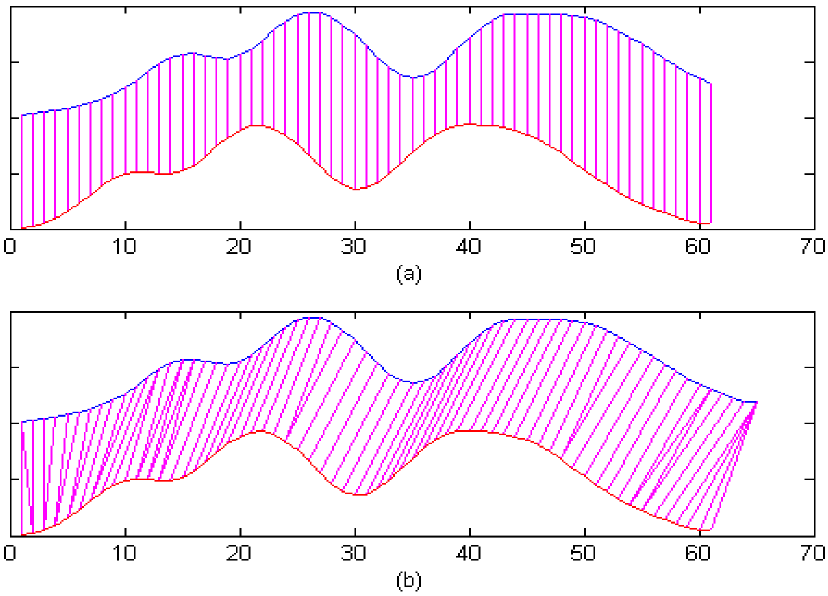


Fig. 5.4 Two different alignments obtained using (a) Euclidean distance and (b) DTW. For clarity, the upper sequence has been shifted upwards appropriately.

5.4.1. Literature review on DTW applications

In addition to speech recognition, dynamic time warping has also been found useful in many other disciplines, including data mining, gesture recognition, anomaly detection, robotics, manufacturing and medicine. Variations of the original algorithm have been proposed to cope with specific problems involving comparison of time sequences with distortion in the time domain.

Nomikos and MacGregor, 1994 reported its use for batch process monitoring. Li et al., 2004 combined DTW with wavelet decomposition for synchronising batch trajectories. The original signals were decomposed into approximations and details at different scales and matched at each scale separately, using DTW. The matched signals were used, rather than the reconstructed signals, to obtain the synchronised signal.

In Kassidas et al., 1998a an iterative method based on DTW was proposed in order to synchronise signals from industrial batch processes to a common length. Then they can build a MPCA/MPLS batch monitoring model. The method is multivariate and it assigns a weight for each single variable after 10 iterations depending on the smallest deviation of its average trajectory. The authors applied the same multivariate method in Kassidas et al., 1998b for pattern comparison. On this occasion the signals were filtered to remove the magnitude information and PCA was used to reduce the large number of correlated variables. Then, a set of relative similarity measures permitted fault diagnosis according to the reference pattern that gives the minimum distance. Ramaker et al., 2003 modified the procedure proposed by Kassidas and presented a method for warping spectral batch data. The main changes are related to scaling, the weighting matrix and the number of iterations.

Vullings et al., 1998 uses a PLA algorithm (Piecewise Linear Approximation) to segment the filtered heartbeat signal. After this, an algorithm decomposes the signal into three overlapped parts and it is scaled in magnitude. Finally, each of these parts is compared to reference patterns by DTW. The distance between lines is calculated using the difference of slopes and the length of the line segments.

Another medical application of the DTW algorithm is carried out in Caiani et al., 1998. In the definition of local distance they use the first derivative of the left ventricle volume signal together with amplitude. Both the amplitude and its first derivative were normalised by dividing by their respective peak-to-peak amplitudes.

In Kassidas et al., 1998a DTW is used to synchronise batch trajectories by combining it with Multiway Principal Component Analysis (MPCA) and Multiway Projection to Latent Structures (MPLS). Initially, there is a set of data for each variable belonging to different batch processes. These trajectories must be synchronised and they propose an iterative method based on DTW. After synchronisation all trajectories have the same duration and so an average trajectory could be defined. Using this trajectory as the reference, a normalised weight matrix computing the weight of each variable is calculated. Finally, by using the synchronised trajectories, they can to build a MPCA/MPLS model for monitoring the product quality of new batch processes.

The same authors use DTW in Kassidas et al., 1998b to obtain a distance between three reference patterns representing different faults and several test patterns, previously filtered and scaled. The patterns consist of a total of 26 variables, so that PCA is used to reduce the large number of correlated variables. When a new pattern of an unknown fault becomes available, it is projected onto the subspace defined by the PCA. These principal components can then be compared using the distances obtained from DTW.

Llanos et al., 2003 used dynamic time warping to cope with Case Based Reasoning (CBR). CBR methodology proposes a four-step cycle (retrieve, reuse, revise and retain). It basically consists of retaining experiences as cases for further reuse (submitted to a Revision procedure). Cases are registers containing a description of a problem "symptoms" and its solution "diagnostic". The aim is to reuse these cases for solving new problems by analogy. In the presence of a new problem, the basic procedure consists of retrieving analogue cases, according to their description (attributes defining symptoms), and reusing their solutions (diagnostic). DTW is used as a similarity criteria to implement the retrieval task, with the purpose of reducing the influence of time-misalignments in this task. The method is applied to the diagnosis of voltage faults registered in a 25kV substation. Bregón et al., 2006 carried out similar work on the early classification of several fault modes in a laboratory plant.

5.4.2. Variations

In situations with differences in the magnitude of the two signals, DTW would try to solve the variability in the Y -axis by warping the X-axis. This can lead to unintuitive alignments where a single point on one time series maps onto a large subsection of another time series.

On the other hand, DTW does not obey the triangular inequality, therefore, direct indexing under DTW is impossible and the quadratic time and space complexity of DTW creates the need for more efficient alternatives. To overcome these limitations some variations of the original algorithm have been proposed to cope with specific problems. The approaches fall into three categories:

1) ***Constraints*** – Limit the number of cells that are evaluated in the cost matrix.

2) ***Abstraction*** – Perform DTW on a reduced representation of the data.

3) ***Indexing*** – Use lower bounding functions to reduce the number of times DTW must be run during time series classification or clustering.

Constraints are widely used to speed up DTW. Two of the most commonly used constraints are the Sakoe-Chiba Band and the Itakura Parallelogram (see Appendix A). When constraints are used, DTW finds an optimal warp path through the constraint window. However, the globally optimal warp path will not be found if it is not entirely inside the window. Constraints work well in domains where time series have only a small variance in their temporal alignment, and an optimal warp path is expected to be close to a linear warp and passes through the cost matrix diagonally in a relatively straight line. In fact, for many applications, using constraints leads to better classification accuracy than unconstrained DTW (Ratanamahatana and E., 2004). However, constraints work poorly if time series are of events that start and stop at radically different times. In this scenario, the warp path can stray very far from a linear warp and nearly the entire cost matrix must be evaluated to find an optimal warp path.

Abstraction speeds up the DTW algorithm by operating on a reduced representation of the data. The resulting speedup depends on how much abstraction is used. Obviously, the calculated warp path becomes increasingly inaccurate as the level of abstraction increases. Projecting the low resolution warp path to the full resolution usually creates a warp path that is far from optimal. This is because, even if an optimal warp path passes through the low-resolution cell, projecting the warp path to the higher resolution ignores local variations in the warp path that can be very significant.

In Keogh and Pazzani, 1999 the time series are abstracted using piecewise linear segments. Each segment is characterised by a 4-tuple, representing the four points of its extremes. The distance between two segments is defined as the square of the distance between their means. Apart from this modification the algorithm is essentially unaltered. The algorithm, called Segmented Dynamic Time Warping (SDTW), showed similar results to those obtained by DTW, but with a significant decrease in the computation time. Exactly the same operation is shown in Keogh and Pazzani, 2000a, but this time the time series is represented using PAA, thus the algorithm is called Piecewise Dynamic Time Warping (PDTW). A drawback of PDTW is that it requires the user to choose a compression rate for the dimensionality reduction, and the algorithm is very sensitive to the value chosen. A high compression rate means a coarser approximation, which leads to an increase in false dismissals. A lower compression rate means a finer approximation, but slower computational time. The drawback of PDTW motivates the authors in Chu et al., 2002 to introduce the Iterative Deepening Dynamic Time Warping algorithm (IDDTW). The idea is to execute PDTW at increasingly higher resolutions until a desired “accuracy” is achieved.

In order to find more natural alignments, in Keogh and Pazzani, 2001, a modification of DTW that does not consider the *Y*-values of the data points, but rather considers the higher level feature of "shape", was proposed. Information about shape by considering the first derivative of the sequences is obtained; this algorithm was called Derivative Dynamic Time Warping (DDTW).

Here, the distance measured is the square of the difference of the estimated derivatives of x_i and y_j. For simplicity derivatives $\boldsymbol{x}$’ of $\boldsymbol{x}$ are estimated as:

$$\boldsymbol{x}' = \frac{(x_i - x_{i-1}) + ((x_{i+1} - x_{i-1})/2)}{2} \qquad [5.11]$$

This estimate is simply the average of the slope of the line through the point in question and its left neighbour, and the slope of the line through the left neighbour and the right neighbour. Empirically this estimate is more robust to outliers than any estimate considering only two data points.

Feng and Wah, 2003 propose a new warping technique for signature verification. The new approach, called Extreme Points Warping (EPW), first identifies the important peaks and valleys as extreme points (EP) and later it matches them through DTW. Since the corresponding matching pairs of EPs have to be peak-peak or valley-valley matching and the occurrence of ripples, a new local warping path and global cost equation are defined. Finally, they lineally warp the segments within the consecutive EPs and they use Euclidean distance or correlation coefficient to measure the similarity between the signals.

Lei and Govindaraju, 2004 propose a novel similarity measure based on linear regression analysis and DTW. Their technique, called Regression Time Warping, finds the path according to the closest pairs of points to a regression line, calculated previously. Obviously, this path is a sub-optimal solution, but it does not show any less accuracy than DTW under certain conditions and it is much faster.

Srinivasan and Qian, 2005 proposed an alternative approach that constrains the search for the corresponding points of the two signals, based on singular points in the signal that are derived from the perspectives of the operators. These constraints were used with DTW. The same authors propose a signal comparison-based approach to state identification and fault diagnosis in Srinivasan and Qian, 2006. The methodology, inspired by the Smith-Waterman algorithm (see section 5.6.3), modifies the initialisation of the DTW algorithm and then all the other elements are calculated recursively using the Itakura continuity constraint. This way, the method, called dynamic locus analysis, allows the search for the segment in sequence $\boldsymbol{y}$ that matches the complete sequence $\boldsymbol{x}$, where the longitude of $\boldsymbol{y}$ is much longer than $\boldsymbol{x}$. Sequence $\boldsymbol{y}$ is part of a reference database that has historical data of common transitions as well as abnormal operations. Sequence $\boldsymbol{x}$ is composed of the last m samples from an online sensor. Thus, the strategy is to compare this online signal with a collection of reference signals in order to perform fault diagnosis and state identification.

Somervuo, 2004 represent the data strings by a sequence of vectors in order to compute the average between two symbol strings. Each individual vector encodes one symbol position and the dimension of the vectors corresponds to the size of the alphabet. DTW was used for comparing the input strings against the model sequences for model adaptation. Then arithmetic averaging was used, first to determine the length of a new model vector sequence, and then the vector elements along the warping function. This approach provides an ordered display by means of representative local averages of the data.

Another variation called Segment-wise Time Warping (STW) is presented in Zhou and Wong, 2005. It combines a natural time scaling transformation through stretched segments and the DTW method. The time complexity of STW is quite high, therefore they need to use an index structure based on a lower bound function.

FastDTW algorithm has been introduced by Salvador and Chan, 2007 and was initially designed to cut off the computational cost of the original DTW. FastDTW basically consists of splitting the complexity of standard DTW by recursively down-sampling the time series. The warp path found at each iteration of the algorithm is then projected onto the higher resolution layer and serves as a guide that reduces computational complexity by spatially reducing the area handled by dynamic programming. FastDTW complexity is O(n), and is known to find an accurate minimum-distance warp path between two time series that is nearly optimal. FastDTW is superficially similar to IDDTW because they both evaluate several different resolutions, but IDDTW does not project low resolution solutions to higher resolutions.

Indexing uses ***lower-bounding*** functions (Yi et al., 1998; Kim et al., 2001; Keogh and Ratanamahatana, 2005) to prune the number of times DTW is run for similarity searches. Indexing speeds up applications in which DTW is used, but it does not make the actual DTW calculation any more efficient.

The lower-bound time warping distance function defined in Park et al., 2000 returns a distance value between a numeric value and a category symbol. The function is used to filter out dissimilar subsequences employing a disk-based suffix tree as an index structure. However, they only report the speedup for the indexing scheme (which is relatively small), and there is no comparison in terms of accuracy to the true DTW. Thus we can say nothing about the quality of the answer set. Their technique also allows false dismissals.

Among the initial existing lower bound functions, the one introduced in Keogh, 2002 is the best in terms of the tightness of the lower bound. For each query sequence Q that is defined, two new sequences U and L form a bounding envelope that encloses Q from above and below. Then, given a candidate sequence C and a query sequence Q of the same length n, the lower bound function, *LB_Keogh* is defined as follows:

$$LB\_Keogh(Q,C) = \sqrt{\sum_{i=1}^{n} \begin{cases} (c_i - U_i)^2 & if \quad c_i > U_i \\ (c_i - L_i)^2 & if \quad c_i < U_i \\ 0 & otherwise \end{cases}} \qquad [5.12]$$

Since multi-dimensional index structures begin to degrade rapidly somewhere above 16 dimension, they create a lower N dimension version of the function. For this reason, the sequences are represented by means of PAA (Fig. 5.5). Finally, to allow indexing they define a MINDIST(Q,R) function that returns a lower-bounding measure of the distance between a query Q, and R, where R is a Minimum Bounding Rectangle (MBR) (Vlachos et al., 2003).

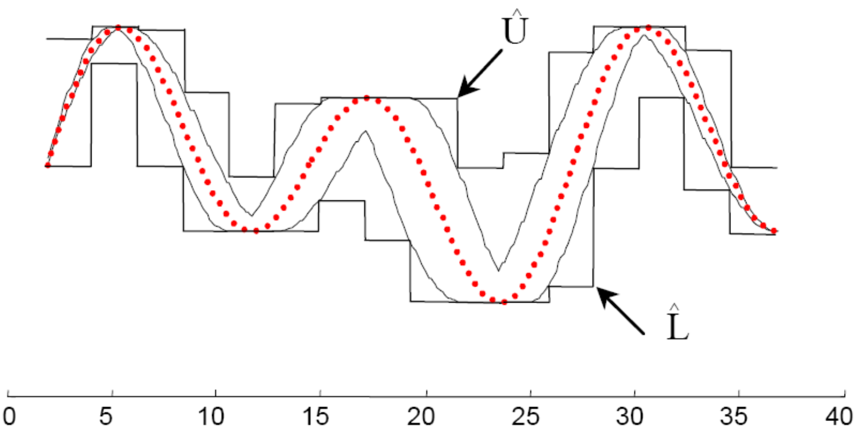


Fig. 5.5 An illustration of the sequences U and L, created for sequence Q (shown dotted) and the approximation of U and L by means of PAA.

In Wong and Wong, 2003, the authors use the same lower-bound function as in Keogh, 2002, but they suggest a new approach based on a sliding window of size *l* in order to handle subsequence matching of arbitrary length. A set of prefixes of length *l* of each possible subsequence is extracted and indexed as *l*-dimensional points by an R-Tree. The performance of the query process depends on the threshold and warping window values.

Shou et al., 2005 adopted a multi-step processing technique for similarity queries using DTW. First, the numerical sequences are decomposed using an Adaptive Piecewise Constant Approximation (APCA). Each segment s_i is represented by a triplet composed of the minimum and maximum value among all the elements contained in s_i, and the number of elements in s_i. Then they use the segments to derive lower bounds for DTW. For this, they employ a new Segmented Dynamic Time Warping (SDTW), which defines the distances between segments. Finally, they develop an index and a multi-step technique that uses the proposed bounds and performs two levels of filtering to efficiently process similarity queries. In the paper they also compare the tightness of the lower bounds proposed by several other authors.

Fu et al., 2005 propose a combination of DTW and uniform scaling called scaled and warped matching (SWM). The technique involve search pruning by means of a lower bounding technique to look for similar patterns under arbitrary time scaling.

Sakurai et al., 2005 proposed a new method FTW (Fast search method for dynamic Time Warping), which efficiently pruned a significant number of the search candidates to reduce the search costs. They use the coarse version of sequences to compute the lower bounding distance. Then it is shown that retrieval under the DTW can be faster by mixing progressively finer resolution and by applying early abandoning to the dynamic programming computation.

The *LB_Keogh* is also used in Xi et al., 2006 where a new approach using numerosity reduction (Pekalska et al., 2006, Wilson and Martinez, 1997) is employed. The authors observe empirically a best accuracy varying the warping window and they present an algorithm which dynamically adjusts warping window size during numerosity reduction.

5.5. Other related methods

This section presents other methods which have often been compared to DTW since all of them are based on temporal alignment.

5.5.1. Longest Common Subsequence (LCS)

A useful measure of similarity is the length of a longest common subsequence (Paterson and Dancik, 1994), often abbreviated to LCSS or LCS. This algorithm is based on the edit distance required in passing from one string to another one. LCS is the longest collection of elements which appears in both sequences and in the same order.

Formally, given a sequence $\boldsymbol{A} = \langle a_1, a_2 ... a_m \rangle$, another sequence $\boldsymbol{C} = \langle c_1, c_2 ... c_k \rangle$ is a subsequence of ***A***, if there is a strictly increasing sequence $[i_1, i_2, \ldots, i_k]$ of indices of ***A*** such as that for all $j = 1 \ldots k$, $a_{ij} = c_j$ (Cormen et al., 2001).

Then, given two sequences $\boldsymbol{A} = \langle a_1, a_2 ... a_m \rangle$ of length m and $\boldsymbol{B} = \langle b_1, b_2 ... b_n \rangle$ of length n, the longest common subsequence of ***A*** and ***B*** is a longest sequence ***C***, which is both a subsequence of ***A*** and ***B***.

In the most straightforward similarity scoring model, each match between aligned string characters contributes 1 to the final similarity score and there is no penalty for inserted spaces or mismatches. The solution to the LCS problem involves solving the following recurrence equation, where the cost for the edit operations is stored in the cumulative cell γ:

$$\gamma(i, j) = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ \gamma(i-1, j-1)+1 & \text{if } i, j>0 \text{ and } x_i = y_j \\ \max\begin{Bmatrix} \gamma(i, j-1) \\ \gamma(i-1, j) \end{Bmatrix} & \text{if } i, j>0 \text{ and } x_i \neq y_j \end{cases} \qquad [5.13]$$

For example, if ***A***= ⟨GAATTCAGTTA⟩ and ***B***= ⟨GGATCGA⟩ the LCS is ⟨GATCGA⟩, that is LCS(***A***,***B***)=6 (Fig. 5.6).

```
G A A T T C A G T T A
|   | |   |   |     |
G G A T - C - G - - A
```

Fig. 5.6 Alignment produced by the LCS algorithm.

Note that instead of using distance, “score" is used. Thus, the higher the score, the more similar the two time series. The LCS score can be converted into distance using the following formula:

$$LCS_{dist}(\boldsymbol{A},\boldsymbol{B}) = 1 - \frac{LCS(\boldsymbol{A},\boldsymbol{B})}{\min(m,n)} \qquad [5.14]$$

LCS applied to time series.

LCS belongs to the edit distance family of string matching algorithms. When LCS applies to time series, the complication is that the elements in ***A*** and ***B*** are not symbols, but real values. To take the real values into account, it relaxes equality to be within a certain tolerance ε, that is, $| x_i - y_i | \leq \varepsilon$.

The concept of longest common subsequences was then used by Das et al., 1997; defining similarity as follows: two sequences ***x*** and ***y*** are *F*-similar, if there is a function $\boldsymbol{f} \in F$ such that a long subsequence ***x***’ of ***x*** can be approximately mapped to a long subsequence ***y***’ of ***y*** using ***f***. They propose ***f*** to be a linear transformation and the similarity is defined as the length of the LCS, where two points are compared using a threshold.

Later, Vlachos et al., 2003 formalise non-metric similarity functions based on LCS. In their functions the stretching that is being provided by the LCS algorithm is within a certain range, as well as global translating of the sequences in space.

Differences between LCS and DTW.

Using LCS and DTW as a similarity measure between two sequences has the advantage that they can be of different lengths. However, there are important characteristics which differentiate both techniques.

DTW handles the expansion and contraction of the sequences but imposes the alignment onto the entire lengths of the sequences. Furthermore, its efficiency deteriorates for noisy data, since by matching all the points, it also matches the outliers, distorting the true distance between the sequences.

On the other hand, the LCS algorithm is very flexible at the starting/end points of the alignment, allowing some elements to be unmatched (Fig. 5.7), but does not deal with expansion/contraction. For example, if ***A***= ⟨`AABBCCDD`⟩ and ***B***= ⟨`ABCD`⟩, the algorithm yields a LCS(***A***,***B***)=4 and $LCS_{dist}(A,B)=0$, since the sequences have the subsequence ⟨ABCD⟩ in common. Now, if ***A***= ⟨AEBFCGDH⟩ and ***B***= ⟨ABCD⟩, the outputs are still the same, since the common subsequence is again ⟨ABCD⟩. Note that, as the first pair of sequences are just expanded/contracted versions of each other, they should be considered as more similar than the second pair.

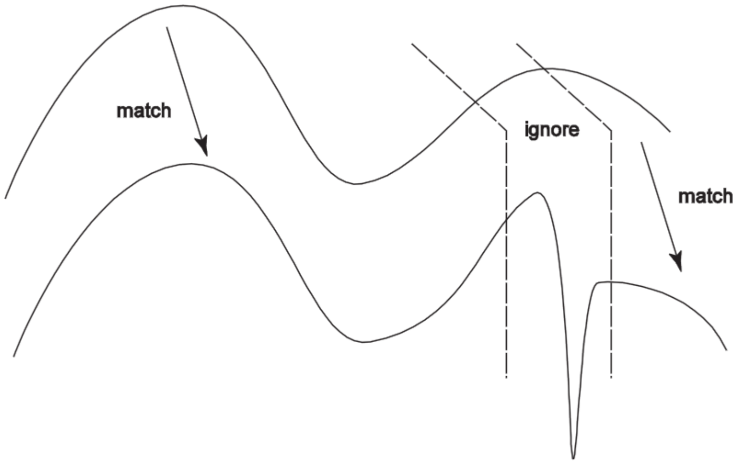


Fig. 5.7 Using LCS only the similar portions are matched, avoiding the outliers, while DTW tries to match all the elements, including the outliers (Vlachos et al., 2004).

On trying to import the expansion/contraction property from DTW to LCS, Guo and Siegelmann, 2004 proposed a change in the recurrence equation [5.13]. The modification allows the addition of the value of adjacent cells when $x_i=y_j$. So eq. [5.13] is substituted by:

$$\gamma(i,j)=\begin{cases}0 & \text{if } i=0 \text{ or } j=0\\ \max\begin{Bmatrix}\gamma(i,j-1)\\ \gamma(i-1,j)\\ \gamma(i-1,j-1)\end{Bmatrix}+1 & \text{if } i,j>0 \text{ and } x_i=y_j\\ \max\begin{Bmatrix}\gamma(i,j-1)\\ \gamma(i-1,j)\end{Bmatrix} & \text{if } i,j>0 \text{ and } x_i\neq y_j\end{cases} \tag{5.15}$$

5.5.2. Qualitative Similarity Index (QSI)

Cuberos et al., 2002 present the Qualitative Similarity Index (QSI) as a similarity measure between two original time series. The idea of this index is the inclusion of qualitative knowledge for the comparison of time series. The measure is based on the matching of qualitative labels (SDL language) that represent the evolution of the series values. Each label represents a range of values that may be assumed as similar from a qualitative perspective.

Let $\boldsymbol{x} = [x_0, ..., x_f]$ be a time series, the proposed approach is applied in three steps. First, a normalisation of the values of $\boldsymbol{x}$ is performed, in the interval [0,1]. Using this series the slope evolution or differences series $\boldsymbol{x}_D = [d_0, ..., d_{f-1}]$ is obtained. A label may be assigned to every different slope, so the range of all the possible slopes is divided into groups and a qualitative label is assigned to each group. The range division is defined depending on the parameter δ which is supplied by the experts according to their knowledge of the system. The value of this parameter has a direct influence on the quality of the results.

In Table 5.1 the first column represents the qualitative label for every range of derivatives, which is shown in the second row. The last column shows the character assigned to each label. The proposed alphabet contains three characters for increases and three for decreases ranges, and one additional character for the constant range. This alphabet is used to obtain the string of characters corresponding to the time series $\boldsymbol{x}$. Fig. 5.8 shows a normalised curve with their derivative values and the assigned label for each transition between adjacent values. This example has been obtained selecting δ=5.

Table 5.1 Qualitative labels for every range of derivatives.

Label	Range	Symbol
High increase	$[1/\delta, +\infty]$	`H`
Medium increase	$[1/\delta^2, 1/\delta]$	`M`
Low increase	$[0, 1/\delta^2]$	`L`
No variation	0	`0`
Low decrease	$[-1/\delta^2, 0]$	`l`
Medium decrease	$[-1/\delta, -1/\delta^2]$	`m`
High decrease	$[-\infty, -1/\delta]$	`h`

Finally, the similarity between two time series is calculated by means of the comparison of the two strings obtained from them, applying the previous transformation process, and then using the LCS algorithm.

Formally, let $\boldsymbol{x}$, $\boldsymbol{y}$ be time series and $\boldsymbol{A}$, $\boldsymbol{B}$ be the sequences of strings obtained when $\boldsymbol{x}$, $\boldsymbol{y}$ are normalised and labelled respectively. The QSI similarity between the strings $\boldsymbol{A}$, $\boldsymbol{B}$ is defined as follows:

$$QSI(\boldsymbol{A},\boldsymbol{B}) = \frac{\nabla(LCS(\boldsymbol{A},\boldsymbol{B}))}{m} \qquad [5.16]$$

where $\nabla\boldsymbol{S}$ is the counter quantifier applied to string $\boldsymbol{S}$. The counter quantifier yields the number of characters of $\boldsymbol{S}$. On the other hand, m is defined as $m = max(\nabla\boldsymbol{A},\nabla\boldsymbol{B})$. Therefore, the QSI similarity may be understood as the number of ordered symbols that we may find in the same order in both sequences simultaneously, and this value is divided by the length of the longest sequence.

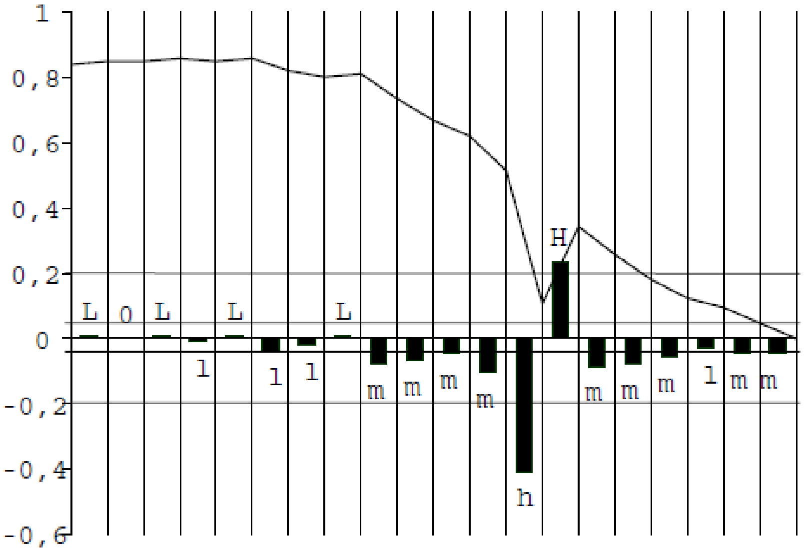


Fig. 5.8 Sample of translation.

5.5.3. SAX similarity

The SAX method (Symbolic Aggregate approXimation) allows a time series of arbitrary length n to be reduced to a string of arbitrary length w where typically $w<<n$ (see section 3.3.3). Then, the authors (Lin et al., 2003) define a distance function MINDIST (eq. [5.17]) that returns the minimum distance between the original time series of two words (Fig. 5.9). The distance measure is based on looking up the distances between each pair of symbols and their Euclidean distance. For example, if the cardinality is 4, the *dist*() function is implemented using the table lookup illustrated in Fig. 5.10.

$$MINDIST(\hat{Q},\hat{C}) = \sqrt{\frac{n}{w}}\sqrt{\sum_{i=1}^{w}(dist(\hat{q}_i,\hat{c}_i))^2} \qquad [5.17]$$

$$\hat{C} = \mathbf{b\,a\,a\,b\,c\,c\,b\,c}$$

$$\hat{Q} = \mathbf{b\,a\,b\,c\,a\,c\,c\,a}$$

Fig. 5.9 The distance between two SAX representations requires looking up the distances between each pair of symbols.

	a	**b**	**c**	**d**
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

Fig. 5.10 The lookup table used by the MINDIST function for an alphabet of cardinality of 4. For example dist(a,c)=0.67.

The value in cell (*i*,*j*) for any lookup table is calculated by the expression:

$$cell_{i,j} = \begin{cases} 0, & if \quad |i-j| \leq 1 \\ \beta_{\max(i,j)-1} - \beta_{\min(i,j)}, & otherwise \end{cases} \qquad [5.18]$$

Where β are the breakpoints that divide a Gaussian distribution in an arbitrary number of equiprobable regions (Fig. 5.11). The new distance lower bounds the Euclidean distance (Keogh et al., 2001a) between the original subsequences.

β_i \ a	3	4	5	6	7	8	9	10
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4			0.84	0.43	0.18	0	-0.14	-0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

Fig. 5.11 A lookup table that contains the breakpoints that divide a Gaussian distribution in an arbitrary number (from 3 to 10) of equiprobable regions.

5.5.4. Correlation Optimised Warping (COW)

Nielsen et al., 1998 described a new method of aligning two chromatographic profiles by piecewise linear stretching and compression of the time axis of one of the profiles. The method is referred to as correlation optimised warping (COW) since the optimal alignment is determined by correlation of the aligned fragments of signals.

Consider two chromatographic profiles that need to be aligned. One of them is chosen as a target profile (*T*) and the second profile (*R*) is to be aligned with it (Fig. 5.12). The unaligned profile *R* with length L_R is divided into *N* sections, each with length *m*. The target profile with length L_T is also divided into *N* sections. Each section, denoted ***w***, is warped (i.e., stretched or compressed) by linear interpolation, so that a signal *R*' with the same length as *T* is obtained.

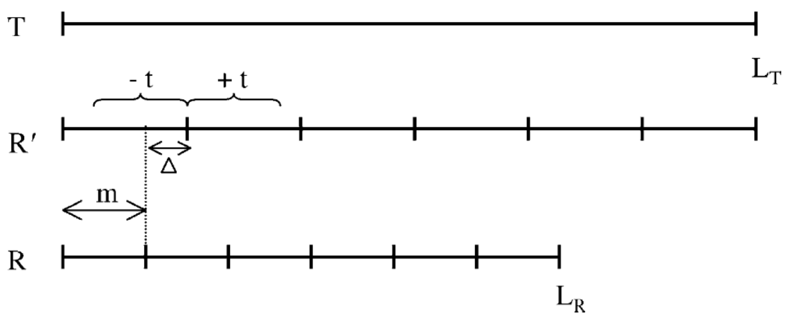


Fig. 5.12 Schematic presentation of the structure of the optimal warping problem.

To treat unequal lengths of two profiles (i.e., *T*, *R*), the difference in section length in *R* and *T* is defined as $\Delta = L_T/N - m$.

A finite number of possible warpings (i.e., sets of indices that allow stretching or compression) is examined for each section. The range of possible warpings is determined by the input parameter t that expresses maximum warping. Warping, denoted u, varies in the range ($\Delta - t$, $\Delta + t$).

In the next step, the matrix **F** with size (N + 1, L_T+1) is generated (Fig. 5.13). All elements of matrix **F** are initialised as minus infinity, except element $f(N+1, L_T+1)$, which equals zero. The zero element respects the fact that the last point of target profile T is aligned with last point of profile R.

During the optimisation procedure, elements of matrix **F** (determined by intervals [J_{start} , J_{end}] for each section) are replaced by cumulative benefit function, constructed based on correlation coefficient cc of corresponding parts in target profile $\boldsymbol{w}_T$ and interpolated profile $\boldsymbol{w}_{R'}$:

$$c\mathrm{c} = \frac{(\boldsymbol{w}_{\mathrm{T}} - \bar{\boldsymbol{w}}_{\mathrm{T}})^{\mathrm{T}} (\boldsymbol{w}_{\mathrm{R'}} - \bar{\boldsymbol{w}}_{\mathrm{R'}})}{std(\boldsymbol{w}_{\mathrm{T}})std(\boldsymbol{w}_{\mathrm{R'}})} \qquad [5.19]$$

where $std(\boldsymbol{w})$ is the standard deviation and $\bar{\boldsymbol{w}}$ is the mean of the vector $\boldsymbol{w}$. The boundary values for intervals (Fig. 5.13) of possible positions of ending points are calculated as:

$$\begin{aligned} J_{start} &= 1 + \max \begin{Bmatrix} (i-1)(m+\Delta-t) \\ L_{\mathrm{T}} - (N-i+1)(m+\Delta+t) \end{Bmatrix} \\ & \qquad\qquad\qquad\qquad\qquad \forall \quad i=1,\ldots,N \qquad [5.20] \\ J_{end} &= 1 + \min \begin{Bmatrix} (i-1)(m+\Delta+t) \\ L_{\mathrm{T}} - (N-i+1)(m+\Delta-t) \end{Bmatrix} \end{aligned}$$

The correlation coefficient is calculated for each possible position of ending point $f(i, j)$ (indicated by squares in Fig. 5.13) varying all possible warping u in the range ($\Delta - t$, $\Delta + t$) and added to the value of the benefit function for the previous ending point. Only the highest value of the cumulative correlation coefficient is kept in matrix **F**. Simultaneously, matrix **U** (Fig. 5.13), with the same size as **F**, is constructed, where corresponding values of u are kept (i.e., for the highest benefit function). The procedure starts from the last but one row of matrix **F** and continues to the first row of matrix **F**. When the value of the benefit function for the first point is calculated, all previous suboptimal solutions are known. The reconstruction of the signal is done using only matrix **U**, which contains the optimal warpings u for each section.

The algorithm has been used by other authors for chromatographic data (Pravdova et al., 2002, Tomasi et al., 2004).

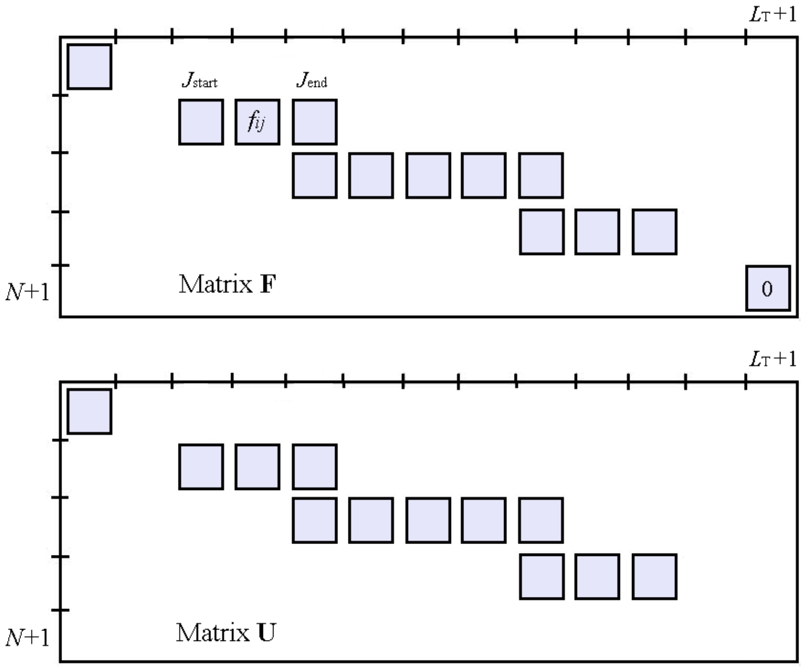


Fig. 5.13 Structure of the matrices *F* and *U*.

5.5.5. Time Warp Edit Distance (TWED)

A recently developed time series matching method was presented in Marteau, 2009.The method, called time warp edit distance (TWED), is defined as an elastic metric with a stiffness parameter to control the elasticity of the metric.

Given two time series $\boldsymbol{x} = [x_1, x_2, \ldots, x_m]$ and $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$, the goal is to edit $\boldsymbol{x}$ and $\boldsymbol{y}$ to completely superimpose the two curves using the three edit operations, $delete_A$, $delete_B$, and *match*, shown in Fig. 5.14.

The editing process is performed from left to right: If i is an index on the segments of $\boldsymbol{x}$ and j on the segments of $\boldsymbol{y}$, then the process initial setting is $i=j=1$. A *match* operation will increment i and j simultaneously: $i=i+1$ and $j=j+1$. A $delete_A$ operation will increment i only: $i=i+1$. A $delete_B$ operation will increment j only: $j=j+1$.

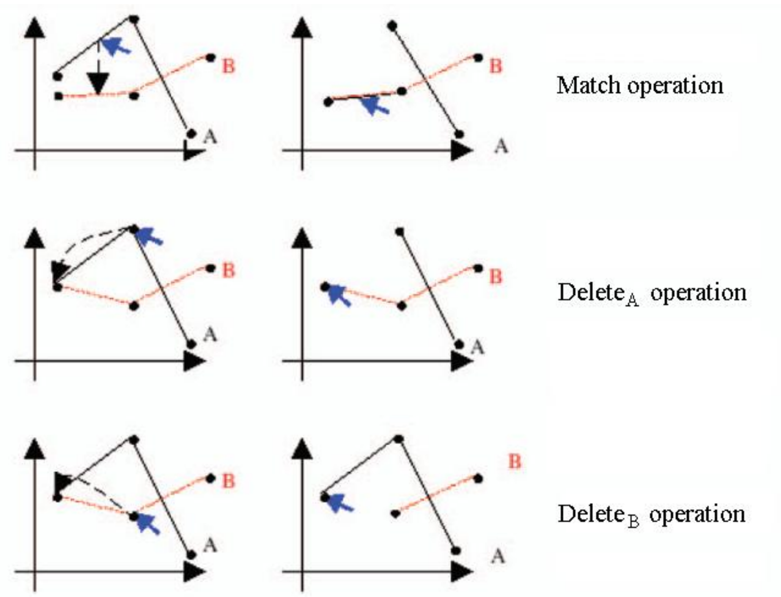


Fig. 5.14 The edit operations in the graphical editor paradigm.

Thus, the discrete time series similarity between two time series is measured as the minimum cost sequence of editing operations needed to transform one time series into another. Expressed as a formula, the distance between time series ***x*** and ***y*** is recurrently calculated by:

$$\delta_{\lambda,\upsilon}(X_i,Y_j) = \min\begin{cases} \delta_{\lambda,\upsilon}(X_{i-1},Y_j) + \Gamma(x_i \to \Lambda) & delete_A \\ \delta_{\lambda,\upsilon}(X_{i-1},Y_{j-1}) + \Gamma(x_i \to y_j) & match \\ \delta_{\lambda,\upsilon}(X_i,Y_{j-1}) + \Gamma(\Lambda \to y_j) & delete_B \end{cases} \qquad [5.21]$$

with

$$\begin{aligned} \Gamma(x_i \to \Lambda) &= d(x_i, x_{i-1}) + \lambda \\ \Gamma(x_i \to y_j) &= d(x_i, y_j) + d(x_{i-1}, y_{i-1}) \\ \Gamma(\Lambda \to y_j) &= d(y_j, y_{j-1}) + \lambda \end{aligned} \qquad [5.22]$$

The recursion is initialised as follows:

$$\begin{aligned} &\delta_{\lambda,\upsilon}(X_0,Y_0) = 0 \\ &\delta_{\lambda,\upsilon}(X_0,Y_j) = \infty \qquad \text{for } j \geq 1 \\ &\delta_{\lambda,\upsilon}(X_i,Y_0) = \infty \qquad \text{for } i \geq 1 \end{aligned} \qquad [5.23]$$

In eq. [5.22] Λ denotes the null sample, λ is a non-negative constant penalty and TWED introduces stiffness by choosing:

$$d(x,y) = d_{LP}(x,y) + \upsilon \cdot d_{LP}(t_x, t_y) \qquad [5.24]$$

Where υ is a non-negative constant to characterise the stiffness, d_{LP} is the l_p norm metric, and t_x,t_y represents the time for x and y respectively.

In Marteau, 2009, the effectiveness of TWED has been empirically proven using 20 different data sets. The value of the stiffness parameter υ is selected from $\langle 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\rangle$ and the value of the constant penalty λ is selected from $\langle 0, 0.25, 0.5, 0.75, 1.0\rangle$.

These values are selected for each data set so as to minimise the classification errors estimated on the training data. If different (υ,λ) values lead to the minimal error rate estimated for the training data, then the pairs containing the highest υ value are selected first, and the pair with the highest λ value is selected last.

The TWED method is adopted in Liu et al., 2010 for classification of wrist pulse blood flow signals.

5.5.6. Edit Distance with Real Penalty (ERP)

Based on L_p-norms, DTW and the domain of strings, Chen and Ng, 2004 proposed the Edit distance with Real Penalty (ERP). The idea is to develop a metric distance function such as the L_1-norm but supporting local time shifting.

During the calculation of the ERP distance, two normalised time series $\boldsymbol{x}$ and $\boldsymbol{y}$ of different lengths, are aligned to the same length by adding some symbols (also called gaps) to them. Then each element in one time series is either matched to a gap or an element in the other time series. The ERP distance is calculated recursively using the expression in eq. [5.25]:

$$ERP(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \sum_{1}^{n} \left| y_j - g \right| & \text{if } m=0 \\ \sum_{1}^{m} \left| x_i - g \right| & \text{if } n=0 \\ \min \begin{Bmatrix} ERP(x_{i-1}, y_{j-1}) + dist_{ERP}(x_i, y_j) \\ ERP(x_{i-1}, y_j) + dist_{ERP}(x_i, gap) \\ ERP(x_i, y_{j-1}) + dist_{ERP}(y_j, gap) \end{Bmatrix} & \text{otherwise} \end{cases} \quad [5.25]$$

where,

$$dist_{ERP}(x_i, y_j) = \begin{cases} \left| x_i - y_j \right| & \text{if } x_i, y_j \text{ not gaps} \\ \left| x_i - g \right| & \text{if } y_j \text{ is a gap} \\ \left| y_j - g \right| & \text{if } x_i \text{ is a gap} \end{cases} \quad [5.26]$$

and g could be any value. The authors of the algorithm pick g=0 and they claim that the experimental results presented are based on 24 benchmark data sets from many areas.

This algorithm yields as result a score equivalent to the number of substitutions and gaps, so the produced alignment may be as unintuitive as DTW if episodes were used. In fact, if g=0 the recursive equation when m>0 and n>0 is equivalent to the string edit distance (Gusfield, 1997) using the discrete metric for two symbols. Moreover, the distance is not a normalised value that would serve as a reference for quantifying significance of the comparison. Finally, like DTW, all elements in the sequences must be aligned, contributing to the final distance.

The ERP algorithm has been used in Zhang et al., 2010 for pulse waveform classification and Afonso et al., 2011 to develop a hurricanes trajectories storage method.

5.5.7. Edit Distance on Real sequence (EDR)

Another distance measure (EDR) based on edit distance on strings is proposed in Chen et al., 2005. Similar to the LCS variation, EDR also uses a tolerance ε to quantify the distance between a pair of points to two values, 0 and 1. Unlike LCS, EDR assigns penalties to the gaps between two matched segments according to the lengths of the gaps.

Formally, the Edit Distance on Real sequence (EDR) between ***x*** and ***y*** is the number of insert, delete, or replace operations that are needed to change ***x*** into ***y***. EDR(***x***, ***y***) is defined as follows:

$$EDR(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} n & \text{if } m=0 \\ m & \text{if } n=0 \\ \min\begin{Bmatrix} EDR(x_{i-1}, y_{j-1}) + subcost \\ EDR(x_{i-1}, y_j) + 1 \\ EDR(x_i, y_{j-1}) + 1 \end{Bmatrix} & \text{otherwise} \end{cases} \qquad [5.27]$$

where

$$subcost = \begin{cases} 0 & \text{if } \left|x_i - y_j\right| \le \varepsilon \\ 1 & otherwise \end{cases} \qquad [5.28]$$

In eq. [5.27] it is assumed that the cost of a replace, insert, or delete operation is only 1, which corresponds to the original definition of edit distance (Gusfield, 1997).

5.6. Pairwise sequence alignment

One way of representing sequences is by means some alphabet, often characteristic of the application. Although these sequences can be a function of time, continuous situations are dealt with by converting them into discrete ones by sampling the time interval. This is the case for the LCS, QSI and SAX methods outlined above, among others (Huang and Yu, 1999, Jonsson and Badal, 1997). This section details a short review of some central themes in comparisons of sequences of symbols which do not involve time.

The procedure for evaluating the difference between two sequences is typically done by first aligning the sequences and then deciding whether that alignment has occurred because the sequences are related. Pairwise alignment is most often described in terms of edit distance or similarity. These are opposite and interchangeable notions. In edit distance the score increases with the number of changes required to transform one sequence into the other, so a smaller score indicates a stronger similarity between the sequences. Similarity follows the idea that the more matched characters between two strings (without changing the order) the larger the positive value for an alignment score.

This type of alignments is widely used for the analysis of biological sequences. Once an alignment has been established, dynamic programming is employed to find the score of the alignment that maximises the similarity. Alignments follow a similar dynamic programming formulation to edit distances.

5.6.1. Sequence alignment

Given a sequence ***A*** = $\langle a_1, a_2... a_{m-1}, a_m \rangle$ of length m, and a sequence ***B*** = $\langle b1, b2..., b_{n-1}, b_n \rangle$ of length n, with both consisting of symbols from an alphabet $\mathcal{Z}$ of size c.

Aligning ***A*** and ***B*** can be considered as the process of transforming sequence ***A*** into sequence ***B*** by:

- Substitutions (or replacements)
- Deletions and insertions (referred to as indels)
- Compressions and expansions

Dealing with differences between sequences due to substitution, deletion-insertion, and compression-expansion is the central theme of sequence comparison. The most basic approaches to analysing the differences between sequences are limited to substitutions and *indels* (insertions and deletions). The common modes for such analyses are by means of a trace or matching (also defined as alignment by some authors). Distinctions between these analyses are illustrated in Fig. 5.15, where the symbol "-" stands for a gap.

```
I N D U S T R Y
|  |
I N T E R E S T          Trace


I N D U S T - R Y - -
                         Matching
I N - - - T E R E S T
```

Fig. 5.15 Two modes for analysing differences between sequences.

A trace from ***A*** to ***B*** consists of the source sequence ***A*** above and the target sequence ***B*** below, usually with lines from some elements in the source to some elements in the target providing a correspondence. An element can have no more than one line, and the lines must not cross each other.

A matching M (or alignment) is a mapping between the elements of two sequences ***A*** and ***B***. Two elements a_i of the source ***A*** and b_j of the target ***B*** are said to be aligned in respect to an alignment M, if the alignment of ***A*** and ***B*** maps a_i and b_j onto each other. Given an alignment M, a gap ("–") replaces elements of sequence ***A*** not aligned with any of those in sequence ***B*** (or vice versa). Relative to sequence ***A***, an insertion is a contiguous stretch of residues from A aligned with the gap character. On the other hand, a deletion is a stretch of gaps aligned with residues from ***B***.

As modes of presentation, traces and alignments each have their own advantages. However, both are used to tell whether two or more sequences are related and to give an impression of how close their relationship is in terms of sequence similarity. In Sankoff and Kruskal, 1983 they gathered an excellent collection of introductory papers on the topics of sequence searching, sequence alignment, pattern recognition and speech processing and recognition.

5.6.2. Scoring Models

The goal of sequence alignment is to pinpoint regions of high similarity between the two sequences. An algorithm for sequence alignment will, in general, pursue this identification by maximising a certain score that quantifies the similarity between the sequences in the particular alignment. Hence, the algorithm for sequence alignment is essentially just a mathematical optimisation procedure based on a predefined scoring scheme.

The simplest way to give a score to an alignment of two sequences is to calculate their Hamming distance: for two sequences of equal length, the different positions are counted. This distance measure is in general not flexible enough. Sequences may have different lengths and corresponding residues may have been shifted to different positions by deletions or insertions. The total score of an alignment should be a sum of terms for each aligned pair of elements plus gap contributions over all aligned residues, so that the best alignment is the one with the highest score. Mainly, the scoring models reward a match between two elements with a positive score, whereas a mismatch and inserted gaps are penalised. In biological applications the scores are usually represented in the form of substitution (score) matrices, where each cell provides a measure of how similar two terms are.

So, the particular alignment of the two sequences $\boldsymbol{A} = \langle a_1, a_2 ..., a_m \rangle$ and $\boldsymbol{B} = \langle b_1, b_2 ..., b_n \rangle$ yields an alignment score. Denoting the similarity score for aligning a_i with b_j by $\delta(a_i,b_j)$ and the number of gaps of length l in the alignment by $N_{\text{gap}}(l)$, the alignment score $\gamma(\boldsymbol{A},\boldsymbol{B})$ is defined as:

$$\gamma(\boldsymbol{A},\boldsymbol{B}) = \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{n} N_{ij}\delta(a_i,b_j)}_{sum\ \ up\ \ similarities} - \underbrace{\sum_{l} N_{gap}(l)g(l)}_{substract\ \ gap\ \ penalties} \qquad [5.29]$$

where N_{ij} =1 if the alignment contains the pair (a_i,b_j), and N_{ij} =0 if not; $g(l)$ represents the (positive) gap penalty function. Since similarity and gap penalty scores reject the likelihood of evolutionary events, the optimal alignment (i.e., the particular alignment that receives the largest alignment score) will delineate the series of evolutionary events most likely to have taken place.

Sellers, 1974 showed that the smallest number of steps required to change one sequence into the other could be calculated by the dynamic programming algorithm. Optimal alignments can be computed via the same schemes for maximum similarity by replacing the minimum distance by a maximal similarity scoring scheme.

5.6.3. Global and local alignments

Generally, one can distinguish between global and local alignment procedures. A global alignment will always cover the entire input of sequences, no matter how different these may be, and the algorithm determines the alignment that maximises the alignment score over the full length of both sequences. Global alignments are a reasonable approach for sequences that are related over their entire lengths. On the other hand, local alignments contain only contiguous parts of the sequence that are “similar”. In such cases, the algorithm computes the optimal local alignment by finding the pair of substrings of the full sequences whose alignment yields the highest alignment score among the set of all substrings and their possible alignments.

Global Alignment (Needleman-Wunsch)

The most widely used dynamic programming algorithm for global sequence alignment is the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). The idea behind all the versions is to build up an optimal alignment using previous solutions for optimal alignments of smaller subsequences.

The maximum match can be determined by representing the two sequences ***A***, ***B*** of length m,n respectively in a matrix indexed by i and j, one index for each sequence. The score value $\gamma(i, j)$ assigned to each cell is built recursively by the following recurrence:

$$\gamma(i.j) = \begin{cases} \text{if } i{=}0 \ \& \ j{=}0: & 0 \\ \text{if } i{>}0 \ \& \ j{=}0: & gap(i-1) \\ \text{if } i{=}0 \ \& \ j{>}0: & gap(j-1) \\ & \\ otherwise: & \delta(a_i, b_j) + \\ & \max\begin{pmatrix} \gamma(i-1, j-1) \\ \max_{1<k\le i}(\gamma(i-k, j-1) + gap(k)) \\ \max_{1<k\le j}(\gamma(i-1, j-k) + gap(k)) \end{pmatrix} \end{cases} \qquad [5.30]$$

In this formulation, $\delta(a_i, b_j)$ is the score given for matching the i^{th} symbol in string ***A*** to the j^{th} symbol in string ***B***. The penalty for a gap of size k is defined by gap(k). In its simplest form, $\delta(a,b)$ can be defined as 1 for a match and 0 for a mismatch. In the original paper, the symbols (amino acids) were numbered from the N-terminal end, although the direction makes no difference to the final result, so the table was filled in starting at the end of the sequences at position (0,0).

Every possible comparison will be represented by pathways through the array. An i or j can occur only once in a pathway because a particular symbol cannot occupy more than one position at one time. Furthermore, the only permissible relationships of their indices are $m>i$, $n>j$ or $m<i$, $n<j$. Any other relationships represent permutations of one or both amino acid sequences which cannot be allowed since this destroys the significance of a sequence.

A pathway is signified by a line connecting cells in the array. Proceeding along complete diagonals with no deviations would imply an alignment without any gaps. The introduction of a gap (either by an insertion or a deletion) in either sequence would correspond to moving either above or below the main diagonal (Fig. 5.16).

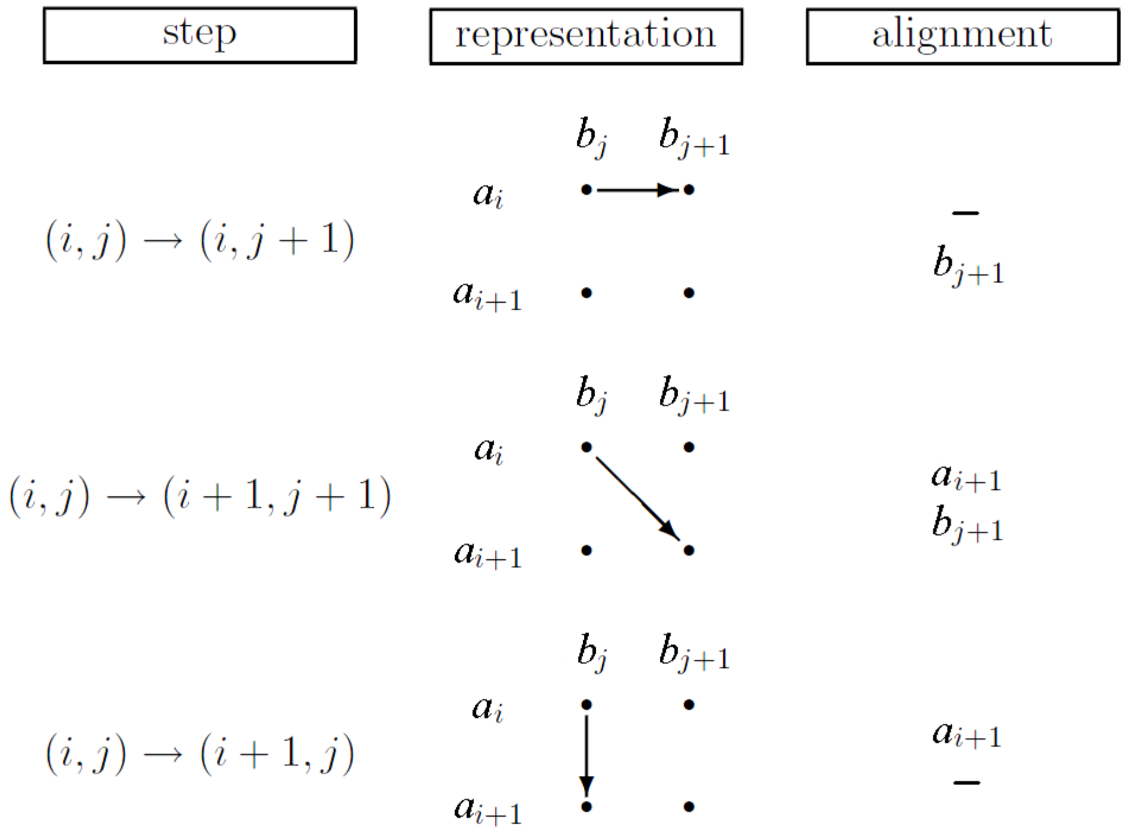


Fig. 5.16 The next three possible steps from the element (i; j), their representation in the alignment matrix and the corresponding alignment.

To find the best route, Needleman and Wunsch suggested modifying the matrix to represent this idea of tracing different pathways through the matrix. From all the possible pathways only the one which is best (in terms of maximising a score) can be chosen. Their method consists of two passes through the matrix. The first pass traces a score for all possible routes and moves right to left, bottom to top. Once the scores for all possible routes are found, the maximum can be chosen (it will be somewhere on the topmost row or leftmost column) and a second pass can be carried out, this time running left to right, top to bottom to find the alignment that gives the maximum score.

The reason that the algorithm works is that the score is made up of a sum of independent pieces, so the best score up to a point in the alignment is the best score up to the point one step before, plus the incremental score of the new step.

Local alignment

It is often the case that one or more regions of high similarity will exist in two sequences that are otherwise dissimilar. Then, short and highly similar subsequences may be missed in the global alignment because they are outweighed by the rest of the sequence. Hence, one would aim to create a locally optimal alignment.

A small modification of the original Needleman-Wunsch algorithm that allows the determination of the optimal local alignment has been introduced by Smith and Waterman, 1981. The key difference is based on the introduction of zero as a new option in the recursion relation (eq. [5.31]) which has the effect of terminating any path in the alignment matrix in which the score drops below zero. The algorithm requires that the scoring function be negatively biased so that regions of low similarity will have negative scores.

This is required in order to cause the score to drop as more and more mismatches are added. Hence, the score will rise in a region of high similarity and then fall outside of this region. If there are two segments of high similarity then these must be close enough to allow a path between them to be linked by a gap or they will be left as independent segments of local similarity. After optimal alignment scores have been calculated for all nodes in the usual recursive way, the optimal local alignment is found by locating the node with the largest alignment score and performing a trace-back starting from this node, until a node is encountered in which the alignment score is equal to zero.

$$\gamma(i.j) = \begin{cases} \text{if } i\text{=0 or } j\text{=0:} \quad 0 \\ otherwise: \qquad \max \begin{pmatrix} 0 \\ \gamma(i-1, j) + \delta(a_i, -) \\ \gamma(i, j-1) + \delta(-, b_j) \\ \gamma(i-1, j-1) + \delta(a_i, b_j) \end{pmatrix} \end{cases} \qquad [5.31]$$

In eq. [5.31] vertical movements are penalised with the insertion function $\delta(a_i,-)$, while the horizontal movements are penalised with the deletion function $\delta(-,b_j)$. Usually these functions correspond to negative scores.

To illustrate the difference between global and local alignments, Fig. 5.17 shows two alignments of the same DNA sequences. The first shows a weak global alignment while the second shows a stronger local alignment.

```
TTGACACCCTCC-CAATTGTA
    ::  ::   ::  :
ACCCCAGGCTTTACACAT---

---------TTGACACCCTCCCAATTGTA                TTGACAC
         :: ::::                   or        :: ::::
ACCCCAGGCTTTACACAT-----------                TTTACAC
```

Fig. 5.17 Global vs. Local Alignment.

More research has been carried out to create gap-costs that allow block insertions and deletions (Gotoh, 1982; Sankoff and Kruskal, 1983). Gotoh, 1982 devised a 3-state, affine gap costs model of mutation which improves time efficiency to scoring gaps in sequence alignment. His work has been used in other research to treat gaps in alignments (Allison, 1993).

The *k*-best variation of the Smith-Waterman algorithm (Waterman and Eggert, 1987) returns non-overlapping local alignments that score at or above a preset level. This is particularly useful if two sequences share multiple regions of similarity interrupted by dissimilar regions and with the order of the similar regions rearranged.

Heuristic methods

Motivated by the problem of finding sequences in large databases, the heuristic similarity search algorithms FASTA (Lipman and Pearson, 1985) and BLAST (Altschul et al., 1990) were created.

FASTA considers exact matches between short sub-strings k. If a significant number of such exact matches are found, FASTA uses the dynamic programming algorithm to compute optimal alignments. This approach allows speed to be traded for precision. The larger the parameter k, the smaller the number of exact matches. This makes the program faster but loses precision as it becomes less likely that the optimal alignment contains enough exact matches of length k and the procedure may find nothing. Nevertheless, experience shows that with sensitively chosen parameters, FASTA misses very few cases of significant homology.

BLAST is another heuristic method based on a similar idea. BLAST focuses on no gap alignments of (again) a certain fixed length k. Rather than requiring exact matches, BLAST uses a scoring function to measure similarity, rather than distance. In particular, for proteins, one can argue that segment pairs with no gaps and high similarity scores indicate regions of functional similarity. For a given threshold parameter S, BLAST reports to the user all database entries which have a segment pair with the query sequence that scores higher than S. If the scoring function used has a probabilistic interpretation, BLAST can also give an assessment of the statistical significance of the matches it reports.

Another heuristic method for sequence alignment was presented in Rognes and Seeberg, 1998. The algorithm, called SALSA, has many similarities to FASTA and BLAST, but includes a post-processing stage that increases sensitivity. The idea is to build an alignment from all the fragments found in the initial stages of the searching process. Then, fragments should be arranged by a gap obtaining the optimal score. The position of the gap is found using dynamic programming only if the score of the partial alignment on either side of the gap is higher than the gap penalty.

Subsequently, the aforementioned author introduced an improved method in Rognes, 2001. This time the algorithm exploits the parallel processing capability of the microprocessor to perform the same operation in parallel on several independent data sources. Like most heuristic algorithms the method can be divided into two phases. First it computes the exact optimal ungapped alignment score of each diagonal. Secondly, a novel heuristic search estimates a gapped alignment score taking into account the amount of sequence similarity on several diagonals. The fraction containing the 1% of the highest scoring database sequences is finally subjected to a rigorous Smith-Waterman alignment.

5.7. Similarities between qualitative trends

Finding a similarity measure for qualitative sequences is not easy because sequences that are qualitatively the same may be quantitatively different. Also the sequences may be of different lengths, making it difficult or impossible to embed the sequences in a metric space, or the sampling rates of sequences may be different. Most of the early researchers estimated a similarity measure between qualitative trends by matching the sequence of primitives or, at a second-level, using a distance measure between the data sets. Next, some methods for estimation of similarity between qualitative trends will be discussed.

A methodology for pattern recognition based on episodes is described in Bakshi and Stephanopoulos, 1994b. Each pattern is represented by a string of primitives identified by means of a pattern grammar. Pattern matching of the various generalised descriptions facilitates extraction of the qualitative and quantitative features, which are then used for solving the classification problem. For example given the two following trends:

A B C B C
A B C B C D A B C

The syntactic descriptions may be matched in three different ways:

A B C B C
A B C B C (D A B C)

A B C   B C
A B C   (B C D A)   B C

A   B C   B C
A (B C) B C (D A) B C

The qualitative features considered for classification are the features that remain unmatched after pattern matching, therefore, the syntactic features that allow complete classification are:

DABC

BCDA

BC**DA

Then each of these features is evaluated until the classification is solved. However, sometimes the simple syntactic features are not sufficient for complete classification. In this case, the quantitative characteristics of the trends are necessary. The quantitative features evaluated for classification in these generalised descriptions is the quantitative information contained in the matched triangles. The classification problem is resolved by means of the technique of induction by decision trees.

Yamanaka, 1997 detects the characteristic points (those having extreme curvature) of the patterns and store patterns to be detected (dictionary). Using characteristic points, the time series data is transformed into a series of line segments. Lastly, these line segments are compared to the dictionary by a dynamic programming matching method. The similarity obtained is the same as the correlation factor between time series data and a word in the dictionary.

Sundarraman and Srinivasan, 2003 describes a process variable as an ordered collection of enhanced atoms. An enhanced atom consists of a first-order shape, the time duration for which that shape is manifested, and the variable magnitudes at the beginning and end of the shape. When this trend includes quantitative information it is called an enhanced trend. For trend comparison, first a real-time trend is synchronised with a dictionary trend by comparing the expected and the observed shapes. Thus, a shape-matching degree (MD_S) is defined, being MD_S=1 if the two shapes match each other and MD_S=0 if they are completely dissimilar. As a second measure, a magnitude-matching degree (MD_m) was defined based on the difference of the variable value during a phase and the expected evolution of the dictionary trend. Analogous to MD_m, a third similarity measure for duration MD_d was defined. A value of MD_m=1 and MD_d=1 signifies an exact match while a smaller or larger value indicates a low similarity. The degree of dissimilarity $f^x(i)$ is a measure of the total deviation between the dictionary trend and the real-time one for variable *x* and is calculated based on the three matching degrees:

$$f^x(i) = \max(|1 - MD_S(i)|, |1 - MD_m(i)|, |1 - MD_d(i)|) \quad [5.32]$$

In Dash et al., 2003 two similarity indices are defined based on the aligned qualitative sequences. The first one is based only on the sum of the similarities between individual primitives according to a similarity matrix, and the second one takes time into account as a weighting factor. It is not clear how the similarities between primitives have been defined and the authors claim that a variety of similarity measures may be defined depending on the particular application.

Maurya et al., 2007 improves the last similarity measure adding shape-based and magnitude-based similarities. So the shape-based similarity is given by the normalised area under primitive P1 divided by the normalised area under primitive P2, where the normalised area under primitive P1 is lesser. The magnitude-based similarity between the primitives P1 with magnitude *a* and P2 with magnitude *b* penalise the difference in the magnitude and is given by:

$$Magnitude-based\_similarity = \exp\left(\frac{((b/a)-1)^2}{2\sigma_{mr}^2}\right) \quad [5.33]$$

where σ_{mr} is the variance parameter for magnitude ratio and it controls the extent of penalty for difference in the magnitude. This value is adjusted empirically to 10. The total distance between primitives is the product of their shape-based and magnitude-based similarities. For multivariate processes, the overall similarity measure or confidence index is given by CI = min(S_1, S_2, …, S_n) where S_k is the similarity measure between the trends of the k^{th} sensors in the two scenarios.

In Hung and Anh, 2007 they use a shape definition hierarchy tree (Xia, 1997) to classify the shapes of the linear segments in terms of slope trends. First, the segments obtained by PLA are converted into three main symbols in agreement to their slopes: U: sharp up, S: stable up and D: sharp down. Besides, each symbol is subdivided according to the angle of the segment. Then a shape definition hierarchy tree which represents a suitable set of slope trends is built (Fig. 5.18).

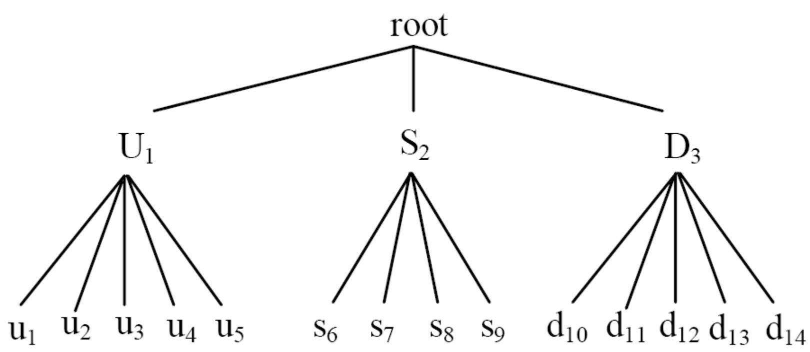


Fig. 5.18 The shape definition hierarchy tree used in Hung and Anh, 2007.

The distance between a pair of symbols a_i, b_j is defined as follows:

$$dist(a_i, b_j) = 2^{\max(0,|k-t|-1)} * \max(0, |j-i|-1) \qquad [5.34]$$

Where the parent nodes of a_i, b_j are A_k and B_t respectively.

Example:

$$dist(u_1, s_8) = 2^{\max(0,|2-1|-1)} * \max(0, |8-1|-1)$$

5.8. Conclusions

There is a lot of work which addresses the problem of similarity search and classification of time series. On examining these papers, it is common to distinguish between those based on transformations and those that propose a signature extraction, but an important problem arises due to time misalignments.

Sequence alignment is a widely used method for performing sequence similarity. An alignment may be constructed as an intermediate step or as a goal in itself. Thus, this chapter has summarised the most important techniques which deal with the temporal alignment problem and pairwise alignment used in the analysis of biological sequences. Again, the problem may arise due to the nature of the time series. For example, some methods based on sequence alignment require working with series of the same length.

A method that tries to solve this inconvenience is Dynamic Time Warping (DTW), which uses dynamic programming to align time series by stretching or shrinking them along its time axis. However, DTW can lead to unintuitive alignments and it is computationally expensive (in both time and memory), which normally makes it useful only for offline applications.

This dissertation proposes the use of qualitative representations of signal trends as representations of system behaviour. This representation reduces the dimension and it can keep the temporal meaning. Several approaches have been developed to estimate a similarity measure between qualitative trends, but they always address a particular problem and have shortcomings in generalised cases.

Based on some ideas from the algorithms presented above, the next two chapters will examine the comparison of process signals when trying to define a similarity function from a time evolution and qualitative perspective.

Chapter 6.

Similarity search based on DTW

6.1. Introduction

The advantage of representing process signals as a set of qualitative sequences has been presented above. To summarise, the benefits are twofold: first, they provide an intuitive representation that is very useful when one examines the process. And second, they reduce the complexity of variables by allowing only a finite set of qualitative descriptions.

On the other hand, many real applications require monitoring or diagnostic systems working online with the process. However, due to the great number of variables or the complexity of systems, this requirement is often impossible to execute.

In the previous chapter, DTW was shown as a good method to determine the similarity between two time series due to its capacity for aligning sequences with temporal misalignments and different longitudes. However, it has the disadvantage of being a computationally expensive algorithm and it could fail in the alignment by trying to solve the variability in the Y-axis by warping the X-axis. In this chapter, two modifications of the DTW algorithm are introduced.

The first modification consists of applying DTW, not in the original time series, but in its episode-based representations (Colomer et al., 2002a; Colomer et al., 2002b; Colomer et al., 2003). The second one proposes a slight modification of DTW in order to adapt it for online applications (Gamero et al., 2004; Llanos et al., 2004).

The following sections are devoted to explaining the main particularities of the new algorithms. Finally, two application examples will illustrate both methods.

6.2. EpDTW

DTW is a technique for aligning time series which has been applied successfully in many fields. However, these approaches are normally used for offline applications due to the expensive computational cost (in both time and memory) of the DTW algorithm. Furthermore, DTW, by matching all the elements is also going to try and match the outliers which, most likely, are going to distort the real distance between the sequences examined. The representation of a sequence as episodes reduces the calculation time by decreasing the amount of manipulated data. Likewise, the qualitative character that defines an episode avoids the problem of the variability in the Y-axis. In addition, the pre-processing data which converts process signals to episodes removes the outliers.

Therefore, DTW can be used to align episodes and obtain a global distance. The next subsection presents an approach called EpDTW which has been created following the basic principles of DTW and using episodes as a higher level representation of variables.

6.2.1. The algorithm

Formally, given two time series $\boldsymbol{x} = [x_1,x_2,\ldots,x_m]$ and $\boldsymbol{y} = [y_1,y_2,\ldots,y_n]$, they are converted to sequences of episodes $\mathbb{X} = \langle \mathbb{X}_1,\ldots, \mathbb{X}_i,\ldots, \mathbb{X}_M\rangle$ and $\mathbb{Y} = \langle \mathbb{Y}_1,\ldots, \mathbb{Y}_j,\ldots, \mathbb{Y}_N\rangle$ where $M<<m$ and $N<<n$. The indices i and j represent the relative position of each episode in the sequence ($1\leq i\leq M$, $1\leq j\leq N$).

As well as DTW, the EpDTW algorithm finds a warping path **W** of k points (eq. [6.1]) in a two-dimensional M by N cost matrix by evaluating the cumulative distance $\gamma(i,j)$. Every cell value $\gamma(i,j)$ in the matrix is calculated recursively as the sum of the local distance found in the current cell and the minimum of the cumulative distances of the adjacent elements (eq. [6.2]). So the distance between two sequences of episodes is obtained in an iterative way by accumulating local distances between those episodes which compose warping path W, thus minimising the global distance.

$$\begin{aligned} &\mathbf{W} = \mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_k \quad | \quad \max(M, N) \leq k \leq M + N \\ &\mathrm{w}_k = (i_k, j_k) \end{aligned} \qquad [6.1]$$

$$\gamma(i,j) = dist(i,j) + \min\begin{Bmatrix} \gamma(i-1, j-1) \\ \gamma(i-1, j) \\ \gamma(i, j-1) \end{Bmatrix} \qquad [6.2]$$

In eq. [6.2] *dist(i,j)* is the local distance $dist(\mathbb{X}_i, \mathbb{Y}_j)$ between episodes $\mathbb{X}_i$ and $\mathbb{Y}_j$ according to a distance matrix. Thus, working with qualitative symbols requires the definition of a local distance between them. In this sense, a distance matrix where each episode is related to the complete set is defined. Distances are based on the qualitative state and auxiliary characteristics that define the different types of episodes. However, these local distances could be subject to the criterion of the user, so one could give more importance to some episodes over others, obtaining a different global distance and preserving the essential features of the process signal. For example, Table 6.1 shows a distance matrix where the criterion has been to distinguish between three groups, with higher distances between types of different groups.

Table 6.1. Local distance $dist(\mathbb{X}_i, \mathbb{Y}_j)$ between episodes.

	⌉	⌈	⎛	∩	⎞	\	−	/	⎝	∪	⎠	⌋	⌊
⌉	0	1	1	.75	.25	.5	1	1	.75	1	1	1	1
⌈	1	0	.25	.75	1	1	1	.5	1	1	.75	1	1
⎛	1	.25	0	.5	1	1	.75	.25	1	1	.5	.75	1
∩	.75	.75	.5	0	.5	.75	.5	.75	1	1	1	1	1
⎞	.25	1	1	.5	0	.25	.75	1	.5	1	1	1	.75
\	.5	1	1	.75	.25	0	.5	1	.25	.75	1	1	.5
−	1	1	.75	.5	.75	.5	0	.5	.75	.5	.75	1	1
/	1	.5	.25	.75	1	1	.5	0	1	.75	.25	.5	1
⎝	.75	1	1	1	.5	.25	.75	1	0	.5	1	1	.25
∪	1	1	1	1	1	.75	.5	.75	.5	0	.5	.75	.75
⎠	1	.75	.5	1	1	1	.75	.25	1	.5	0	.25	1
⌋	1	1	.75	1	1	1	1	.5	1	.75	.25	0	1
⌊	1	1	1	1	.75	.5	1	1	.25	.75	1	1	0

Considering that the episodes are asynchronous, the time duration of each episode could be an important parameter to keep in mind in some applications. The EpDTW algorithm proposes, optionally, to add this information to local distances by modifying the former distance *dist*(*i*,*j*). Then, if time is important, the local distance is defined as:

$$dist(i,j) = dist(\mathbb{X}_i, \mathbb{Y}_j) * \min(d^i, d^j) + cp \qquad [6.3]$$

where d^i and d^j is the temporal extension of episodes $\mathbb{X}_i$ and $\mathbb{Y}_j$. The variable *cp* is a compression penalty calculated (eq. [6.4]) according to the maximum local warping w_{MAX}. The value w_{MAX} represents the maximum number of samples from one sequence that can be compressed to be aligned to one sample in the other sequence. Intuitively, there is some resemblance to the warping window used as a constraint in the original DTW algorithm.

$$cp = \max \begin{Bmatrix} \left\lceil \dfrac{\max(d^i, d^j)}{w_{MAX}} \right\rceil - \min(d^i, d^j) \\ \\ 0 \end{Bmatrix} \qquad [6.4]$$

Note that eq. [6.4] uses the *ceiling* function $\lceil z \rceil$. It returns the smallest integer value that is greater than or equal to a number. That is:

$$\lceil z \rceil = \min \quad n \in \mathbb{Z} \quad | \quad n \geq z \qquad [6.5]$$

Finally, the score obtained should be normalised according to the weight of each cell belonging to path W. Thus, in this case the final distance is:

$$dist_{EpDTW}(\mathbb{X}, \mathbb{Y}) = \frac{\gamma(M, N)}{\sum_{h=1}^{k} weight(\mathrm{w}_h)} \qquad [6.6]$$

The weight for each cell w_k represents the number of links between two episodes. This weight has a value of 1 when time is not considered. Thus, the denominator can be substituted by *k*. However, when episode duration is involved the weight for each cell is calculated as:

$$weight(\mathrm{w}_k) = weight(i_k, j_k) = \max \begin{Bmatrix} \min(d^i, d^j) \\ \\ \left\lceil \dfrac{\max(d^i, d^j)}{w_{MAX}} \right\rceil \end{Bmatrix} \qquad [6.7]$$

6.2.2. Illustrative examples

Example 1

Consider the four signals and their qualitative representation in Fig. 6.1. These signals have the same duration in time, but different temporary misalignments and variability in the Y-axis. The goal is to obtain a measure of similarity between them. For simplicity, each symbol can be represented by a letter as in Fig. 6.2.

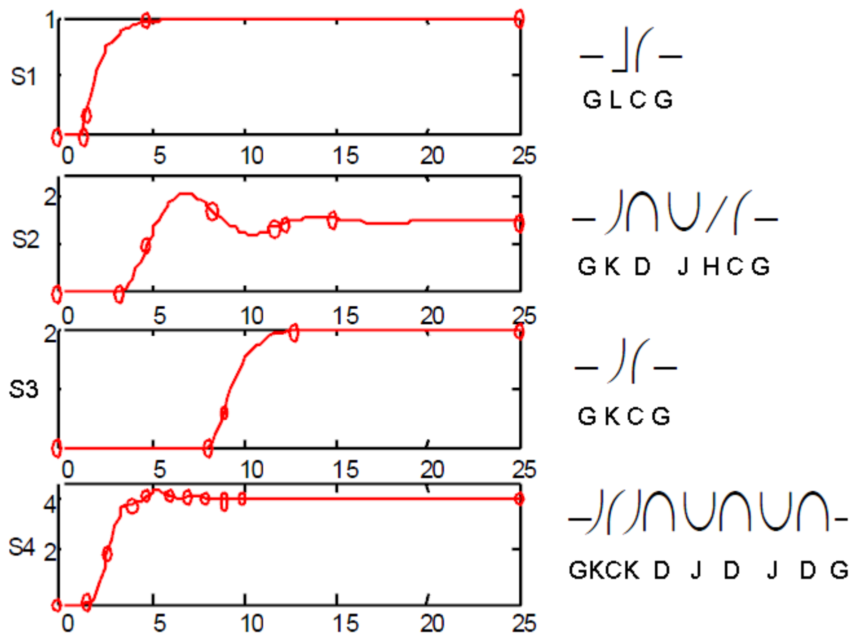


Fig. 6.1 Signals S to be compared and their representation *S*.

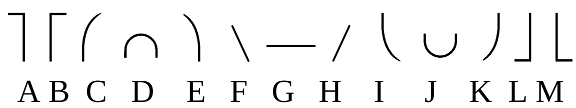


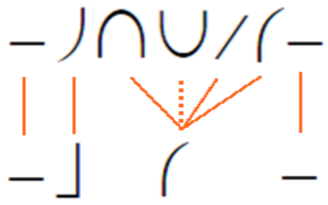
Fig. 6.2 Useful set of episodes.

In order to illustrate the EpDTW algorithm, signals S1 and S2 will be examined. By applying eq. [6.2], the cumulative distance matrix is filled in as shown in Fig. 6.3. The value 2 in the last cell represents the cost of the alignment. This value can be normalised to a distance between 0 and 1 by dividing the cost by the length of the path. Note that the italicised numbers in the matrix indicate a path **W** of 7 points. So, the distance between qualitative sequences *S*1 and *S*2 is 2/7=0.285.

Sequence $\mathbb{S}2$

Seq. $\mathbb{S}1$	—	)	∩	∪	/	(	—
—	***0***	0’75	1’25	1’75	2’25	3	3
⌋	1	***0’25***	1’25	2	2’25	3	4
(	1’75	0’75	***0’75***	***1’75***	***2***	***2***	2’75
—	1’75	1’5	1’25	1’25	1’75	2’5	***2***

(a)



(b)

Fig. 6.3 (a) The cumulative distance matrix for $\mathbb{S}1$ and $\mathbb{S}2$. (b) Temporal alignment deduced from the obtained path.

Following the same procedure, the distances between the four signals are obtained (Table 6.2). The values obtained are a normalised distance, so 0 represents a complete equality.

Table 6.2 Similarity using proposed EpDTW and Table 6.1.

	$\mathbb{S}1$	$\mathbb{S}2$	$\mathbb{S}3$	$\mathbb{S}4$
$\mathbb{S}1$	0	0.285	0.0625	0.325
$\mathbb{S}2$	0.285	0	0.214	0.25
$\mathbb{S}3$	0.0625	0.214	0	0.3
$\mathbb{S}4$	0.325	0.25	0.3	0

Example 2

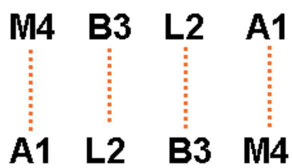
In this second example the importance of time will be illustrated by a simple case. Consider the two sequences of episodes: $\mathbb{X} = \langle$M4,B3,L2,A1$\rangle$ and $\mathbb{Y} = \langle$A1,L2,B3,M4$\rangle$, where each episode is represented by type (Fig. 6.2) and duration. In this case the distance between different types of episodes is always 1.

First, the distance $dist_{\mathrm{EpDTW}}$ without the temporal character is calculated. Fig. 6.4 shows the distance matrix and the alignment produced.

Sequence $\mathbb{Y}$

Sequence $\mathbb{X}$	**A1**	**L2**	**B3**	**M4**
M4	***1***	2	3	3
B3	2	***2***	2	3
L2	3	2	***3***	3
A1	3	3	3	***4***

(a)



(b)

Fig. 6.4 (a) Cumulative distance matrix for $\mathbb{X}$ and $\mathbb{Y}$. (b) Temporal alignment deduced from the obtained path.

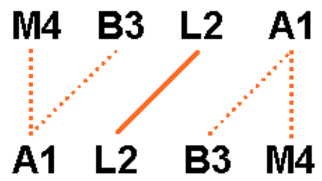
The normalised distance value is obtained by dividing the cell value $\gamma(4,4)$ by the length of the path (k=4), so $dist_{\mathrm{EpDTW}}$ =4/4=1. That is, the sequences are completely dissimilar, as can be observed in the alignment in Fig. 6.4 (b).

On the other hand, Fig. 6.5 shows the distance matrix and the alignment produced when the temporal extension is included. In order to see the effect of the maximum local warping, w_{MAX} has been fixed to 3. On this occasion the temporal alignment produced is more natural and it reflects a certain similarity between the sequences.

Sequence $\mathbb{Y}$

Sequence $\mathbb{X}$	A1	L2	B3	M4
M4	***2***	4	7	7
B3	***3***	4	4	7
L2	4	***3***	5	6
A1	4	4	***4***	***6***

(a)



(b)

Fig. 6.5 (a) Cumulative distance matrix for $\mathbb{X}$ and $\mathbb{Y}$ when the time is considered and w_{MAX} =3.

(b) Temporal alignment deduced from the obtained path.

Again, the normalised distance value is obtained by dividing the cell value $\gamma(4,4)$ by the sum of weights for each cell w_k along the path **W**=[(1,1),(2,1),(3,2),(4,3),(4,4)]. This time the weight needs to be calculated. This operation can be done in parallel with the calculation of the cumulative distance matrix and the weights are stored in a weight matrix (Fig. 6.6). Then, once the path is established, it is easy to add up the individual weights and $\Sigma_{k=1\text{-}5}\mathit{weight}(w_k)$=8. Thus, finally, the new distance $dist_{\mathrm{EpDTW}}$ =6/8=0.75.

Sequence $\mathbb{Y}$

Sequence $\mathbb{X}$	A1	L2	B3	M4
M4	2	2	3	4
B3	1	2	3	3
L2	1	2	2	2
A1	1	1	1	2

Fig. 6.6 Weight matrix.

Fig. 6.7 illustrates the intuitive idea of the maximum local warping w_{MAX} and the weights for each cell w_k. First, the elements are matched according to the path obtained by the cumulative distance matrix (Fig. 6.5). Each matched element contains a number of links equal to its duration. Then, in order to align the sequences, the algorithm tries to compress some elements. The maximum compression is equivalent to when time is not considered. Thus, w_{MAX} limits the compression. In the example, w_{MAX} =3, so 3 links can be compressed to 1 and the result of this procedure for the two sequences is the weight matrix shown in Fig. 6.6.

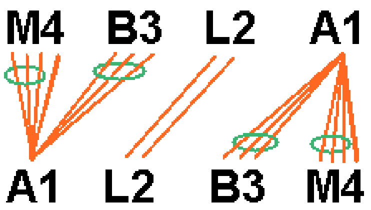


Fig. 6.7 Intuitive significance of the maximum local warping w_{MAX}.

6.3. Application example: Situation assessment in a two tank system

As an application example, the proposed approach has been used in a laboratory plant for situation assessment purposes. In this plant (Fig. 6.8), the level in tank A is controlled by means of a PID controller by pumping water from a reservoir (tank B).

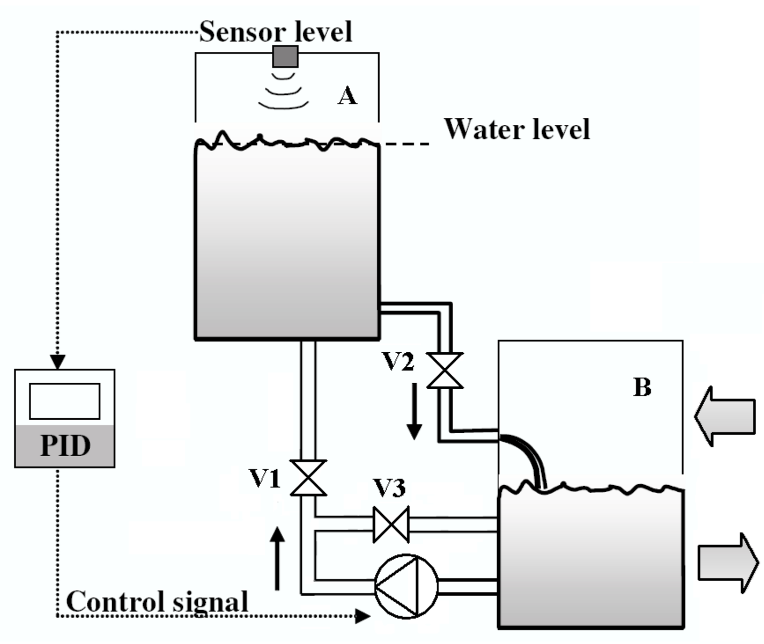


Fig. 6.8 Two tank system.

The level in tank A and the control signal (pump) are the monitored process variables. Three valves (V1, V2 and V3) can be manipulated in order to simulate obstructions and leakages. Then, several situations are possible by the appropriate combination of opening and closing valves (Table 6.3). Additionally, the system dynamics can be modified slightly by filling or emptying the reservoir with external water. Then, the input and output of external water also creates interesting situations to be detected. The experiments have been developed under the assumption that two situations cannot be overlapped. Thus, changes in the configuration of valves are only performed when the process is in a steady state.

Table 6.3 List of situations as a result of the combination valve.

Description	V1	V2	V3
Normal Behaviour	OPEN	OPEN	CLOSED
Obstruction in pump or input pipe	CLOSED	OPEN	CLOSED
Obstruction in the output pipe	OPEN	CLOSED	CLOSED
Leak in input pipe or pump	OPEN	OPEN	OPEN

The monitoring system will be able to assess such situations and diagnose the origin of abnormal behaviour according to the behaviour of measured signals described by the sequences of episodes. A representation based on a set of 13 types of episodes (Fig. 6.2) is used to represent symptoms in the case definition.

Table 6.4 Local distance between episodes.

	⎤	⎡	⎛	∩	⎞	\	−	/	⎝	∪	⎠	⎦	⎣
⎤	0	.72	.85	.7	.62	.67	.75	.9	.8	.87	.95	1	.67
⎡	.72	0	.7	.62	.77	.82	.75	.75	.87	.95	.87	.67	1
⎛	.85	.7	0	.52	.8	.85	.6	.27	.9	.82	.65	.8	.87
∩	.7	.62	.52	0	.45	.6	.6	.6	.82	.9	.82	.87	.95
⎞	.62	.77	.8	.45	0	.27	.6	.85	.65	.82	.9	.95	.87
\	.67	.82	.85	.6	.27	0	.55	.8	.27	.6	.85	.9	.75
−	.75	.75	.6	.6	.6	.55	0	.55	.6	.6	.6	.75	.75
/	.9	.75	.27	.6	.85	.8	.55	0	.85	.6	.27	.67	.82
⎝	.8	.87	.9	.82	.65	.27	.6	.85	0	.4	.8	.85	.7
∪	.87	.95	.82	.9	.82	.6	.6	.6	.4	0	.45	.7	.62
⎠	.95	.87	.65	.82	.9	.85	.6	.27	.8	.45	0	.57	.77
⎦	1	.67	.8	.87	.95	.9	.75	.67	.85	.7	.57	0	.72
⎣	.67	1	.87	.95	.87	.75	.75	.82	.7	.62	.77	.72	0

A major benefit of this set of episodes for situation assessment is that discontinuities and stability periods (usual in abnormal and in normal situations respectively) are explicitly represented by means of 5 types of episodes (⌉ ⌈—⌋ ⌊). Local distances between episodes are defined in Table 6.4. Note that these distances are different to the ones in Table 6.1. The difference in local distances between types of episodes does not affect the global measure of similarity too much, although it allows a finer measure to be obtained. The main difference resides in defining a major or minor distance between those episodes with different behaviour.

A test case base has been built by obtaining the sequence of episodes for the two monitored variables (Fig. 6.9) in a time window of 70 seconds. After testing the most common situations, the case base is composed of 29 cases (Table 6.5) with the description of the situation. This information associated with faults is structured according to three criteria: the part of plant or operation that is being affected, the affected component of the plant and the corresponding diagnosis. This is useful in the final decision about the situation in order to distinguish ambiguous situations when multiple diagnoses are retrieved.

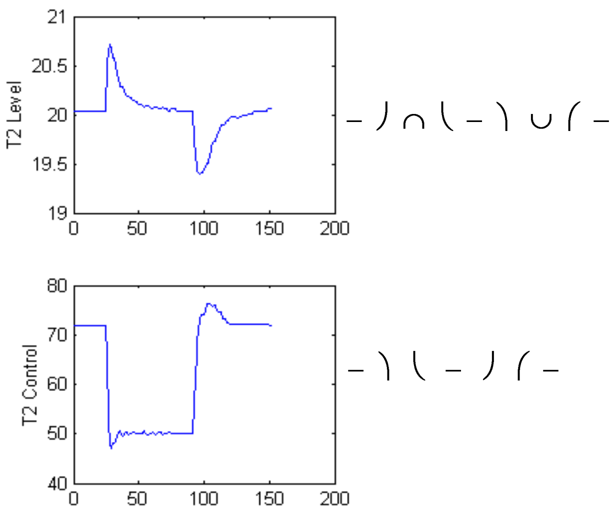


Fig. 6.9 Level and control signals: acquired signal and the corresponding episode-based representation.

In order to test the operability of the EpDTW algorithm, each one of the 29 cases has been compared to the others. Then, 841 similarity measures are carried out taking into account the pattern composed of the level and control signals. The similarity obtained by means of EpDTW gives a normalised value where zero corresponds to identity. Then, similarity between cases is obtained from the average of the similarity for the two signals. From a general point of view, if the 29 cases are analysed by ordering the value of similarity obtained in comparison to the rest of cases, it can be deduced that a threshold of 0.1 allows enough cases for a correct situation assessment to be obtained. Thus, this threshold has been applied to consider the *K*-nearest neighbour symptoms to retrieve.

Table 6.5 The Case Base.

Case	Level	Control	Situation		
			Location	Component	Diagnosis
1	—)\	—)(—	Input tank A	Input pipe or pump	Obstruction
2	—)\(—	—)(—	Input tank A	Input pipe or pump	Obstruction
3	—)(—	—)(—	Input tank A	Input pipe or pump	Obstruction
4	∪(—	—)(—	Input tank A	Input pipe or pump	Obstruction restored
5	—)(—	—)(—	Input tank A	Input pipe or pump	Obstruction restored
6	—)/(—	—)\(—	Input tank A	Input pipe or pump	Obstruction restored
7	∪/(—	—)\(—	Input tank A	Input pipe or pump	Obstruction restored
8	—)∩(—	—)(—	Output tank A	Output pipe	Obstruction
9	—)∩(—	—)\(—	Output tank A	Output pipe	Obstruction
10	—)/∩\(—	—)\(—	Output tank A	Output pipe	Obstruction
11	—)∪(—	—)(—	Output tank A	Output pipe	Obstruction restored
12	—)\∪/(—	—)/(—	Output tank A	Output pipe	Obstruction restored
13	—)∪(—	—)/(—	Output tank A	Output pipe	Obstruction restored
14	—)∪(—	—)/(—	Input tank A	Input pipe or pump	Leakage
15	—)∪—	—)/(—	Input tank A	Input pipe or pump	Leakage
16	—)∪/(	—)/(—	Input tank A	Input pipe or pump	Leakage
17	—)∪(	—)(—	Input tank A	Input pipe or pump	Leakage
18	—)∩\—	—)\(—	Input tank A	Input pipe or pump	Leakage restored
19	—)∩(—	—)(—	Input tank A	Input pipe or pump	Leakage restored
20	)∩\(	)\(—	Input tank A	Input pipe or pump	Leakage restored
21	—)∩(	—)\(—	Input tank A	Input pipe or pump	Leakage restored
22	—	—)	Input tank B	External water	Input of ext. water
23	—	—\	Input tank B	External water	Input of ext. water
24	—	\—	Input tank B	External water	Conclude input
25	—	(—	Input tank B	External water	Conclude input
26	—	—)	Output tank B	External water	Output of ext. water
27	—	—/	Output tank B	External water	Output of ext. water
28	—	(—	Output tank B	External water	Conclude output
29	—	/—	Output tank B	External water	Conclude output

In the following paragraphs, three illustrative examples are presented using this short case base. For this purpose cases 2, 15 and 8 have been used to simulate a new solution (obviously the distance between these cases and themselves is zero). They have been compared to the whole case base and those that have a distance degree inferior to the threshold (0.1) have been retrieved. Table 6.6 to Table 6.8 represent the retrieved cases (first row) and the associated distances (second row).

Table 6.6 Retrieved cases and similarity for case 2.

Case	2	3	1
Distance	0	0.0306	0.0516

In Table 6.6 (Case 2), the diagnosis corresponding to each case is the same and therefore the situation assessment is correct. In the assumption that case 2 is a new situation, the similarity to the registered cases does not make the inclusion of this new case in the case base necessary.

Table 6.7 Retrieved cases and similarity for case 15.

Case	15	13	14	11	16	17	3
Distance	0	0.0333	0.0333	0.0639	0.0639	0.0681	0.0806

Case 15 (Table 6.7) yields 6 cases with a distance inferior to 0.1 as result. The two most similar cases (13 and 14) offer different symptoms, nevertheless, case 13 corresponds to a situation obtained by the restoration of a previous obstruction. Considering that two cases cannot take place simultaneously and that in previous instants an obstruction has not been noticed, cases 13 and 11 can be discarded because they give an inconsistent assessment. Finally, the final selection can be decided by a simple frequency analysis: 3 of 4 cases diagnose an input leakage (pump or pipe) in tank A against one different case. Therefore, this would be the diagnosis.

If previous states are not considered, then cases corresponding to restoration of previous obstructions cannot be deleted. Then, evaluating the frequency, 4/6 indicate problems in the input of tank A, while the remaining 2/ 6 indicate problems in the output. From the cases related to input, all the cases indicate that the failure is located in the pump or pipe. So, this would be the assigned diagnosis, with the probability that the failure is caused by leakage. With this incomplete result the case would be stored in the case base.

Table 6.8 Retrieved cases and similarity for case 8.

Case	8	19	9	18	21	5
Distance	0	0	0.0306	0.0581	0.0639	0.0917

Finally, in the example of case 8 (Table 6.8), it will be supposed that the registered case exists but the previous states are not considered; therefore the cases corresponding to a restoration of a previous failure are not annulled. So, case 8 offers 5 similar (including case 8) cases but with a different situation assessment; one case is even completely identical but with different symptoms. On evaluating frequency, 4/5 of these cases correspond to problems with the input of tank A, while 1/5 correspond to the output. This would indicate that the problems are with the input of tank A, which is an incorrect diagnosis. This only happens if the previous states are not considered, since cases 19 (perfect matching), 18 and 5 correspond to restoration failure.

6.4. Online DTW

As mentioned above, a shortcoming of DTW is its expensive computational cost which normally makes it useful only for offline applications. On the other hand, many real applications demand techniques working online with the process. This has motivated a slight modification of DTW in order to adapt it for online application (Gamero et al., 2004; Llanos et al., 2004).

6.4.1. The algorithm

The new online time warping algorithm aligns two sequences arriving in real time. Mainly, the algorithm calculates the optimal path incrementally in real time as soon as new data is received. The algorithm is initialised by defining a sliding window size w and the warping window width, r. As main particularities, the two sequences have the same length and the new algorithm returns a distance value at each sample time.

The algorithm starts calculating the cumulative distance at time $t = 2$, $D(i,j)$ for each grid element of the squared matrix. Later on, the matrix grows and only cumulative distances for new cells in the matrix are calculated. Next, the matrix reaches the maximum value w and becomes a sliding window. At each sample time, the oldest cells in the matrix are deleted and cumulative distances are calculated for empty cells corresponding to the new sample (Fig. 6.10). Therefore, the algorithm runs more quickly since it uses previously calculated values and the global path constraint *r* restricts the area calculation. Nevertheless, a new path must be found for each window and the distance value is obtained by calculating the total distance according to this new path.

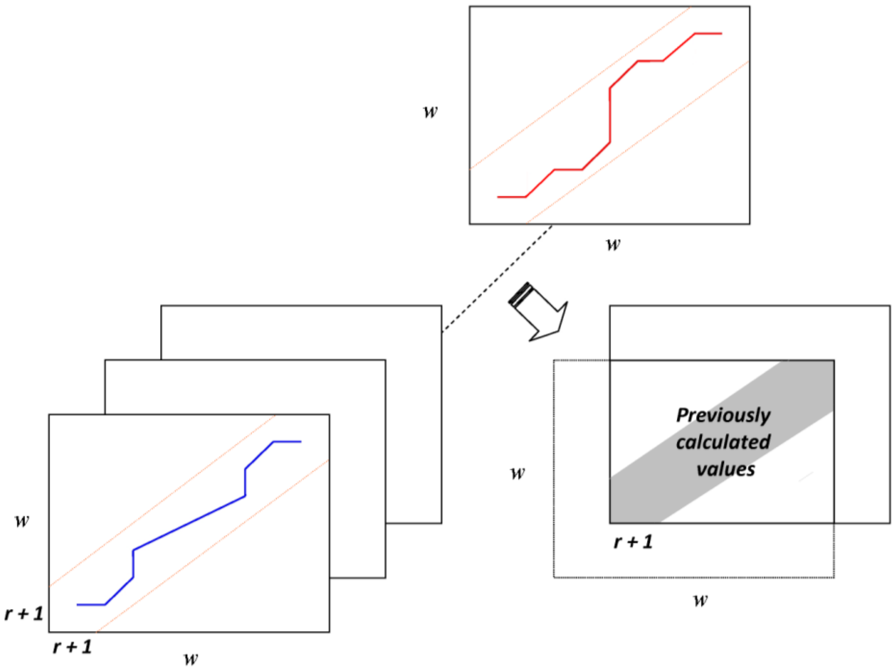


Fig. 6.10 Online DTW process.

6.4.2. Illustrative example

To further illustrate the sliding window technique, consider the following two sequences: $\boldsymbol{x}$ = [2, 5, 2, 5, 2, 1] and $\boldsymbol{y}$ = [0, 3, 6, 0, 6, 0].

First, by applying the dynamic programming algorithm, the cumulative distance matrix is filled as shown in Fig. 6.11. Each value in the cell represents the cumulative distance of that cell. The optimal warping path can be found by tracing backward from the lower right corner, (6,6), towards the upper left corner, (1,1). At each cell, the previous neighbouring cell with minimum cumulative distance is chosen.

Sequence ***y***

Sequence ***x***	0	3	6	0	6	0
2	*2*	*3*	7	9	13	15
5	7	4	*4*	9	10	15
2	9	5	8	*6*	10	12
5	14	7	6	11	*7*	12
2	16	8	10	8	11	*9*
1	17	10	13	9	13	*10*

Fig. 6.11 A cumulative distance matrix for sequences *x* and *y*.

Now the online DTW will be applied to the same sequences ***x*** and ***y***, with the width of the warping windows r = 2 and the sliding window size w=3. Fig. 6.12 shows the cumulative distance at the sample time t = 3, which is when the matrix reaches a maximum value established according to w and it becomes a sliding window. In this particular case r has been defined to fill the entire matrix.

y

x	0	3	6
2	2	3	7
5	7	4	4
2	9	5	8

Fig. 6.12 A cumulative distance matrix for sequences *x* and *y* at the sample time t = 3.

Fig. 6.13 shows the cumulative distance matrix at the sample time t = 4. The sliding windows are overlapping and the coloured grids denote cumulative distances computed previously and which do not have to be computed again. The oldest cells (computed at t = 1) are not taken into account and cumulative distances are calculated just for the new values.

y

x	0	3	6	0
2				
5		4	4	9
2		5	8	6
5		7	6	11

Fig. 6.13 Cumulative distance values of the sliding window for sequences *x* and *y* at time t = 4.

Fig. 6.14 and Fig. 6.15 show the cumulative distance values of the sliding windows at time t = 5 and t = 6, respectively. Note that the cumulative distance $\gamma(6,6) = 10$ is the same distance value as when applying the traditional DTW algorithm. Nevertheless, both methods cannot be compared because DTW needs all the sequences $\boldsymbol{x}$ and $\boldsymbol{y}$ to find an alignment between them, while the online DTW uses subsequences of $\boldsymbol{x}$ and $\boldsymbol{y}$ to find an alignment within the warping window. This is an important factor to be considered since the band global constraint will prevent large deviations from the linear path.

$\boldsymbol{x}$ \ $\boldsymbol{y}$	0	3	6	0	6
2					
5					
2			8	6	10
5			6	11	7
2			10	8	11

Fig. 6.14 Cumulative distance values of the sliding window for sequences *x* and *y* at time t = 5.

$\boldsymbol{x}$ \ $\boldsymbol{y}$	0	3	6	0	6	0
2						
5						
2						
5				11	7	12
2				8	11	9
1				9	13	10

Fig. 6.15 Cumulative distance values of the sliding window for sequences *x* and *y* at time t = 6.

6.4.3. Related work

A few authors have presented their solutions for dealing with the adaptation of DTW to online operation. Some of them present ideas very similar to our algorithm. Some of these approaches are summarised below.

Dixon, 2005 presents an online time warping algorithm which aligns a sequence arriving in real time with a stored sequence. The idea is to define dynamically the search band. The algorithm is initialised by computing a square matrix of size c and the path is calculated using the standard recursion formula. Then the calculated area is iteratively expanded by evaluating rows or columns of length c. The direction of expansion is determined by the location of the cell in the active area with the lowest minimum path cost. Another parameter limits the number of successive computations in the same direction. Finally, when the ends of both files are reached, the optimal path is traced backwards using the standard DTW algorithm, constrained by the cells calculated previously during the forward path calculation.

In Ko et al., 2005 a system using a sliding window is also proposed. First, the distance table is constructed while the data is acquired. In their approach the distance table is composed of a class template (known sequence) and real-time data from sensors. Once the distance table reaches the maximum buffer size it becomes a sliding window. The algorithm uses previously calculated values and calculates a new warping path at each time instance. The main difference with respect to DTW is that this approach proposes the relaxing of the start and end points. The DTW distance is obtained by calculating the cumulative distance along the new warping path.

Capitani and Ciaccia, 2006 propose another online DTW-like distance measure exploiting previously performed computations and using a sliding window. The method, called Stream-DTW (SDTW), computes the measure by obtaining the cost from different warping paths in which boundary conditions were relaxed. The SDTW distance lower bounds the DTW distance and it shows good accuracy in a streaming environment.

In Rajshekhar et al., 2007 a method for fault diagnosis employing an online variation of DTW is presented. The idea is very simple: they select a window from the current sequence of measured variables to compare it with several windows from the reference dataset. The vast number of operations is minimised by using a lower-bounding measure and the similarity measurement is based on a probability function. The method was tested in two simulated case studies with promising classification accuracy but with late detection times.

6.5. Application example: Residual computation in a three tank system

6.5.1. Motivation

Fault Detection and Isolation (FDI) methods based on analytical redundancy (Chow and Willsky, 1984) are widely used to diagnose systems for which a mathematical model is available. The task of FDI is typically accomplished in two steps, namely residual generation and residual evaluation. A residual is a signal generated from a computation based on measured variables. It is ideally zero in the fault-free case and different from zero in the faulty case. In practice, the generated residuals are not identically zero, due to various errors (measurement noises, modelling uncertainties).

Residual generation consists of designing fault indicators to satisfy specifications such as sensitivity to faults and robustness to disturbance, modelling errors and noise. Commonly, residuals are analytical symptoms (Blanke et al., 2006).

Residual evaluation (known also as decision procedure) consists of translating the symptoms into information about the faults that may have occurred.

A possible structure for fault detection is presented in Fig. 6.16. Residual r is obtained as a result of comparison of the model output y_M to real process measure y.

This example uses the online DTW presented above in order to carry out the residual computation and evaluation since DTW is especially suitable for those errors related to time distortions.

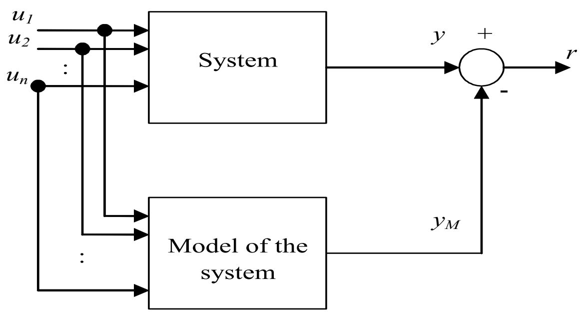


Fig. 6.16 Diagram of residual generation.

6.5.2. The laboratory plant

In order to illustrate the proposed method a laboratory plant was used to test this approach.

To illustrate the proposed method, consider the laboratory plant depicted in Fig. 6.17. Three tanks are connected by pipes which can be controlled by several valves. The main aim of the two tanks (TANK 2 and TANK 3) is to provide a continuous water flow "Q_N" to a consumer. The water level in TANK 2 has, therefore, to be maintained at a constant level. TANK 3 is filled by PUMP 2 up to a nominal water level. Water flowing between the tanks can be controlled by on/off valves V9 and V10. Valves V9 and V10 are used to control the water level in TANK 2. Valve VR2, which can be used to simulate a leakage in TANK 3, is normally closed. Valve VR1, which can be used to simulate an obstruction between TANK 2 and the consumer, is normally open.

The measurements available from the process are the continuous water levels (level in TANK 2 "h_2" measured by SENSOR 2 and level in TANK 3 "h_3" measured by SENSOR 3) and control signal "P" of PUMP 2. The level in TANK 3 is controlled by means of a PID regulator by pumping water from the reservoir (TANK 1).

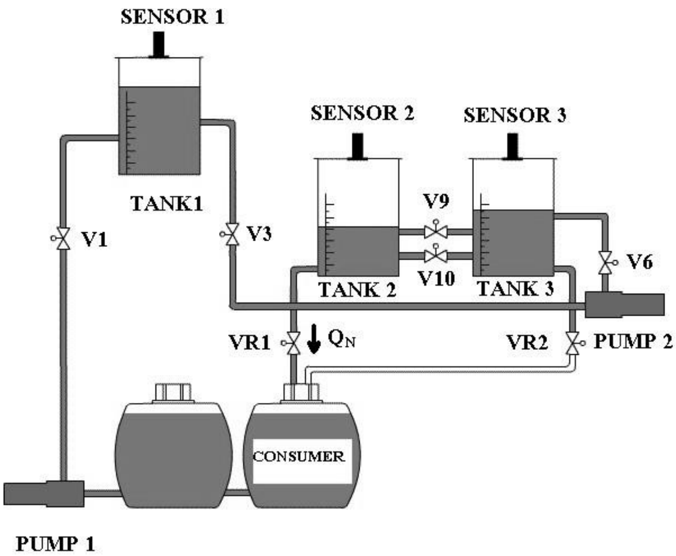


Fig. 6.17 The laboratory plant.

The process variables available are used to simulate system behaviour and the following models can be used for fault diagnostic purposes:

- level in TANK 3,

$$\hat{h}_3 = f(SPh_3)$$

where, SPh_3 is the level set point of h_3.

- level in TANK 2,

$$\hat{h}_2 = f(SPh_2)$$

where SPh_2 is the level set point of h_2.

- control signal of pump 2,

$$\hat{P} = f(SPh_3, \hat{h}_3)$$

Based on simulated model outputs, the residuals are calculated as a difference (by using online DTW) between the process variable and corresponding simulated variables. So the following residuals are generated as:

$$r_1 = DTW_{ONLINE}(h_3, \hat{h}_3)$$

$$r_2 = DTW_{ONLINE}(h_2, \hat{h}_2)$$

$$r_3 = DTW_{ONLINE}(P, \hat{P})$$

The laboratory plant has been modelled with Bond Graph and simulated with Matlab-Simulink.

Six different fault scenarios can be considered (Table 6.9). Situations when more than one of these faults happen simultaneously are not considered. Next, the advantages and disadvantages of the proposed method are summarised.

Table 6.9. Faults and their corresponding description.

Fault	Description
F_1	Leakage in tank 3, by opening VR2
F_2	Blockage of valve V10 in closed position.
F_3	Blockage of valve V10 in open position.
F_4	Pump 2 blockages 20% of its capacity.
F_5	Pump 2 blockages 80% of its capacity.
F_6	Obstruction between TANK 2 and the consumer, by closing VR1.

6.5.3. The residual generation and evaluation approach

This example uses a method based on thresholds and fault signatures for residual evaluation. A residual will be coherent with the model of the system if it is zero or less than the chosen thresholds. The coherence of each residual is tested using a direct comparison between its value and thresholds [ε,-ε]. This test is applied to the set of residuals r_i leading to a vector S = [s_1, s_2, s_3,..., s_n]. Each component s_i of S is obtained using the following rule:

$$s_i = \begin{cases} 1 & \text{if} \quad r_i > \varepsilon \\ 0 & \text{if} \quad -\varepsilon \le r_i \le \varepsilon \\ -1 & \text{if} \quad r_i < -\varepsilon \end{cases} \qquad [6.8]$$

The aim of the isolation step is to provide a list of elements which are failing. The decision procedure works as follows: the residual is computed first and then checked to see if it is not different from zero. If the residual is stochastic then thresholds have to be determined inside which the residual occurs. The recognition procedure compares the vector S with all the signatures and determines the corresponding fault.

First, the three available measurements are studied under normal situation for a threshold definition. In this example, the threshold ε is defined as 3σ, where σ is the standard deviation observed under the faultless mode. Other settings are the sliding window W=40 and the warping window size r=10.

Fig. 6.18 to Fig. 6.20 show system measures and simulated variables under normal conditions. Dotted lines show simulation of the system; the misalignment between both signals can be seen. The measures have been normalised between 0 and 1. The residuals computed by using Euclidean distance and DTW_{ONLINE} are shown below the signals.

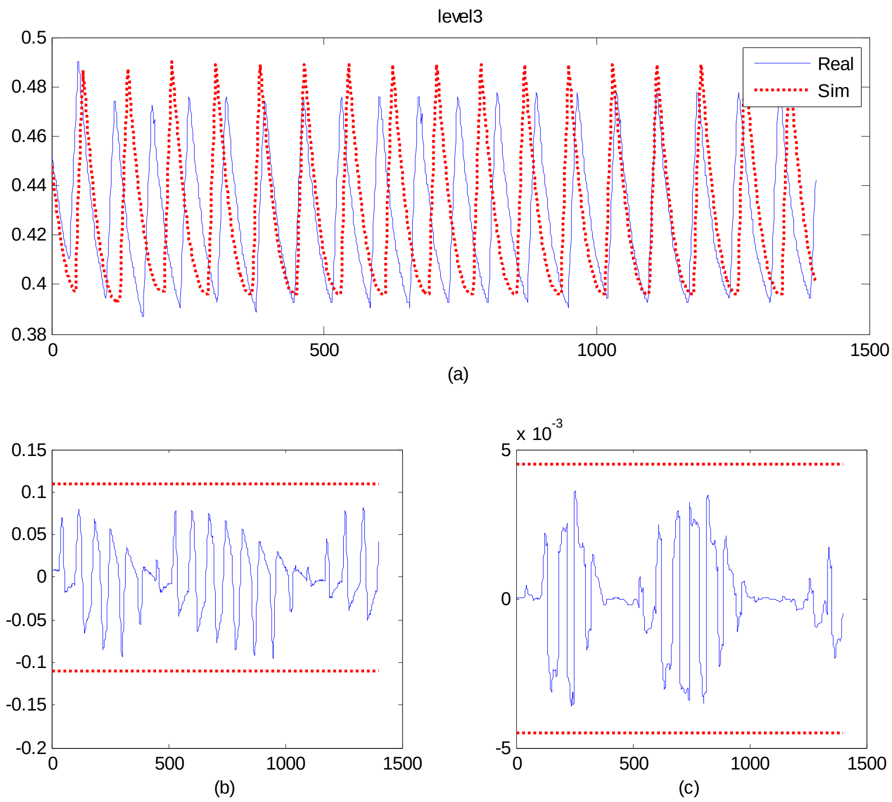


Fig. 6.18 (a) Real and simulated level in tank 3, system in normal operation. (b) Residual r_1 obtained using Euclidean distance. (c) Residual r_1 using DTW_{ONLINE}.

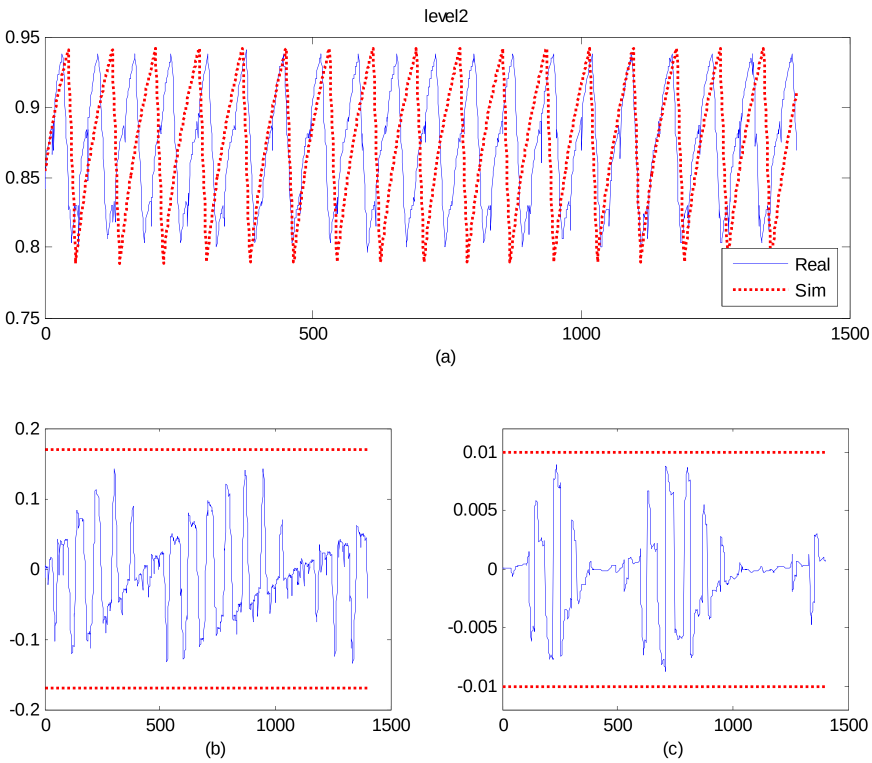


Fig. 6.19 (a) Real and simulated level in tank 2, system in normal operation. (b) Residual r_2 obtained using Euclidean distance. (c) Residual r_2 using DTW_{ONLINE}.

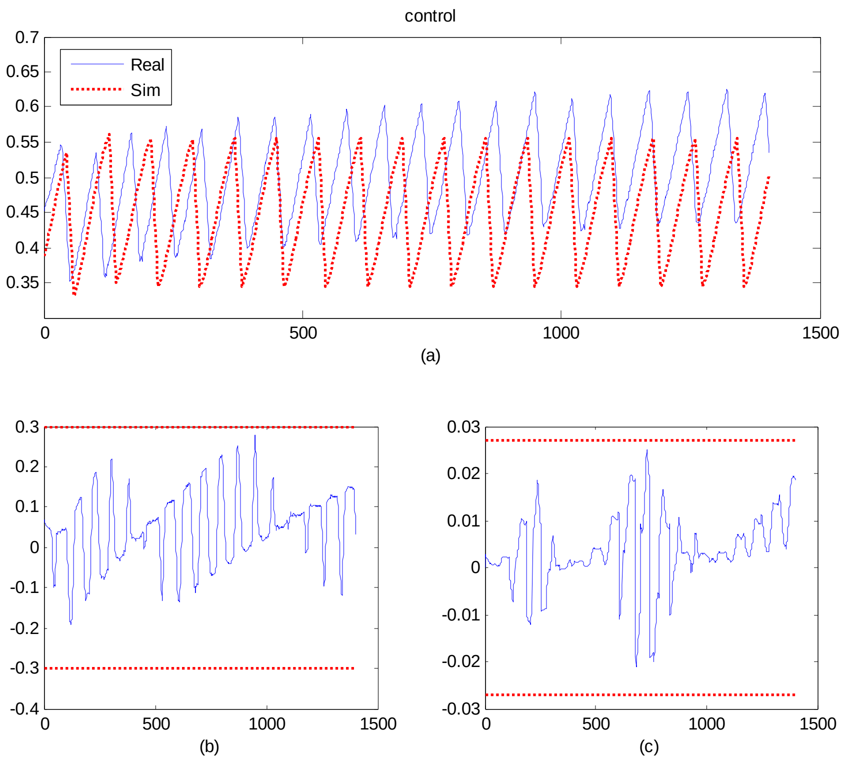


Fig. 6.20 (a) Real and simulated control signal of pump, system in normal operation. (b) Residual r_3 obtained using Euclidean distance. (c) Residual r_3 using DTW_{ONLINE}.

Then, through the application of $DTW_{ONLINE,}$ a set of residuals is obtained and a simple fault signature table can be easily built for fault isolation (Table 6.10). In this table a "±1" denotes that the residual is sensitive to the fault. The sign of the distance value is added according to the sign of the difference between the two signals. Thus, a recognition procedure determines the corresponding fault according to the table. Note that all the faults are perfectly isolable within the selected thresholds. Only the blockage of V10 in the open position (F3) may not be detected by either method.

Table 6.10 Residual structure.

	$s1$	$s2$	$s3$
F0	0	0	0
F1	-1	0	+1
F2	+1	-1	-1
F3	0	0	0
F4	-1	0	-1
F5	+1	0	+1
F6	+1	+1	-1

Finally, three faulty scenarios are analysed. Fig. 6.21 and Fig. 6.22 show the level in tank 3 and the respective control signal for fault $F4$. This fault consists of a blockage in pump 2 of 20% of its capacity. The residual r_1 (Fig. 6.21c) computed by DTW_{ONLINE} shows a higher robustness than the residual calculated by means of the difference (Fig. 6.21b). In fact, the results obtained showed fewer false alarms than when using DTW. In addition, the normal residual r_3 (Fig. 6.22b) manifests an unstable state when the fault occurs. Note that the residual goes above and under the threshold several times during the fault duration. This behaviour could be interpreted as a false alarm. As can be observed, the main problem is originated by the misalignment between both signals and this causes the difference to have oscillatory values.

Fig. 6.23 and Fig. 6.24 show the level in tank 3 and the control signal respectively for fault $F1$. This fault consists of a leakage in tank 3. Note that the residuals computed by DTW_{ONLINE} during the fault have the opposite sign as in Table 6.10 (S=[-1, 0,+1]).

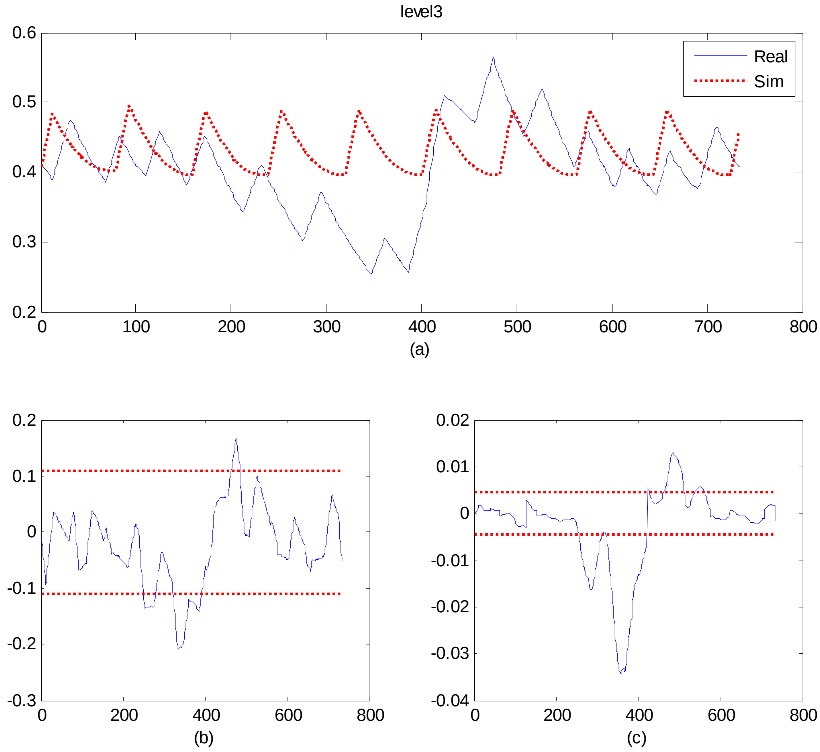


Fig. 6.21 (a) Real and simulated level of tank 3, faultless mode F4. (b) Residual r_1 obtained using Euclidean distance. (c) Residual r_1 using DTW$_{\text{ONLINE}}$.

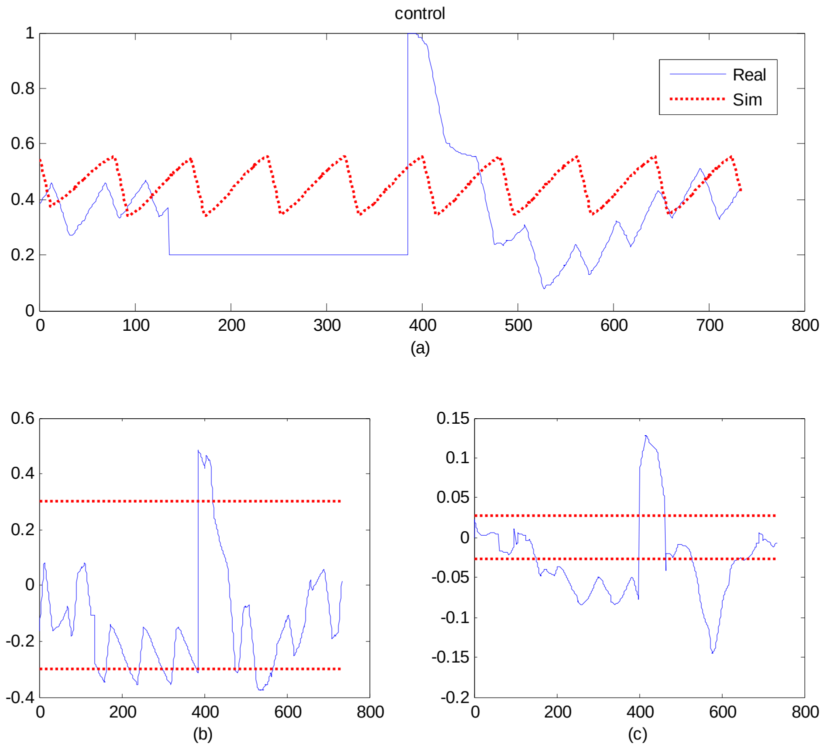


Fig. 6.22 (a) Real and simulated control signal of pump, faultless mode F4. (b) Residual r_3 obtained using Euclidean distance. (c) Residual r_3 using DTW$_{\text{ONLINE}}$.

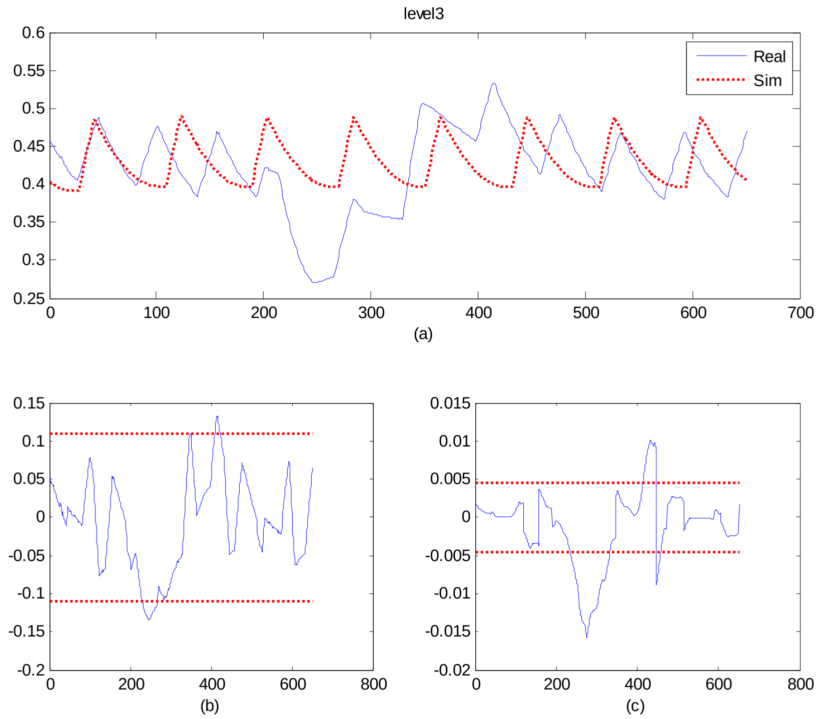


Fig. 6.23 (a) Real and simulated level of tank 3, faultless mode F1. (b) Residual r_1 obtained using Euclidean distance. (c) Residual r_1 using DTW$_{ONLINE}$.

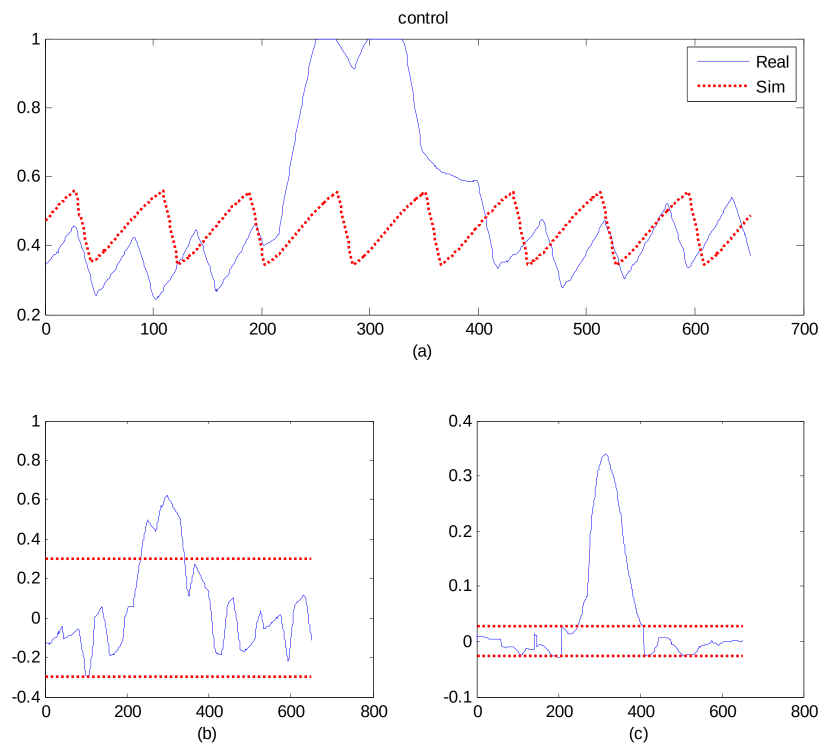


Fig. 6.24 (a) Real and simulated control signal of pump, faultless mode F1. (b) Residual r_3 obtained using Euclidean distance. (c) Residual r_3 using DTW$_{ONLINE}$.

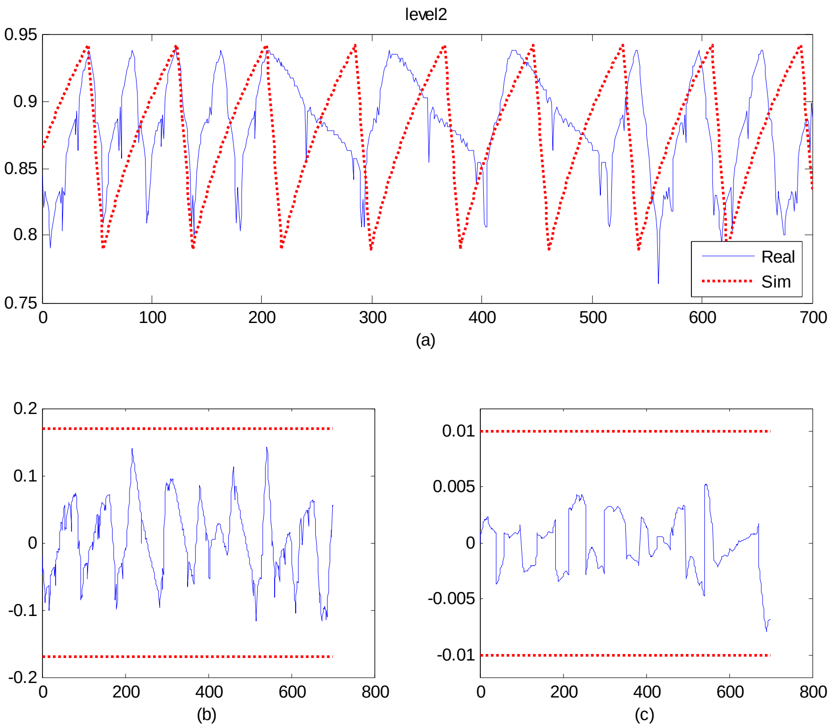


Fig. 6.25 (a) Real and simulated level of tank 2, faultless mode F3. (b) Residual r_2 obtained using Euclidean distance. (c) Residual r_2 using $\mathrm{DTW}_{\mathrm{ONLINE}}$.

The last case analysed corresponds to fault *F*1. Fig. 6.25 shows the level in tank 2 as a representative signal. This fault could not be detected by any residual. However, the real signal presents visible abnormal behaviour during the fault duration; note that a frequency variation is notably in evidence. Thus, further work should consider some variation of DTW as the Derivative Dynamic Time Warping (DDTW) as proposed in Keogh and Pazzani, 2001 or episodes representing different slopes. Therefore, the information about shape, as the first derivative, could be used to improve the $\mathrm{DTW}_{\mathrm{ONLINE}}$ algorithm.

6.6. Conclusions

In this chapter, two modifications of the DTW algorithm have been introduced. The first one consists of operating, not in the original time series, but in its episode-based representations. Since different patterns belonging to the same class can have different time duration or magnitudes, a combination of DTW and qualitative representations tries to solve the problem of the variability in the Y-axis. So, the advantage of the temporal alignment produced by DTW is added to the advantage of the representation in episodes, which takes into account the behaviour of the signals. In addition, qualitative representations solve the problem of the lengthy computing time when dynamic programming is used.

The utility of the new algorithm is shown in several examples and the diagnosis of a level control system where the correct identification of operating situations is obtained through the comparison of the current pattern with well-known reference patterns.

The second modification proposes a slight variation of DTW in order to adapt it for online application. In the way that the algorithm has been defined, this works only for a priori unknown signals that evolve at the same time. Nevertheless, the same idea could be applied if one of the sequences is a known reference sequence. Important factors to be considered are the window size and the band global constraint. The first could produce a filtering effect, for example: identical but shorter events can be missed if the window size is much greater than event duration and the second will prevent large deviations from the linear path.

As an example, the approach has been used in order to improve the residual computation from a laboratory plant. Since this approach is especially suitable for those errors related to time distortions, it will be useful for distributed systems with communication delays and for hybrid systems with on/off sensors or actuators causing misalignments between real and simulated signals.

Both algorithms allow the DTW algorithm to be speeded up, but there are still two significant drawbacks. First, all the elements in the alignment must be matched, contributing to the final distance. And second, the endpoint constraints require that the warping path start and finish in diagonally opposite corner cells of the distance table.

The solution involves developing other algorithms in which these features will not occur. Methods based on traces meet these specifications and will be the subject of the next chapter.

Chapter 7.

A similarity index for qualitative sequences

7.1. Introduction

In the previous chapter a variation of DTW which allows working with episodes was presented. Although the results obtained in the application example are promising, methods based on DTW have some drawbacks. First, DTW matches all the elements in the sequences, even those that are not similar. Taking into consideration that a large portion of the sequences may be just outliers, the similarity function must be robust under noisy conditions and it should not match the incorrect parts. Secondly, DTW is a method of global alignment, no matter how different the sequences may be. This is a reasonable approach only for sequences that are related over their entire length.

The aim of this thesis is the interpretation of continuous signals involved in process monitoring as sequences of qualitative representations. The objective is to perform situation assessment by comparing current trends based on episodes with other well known patterns, since it is possible to associate the latter with certain process states or symptoms. Thus, a whole sequence can represent a particular class, while the length of the current sequence extracted from the process is arbitrary and could represent more than one class, or even none. Then, the classification by means of a similarity measure is difficult when methods that return a global similarity are used.

This chapter is focused on defining and implementing a new normalised index (QSSI) related to the degree of similarity between qualitative trend signals. In contrast to DTW, the proposed algorithm only matches similar items thus allowing greater flexibility and the alignment has been formulated in terms of similarity instead of distance between sequences. Moreover, the QSSI algorithm does not align both sequences over their full length.

Following the presentation of the QSSI similarity principle, two application examples will be presented. The first example shows a methodology for classification of voltage sags according to their origin in the power system. The second example describes an approach for online monitoring and situation assessment applied to a steam generator, using online data from sensors providing both quantitative and qualitative information.

7.2. Qualitative Sequence Similarity Index (QSSI)

The methods for computing similarity in the domain of strings over finite alphabets are the basis of the similarity index presented here. A way of representing time sequences is by means of an alphabet, which is often characteristic of the application. This conversion of time series into qualitative sequences implies a discretisation in both time (resampling) and magnitude (alphabet). Thus, each pattern is represented by a string of episodes identified by means of a pattern grammar. Then, the sequence comparison problem is reduced to quantifying the degree of similarity or, equivalently, the distance between qualitative sequences. To do this, however, the sequences to be compared must first be aligned.

An algorithm for sequence alignment will, in general, attempt to identify regions of high similarity by maximising a certain score that quantifies the similarity between the sequences in an optimal (or suboptimal) alignment. Most of the alignment algorithms are based on the technique of dynamic programming (Gusfield, 1997). These algorithms search optimal solutions for a given scoring or cost function. Usually, the scoring function is based on the definition of a set of operations, basically insertions, deletions and substitutions, with a certain cost associated with each one. The algorithm searches for the optimal alignment based on the application of these operations. Next, a criterion is needed to judge whether the two sequences share a sufficient degree of similarity. However, the majority of alignment algorithms lack the normalisation that would appropriately rate the obtained score with respect to the length of the sequences or would serve as a reference index for quantifying the significance of the comparison. The proposed method, called Qualitative Sequence Similarity Index (QSSI), represents a normalised measure of the maximum similarity score between two qualitative sequences, $\boldsymbol{S}$ and $\boldsymbol{Q}$. The analysis of similarity between two sequences has been defined in terms of traces (Sankoff and Kruskal, 1983). A trace from $\boldsymbol{S}$ to $\boldsymbol{Q}$ consists of connecting the sequences, usually with a set of links between the elements that exist in both, *S* and *Q* (Fig. 7.1). Each pair of elements connected to one of these links constitutes a match according to the terminology used by this author. The following subsections are devoted to explaining the algorithm in detail while an example is shown in Appendix B.

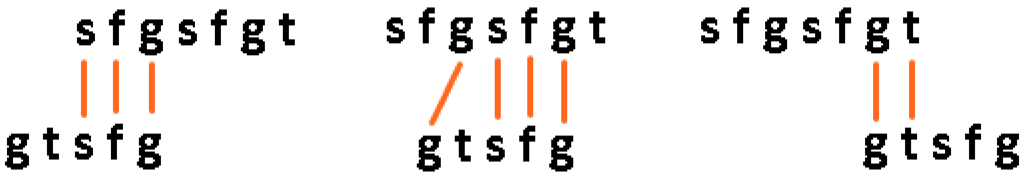


Fig. 7.1 Three possible traces for the same sequence.

7.2.1. Description of the basic algorithm

This section will explain the operation of the basic algorithm when is applied to a series of symbols. Later, the description will be expanded by the introduction of episodes. The idea behind the approach is to build up an optimal temporal alignment between two time-dependent qualitative sequences and then to calculate the similarity using the total score of the aligned elements. The QSSI will be performed in three stages:

- Obtaining matches (pairs of elements).
- Retracing the optimal path.
- Minimisation of the temporal misalignment, that is, the selection of the optimal trace (with the highest score).
- Calculate the normalised similarity index.

Obtaining matches

Given two symbol sequences ***Q*** and ***S***, with lengths *m* and *n* respectively:

$$\boldsymbol{Q} = \langle q_1, q_2, \ldots, q_i, \ldots, q_m \rangle \qquad \boldsymbol{S} = \langle s_1, s_2, \ldots, s_j, \ldots, s_n \rangle \qquad [7.1]$$

The smallest unit of comparison is a pair of elements, one from each sequence. The maximum match can be defined as the largest number of elements from one sequence that can be matched to those of another sequence. If the sequences are composed of temporal series the maximum match represents a temporal alignment, which is determined by representing all the possible pair combinations that can be constructed from the two sequences in a two-dimensional array, or alignment matrix **M** (Fig. 7.2). So, the alignment matrix is initialised first by scoring cells with similar pairs.

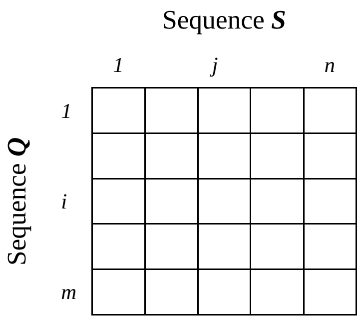


Fig. 7.2 Arrangement of the sequences *S* and *Q* in the alignment matrix.

In the simplest method, the score of each pair combination is defined by using a similarity (or scoring) matrix containing the distances (or similarities) between all the existing elements. The sophistication of the comparison can be increased if each cell value is defined as a function of the properties of the elements, such as time duration and other qualitative values (see application examples).

The value of each cell in the alignment matrix is calculated by the following recurrence:

$$\gamma(i,j)=\begin{cases}\text{if } (i{=}\text{m} \;\&\; j{=}\text{n}): \\ \qquad \delta(\text{q}_m,\text{s}_n) \\ \text{if } (i{<}\text{m} \;\&\; j{=}\text{n}) \\ \qquad \begin{pmatrix}\delta(\text{q}_i,\text{s}_n)+\gamma(i+1,n) & \text{if} & \delta(\text{q}_i,\text{s}_n)>0 \\ \gamma(i+1,n)-1 & \text{if} & \delta(\text{q}_i,\text{s}_n)=0\end{pmatrix} \\ \text{if } (i{=}\text{m} \;\&\; j{<}\text{n}): \\ \qquad \begin{pmatrix}\delta(\text{q}_m,\text{s}_j)+\gamma(m,j+1) & \text{if} & \delta(\text{q}_m,\text{s}_j)>0 \\ \gamma(m,j+1)-1 & \text{if} & \delta(\text{q}_m,\text{s}_j)=0\end{pmatrix} \\ \\ \textit{otherwise}: \\ \qquad \delta(\text{q}_i,\text{s}_j)+ \\ \qquad \max\begin{pmatrix}\delta(\text{q}_i,\text{s}_j)+\gamma(i+1,j+1) & & \\ \gamma(i,j+1)-1 & \text{if} & \mathbf{Md}(i,j+1)\neq 3 \\ \gamma(i+1,j)-1 & \text{if} & \mathbf{Md}(i+1,j)\neq 2\end{pmatrix}\end{cases} \qquad [7.2]$$

In the equation, $\delta(q_i, s_j)$ is the score given for matching the i^{th} symbol in string $\boldsymbol{Q}$ with the j^{th} symbol in string $\boldsymbol{S}$. The matrix is filled in starting at the end of the sequences at position (m,n). Thus, the three first lines represent the initialisation of the matrix, starting with the last column ($1 \le i \le m$ and $j=n$) and the last row ($i=m$ and $1 \le j \le n$).

In parallel to the alignment matrix, a direction matrix **Md,** containing directions of propagation, is constructed (see the example in Appendix B). This is equivalent to recording for each node (i,j) from which direction v the calculation of the optimal alignment score has been propagated from (Fig. 7.3). For practical realisations of the algorithm, saving directions is usually the preferred method since it requires no additional computations which reduce both the necessary execution time and overall complexity of the algorithm.

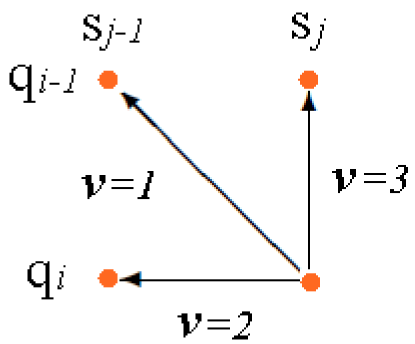


Fig. 7.3 The encoded variable *v* specifies the direction of propagation.

Furthermore, the maximum values obtained from eq. [7.2] must meet the permitted directions in order to not violate the “N”-shape configuration (see Fig. 7.4). This constraint only affects the last two lines in the recurrence formula since they can produce an “N”-shape. Thus, cell $\gamma(i,j+1)$ can only be added if it comes from direction v=1 or v=2. Similarly, cell $\gamma(i+1,j)$ can only be added if it comes from direction v=1 or v=3.

Retracing the optimal path

The optimal temporal alignment is represented by a pathway **W** of length K through the alignment matrix,

$$\mathbf{W} = w_1, w_2, \cdots, w_k, \cdots, w_K \qquad [7.3]$$

where $w_k = (i_k, j_k)$, and i_k and j_k denote the time index of each element in $\boldsymbol{Q}$ and $\boldsymbol{S}$ respectively.

To find pathway **W**, the alignment matrix is first filled in according to eq. [7.2]. Then, the optimal matched alignments forming the path **W** are available by a backtrace through the alignment matrix. The path starts from the cell $w_1 = (i_1, j_1)$ with the highest score and, guided by the recorded directions of propagation, it is completed until reaching the final element (i_k, j_k) with the lowest score. The path **W** will consist of those pairs with similarity between their elements.

If there are two or more possible pathways, the algorithm chooses the one which maintains the bigger proportion of matches for the maximum number of episodes of each sequence. The path obtained complies with the following properties:

- The pathway is monotonic. That is, $i_{k+1} \geq i_k$ and $j_{k+1} \geq j_k$. An immediate consequence is that the lines from each pair of elements cannot cross other links. In this way any relationship representing permutations is avoided, since this destroys the physical significance of a sequence.
- An "N"-shaped configuration is not allowed. If a term has multiple lines, the terms at the other ends of these lines cannot have multiple connections (Fig. 7.4). This is also a common constraint in speech recognition (Sankoff and Kruskal, 1983).
- Multiple connections are allowed. An element can have more than one match (Fig. 7.5).
- The scoring process allows stepping in one element in the same row or column. This is to minimise the introduction of noise episodes between two identical episodes.

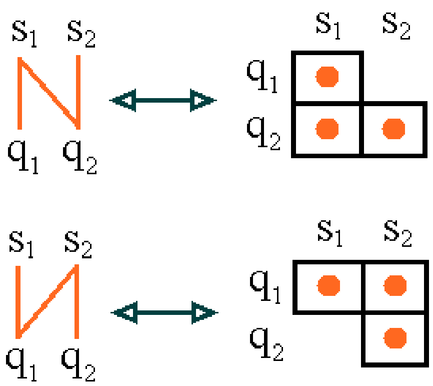


Fig. 7.4 The "N"-shaped constraint and its equivalence in the alignment matrix.

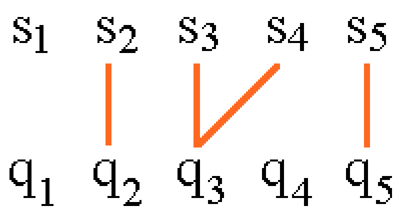


Fig. 7.5 Example of multiple connections.

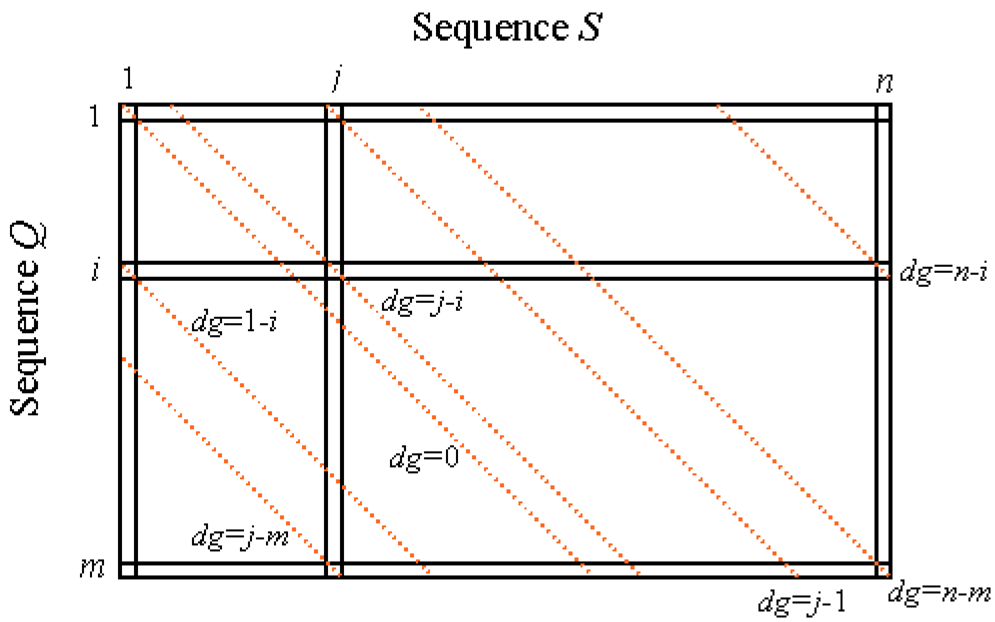


Fig. 7.6 The diagonals are numbered according to the difference *j-i* between the coordinates of the cells on the diagonal.

Minimisation of the temporal misalignment

The next step is to choose the optimal temporal alignment based on an accurate analysis of diagonals. A diagonal *dg* in the alignment matrix is defined as the cells with a position *(i,j)* where *j-i=dg* (Fig. 7.6). All the matches over a diagonal *dg*, are perfectly aligned in time, whereas matches out of the diagonal indicate the presence of a time misalignment. The goal in this step is to identify the diagonal which minimises the temporal misalignment between sequences ***Q*** and ***S***.

Once the path **W** is obtained, the optimal diagonal dg_{opt} is selected by calculating the distance of every match to the diagonal. The diagonal that has the minimum accumulated distance, *f(dg),* is the optimal one. Only the set of diagonals *Dg* containing matches are considered when computing the accumulated distance:

$$dg_{opt} = \underset{dg \in Dg}{\arg\min}\ f(dg) \quad \forall Dg = \left\{ dg \in \left[-(m-1), n-1\right] \ \middle|\ dg = j_k - i_k \right\} \qquad [7.4]$$

with

$$f(dg) = \sum_{k=1}^{K} \left| j_k - i_k - dg \right| \qquad [7.5]$$

A temporal alignment according to diagonals represents the shift in the sequences in either direction to synchronise the matched elements in time. This movement allows a sequence to overlap or enclose the other one, that is, for two sequences ***Q*** and ***S***, with longitude m and n respectively, and $n>m$, the possible $n+m-1$ traces can be aligned as shown in Fig. 7.7.

$$\begin{array}{lllllll}
 & s_1\ s_2\ \ldots\ \ldots\ s_{n-1}\ s_n & & & \\
q_1\ \ldots\ \ldots\ q_m & & & & \\
 & q_1\ \ldots\ \ldots\ q_m & & & \\
 & \qquad\qquad \ldots & & & \\
 & \qquad\qquad\quad q_1\ \ldots\ \ldots\ q_m & & & \\
 & \qquad\qquad\qquad\quad q_1\ \ldots\ \ldots\ q_m & & &
\end{array}$$

Fig. 7.7 Possible temporal alignments of two sequences.

Calculating the similarity index

Once the sequences are aligned in time, the algorithm calculates a normalised value representing similarity. The index of similarity for QSSI is defined as:

$$QSSI(\boldsymbol{Q}, \boldsymbol{S}) = \frac{1}{F_N} \cdot \sum_{k=1}^{K} Sim\left(i_k, j_k\right) \cdot \frac{p(i_k, j_k)}{p_0} \qquad [7.6]$$

where $Sim(i_k,j_k)$ is the local similarity associated to the match k. The function $p(i_k,j_k)$ is a penalty factor motivated by the temporal misalignment and $\boldsymbol{p_0}$ is the maximum value of this function when two characters are perfectly aligned. Finally, F_N is a normalisation factor, so the final value $QSSI(\boldsymbol{Q},\boldsymbol{S})$ is normalised between 0 (completely dissimilar) and 1 (equal). The next subsection addresses the issue of normalisation factor F_N.

The simplest local similarity, *Sim()* function, assigns a value of 1 for identical characters and 0 if they are different. In this case, since the path **W** contains only matches, all values for $Sim(i_k,j_k)$ are 1. Then,

$$QSSI(\boldsymbol{Q},\boldsymbol{S}) = \frac{1}{F_N \cdot p_0} \cdot \sum_{k=1}^{K} p(i_k, j_k) \qquad [7.7]$$

The function $\boldsymbol{p}(i_k,j_k)$ can be designed to deal with the different durations of qualitative symbols according to current requirements. For example, it can assign a lower weight to further matching, which is measured from the optimal diagonal. The extreme case will happen when the time influence is not considered by defining $p(i_k,j_k) = p_0$.

The complete flow diagram of the QSSI algorithm is summarised in Fig. 7.8.

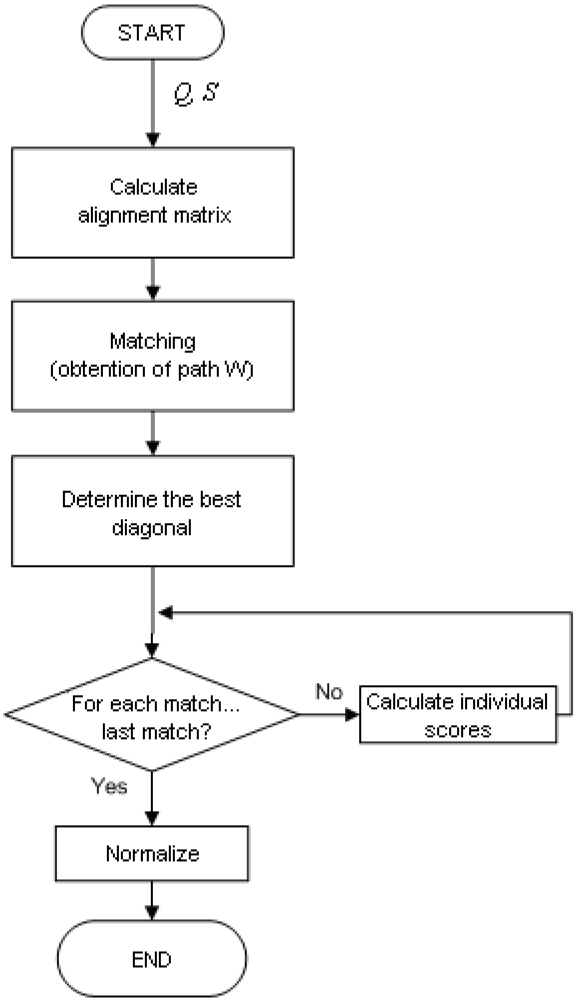


Fig. 7.8 Structure of the QSSI algorithm.

7.2.2. Extending the algorithm to use episodes

So far, the operation of the algorithm has been presented. However, a qualitative representation describing the behaviour of process signals is needed. This representation is performed by episodes.

Formally, given two time series (process measurements) $\boldsymbol{x} = [x_1,x_2,\ldots,x_m]$ and $\boldsymbol{y} = [y_1,y_2,\ldots,y_n]$, they are converted to sequences of episodes $\mathbb{X} = \langle \mathbb{X}_1,\ldots, \mathbb{X}_i,\ldots, \mathbb{X}_M\rangle$ and $\mathbb{Y} = \langle \mathbb{Y}_1,\ldots, \mathbb{Y}_j,\ldots, \mathbb{Y}_N\rangle$ where $M<<m$ and $N<<n$. The indices i and j represent the relative position of each episode in the sequence ($1\le i\le M$, $1\le j\le N$). According to section 3.2.4, an episode is defined as the set:

$$\mathbb{X}_i = \left\langle QS^i, d^i, \text{ auxiliary characteristics } \right\rangle \qquad [7.8]$$

The main problem when dealing with episodes is their duration (d^i). For example, a pressure signal increases for 1 minute, then becomes steady for 5 minutes and finally increases again for 10 minutes. Are these rising episodes similar or different according to their duration? The answer is not simple and most times it is application dependent.

An important cause of temporal misalignment is the different length of episodes. Consequently the similarity between sequences could be altered. In practical application, the duration of episodes acts as a weight in order to align sequences by shifting the diagonal. In QSSI a pre-processing of durations will allow the sequences to be compared without considering duration or dealing with time strictly. Thus, the effective length d_i' shall be determined by applying equation [7.9], where d^i is the current length of episode $\mathbb{X}_i$ and κ a length reduction coefficient between 0 and 1.

$$\mathrm{d}_i' = (d^i)^\kappa \qquad [7.9]$$

If κ=0, then the effective length is 1 for each episode. This value is used if the user does not give importance to time. The opposite value, κ=1, means that current lengths are very important and so the duration of episodes remains unchanged. Between these values time is compressed exponentially, reducing more significantly longer episodes in relation to the shorter ones. Therefore, equation [7.9] allows the importance of time to be adjusted to each application.

Another factor to keep in mind is that some matched elements have more weight than others because of their duration. This must be reflected in the construction of the alignment matrix. Thus, the initial equation [7.2] has been slightly modified as shown in eq. [7.10].

$$\gamma(i, j) = \begin{cases} \text{if } (i{=}M \;\; \& \; j{=}N): \\ \qquad \delta_d(\mathbb{X}_M, \mathbb{Y}_N) \\ \text{if } (i{<}M \;\; \& \; j{=}N) \\ \qquad \begin{pmatrix} \delta_d(\mathbb{X}_i, \mathbb{Y}_N) + \gamma(i+1, N) & \text{if} & \delta_d(\mathbb{X}_i, \mathbb{Y}_N) > 0 \\ \gamma(i+1, N) - g & \text{if} & \delta_d(\mathbb{X}_i, \mathbb{Y}_N) = 0 \end{pmatrix} \\ \text{if } (i{=}M \;\; \& \; j{<}N): \\ \qquad \begin{pmatrix} \delta_d(\mathbb{X}_M, \mathbb{Y}_j) + \gamma(M, j+1) & \text{if} & \delta_d(\mathbb{X}_M, \mathbb{Y}_j) > 0 \\ \gamma(M, j+1) - g & \text{if} & \delta_d(\mathbb{X}_M, \mathbb{Y}_j) = 0 \end{pmatrix} \\ \\ \text{otherwise:} \\ \qquad \delta_d(\mathbb{X}_i, \mathbb{Y}_j) + \\ \qquad \max \begin{pmatrix} \delta_d(\mathbb{X}_i, \mathbb{Y}_j) + \gamma(i+1, j+1) & & \\ \gamma(i, j+1) - g & \text{if} & \mathbf{Md}(i, j+1) \neq 3 \\ \gamma(i+1, j) - g & \text{if} & \mathbf{Md}(i+1, j) \neq 2 \end{pmatrix} \end{cases} \quad [7.10]$$

Where the new score $\delta_d(\mathbb{X}_i, \mathbb{Y}_j)$ used in eq. [7.10] is calculated as:

$$\delta_d(\mathbb{X}_i, \mathbb{Y}_j) = \delta(\mathbb{X}_i, \mathbb{Y}_j) \cdot g \quad [7.11]$$

and

$$g = \min(d_i', d_j') \quad [7.12]$$

The real value g also penalises vertical and horizontal movements, while the score $\delta(\mathbb{X}_i, \mathbb{Y}_j)$ given for matching the i^{th} episode in sequence $\mathbb{X}$ with the j^{th} episode in sequence $\mathbb{Y}$ can be based on the qualitative state and auxiliary characteristics that define the different types of episodes.

7.2.3. Normalisation and returned values

Normally, a criterion is needed to judge whether the two sequences share a sufficient degree of similarity. Most alignment algorithms lack some type of normalisation that would appropriately rate the score obtained in respect to the longitudes of the sequences or a reference for quantifying significance.

The QSSI algorithm uses several normalisation policies by varying the normalisation factor according to the importance given to the final alignment. In any case, the similarity index returned by QSSI will be a normalised value between 0 and 1. The variations are described below.

Global alignment

This is the generalised case; the normalisation factor is defined as:

$$F_N = \max\left(m, n, \mathfrak{K}\right) \qquad [7.13]$$

where

$$\mathfrak{K} = \sum_{k=1}^{K} \max\left(\left[\mathrm{d}^i, \mathrm{d}^j\right]_k\right) \qquad [7.14]$$

The parameters m,n are the lengths of the time series $\boldsymbol{x}$ and $\boldsymbol{y}$, or likewise, the sum of the durations of the episodes in $\mathbb{X}$ and $\mathbb{Y}$. The pair $[\mathrm{d}^i, \mathrm{d}^j]_k$ are the maximum length of the episodes $\mathbb{X}_i$ and $\mathbb{Y}_j$ associated to the match k. Therefore $\mathfrak{K}$ is the total number of links between the two sequences.

This is an appropriate normalisation when the two sequences have identical importance and the similarity index is obtained based on the entire lengths.

Reduced alignment

This case is motivated by the fact that in some alignments there are no matched elements at the beginning or the end of the aligned sequences. So these elements are not considered for normalisation of similarity.

$$F_N = \max\left(m', n', \mathfrak{K}\right) \qquad [7.15]$$

Where m' and n' are the lengths of subsequences aligned, or similarly, the sum of durations of the episodes in those subsequences of $\mathbb{X}$ and $\mathbb{Y}$ that were aligned. For example, in Fig. 7.9 the subsequences are $\mathbb{X}' =\langle s,f,s,g\rangle$ and $\mathbb{Y}' =\langle s,f,g\rangle$. In these subsequences the length of each episode is 1 and then m'=4 and n'=3. On the other hand, if durations are different from 1, for example $\mathbb{X}' =\langle s2,f5,s1,g10\rangle$ and $\mathbb{Y}' =\langle s3,f3,g1\rangle$, then m'=18 and n'=7.

Overlapped alignment

Occasionally the high requirements of the application need to consider the full temporal extension that occurs when the sequences are aligned. This mode penalises overlapping sequences as shown in Fig. 7.9.

$$F_N = \max \; longT, \mathfrak{K} \qquad [7.16]$$

The parameter *longT* is the sum of the 2 lengths (again in terms of duration of episodes) once the sequences were aligned. In the example in Fig. 7.9 it has a value of 8. Note that, if the sequences are not overlapped, F_N is identical to the value obtained for the global alignment.

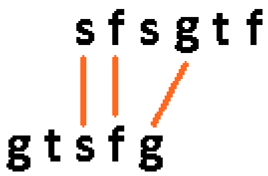


Fig. 7.9 Two overlapped sequences. The duration of each episode is 1.

Other returned values

It is possible that two sequences share significant similarity only in partial sequences, whereas the remaining regions are mainly unrelated. Then, regardless of the normalisation factor, the algorithm returns the ratio of the aligned subsequence with respect to the total length in terms of percent for each sequence. For this purpose a subsequence is considered the longest fragment of sequence with their terminal elements matched.

In the example in Fig. 7.9 the sequences were aligned in 66.66% and 60% of their length. This is important and complementary information if, for example, reduced alignment normalisation is used.

Next two sections present two complete examples of application of the QSSI algorithm.

7.3. Application example: classification of voltage sags

In this application the visual interpretation of original signals is simple for human experts. Thus, an operator could categorize the time series into a finite number of classes and perform the classification task intuitively. In these cases, the representation of process signals as sequences of qualitative episodes facilitates their integration with expert knowledge bases (KB) and hence, it is possible to automate the classification by comparing the new patterns with others registered in the KB.

This section presents the application of the above mentioned QSSI algorithm on the classification of voltage sags (transient reduction of voltage magnitude) gathered at 25kV distribution substations (Gamero et al., 2011). The objective is to assist monitoring systems in locating the origin of such disturbances in the transmission (HV) or distribution (MV) system.

7.3.1. Motivation and problem overview

The current dependence of industry, commerce, and services on electricity has led to the regulation of power quality. The most common disturbances affecting power quality are voltage sags. Sags are defined as the time interval between the instant when the root mean square (RMS) voltage decreases down to 90% of its nominal value and the instant when it returns to its normal level (a three-phase unbalanced voltage sag is shown in Fig. 7.10). The origin of these alterations may be faults (shortcircuits) or the operation of heavy loads (motors, transformer energization, etc.), and the duration depends directly on the reaction time of the power system, including firing protective system in presence of faults. These disturbances propagate through the power system according to electrical laws Bollen, 2000; consequently once a disturbance is detected and registered in a monitoring point (typically in the secondary of distribution transformers) it is necessary to deduce its origin (up or downstream) (Fig. 7.11) for two main reasons: a rapid maintenance intervention (when required) and to assign responsibilities for damages suffered by customers (if any).

With this aim, QSSI has been applied as a similarity measure to discriminate between voltage sags originated upstream (in the transmission network, HV) and downstream (in the distribution system, MV). Moreover, determining whether a sag has occurred in the distribution or transmission networks precedes the localization and mitigation stages Hamzah et al., 2004.

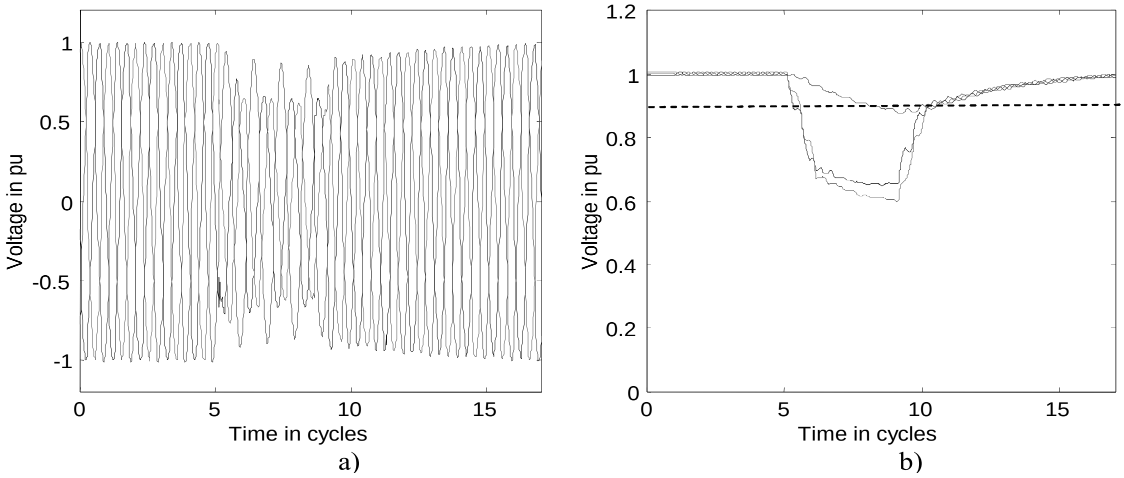


Fig. 7.10 a) Example of a three-phase unbalanced voltage sag. b) RMS voltage.

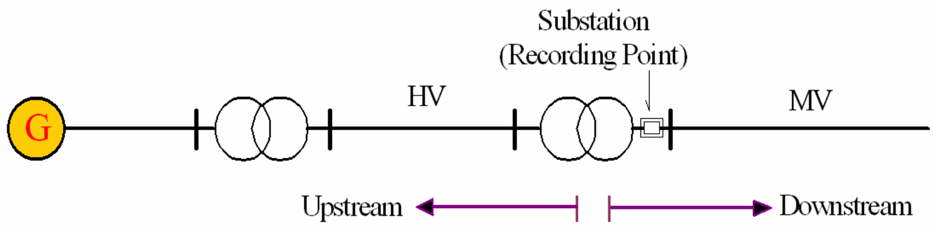


Fig. 7.11. Situation of Power Quality Monitors in the secondary wiring of transformers in distribution substations.

7.3.2. Qualitative representation of sags

Power quality monitors (Fig. 7.11), installed in the secondary winding of nine 132/25kV transformers in power distribution substations have been used in this application example. These instruments have been configured to detect variations of voltage measured in delta configuration (phase to phase) and they register voltage and currents as time series during 39 of periods (1 period = 20 msec.), starting two periods before the disturbance is detected (128 samples per period).

Information contained in each sample is individually irrelevant; the significant information is given by the existence of specific primitives and their duration. So, previous to the classification, the voltage and current waveforms are transformed into a sequence of episodes, described by a set of ordered qualitative symbols and their duration. The pre-processing and discretisation stages to obtain them are described below.

Sag pre-processing

Since voltage sags are characterized by the RMS waveform of voltages, simple pre-processing has to be applied to the instantaneous waveforms to obtain the RMS values. These values were obtained using a 1 cycle (20msec, 128 samples) sliding window and applying the Short Fourier Transform (SFT) to estimate the magnitude of the fundamental frequency (50Hz). This simple pre-processing is commonly used to identify basic characteristics of sags such as magnitude and duration Bollen, 2000. In this work the pre-processing has been applied before obtaining the qualitative description of waveforms in episodes as described in the next subsection. Using instantaneous values of waveforms, instead of RMS, to describe sags is only applicable when interest relies on phase shift, harmonic distortion or other transient characteristics Djokic et al., 2005.

Discretisation of raw data and qualitative representation

RMS series, describing voltage sags have been converted into sequences of episodes using the magnitude of waveforms and their first derivative. These parameters allow the signals to be represented in the same way a human expert would do it. The premise is that this human-based representation is sufficient to distinguish the different classes. Moreover, instead of considering the three phases in the conversion procedure, only the phase falling lowest during the sag was used. This is possible because the interest does not reside on the analysis of faulted phases but in the location up/downstream of the fault independently of which phase was affected. The results corroborate that this simplification is a valid way to obtain a useful representation for classification purposes. Then, the qualitative conversion procedure is performed following four basic steps:

- *Piecewise Aggregate Approximation (PAA)* is used to reduce the length of RMS waveforms without losing information. PAA Keogh et al., 2001b is a dimensionality reduction technique that approximates a time series by dividing it into *M* equal-length segments and using the average value of the samples in each segment as the data reduced representation. *M* must be selected to preserve the shape of the original time series. In this application, since the original waveforms contain 128 samples per cycle, a good choice for the segment length is 64 samples. Disturbances lasting less than half a cycle (64 samples) are not considered sags. Thus, dividing the original 4992 samples by 64 results in a sequence of length M=78, where each sample represents the equivalent of a half cycle while the original shape and the minimal information required are preserved.

- *Low Pass Filtering*: A 3-sample-length mean filter has been applied to the RMS waveform. The purpose of this filter is to smooth the dynamics of the signal to avoid spurious transitions between qualitative states. A similar effect would be obtained by defining a dead-band in the transitions between qualitative states.
- *Computing the first derivative.* The derivative of the filtered sequence is calculated to identify transitions between qualitative states.
- *Qualitative representation*. Finally, a qualitative representation is obtained from the evaluation of the magnitude and the derivative. A qualitative state is assigned to every sample according to the qualitative value of the derivative. According to the previous definition of episode, the consecutive samples with the same qualitative state constitute an episode and their length is defined by the number of them. The type of the episode, useful for classification purposes, is then obtained by adding the qualitative value of the magnitude. This gives information about the depth of the sag. In this way, episodes represent a constant behaviour (derivative) between two time instants preserving the magnitude at these instants.

In the application five categories (Steady State, Fall, Rapid Fall, Rise and Rapid Rise) have been defined for the derivative and six for the magnitude, resulting in an alphabet of 30 possible types of episodes (Table 7.1). The most obvious method to assign symbols to single values is range partitioning (Daw et al., 2003), e.g., by histograms. The six levels of magnitude have been defined according to the analysis of distribution of the depth of the sags. The histogram in Fig. 7.12 shows this distribution. Most of them (61%) have a minimum magnitude between 24kV and 18kV with a majority arriving between 20 kV and 22kV. Thus, the cut points have been defined for each 2kV around this majority class. Below 18kV a reduced number of sags is given, resulting in a more disperse distribution. This band has been split into two categories with the threshold at 12kV to have a similar *representativity* of the resulting categories. This categorization results in a minimum of 10% of available sags in each category.

Table 7.1 Types of episodes as ASCII symbols (ASCII code).

	1.Rapid Fall	2.Fall	3.Steady State	4.Rise	5.Rapid Rise
6:Level High (≥24KV)	u	v	w	x	y
5: ↓↓↓↓ (≥22KV)	k	l	m	n	o
4: ↓↓↓ (≥20KV)	a	b	c	d	e
3: ↓↓ (≥18KV)	V	W	X	Y	Z
2: ↓ (≥12KV)	L	M	N	O	P
1: Level Low (<12KV)	A	B	C	D	E

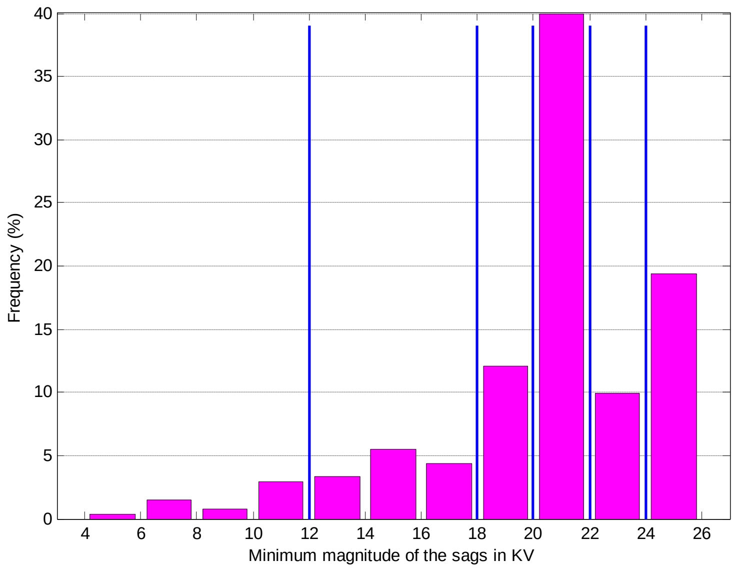


Fig. 7.12 Frequency chart of minimum voltage for sags

The application of this procedure results in a qualitative description of the waveforms of sags. This information can be represented by a string using the ASCII characters in Table 7.1 followed by the duration expressed as the number of consecutive samples with the same qualitative state.

The whole procedure can be observed in Fig. 7.13: (a) the three voltage phases are evaluated to select the deepest one; (b) the waveform is pre-filtered using PAA representation with ½ period (64 samples) length segments; (c) the first-derivative is calculated from the PAA waveform; and (d) the two waveforms (magnitude and derivative) are qualified (in the figure integer indexes are used for simplicity) before being converted into a discrete string. The evaluation of the first derivative is used to determine the duration of the episodes, since that establishes an interval with the same behaviour. The qualified magnitude at the end of the episode is preserved in the definition of each episode as an auxiliary characteristic. Notice that the level at the beginning of the first episode is always the same, around 25kV. Thus, the resulting string for the previous example is “v3a3b4d1e2d1m33x5w25”. According to Table 7.1, it is read as follows: during 3 samples (segments) the waveform falls and finishes at the highest level, during 3 more samples the signal presents a *rapid fall* and finishes at level 4, then it *falls*, slightly, during the following 4 samples remaining at the same level 4, and so on. This string containing the distinguished qualitative features will be used to compare this sag against others.

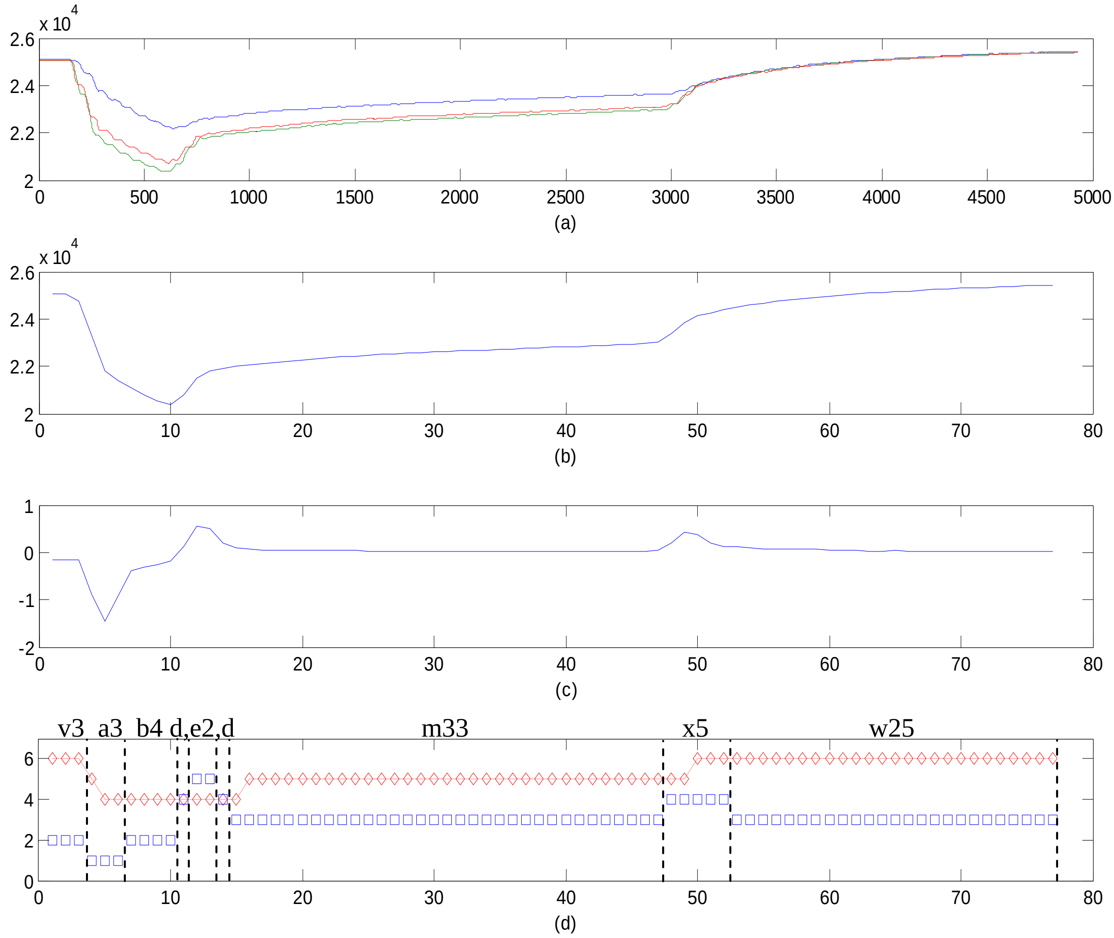


Fig. 7.13 Basic procedure to obtain the qualitative representation of waveforms. (a) Selection of the phase with greater depth. (b) The waveform is pre-filtered using the PAA representation. (c) The first-derivative is calculated from the PAA waveform. (d) The two waveforms are qualified according to breakpoints.

7.3.3. Classification of voltage sags

The whole set of data collected from nine substations was divided into two subsets. The first subset consists of a selection of representative registers to be used as a reference *dictionary,* and both subsets were used to test the performance of the QSSI algorithm. In the example, due to confidential restrictions on the use of real data, the names of the substations have been omitted and substituted with capital letters from A to I. The proposed method includes three main parts: first, preparation of the data as explained in the previous section. Next, each test case T is compared with the representative data set in the *dictionary* constructed previously. Finally, the test case is classified following a simple adaptive *k*-NN (*k* nearest neighbours) approach. These two last steps are described below.

Building the dictionary

In the first stage, a subset of four representative substations (labelled as A, E, F and I) were studied in order to build a *dictionary* of representative waveforms of sags. A total of 253 voltage sags classified as upstream/HV (transmission, 128 sags), or downstream/MV (distribution, 125 sags) according to the location of their origin in the power network was available from these substations. After transforming these waveforms into qualitative sequences a total of 94 representative patterns (40 labelled as HV and 54 labelled as HV) were retained to build the reference *dictionary*. Only the sags different enough from the others have been retained in the dictionary. Thus, a sag with a similarity greater than 2/3 with respect to another already in the dictionary is discarded. The dictionary size reveals that even though all substations are connected to a similar HV network, their behaviour is not completely equal.

Test

The waveforms registered in the second subset of substations (B, C, D, G and H) were added to the first one, obtaining 528 voltage sags for testing. QSSI has been used to determine the similarity between these sags and those ones in the reference *dictionary*. The goal is to determine the class of new sags based on the similarity. The complete set of sags has been classified using a simple *k*-NN approach as it is explained in the next subsection.

Classification method

Given a test case T, an adaptive *k*-nearest neighbour search algorithm Ougiaroglou et al., 2007 returns the set of *k* most similar cases, $C_1,\ldots,C_k$. In this particular example a minimum of $k=3$ nearest neighbours must be used to infer a class for the case T, and we have fixed $k=5$ as the maximum. Thus, the number of nearest neighbour will vary between $3\leq k\leq 5$. Similarity is used to order the cases as they are retrieved allowing an early-break heuristic to interrupt the retrieval task once enough cases (a minimum of 3) of the same class have been retrieved. Here, two classes conforms the solution space based on the origin of the sag: upstream (HV) or downstream (MV). So the class corresponding to T could be determined according to the majority following a voting strategy.

7.3.4. Results

Table 7.2 shows classification results according to the majority rule for the whole set of substations. The last two columns indicate the assigned class, labelled as HV or MV. For example in the substation A there are 48 sags originated in HV and the approach classifies 47 as HV and 1 sag as MV. For those substations used to build the dictionary, an accurate classification would be expected; however, there are some misclassifications. An exhaustive analysis reveals that the main cause of misclassifications is because the majority rule is independent of the similarity degree of the retrieved cases (Table 7.3). This suggests the need to improve the classification approach in the final application to take into account the closeness degree of similarity between the test and the retrieved cases (see example in Appendix C).

On the other hand, a rare example is found in Substation I. Table 7.4 shows the retrieved cases for this sag. Although a complete similitude was found to MV the majority belongs to HV, including a case with a similitude very close to 1. Fig. 7.14 shows the waveform of this sag and the waveform of the most similar one in HV. A revision of the information associated with these registers revealed that both corresponds to the same class and that the misclassification was due to a labelling mistake in the utility.

Table 7.2. Classification results.

			Majority rule	
		# sags	HV	MV
Substation A	sags HV	48	47	1
	sags MV	45	1	44
Substation B	sags HV	20	20	
	sags MV	16	2	14
Substation C	sags HV	58	58	
	sags MV	24	3	21
Substation D	sags HV	35	35	
	sags MV	27	2	25
Substation E	sags HV	33	33	
	sags MV	23		23
Substation F	sags HV	24	24	
	sags MV	25		25
Substation G	sags HV	17	17	
	sags MV	7		7
Substation H	sags HV	38	38	
	sags MV	33	6	27
Substation I	sags HV	23	23	
	sags MV	32	5	27

Table 7.3. Retrieved cases for the sag originated in HV and classified as MV in substation A.

Class	HV	MV	HV	MV	MV
Similarity	0.907	0.554	0.474	0.457	0.455

Table 7.4. Retrieved cases for a sag wrongly misclassified in substation I.

Class	HV	HV	HV	MV
Similarity	0.944	0.666	0.659	1

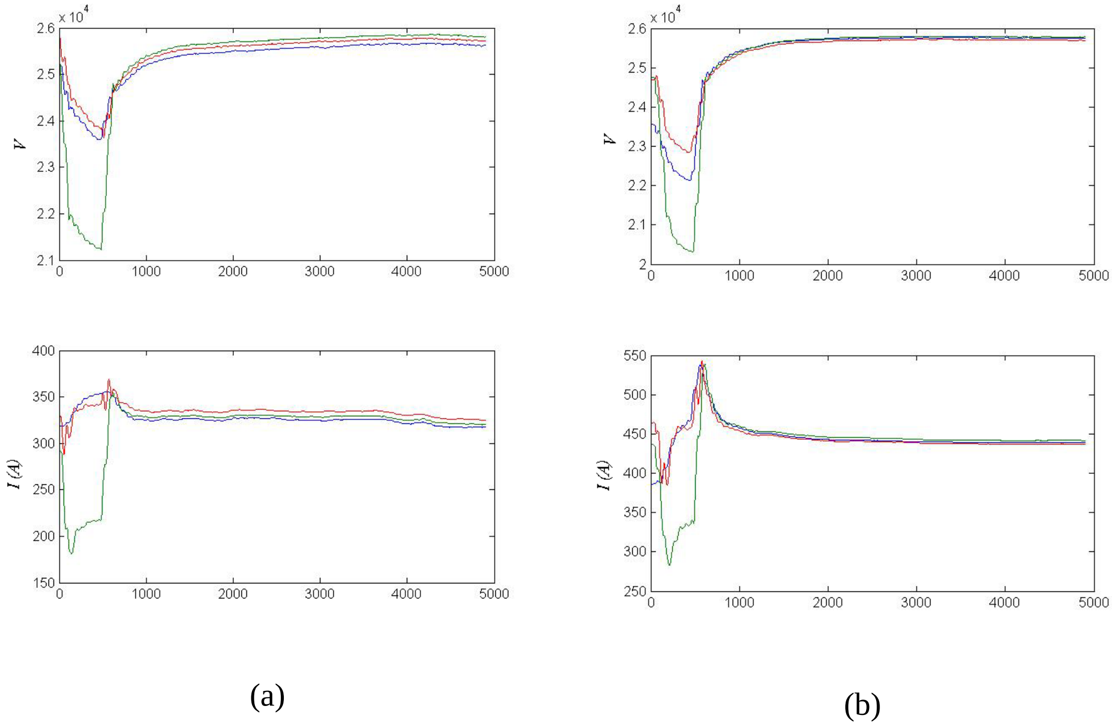


Fig. 7.14 Two waveforms with a similar QSSI. (a) Classified as HV and (b) originally bad classified as MV in the dictionary.

The main advantage of using the proposed methodology to locate the origin of sags is that it allows sags gathered in different substations to be compared and similarities to be found, despite the existence of quantitative differences in the waveforms associated with the electrical characteristics (loads, network topology, transformers, etc.) of each substation.

The next example will extend the algorithm to deal with complex sequences defined by multiple attributes instead of just a string with time.

7.4. Application example: situation assessment of a steam generator process

The aim of this example is to perform situation assessment in a steam generator. This means supplying relevant information about the state of the process to the process operator or to the expert. These states can either be normal, which corresponds to an operational state of the process, or abnormal, i.e., a failure.

7.4.1. A situation assessment approach based on QTA

The initial operation consists of representing the measured signals as sequences of episodes (qualitative and auxiliary quantitative data). This qualitative representation can be obtained offline (from a recorded signal), or online.

The trends identified are used for the purpose of matching against a knowledge base for the classification of known states. The functional states of the process can be identified using the process-history data (or by simulation). The experience of an expert or a collection of registered events is required in order to characterise the known behaviour of the system. Thus, a dictionary of events composed by qualitative sequences and descriptions of faults and normal states is defined. Afterwards, situations recognition is then based on the comparison between the trends from the dictionary and the observed online qualitative trends.

In this approach, the fault detection algorithm is an optional issue. If it does not exist, only existing events in the dictionary will be detected. Then two strategies could be followed: the first is based on the addition of all abnormal situations to the dictionary. The second is to add all the normal operations. So any deviation from expected behaviour leads to symptom detection. This detection procedure, based on such a discrepancy principle, allows abnormal situations due to real failures in the process sensors and/or actuators to be taken into account as well as unexpected situations which correspond to a normal operating of the process not considered in the elaboration of the dictionary. So, the classification stage would involve fault detection.

If the approach is ready for fault detection, new sequences are captured and reported as unknown events whenever a new fault occurs. Then, it is possible to build and/or extend the dictionary in online mode. The possibility of extending the dictionary at any time is an important characteristic. From the initial analysis a "reference model" of the process behaviour can be obtained and stored in the dictionary. This model obviously does not include modelling errors; nevertheless, it is not exhaustive as the historical data cannot cover the entire “life” of the process. This lack of completeness implies that an abnormal situation can either characterise a failure situation or an unexpected normal situation. In some cases, this pattern can include situations that correspond to critical states of the process in terms of equipment or operator safety. Thus, whenever a new unknown event is available, an expert can interpret the meaning of the event and add it to the dictionary.

Finally, new situations can be easily introduced manually. This is due to the use of qualitative representations that resemble human perception. This QTA framework follows the general scheme of Fig. 7.15.

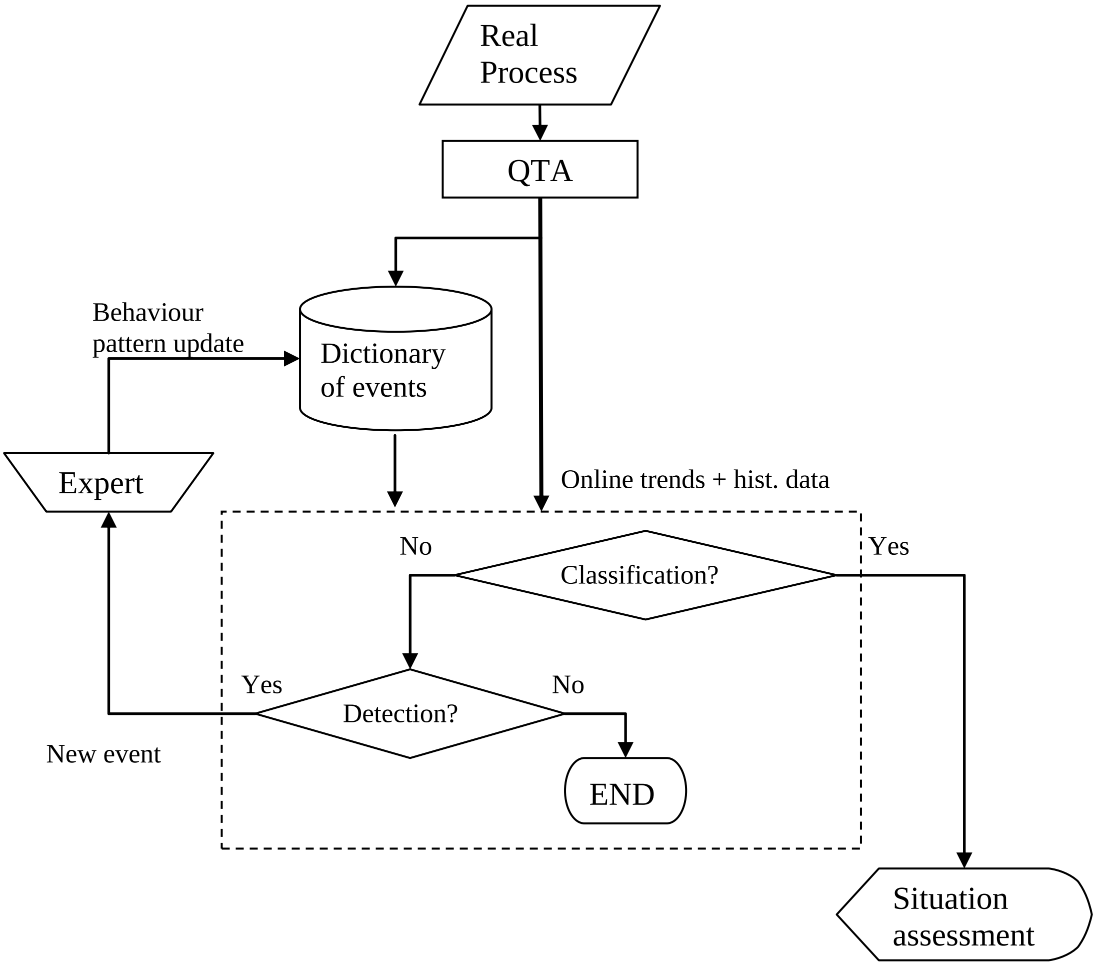


Fig. 7.15 A generalised scheme for situation assessment.

7.4.2. Episode-based representations

The proposed approach uses qualitative episodes as a representation of signals. Thus, the fault dictionary can be easily interpreted, constructed or expanded by an expert operator.

Two methods were used to obtain episodes. The aim is to test the accuracy of the QSSI algorithm by employing different qualitative representations, including quantitative data, but describing the same dynamics. The next two sections summarise these methods.

Qualtras

The first episode-based representation is applied by means of the tool *Qualtras*. The episodes can be obtained offline (from a recorded signal), or online, supplying the last finished episodes and the current episode at each sample time. When the episodes are extracted online, Qualtras introduces a delay of 4 samples in communicating a finished episode. This is because the tool checks if the episode is really finished.

The first derivative and its signal have been chosen as used features in order to obtain the episode representation. This simple representation, however, offers enough information to characterise the signals of the analysed process. So, the series of episodes obtained can now be used to describe patterns that identify particular classes of operating situations.

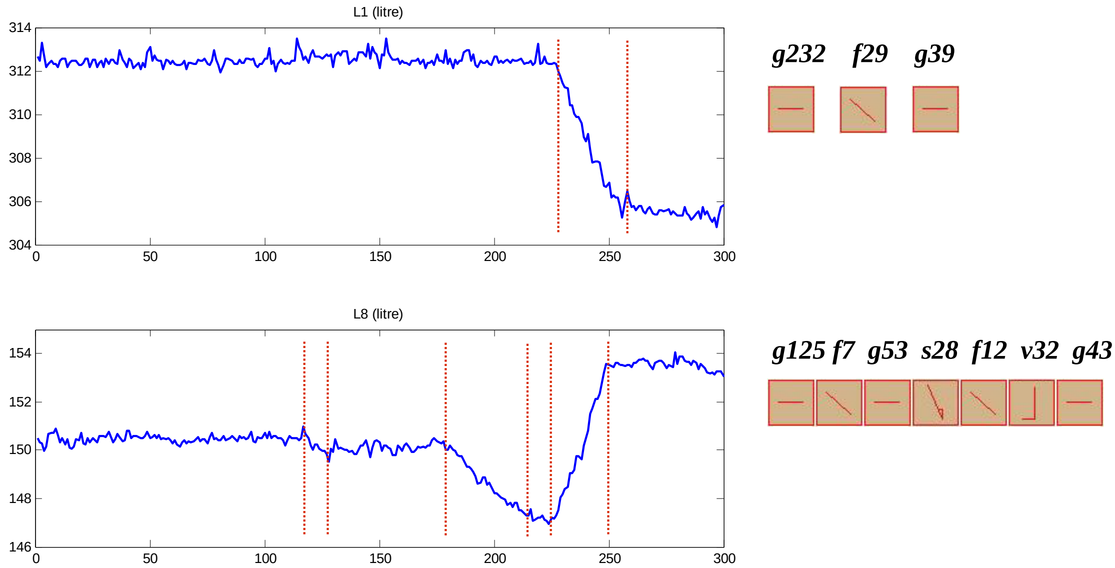


Fig. 7.16 Two signals and their qualitative representation. Vertical dotted lines indicate a change of episode.

The example in Fig. 7.16 shows two signals and their episodes obtained by Qualtras. For a visual representation, the user can select the icon from an extensive library (Fig. 4.4) so that it represents the most intuitive qualitative behaviour. In order to handle the qualitative sequence with other algorithms the episodes are described as different letters. In the example the resulting string for L1 is “g232f29g39”, this is read as follows: during 232 samples the signal L1 is steady, then it falls during 29 samples and finally it returns to steady again. In L8 the episodes ***s*** and ***v*** represent a *rapid fall* and an *abrupt rise* respectively.

An auto-tuning online trend extraction method

The second qualitative representation used in this approach has been obtained by means of the online auto-tuning trend extraction method presented in Charbonnier and Damour, 2008. The methodology splits the data into successive line segments and the signal variability is estimated. In this method an episode is increasing or decreasing if its amplitude is significantly greater than the level of noise, from a statistical point of view. Thus, the tuning parameters are modified at each segmentation time according to the signal variability. Finally, a parameter Δ expresses the delay the algorithm takes to establish the segmentation time, but the same value can be given to any variable recorded on the process.

Since only 3 primitives are available (*increasing*, *decreasing*, *steady*), the output of the algorithm will be episodes containing the slope and the final value of the primitive during a time interval. Fig. 7.17 shows a noisy pressure signal and the generated linear representation. The dotted lines are the primitives obtained in step 1, while the solid lines are the definitive episodes. Just as the above representation, to handle the qualitative sequence by other algorithms, the episodes are described as different letters. For example, the first 4 episodes are:

[g15,0,8.25][f110,-0.0037,7.84][t43,0.0078,8.18][g6,0,8.18]

with the same temporal significance as the sequences in section 3.1 and including the slope and final value. Episodes ***f, t*** and ***g*** represent decreasing, increasing and steady. In addition, two special episodes have been included, ***a*** and ***v***, which represents a *rapid fall* and a *rapid rise* respectively, with a duration of one sample.

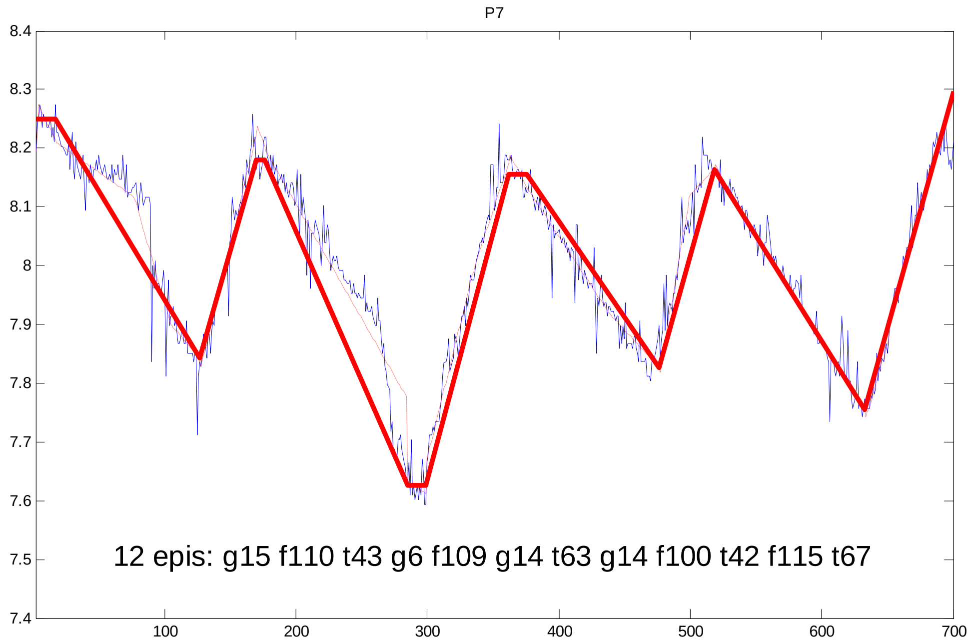


Fig. 7.17 A signal and its qualitative representation obtained by means of the auto-tuning online trend extraction method.

7.4.3. The steam generator process

The classification ability and performance of the proposed approach has been tested using data from a pilot plant. The process, a steam generator (Fig. 7.18), is designed to be a scale-model of part of a nonlinear power station which reproduces the same thermodynamic phenomena as the real industrial process.

The aim of this installation is to implement and test different methods and algorithms using the design of the advanced Decision Support System (Medjaher et al., 2006; Kempowsky et al., 2006) which can be used, for example, in the supervision of chemical and petrochemical process plants.

The main characteristic of the steam generator cycle and the faults which can occur in each cycle substructure are outlined below.

System description

The installation is mainly constituted of four subsystems: a feed water supply system, a boiler heated by a 60kW thermal resistor, a steam flow system and a complex condenser coupled with a heat exchanger.

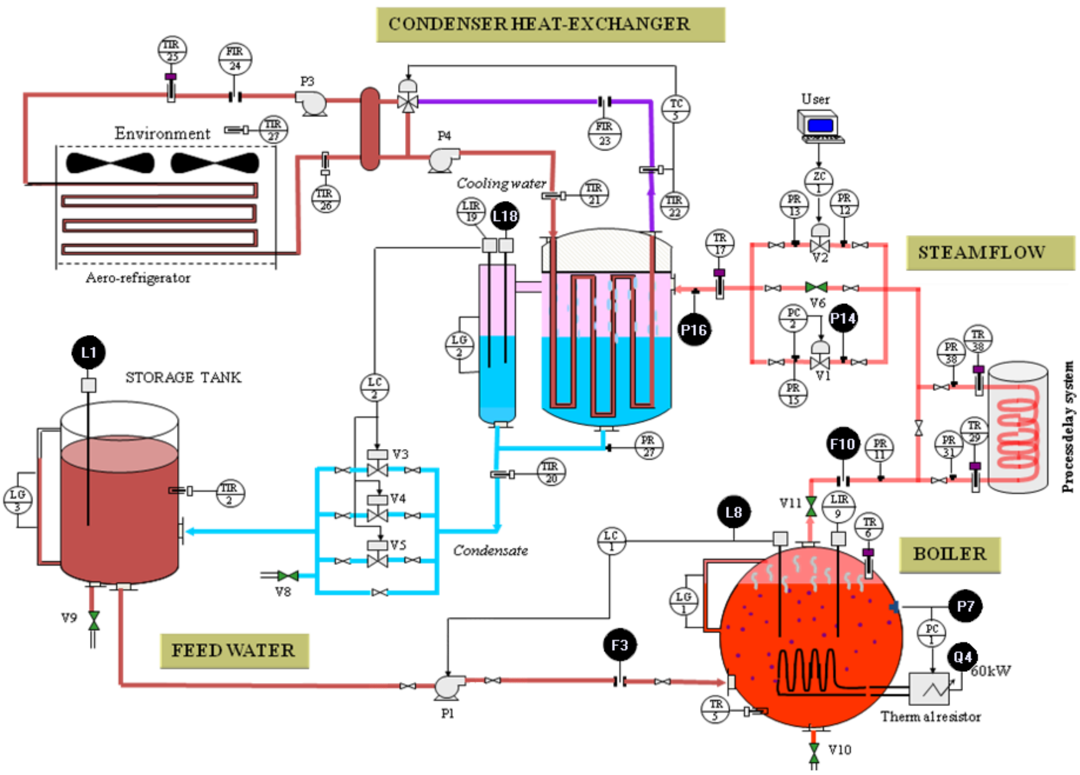


Fig. 7.18 Steam generator.

The feed water supply system consists of a receiver and the pump P1. The tank stores water and it feeds the boiler. The pump is controlled by an on-off controller to maintain a constant water level L8 in the steam generator.

The boiler subsystem produces under-pressure steam. Regulation of the steam generator process is carried out by controlling water level L8 and pressure P7 within the boiler. To maintain a constant water level L8 in the boiler, an on-off controller acts on the pump P1 feeding water flow F3. The water level must always be within a limited range of the set-point ±4 litres.

The heat/power value Q4 depends on the available accumulator pressure P7. When the accumulator pressure drops below a minimum value, the heat resistance delivers maximum power, and when the accumulator reaches a maximum pressure the heat resistance is cut off. As for the pump, good operation of the boiler is preserved by considering a hysteresis of ± 0.2 bar around the set-point.

The steam flow system replaces the turbine of normal power installations. The expansion of the generated steam is carried out by three valves in parallel connection. The first one is a manual bypass valve which is normally closed. It allows a leak in the system to be simulated. The second one is a controlled position valve, simulating the pass around of the steam flow to a condenser. The third valve is automatically controlled to maintain the proper inlet pressure P15 to the condenser.

The condenser heat exchanger system consists of two different circuits: a short circuit, if the outlet temperature is lower than a fixed value, and a long circuit, if the outlet temperature is higher than a fixed value.

To achieve situation assessment, including faults and other unexpected states, the signals shown in Table 7.5 are used. These sensors were chosen after studying the process faults. For processes with a higher number of sensors, Maurya et al., 2010 outlines the steps of an algorithm for optimal selection of fault-specific sensors.

Table 7.5 Measurement space range value. The data sampling is one sample per second.

Signal	Measurement variable	Unit	Scale	Nominal value
L1	Level in the storage tank	L	0-420	350
F3	Water flow from pump P1	L/h	0-1600	950
Q4	Heat power	Kw	0-60	55
P7	Boiler pressure	bar abs	0-16	8
L8	Boiler water level	L	143-156	150
F10	Steam flow	kg/h	0-100	Unsteady (about 30)
P14	Upstream pressure of valve V6	bar abs	0-16	8
P15	Downstream pressure of valve V6	bar abs	0-16	Unsteady (about 2.5)
P16	Inlet pressure to the condenser	bar abs	0-16	Unsteady (about 2.5)
L18	Condenser water level	L	0-14	6
Heat_Com	“on-off” heater control		0-60	55
Pump_Com	“on-off” pump control			0 or 1

Overview of faults

Several kinds of faults can be introduced in the process. In this example, the 6 fault classes described in Table 7.6 have been examined. Each fault class is described through the most relevant measurements carried out in the process (Table 7.7) and describes a specific situation to be diagnosed.

Table 7.6 List description of detectable faults.

Fault reference	Description
F1	Water leakage in the boiler
F2	Water leakage in the tank
F3	Water leakage inside the condenser
F4	Boiler output blockage
F5	Heater failure
F6	Pump fault

Table 7.7 Relation of signals involved for each fault.

	L1	F3	Q4	P7	L8	F10	P14	P16	L18	HeatC	PumpC
F1	✓				✓						
F2	✓				✓						
F3	✓							✓	✓		
F4						✓	✓				
F5			✓	✓			✓			✓	
F6		✓									✓

Data sets

Two data sets were exploited to illustrate the operating capacity of the framework. The first set, called *MIX1*, corresponds to the complete set of fault situations. Fig. 7.19 shows the evolution of these signals for scenario *MIX1*.

The second one, called *BoilerPressureControllerFault* (Fig. 7.20), corresponds to a new fault that was not included in the initial fault list. Therefore, the approach should recognise it as an unknown state.

The case of multiple process faults occurring simultaneously has not been considered. Results are presented in the following sections.

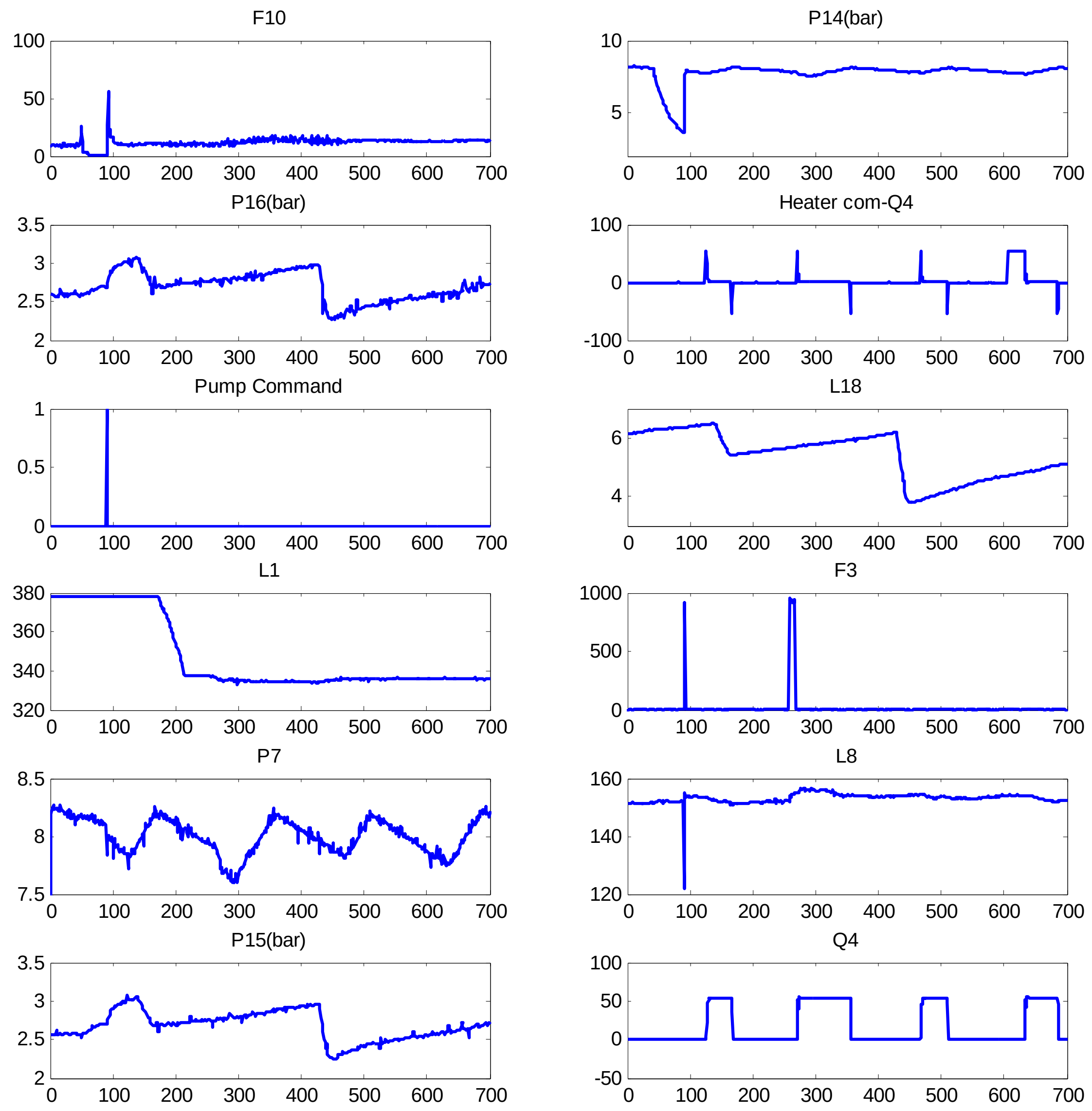


Fig. 7.19 Signals from the data set *MIX1*.

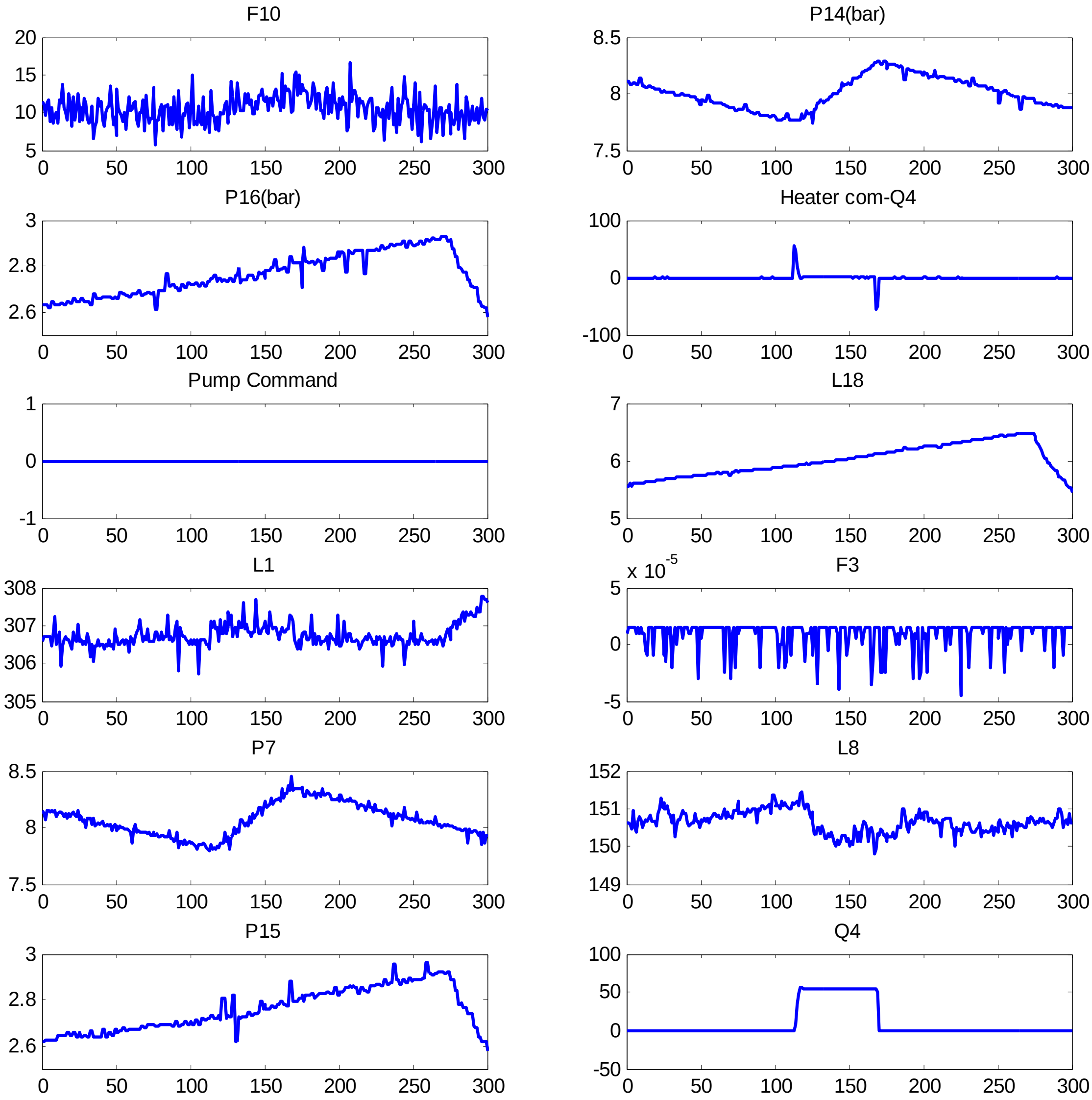


Fig. 7.20 Signals from the data set *BoilerPressureControllerFault.*

7.4.4. The framework operation

In this illustrative example, the trends of observed variables for the 6 fault classes are stored in a trend dictionary as an event. In this case the dictionary is created offline. Each event is made up of the temporal qualitative sequence from sensors, the definition of the fault, if it is beginning or ending and a flag indicating if the event was detected when a regulated variable (P7 and L8) was out of control.

The resulting dictionary is then used to perform the situation assessment. Thus, for each sequence in the dictionary, the QSSI algorithm computes an alignment score that represents the degree of similarity between the current trends and the dictionary sequence. If the obtained similarity index QSSI becomes high enough then the fault with the highest QSSI is declared as the diagnosed state.

The presented framework was run online when episodes were extracted with Qualtras. Later, it was run offline using the auto-tuning formalism described in section 7.4.2. In both cases the QSSI was calculated when a new episode finished or if a regulated variable reached an abnormal value without changing the setpoint.

If the QSSI is evaluated when a regulated variable is out of control but no state could be identified, a new state is deemed to have occurred and the system assists the operator in introducing this new event into the dictionary or discarding it. The results are presented in the following subsections.

7.4.5. Case study I: online operation

This case study uses the sequences of episodes obtained by Qualtras. According to the available symbols used to represent episodes (Fig. 7.21), the local similarity chart between episodes is defined (Fig. 7.22).

Similarities can be based on the qualitative state and auxiliary characteristics, or they can be subject to the criterion of the user, so one could give more importance to some episodes concerning the others. In Fig. 7.22, the first 9 episodes represent different degrees of slope, ordered from decreasing to increasing, and the last 3 episodes are "low", "normal" and "high". In this chart, if two episodes are exactly the same, the value of similarity will be "1" and when they are totally different, the value of similarity will be "0". Distances are based on the following criteria:

- The special episodes "low", "normal" and "high" (*l*, *n*, *h*) have no similarities to any other. Typically these symbols represent episodes based on the signal value.
- About the normal episodes (*a*, *s*, *f*, *o*, *g*, *p*, *t*, *r*, *v*):
    - The first neighbour on the diagonal receives a similarity value of 0.5.
    - The second neighbour on the diagonal receives a similarity value of 0.2 if the episodes represent slopes with the same sign.
    - All the other combinations have totally no similarities between them.

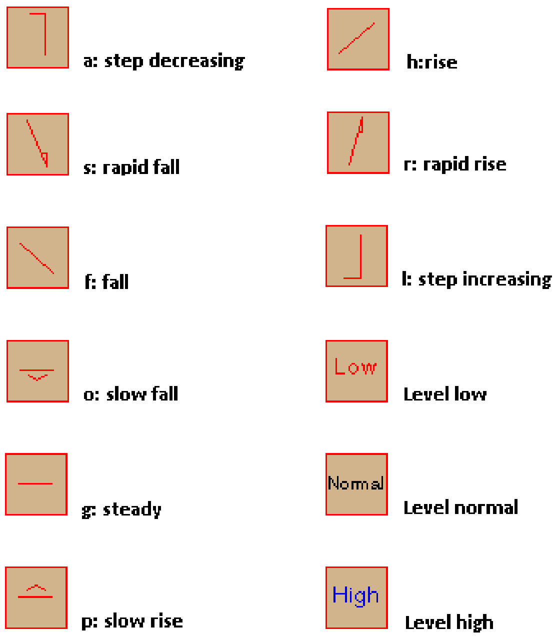


Fig. 7.21 List of available symbols and their meaning.

EPIS	a	s	f	o	g	p	t	r	v	l	n	h
a	1	0,5	0,2	0	0	0	0	0	0	0	0	0
s	0,5	1	0,5	0	0	0	0	0	0	0	0	0
f	0,2	0,5	1	0,5	0	0	0	0	0	0	0	0
o	0	0	0,5	1	0,5	0	0	0	0	0	0	0
g	0	0	0	0,5	1	0,5	0	0	0	0	0	0
p	0	0	0	0	0,5	1	0,5	0	0	0	0	0
t	0	0	0	0	0	0,5	1	0,5	0,2	0	0	0
r	0	0	0	0	0	0	0,5	1	0,5	0	0	0
v	0	0	0	0	0	0	0,2	0,5	1	0	0	0
l	0	0	0	0	0	0	0	0	0	1	0	0
n	0	0	0	0	0	0	0	0	0	0	1	0
h	0	0	0	0	0	0	0	0	0	0	0	1

Fig. 7.22 The Similarity chart used for qualitative episodes. The values are normalised between 0 and 1.

Next, the online operation is executed and the episodes are continuously extracted. Classification by QSSI is performed whenever a new episode is available or if a controlled variable is out of control. So, under normal conditions the evaluation frequency depends on the episode duration. This feature is the main cause of delay in diagnosis time. On the other hand, the occurrence of unusual primitives in some sequences allows faster diagnosis.

The results of the situation assessment for scenario MIX1 are presented in Table 7.8. One can see that the states are identified within a short time (column 2) after the introduction of the fault (column 1). Column 3 shows the diagnosed fault and the obtained QSSI value for the two variables involved for each fault at the instant of detection. Column 4 shows the following diagnosed state and, as can be seen, at least one of the variables does not have enough similarity and a wrong diagnosis is avoided.

Table 7.8 Results of QSSI based situation assessment for scenario MIX1.

Introduced fault	Diagnosis time	Top two diagnosed faults and QSSI value	
(From t_{ini} to t_{end})	(From t_{ini} to t_{end})	Fault 1 (QSSI)	Fault 2 (QSSI)
F4 (41s – 90s)	46 – 93	F4 (0.521,0.668)	F5 (0.738,0.445)
F2 (172s – 212s)	177 – 227	F2 (0.543,0.796)	F1 (0.656,0.347)
F6 (257s – 268s)	263 – 274	F6 (0.542,0.89)	F5 (0.738,0.445)
F3 (429s – 458s)	433 – 467	F3 (0.91,0.71,0.67)	F5 (0.73,0.445)
F5 (606s – 634s)	609 – 643	F5 (0.738,0.589)	F1 (0.738,0.347)
F1 (647s – 684s)	657 – 688	F1 (0.738,0.695)	F4 (0.521,0.411)

As an example, Fig. 7.16 shows the sensor signals L1 and L8 involved for the faults F1 and F2. In this particular case, a boiler leakage (F1) is introduced from the sample time 180 until 225. When the fault begins, L8 shows a drop in the water level of the boiler while L1 just operates as normal. After, the fault is restored and L1 will be decisive in detecting the end of the fault when the boiler becomes full. Table 7.9 shows the messages created by the approach when the fault F1 is detected and later when the plant returns to normal operation. The variable *PAlign* indicates the percent of the current signal pattern and the dictionary sequence that has been aligned respectively. This is an important characteristic since the known events in the dictionary are represented by the entire sequence. Thus, a successful classification should align the whole recognised sequence. Note that the situation assessment is performed 4 seconds after an episode has concluded. This is because the algorithm waits this time in order to determine whether the next episode is a new type or if it can be eliminated or aggregated to the previous one.

Table 7.9 Messages returned by the approach for fault F1.

```
(time instant: 653 - Comm. time: 657) <<<<<
Boiler Leakage FBegin
current L1: g25 and LibL1(class 1): g40. Sim.score: 0.73893 PAlig: 100  100
current L8: g9s16 and LibL8(class 1): g10s25f5. Sim.score: 0.6958 PAlig: 100  100
Sit. Assessment nr.1: F1.1 with similarities: 0.73893     0.6958
Sit. Assessment nr.2: F4.1 with similarities: 0.52176     0.4117

(time instant: 684 - Comm. time: 688) <<<<<
Boiler Leakage FEnd
current L1: g25 and LibL1(class 0): g40. Sim.score: 0.73893 PAlig: 100  100
current L8: s12f6g7 and LibL8(class 0): g20. Sim.score: 0.5576 PAlig: 55   100
Sit. Assessment nr.1: F1.0 with similarities: 0.73893     0.5576
Sit. Assessment nr.2: F2.1 with similarities: 0.21354      0.4725
```

Regarding specific situations, it can be seen in Fig. 7.19 that the variables L8 and P7 are out of control at approximately samples 260 and 270 respectively. This situation matches with the beginning and end of the pump failure (F6), which has this effect on the two variables at different instants. The variable P7 is also out of control at sample 623 until sample 641. Now the situation corresponds to the heater fault (F5) detected at sample 609 and corrected at sample 643.

If the QSSI is evaluated when a regulated variable is out of control but no state can be identified, a new state is deemed to have occurred and the system assists the operator in introducing this new event into the dictionary or discarding it.

The second data set (Fig. 7.20) corresponds to the detection of a fault that was not learned during the process characterisation. The fault is introduced in the boiler pressure controller at sample 151 and is back to normal at sample 168. The evaluation is executed at sample 158 when the detection algorithm detects a deviation in P7 and concludes the variable is out of control. Since the trend dictionary does not include the signature for this fault the approach presents this state as a novel event. At this moment the approach registers the episodes for all variables (Table 7.10) and it shows them to the operator. Then the operator is requested to add this fault to the dictionary by performing the diagnosis manually, wait until the fault has sufficiently evolved, or discard it.

Table 7.10 Messages returned by the approach when a variable is out of control.

```
T: 158   WARNING: P7 OUT OF CONTROL!!!
F10; n30;
P14; p3t27;
P16; p30;
HEATER_COM-Q4; n30;
PUMP_COMMAND; l30;
L18; p30;
L1; g30;
F3; l30;
P7; p2t28;
L8; f8g22;
HEATER_COM;
Q4; h30;
```

Note that the maximum duration for each sequence is 30 seconds as this length is enough to represent the current state. However, there is not a critical value since the QSSI algorithm returns information about the percentage of the aligned sequence and it returns a similarity value according to this percentage. Regarding our example, the correct action by the operator is to add a new diagnosis relating variables P7 and Q4 and to flag the new diagnosis as "Out of control".

7.4.6. Case study II: offline operation

This section presents a similar approach to that described above but running offline and using the auto-tuning trend extraction method described in section 7.4.2. The aim is to test the QSSI algorithm with a different qualitative representation, including quantitative data, over the same scenario. Thereby, it is possible to test the robustness of the QSSI algorithm against different qualitative representations. Although the method allows online trend extraction, it has not been implemented in Qualtras yet, where it would be capable of connecting to the real process. Thus, this case study was analysed in offline mode.

The steps to follow are the same as in the previous method. First, the episodes are extracted in offline mode and a new trend dictionary of events is created adding new trends representing the 6 fault classes. Since the trend algorithm only uses 3 types of episodes, 2 special episodes representing steps or discontinuities (types *l* and *v*) have been incorporated. That is, increasing and decreasing episodes but with a very short time duration. Moreover, episodes include quantitative data to enhance the comparison. For successful situation assessment the use of qualitative data and the slope is enough.

Secondly, another local similarity chart between episodes is defined (Fig. 7.23). Only the episodes based on the signal value (*low*, *high*) and the 3 episodes (*a*, *g*, *v*) with a quantitatively similar slope have a similarity value of 1 if they are compared to an identical one. Cells marked with an asterisk have a similarity value calculated according to their slopes, such as:

$$Sim(\mathbb{Q}^i, \mathbb{S}^j) = 1 - \frac{|attribute_i - attribute_j|}{\max(|attribute_i, attribute_j|)} \qquad [7.17]$$

With *attribute* being the slope and $Sim(\mathbb{Q}^i, \mathbb{S}^j)$ the normalised similarity between two episodes $\mathbb{Q}^i$ and $\mathbb{S}^j$. If the slopes are the same, the similarity is 1.

EPIS		a	f	g	t	v	Low l	High h
	a	1	0	0	0	0	0	0
	f	0	*	*	0	0	0	0
	g	0	*	1	*	0	0	0
	t	0	0	*	*	0	0	0
	v	0	0	0	0	1	0	0
Low	l	0	0	0	0	0	1	0
High	h	0	0	0	0	0	0	1

Fig. 7.23 The Similarity chart used for qualitative-quantitative episodes.

Finally, the QSSI algorithm runs when a new episode is finished. Fig. 7.24 illustrates the offline recognition performed for a period of 700 seconds. The figure shows evolution of the observed variables and their corresponding episodes. Vertical lines mark the beginning and the end of the faulty situations detected. These results are presented in Table 7.11.

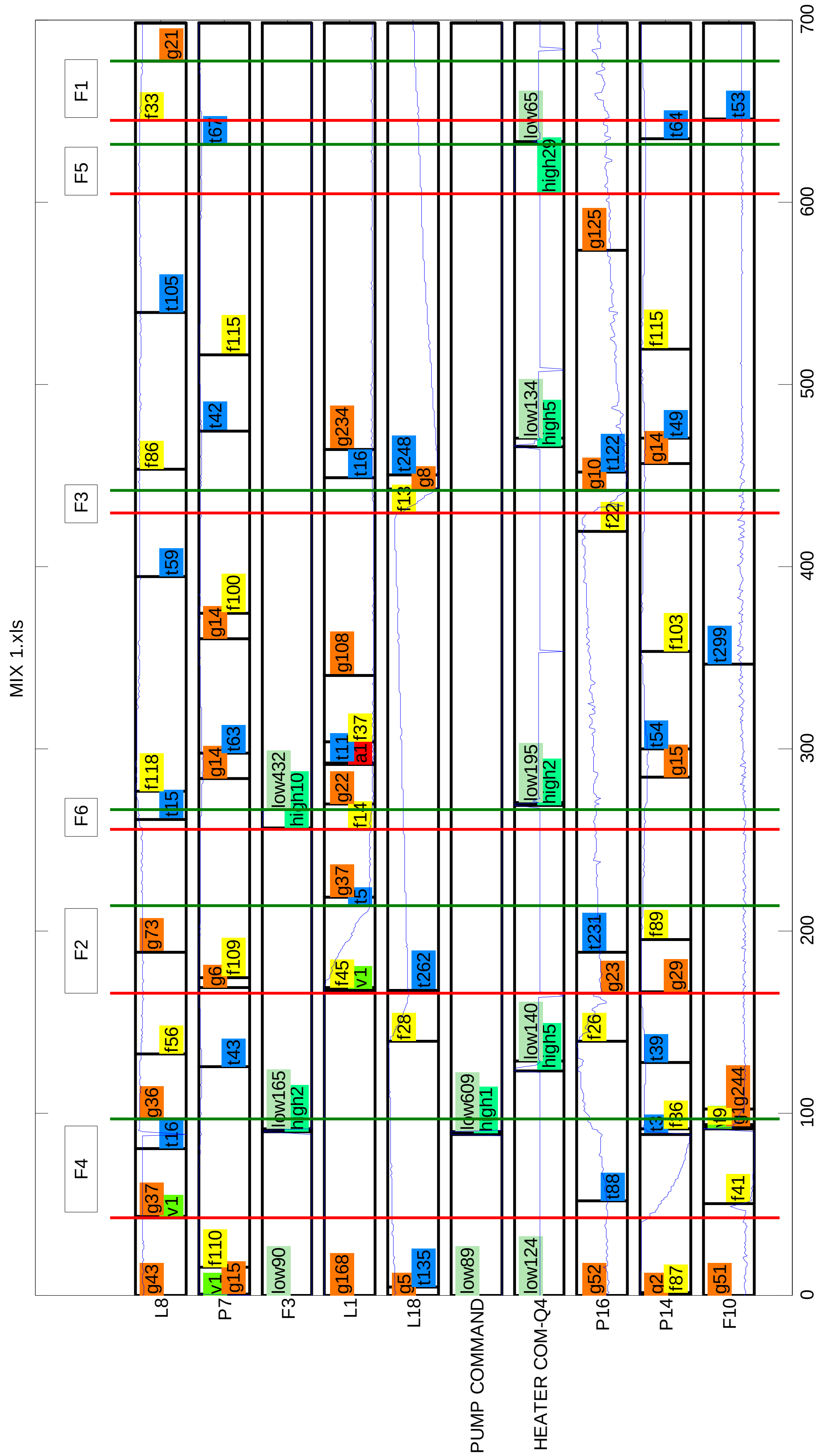


Fig. 7.24 Timeline representation of episodes and identified faulty situations.

Table 7.11 Results of QSSI-based situation assessment for scenario MIX1.

Introduced fault (From t_{ini} to t_{end})	Diagnosis time (From t_{ini} to t_{end})	Top two diagnosed faults and QSSI value	
		Fault 1 (QSSI)	Fault 2 (QSSI)
F4 (41s – 90s)	43 – 97	F4 (0.803,0.438)	F4 (0.54,0.572)
F2 (172s – 212s)	166 – 214	F2 (0.567,0.726)	F1 (0.624,0.381)
F6 (257s – 268s)	256 – 267	F6 (0.618,0.89)	F2 (0.497,0.73)
F3 (429s – 458s)	430 – 452	F3 (0.58,0.89,0.56)	F5 (0.821,0.323)
F5 (606s – 634s)	605 – 632	F5 (0.838,0.571)	F1 (0.333,0.766)
F1 (647s – 684s)	645 – 678	F1 (0.803,0.591)	F2 (0.333,0.689)

As this case study was performed in offline mode the time diagnosis obtained was slightly better than those shown in the previous case. This is because the analysis was run at the end of each episode, but data is captured by a window centered on this point. Thus, the sequences evaluated by the QSSI algorithm contain future episodes and episodes already elapsed.

7.5. Conclusions

This chapter has focused on defining and implementing a new similarity algorithm (QSSI) able to deal with the temporal misalignment problem caused by dynamic processes. The algorithm returns a normalised index related to the degree of similarity between qualitative trend signals.

The idea behind the algorithm is the computing of the optimal alignment, determined by finding the longest pair of subsequences in the full sequences which yields the highest alignment score among the set of all subsequences and their possible alignments. Afterwards the similarity is calculated using the aligned elements.

Therefore, since QSSI finds the longest related subsequences, it is not considered a global alignment method. However, the different normalisation modes allow the similarity values to be adapted to the necessities of the user. Moreover, additional information, such as the percentage of aligned sequences, complements the meaning of the returned value. It must be remembered that those elements not related within the subsequences cannot be removed because they constitute part of the signal evolution, and hence they should also participate in the final similarity value.

The performance of this method has been illustrated by two application examples. The first example shows the classification of voltage sags gathered at 25kV distribution substations. QSSI is used to return the similarity between new and previously registered and labelled sags.

The second example describes an approach for online monitoring and situation assessment applied to a steam generator. In this application the accuracy of the QSSI algorithm has been validated using two different qualitative representations providing both quantitative and qualitative information.

Thus, the QSSI algorithm has been shown as a good method for determining the similarity between two qualitative sequences due to its ability to align sequences with different longitudes. The promising classification accuracy suggests that the algorithm presented could be applied satisfactorily and confirms its usefulness in classification approaches.

Chapter 8.

Comparison of similarity measures on time series

8.1. Introduction

This chapter is dedicated to the comparison between the three methods developed in this thesis and the different similarity measures used when dealing with numerical series. This is an interesting study, since methods that work with qualitative representations are compared to traditional algorithms.

A comprehensive evaluation of distance and similarity measures is presented in Ding et al., 2008, where the authors conclude that the best accuracy is obtained by the DTW and EDR algorithms. Their conclusion motivated the inclusion of these two methods in this comparison study as well as inspiring the methodology followed.

8.2. Methodology and parameters

In order to carry out the comparison the experiments were performed on 15 time series data sets provided by the UCR Time Series repository (Keogh et al., 2011). The data sets come from a wide variety of applications and they are independent from the application examples shown so far. This makes the comparison meaningful, especially when these data sets are often used by algorithms that deal with numerical series.

In this study some of the measures outlined (ED, DTW, ERP, EDR and LCS) in Chapter 5 and the new algorithms created (QSSI, EpDTW and DTW_{ONLINE}) were included.

The evaluation method uses a one nearest neighbour (1NN) classifier such as that proposed in Keogh et al., 2011 to evaluate the efficacy of the distance measure used. Specifically, each time series has a correct class label, and the classifier tries to predict the label as that of its nearest neighbour in the training set. Hence, the accuracy of the 1NN classifier directly reflects the effectiveness of the similarity measure.

Several techniques require the setting of one or more parameters. For methods that deal with numerical series, the setting parameters listed in Table 8.1 were used and the results are based on the best error ratio obtained. A common parameter is the window size (or warping window) expressed as a percentage of the length *n* of the time series, while the algorithms based on edit distances use a matching threshold parameter ε (Section 5.5).

The setting parameters used for each data set are indicated in Table 8.2. The parameters were chosen by testing a small group for each parameter according to a preview representation of time series.

Table 8.1 Parameter tuning for similarity measures based on numerical series.

Method	Parameters	
	Window size ($\%\cdot n$)	ε
DTW	3,6,9,12	-
DTW_{ONLINE}	3,6,9,12	-
EDR	3,6,9,12	0.02,0.1,1,3
ERP	3,6,9,12	
LCSw	3,6,9,12	0.02,0.1,1,3

Regarding EpDTW and QSSI, the time series must be converted first to qualitative sequences. Until now, each application has been studied individually and the parameters needed for the qualitative representation have been optimised. In this experiment a common method to obtain qualitative representation from all time series was used. Then, the obtained representations could be not optimal. The chosen method is a variation of the SAX approach. Thus, each episode is represented by the symbol returned by the SAX algorithm and the modification is based on considering the slope in every word. In this way the method generates two new groups of episodes; O, P and U, V corresponding to falling and rising episodes respectively. The first episode of each group is assigned if the slope exceeds a certain threshold, and the second if the slope exceeds the threshold*3.5. This value is arbitrary and in practice there is hardly any episode that exceeds this second threshold. This modification was introduced to enhance the qualitative representation, obtaining a closer representation to the dynamics of signals. For example, in Fig. 8.1 the second segment has a slope greater than 0.15, so this segment is labelled as U. The name of the label is irrelevant and it should only be different from the other labels used by the SAX algorithm. It can be observed from this figure that the sequence of episodes is ⟨a1,U1,g1,f2,e2,c3,b2⟩. Then, since each segment contains 10 samples (PAA=10), the final sequence is ⟨a10,U10,g10,f20,e20,c30,b20⟩. Another example is shown in Fig. 8.2, where the sequence of episodes obtained for a PAA=4 is ⟨O8,d8,c12,b16,O4,U16,O12,a12, b4,c24,d4,U8⟩.

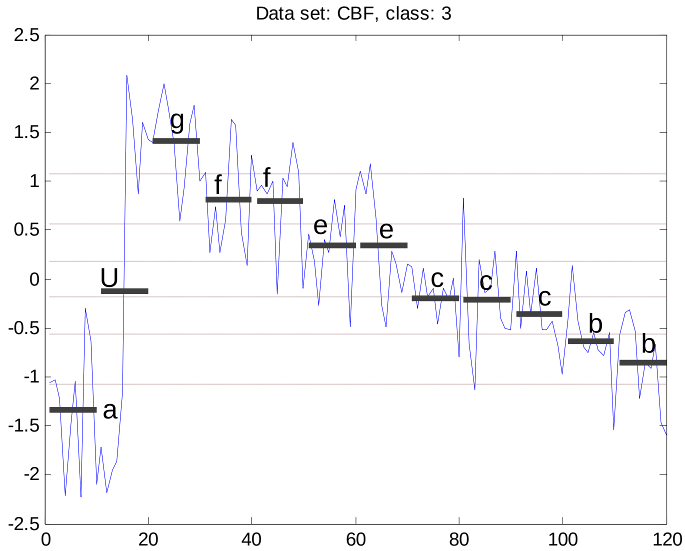


Fig. 8.1 Example of qualitative representation based on SAX (PAA=10, alphabet=7, slope threshold=0.15).

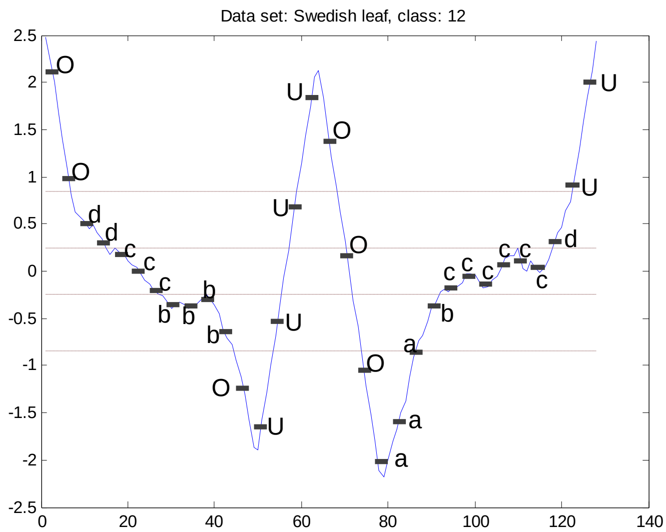


Fig. 8.2 Example of qualitative representation based on SAX (PAA=4, alphabet=5, slope threshold=0.15).

The scores used by QSSI and EpDTW follow a simple criterion: if the episodes are the same the score is 1, and if the episodes are neighbours the score is 0.5. This rule can be represented in the score matrix shown in Fig. 8.3. The first 9 episodes correspond to the alphabet generated by SAX while episodes O, P and U, V correspond to the two new groups according to the slope (modified SAX).

	a	b	c	d	e	f	g	h	i	O	P	U	V
a	1	0.5	0	0	0	0	0	0	0	0	0	0	0
b	0.5	1	0.5	0	0	0	0	0	0	0	0	0	0
c	0	0.5	1	0.5	0	0	0	0	0	0	0	0	0
d	0	0	0.5	1	0.5	0	0	0	0	0	0	0	0
e	0	0	0	0.5	1	0.5	0	0	0	0	0	0	0
f	0	0	0	0	0.5	1	0.5	0	0	0	0	0	0
g	0	0	0	0	0	0.5	1	0.5	0	0	0	0	0
h	0	0	0	0	0	0	0.5	1	0.5	0	0	0	0
i	0	0	0	0	0	0	0	0.5	1	0	0	0	0
O	0	0	0	0	0	0	0	0	0	1	0.5	0	0
P	0	0	0	0	0	0	0	0	0	0.5	1	0	0
U	0	0	0	0	0	0	0	0	0	0	0	1	0.5
V	0	0	0	0	0	0	0	0	0	0	0	0.5	1

Fig. 8.3 Score chart used by EpDTW and QSSI.

8.3. Results

The results obtained from this experiment are shown in Table 8.2. The tests were carried out on a computer with an Intel Core2 Duo 2’93GHz processor and 2GB of RAM.

Although the table shows the results for all the methods, the DTW_{ONLINE} algorithm should only be compared to the classic DTW algorithm in order to appreciate the difference in time execution. Aside from this exception, the other methods can be compared to each other.

8.3.1. DTW vs. DTW_{ONLINE}

Here, the DTW_{ONLINE} algorithm is used in order to reduce the calculation time with regard to classic DTW. In this case, the sliding window size *w* and the warping window width, *r*, have the same value. If the window *w* does not reach the length of the time series, the algorithm returns the minimum distance from the last column or row in the cumulative distance matrix. Then, if this value exceeds the minimum distance obtained earlier, the calculation of the remaining cells in the distance matrix can be aborted. This strategy is valid because a one nearest neighbour classifier is used. Thus, in this application the path does not need to be recalculated.

The columns dedicated to DTW and EpDTW methods in Table 8.2 shown the error ratio and the execution time in seconds. It can be observed that the difference in execution time can be very important while the error ratio is very similar.

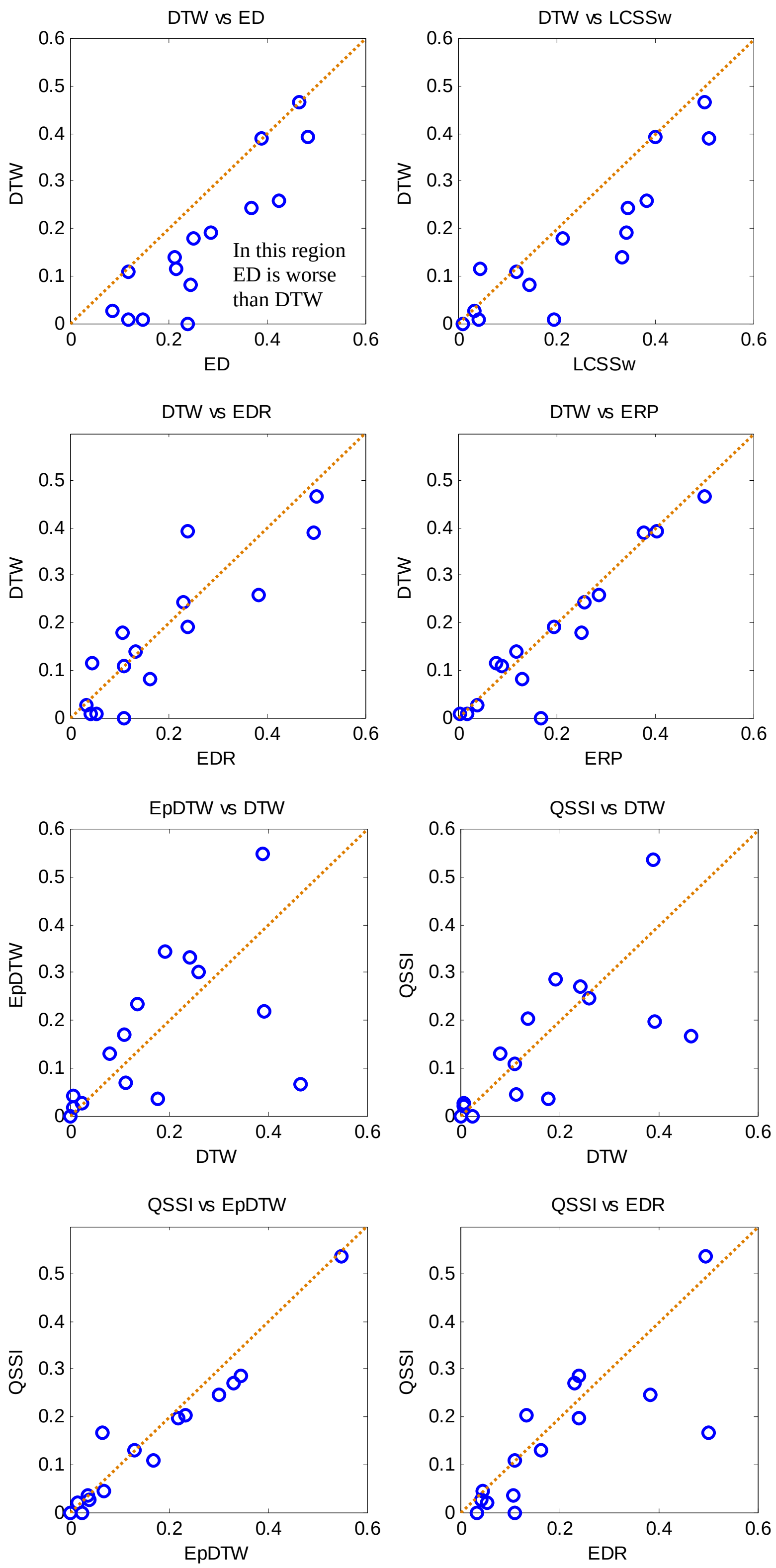


Fig. 8.4 Illustration of error ratios.

8.3.2. Quantitative vs. Qualitative methods

In these experiments DTW, EDR, ERP and the variation of LCS (eq. [5.15]) developed byGuo and Siegelmann, 2004 (here named LCSw for brevity), were used for comparison with EpDTW and QSSI, two methods based on qualitative representations.

Fig. 8.4 provides a more intuitive illustration of the performance of the similarity measures compared in Table 8.2. The error ratios of the two methods under comparison are used as the x and y coordinates of a dot, where each dot represents a particular data set. Thus, the area with highest number of dots indicates the method with worse results.

Based on the error ratio obtained, there is not any method categorically superior to DTW, but some methods are more effective than others on certain data sets.

If only the EpDTW and QSSI methods are compared, QSSI proves to be superior to EpDTW as was concluded in previous chapters.

Regarding the execution of individual methods, some conclusions can be extracted:

- Constraining the warping window size value reduces the computation cost but does not always yield better accuracy, and a larger size does not ensure greater accuracy either.
- The parameter ε used by the edit distance-based similarity measures has a significant influence on the accuracy. Although this parameter can be effective for filtering noise, it also filters the small differences that characterise different classes (Fig. 8.5). Likewise, the algorithms based on qualitative representations (especially if SAX is used) obtain a poor error ratio too.

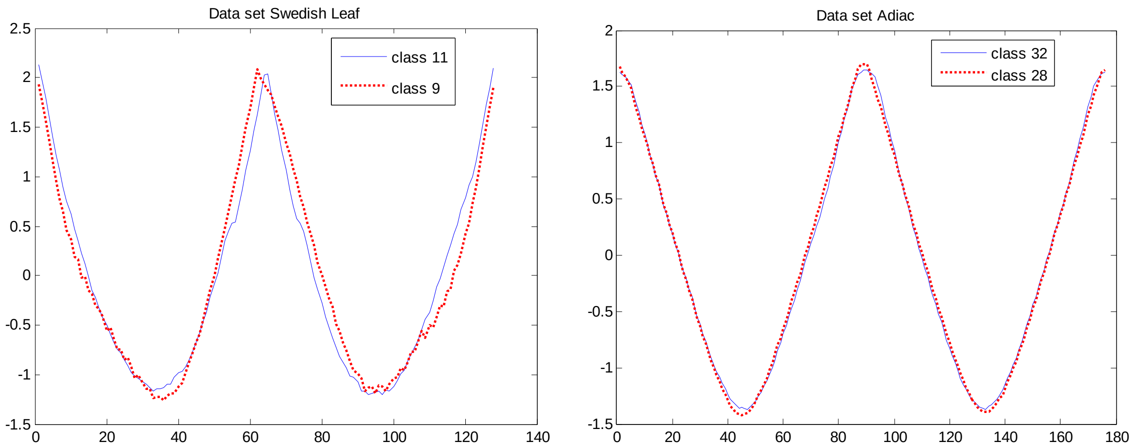


Fig. 8.5 Example of series belonging to different classes but with small differences.

Table 8.2 Error ratio of different similarity measures.

Name	Number of classes	Size of training set	Size of testing set	TS length	ED	DTW[1]	DTW$_{ONLINE}$[1]	EDR[2]	ERP[3]	LCSw[2]	EpDTW[4]	QSSI[5]
Synthetic Control	6	300	300	60	0.12	0.007 (12-197.2s)	0.013 (6-101.26s)	0.043 (3 – 1)	0.02 (6)	0.197 (3 – 1)	0.04 (10,.15,7,3)	0.026 (10,0.2,7,1)
Gun-Point	2	50	150	150	0.087	0.026 (3-48.3s)	0.027 (6-28.45s)	0.033 (3 – 0.1)	0.04 (6)	0.033 (6 – 0.1)	0.026 (3,.01,6,5)	0 (3,.015,5,0.6)
CBF	3	30	900	128	0.148	0.007 (12-296.2s)	0.002 (12-207.04s)	0.054 (6 – 1)	0.003 (12)	0.043 (6 – 1)	0.017 (10,.15,7,3)	0.019 (10,0.15,7,0.8)
Face (all)	14	560	1690	131	0.286	0.192 (3-2933s)	0.19 (6-1617s)	0.241 (6 - 1)	0.195 (3)	0.343 (3-1)	0.345 (3,0.75,5,1)	0.287 (3,0.75,5,1)
OSU Leaf	6	200	242	427	0.483	0.393 (6-2272.2s)	0.397 (6-552.7s)	0.239 (12 – 0.1)	0.405 (12)	0.401 (9-0.1)	0.219 (4,0.05,5,3)	0.198 (4,0.05,5,1)
Swedish Leaf	15	500	625	128	0.213	0.138 (3-966s)	0.184 (6-471.16s)	0.133 (3 – 0.1)	0.118 (3)	0.334 (3-0.1)	0.233 (4,0.15,4,1)	0.203 (4,0.15,5,0.8)
50Words	50	450	455	270	0.369	0.242 (6-3954.6s)	0.233 (6-760.65s)	0.23 (9-0.1)	0.257 (6)	0.345 (9-0.1)	.332 (4,0.15,3,1)	0.272 (4,0.15,3,1)
Trace	4	100	100	275	0.24	0 (6-206.12s)	0 (6-40.2s)	0.11 (3 – 0.1)	0.17 (3)	0.01 (12– 0.1)	0 (4,0.15,6,3)	0 (5,0.1,6,0)
Face (four)	4	24	88	350	0.216	0.114 (3-55s)	0.113 (6-45.4s)	0.045 (3 – 0.1)	0.079 (3)	0.045 (3– 0.1)	0.068 (4,0.5,9,1)	0.045 (4,0.5,9,0.8)
Lightning-2	2	60	61	637	0.246	0.082 (12-649.4s)	0.098 (12-379.5s)	0.164 (3–0.1)	0.131 (3)	0.147 (3 – 0.1)	0.131 (4,0.5,7,2)	0.131 (10,0.1,6,1)
Lightning-7	7	70	73	319	0.425	0.26 (6-143.2s)	0.246 (6-80.74s)	0.384 (12– 0.1)	0.287 (3)	0.384 (6 – 1)	0.301 (10,0.2,7,5)	0.246 (6,0.2,8,0.8)
ECG	2	100	100	96	0.12	0.11 (3-28.5s)	0.12 (6-17.37s)	0.11 (3–1)	0.09 (3)	0.12 (3 – 0.1)	0.17 (3,0.45,6,1)	0.11 (3,0.45,6,1)
Adiac	37	390	391	176	0.389	0.391 (3-888s)	0.401 (6-412.24s)	0.496 (3– 0.02)	0.378 (3)	0.509 (3-0.02)	0.549 (5,0.1,9,3)	0.537 (5,0.05,9,.6)
Beef	5	30	30	470	0.467	0.467 (3-37.4s)	0.467 (6-18.6s)	0.5 (3– 0.02)	0.5 (3)	0.5 (3– 0.02)	0.067 (4,0.01,4,1)	0.166 (4,0.01,4,1)
Coffee	2	28	28	286	0.25	0.179 (3-14.5s)	0.179 (6-9.38s)	0.107 (3 – 3)	0.25 (3)	0.214 (3– 3)	0.036 (10,0.1,6,1)	0.036 (10,0.1,6,1)

[1]Error ratio. (warping window width r (%) - execution time)
[2]Error ratio. (warping window width r (%) - ε)
[3]Error ratio.(warping window width r (%))
[4]Error ratio. (PAA, Slope, Alphabet, Stiffness)
[5]Error ratio. (PAA, Slope, Alphabet, Length reduction coefficient κ)

8.3.3. Improving the worst result

As was mentioned above, some data sets have very similar time series but are labelled as different classes. Thus, the resulting qualitative representation is practically identical. This fact has repercussions on a worse error ratio compared to methods based on numerical series. Then, when the time series are converted to qualitative representations the results could be improved if some type of quantitative information is added as auxiliary data. Note that this enhanced representation follows the definition of episode such as stated in eq. [3.15]. Moreover, by adding this information the parameters used for the SAX algorithm can be relaxed in order to optimise the performance of QSSI. Consequently, it is possible to shorten the sequence of episodes.

The next example shows how to improve the result obtained by the data set Adiac, which is the data set with the worst error ratio (0.537). This is because different classes have identical qualitative representation (Fig. 8.6) or very similar (Fig. 8.7).

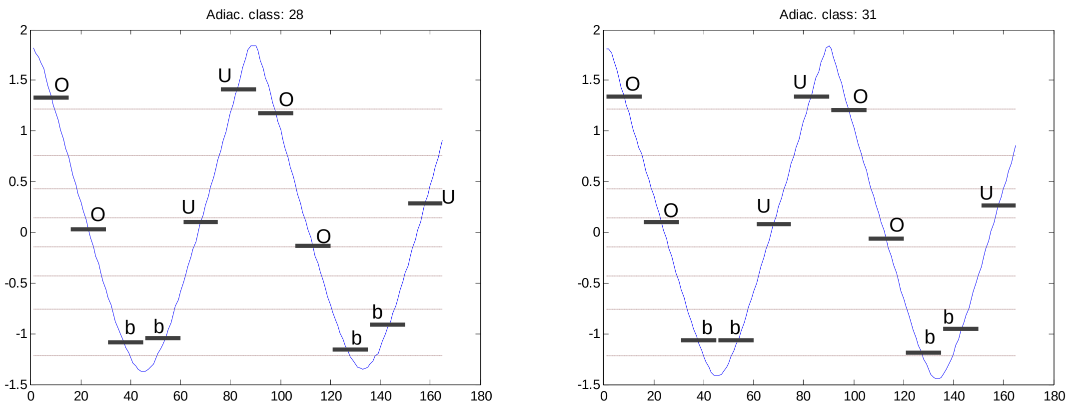


Fig. 8.6 Two classes with the same qualitative representation.

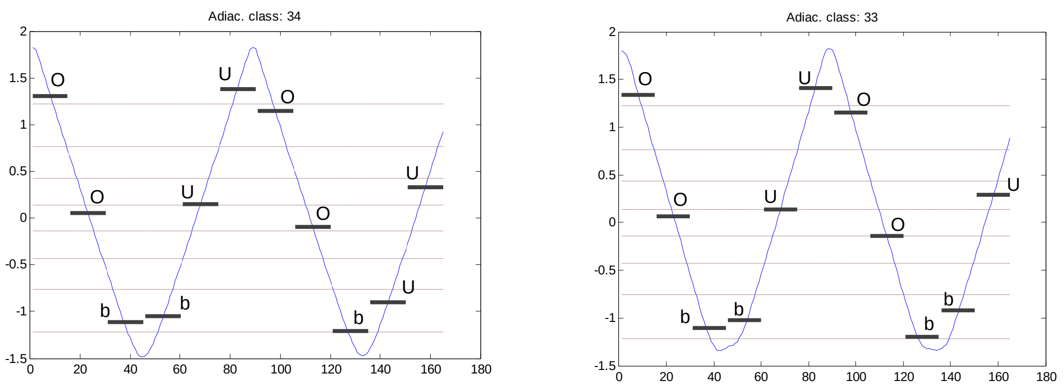


Fig. 8.7 Two classes with a very similar qualitative representation.

Now, the modified SAX algorithm adds quantitative information to each episode, such as the slope and the maximum value. As can be observed in Fig. 8.6 and Fig. 8.7, the main differences are in the maximum and minimum values of the time series. Thus, only the first two (a, b) and the last two (h, i) episodes are used to calculate the score according to their maximum values.

Therefore, the score chart used before is replaced by the chart shown in Fig. 8.8. Here, values 0.1 and 0.2 are codified values indicating episodes with auxiliary data. So, the score (or similarity) for episodes codified as 0.1 is calculated according to eq. [7.17], with the attribute being the maximum value. On the other hand, the score for episodes codified as 0.2 is calculated as a function of their slopes using the same equation.

	a	b	c	d	e	f	g	h	i	O	P	U	V
a	0.1	0.1	0	0	0	0	0	0	0	0	0	0	0
b	0.1	0.1	0.5	0	0	0	0	0	0	0	0	0	0
c	0	0.5	1	0.5	0	0	0	0	0	0	0	0	0
d	0	0	0.5	1	0.5	0	0	0	0	0	0	0	0
e	0	0	0	0.5	1	0.5	0	0	0	0	0	0	0
f	0	0	0	0	0.5	1	0.5	0	0	0	0	0	0
g	0	0	0	0	0	0.5	1	0.5	0	0	0	0	0
h	0	0	0	0	0	0	0.5	0.1	0.1	0	0	0	0
i	0	0	0	0	0	0	0	0.1	0.1	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0.2	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0.2	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0.2	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0.2

Fig. 8.8 Score chart used by QSSI and episodes containing qualitative information.

Finally, the SAX settings are established as: PAA=15, Alphabet=9, Threshold slope=0.075. The QSSI was executed using a length reduction coefficient κ of 0 and 1, giving a result of the best error ratio for κ=0 with a value of 0.378. This result matches the best value obtained in the experiment (Table 8.3), so QSSI becomes the best algorithm for this data set.

Table 8.3 Error ratio of different methods for data set Adiac. The methods have been listed in order of accuracy.

Method	QSSI, ERP	ED	DTW	EDR	LCSw	EpDTW
Error ratio	0.378	0.389	0.391	0.496	0.509	0.549

8.4. Conclusions

In this chapter, the performances of EpDTW and QSSI have been compared to other techniques and time series widely used by the data mining/machine learning community. The results indicate that, in general, QSSI is not worse than other well known methods and, in some cases, QSSI is the best option, obtaining a high level of accuracy.

Furthermore, it has also been shown that the accuracy of methods based on episodes can be improved by extending the episodes with quantitative information. In the above example QSSI became the best option, obtaining the best result while still being much faster than DTW due to the abstraction of the signal. From this result, it can be concluded that QSSI is the best similarity algorithm amongst the methods analysed.

Although similarity methods based on qualitative representations are often used in processes where they are the only solution, the results shown in this chapter suggest that their use as an alternative to numerical similarity techniques should be studied.

All the code files used for these experiments are available on the web site (Gamero, 2012), while time series data should be downloaded from Keogh et al., 2011.

Chapter 9.

Conclusions and future work

In this final chapter, a summary of the work presented in this thesis and contributions achieved are reviewed first of all. Finally, a perspective of possible future improvements is proposed.

9.1. Summary

FDD methods have been classified into two categories: model-based methods and non model-based methods. The use of one or the other category is only subject to the knowledge of process behaviour or faults. When analytical models are hardly available non model-based methods are a realistic alternative, but often expert knowledge of the process is needed. The main difficulty with knowledge-based methods is the translation of the numeric values (data coming from the process) to qualitative data (symbols) that can be used with these techniques.

Using variables as qualitative trends based on episodes reduces complexity and allows the extraction of meaningful information. So, qualitative process trends which provide an intuitive meaning about process behaviour can be used for monitoring, or could be supplied to other fault detection and diagnosis tools, expert systems or classification methods.

The usefulness of qualitative information depends on the formalism used to extract the qualitative trends. Likewise, this formalism should be adaptable to user requirements or complementary tools. Therefore, it is desirable to use a general formalism able to represent the signals based on any user-defined characteristics according to signal behaviour and supervision objectives. Moreover, in order to improve the representation usefulness, some *auxiliary* characteristics of the episodes can be used.

This issue has motivated the development of the tool Qualtras as the first contribution. This tool facilitates the abstraction of the most significant characteristics of the signals, representing any process signal by means of episodes. Each episode is composed of a time interval, a symbol describing its behaviour and a set of quantitative and/or qualitative data with additional information.

The implemented set of basic functions to detect changes between episodes requires specific tuning of parameters and thresholds to achieve the desired performance levels. These requirements require a priori knowledge of the process. However, the open architecture of Qualtras allows functions designed specifically for a particular process to be added or new techniques without, or almost without, settings. Then, if only useful characteristics are used to construct episodes, the obtained representation will be, at the same time, the most simplified and the most significant from the supervisory system point of view.

Afterwards, reasoning about process dynamics implies dealing with the evolution of variables, and the dynamics of process variables represented by sequences of episodes can be representative of specific states. Thus, the task of diagnosis can be viewed as a classification problem or a pattern recognition task. Classification by comparison and the matching of temporal sequences is an active area of research in the study of time series. However, signal comparison can be affected by several problems as differences in the length (total time) of the two signals, or in the magnitudes, or time misalignments.

A method that deals with time misalignments and different lengths is Dynamic Time Warping (DTW). DTW uses dynamic programming to align time series by stretching or shrinking them along the time axis. Since different patterns belonging to the same class can have different time duration or magnitudes, a new algorithm (EpDTW) based on qualitative representations and DTW was developed. So, the advantage of the temporal alignment produced by DTW is added to the advantage of representation by episodes, which takes into account the behaviour of the signals. In addition, qualitative representations solve the problem of the lengthy computing time when dynamic programming is used.

After this first variation of DTW, a second variation is proposed in order to adapt the original DTW to online applications. These two last algorithms constitute the second and third contribution of this thesis. However, methods based on DTW may not give an appropriate result when the sequences to be compared include unnecessary data like extra length or outliers.

In order to solve this inconvenience a new similarity index (QSSI) has been defined and implemented as the more important contribution. The idea behind the algorithm is the computation of the optimal alignment, determined by finding the pair of longest subsequences of the full sequences whose alignment yields the highest alignment score among the set of all subsequences and their possible alignments. The subsequences can include elements not related but within the aligned subsequences. These elements cannot be removed because they constitute part of the signal evolution, and hence they should also participate in the final similarity value. Finally, the algorithm returns a normalised index related to the degree of similarity between qualitative trends signals.

Generally, qualitative representations are applied in situations where other techniques are difficult or impossible to be implemented. We have demonstrated the performance of the tools created through a number of practical applications throughout this thesis. In view of the results obtained, this representation becomes another option in the field of fault diagnosis and monitoring of process data. Moreover, the algorithms developed for comparing qualitative sequences applied to dynamic systems (or time series) have been shown to be as efficient as any other existing technique, even surpassing them. The drawback of this type of representation is only the settings of data abstraction methods which represent additional effort. This is a problem to be solved with the development of new methods.

The following section summarises the results obtained.

9.2. Discussion of results

Each contribution developed throughout this thesis has been tested on an illustrative example. The following paragraphs summarise the main points.

Qualtras

The applicability of Qualtras (Gamero et al., 2009) is demonstrated in a pre-fault detection approach of a blast furnace (Gamero et al., 2006, Mora et al., 2004). The application shows how Qualtras can complement other applications to improve diagnosis strategies. This example is a real case since the software was installed at the Corus plant in the UK and operated successfully.

EpDTW

The utility of the EpDTW algorithm (Colomer et al., 2002a, Colomer et al., 2002b) is shown in the diagnosis of a level control system where the correct identification of operating situations is obtained through the comparison of the current pattern with well-known reference patterns (Colomer et al., 2003, Gamero et al., 2002). This is an application carried out in a simple pilot plant where the effectiveness of the algorithm is demonstrated even without using the length of the episodes.

A more comprehensive study is shown in Chapter 8, where the results obtained present an accuracy comparable to other methods.

Despite promising results, this method still has two significant drawbacks. First, all the elements in the alignment must be matched contributing to the final distance. And second, the endpoint constraints require that the warping path starts and finishes in diagonally opposite corner cells of the distance table. That is, it is a global alignment method.

Thus, although the accuracy of EpDTW can be improved considerably using episodes with auxiliary data, it is less accurate than QSSI. However, EpDTW is also less complex than QSSI and therefore it can be considered when time execution is very important.

DTW_{ONLINE}

The way that this algorithm is defined, it is valid only for those signals that evolve at the same time. As an example, the method has been used in order to improve the residual computation from a laboratory plant (Gamero et al., 2004, Llanos et al., 2004). This approach is especially suitable for those errors related to time distortions, therefore it will be useful for distributed systems with communication delays and for hybrid systems with on/off sensors or actuators causing misalignments between real and simulated signals.

The algorithm has also been compared to classic DTW in Chapter 8. In this case the online DTW algorithm is used in order to reduce the calculation time but this strategy is valid because a one nearest neighbour classifier is used.

QSSI

As a solution for improving the results of DTW-based techniques a new technique called QSSI has been developed. Performance of QSSI is illustrated by two application examples; the classification of voltage sags (Gamero et al., 2011) and the situation assessment of a steam generator process (to be published shortly). The accuracy of the algorithm has been validated using two different qualitative representations providing both quantitative and qualitative information. The promising classification accuracy suggests that the algorithm presented could be applied satisfactorily and confirms its utility in classification approaches.

This algorithm has also been compared to the others in Chapter 8 (to be published shortly). The results indicate that, in general, QSSI is not worse than other well-known methods and, in some cases, QSSI is the best option for obtaining high accuracy. Even when the accuracy is somewhat lower than that obtained by other algorithms, QSSI is an algorithm to take into account due to its lower execution time.

Regarding the worst values obtained by EpDTW and QSSI, they are produced because some classes have a very similar qualitative pattern. However, these values are much better if some quantitative information is included or the qualitative representation is improved. The first option is easily achieved as shown in the example of Chapter 8, while the second one involves an additional task to be carried out individually for each data set.

9.3. Recommendations for further work

With the contributions listed above, the initial objectives of this thesis have been accomplished. However, there are a few issues that require further investigation or improvements. The important ones are the following:

Qualtras

It is necessary to implement other techniques such as the online auto-tuning trend extraction method (Charbonnier and Damour, 2008) or the SAX variation used in Chapter 8.

The settings of any algorithm used to obtain an adequate qualitative representation are the main drawback of using episodes. In this sense the algorithm proposed by Charbonnier and Damour, 2008 could be an important improvement. In any case it should be considered an improvement in the configuration of the techniques employed.

Qualtras was developed in G2, a software rarely used by the scientific community. It would be desirable to transfer it to an open language for a shared evolution and to expand its capabilities to connect to the real world.

EpDTW

The algorithm could possibly be improved if the maximum local warping were applied to contiguous matched cells (in the same horizontal or vertical), instead of individual cells.

QSSI

Although the results obtained by the algorithm are very good, some aspects can be studied in order to improve performance:

- Reduce the complexity and execution time. For example, by trying to import the warping window concept to qualitative representations.
- Transfer the idea of local warping used in EpDTW to QSSI.
- New functions $p(i_k,j_k)$ for penalising the time misalignment can be studied. For example, the time misalignment could be analysed by quantifying the slope variation regarding the diagonal.
- Study the implementation of an indexing technique, so that similar items can be grouped together and also to facilitate the pruning of irrelevant data.

Appendix A.

DTW. Problem formulation

The idea of the Dynamic Time Warping problem is stated as follows: given two time series ***x*** and ***y***, of lengths *m* and *n* respectively,

$$\begin{aligned} \boldsymbol{x} &= [x_1, x_2, \cdots, x_{\mathrm{i}}, \cdots, x_{\mathrm{m}}] \\ \boldsymbol{y} &= [y_1, y_2, \cdots, y_{\mathrm{j}}, \cdots, y_{\mathrm{n}}] \end{aligned} \quad \text{[A.1]}$$

In order to align the two sequences, DTW finds a warping path **W** of *k* points in a two-dimensional *m* by *n* cost matrix where every cell value (*i*,*j*) will contain the distance (i.e., $dist(x_i,y_j) = \| x_i - y_j \|$) between points x_i and y_j. The warping path W is a sequence of contiguous elements that defines a mapping (or alignment) between ***x*** and ***y*** minimising the cumulative distance (Fig. A.1).

$$\mathbf{W} = \mathrm{w}_1, \mathrm{w}_2, ..., \mathrm{w}_{\mathrm{k}} \quad \max \; m, n \; \leq k \leq m + n \quad \text{[A.2]}$$

$$\mathrm{w}_k = \mathrm{i}_k, \mathrm{j}_k \quad \text{[A.3]}$$

where $(\mathrm{i}_\mathrm{k},\mathrm{j}_\mathrm{k})$ corresponds to the k^{th} grid element in the warping path.

If the warping path passes through a cell (*i*,*j*) in the cost matrix, it means that the i^{th} point in time series ***x*** is warped to the j^{th} point in time series ***y***. Note that where there are vertical sections of the warping path, a single point in the time series ***y*** is warped to multiple points in time series ***x***, and the opposite is also true where the warping path is a horizontal line. Since a single point may map to multiple points in the other time series, the time series do not need to be of equal length.

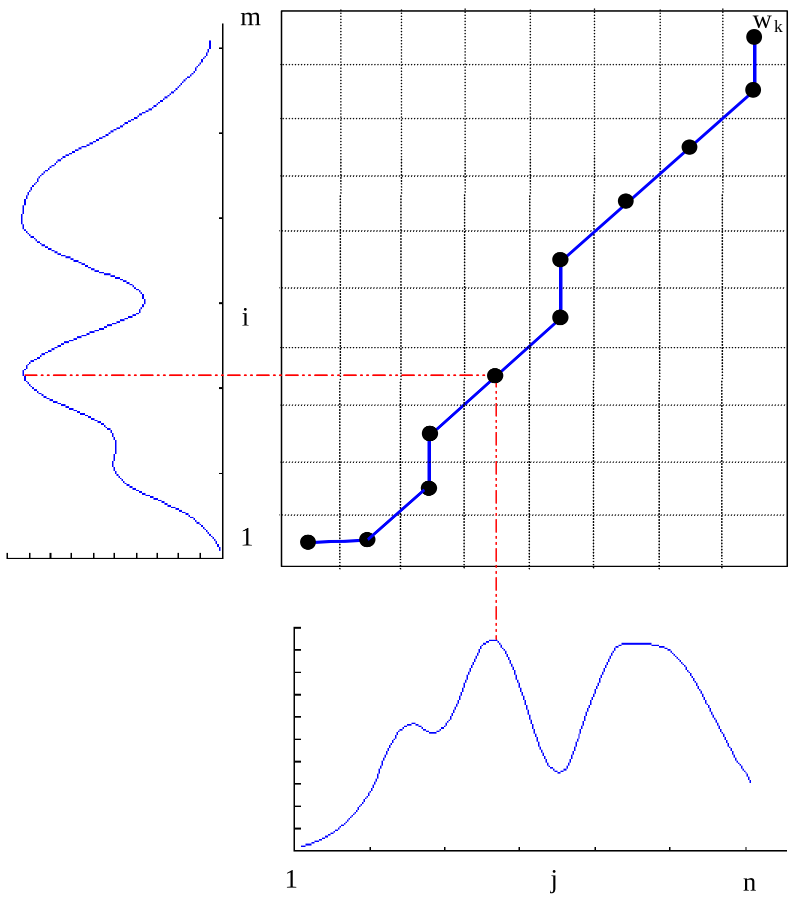


Fig. A.1 Cost matrix and the warping path W traced through it from (1,1) to (m,n).

The warping path is typically subject to several constraints:

Boundary conditions: the warping path should start at w_1=[1,1] and end at w_k=[m,n].

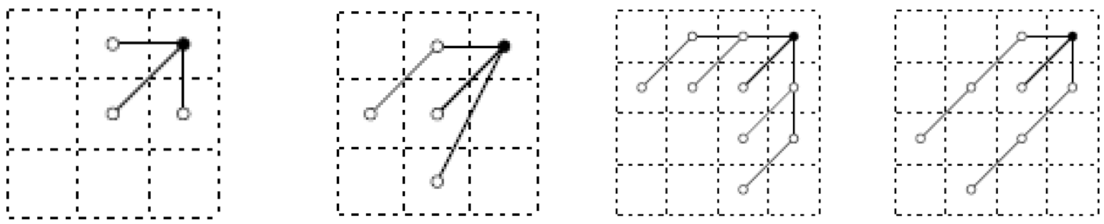
Monotonicity and continuity: given two grid elements in a warping path, (i_k, j_k) and (i_{k+1},j_{k+1}), then $1 \geq i_{k+1} - i_k \geq 0$ and $1 \geq j_{k+1} - j_k \geq 0$. This restricts the allowable transitions of a node to adjacent elements, that is, the path is not allowed back in time ($i_{k+1} \geq i_k$ and $j_{k+1} \geq j_k$).

Slope constraints: this condition is carried out as a restriction on the possible relation between several consecutive points on the warping function (excessive expansion or compression). It can be avoided by not allowing the local slope of the path to exceed a specified range (Fig. A.2).

Global path constraint: defines the region of grid elements that are searched for the optimal warping path (Fig. A.3). Intuitively, the grid constrains the warping path in a global sense by limiting how far it may stray from the diagonal (Itakura, 1975; Sakoe and Chiba, 1978). The aim of global constraints is to limit the number of cells that are evaluated in the cost matrix and speed up the DTW calculation. For example, the Sakoe-Chiba band illustrated in Fig. A.3 can be defined as follows:

$$\forall (i_k, j_k) \in \mathrm{W}, \quad i_k - r \leq j_k \leq i_k + r \qquad \text{[A.4]}$$

Where r is the width of the warping window.



(a) (b)

Fig. A.2 (a) Trivial case of no constraint. (b) Some examples of slope constraints.

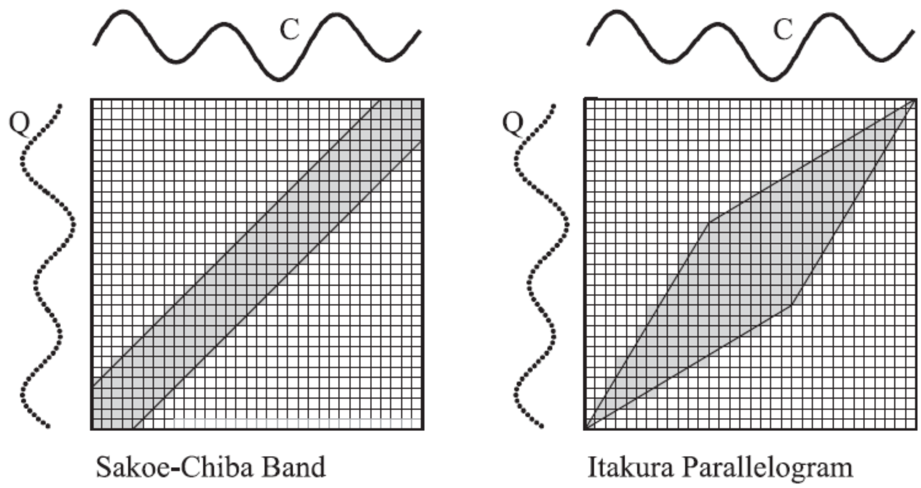


Fig. A.3 Global constraints limit the scope of the warping path, restricting them to the gray areas. The two most common constraints in the literature are the Sakoe-Chiba band and the Itakura parallelogram.

As there are many warping paths, the search is resolved by means of dynamic programming. Then, the path is extracted by evaluating the cumulative distance $\gamma(i,j)$ as the sum of the distance $dist(x_i,y_j)$ found in the current cell and the minimum of the cumulative distances of the adjacent elements:

$$\gamma(i,j) = \min \begin{Bmatrix} 2 \cdot dist(x_i, y_j) + \gamma(i-1, j-1) \\ dist(x_i, y_j) + \gamma(i-1, j) \\ dist(x_i, y_j) + \gamma(i, j-1) \end{Bmatrix} \quad [A.5]$$

$$\gamma(1,1) = 2 \cdot dist(x_1, y_1)$$

In this case the equation uses a weight of 2 for diagonal steps. It is also possible to find the above expression using the smaller weight of 1 (eq. [A.6]), the optimal path is prioritised and diagonal local transitions are preferred over horizontal or vertical ones producing smaller distortions of the time axes.

$$\gamma(i,j) = dist(x_i, y_j) + \min \begin{Bmatrix} \gamma(i-1, j-1) \\ \gamma(i-1, j) \\ \gamma(i, j-1) \end{Bmatrix} \quad [A.6]$$

$$\gamma(1,1) = dist(x_1, y_1)$$

Practically, the optimal warping path can be found by tracing backward from $\gamma(m,n)$ towards $\gamma(1,1)$. At each cell the previous neighbouring cell with minimum cumulative distance is chosen.

Finally, the normalised total distance is defined as:

$$dist(\boldsymbol{x}, \boldsymbol{y}) = \underset{W}{Min} \left[\frac{\sum_{k=1}^{K} dist(i_k, j_k) * w(k)}{N(w)} \right] \quad [A.7]$$

where ***w(k)*** is a nonnegative weighting function and *N(w)* is a normalisation factor which is a function of ***w(k)***. The value of *N(w)* depends on the type of weighting function ***w(k)*** used. The most common weighting function for a symmetric algorithm is:

$$\boldsymbol{w}(k) = (i_k - i_{k-1}) + (j_k - j_{k-1}) \quad [A.8]$$

Thus, $\boldsymbol{w}(k)$ weights show a local transition from the $(k-1)^{th}$ path point to the k^{th} path point, according to the number of horizontal and vertical steps (both equivalents) that need to be taken for that particular local transition. Then:

$$N(\boldsymbol{w}) = \sum_{k=1}^{K} \boldsymbol{w}(k) = m + n \qquad \text{[A.9]}$$

where m and n are the longitudes of sequences X and Y respectively. This is true if eq. [A.5] is used, while for eq. [A.6] the normalisation factor $N(\boldsymbol{w})=K$.

Appendix B.

Example of the QSSI algorithm

Given two string sequences ***S*** and ***Q***:

$$
\begin{aligned}
S &= \langle s, f, g, s, f, g, t \rangle \\
Q &= \langle g, t, s, f, g \rangle
\end{aligned}
\qquad [B.1]
$$

First, the alignment matrix **M** is initialised by scoring cells with similar pairs (Fig. B.1a). In this example the score is 1 if the episodes are the same and 0 if they are different. Each coloured cell in the alignment matrix constitutes a match. In parallel, a direction matrix **Md** containing zeros is constructed (Fig. B.1b).

	s	f	g	s	f	g	t
g	0	0	1	0	0	1	0
t	0	0	0	0	0	0	1
s	1	0	0	1	0	0	0
f	0	1	0	0	1	0	0
g	0	0	1	0	0	1	0

(a)

	s	f	g	s	f	g	t
g	0	0	0	0	0	0	0
t	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0
g	0	0	0	0	0	0	0

(b)

Fig. B.1 (a) Alignment matrix M initialised with matches and (b) initial direction matrix Md.

After initialisation, the alignment matrix is filled beginning with the last column and row (Fig. B.2), and then by operating successive summations on the cells from the end of the sequences toward the origin according to eq. [7.10] (Fig. B.3).

Regarding the values of the direction matrix, in addition to the three directions shown in Fig. 7.3, a special value v=0 indicates a possible candidate to finish the pathway **W**.

	s	f	g	s	f	g	t
g	0	0	1	0	0	1	0
t	0	0	0	0	0	0	1
s	1	0	0	1	0	0	0
f	0	1	0	0	1	0	0
g	0	0	1	0	0	1	0

(a)

	s	f	g	s	f	g	t
g	0	0	0	0	0	0	3
t	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0
f	0	0	0	0	0	0	0
g	0	2	0	0	2	0	0

(b)

Fig. B.2 The alignment matrix (a) and direction matrix are filled beginning with the ends of sequences.

	s	f	g	s	f	g	t
g	4	5	6	3	2	3	0
t	4	4	5	4	1	0	1
s	5	3	4	5	2	0	0
f	2	3	1	2	3	0	0
g	0	0	1	0	0	1	0

(a)

	s	f	g	s	f	g	t
g	1	1	1	3	2	1	3
t	3	1	1	3	3	1	0
s	1	2	2	1	3	1	0
f	2	1	2	2	1	1	0
g	0	2	0	0	2	0	0

(b)

Fig. B.3 (a) Completed alignment matrix, the arrows show the path W. (b) Direction matrix, the arrows show how the path W is obtained.

Now, the necessary pathway **W** through **M** is defined as one which begins at a cell with maximum value and with similarity between their items. In the example, the cell (1,3) has a value of 6. Then, starting at w_1= (1,3) the path is obtained by a backtrace through the alignment matrix and guided by the recorded directions of propagation. That is, the cell (1,3) in the direction matrix has a value k=1 (diagonal movement), so the next cell is (2,4). This procedure is repeated until a cell with direction 0 is reached and the alignment is finished (Fig. B.3b). According to these steps the complete path yields **W**=[(1,3),(2,4),(3,4),(4,5),(5,6)], and eliminating those cells without similarity, **W**=[(1,3),(3,4),(4,5),(5,6)] as the optimal path. The maximum-match pathway then, is that pathway for which the sum of the assigned cell values (less any penalty factors) is largest.

The next step is to choose the optimal temporal alignment according to diagonals. Since the alignment is determined by minimising the distance to diagonals, only diagonals containing matches are evaluated. The set of diagonals in the example is Dg=[1, 2] and the values obtained by equations [7.4] and [7.5] are 1 and 3 respectively. So the optimal diagonal is the one labelled '1'.

Fig. B.4 displays from left to right the two alignments derived from these diagonals. Intuitively, one can see that diagonal 1 produces a better temporal alignment. The time misalignment, represented by oblique lines in Fig. B.4, is produced by the existence of symbols without a correspondence in the other sequence. Due to their physical significance, these non-matched elements cannot be ignored. Nevertheless, the similarity between two sequences is penalised by the existence of time misalignments.

Fig. B.4 Alignments produced by diagonals 1 and 2.

Finally, the QSSI value is calculated. Table B.1 reports the values for the two possible diagonals. The final normalised similarity is 0.5, and the percentage of aligned subsequence is 57.14% and 100% for ***S*** and ***Q*** respectively.

Table B.1 Example of calculation of QSSI values.

	$p(i_k, j_k) = \frac{1}{2^{\lvert j-i-dg \rvert}}$ and $p_0=1$	
Diagonal	*dg*=1	*dg*=2
$\sum_{k=1}^{K} p(i_k, j_k)$	1*0.5+3*1=3.5	1*1+3*0.5=2.5
max(m,n, $\mathfrak{K}$)*p_0	7*1	
QSSI(S,T)	3.5/7=0.5	2.5/7=0.35

Now the example is extended using episodes:

$$\mathbb{S}=\langle s1,f1,g3,s2,f2,g1,t1\rangle \qquad \mathbb{Q}=\langle g3,t1,s1,f1,g1\rangle$$

The length of the episodes does not change the path but can vary the optimal diagonal since different durations act as weights. Thus, the similarity index could be altered (Table B.2). In this case the new diagonal is 2 and the temporal alignment is shown in

Fig. B.5. Now the final normalised similarity is 0.59, and the percentage of aligned subsequence is 72.72% and 100% for *S* and *Q* respectively.

Table B.2 Calculation of QSSI values for *dg*=2.

	$p(\mathrm{i}_k, \mathrm{j}_k) = \frac{1}{2^{\lvert j-i-dg \rvert}}$ and p_0=1
Diagonal	*dg*=2
$\sum_{k=1}^{K} p(\mathrm{i}_k, \mathrm{j}_k)$	3*1+3*0.5+2*1=6.5
max(m,n, $\mathfrak{K}$)*p_0	max(11,7,8)=11
QSSI(S,T)	6.5/11=0.59

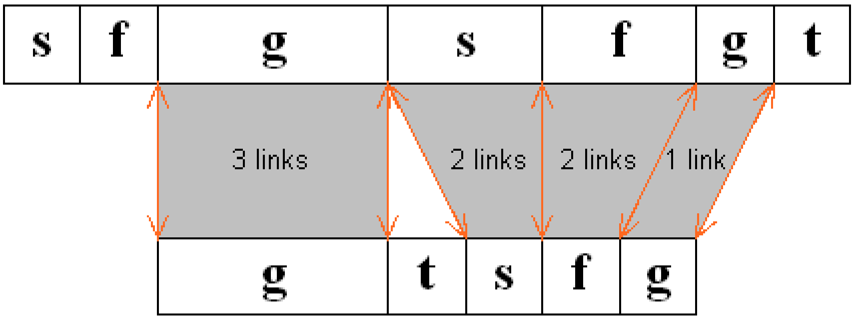


Fig. B.5 Alignment produced by diagonal 2 in sequences of episodes.

Finally, consider the following two sequences:

$$\mathcal{S}=\langle f4,v1,g1,f7,g4\rangle \qquad \mathcal{Q}=\langle v1,F5,g5\rangle$$

The local similarity chart between episodes is defined in Fig. B.6.

EPIS		*f*	*F*	*g*	*v*
	f	1	0.5	0	0
	F	0.5	1	0	0
	g	0	0	1	0
	v	0	0	0	1

Fig. B.6 Local similarity chart. Episodes f and F are qualitatively similar but their slopes are different.

The complete alignment matrix **M** and direction matrix **Md** are presented in Fig. B.7. Graphically, the alignment is shown in Fig. B.8.

	f4	v1	g1	f7	g4
v1	9	10	9	8	0
F5	5	7	8	9	0
g5	0	0	1	0	4

(a)

	f4	v1	g1	f7	g4
v1	2	1	1	3	0
F5	2	2	2	1	3
g5	0	2	0	2	0

(b)

Fig. B.7 (a) Completed alignment matrix, the arrows show the path W. (b) Direction matrix, the arrows show how the path W is obtained.

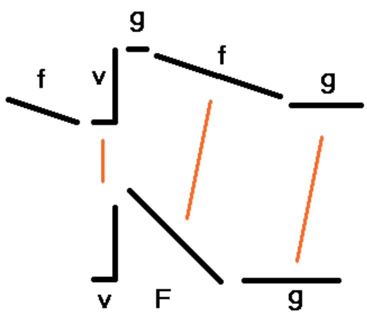


Fig. B.8 Alignment obtained by QSSI.

For these sequences, the QSSI value is 0.485 and 0.634 using global and reduced alignment as the normalisation option respectively.

Appendix C.

A method for improving the classification rate

Although this is a subject out of the scope of this thesis, this appendix shows a method for improving the classification rate when a normalised similarity measure is obtained. The method is illustrated by following the application of section 7.3.

In the example, a voting strategy using an adaptive *k*-nearest neighbour search algorithm is followed. However, the closeness degree of similarity between the test and the retrieved cases is not taken into account. As result of majority rule, the mean value of the success rate is about 93.1%.

An alternative approach is to define a decision variable. This definition represents the likelihood of a specific class in a test case. We define an adaptive/exponential decision variable DV_{aX}^{HV} as:

$$DV_{aX}{}^{HV} = \frac{\sum_{C_i^A} Sim(C_i^{HV}, T)^r}{\sum_{i=1}^{k} Sim(C_i, T)^r} \qquad [C.1]$$

C_i^{HV} are the cases in $C_1, \ldots, C_k$ which belong to class HV, and $Sim(C_i,T)$ is the similarity between a case C_i and the test case T, where similarity values are normalised between 0 and 1. Regarding the exponent value r, it could be dependent on heuristics according to the number of NN for each class. In this example, because of the reduced and fixed number of retrieved cases, r has adopted the minimum number of k-NN retrieved, so, r is set to 3. That is,

$$DV_{aX}^{HV} = \frac{\sum_{C_i^A} Sim(C_i^{HV},T)^3}{\sum_{i=1}^{k} Sim(C_i,T)^3} \quad [C.2]$$

Then, the decision variable is compared against a given threshold τ. Whenever the current value of the decision variable equals or goes beyond this threshold, the test case T is classified as belonging to class HV or not.

As an example, Table C.1 shows some representative situations and values returned by DV_{aX}^{HV}. Note that DV_{aX}^{HV} classifies the test cases correctly because it captures the closeness of the similarity to the test case involved in the selected class.

The third situation is a special test where $Sim(C_1^{HV},T)$ returns a complete similarity while the second case of the same class returns a very low similarity. This last case penalises the DV value, however DV_{aX}^{HV} yields a high value, as is expected.

Table C.1 Different situations and values returned by DV_{aX}^{HV} .

	$Sim(C_1^{HV},T)$	$Sim(C_2^{HV},T)$	$Sim(C_3^{MV},T)$	$Sim(C_4^{MV},T)$	DV_{aX}^{HV} / T class (τ=0.5)
Ex. 1	0.8	0.8	0.5	0.5	0.80 / HV
Ex. 2	0.3	0.3	0.6	0.6	0.11 / MV
Ex. 3	1	0.1	0.4	0.4	0.88 / HV

Finally, Table C.2 shows the classification results according to the majority rule and using the decision variable (τ=0.5) for the whole set of substations.

Regarding the misclassifications when using the decision variable, Table C.3 shows the returned cases for each one, T_1 (substation C) and T_2 (substation I). Both tests belong to class MV, but they have been classified as HV. Similarities returned for T_1 prove there is no room for precision on the decision variable. On the other hand, a posterior revision revealed that the misclassification in T_2 was due to a labelling mistake in the utility.

Thus, when the *k*-NN algorithm is complemented by the decision variable the success rate achieved is close to 100%. This is because this definition takes into account the closeness degree of similarity between the test and the retrieved cases.

Table C.2 Classification results.

			Majority rule		Decision variable	
		# sags	HV	MV	HV	MV
Substation A	sags HV	48	47	1	48	
	sags MV	45	1	44		45
Substation B	sags HV	20	20		20	
	sags MV	16	2	14		16
Substation C	sags HV	58	58		58	
	sags MV	24	3	21	1	23
Substation D	sags HV	35	35		35	
	sags MV	27	8	19		27
Substation E	sags HV	33	33		33	
	sags MV	23	1	22		23
Substation F	sags HV	24	24		24	
	sags MV	25		25		25
Substation G	sags HV	17	17		17	
	sags MV	7		7		7
Substation H	sags HV	38	38		38	
	sags MV	33	6	27		33
Substation I	sags HV	23	23		23	
	sags MV	32	5	27	1	31

Table C.3 Retrieved cases for misclassifications using the decision variable.

	Sim / Class	Sim / Class	Sim / Class	Sim / Class	Sim / Class	DV_{aX}^{HV}
T_1 Subst. C	0.759 / HV	0.426 / HV	0.407 / HV	0.769 / MV	0.419 / MV	0.524
T_2 Subst. I	0.944 / HV	0.666 / HV	0.659 / HV	1 / MV		0.587

References

Aach, J. and Church, G. (2001). Aligning gene expression time-series with time warping algorithms. Bioinformatics 17 pp.: 495-508.

Afonso, F.; Barbosa, F. and Rodrigues, A. (2011). SimTraj: An Approach to Similar Queries over Trajectories in Metric Spaces. Proceedings of GeoComputation 2011, University College London, UK.

Agrawal, R.; Faloutsos, C. and Swami, A. (1993). Efficient similarity search in sequence databases. Conf. on Foundations of Data Organization and algorithms. 69-84

Agrawal, R.; Psaila, G.; Wimmers, E. L. and Zaït, M. (1995). Querying shapes of histories. Proc. of 21st VLDB Conf., Switzerland. 502-514

Alcorta García, R. and Frank, P. (1997). Deterministic nonlinear observer-based approaches to fault diagnosis: a survey. Control Engineering Practice 5(5) pp.: 663-670.

Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W. and Lipman, D. J. (1990). Basic local alignment search tool. J. Mol. Bio. 215 pp.: 403-410.

Allison, L. (1993). Normalisation of affine gap costs in optimal sequence alignment. J. Theor. Biol. 161(2) pp.: 263–269.

Angeli, C. (2008). Online expert systems for fault diagnosis in technical processes. Expert Systems 25(2) pp.: 115-132.

Antunes, C. M. and Oliveira, A. L. (2001). Temporal data mining: an overview, San Francisco, USA.

ASM. (n.d.). "The Abnormal Situatiom Management (ASM) Consortium." from http://www.asmconsortium.net.

Ayrolles, L. (1996). Abstraction Temporelle et Interpretation Quantitative/Qualitative de Procesus a Dynamiques Multiples. Toulouse (France), Universite Paul Sabatier. These de Doctorat.

Bakshi, B. R.; Locher, G. and Stephanopoulos, G. (1994). Analysis of Operating Data for Evaluation, Diagnosis and Control of Batch Operations. Journal of Process Control 4(4) pp.: 179-194.

Bakshi, B. R. and Stephanopoulos, G. (1994a). Representation of process trends part III. Multiscale extraction of trends from process data. Computers & Chemical Engineering 18 pp.: 267-302.

Bakshi, B. R. and Stephanopoulos, G. (1994b). Representation of process trends part IV. Induction of real-time patterns from operating data for diagnosis and supervisory control. Computers & Chemical Engineering 18 pp.: 303-332.

Bar-Joseph, Z.; Gerber, G.; Gifford, D.; Jaakkola, T. and Simon, I. (2002). A new approach to analyzing gene expression time-series data. 39-48

Basseville, M. (1988). Detecting Changes in signals and systems - a survey. Automatica 24(3) pp.: 309-326.

Beard, R. V. (1971). Failure Accommodation in Linear Systems Through Reorganization, Mass.Inst. Techno.

Beckmann, N.; Kriegel, H. P.; Schneider, R. and Seeger, B. (1990). The R*-Tree: an efficient and robust access method for points and rectangles. ACM SIGMOD Conf. 322-331

Bellman, R. (1961). Adaptive Control Processes: A Guided Tour, Princeton University Press.

Bellman, R. and Stuart, D. (1962). Applied Dynamic Programming. Princeton, New Jersey.

Bierman, G. M. (2000). Using XML as an Object Interchange Format, Department of Computer Science, University ofWarwick.

Blanke, M.; Kinnaert, M.; Lunze, J. and M.Staroswiecki (2006). Diagnosis and Fault-Tolerant Control, 2nd Edition. Germany, Springer Verlag.

Bollen, M. H. J. (2000). Understanding Power Quality problems. New York, 2000, IEEE Press.

Boschen, F. J. and Weise, C. (2003). What starts inflation: Evidence from the OECD countries. Journal of Money Credit and Banking 35(3) pp.: 323-349.

Bozkaya, T.; Yazdani, N. and Ozsoyoglu, M. (1997). Matching and indexing sequences of different lengths. In Proc. of the Sixth Int. Conf. on Information and Knowledge Management (CIKM'97), November 10-14, 1997, Las Vegas, Nevada.

Bregón, A.; Simón, M. A.; Rodríguez, J. J.; Alonso, C.; Pulido, B. and Moro, I. (2006). Early Fault Classification in Dynamic Systems Using Case-Based Reasoning. Lecture Notes in Computer Science. 4177/2006: 211-220.

Caiani, E.; Porta, A.; Baselli, G.; Turiel, M.; Muzzupappa, S.; Pieruzzi, F. and Crema, C. (1998). Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on the left ventricular volume. IEEE Computers in Cardiology pp.: 73-76.

Capitani, P. and Ciaccia, P. (2006). Warping the time on data streams. Data & Knowledge Engineering doi:10.1016/j.datak.2006.08.012.

Carreira-Perpiñán, M. Á. (2001). Continuous latent variable models for dimensionality reduction and sequential data reconstruction, University of Sheffield, UK.

Cauvin, S. and Celse, B. (2004). CHEM: Advanced Decision Support Systems for Chemical/Petrochemical Process Industries. ESCAPE-14 Congress, Lisbon. Portugal.

Colomer, J. (1998). Representació Qualitativa Asíncrona de Senyals per a la Supervisió Experta de Processos, University of Girona (UdG),Catalonia, Spain. Ph.D. dissertation.

Colomer, J. and Meléndez, J. (2001). A family of FIR differentiators based on a polynomial least squares estimation. Proc. Of the European Control Conference. 2802-2807

Colomer, J.; Melendez, J. and Gamero, F. (2003). A Qualitative Case-Based Approach for situation assessment in Dynamic Systems. Application in a two tank system. . SAFEPROCESS 2003, Washington (USA).

Colomer, J.; Meléndez, J. and Gamero, F. I. (2002a). Pattern recognition based on episodes and DTW. Application to diagnosis of a level control system. 16th International Workshop on Qualitative Reasoning, Sitges, Spain, 2002. N. Angell and J. A. Ortega (eds). 37-43

Colomer, J.; Meléndez, J. and Gamero, F. I. (2002b). Qualitative representation of process trends for situation assessment based on cases. 15th IFAC World Congress, Barcelona, 2002.

Cormen, T.; Leiserson, C.; Rivest, R. and Stein, C. (2001). Introduction to Algorithms. Cambridge, MA.

Cuberos, F. J.; Ortega, J. A.; Gasca, R. M. and Toro, M. (2002). QSI – Qualitative Similarity Index. QR2002 16th International Workshop on Qualitative Reasoning, Sitges, Spain, 2002. 45-51

Chakrabarti, K. and Mehrotra, S. (1999). The hybrid tree: An index structure for high dimensional feature spaces. ICDE. 440-447

Chan, F. K.-P.; Fu, A. W.-C. and Yu, C. (2003). Haar wavelets for efficient similarity search of time-series: with and without time warping. IEEE Transactions on Knowledge and Data Engineering 15(3) pp.: 686-705.

Chan, K. and Fu, W. (1999). Efficient time series matching by watching wavelets. Proc. of the 15th IEEE Int. Conf. on Data Engineering. 126-133

Charbonnier, S. and Damour, D. (2008). A robust auto-tuning on -line trend extraction method. Proc. of the 17th IFAC World Congress, Vol. 17, Part 1. Korea, South.

Charbonnier, S. and Gentil, S. (2007). A trend-based alarm system to improve patient monitoring in intensive care units. Control Eng. Practice 15 pp.: 1039-1050.

Chen, J. and Patton, R. J. (1999). Robust model-based fault diagnosis for dynamic systems, Kluwer Academic Publishers.

Chen, L. and Ng, R. T. (2004). On The Marriage of Lp-norms and Edit Distance. Thirtieth international conference on Very large data VLDB '04, 04. VLDB Endowment. 792-803

Chen, L.; Özsu, M. T. and Oria, V. (2005). Robust and fast similarity search for moving object trajectories. Proceedings of the 2005 ACM SIGMOD international conference on Management of data, Baltimore, Maryland. 491-502

Chen, Q.; Chen, L.; Lian, X.; Liu, Y. and Yu, J. X. (2007). Indexable PLA for efficient similarity search. Proceedings of the 33rd international conference on Very large data bases. Vienna, Austria pp.: 435-446.

Cheung, J. T. and Stephanopoulos, G. (1990). Representation of Process Trends , parts I and II. Computers & Chemical Engineering 14 pp.: 495-540.

Chow, E. Y. and Willsky, A. S. (1980). Issues in the development of a general algortihm for reliable failure detection, Albuquerque,USA.

Chow, E. Y. and Willsky, A. S. (1984). Analytical redundancy and the design of robust failure detection systems. IEEE Trans. on Automatic Control AC-29 pp.: 603-614.

Chu, S.; Keogh, E.; Hart, D. and Pazzani, M. (2002). Iterative deepening Dynamic Time Warping for time series. The 2nd SIAM International Conference on Data Mining.

Das, G.; Gunopulos, D. and Minnila, H. (1997). Finding similar time series. PKDD '97 Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery. 88-100

Dash, S.; Maurya, M. R. and Venkatasubramanian, V. (2004). A novel interval-halving framework for automated identification of process trends. AIChE Journal 50(1) pp.: 149-162.

Dash, S.; Rengaswamy, R. and Venkatasubramanian, V. (2003). Fuzzy-logic based trend classification for fault diagnosis of chemical processes. Computers & Chemical Engineering 27 pp.: 347-362.

Dash, S. and Venkatasubramanian, V. (2000). Challenges in the industrial applications of fault diagnostic systems. Computers & Chemical Engineering 24 pp.: 785-791.

Daw, C. S.; Finney, C. E. A. and Tracy, E. R. (2003). A review of symbolic analysis of experimental data. Review of Scientific Instruments 74(2) pp.: 916-930.

Delmaire, G.; Cassar, J. P. and Staroswiecki, M. (1994). Identification and parity space techniques for failure detection in siso systems including modelling errors. Proceedings of the 33rd IEEE Conference on Decision and Control.

Ding, H.; Trajcevski, G.; Scheuermann, P.; Wang, X. and Keogh, E. (2008). Querying and mining of time series data: experimental comparison of representations and distance measures. Proceedings of the VLDB Endowment, 1. 1542-1552

Ding, S. (2008). Model-based fault diagnosis techniques: design schemes, algorithms, and tools, Springer Verlag.

Ding, X. and Jeinsch, T. (1999). An approach to analysis and design of observer and parity relation based FDI systems. 14th IFAC World Congress.

Dixon, S. (2005). Live Tracking of Musical Performances using On-Line Time Warping. Proceedings of the 8th International Conference on Digital Audio Effects (DAFx05), Madrid, Spain. 92-97

Djokic, S. Z.; Milanovic, J. V.; Chapman, D. J. and McGranaghan, M. F. (2005). Shortfalls of Existing Methods for Classification and Presentation of Voltage Reduction Events. IEEE Trans. on Power Delivery 20(2) pp.: 1640-1649.

Dong, D. and McAvoy, T. J. (1996). Nonlinear principal component analysis based on principal curves and neural networks. Computers & Chemical Engineering 20 pp.: 65-78.

Dunia, R. and Qin, S. J. (1998). Joint diagnosis of process and sensor faults using principal component analysis. Control Engineering Practice 6(4) pp.: 457-469.

Efthimiadu, I.; Tham, M. T.; Adgar, A. and Cox, C. S. (1995). Integrating Statistical and Engineering Control Techniques. Measurement and Control 28(3) pp.: 78-82.

Falkenhainer, B. and Forbus, K. (1991). Compositional modeling: finding the right model for the job. Artificial Intelligence 51 pp.: 95–143.

Faloutsos, C.; Ranganathan, M. and Manolopoulos, Y. (1994). Fast subsequences matching in time-series databases. Proceedings of the 1994 ACM SIGMOD Int. Conf. on Management of Data, 1994, Mineapolis, USA. 419-429

Feng, H. and Wah, C. C. (2003). Online signature verification using a new extreme points warping technique. Pattern Recognition Letters 24 pp.: 2943-2951.

Flehmig, F.; Watzdorf, R. V. and Marquardt, W. (1998). Identification of trends in process measurements using the wavelet transform. Computers & Chemical Engineering 22 pp.: 491-496.

Forbus, K. (1996). Qualitative Reasoning, CRC Handbook of Computer Science.

Fourie, S. H. and de Vaal, P. (2000). Advanced process monitoring using an on-line nonlinear multiscale principal component analysis methodology. Computers & Chemical Engineering 24 pp.: 755-760.

Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - A survey and some new results. Automatica 26 pp.: 459-474.

Frank, P. M. (1996). Analytical and qualitative model-based fault diagnosis - A survey and some new results. European Journal of Control 2 pp.: 6-28.

Frank, P. M.; Ding, S. X. and Koppen-Seliger, B. (2000). Current developments in the theory of FDI. IFAC Symposium Safeprocess. 16-27

Frank, P. M. and Ding, X. (1997). Survey of Robust Residual Generation and Evaluation Methods in Observer-based Fault Detection Systems. Journal of Process Control 7(6) pp.: 403-424.

Frank, P. M. and Koppen-Seliger, B. (1997). New developments using AI in fault diagnosis. Eng. Applications of Artificial Intelligence 10(1) pp.: 3-14.

Fried, R.; Bernholt, T. and Gather, U. (2006). Repeated median and hybrid filters. Computational statistics & data analysis 50 pp.: 2313-2338.

Fu, A. W.-c.; Leo, E. K.; Lau, Y. H. and Ratanamahatana, C. A. (2005). Scaling and Time Warping in Time Series Querying. Proc. of the 31st international conference on Very large data bases VLDB 2005. VLDB Endowment. 649-660

Fu, T.-c. (2011). A review on time series data mining. Eng. Appl. Artif. Intell. 24(1) pp.: 164-181.

Fu, T. C.; Chung, F. L.; Luk, R. and Ng, C. M. (2008). Representing financial time series based on data point importance. Engineering Applications of Artificial Intelligence 21(2) pp.: 277–300.

Gaede, V. and Günther, O. (1988). Multidimensional access methods. ACM Computing Surveys 30(2).

Galati, D. G. and Simaan, M. A. (2006). Automatic decomposition of time series into step, ramp, and impulse primitives. Pattern Recognition 39(11) pp.: 2166-2174.

Gamero, F. I. (2012). from http:\\eia.udg.edu\~gamero\TestCh8.

Gamero, F. I.; Colomer, J. and Meléndez, J. (2002). Situation assessment based on cases. A qualitative approach for cases representation and retrieval. Iberamia, Sevilla (Spain). November, 2002.

Gamero, F. I.; Colomer, J.; Meléndez, J. and Berjaga, X. (2009). Qualtras: a Tool for Qualitative Trend Representations. 20th International Workshop on Principles of Diagnosis, DX'09, Stockholm. 187-194

Gamero, F. I.; Colomer, J.; Meléndez, J. and Warren, P. (2006). Predicting Aerodynamic Instabilities in a Blast Furnace. Eng. Applications of Artificial Intelligence 19(1) pp.: 101-111.

Gamero, F. I.; Llanos, D. A.; Colomer, J. and Meléndez, J. (2004). Residual Generation and Evaluation using Dynamic Time Warping. IMAACA'04 - The International Conference on Integrated Modeling & Analysis in Applied Control & Automation. (Part of International Mediterranean Modeling Multiconference, I3M 2004), Genoa (Italy), 2004.

Gamero, F. I.; Meléndez, J. and Colomer, J. (2011). QSSI: A new similarity index for qualitative time series. Application to classify voltage sags. Applied Artificial Intelligence 25(2).

Ge, Z. and Song, Z.-H. (2008). Latent and Small Fault Detection and Diagnosis for Dynamic Processes. Proceedings of the 17th IFAC World Congress, 17. COEX, Korea, South. 2576-2581

Gentil, S. (1996). Intelligence Artificialle pour la Surveillance des Procédés Continus. Surveillance des systèmes continus. Ecole d'Eté d'Automatique de Grenoble Tome 1.

Gertler, J. (1988). Survey of model-based failure detection and isolation in complex plants. IEEE Control Systems Magazine 8(6) pp.: 3-11.

Gertler, J. (1995). Diagnosing parametric faults-from identification to parity relations, Seattle, WA. 1615-1620

Gertler, J. (1998). Fault Detection and Diagnosis in Engineering Systems. New York, Marcel Dekker.

Gertler, J. (2000). All linear methods are equal and extendible to nonlinearities. IFAC Safeprocess, Budapest, Hungary. 52-63

Gertler, J.; Costin, M.; Fang, X.; Kowalczuk, Z.; Kunwer, M. and Monajemy, R. (1995). Model based diagnosis for automotive engines-algorithm development and testing on a production vehicle. IEEE Trans. on Control Systems Technologies 3 pp.: 61-69.

Gertler, J.; Fang, X. and Luo, Q. (1990). Detection and diagnosis of plant failures: the orthogonal parity equation approach. Control and Dynamic Systems 37 pp.: 159-216.

Gertler, J. and Singer, D. (1990). A new structural framework for parity equation-based faiulure detection and isolation. Automatica 26 pp.: 381-388.

Gertler, J. J. (1991). Analytical redundancy methods in fault detection and isolation. IFAC Symposium Safeprocess. 9-21

Geurts, P. (2001). Pattern extraction for time series classification. Proc. of PKDD 2001, 5th European Conference on Principles of Data Mining and Knowledge Discovery. 115-127

Gomm, J. B.; Weerasinghe, M. and Williams, D. (1998). Pruning and extractions of qualitative fault diagnosis rules from a neural network. Workshop on On-line Fault detection and supervision in the chemical process industries, Lyon, France.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. Journal of Molecular Biology 162 pp.: 705-708.

Guo, A. and Siegelmann, H. (2004). Time-Warped Longest Common Subsequence Algorithm for Music Retrieval. Proc. of the 5th International Conference on Music Information Retrieval (ISMIR'04), Barcelona, Spain.

Guo, J. J.; Curkendall, S. J.; Jones, J.; Fife, D.; Goehring, E. and She, D. W. (2003). Impact of cisapride label changes on condispensing of contraindicated medicaments. Pharmaceopidemiology and Drug Safety 12(4) pp.: 295-301.

Gusfield, D. (1997). Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.

Hamzah, N.; Mohamed, A. and Hussain, A. (2004). A new approach to locate the voltage sag source using real current component. J. of Electric Power Systems Research 72 pp.: 113-123.

Hetland, M. L. (2001). A Survey of Recent Methods for Efficient Retrieval of Similar Time Sequences. In Data Mining in Time Series Databases. M. Last, A. Kandel, and H. Bunke, Eds. Singapore: World Scientific.

Himmelblau, D. M. (1978). Fault detection and diagnosis in chemical and petrochemical processes. New York, Elsevier Scientific Publishing Company.

Hoskins, J. C. and Himmelblau, D. M. (1988). Artificial neural network models of knowledge representation in chemical engineering. Computers & Chemical Engineering 12(9) pp.: 881-890.

Huang, Y. W. and Yu, P. S. (1999). Adaptive query processing for time-series data. Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and DataMining. 282–286

Hung, N. Q. V. and Anh, D. T. (2007). Combining SAX and Piecewise Linear Approximation to Improve Similarity Search on Financial Time Series. Information Technology Convergence, 2007. ISITC 2007.

Isaza, C.; Orantes, A.; Kempowsky-Hamon, T. and Lann, M.-V. L. (2009). Contribution of fuzzy classification for the diagnosis of complex systems. Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain. 1132–1139

Isermann, R. (1984). Process Fault Detection Based on Modeling and Estimation Methods. Automatica 20(4) pp.: 387-404.

Isermann, R. (1997). Supervision fault-detection and fault-diagn methods - An introduction. Control Engineering Practice 5(5) pp.: 639-652.

Isermann, R. (2005). Model-based fault-detection and diagnosis–status and applications. Annual Reviews in control 29(1) pp.: 71–85.

Isermann, R. (2006). Fault-diagnosis systems: an introduction from fault detection to fault tolerance, Springer Verlag.

Isermann, R. and Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. Control Engineering Practice 5(5) pp.: 709-719.

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

Izadi-Zamanabadi, R. and Staroswiecki, M. (2000). A structural analysis method formulation on fault-tolerant control system design. Proc. of the 39th IEEE Conference on Decision and Control, Sydney, Australia.

Janusz, M. and Venkatasubramanian, V. (1991). Automatic generation of qualitative description of process trends for fault detection and diagnosis. Eng. Applications of Artificial Intelligence 4 pp.: 329-339.

Ji, X.; Li-Ling, J. and Sun, Z. (2003). Mining gene expression data using novel approach based on hidden Markov models. FEBS 542 pp.: 125-131.

Jolliffe, I. T. (1986). Principal Component Analysis. New York, Springler-Verlag.

Jonsson, H. A. and Badal, Z. (1997). Using signature files for querying time-series data. Proc. of the First European Symposium on Principles and Practice of Knowledge Discovery in Databases. 211–220

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

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

Katipamula, S. and Brambley, M. R. (2005). Methods for fault detection, diagnostics, and prognostics for building systems - A review, Part I. HVAC&R Research 11(1) pp.: 3-25.

Kempowsky, T.; Subias, A. and Aguilar-Martín, J. (2006). Process situation assessment: From a fuzzy partition to a finite state machine. Eng. Applications of Artificial Intelligence 19 pp.: 461-477.

Kempowsky, T.; Subias, A.; Aguilar-Martín, J. and Le Lann, M. V. (2005). Online continuous process monitoring by means of a finite state machine generated using learning techniques. 18th International Congress on Condition Monitoring and Diagnostic Engineering Management (COMADEM'2005), Cranfield (GB), 2005. 221-231

Keogh, E. (1997). A fast and robust method for pattern matching in time series databases.

Keogh, E. (2002). Exact indexing of dynamic time warping. Proceedings of the 28th international conference on Very Large Data Bases, Hong Kong, China. 406-417

Keogh, E.; Chakrabarti, K.; Mehrotra, a. S. and Pazzani, M. (2001a). Locally adaptive dimesionality reduction for indexing large time-series databases. Proceedings of ACM SIGMOD. pp.: 151-162

Keogh, E.; Chakrabarti, K.; Pazzani, M. and Mehrotra, S. (2001b). Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowledge and Information Systems J. 3(3) pp.: 263-286.

Keogh, E.; Chu, S.; Hart, D. and Pazzani, M. (2003). Segmenting Time Series: A Survey and Novel Approach. Data Mining in Time Series Databases. World Scientific Publishing Company, 2003.

Keogh, E. and Pazzani, M. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. Proceedings of the Fourth conference on Knowledge Discovery in Databases and Data Mining, New York. 239-241

Keogh, E. and Pazzani, M. (1999). Scaling up dynamic time warping to massive datasets. PKDD '99 Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery, Prague, Czech Republic. Springer-Verlag London, UK.

Keogh, E. and Pazzani, M. (2000a). Scaling up dynamic time warping to massive datasets. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, Massachusetts, United States. ACM - New York. 285-289

Keogh, E. and Pazzani, M. (2000b). A simple dimensionality reduction technique for fast similarity search in large time series databases. Fourth Pacific- Asia Conference on Knowledge Discovery and Data Mining, Kyoto, Japan.

Keogh, E. and Pazzani, M. (2001). Derivative Dynamic Time Warping. First SIAM Int.l Conference on Data Mining (SDM'2001), Chicago, USA.

Keogh, E. and Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. Knowledge and Information Systems 7(3) pp.: 358-386.

Keogh, E.; Zhu, Q.; Hu, B.; Y., H.; Xi, X.; Wei, L. and Ratanamahatana, C. A. (2011). "The UCR Time Series Classification/Clustering Homepage: www.cs.ucr.edu/~eamonn/time_series_data/."

Kim, S.-W.; Park, S. and Chu, W. W. (2001). An index-based approach for similarity search supporting time warping in large sequence databases. Proc. of the 17th International Conference on Data Engineering. 607-614

Kinnaert, M. (2003). Fault diagnosis based on analyticals models for linear and nonlinear systems - a tutorial., Washington DC, USA. 38-50

Kleer, J. and Kurien, J. (2003). Fundamentals of model-based diagnosis, Washington DC, USA. 25-36

Ko, M. H.; West, G.; Venkatesh, S. and Kumar, M. (2005). Online context recognition in multisensor systems using dynamic time warping. Conference on Intelligent Sensors, Sensor Networks and Information Processing. 283-288

Koivo, H. N. (1994). Artificial neural networks in fault diagnosis and control. Control Engineering Practice 2(1) pp.: 89-101.

Konstantinov, K. and Yoshida, T. (1992). Real-Time Qualitative Analysis of the Temporal Shapes of (Bio) Process Variables. AIChE Journal 38(11) pp.: 1703-1715.

Korn, F.; Jagadish, H. and Faloutsos, C. (1997). Efficiently supporting ad hoc queries in large datasets of time sequences. Proc. ACM SIGMOD Int. Conf. on Management of Data. 289-300

Kourti, T. and MacGregor, J. F. (1996). Multivariate SPC Methods for Process and Product Monitoring. J. Quality Technology 28 pp.: 409–428.

Kramer, M. A. (1992). Autoassociative neural networks. Computers & Chemical Engineering 16(4) pp.: 313-328.

Krzanowski, W. J. (1979). Between-group comparison of principle components. J. Am. Stat. Assoc. 74(367) pp.: 703-707.

Kuipers, B. (1986). Qualitative simulation. Artificial Intelligence 29(3) pp.: 289–338.

Kuipers, B. (2001). Qualitative simulation. Encyclopedia of Physical Science and Technology, Third Edition. E.-i.-C. R. A. Meyers pp.: 287-300.

Lapp, S. A. and Powers, G. A. (1977). Computer-aided synthesis of fault trees. IEEE Trans. on Reliability 26(1) pp.: 2-13.

Lei, H. and Govindaraju, V. (2004). Regression Time Warping for Similarity Measure of Sequence. CIT'04, pp. 826-830.

Li, R. F. and Wang, X. Z. (2001). Qualitative/Quantitative simulation of process temporal behavior using clustered fuzzy digraphs. AICHE Journal, 2001 47(4) pp.: 906-919.

Li, Y.; Wen, C. L.; Xie, Z. and Xu, X. H. (2004). Synchronisation of batch trajectory based on multi-scale dynamic time warping. 2403-2408

Liggesmeyer, P. and Rothfelder, M. (1998). Improving system reliability with automatic fault tree generation. Proc. 28th Annual Fault Tolerant Computing Symposium, Munich, June 1998. 90-99

Lin, J.; Keogh, E.; Lonardi, S. and Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, CA. June 13. ACM. New York, USA.

Lin, J.; Keogh, E.; Lonardi, S. and Patel, P. (2002). Finding Motifs in Time Series. In proc. of the 2nd Workshop on Temporal Data Mining, at the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada. July 23-26, 2002.

Lin, J.; Keogh, E.; Wei, L. and Lonardi, S. (2007). Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Discov. 15(2) pp.: 107-144.

Lipman, D. J. and Pearson, W. R. (1985). Rapid and sensitive protein similarity searches. Science 227 pp.: 1435-1441.

Liu, L.; Zuo, W.; Zhang, D.; Li, N. and Zhang, H. (2010). Classification of Wrist Pulse Blood Flow Signal Using Time Warp Edit Distance. ICMB 2010, LNCS 6165. 137–144

Lu, N.; Wang, F. and Gao, F. (2003). Combination Method of Principal Component and Wavelet Analysis for Multivariate Process Monitoring and Fault Diagnosis. Industrial & Engineering Chemistry Research 42(18) pp.: 4198-4207.

Llanos, D.; Mora, J.; Meléndez, J.; Colomer, J.; Corbella, X. and Sánchez, J. (2003). Abstraction of significant temporal features of voltage sags recorded in a 25kV substation. International Conference on Renewable Energy and Power Quality (ICREPQ '03), Vigo (ESP).

Llanos, D. A.; Gamero, F. I.; Colomer, J. and Meléndez, J. (2004). Residual Computation using Dynamic Time Warping. Proc. of World Automation Congress WAC'04, 16. Seville (Spain), 2004. 143-148

Mäckel, O. and Rothfelder, M. (2001). Challenges and solutions for fault tree analysis arising from automatic fault tree generation: some miletones on the way. ISAS-SCI 1 pp.: 583-588.

Mallat, S. and Zhong, S. (1992). Characterization of signals from multiscale edges. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(7) pp.: 710-721.

Manders, E.-J.; Mosterman, P. J. and Biswas, G. (1999). Signal to symbol transformation techniques for robust diagnosis in TRANSCEND. Working Papers of the Tenth Intl. Workshop on Principles of Diagnosis, Loch Awe, Scotland, June 1999. 155-165

Mangoubi, R. S. (1998). Robust estimation and failure detection, Springer-Verlag.

Marteau, P.-F. (2009). Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching. IEEE Trans. Pattern Analysis and Machine Intelligence 31 pp.: 306–318.

Martin, E. B. and Morris, A. J. (1996). An Overview of Multivariate Statistical Process Control in Continuous and Batch Process Performance Monitoring. Trans. of the Inst. of Meas. and Control 18 pp.: 51-60.

Maurya, M. R.; Paritosh, P. K.; Rengaswamy, R. and Venkatasubramanian, V. (2010). A framework for on-line trend extraction and fault diagnosis. Engineering Applications of Artificial Intelligence 23(6) pp.: 950-960.

Maurya, M. R.; Rengaswamy, R. and Venkatasubramanian, V. (2003). Fault diagnosis by qualitative trend analysis of the principal components: prospects and some new results. In proc. of the 5th IFAC symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS-2003), Washington D.C., USA, June 9-11, 2003. 861-866

Maurya, M. R.; Rengaswamy, R. and Venkatasubramanian, V. (2007). Fault diagnosis using dynamic trend analysis: A review and recent developments. Eng. Applications of Artificial Intelligence 20 pp.: 133-146.

Medjaher, K.; Samantaray, A. K.; Bouamama, B. O. and Staroswiecki, M. (2006). Supervision of an industrial steam generator. Part II: Online implementation. Control Engineering Practice 14(1) pp.: 85-96.

Megalooikonomou, V.; Wang, Q.; Li, G. and Faloutsos, C. A. (2005). A Multiresolution Symbolic Representation of Time Series. ICDE ‘05, pp. 668-679.

Meléndez, J. and Colomer, J. (2001). Episodes representation for supervision. Application to diagnosis of a level control system. Workshop on Principles of Diagnosis DX’01, Sansicario (Italy). 119-126

Mendonça, L.; Sousa, J. and Sà da Costa, J. (2009). An architecture for fault detection and isolation based on fuzzy methods. Expert Systems with Applications 36(2) pp.: 1092-1104.

Minnen, D.; Isbell, C.; Essa, I. and Starner, T. (2007). Detecting subdimensional motifs: an efficient algorithm for generalized multivariate pattern discovery. ICDM 2007. 601-606

Mo, K. J.; Shin, D. S. and Chang, K. S. (1998). Process monitoring using the clustering method and functional link associative neural networks. Proc. of the Foundations on Computer Aided Process Operations (FOCAPO 98), Snowbird, Utah (USA).

Moller-Levet, C.; Cho, S. and Wolkenhauer, O. (2003a). Microarray data clustering based on temporal variation: Fcv and tsd preclustering. Applied Bioinformatics 2(1) pp.: 35–45.

Moller-Levet, C. S.; Cho, K.-H.; Yin, H. and Wolkenhauer, O. (2003b). Clustering of gene expression time series data. Tech. Rep., Systems Biology and Bioinformatics Group. University of Rostock, Germany. http://www.sbi.uni-rostock.de/publications.htm.

Moon, Y. S.; Whang, K. Y. and Han, W. S. (2002). GeneralMatch: A subsequence matching method in time-series databases based on generalized windows. Proc. of the ACM SIGMOD Int. Conf. on Management of Data, Madison, Wisconsin. 382-393

Mora, J.; Colomer, J.; Meléndez, J.; Gamero, F. and Warren, P. (2004). Hybrid approach based on temporal representation and classification techniques used to determine unstable conditions in a blast furnace. . Lecture Notes in Computer Science. Springer-Verlag Heidelberg 3040 pp.: 404-414.

Mylaraswamy, D. (1996). DKit: A Blackboard-based Distributed, Multi-Expert Environment for Abnormal Situation Management. West Lafayette, Purdue University.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48(3) pp.: 443-453.

Nielsen, N.-P. V.; Carstensen, J. M. and Smedsgaard, J. (1998). Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. Journal of Chromatography A 805 pp.: 17-35.

Nomikos, P. and MacGregor, J. F. (1994). Monitoring batch process using multiway principle component analysis. AIChEJ 40(8) pp.: 1361-1375.

Nong, Y., Ed. (2003). The Handbook of Data Mining. Mahwah, New Jersey, Lawrence Erlbaum Associates.

Notohardjono, B. D. and Ermer, D. S. (1986). Time-series control charts for correlated and contaminated data. J. Eng. Ind-T. ASME 108(3) pp.: 219-226.

Oppenheim, V. and Schafer, R. (1975). Digital Signal Processing, Prentice Hall, N.J.

Ougiaroglou, S.; Nanopoulos, A.; Papadopoulos, A. N.; Manolopoulos, Y. and Welzer-Druzovec, T. (2007). Adaptive k -Nearest-Neighbor Classification Using a Dynamic Number of Nearest Neighbors. ADBIS 2007, Springer-Verlag. Berlin, Heidelberg. 4690 pp.: 66-82.

Oyeleye, O. O. (1989). Qualitative modeling of continuous chemical processes and applications to fault diagnosis, MIT. Sc.D. Thesis.

Park, S.; Chu, W. W.; Yoon, J. and Hsu, C. (2000). Efficient Searches for Similar Subsequences of Different Lengths in Sequence Databases. Proceedings of the 16th International Conference on Data Engineering.

Paterson, M. S. and Dancik, V. (1994). Longest common subsequences. Proceedings of 19th Int. Symposium Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 841, 1994. 127-142

Patton, R. J.; Fantuzzi, C. and Simani, S. (2003). Model-based fault diagnosis in dynamic systems using identification techniques, Springer-Verlag.

Patton, R. J.; Frank, P. and Clark, R. (1989). Fault diagnosis in dynamic systems, theory and application, Prentice Hall.

Patton, R. J.; Lopez-toribio, C. J. and Uppal, F. J. (1999). Artificial intelligence approaches to fault diagnosis for dynamic systems. Int. J. Appl. Math. and Comp. Sci. 9(3) pp.: 471-518.

Pau, L. F. (1981). Failure diagnosis and performance monitoring. New York, Marcel Dekker.

Pekalska, E.; Duin, R. P. W. and Paclik, P. (2006). Prototype Selection for Dissimilarity-Based Classifiers. Pattern Recognition 39(2) pp.: 189-208.

Perng, C. S.; Wang, H.; Zhang, S. R. and Parker, D. S. (2000). Landmarks: a new model for similarity-based pattern querying in time series databases

Proc. IEEE ICDE. 33-42

Popivanov, I. and Miller, R. J. (2002). Similarity search over time-series data using wavelets. ICDE '02 Proceedings of the 18th International Conference on Data Engineering. 212-221

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

Puig, V.; Quevedo, J.; Escobet, T. and Pulido, B. (2004). On the integration of fault detection and isolation in model-based fault diagnosis. 15th International Workshop on principles of diagnosis DX, 2004. 227-232

Puig, V.; Quevedo, J.; Escobet, T. and Tornil, S. (2002). Model based fault diagnosis of dynamic processes: comparing FDI and DX methodologies. IV Jornadas de

ARCA: Sistemas Cualitativos y Diagnosis, Vilanova i la Geltrú, Barcelona, Catalonia, Spain, 2002. 91-99

Rafiei, D. and Mendelzon, A. (1997). Similarity based-queries for time-series data. Proc. ACM SIGMOD Conf. 13-25

Rajshekhar; Gupta, A.; Samanta, A.; Kulkarni, B. and Jayaraman, V. (2007). Fault Diagnosis Using Dynamic Time Warping. Pattern Recognition and Machine Intelligence. A. Ghosh, R. De and S. Pal, Springer Berlin / Heidelberg. 4815: 57-66.

Ramaker, H.-J.; Van Sprang, E. N. M.; Boelens, H. F. M.; Westerhuis, J. A. and Smilde, A. K. (2003). Dynamic time warping of spectroscopic BATCH data. Analytica Chimica Acta 498 pp.: 133-153.

Ramil, J. F. and Smith, N. (2002). Qualitative simulation of models of software evolution. Journal of Software Process: Improvement and Practice 7(3-4) pp.: 95-112.

Ramoni, M. F.; Sebastiani, P. and Kohane, I. S. (2002). Cluster analysis of gene expressions dynamics. PNAS 99(14) pp.: 9121-9126.

Ratanamahatana, C. A. and E., K. (2004). Making Time-series Classification More Accurate Using Learned Constraints. In proc. of SIAM International Conference on Data Mining (SDM '04), Lake Buena Vista, Florida, April 22-24, 2004. 11-22

Rengaswamy, R. (1995). A framework for integrating process monitoring, diagnosis and supervisory control. Purdue University. Doctoral Thesis.

Rengaswamy, R.; Hagglund, T. and Venkatasubramanian, V. (2001). A qualitative shape analysis formalism for monitoring control loop performance. Eng. Applications of Artificial Intelligence 14(1) pp.: 23–33.

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

Rengaswamy, R. and Venkatasubramanian, V. (2000). A fast neural network and its updation for incipient fault detection and diagnosis. Computers & Chemical Engineering 24 pp.: 431-437.

Rognes, T. (2001). ParAlign: a parallel sequence algorithm for rapid and sensitive database searches. Nucleic Acids Research 29 pp.: 1647-1652.

Rognes, T. and Seeberg, E. (1998). SALSA: improved protein database searching by a new algorithm for assembly of sequence fragments into gapped alignments. Bioinformatics 14(10) pp.: 839-845.

Ruiz, D.; Nougues, J. M. and Puigjaner, L. (2001). Fault diagnosis support systems for complex chemical plants. Computers & Chemical Engineering 25 pp.: 151-160.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoustics, Speech, and Signal Proc. ASSP-26(1) pp.: 43-49.

Sakurai, Y.; Yoshikawa, M. and Faloutsos, C. (2005). FTW: Fast similarity search under the time warping distance. Proc. of the 24th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'05), Baltimore, MD, USA.

Salvador, S. and Chan, P. (2007). Toward Accurate Dynamic Time Warping in Linear Time and Space. Intell. Data Analysis 11(5) pp.: 561-580.

Sankoff, D. and Kruskal, J. (1983). Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, Addison Wesley. Reading, Massachusetts.

Sellers, P. H. (1974). An algorithm for the distance between two sequences. J. of Combinatorics Theory 16 pp.: 253-258.

Shatkay, H. and Zdonic, S. (1996). Approximate queries and representation for large data sequences. 12th International Conference on Data Engineering. 546-553

Shou, Y.; Mamoulis, N. and Cheung, D. W. (2005). Fast and Exact Warping of Time Series Using Adaptive Segmental Approximations. Machine Learning 58(2-3) pp.: 231-267.

Silverman, H. F. and Morgan, D. P. (1990). The application of dynamic programming to connected speech recognition. IEEE Signal Processing Magazine 7 pp.: 6-25.

Singhal, A. and Seborg, D. E. (2001). Matching patterns from historical data using PCA and distance similarity factors. In Proc. of the American Control Conference, Arlington, VA June 25-27, 2001.

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

Singhal, A. and Seborg, D. E. (2002b). Pattern Matching in Multivariate Time Series Databases Using A Moving Window Approach. Industrial & Engineering Chemistry Research 41 pp.: 3822-28.

Singhal, A. and Seborg, D. E. (2006). Evaluation of a pattern matching method for the Tennessee Eastman challenge process. Journal of Process Control 16 pp.: 601–613.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. J Mol Biol. 147(1) pp.: 195-197.

Somervuo, P. J. (2004). Online algorithm for the self-organizing map of symbol strings. Neural Networks 17 pp.: 1231-1239.

Srinivasan, R. and Qian, M. S. (2005). Off-line temporal signal comparison using singular points augmented time warping. Industrial & Engineering Chemistry Research 44(13) pp.: 4697-4716.

Srinivasan, R. and Qian, M. S. (2006). Online fault diagnosis and state identification during process transitions using dynamic locus analysis. Chemical Engineering Science 61 pp.: 6109-6132.

Srinivasan, R.; Wang, C.; Ho, W. K. and Lim, K. W. (2005). Neural networks systems for multi-dimensional temporal pattern classification. Computers & Chemical Engineering 29 pp.: 965-981.

Staroswiecki, M.; Cassar, J. P. and Declerck, P. (2000). A structural framework for the design of fdi systems in large scale industrial plants, Springer-Verlag.

Staroswiecki, M. and Comtet-Varga, G. (2001). Analytic redundancy relations for fault detection and isolation in algebraic dynamic systems. Automatica 37(5) pp.: 687-699.

Stephanopoulos, G.; Locher, G.; Duff, M. J.; Kamimura, R. and Stephanopoulos, G. (1997). Fermentation Database Mining by Pattern Recognition. Biotechnology and Bioengineering 53(5) pp.: 443–452.

Struzik, Z. R. and Siebes, A. (1999). The Haar Wavelet Transform in the Time Series Similarity Paradigm. Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery, Springer-Verlag pp.: 12-22.

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

Tanaka, Y.; Iwamoto, K. and Uehara, K. (2005). Discovery of time series motif from multi-dimensional data based on MDL principle. Machine Learning 58 pp.: 269-300.

Tarifa, E. and Scenna, N. (1997). Fault diagnosis, direct graphs, and fuzzy logic. Computers & Chemical Engineering 21 pp.: 649-654.

Tilman, D. and Wedin, D. (1991). Oscillations and chaos in the dynamics of a perennial grass. Nature 353(6345) pp.: 653-655.

Tomasi, G.; Van den Berg, F. and Andersson, C. (2004). Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. Journal of Chemometrics 18 pp.: 231-241.

Tsuge, Y.; Hiratsuka, K.; Takeda, K. and Matsuyama, H. (2000). A fault detection and diagnosis for the continuous process with load fluctuations using orthogonal wavelets. Computers & Chemical Engineering 24 pp.: 761-767.

Tversky, A. and Gati, I. (1982). Similarity, separability, and the triangle inequality. Psychological Review 89(2) pp.: 123-154.

Vahdatpour, A.; Amini, N. and Sarrafzadeh, M. (2009). Toward Unsupervised Activity Discovery Using Multi Dimensional Motif Detection in Time Series. Twenty-first International Joint Conference on Artificial Intelligence (IJCAI) 2009, Pasadena, California.

Vedam, H. and Venkatasubramanian, V. (1997). A Wavelet Theory-Based Adaptive Trend Analysis System for Process Monitoring and Diagnosis. Proc. of the American Control Conference, Piscataway/NJ, USA. 309–313

Vedam, H.; Venkatasubramanian, V. and Bhalodia, M. (1998). A B-Spline based method for data compression, process monitoring and diagnosis. Computers & Chemical Engineering 22 pp.: 827–830.

Venkatasubramanian, V. and Chan, K. (1989). A neural network methodology for process fault diagnosis. AIChE Journal 35(12) pp.: 1993-2002.

Venkatasubramanian, V.; Rengaswamy, R.; Yin, K. and Kavuri, S. H. (2003a). A review of process fault detection and diagnosis, Part I: Quantitative model-based methods. Computers & Chemical Engineering 27 pp.: 293-311.

Venkatasubramanian, V.; Rengaswamy, R.; Yin, K. and Kavuri, S. H. (2003b). A review of process fault detection and diagnosis, Part II: Qualitative models and search strategies. Computers & Chemical Engineering 27 pp.: 313-326.

Venkatasubramanian, V.; Rengaswamy, R.; Yin, K. and Kavuri, S. H. (2003c). A review of process fault detection and diagnosis, Part III: Process history based methods. Computers & Chemical Engineering 27 pp.: 327-346.

Venkatasubramanian, V.; Vaidyanathan, R. and Yamamoto, U. (1990). Process Fault Detection and Diagnosis using neural networks - I Steady State processes. Computers & Chemical Engineering 14(7) pp.: 699-712.

Villez, K.; Keser, B. and Rieger, L. (2009). Qualitative representation of trends (QRT) as a tool for automated data-driven monitoring of on-line sensors. 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona (Spain). 1276-1281

Villez, K.; Rosén, C.; Anctil, F.; Duchesne, C. and Vanrolleghem, P. A. (2008). Qualitative representation of trends: An alternative approach to process diagnosis and control. Water Science & Technology 57(10) pp.: 1525–1532.

Vlachos, M.; Gunopulos, D. and Das, G. (2004). Indexing time-series under conditions of noise. Data Mining In Time Series Databases. A. K. M. Last, and H. Bunke Eds., World Scientific. 57: 67-100.

Vlachos, M.; Hadjieleftheriou, M.; Gunopulos, D. and Keogh, E. (2003). Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures. In the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Washington, DC, USA. August 24 - 27, 2003. 216-225

Vullings, H. J. L. M.; Verhaegen, M. H. G. and Verbruggen, H. B. (1998). Automated ECG segmentation with dynamic time warping. Proceedings of the 20th Int. Conf. Of the IEEE Engineering in Medicine and Biology Society 20(1) pp.: 163-166.

Watanabe, K.; Matsura, I.; Abe, M.; Kubota, M. and Himmelblau, D. M. (1989). Incipient Fault Diagnosis of chemical processes via artificial neural networks. AIChE Journal 35(11) pp.: 1803-1812.

Waterman, M. S. and Eggert, M. (1987). A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. J. Mol. Biol. 197 pp.: 723-728.

Weber, R.; Schek, H.-J. and Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. VLDB '98. 194-205

Wilcox, N. A. and Himmelblau, D. M. (1994). The possible cause and effect graphs (PCEG) model for fault diagnosis - I. Methodology. Comp. Chem. Eng. 18(2) pp.: 103-116.

Wilson, D. R. and Martinez, T. R. (1997). Instance Pruning Techniques. ICML'97, Morgan Kaufmann, pp. 403-411.

Williams, B. C. (1986). Doing Time: Putting qualitative reasoning on firmer ground. Proc. of AAAI-86, National Conf. on Artif. Intel. pp.: 105-112.

Willsky, A. S. (1976). A Survey of Design Methods for Failure Detection in Dynamic Systems. Automatica 12 pp.: 601-611.

Wong, J. C.; McDonald, K. A. and Palazoglu, A. (2001). Classification of abnormal operation using multiple process variable trends. Journal of process control 11 pp.: 409-418.

Wong, T. S. F. and Wong, M. H. (2003). Efficient subsequence matching for sequences databases under time warping. Seventh International Database Engineering and Applications Symposium (IDEAS'03). 139-148

Wu, Y.; Agrawal, D. and Abbadi, A. E. (2000). A comparison of DFT and DWT based similarity search in time-series databases. Proc. of the 9th International Conference on Information and Knowledge Management, McLean, Virginia, United States. 488-495

Xi, X.; Keogh, E.; Shelton, C. and Wei, L. (2006). Fast time series classification using numerosity reduction. Proc. of the 23nd International Conference on Machine Learning, Pittsburgh.

Xia, B. (1997). Similarity Search in Time Series Data Sets. School of Computing Science, Simon Praser University. M.S. Thesis.

Yamanaka, F. (1997). Application of the intelligent alarm system for the plant operation. Computers & Chemical Engineering 21, suppl. pp.: 625-630.

Yamashita, Y. (2006). An automatic method for detection of valve stiction in process control loops. Control Engineering Practice 14(5) pp.: 503-510.

Yi, B. K. and Faloutsos, C. (2000). Fast time sequence indexing for arbitrary Lp norms. 26th Intl. Conf. on Very Large Databases, Cairo. 285-394

Yi, B. K.; Jagadish, H. and Faloustsos, C. (1998). Efficient retrieval of similar time sequences under time warping. 23-27

Yoon, S. and MacGregor, J. F. (2004). Principal-component analysis of multiscale data for process monitoring and fault diagnosis. AIChE Journal 50(11) pp.: 2891-2903.

Yu, D. L.; Gomm, J. B. and Williams, D. (1999). Sensor fault diagnosis in a chemical process via RBF neural netwoks. Control Engineering Practice 7 pp.: 49-55.

Yum, M. K. and Kim, J. H. (2003). A very-short-term intermittency of fetal heart rates and developmental milestone. Pedriatic Research 53(6) pp.: 915-919.

Zhang, D.; Zuo, W.; Zhang, D.; Zhang, H. and Li, N. (2010). Classification of pulse waveforms using edit distance with real penalty. . EURASIP J. Adv. Signal Process 2010(Article 28).

Zhang, X.; Polycarpou, M. M. and Parisini, T. (2002). A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. Automatic Control, IEEE Transactions on 47(4) pp.: 576-593.

Zhao, J.; Chen, B. and Shen, J. (1998). Multidimensional non-orthogonal wavelet-sigmoid basis function neural network for dynamic process fault diagnosis. Computers & Chemical Engineering 23 pp.: 83-92.

Zhou, M. and Wong, M. H. (2005). A segment-wise time warping method for time scaling searching. Information Sciences 173 pp.: 227-254.