Departament d'Enginyeria Química

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

Universitat Politècnica de Catalunya

A global approach for supporting operators' decision-making dealing with plant abnormal events

Thesis

In partial fulfillment of the requirements for the degree of Doctor of Philosophy of the Universitat Politècnica de Catalunya, supervised by professors Dr. Lluis Puigjaner Corbella and Dr. Moisés Graells Sobre, in Barcelona, April 2008.

Ignacio Yélamos Ruiz

Chemical Engineer

A mis padres.

El hombre y su seguridad deben constituir la preocupación fundamental de toda aventura tecnológica. No olvidéis nunca esto cuando estéis metidos de lleno en vuestros planos y en vuestras ecuaciones.

Albert Einstein (1879-1955)

from *Era medianoche en Bhopal*, D. Lapierre and J. Moro (2001)

Summary

The high automation acquired in chemical industry during last decades has made supervision a delicate and complex task. Therefore, current plants supervision requires of sophisticated systems and tools that can create profit from the information installed modules. Thus, the precise tracking of process variables or the high plant operability, achieved by the current regulatory control, are aspects that must be contemplated when the plant has to give a global response against deviations from normal operating regime.

This thesis presents a global approach for the management of abnormal situations in chemical plants. In this proposal the complete flow of information required to respond to any nonstandard situation is considered. This global approach incorporates several key aspects: first, all the plant modules that are necessary in the faults management are presented; secondly, this thesis focuses on improving the techniques used in each of these modules so far. Lastly, synergies discovered by the proposed global approach are used to develop novel and promising solutions to address process safety and optimization difficulties.

Thus, chapter 1 establishes a first general approach to the motivations and scope of the thesis. It describes fundamental characteristics of the chemical industry and its progression during last years as well as the supervision requirements associated to this new and evolving industry. The second chapter summarizes past and current techniques applied to reduce the risk of incidences and accidents in chemical processes. This summary is primarily centered on the methodologies most commonly cited in recent literature and on those which are generally accepted in the industry. Once analyzed process supervision basis, third chapter sets out our global approach for the abnormal events management. It briefly presents the modules of the whole abnormal events management chain. These modules will be discussed in more detail in the remaining chapters.

Chapter 4 focuses on the improvement of data acquisition systems and its later treatment by means of data reconciliation with process models. Chapters 5, 6, 7 and 8 are devoted to the study of the central part of any abnormal events management system, the diagnosis module. Chapter 5 formalizes the diagnosis problem and standardizes the diagnosis systems performance evaluation indices. Chapters 6 and 7 present two new diagnosis systems based on the use of historical data. Chapter 6 describes the first FDS. It discusses

the effective diagnosis of simultaneous faults through the use of a machine learning algorithm, namely "Support Vector Machines," (SVM) by adopting a "MultiLabel" approach. The second system (chapter 7) combines a detection module based on a Principal Component Analysis (PCA) model and on a "if-then" rules diagnosis module. As a compendium of the diagnosis analysis, chapter 8 studies the strengths and weaknesses of the proposed diagnosis systems and proposes an integration of complementary diagnosis modules that surpasses the yield of anyone of the individual parts.

Chapters 9 and 10 are devoted to the support of process operators' decision-making when managing deviation from the normal plant operating regime. Chapter 9 describes a novel integration methodology of on-line and off-line process knowledge that allows generating substantial support information for the operator decision-making. Chapter 10 is also centered in the decision-making process and describes synergetic results that may be generated when integrating the abnormal events management system with other plant modules. In this chapter the global abnormal events management system incorporates an on-line optimization module. By this way, the decision making process for abnormal events does not just take into account plant safety issues but also the economic issues. Furthermore, this modules integration makes the on-line optimization technique a more trustworthy system in practice.

All chapters begin with a theoretical section followed by a second section that focuses on academic and industrial validation. Those subjects that exceed this thesis scope are commented and proposed as future work in chapter 11.

Resumen

El alto grado de automatización adquirido en las plantas químicas durante las últimas décadas hace que las tareas de supervisión sean ahora más complejas y delicadas. Esta supervisión requiere de sistemas y herramientas sofisticadas que puedan sacar provecho de los módulos de adquisición de información instalados en planta. Así, el preciso seguimiento de las variables de proceso o la fácil operatividad de los procesos, gracias a los sistemas de control regulatorio actuales, son aspectos relevantes que deben ser contemplados a la hora de dar una respuesta global a las desviaciones del régimen normal de operación.

Esta tesis presenta un enfoque global para la gestión de situaciones anormales en plantas químicas. En esta propuesta se contempla el flujo completo de información requerido para responder efectivamente a cualquier situación anormal que se pueda presentar. Mediante esta visión global, primeramente se identifican todos los módulos de planta involucrados en la gestión de fallos; luego se focalizan esfuerzos en mejorar las técnicas que estos módulos usan para su operación; por último, se aprovechan algunas de las sinergias descubiertas mediante esta visión global de la gestión de eventos anormales.

De esta forma, el primer capítulo establece un primer acercamiento general a las motivaciones y ámbito de la tesis, describiendo rasgos fundamentales en la evolución de la industria química durante los últimos años y los requerimientos asociados al nuevo modelo de supervisión. El segundo capítulo resume las técnicas y aplicaciones actuales para reducir el riesgo de incidencias y accidentes en procesos químicos. Este resumen se centra principalmente en aquellas metodologías más empleadas en la literatura y aquellas con más aceptación en ambientes industriales. Una vez analizado el estado del arte en la supervisión de procesos, se propone un enfoque global de gestión de eventos anormales en el tercer capítulo, que presenta los eslabones de la cadena de gestión de eventos anormales, los cuales serán abordados en detalle en los capítulos restantes.

De esta forma, el capítulo 4 se centra en la mejora de los sistemas de adquisición de datos y su posterior tratamiento mediante reconciliación con modelos del proceso. Los capítulos 5, 6, 7 y 8 se dedican al estudio de la parte central de cualquier sistema de respuesta a eventos anormales, el módulo de diagnóstico. El capítulo 5 formaliza el problema de diagnosis y estandariza los índices de evaluación de funcionamiento de los sistemas de

diagnóstico. Los capítulos 6 y 7 presentan dos nuevos sistemas de diagnosis basados en el uso de datos históricos. El primero, desarrollado en el capítulo 6, implementa un algoritmo de aprendizaje llamado "Máquinas de soporte vectorial" (SVM) adoptando un enfoque "MultiEtiqueta" que permite el diagnóstico eficaz de fallos simultáneos. El segundo sistema (capítulo 7) integra un módulo de detección basado en un modelo de Análisis de Componentes Principales y un módulo de diagnóstico basado en reglas "si-entonces". Como compendio de la diagnosis, el capítulo 8 estudia las fuerzas y debilidades de los sistemas de diagnóstico propuestos y propone una integración de módulos de diagnóstico complementarios que supera el rendimiento de cualquiera de los sistemas por separado.

Los capítulos 9 y 10 están centrados en la toma efectiva de decisiones frente a desviaciones del régimen normal de operación. El capítulo 9 presenta una metodología novedosa de integración de conocimiento del proceso en línea y fuera de línea, que permite generar información sustancial de soporte al operador en la toma de decisiones. El capítulo 10 se centra también en la toma de decisiones, mostrando las sinergias generadas al integrar el sistema de diagnosis con otros módulos de planta. En este capítulo el sistema global de gestión de eventos anormales es complementado con un módulo de optimización en línea. De esta forma el nuevo soporte a la toma de decisiones frente a perturbaciones no sólo tiene en cuenta aspectos relacionados con la seguridad sino también con la economía de la planta. Además, la integración permite que la técnica de optimización empleada sea más fiable en su aplicación.

Todos los capítulos incluyen una primera parte teórica seguida de una segunda parte centrada en la validación académica e industrial. Aquellos temas que exceden el alcance de esta tesis, son comentados y propuestos como trabajo futuro en el capítulo 11.

Acknowledgements

Firstly, I would like to thank Professor Lluis Puigjaner for its confidence during this thesis work. I would like to remark the excellent work environment that he created and the invaluable professional opportunities that I had under his supervision throughout this period of time. I would also want to express my gratefulness towards Sergio Ferrer, who animated me to make this thesis and who has been not only a work companion, but a partner of experiences during these years. Also I am thankful to Moisés Graells by its collaboration and guide; this thesis must a lot to their comments and reflections.

Because of the great number of people to mention, I am going to thank generally the aid and friendship found in the colleagues of doctorate and students of computer science engineering of group CEPIMA. The found atmosphere and comradeship in these four years has been crucial for the development and success of the PhD thesis as well as for my personal growing. I would like to make special mention to those with which I will not be able to share the moment of the defense of this work; although they are far in distance I continue feeling them near.

I would like to express my gratitude towards professors John MacGregor and Venkat Venkatasubramanian who welcomed me warmly at McMaster University and Purdue University respectively. Also I want to thank the master and the doctorate students of both universities by their kindness and friendship. These stays certainly improved the contents of this thesis, as well as contributed to my professional and personal growing.

Finally I must emphasize the economic support that has sustained and has allowed the accomplishment of the goals of this thesis. It came from the FPI program of the Ministry of Education and Science of the government of Spain.

Agradecimientos

Primero querría agradecer al profesor Lluis Puigjaner la confianza y apoyo dados durante todo el periodo de tesis. Siempre propició unas condiciones de trabajo excelentes y me ofreció grandes oportunidades de desarrollo profesional. Querría expresar mi agradecimiento hacia Sergio Ferrer, quién me animó a hacer esta tesis y que ha sido un compañero no solo de trabajo, sino de vivencias en estos años. También agradezco a Moisés Graells por su colaboración y guía; esta tesis debe mucho a sus comentarios y reflexiones.

Por el gran número de personas a mencionar, voy a agradecer de modo general la ayuda y amistad encontrada en los colegas de doctorado y estudiantes de ingeniería informática del grupo CEPIMA. El ambiente y compañerismo encontrados en estos cuatro años fue clave para el desarrollo de la tesis y el mío propio. Me gustaría hacer mención especial a aquellos con los que no podré compartir el momento de la defensa de este trabajo, están lejos en distancia pero los sigo sintiendo cercanos.

Me gustaría también expresar mi gratitud hacia los profesores John MacGregor y Venkat Venkatasubramanian por permitirme realizar sendas estancias en las universidades de Purdue y McMaster. También a los estudiantes de master y doctorado en ambas universidades, por su acogida y amistad. Estas estancias ciertamente mejoraron los contenidos de esta tesis, así como mi conocimiento profesional y personal.

Finalmente debo resaltar el soporte económico que ha sustentado y permitido la realización de la tesis, proveniente del programa FPI del Ministerio de Educación y Ciencia del gobierno de España.

Contents

CHAPTER 1

Motivation and Scope

In the last 30 years, market globalization has induced an intense effort and the application of very restrictive policies in the process industry to reduce operational costs, increase process efficiency and reduce the number of operators paying attention to the processes. In parallel, an increment on energy efficiency has been reached taking advantage of economies of scale by building very large sites (process size has been increased up to 10 times). The spacing between process equipment in plants has been also reduced to save money on energy, piping and instrumentation. In addition, it should be noted that production sales have been growing, leading to an important increment on the productivity and constraining some plants to work in excess of the capacities for which were designed.

In such large site environments it has become crucial to implement a high level of automation to manage complexity. For instance, nowadays there is no oil refinery that does not use advanced process engineering, such as advanced control systems, to improve process operations (Moro, 2003). As a result, traditional human supervision is not well suited anymore because of the nature of new plant information. Thus, novel and more complex automatic supervision systems are required to deal with this new industrial paradigm.

Under normal situations, the usual control system acts partly as operators to keep a satisfactory production process. However, abnormal situations from process malfunctions, out of the scope of the regulatory control, will continue arising in the plants. These long-time known situations now require novel solutions. In that way, a new supervisory control is needed to keep the automatically controlled plant under the established normal operating conditions. Policies that adopt new production standards from the economy point of view but neglect the associated safety requirements may harm robustness and safety of the process as well as the initial motivations of change, the plant economic competitiveness.

In order to face these frequent incidences, process industry generally has preferred to use simple detection tools that assure scarce but reliable support information instead of complex systems with extended capacities but with a lack of robustness. In that sense, simple but robust procedures as standard control alarms (e.g. high level alarms, high pressure

alarms, etc.) ordered by importance, have been widely extended in process industry. In other cases, only monitoring tools informing about some specific deviations are implemented, not giving any helpful information about root causes of plant disturbances.

The main problem of these methods is that they are, in some cases, too simple to deal with complex sites abnormal events. Sometimes after an abnormal event occurs, control systems are not able to work (e.g. they cannot face erroneous measurements). In other occasions, processes are highly coupled causing that one deviation can propagate, and very fast, through several equipments making hard and even impossible to process operators to follow the actual abnormal events. Moreover, the common high process interdependence increases the possibility that a control action, through which the control system is trying to correct a disturbance in a particular process unit, will affect other units of the plant. It is for all these reasons that more sophisticated detection and diagnosis tools as well as complete decision support systems are needed to manage daily incidences.

Remarkable examples of bad abnormal events management during last years causing awful consequences are easy to find. The explosion at the Kuwait Petrochemical's Mina Al-Ahmedhi refinery in 2000, which resulted in about $400 million in damages, the explosion at the offshore oil platform P-36 of Petrobras, Brazil, in 2001, with estimated loss about $5 billion (Venkatasubramanian, 2003), or the more distant disaster in Bhopal (1984) have shown that advanced control systems and simple supervision methods may not be sufficient to deal with plant abnormal situations.

Although these catastrophic accidents are not frequent and represent extreme cases within the spectrum of major abnormal situations, minor accidents are very common with significant effects on economy, safety and environment. According to Laser (2002), the inability of automated control systems and plant operators to deal with abnormal situations costs the British economy up to $27 billion per year. Another study indicates that elimination of all abnormal situations in the petrochemical industry could increase profits up to 5% (Harrold, 1998). Thus, all these new Process System Engineering (PSE) innovations addressed to increase process efficiency may not only put at risk the plant safety, but also the initial objective of minimizing cost (with the aim to compete in such aggressive markets) could be highly hampered without a carefully designed process supervision policy.

Abnormal events management (AEM) is the subject which arose to address all these new necessities. It is in charge of providing tools that help operators to face abnormal events and to drive back process to control. Therefore, new supervision systems must be able to manage and take advantage of the plants available information, supply reliable on-line plant state knowledge, as well as provide decision-making support to humans in charge of processes.

Hence, fault diagnosis has become the core of the on-line safety plant response (or AEM) in the new highly automated heavy industry, as it is who determines whether the plant is running correctly or not. In large industrial complexes such as chemical, petrochemical, cement, steel, or power, it is very difficult for process operators to detect the existence of a fault by examining time-sequenced data, among the hundreds of available variables, and it is even more difficult to find out the root cause of the detected anomalies. During last two decades an immense amount of research has been done in that specific field, resulting in a great variety of methods with increasing acceptance in practice. Nevertheless, none

of the multiple approaches reported in the literature offers an overall solution to the fault detection and diagnosis problem. They do not satisfy the accuracy and reliability that such kind of automatic systems require and even less can be said about the success reached in the decision-making support field.

Obviously, problem identification is the first and main step to put an effective stop to any abnormal situation. However, even under good diagnosis, the usual lack of detailed and structured diagnosis information makes plant operators not relying on the diagnosis results. Such distrust highly influences at the time to apply several corrective actions when fast responses are required. Moreover, plant managers and operators face two apparently opposite objectives, safety and economy, which puts additional pressure on operators decision-making in critical instants. Hence, people in charge of plant control are subject to a great stress having to make fast decisions under complex situations, with insufficient information and subject to a very high load of responsibility to evaluate the trade-off between risk and economy. It is at that point when an additional decision-making support tool becomes clearly indispensable.

In order to cope with these limitations, the supervisory system should overcome the industry reluctance by offering the same reliability achieved at other plant information levels (e.g. in the regulatory control). Such supervisory system should involve more than just a reliable Fault Diagnosis System (FDS), but a complete decision-making support methodology based on a first fault identification and a later intervention protocol proposal. It should work in real time detecting abnormal events as soon as possible (much before than control alarms), informing the operator about process deviations, explaining their root causes and suggesting solutions or corrective actions. Therefore, appropriate measures could be taken in real time to avoid fault propagation and tending to reduce possible dangerous consequences of such abnormal events from different points of view.

In order to create a robust operator support tool, a cautious information management is crucial. In that sense, a careful data acquisition and data consistency verification are the initial steps to assure a rigorous methodology. Then, a flexible and robust diagnosis system will be required to identify the normal or abnormal plant conditions from consistent plant information. Finally, an intelligent decision support module which gathers current and off-line process knowledge will be necessary to propose the corrective actions and avoid undesirable consequences.

This thesis makes an effort to globally tackle the overall AEM and its related decision-making support. In that sense, innovative techniques and solutions are proposed in the data acquisition system, the data rectification module, the fault detection and diagnosis systems, as well as in the final support to the process operators. The state of the art for this wide field and related sub-topics are given in chapter 2.

CHAPTER 2

State of the Art

This chapter presents the State of the Art of the main fields involved in the Abnormal Events Management. The approach focuses on recent published methodologies and outlines weaknesses still presents at each field.

2.1 Sensor placement and data acquisition

Plant decision-making relies on the information acquired from the process. Hence, a reliable and accurate data acquisition system is a prerequisite for the correct performance of any computer-based process supporting tool relying on process measurements. In process industry sensors are the core of the plant data acquisition system. Thus, it will be crucial that the designed sensor network (SN) supplies relevant plant information to the Process System Engineering (PSE) tools, upon which correct performance of actual industry competitiveness strongly depends.

The SN design and upgrade are well known problems for the plant design engineers (Carnero et al., 2005; Gala and Bagajewicz, 2006). Depending on the main use that collected data will have on the process computer-aided supporting system, different variable sets should be measured. In that way, different SN design criteria can be imposed whether the data is going to supply the regulatory control, the monitoring system, the on-line optimization module or another one. On the other hand, different sensors may be available for measuring a variable with widely varying capabilities such as the measuring range, the reliability or accuracy. The cost of the sensor will be a function of its capabilities and its performance. This information must typically be obtained from instrumentation manufacturers or suppliers and will be vital together with the measuring criteria at the time of designing the SN.

From the cost, complexity and feasibility point of views it is not a good practice to measure each of the hundreds or thousands of variables existing in an industrial process. Therefore, it is usual to constrain the SN design problem by only focusing on the measurement of some critical variables, which depends on the selected design criteria in each

case. On this basis, different approaches have been proposed in the related literature from three decades ago.

The first reported SN design procedure was focused on maximizing the estimation accuracy of a data reconciliation methodology for linear processes (Vaclavek, 1969). Since then, different criteria and approaches have been proposed so far. The observability criteria, that demands some key variables to be always known or observed by direct measurements or by model estimations has been widely studied (Vaclavek and Loucka, 1976; Ragot et al., 1992). Other criteria have been added together with the observability, developing multi-criteria designs. In this sense, accuracy and reliability have been included as complementary important criteria for the SN design (Kretsovalis and Mah, 1987; Ali and Narasimhan, 1995, 1996) as well as the cost of the SN (Madron and Veverka, 1992; Bagajewicz and Sánchez, 2000).

The SN design has been also formulated as a classical optimization problem using traditional and new algorithms for solving it (Bagajewicz, 1997). Multiple formulation and optimization approaches have been proposed to solve the SN design since it was formulated in these terms. Approaches based on stochastic methods using genetic algorithms (Benquilou et al., 2004; Musulin et al., 2005) or Mixed Integer Non Linear Programming (MINLP) methodologies have been proposed obtaining interesting results (Angelini et al., 2006a).

Much has been done in terms of guaranteeing variables observability, assuring high measurements reliability or focusing on controllable factors but few have looked at the data acquisition as the source of information of the diagnosis system. Fault monitoring, diagnosability and isolability criteria have been scarcely considered (Raghuraj et al., 1999; Bhushan and Rengaswamy, 2002a,b) though successful combinations using an optimization approach (Musulin et al., 2004; Angelini et al., 2006b) have been obtained. In any case, low research works have been devoted to address the sensor placement issue from the diagnosis point of view and additional SN design criteria could have been proposed to extend the Fault Diagnosis Systems (FDSs) capabilities. For more details about SN designs, the reader is addressed to Bagajewicz (2002).

2.2 Data rectification

As it has been described previously, process industry regularly measures many variables to supply on-line plant information to the PSE modules. These data are crucial for the plant economy and safety. However, they are inherently corrupted by errors during the measuring, processing and signal transmissions, even if the best available equipments are chosen.

Data rectification focuses on erasing those measurement errors dragged in the measuring process. The worth of data rectification depends on the installed SN, whose design has been previously discussed, and is significantly linked upon the location of measured variables. The errors that affect the measured data can be generally categorized into two main classes (Narasimhan and Jordache, 2000): *Random* and *Gross* errors.

- **Random errors** are usually assumed to be normally distributed with zero mean and are described as common cause variation. This background noise is found in all real processes and measuring systems at some degree.

- **Gross errors** (GE) are those errors that are known as special case of variation and are considered as outliers when they are detected.

The uncertain measurement of variables may affect the achievement of an adjusted regulatory control, an efficient on-line plant optimization and even derive to unsafe operations. Hence, there is a clear need of increasing the confidence on measurements by compensating the random and gross errors before accepting them as reliable information for correct decision-making. Therefore, the first step in data analysis consists of rectifying process data by approximating to the true values of process variables. Due to the positive effects achieved by the data rectification on different PSE tasks (regulatory control, optimization, monitoring and so on) it has been increasingly incorporated at the chemical and petrochemical industry (Narasimhan and Jordache, 2000).

Most of the data rectification techniques and their developments have been reported during the last three decades. All of them can be clearly classified in two different groups: 1) filtering-based rectification methods, and 2) model-based rectification methods. Both are described in the two next subsections.

2.2.1 Filtering-based rectification methods

Filtering rectification methods are based on analog or digital filters to erase high-frequency noise of process values. The lack of robust process models in real life has made of filtering based rectifying techniques an extended tool in current industry applications. It is at that point where the use of univariate filtering based rectification methods get a more significant value. Thus, statistical filters have been successfully applied in the literature (i.e. Exponential Weighted Moving Average (EWMA), Moving average (MA), etc.). A wide range of statistical filters are available in the literature (Narasimhan and Jordache, 2000).

Moreover, they are easily implementable and due to their evaluation simplicity have become very popular in on-line applications. They have been broadly used in data pretreatments as data conditioning tools, being extensively incorporated to many Distributed Control System (DCS) during last years. In contrast to model based data rectification methods, they can be also used to eliminate large outliers in a data pre-processing stage (previous to data reconciliation) by just setting the tolerable lower and upper bounds of process variables.

New univariate filtering techniques have also arisen during two last decades as a result of deep discussion about the multi-scale nature of chemical process data (Stephanopoulos and Han, 1996). These new techniques are focused on exploiting the wide and rich multi-scale details of chemical process signals to overcome some limitations imposed by simplifying such signals as mono-scale traditional filters do. Among all these new techniques, wavelets functions (Haar, 1910; Daubechies, 1992; Mallat, 1999) have been gaining great popularity (Nounou and Bakshi, 1999; Jiang et al., 2003).

However and despite of the mentioned advantages regarding data based rectification methods, filtering techniques have a main weakness: they do not guarantee consistent estimations with respect to the fundamental balances that rule the process. Furthermore, as they are univariate techniques (estimate each variable separately) the redundancy achieved by using rigorous models can not be used, then losing estimation capabilities of model based methods (i.e. estimation of not measured process variables).

2.2.2 Model-based rectification methods:

The model-based rectification methods are more commonly identified as Data Reconciliation (DR) methods. They use a process rigorous model to make the estimations consistent with the model. The estimation achieved results of exploiting the redundancy of process measurements and a model of physical constraints as material or energy balances upon which the model is based.

DR compensate and reduce GE since these alter the methodology basis causing bad estimations. Thus, for an effective reconciliation, GE should have been previously erased by a companion and complementary GE detection technique. Both techniques are then normally applied together for improving accuracy of measured data.

DR can be formulated as an optimization problem. Thus, process data accuracy is improved by means of adjusting the measured values so that they satisfy a set of equality (mass and energy balance, reaction equations, etc.) and inequality (variables physical bounds) constraints. It is attained by minimizing the difference between the measured (y) and the estimated ($\widehat{y}$) variable values subject to the model constraints:

$$\begin{aligned} & min \sum_{i=1}^{n} w_i \, (\widehat{y_i} - y_i)^2 \\ \text{subject to} \quad & f(\widehat{y_i}, u_k) = 0 \end{aligned} \tag{2.1}$$

where, f represents the process model associated constraints, w_i is the weight for each variable (i) and u_k is the estimate of each unmeasured variable (k).

DR was first applied to linear processes using steady state models (Kuehn and Davidson, 1961) but later many other contributions have extended it (Mah et al., 1976; Crowe et al., 1983; Sánchez and Romagnoli, 1996). Throughout last decades, the original problem has been also extended to nonlinear (Tjao and Biegler, 1991; Kelly, 2004) and dynamic processes. In that sense, several approaches and proposals have arisen during last years. Noticeable solutions are next remarked:

- Dynamic programming based solutions for the optimization problem (Liebman et al., 1992; Mingfang et al., 2000).
- Analytical and numerical solutions derived from polynomial representations of the process model (Bagajewicz and Jiang, 2000).
- Kalman Filter based analytic discrete solutions (Narasimhan and Jordache, 2000; Abu-el Zeet et al., 2002).

Literature assessment As it has been reviewed, different data rectification approaches have been proposed revealing capabilities to face linear or non-linear problems, from these cases in which first principle models are available to those in which the only reliable information is process data. From the scope of that thesis, main drawbacks uncovered stem from the scarce reported on-line applications, crucial for the Abnormal Events Mamagement (AEM) systems. Few works have focused on usual on-line implementation difficulties as the wavelet "end-point" effect (Addison, 2002; Misiti et al., 2004), the lack of correlation between variables with different measuring frequency or the use of more efficient solvers for the on-line application of these techniques. These facts make this area an open field yet requiring of further work. More details and some extra-challenges of DR are reviewed by Crowe (1996) and Narasimhan and Jordache (2000).

2.3 Fault diagnosis

As it was mentioned in chapter 1, fault diagnosis is the nucleus of the AEM. The fault diagnosis in the chemical industry has been extensively studied in the literature and alternative approaches have been reported to address the complex character of the problem. A general overview of the process, the regulatory control system and the diagnostic module are physically distinguished in Figure 2.1.

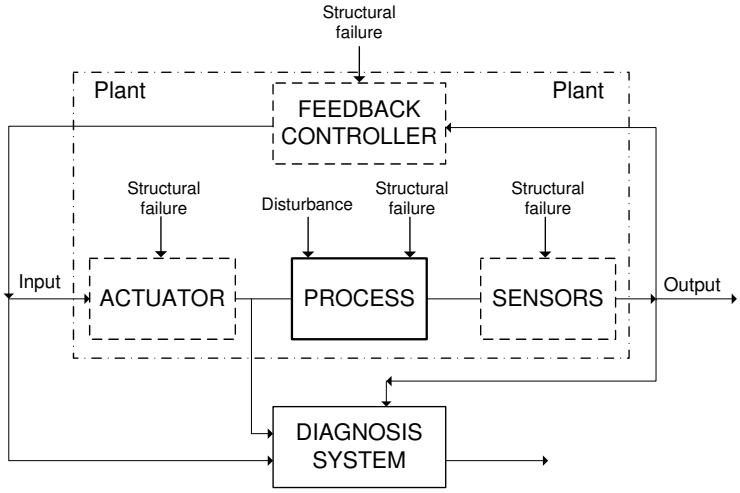


Figure 2.1: General overview of process control under presence of failures and disturbances.

Furthermore, Figure 2.1 represents the wide variety of possible faults that can arise in a plant. They are classified in two separated groups:

Structural failures (faults): They refer to changes in the process itself. Any equipment failure, from tank leakages, compressor or pump breakages, to sensor or controller malfunctions can be considered as structural failures.

Disturbances: They refer to exogenous changes in process parameters (coming from the process environment). Circumstances such as the change of a heat transfer

coefficient due to the equipment performance decay or the lack of a good equipment dynamic modeling are considered as process disturbances.

Two main characteristics can be used to classify a FDS:

1. The type of knowledge it uses.
2. The type of diagnosis strategy applied.

In general, the strategy used to find out a fault root cause is highly dependent on the knowledge representation it uses. Therefore, it results more appropriated to establish such FDS classification based on the "a priori" knowledge used. The fault diagnosis review presented next, follows a similar structure as that one proposed by Venkatasubramanian et al. (20003a,b,c). In that classification, these authors establish three main sets. Qualitative Model Based Methods, Quantitative Model Based Methods and Historical Based Methods (Figure 2.2).

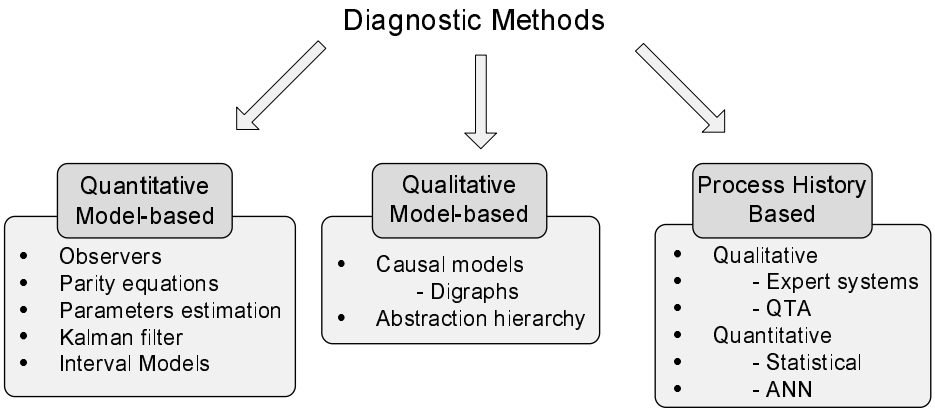


Figure 2.2: Classification of approaches for process monitoring and diagnosis, Venkatasubramanian et al. (2003b).

Next, it is presented a review of the diagnosis methodologies reported recently.

2.3.1 Quantitative model-based approaches

To detect and isolate faulty components, some sort of redundancy is required. The redundancy is used to make consistent checks between related variables. In applications with very high security demands such as aircraft control-systems, redundancy can be supplied by extra hardware; this is known as hardware redundancy. A critical component, for example a sensor, is then duplicated or triplicated and voting schemes can be used to monitor signal levels and trends to detect and locate faulty sensors. Hardware redundancy has the advantage of providing high reliability, but the approach presents several important drawbacks, e.g. extra hardware costs, space and weight consideration, and the fact that some components can not be always duplicated.

Instead of using hardware redundancy, analytical redundancy can be used where redundancy is supplied by a process model instead of extra hardware. Such redundancy supplied by a model can be defined more formally as:

Definition (Analytical Redundancy):

There exists analytical redundancy if there exists two or more different ways to determine a variable x by only using the observations z(t), i.e. $x = f_1(z(t))$ *and* $x = f_2(z(t))$*, and* $f_1 \neq f_2$.

Thus, the existence of analytical redundancy makes it possible to check the validity of the assumptions made to ensure that $f_1(z(t)) = f_2(z(t))$.

As a clarifying example, let's assume that two sensors measure the variable x according to $y_1 = x \wedge y_2 = 2 \cdot x$. The integrity of the two sensors can then be validated by ensuring that the relation, $y_2 - 2y_1 = 0$ holds. Following that example, it is easy to see that a malfunction in any of the two sensors would invalidate the relation and a fault could be detected.

In more general cases and in order to facilitate not only fault detection but also fault isolation, there is a need to describe fault influence on the process more formally, i.e. fault models of some sort are needed. Such diagnosis systems based on analytical redundancy are also called quantitative model-based approaches (Frank et al., 2000; Isermann, 2005). The first step in these approaches consists of generating an explicit mathematical model of the process. Then, the process can be validated by comparing a measured variable with an estimate of the variable, obtained using the model. These model formulations are typically differential/difference equations, transfer functions, and/or static relations.

A wide variety of models has been used (first principles, frequency response, etc.). In recent years, first principles models based on physical understanding of the process are becoming very popular due to the availability of high performance computers. In any quantitative model-based fault diagnosis system, two main phases can be distinguished: the residual generation and the subsequent residual evaluation.

- The **residual generation** consists of creating a set of residuals which detect and unequivocally identify different faults. The way in which the residuals are generated by a model is the basis from which most quantitative model-based fault diagnosis systems are differentiated.

- The **residual evaluation** is a decision-making process that transforms the generated residuals into quantitative values from which the current plant state can be identified by using a logic decision function. In practice, there are usually a great number of unknown inputs and uncertainties that make the complete decoupling of all unknown inputs hardly achievable. Hence, residuals (or any decision functions built from them) always deviate from zero even if no fault is present. Thus, robust residual evaluation has become the main way to keep the false alarm rate small with an acceptable sensitivity to faults. In the related literature robust residual evaluation has been accomplished by different manners, for example by statistical data processing, data reconciliation, correlation, pattern recognition, fuzzy logic or adaptive thresholds.

As it was remarked above, the main difference among the quantitative model based approaches (observers, parity space equations, Kalman filter, parameter estimation, etc.) basically lie on the way in which the models are developed to facilitate the residuals evaluation.

2.3.1.1 Observers

Observers are diagnostic-oriented models (see examples by García and Frank (1997) or by Sotomayor and Odloak (2005)) which are sensitive just to a specific fault while are insensitive to the rest. The approach consists of generating the set of observers which accounts for the detection and isolation of those faults considered in the problem. Thus, under normal operating conditions, the observers track the process closely and the residuals from the unknown inputs will be small. When a fault occurs, all the insensitive observers will remain in showing small residuals whereas the generated residuals of the sensitive observer will deviate in a sensible and clear way. The set of observers is so designed that residuals show different patterns for each fault, which makes the fault isolation possible.

An important fact outlined by Frank and Ding (1997) is that the observer-based design does not need application of state estimation theory, instead, only output estimators are required which are generally realized as filters. This has often been overlooked in the literature and has misled many practitioners to the erroneous opinion that for the observer-based approach the knowledge of state-space theory would be indispensable. A number of methods for observer-based residual generation have been proposed over the past two decades. The most significant approaches are the fault detection filter, the innovation test, the dedicated and generalized observer scheme, and the unknown input observer scheme (Patton et al., 1989). Readers may refer to the review by Frank and Ding (1997) for further insight on the subject.

2.3.1.2 Parity equations

The basis of parity equations is the re-arrangement of the model equations in order to obtain a model structure maximizing fault isolation performance. The parity (consistency) equation method is the direct implementation of the analytical redundancy extracted from comparison of a mathematical model and the real behavior of the plant. Primary residuals are formed as the difference between the actual plant outputs and those predicted by the model. These are then subject to a linear transformation, to obtain the desired fault detection and isolation properties.

The parity relation techniques have been approached from two different areas. In one hand much research has been done from the data reconciliation and gross error-detection area by relying on material and energy balance calculations (Himmelblau, 1978; Romagnoli and Stephanopoulos, 1981; Puig and Quevedo, 2002). On the other hand, the aerospace research has also made use of these model-based diagnosis techniques by using first static and later dynamic models implementing state-space and transfer function forms. For details about the parity equations approach, interested readers are addressed to Isermann (1993), Patton and Chen (1991) and Staroswiecki et al. (1993).

2.3.1.3 Parameter estimation

When designed with the same objectives, parity equations and observers based techniques are equivalents (Gertler, 1991). However, parameter estimation is somewhat a different quantitative model based approach, despite the relationship between parameter estimation and the parity space has recently been worked out analytically (Delmaire et al., 1994).

In order to apply that modeling strategy, accurate process parametric models are required for describing the process behavior (Hestetun and Hovd, 2006). Through this parametric models, the parameters are evaluated when process measurements are available. Possible relevant changes in such estimated parameters with respect to the normal operating standards are then calculated. Such parameters residuals estimation will be used by properly trained pattern recognition techniques able to relate such residuals to process faults.

Very often, process faults cause changes on physical parameters. Continuous-time model descriptions are used, so their model parameters are directly coupled to the physical parameters of the process. Then, depending on the complexity of the system considered and the available measurements, the physical parameters can frequently be estimated in a straight way. At least the relationship between model parameters and physical parameters is simpler than in discrete-time systems. Wide parameter estimation reviews are given by Gertler (1995) and Höfling and Isermann (1996).

2.3.1.4 Kalman filter

Kalman filter is a recursive algorithm able to evaluate state estimation, founding wide applications in chemical industry, from data reconciliation (Tjao and Biegler, 1991) to fault detection and diagnosis (Fathi et al., 1993; Liu, 1999). There exist different versions of the Kalman filter (KF), being the basic one derived for linear systems and an extended version (EKF) for nonlinear systems the most common applied versions.

Kalman filter (Willsky, 1976) can be used in order to optimally estimate the state of the process by monitoring changes in the prediction error. It combines state prediction from previous state information and a later correction of such estimations using current process measurements. A bank of Kalman filters designed for the existing faults can be used to isolate them (Basseville and Benveniste, 1986). When using these approaches, it is essential to have the knowledge about the relationships between faults and the model parameters.

2.3.1.5 Interval models

Research on robust fault detection methods dealing with modeling uncertainty has been very intense on the Fault Detection and Isolation field. Two different approaches exist:

1. **Active** methods based on generating residuals insensitive to uncertainty (modeling errors and disturbances) and sensitive to faults using some decoupling methods.

2. **Passive** methods, which by means of adaptive threshold improves the robustness of the fault detection at the decision-making stage.

In passive methods, the effect of model uncertainty is taken into account when determining the optimal threshold to be used in the residual evaluation, either in the time domain

(Horak, 1988) or in the frequency domain (Hamelin et al., 2001). In particular, interval models which models uncertainty considering a normal operation model plus the uncertainty in every parameter, results in an effective fault detection system.

An interval model consists of a process model that includes vectors of uncertain parameters bounded by intervals (Puig et al., 2002). Such parameters include all the possible values of the nominal operation. The key to reach confident robustness without losing detection effectiveness is the properly selection of the adaptive thresholds intervals. Fault detection is based on comparing the system responses with the set of model responses, obtainable by varying the uncertain parameters within their intervals. The detection is performed by testing whether the measured input and output from the system lie within the behavior described by the model of the faultless system. An example of interval model for fault detection is formulated in Ljung (1999). Diagnosis is carried out by implementing models for each fault considered in the studied system.

2.3.2 Qualitative model-based approaches

In many chemical processes, operating conditions change frequently and only the qualitative relationship among variables remains constant. It is under such circumstances when the qualitative causal models have been extensively applied to deal with the fault diagnosis problem. The reasoning now is to identify functional changes which resulted in the malfunctioning of the process.

The concepts under qualitative model-based methods are similar to those of quantitative model based methods. Both use a priori knowledge to infer the searching strategy. While in quantitative model-based the fundamental understanding of the physics of the process is expressed in terms of mathematical models among system inputs and outputs, in qualitative model-based, these relationships are expressed in terms of qualitative functions centered around different units in the process. In addition, these qualitative methods can be supported by the inclusion of quantitative process information. Thus, some restrictions may generally be added to overcome qualitative inherent drawbacks as the inaccuracy when generating too large fault candidate sets (Maurya et al., 2004).

These techniques are used to predict the behavior of the process under normal operation conditions and some faulty conditions. The fault diagnosis is then achieved by comparing the predicted behavior with the actual behavior. Depending on the way the qualitative knowledge is organized, different methodologies can be distinguished. Next, two of the main qualitative diagnosis methodologies are described:

2.3.2.1 Digraph-based causal models

The process causal information can be represented by a digraph (DG), in which the process variables (and parameters) are represented as nodes and causal relations are represented by directed arcs. In that framework, arcs go from *cause* nodes to *consequence* nodes.

A signed DG (SDG) is a relevant variant of a simple DG that includes in the directed arcs representation of the sign of the influence (Vedam and Venkatasubramanian, 1999). DG or SDG models used for fault diagnosis also include root nodes that are used to

model exogenous causes such as process faults. Vaidhyanathan and Venkatasubramanian (1996) extended the SDG to automatically perform a HAZOP analysis creating the so called HAZOP-digraph model (HDG). Although SDG guarantees that the root cause is a subset of the proposed set of root causes, it generates several spurious solutions. Several approaches have been proposed for filtering these solutions, such as including process knowledge (Maurya et al., 2004). Hence, this matter continues being a field of research.

2.3.2.2 Abstraction hierarchy

Abstraction hierarchy modeling (de Kleer and Brown, 1984) consists of decomposing the behavior of the whole system in the behavior of its subsystems. Such decomposition makes easier to diagnose the behavior of the overall system solely from the laws governing the behavior of its subsystems. This structured framework is very suitable to deal with very complex processes. The decomposition can be structural or functional (Rasmussen, 1986). The first one is related to the connectivity information of a unit whereas the second one specifies the output of a unit in function of its inputs. Thus, abstraction hierarchy can be seen as a helping qualitative methodology more than a diagnosis technique itself.

2.3.3 Process history based methods

Fault diagnosis in complex systems generally lacks of sufficiently accurate mathematical models. Besides, in that complex situations, plant knowledge to generate alternative qualitative diagnosis procedures is often unavailable. Such limitations make the use of data-based techniques, either in the framework of diagnosis expert systems or in combination with a human expert, the only feasible way to face the diagnosis problem.

Process history-based methods take advantage of the availability of the large amount of historical process data stored in the data bases of current chemical plants to extract a priori knowledge for fault diagnosis. They are classified as qualitative or quantitative according to the way in which data are transformed and presented as a priori knowledge (namely feature extraction). Thus, two main groups can be distinguished: 1) methods that extract qualitative knowledge from the stored historical data and 2) methods that extract quantitative knowledge.

2.3.3.1 Qualitative feature extraction

Two of the major qualitative feature extraction methods are next presented:

Expert systems Standard expert systems are artificial intelligence applications which perform a task, such as the decision-making required in fault diagnosis, that would otherwise be performed by a human expert. They use expertise from operators, engineers, process history data, etc., stored as knowledge in form of if-then rules. Then, they diagnose new process data by running the if-then rules set through an inference engine, which is software which provides diagnostics to the user. Such systems are not based on deep knowledge of the physics of the system but on the relations of some crucial variables and eventually their behavior under abnormal states.

Different approaches have been proposed from the initial papers by Henley (1984) to more recent works as that one diagnosing a multistage flash desalination process proposed by

Tarifa and Scenna (1998).

The most interesting features of expert systems are:

- Easiness on its development.
- Transparent and comprehensible reasoning for plant operators.
- Capability of reasoning under uncertainty, as they can implement fuzzy logic.
- Ability to justify its own line of reasoning in a manner directly intelligible to the enquirer.

The main weaknesses of expert systems can be summarized on the next points:

- They fail to handle any situation that is not included in the knowledge base.
- They are very system specific.
- Its representation power is quite limited.
- They are difficult to update.

Qualitative Trend Analysis (QTA) Process trend analysis is a useful approach to exploit the temporal information of variables to diagnose faulty process states. By using QTA, some significant events seen on process variables can be straightly linked to different abnormal states of the plant. The representation allows inferring a proper reasoning about the process behavior for fault diagnosis. In that sense, Cheung and Stephanopoulos (1990) introduced a representation of trends based on triangulation. There, each segment (between two critical points) of a trend was represented by its initial slope, its final slope and a segment connecting the two critical points.

The problems of qualitative representation are the noise and the multi-scale nature of data that can corrupt the representation. Bakshi and Stephanopoulos (1992) faced that problem using multi-scale filtering. Dash et al. (2003) proposed a new trend classification that includes fuzzy logic to avoid misclassification from large knowledge-based faults signatures. In such methodology a three-step based procedure is implemented.

1. Firstly a language to represent the existing trends is established (based on some "primitives" trends).
2. Then, a trends identification technique is proposed (based on an "interval-halving" approach).
3. Finally, a mapping from trends to operational condition is implemented, based on a pattern recognition approach.

2.3.3.2 Quantitative feature extraction

Such methodologies face the diagnosis problem as a pattern-recognition problem, that is, a classification of current plant data in some pre-determined classes. Two main approaches can be outlined.

Statisticals Statistical quantitative methods have become really useful in process industry since the necessity/opportunity of dealing with all the information provided by current data acquisition systems. In that sense, the main requirement of most of these techniques is the existence of good historical data.

From the statistical process monitoring viewpoint, plant measurements are considered time series under random influences. Therefore, it is essential to formulate the systems in probabilities settings. When the process is under control, the observations have probability distributions corresponding to the normal mode of operation. The underlying distributions change when the process is out of control.

Currently, Multivariate Statistical Process Monitoring (MSPM) is the most extended trend. It has been favored in industrial practice because of its ability to handle large number of highly correlated variables, measurements errors, non-linearities and missing data. Among MSPM methods, principal component analysis (PCA) has become the basis of the majority of techniques implemented in the literature. It allows reducing the dimension of plant models by using lineal dependencies among the variables. By adopting the first few principal components, a new coordinated system is obtained by rotating the original variables. Then, projecting process data into that reduced space allows describing adequately and in a simpler and more meaningful way the process state.

The first descriptions of PCA were presented by Pearson (1901) and Hotelling (1933). Recently, PCA methods have been introduced as an extension of monovariate Statistical Process Control (SPC) charts to monitor modern multivariate processes aimed at increasing quality and yield, and to reducing costs (Jackson, 1980; Wise et al., 1989; Kresta et al., 1991). Partial Least Squares (PLS) is also a good tool for predicting and detecting changes in process quality variables. PLS rearranges a new coordinates space not only taking into account process variables (as done by PCA) but also product quality variables. Hence, a link between process and quality variables is created in a reduced and meaningful space. Readers interested on further details of PCA or PLS could review the work given by Kourti and MacGregor (1995).

Several extensions of PCA have been developed for addressing the limitations of the conventional PCA (linearity, time invariability, gaussian function based, etc.). Multiway PCA allows analysis of multidimensional matrices (Nomikos and MacGregor, 1994) and it is useful on batch processes monitoring. Hierarchical or multi-block PCA permits easier modeling and interpretation of a larger matrix by decomposing it into smaller matrices or blocks (Wold et al., 1996). Dynamic PCA extracts time-dependent relationships in measurements by augmenting data matrix by time lagged variables (Ku et al., 1995). Nonlinear PCA extends to extracting nonlinear relationships among process variables (Dong and McAvoy, 1996). Multiscale PCA combines the ability of PCA to extract information from data with that of wavelet analysis to extract deterministic features by decorrelating autocorrelated measurements (Bakshi, 1998).

Neural networks Artificial Neural Networks (ANN) have been extensively studied in fault diagnosis literature. They have mainly been used as data based models in function approximation problems and as classifiers in classification problems. An ANN consists of neurons, simple processing elements, which are activated as soon as their inputs exceed certain thresholds. The neurons are arranged in layers, which are inter-connected in such

a way that input signals are propagated through the complete network to the output. The choice of the transfer function of each neuron (e.g. a sigmoidal function) contributes to the overall nonlinear behavior of the network.

In general, neural networks that have been used for fault diagnosis can be classified from two points of view: 1) the network architecture, which gives rise to sigmoidal networks, radial basis networks and so on; 2) the learning strategy, resulting in two methodologies main lines, supervised and unsupervised learning.

In the **supervised strategy**, during the training phase the connection weights between neurons are learned from mismatching of the desired and actual values to guide the search. This procedure allows obtaining a classifier able to distinguish the different trained sets. Applications of neural networks to fault diagnosis (Hoskins et al., 1991; Sorsa and Koivo, 1991; Hernández et al., 2006) demonstrated the great potential and practical relevance of this approach, specially that of coping with oversize when dealing with dynamic problems. On the other hand, they are black-box methods and often the selection of the structure is not systematic and generates big losses of time. The most used supervised ANN in diagnosis is the back-propagation feed-forward neural network (Venkatasubramanian, 1985; Venkatasubramanian et al., 1990).

In the **unsupervised strategy**, ANN are used as clustering tools that allow identifying different classes from a set of confusing data on the basis of some clear feature. These networks are known as self-organizing since the structure is adaptively determined based on inputs to the network. In this kind of networks the number of classes expected are usually given before training. The basis of such ANN are described in Kohonen (1984).

2.3.4 General conclusions

Reviewing the literature of the three last decades, the overall conclusions about fault diagnosis in chemical plants can be summarized in some few points.

2.3.4.1 Desirable characteristics of a fault diagnosis system

The essential features of a FDS are basically those proposed by Venkatasubramanian et al. (2003b).

- **Quick response**: Capacity to quickly detect and diagnose the abnormal events.
- **Isolability**: Capacity to provide accurate information about the detected abnormal event.
- **Robustness**: Avoid false alarms even in case of disturbances and noise.
- **Novelty identifiability**: Faults that have not been modeled a priori have to be identified as novel malfunctions.
- **Classification error estimate**: The alarms generated by the diagnosis system have to be related to some degree of confidence.
- **Adaptability**: The system has to be flexible enough in order to easily adapt to process changes such as grade transitions.

- **Explanation facility**: Capacity to give enough information not only about the abnormal event but also about the reasons leading to the diagnosis.
- **Modeling requirements**: The system has to be implemented as fast and easily as possible. The knowledge acquisition and modeling efforts should be minimized.
- **Storage and computational requirements**: In general, on-line fault diagnosis algorithms are not too complex but require of a high data storage capacity. It would be desired to achieve a reasonable balance on these both competing requirements.
- **Multiple fault identifiability**: The combined effects of process faults have to be taken into account, mainly for those process faults that are more likely to happen or those that can cause the most hazardous situations.

2.3.4.2 Lack of comparative standards

Several qualitative analysis in the literature have confronted different fault diagnosis techniques (Venkatasubramanian et al., 2003a) pointing out the overall advantages and weaknesses of each diagnosis method. Nevertheless, an evident lack of comparative indices prevents of removing ineffective techniques. Although many tools have been proposed in the fault diagnosis literature, this lack of a standard comparative analysis has not permitted to discard methodologies or confront different approaches in the same conditions in order to extract overall conclusions.

When a FDS must be chosen for a specific process the question is clear, which FDS is the most appropriate for such case? It is evident from a literature review, that the choice of the FDS depends on the available a priori knowledge of the process. But, because of the great variety of existing approaches therein each FDS sub-classes, it is difficult to opt clearly for the best option and specific analysis must be made by the concerned process engineer. This is mainly due to diagnosis literature has generally faced partial and different case studies, has not offered complete information to reproduce the experiments and has not confronted new methodologies with previous related techniques.

Some fault detection works have presented rigorous comparative measures (Chen and Liao, 2002) that clearly show their quantitative detection rates fault by fault at the same time that compare their results with other methods. However, there exist many works that only refer their results in a qualitative way, demonstrating how their models explain more process variance than a former approach or just showing graphical illustrative improvements, as Wang and Romagnoli (2005) or Dong and McAvoy (1996). Such kind of works are hardly comparable and the goodness of the methodologies remains confuse regarding all the existing approaches.

Even more confusing and hardly comparable are the results presented in the diagnosis related works. The approaches presented frequently address different problems usually applied to particular cases. In such works, FDS performance measurements are quite disperse and even contradictory. An extensive review of present literature shows how often the same author proposes different diagnosis approaches giving different results pattern. For instance, Dash et al. (2003) proposed a fuzzy logic based trend classification and evaluated the FDS performance applied on a CSTR (Luyben, 1990) by means of similarity indices and severity levels, while later on 1999 he presented a PCA-SDG fault diagnosis system (Vedam and Venkatasubramanian, 1999) in which the performance was evaluated

by giving an overall correct diagnosis percentage, showing some qualitative improvements, in a different case study (a fluidized catalytic cracking unit).

From this previous analysis it is clear to conclude the necessity of comparison standards. Although reported techniques conform an invaluable background for chemical processes diagnosis, offering a wide range of alternatives, the need to establish standard performance indices and procedures become essential to standardize the comparison between the existing diagnosis methodologies.

2.3.4.3 Need of further research on novel diagnosis tools

Classification tools are one of the most extended fault diagnosis systems in literature. They are based on mapping input data from process to different plant states. The classification methodologies have succeeded because of their implementation simplicity, so they do not require of process operators experience or process first principle knowledge.

Among these classification methodologies, data-based rules expert systems and machine learning are the most commonly applied methodologies. However, machine learning has not been properly exploited in the fault diagnosis literature and only neural network, a technique overcome in other technical fields -as in Word Sense Disambiguation (WSD) (Escudero et al., 2004), or in Text Categorization (Joachims, 1998)- has been widely applied as a pattern learning technique.

The huge machine learning area, developed in parallel to chemical fault diagnosis during recent years does not only consider classical neural networks, but more recent methodologies that have succeeded in other technical fields. From the analysis of that engineering field, it is straight to conclude that the fault diagnosis problem has not been widely considered from the machine learning point of view. Then, it arises the possibility of applying such algorithms and techniques to fault diagnosis of chemical processes in order to take advantage of such knowledge. Among the most widely used machine learning algorithms some could be remarked:

- Naive Bayes (Duda and Hart, 1973), the simplest representative of probabilistic learning methods.
- Decision list, applied by Yarowsky (1995).
- AdaBoost, Abney et al. (1999) gave a generalized version of that technique.
- Support Vector Machines (SVM) introduced by Boser et al. (1992).

2.3.4.4 Integration at different levels

Analysis of current diagnosis literature also shows that any isolated approach may, working independently, fulfill all the requirements (subsection 2.3.4.1) of an ideal FDS. Therefore, most reviews (Venkatasubramanian et al., 2003a; Frank, 1997) have emphasized the possibility of combination of different and complementary tools in order to overcome the limitations of isolated tools strategies. Such combination of tools would allow using different kinds of knowledge in one single framework for better decision-making. In that sense, not only combination of techniques but complete FDSs could be integrated to take advantage of complementary characteristics.

Furthermore, the integration within the supervisory control module could also be extended to an upper level at the plant control system to achieve additional synergies. Thus, the resulting FDS could be integrated to the scheduling or on-line optimization layers in the plant control hierarchy. Next, some details of previously described integration opportunities are given.

Integration of techniques The integration of isolated fault diagnosis methodologies has been widely and successfully used to improve diagnosis performance. Many kinds of combinations of the previously described diagnosis techniques (section 2.3) have been merged in order to gather the FDS optimal abilities mentioned in subsection 2.3.4.

In that sense, it should be pointed out the integration of neural networks and fuzzy-logic based expert systems that have originated a new fault diagnosis sub-branch known as neuro-fuzzy systems. These methods take advantage of ANN classification and non-linearities capabilities and fuzzy-logic comprehensibility for operators (Ayoubi and Isermann, 1997; Castellano et al., 2005) as well as its good uncertainty management.

Combinations of good residuals generators and modules that properly tackle with such data have also successfully been applied in the literature. Residuals generated from observers or PCA models have been successfully linked with fuzzy inference systems (Frank and Ding, 1997) or SDG (Vedam and Venkatasubramanian, 1999) to interpret such residuals.

Integration of different FDSs Although tools integration have been widely implemented, the combination of independent FDSs have been hardly considered in the fault diagnosis literature. The complementarity of different fault diagnosis systems in specific case studies suggests to combine the FDSs to obtain improved and more reliable diagnostics.

The work presented by Fröhllich and Nejdll (1996) deals with the conflict resolution problem between different isolated diagnosis responses, but few contributions have been found in diagnosis reported works in order to take advantage of such integration opportunities.

Integration of FDS with other plant supporting modules Fault diagnosis is one of the basic control tools within current process automation framework (Figure 2.3). Throughout last years, much attention have been paid to the concept of total process control and to the necessity of providing the isolated information from each plant level to the rest. In such way, information used on a data reconciliation module could be of crucial interest for the inference diagnosis procedure. Also, the information from fault diagnosis could be incorporated into the traditional solution paradigms of other process operations.

The integration of the FDS in the overall information plant framework (Figure 2.3) has been outlined in different fault diagnosis reviews, but there are very few applications and approaches that integrate different information levels to obtain beneficial synergies. Such integration has put relative attention on three main areas: optimal sensor location, data reconciliation and supervisory control (Venkatasubramanian et al., 2003a).

Bhushan and Rengaswamy (2002a,b) have investigated the problem of sensor location based on various fault diagnosis and observability criteria. Raghuraj et al. (1999) have

proposed a way to allocate a sensor network based on a SDG. Tanaka (1989) also worked in the sensor location enhancing process observability and detectability. Data reconciliation has been considered by Vachhani et al. (2001) and supervisory control has been taken into account in some other few works (Rengaswamy, 1995). Diagnosis knowledge has been also used as a way to reduce the search space and provide good initial values to enhance data reconciliation (Vachhani et al., 2001).

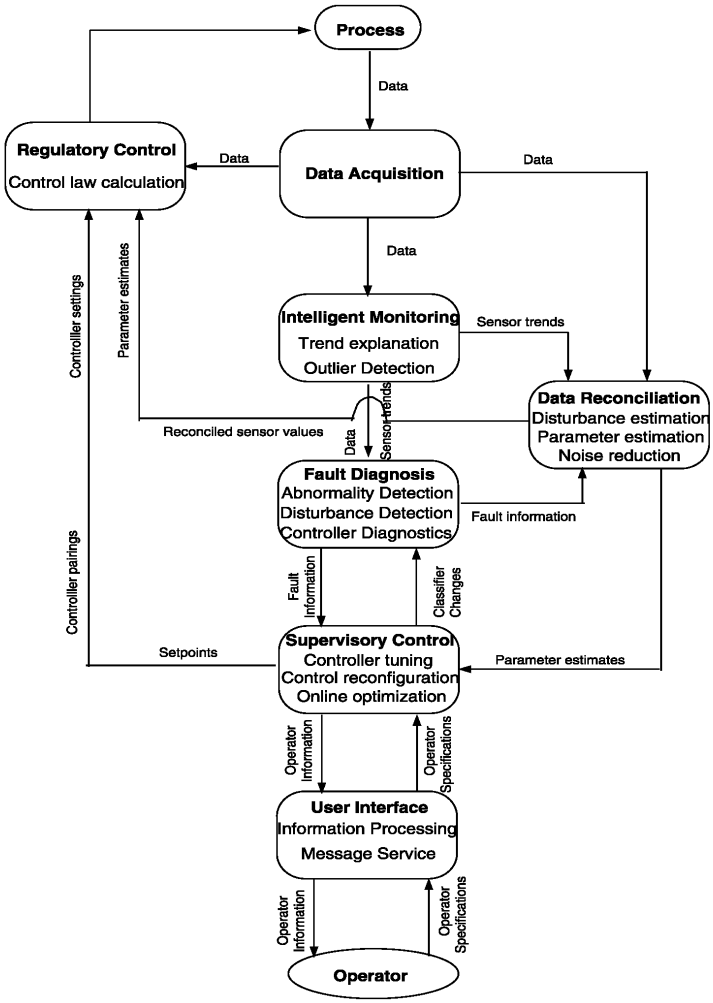


Figure 2.3: Integration of diagnosis tasks. Venkatasubramanian et al. (2003a).

Nevertheless, much more effort is required to properly exploit all the integration opportunities. Very few works have taken advantage of these integration opportunities and few applications have been reported despite of the theoretical benefits remarked on most of fault diagnosis review conclusions.

2.4 Decision-making support: Corrective actions

Final decision on AEM is still an expert human issue because of the lack of robustness of current automatic systems when implementing corrective actions directly to the plant. Nowadays, operators are in charge of making final decisions under abnormal situations in order to avoid further plant or personal damages. Such decisions are difficult to make due to the high pressure put on process operators. This high pressure results from the short time available to make decisions as well as the significant impact these actions normally have on the process. So, in order to avoid major mistakes on the decision-making a deep and wide support in form of information from which to take correct actions would be of great help. All these modules described in previous subsections (sensor placement, data reconciliation modules or diagnosis systems) are part of the overall supervision system whose final goal is to supply that additional supporting information. However, a final system that is able to manage the diagnosis and current process information, by reasoning and proposing straight corrective actions, would be crucial to the success of the decision-making.

Monitoring and diagnosis systems are the core of the current chemical plant supervision systems. They provide on-line information helping to interpret the actual behavior of the process, however, such information is partial and not generally sufficient to make fast decisions. They are not conceived so as to replace completely the humans in charge of the plant, but rather to detect early abnormal situations and to explain them so that operators can take right and efficient decisions. At the end, the operators have still to decide whether the proposed diagnosis or action is valid or not; as a consequence the previous supervision system must be able to justify at any time its reasonings with explanations. Besides that, additional tools that gather complementary plant knowledge are required to extend faults causes explanation, propose actuation alternatives and help the operator in charge to choose the best protocol to follow.

Many efforts have been made in solving the fault detection and diagnosis problems. A sample of these efforts is summarized in the literature review presented in section 2.3. Nevertheless, research on tools that interpret plant information (included fault diagnosis results) and support plant operators decision-making, has been somehow neglected and much less work and contributions can be found in the literature. Several attempts have been reported during last decade. Some of them have approached that issue from a qualitative point of view, considering that a qualitative diagnosis reasoning may facilitate the decision-making. Others have considered the corrective actions application an off-line issue assuming that these corrective actions are part of the maintenance policy, or have based the decision-making on a risk environmental analysis.

Leyval et al. (1994) used qualitative causal graphs to detect and diagnose process deviation from normal conditions guaranteeing a causal explanation of their diagnosis to facilitate human decision-making. Besides fuzzy logic theory was applied as an adequate method to manage imprecision of modeling and process data to obtain an effective interface between low level information (data supplied by sensors) and high level concepts (normal or faulty situations). Cerrada et al. (2006) outlined the importance of the maintenance systems supporting the decision-making tools. They were able to schedule new preventive and corrective maintenance tasks as part of the decision making process under a multi-agent distributed control system. In that way, they modify maintenance policy when uncovering plant faults, thus considering decision-making at mid-term not at real

time. Gheorghe et al. (2005) presented a decision support scheme under a risk approach applied to safety in rail transportation. The framework gathers relevant safety information related to technical infrastructure, human action, management procedures and determine the societal and potential accident risks, also revealing possible causes of the accidents.

Several works (Wörn et al., 2002) have theoretically considered the necessity of reflecting on their overall decision systems support a final advice generation stage, but few have been who have provided practical solutions. An interesting application was given by Dash and Venkatasubramanian (2003) who partially addressed the decision-making problem by providing causes and consequences associated to some biased variables. Although they did not show a methodology to propose final decision support in terms of corrective actions or recommendations, made of the PHA-AEM integration a new source of information upon which to obtain additional decision-making support.

In summary, reported works facing abnormal events normally stop at the diagnosis step by finding out key information of process deviations. However, later and fundamental decision support steps which can help to clarify diagnosis and to propose corrective actions protocols are hardly found in the literature. Global AEM systems should take into account these supporting tools. However, there is an evident lack of practical implementations which demonstrate in which way these modules integration affect the general AEM performance.

CHAPTER 3

Thesis Outline: An AEM Proposal

This thesis proposes a global approach for the AEM of chemical plants. Thus, it addresses the whole chain of tools required for the AEM. In this thesis not just novel solutions are given for each chain module but additional linking systems and synergies are obtained as a result of the general approach considered. The contributions are directed towards two different aspects related to limitations uncovered on the literature review (chapter 2). In one hand, specific works are focused on improving each of the modules involved in the AEM chain (sensor network, data rectification, fault detection/diagnosis and decision-making support system). On the other hand, some solutions to allow an integrated and coordinated AEM policy are also proposed to improve the general framework.

Specific contributions of this thesis address some of those subjects not properly tackled in the literature review (chapter 2). They are summarized next in order to clarify the thesis objectives:

Sensor Networks design

- Lack of practical SN designs from the diagnosis and AEM point of view.

Data rectification

- Lack of studies approaching data reconciliation from the on-line perspective to solve real time difficulties.

Fault detection and diagnosis

- Lack of FDS evaluation standards to confront different methodologies.
- Deficiencies on isolated diagnosis tools performance, for instance, the application of Machine Learning area to chemical processes fault diagnosis.
- Required proper combination of tools in order to overcome some techniques limitations.

- Opportunity of integrating several diagnosis systems to take advantage of complementary capabilities from their natural differences.
- Lack of intelligent information exchange from the FDS level to other levels of the process operations hierarchy (Figure 3.1).
- Poor management of simultaneous faults. Adoption of limited approaches focused on solving the partial diagnosis problem which ignore possible faults simultaneity.

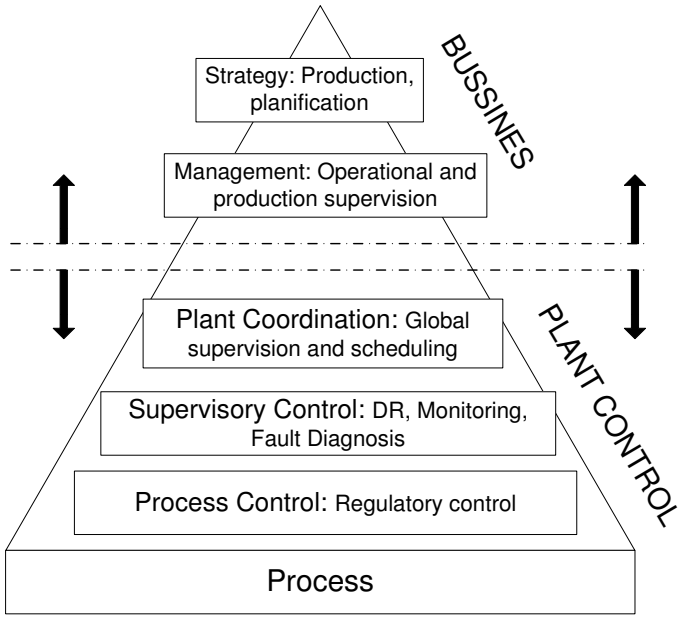


Figure 3.1: Process operations hierarchy.

Decision-making support system

- Scarce contributions on practical systems to support the decision-making process.

The objective of this thesis is to address these limitations by proposing new contributions at each module involved in the AEM.

Contributions to Sensor Network Design

A simple and efficient genetic algorithm based methodology was proposed as a novel way of adjusting sensor network design cost while keeping the system diagnosis performance.

Contributions to Data Reconciliation

Data reconciliation was improved on its on-line version by adjusting usual time delays between measured process variables. A genetic algorithm optimization approach was

applied to enhance the reconciliation results of an Extended Kalman filter applied to an industrial case study.

Contributions to faults diagnosis

The efforts to improve the isolated diagnosis tools mainly focused on the process history based methods, owed to they are more adapted to the intrinsic characteristics of chemical processes. In one hand, difficult processes which chemical industry have to deal with makes the development of accurate mathematical models or causal knowledge relationship between process variables more difficult than it would be desired. That fact makes these kind of methodologies unsuitable for many industrial problems. On the other hand, the availability of big amounts of process data from complex implemented instrumentation networks of current plants, allows easily developing data based methods to tackle with the fault diagnosis issue. Then, the key will be how to properly take advantage of such overwhelming amount of information in form of data to increase the diagnosis efficiency.

The general target at that step was to build a diagnosis framework able to gather the individually required characteristics of a good FDS (Venkatasubramanian et al. (2003b) and section 2.3.4). In that sense, improvements on isolated diagnosis tools, diagnosis tools combination, merging of different FDS and integration of the FDS level on the process operations hierarchy (section 2.3.4) were carried out.

- Support Vector Machines (SVM), a recently developed algorithm from machine learning area was applied as a novel quantitative data-based diagnosis technique in order to widen the range of fault diagnosis available techniques. It was applied to the diagnosis of a challenging problem as well as used to improve diagnosis of simultaneous faults.
- A MultiLabel approach (from machine learning area) was adopted to address simultaneous faults. Under that view only single faults are trained, thus the system considers any faults simultaneity.
- Data based fault diagnosis systems performance was enhanced by means of improving the problem information representation.
- A combination of tools was proposed to generate a complemented fault diagnosis. It was done by connecting a PCA detection module with a Fuzzy Logic Knowledge Rules system. This second FDS combines complementary benefits of PCA as a monitoring tool with reasoning and interpretation capabilities of a fuzzy logic expert system.
- The final contribution related to fault diagnosis consisted of the integration of different FDSs. A general framework which includes different FDSs and a conflict diagnosis solver was proposed. It balances each independent FDS response and decides the most probable plant state based on previous FDSs knowledge. The more different and complementary FDSs, the more benefits achieved by the integration.

Contributions to the decision-making support

The corrective actions in most of industrial processes strictly depends on the ability, knowledge and expertise of plant operators. However, operators are normally subject

to high pressure when making decisions under the presence of process faults due to urgencies and lack of clear diagnostic information. As a result, most of industrial accidents and economical losses are due to human errors (i.e., wrong corrective actions). Therefore, the need of a computer aided decision support system to assist process operators in understanding and assessing process status becomes crucial.

The thesis contribution to the decision making stage is not just a new way of generating corrective action protocols but a PHA and AEM practical integration to support the final decision-making process.

General AEM approach

These previously mentioned contributions are included on the modules which conform the general AEM chain. However there are other contributions that could be better evaluated after showing this thesis AEM system proposal.

The proposed AEM framework is wrapped by the dotted line in Figure 3.2. The system receives as inputs, data and knowledge from plant through the SN and off-line safety analysis (Process Hazard Analysis at the design phase) respectively, and survey supporting corrective actions to the experts in charge of the process.

In that framework, all the crucial modules on the AEM policy are included, from sensor network to the decision-making support system. However some additional contributions are here included and should be outlined.

The conflict solver module is a novel solution built to manage simultaneously different diagnosis systems. Another general contribution is the Process Hazards Analysis and AEM practical integration to support the final operator decision-making. It is represented by the Process Knowledge module in Figure 3.2 which incorporates off-line Process Hazards Analysis into the on-line AEM.

It is important to note that the presented AEM system is arranged to support process operators in the decision-making process. Thus, the information flows outlined on Figure 3.2 are those required to provide on-line advises to the person in charge of the process. Nevertheless, there are other information exchanges between modules that can be established to improve the overall AEM system performance. Thus, difficulties at any of the presented modules can be better addressed by incorporating additional design criteria from other chain's links than investing great efforts on the specific challenging module.

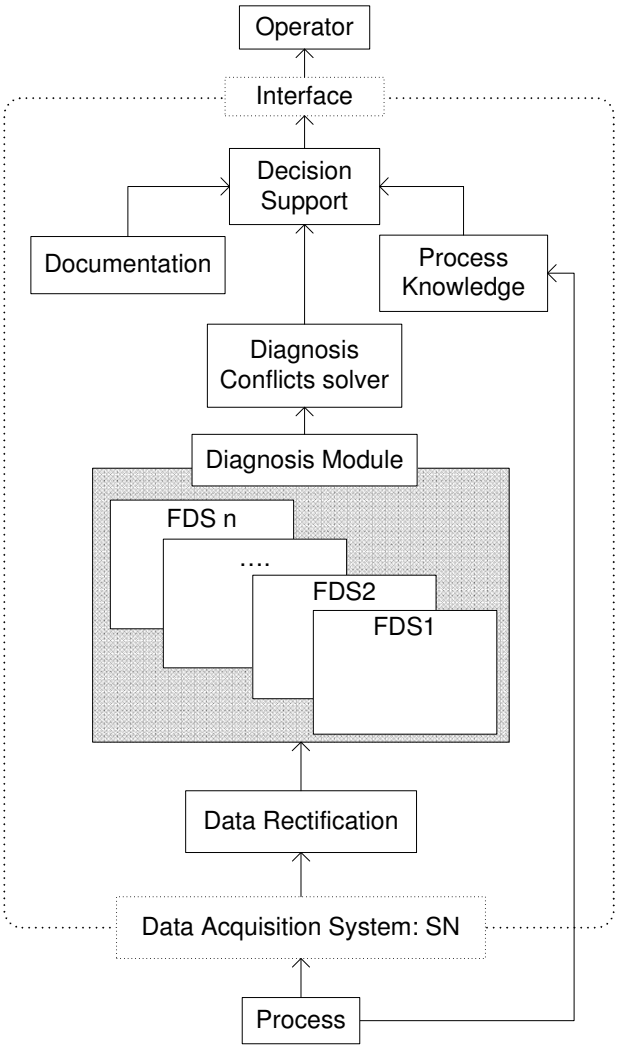


Figure 3.2: Abnormal Events Management system proposal.

CHAPTER 4

Acquiring and Verifying Valuable Process Data

One of the keys to make correct decisions in plant is having in every moment trustworthy and precise information about the process. That's why the stage of capture and ratification of information is a critical point to extract true knowledge of the condition of the plant. This chapter centers principally on the ratification of information, although it also includes a mechanism of sensor placement based on fault diagnosis criteria.

4.1 Introduction

As many of PSE developed tools (monitoring and diagnosis systems, optimization systems, etc.) rely on process data, SN design becomes critical on new operating chemical plants. So, the SN design issue requires of a thorough study to select the most suitable design criteria depending on the goals to accomplish. The design requires accounting for different aspects as the sensors type, number, reliability or position, while keeping some economy thresholds as for example minimum acquisition and installation costs.

Observability has been one of the most widely used criteria in the sensor placement research. Sensors failures may lead to a reduction of measurements affecting seriously control, monitoring and optimization systems, hence the whole process performance. Thus, observability makes possible to observe key process variables even if one or more sensors fail. Besides, other criteria as the sensor accuracy or reliability have also been widely considered in the sensor placement research community.

During last years the optimal design of SNs has received an increasing attention. A wide variety of approaches relying on exhaustive enumeration, rigorous mathematical models, etc., were developed to address the complexity of SN design problem. Within this context, the sensor placement can be regarded as a high combinatorial optimization problem where the main goal is to find the optimal balance between the indicators cost and the network performance.

On the other hand, in most of the papers dealing with fault diagnosis it is assumed that the SN is already installed and the relationship between sensor location and the FDS performance is rarely discussed. In this chapter, both issues are considered and a method to optimally design a SN which on-line supplies reliable data to a FDS is presented. Sensors are placed to minimize the sum of investment costs while keeping the FDS performance when all original measured variables are considered.

Once the SN measures the chosen process variables, a data validation is also essential for checking the information consistency as well as for obtaining more accurate information. Data Reconciliation (DR) is a model-based filtering technique that attempts to reduce the inconsistency between measured process variables from plant sensors and a process model. Thus, by decreasing this inconsistency, measurement errors are reduced and more accurate process variable values are obtained for their use on process control, monitoring and optimization.

When applying DR one has to be careful of avoiding circumstances that can harm the rectification effectiveness. In that sense, performance of DR can be seriously affected by the presence of events that increase variables inconsistency such as modeling errors, gross-errors and/or delays in sampling data. In such situations, an error in the estimation can be generated by forcing a matching of the process model and measured signals that are incompatible. In order to deal with the presence of gross-error(s) several approaches have been presented in the literature (Narasimhan and Jordache, 2000), covering steady and dynamic systems as well as different kinds of errors (e.g. bias, process leaks). However, less attention has been devoted to tackle with situations in which delays are present. Since time delays appear as a time dependent variable, they introduce non-linearities even if the process is linear.

Correlation measures have been previously used in order to determine process delays. Wachs and Lewin (1999) proposed an algorithm that applies relative shifts between process signals within a process data matrix in order to find out the position in which the correlation among variables is maximum. The optimal shifts are those that minimize the determinant of the associated correlation matrix. Such mechanism was applied for improving resolution of a PCA based fault diagnosis system.

When the number of process variables n is large or the maximum delay d_{max} is high, in terms of sampling time, exhaustive search of the maximum correlation by direct data matrix manipulation is impracticable due to the unmanageable combinatorial size (it involves $(d_{max})^n$ determinant calculations). Wachs and Lewin (1999) realized this problem and proposed a new algorithm that reduces the problem size to $d_{max} \times s \times (n - s)$, where s is the number of inputs. However, this algorithm assumes that the output variables are correlated among themselves with no delays present and that the process inputs are independent. Apart of that limitation, the reduced but exhaustive searching strategy maintained in the methodology, makes it unsuitable for actual industrial problems involving continuous (not discrete) time delays. Since real industrial problems have to deal with continuous operation times, such delays identification methodology becomes infeasible at the on-line implementations because of the CPU time limitations. In that sense, more efficient and systematic approaches should be developed in order to deal with real time delays identification.

In addition to the SN design procedure previously described, this chapter also presents an efficient Time Delay Identification (TDI) procedure to improve real time data reconciliation. This TDI is based on a genetic algorithm (GA) optimization which determines all the variable delays with respect to a reference process signal (e.g. the most or the least delayed process signal). These time delays are later removed from the measurements error model to improve the Dynamic Data Reconciliation (DDR) performance. An academic case study with recycles and a challenging industrial control problem, the Tennessee Eastman (TE) benchmark (Downs and Vogel, 1993), were selected to illustrate the advantages of the DDR approach. Thus, the main goal of that part of this chapter is to properly combine DDR and TDI in such a way that data delay mismatches can be minimized.

4.2 Data acquisition: A diagnosis-based sensor network design

The proposed data acquisition system design consists of constraining the SN for only measuring some process key variables while maintaining the FDS performance. These variables must gather the essential process information from the diagnosis point of view. An optimization method that assures process observability and a minimum demanded reliability of process key variables is applied to allocate the required instrumentation at a minimum investment cost.

4.2.1 Problem formulation

The complete proposed procedure can be divided in two consecutive steps. Firstly, the diagnosis key variables must be found. Then, the SN design methodology focuses on these variables, which are crucial for the correct supervision of the process.

4.2.1.1 Key diagnosis variables searching approach

The presented approach use an optimization procedure to extract the key diagnosis variables. It consists of minimizing the number of process variables to be measured by considering some diagnosis performance constraints. Because of the high non-linearity shown by most of Fault Diagnosis Systems (FDS), a genetic algorithm (GA) was proposed to search this diagnosis key variables (see Figure 4.1).

GA constitutes a vast class of stochastic algorithms used to solve optimization problems (Holland, 1975; Goldberg, 1989). In comparison with other stochastic procedures for optimization, GAs consider many points in the search space simultaneously and therefore have a reduced chance of converging to local optima (Karr, 1993).

In the classical GA formalism (Figure 4.1), a set of N_{ind} candidate solutions (population) is generated randomly. Each potential solution (individual) at the multidimensional search space is coded as a vector called chromosome (which consists of a string of genes, each one representing a feature). The goodness of each individual/chromosome at the population is evaluated by utilizing a pre-specified fitness criterion Φ. Upon assessing fitness of each chromosome in the population, a new generation of individuals is created from the current population by using the selection, crossover and mutation operators.

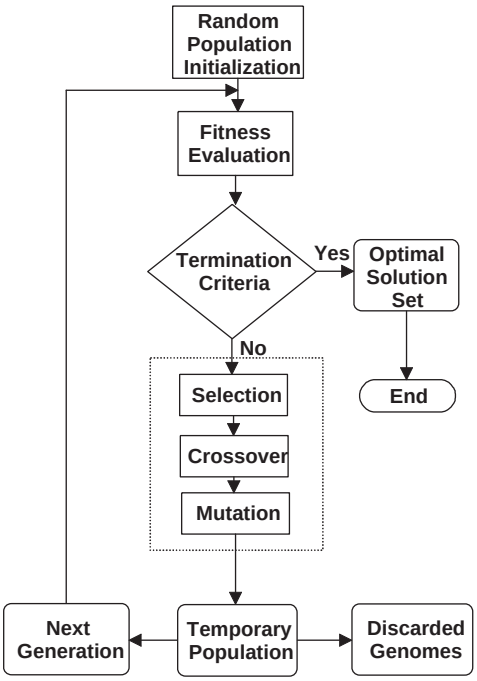


Figure 4.1: Genetic algorithm loop.

Here, the selected chromosome (z) is a vector of n binary variables corresponding to any of the considered process variables. The generic chromosome corresponding to the set of measured process variables (MPV) is given next:

$$MPV_z = [v_{1,z}, v_{2,z}, \ldots, v_{i,z}, \ldots v_{n,z}], \qquad v_{i,z} \quad \epsilon \quad [0, 1] \tag{4.1}$$

where n is the number of considered variables and $v_{i,z}$ are binary variables which determines if the corresponding process variable will or will not be measured and used in the diagnosis system. Hence, if $v_{i,z}$ equals 1 the corresponding process variable will be used in the FDS design and its values will be later collected to infer the on-line process diagnostic.

The fitness criteria which drives the key diagnosis variables search relies on two different factors. The first one is the diagnosis performance represented by a quantitative comparative index: Diagnosis Performance (DP). In this work, Recall was used as the comparative index. The second factor weights positively these individuals which involve less process variables. This second factor is called Variables Reduction Factor (VRF) and it proportionally increases as less are the process variables considered in the individual (z). It accounts for the difference between variables involved ($v_{i,z} = 1$) on the current considered individual regarding the complete set of process variables. Besides that, the optimization is subject to a constraint which, weighting negatively these individuals that reduce the initial FDS performance, prevents reductions in the set of variables at the expense of FDS performance. Next, the optimization problem is formulated:

$$\begin{aligned} Max \qquad & \Phi_z = DP_z + VRF_z \\ \text{subject to} \qquad & DP_z \geq \gamma * DP_o \end{aligned} \tag{4.2}$$

where γ is a parameter representing the percentage of the diagnosis performance that the system allows to be reduced by removing some of the original process variables. DP_z and DP_o are the diagnosis performances for each individual (z) and the original individual (o), which considers all the process variables ($v_{i,z} = 1, \forall i$), respectively. VRF_z is formulated in equation 4.3. $alpha$ allows the appropriate scaling of VRF_z at equation 4.2.

$$VRF_z = \alpha * (n - \sum_{\forall i} v_{i,z}) \tag{4.3}$$

Thus, the final solution is an individual which provides the minimum process variables to be measured keeping close (by $\gamma \approx 1$) to the original FDS performance (DP_o).

4.2.1.2 Sensor network design procedure

Once the key diagnosis variables have been selected, a MINLP-based optimization approach (Angelini et al., 2006a) was adopted to focus on the cost and reliability of the SN. Thus, the SN design procedure minimizes the instrumentation investment cost (IC) while guaranteeing observability and high reliability on the diagnosis key variables. The complete model applied in this chapter was given by Angelini et al. (2006a). Equation 4.4 shows a summary of the complete formulation (Angelini et al., 2006a).

$$\begin{aligned} Min \qquad & IC = \sum_{\forall i} \sum_{\forall k} c_k n_{i,k} \\ \text{subject to} \qquad & r_i^{\min} \leq r_i \end{aligned} \tag{4.4}$$

c_k represents the investment cost of sensor k including the acquisition, installation and de-installation costs. $n_{i,k}$ is a binary variable meaning that variable i is going to be measured by sensor k. The constraint specified on equation 4.4 guarantees a minimum reliability at any of the considered process variables.

4.2.2 Validation of the methodology

The methodology was checked in the TE challenging industrial problem (Downs and Vogel, 1993) by using a data based FDS. This case is detailed on subsection 4.6.2. The FDS is based on a Principal Component Analysis (PCA) detection module integrated with a rule-based fuzzy-logic system that on-line interprets the statistics calculated from PCA (Musulin et al. (2006) and chapter 7). In order to estimate its performance, Recall index defined as the rate of right diagnostics and the total diagnosis responses (see chapter 5) was adopted as the DP index. Parameter γ was made equal 0.98 for guaranteeing at least a 98% of FDS Recall regarding the original FDS performance, when all process variables were considered.

In order to check the SN design procedure benefits, three different design cases were taken into account:

- Design when the key variables are not known.

- Retrofit, knowing which are the key variables, of a SN solution evaluated without knowing which were the key variables.
- Design knowing which are the key variables.

Results show how knowledge on key variables at the design phase, allows accomplishing with the reliability requirements with less investment than that needed during the retrofitting phase. Results obtained by applying the MINLP (Angelini et al., 2006a) take into account the sensor catalogue shown at Table 4.1 (LM and FM stand for level and flow meters respectively), where AC, IC and DC mean sensor acquisition, installation and de-installation costs, and k is the generic name for each sensor. It is worth to remark that the method used for evaluating the functional reliability equation of process variables only deals with mass flow balances (linear cases); this means that the attention is paid to the process variables involved on the flow balances. Within these flow balances only variables F2, F9, F10, F14, and F15 were evaluated as FDS key variables.

Table 4.1: Tennessee Eastman Process: Sensor catalogue.

Sensor	AC_k (m.u.)	IC_k (m.u.)	DC_k (m.u.)	Reliability	Type
S1	250	100	50	0.70	FM
S2	500	100	50	0.75	FM
S3	800	100	50	0.80	FM
S4	1700	200	100	0.85	FM
S5	200	200	100	0.90	FM
S6	300	100	50	0.75	LM
S7	500	100	50	0.80	LM
S8	800	100	50	0.85	LM
S9	1000	100	50	0.90	LM

Table 4.2 shows the optimal SN attained when considering reliability constraints of Table 4.3 and multiplicity equals 2 (only two sensors allowed for measuring each variable). Total SN cost was 5400 monetary units (m.u.), from which 3900 are due to the AC_k (3000+900) and 1500 are due to IC_k (1200+300). The solution consists of using 12 s1-type sensors (FMs) and 3 s6-type sensors of (LMs). Note that $N_{i,k}$ relates sensors (k) to process variable (i).

Table 4.2: Initial optimal sensor network.

$N_{i,k}$	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F14	F15	L1	L2	L3	SR_k	AC_k	IC_k	DC_k
S1	1	1	1	1	1	1	0	1	1	1	1	1	1	-	-	-	0.7	3.0	1.2	0.0
S2	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.8	0.0	0.0	0.0
S3	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.8	0.0	0.0	0.0
S4	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.9	0.0	0.0	0.0
S5	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.9	0.0	0.0	0.0
S6	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1	1	0.8	0.9	0.3	0.0
S7	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.8	0.0	0.0	0.0
S8	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.9	0.0	0.0	0.0
S9	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.9	0.0	0.0	0.0

Table 4.3: Tennessee Eastman Process Reliability Constraints.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F14	F15	L1	L2	L3
r	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8

Table 4.4 shows a retrofitting design obtained from the previous solution. The corresponding reliability constraints are shown in Table 4.5. Now, minimum reliability demands for the FDS key variables were increased. In order to accomplish with the new reliability constraints 2200 additional m.u. were required. The additional cost is due to 1550 m.u. corresponding to *AC*, 600 m.u. corresponding to *IC* and 50 m.u. corresponding to *DC*. Thus, in order to increase key variables functional reliability up to 0.98, one needs 5 additional sensors (3 S1 sensors, 1 S2 sensor and 1 S6 sensor) regarding the initial design.

Table 4.4: Optimal Sensor Network: Retrofitting.

$N_{i,k}$	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F14	F15	L1	L2	L3	SRk	ACk	ICk	DCk
S1	0	**2**	**1**	**1**	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	-	-	-	0.70	0.75	0.40	0.05
S2	**1**	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.75	0.50	0.10	0.00
S3	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.80	0.00	0.00	0.00
S4	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.85	0.00	0.00	0.00
S5	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.90	0.00	0.00	0.00
S6	-	-	-	-	-	-	-	-	-	-	-	-	-	**1**	**1**	**2**	0.75	0.30	0.10	0.00
S7	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.80	0.00	0.00	0.00
S8	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.85	0.00	0.00	0.00
S9	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.90	0.00	0.00	0.00

Table 4.5: Tennessee Eastman Process Reliability: Retrofitting.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F14	F15	L1	L2	L3
r	0.8	0.98	0.8	0.8	0.8	0.8	0.8	0.8	0.98	0.98	0.8	0.98	0.98	0.8	0.8	0.8

Table 4.6 shows the SN solution evaluated considering the reliability constraints values depicted in Table 4.5. The solution counts on 16 S1 sensors, 1 S2 sensor and 3 S6 sensors for a total cost of 7400 m.u., which represent 200 m.u. less than the previous solution evaluated as a retrofitting design. Thanks to the knowledge in advance of the key variables, it is possible to accomplish the reliability requirements with lower investment.

4.3 Data Reconciliation

DR is a technique that takes advantage of the redundancy between the measurements model (Eq. 4.5) and the process model (Eq. 4.6). Therefore, the existence of both models is a prerequisite to reconcile process measured data.

The measurements model is constructed by assigning to each one of the n measured process variables a normal probability distribution around its true value. In the absence of

Table 4.6: Optimal Sensor Network: Retrofitting knowing key variables from the beginning.

$N_{i,k}$	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F14	F15	L1	L2	L3	SRk	ACk	ICk	DCk
S1	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	-	-	-	0.70	4.00	1.60	0.00
S2	0	**1**	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.75	0.50	0.10	0.00
S3	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.80	0.00	0.00	0.00
S4	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.85	0.00	0.00	0.00
S5	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	0.90	0.00	0.00	0.00
S6	-	-	-	-	-	-	-	-	-	-	-	-	-	**1**	**1**	**1**	0.75	0.90	0.30	0.00
S7	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.80	0.00	0.00	0.00
S8	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.85	0.00	0.00	0.00
S9	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0.90	0.00	0.00	0.00

gross-errors the measurements model can be written as:

$$\mathbf{y}_{t_k} = \mathbf{y}^*_{t_k} + \varepsilon_{t_k}, \quad \mathbf{y}_{t_k} \in \mathbb{R}^n \tag{4.5}$$

where $\mathbf{y}$ is the $n\times 1$ measurements vector, $\mathbf{y}^*$ is the $n\times 1$ vector of unknown true values, and ε stands for the $n \times 1$ vector of random measurement errors whose expected value is the null vector and has a known positive definite variance matrix (measurements are assumed to be independent). Additional information must be introduced through the process model equations (constraints). The process model is in general represented by a set of differential equation constraints:

$$f\left(\frac{d\widehat{\mathbf{y}}_{t_k}}{dt}, \widehat{\mathbf{y}}_{t_k}, \widehat{\delta}_{t_k}, \widehat{\theta}_{t_k}\right) = 0, \ f \in \mathbb{R}^f \tag{4.6}$$

algebraic equality constraints:

$$g(\widehat{\mathbf{y}}_{t_k}, \widehat{\delta}_{t_k}, \widehat{\theta}_{t_k}) = 0, \ \ g \in \mathbb{R}^g \tag{4.7}$$

and inequality constraints:

$$h\left(\widehat{\mathbf{y}}_{t_k}, \widehat{\delta}_{t_k}, \widehat{\theta}_{t_k}\right) \leq 0, \ \ h \in \mathbb{R}^h \tag{4.8}$$

where $\widehat{\mathbf{y}}_{t_k}$ represents the values of the estimated vector at discrete time t_k and $\widehat{\delta}_{t_k}$ and $\widehat{\theta}_{t_k}$ are the $p \times 1$ vector of estimated unmeasured variables and the $q \times 1$ vector of adjusted model parameters at discrete time t_k respectively.

The adjustment of measurements in order to compensate random errors usually leads to a constrained optimization problem. In most of applications, the objective function (OF) to be minimized is the squared and weighted difference between measured values and the corresponding process model estimations.

$$OF(\mathbf{y}, \widehat{\mathbf{y}}, \sigma) = \sum_{t_k=0}^{K} [\widehat{\mathbf{y}}_{t_k} - \mathbf{y}_{t_k}]^T \Sigma^{-1} [\widehat{\mathbf{y}}_{t_k} - \mathbf{y}_{t_k}] \tag{4.9}$$

where Σ is a diagonal matrix $\Sigma = diag(\sigma_1^2, ..., \sigma_n^2)$ and σ_i^2 is the variance of the $i - th$ measured variable. Different approaches have been proposed to solve this optimization problem (Liebman et al., 1992; Albuquerque and Biegler, 1996). If the process does not show high nonlinearities and no inequality constraints are present, Kalman Filter (KF) can be efficiently used (Lewis, 1986).

4.3.1 Kalman filter

Kalman filter (KF) is an efficient recursive filter which estimates the state of a dynamic system from a series of incomplete and noisy measurements, assuming that the true current state evolves from the previous one. The algorithm uses the following state-space and measurements models respectively:

$$\mathbf{x_{t_k}} = A_{t_k} x_{t_{k-1}} + B_{t_k} u_{t_k} + w_{t_k} \tag{4.10}$$

$$\mathbf{y_{t_k}} = H_{t_k} x_{t_k} + v_{t_k} \tag{4.11}$$

where $\mathbf{t_k}$ represents a sampling time, $\mathbf{x_{t_k}}$ is the $\mathbf{n_x}$ dimensional vector of state variables, $\mathbf{u_{t_k}}$ is the $\mathbf{n_u}$ dimensional vector of manipulated input variables and $\mathbf{y_{t_k}}$ is the $\mathbf{n_y}$ dimensional vector of measured variables. The state transition, $\mathbf{A_{t_k}}$, the control gain, $\mathbf{B_{t_k}}$ and the observation, $\mathbf{H_{t_k}}$, are matrices of an appropriate dimension and if their coefficients are time-independent the subscript $\mathbf{t_k}$ can be removed.

The Kalman filter assumes errors in the process model and in the measured data. Process noise $\mathbf{w_{t_k}}$ represents errors in the state transition model. This noise is assumed to be white, with zero mean and a variance Q. $\mathbf{v_{t_k}}$ represents the measurement noise with variance R. Using process measurements, the error covariance matrix $\mathbf{P_{t_k/t_k}}$ associated with the estimated state vector $\widehat{\mathbf{x}}_{\mathbf{t_k/t_k}}$ is updated as follows:

$$\mathbf{P_{t_k|t_k}} = (I - K_{t_k} H_{t_k}) P_{t_k|t_{k-1}} \tag{4.12}$$

where $\mathbf{K_{t_k}}$ is the Kalman filter gain given by:

$$\mathbf{K_{t_k}} = P_{t_k|t_{k-1}} H_{t_k}^T (H_{t_k} P_{t_k|t_{k-1}} H_{t_k}^T + R_{t_k})^{-1} \tag{4.13}$$

Then, process expected measurements and current state estimation can be evaluated by means of equations 4.14 and 4.15 respectively.

$$\widehat{\mathbf{y}}_{\mathbf{t_k}} = y_{t_k} - H_{t_k} \widehat{x}_{t_k|t_{k-1}} \tag{4.14}$$

$$\widehat{\mathbf{x}}_{\mathbf{t_k|t_k}} = \widehat{x}_{t_k|t_{k-1}} + K_{t_k} \widehat{y}_{t_k} \tag{4.15}$$

where $\widehat{x}_{t_k|t_k}$ and $\widehat{y}_{t_k}$ are the state and measurements predictions given by the model. $y_{t_k} - H_{t_k} \widehat{x}_{t_k|t_{k-1}}$ is called the residual or innovation and reflects the discrepancy between the predicted measurements $H_{t_k} \widehat{x}_{t_k|t_{k-1}}$ and the real measurement y_{t_k}.The gain matrix K_{t_k} is a factor that allows to weight more or less the residual depending on the confidence of the system in the process measurements or in the previous process estimations.

In industrial cases, applicability of simple Kalman filter is limited because of the common non-linear nature of real life problems. Thus, more accurate non-linear techniques, as the Extended Kalman filter, have been proposed to deal with these problems in the literature.

4.3.2 Extended Kalman filter

In case the process to be estimated or the measurements relationship are not linear, the simple Kalman filter is in many cases not a sufficiently accurate state estimator. In such

cases, the Extended Kalman Filter (EKF) that linearizes about the current mean and covariance is the best choice.

EKF assumes that the process has an associated state vector $x \in \mathbb{R}^n$, but now the process is ruled by the next *non-linear* stochastic difference equations:

$$\mathbf{x_{t_k}} = \gamma(x_{t_{k-1}}, u_{t_k}, w_{t_{k-1}}) \tag{4.16}$$

$$\mathbf{y_{t_k}} = \lambda(x_{t_k}, v_{t_k}) \tag{4.17}$$

where the function γ is used to compute the predicted state from the previous estimate and similarly the function λ is used to compute the predicted measurement from the predicted state. However γ and λ cannot be applied to the covariance directly. Instead a Jacobian matrix is computed. At each time-step (t_k) the Jacobian is evaluated with current predicted states. Basically, what this technique does is to linearize the non-linear function around the current estimate allowing to update the error covariance matrix $P_{t_k|t_k}$, to estimate the current process state $\widehat{x}_{t_k|t_k}$ and the process measurements vector $\widehat{y}_{t_k}$, by using the real process measurements, as it was discussed in the simple Kalman filter. In practice, the individual values of the noise w_{t_k} and v_{t_k} are not known, but the state and measurement vector can be approximated.

For more details on the EKF formulation we refer interested readers to the work developed by Lee and Ricker (1994) which also includes the models and one of the case studies used in that chapter.

4.4 The role of time delay in data reconciliation

In most cases, Kalman filter and EKF could be able to manage certain modeling errors by assuming uncertainties in the process model. However, a poor performance can be obtained when data correlation is seriously affected by unknown time delays between measured process variables. This section presents an efficient method for identifying these time delays in order to overcome such limitation.

4.4.1 Time delay identification

Time delays are present in almost all industrial processes and normally affect the control, monitoring and DDR systems performance. Nevertheless, usual approaches to address these issues do not take into account these delays or simply consider them as known and constant over time.

Two types of time delays can be distinguished:

1. **Process Related Delays (PRD):** They are caused by intrinsic process dynamics, the performance of the implemented control algorithm and/or the way that sensors were placed around the plant (e.g. thermocouple or flow-meter in a pipeline); generally these delays are propagated to the subsequent part of the process under consideration and are unknown for process operators.

2. **Sensor Related Delays (SRD):** They provoke a local effect and can be obtained from suppliers specifications. Nevertheless, from DR perspective, these "local"

delays will certainly induce a biased estimation on the rest of involved redundant process variables.

Although different, both delays cause a decrease in the DR performance. Independently of which is the delay source, the knowledge of delays between process variables will be crucial to obtain better DDR results.

As it was discussed in previous sections, the approach presented by Wachs and Lewin (1999) assumes that output process variables are correlated among themselves with no delays present and that the inputs are independent. Nevertheless, in real processes it is difficult to discriminate between input and output variables, since this distinction is process unit dependent (a variable may be at the same time an output of a process unit and the input of the subsequent one). These distinctions are even more difficult to be assumed in industrial problems, where it is common the use of recycles and feedback control systems.

The TDI method presented here improves all the limitations discussed above. This approach also is able to tackle with real situations without requiring any a priori knowledge about the number and location of time delays or the process input and output variables.

Since TDI requires of an intense computational effort in those problems with high number of variables and delays, a genetic algorithm (GA) based optimization method was adopted to increase the search efficiency. The proposed stochastic-based optimization approach can not guarantee the global optimality of the solution. However, it is able to generate near-optimal solutions in a short time, which is crucial in on-line applications where techniques that give good solutions with modest CPU time are preferred to those that can only generate optimal solutions in a time that exceeds the dynamic process demands. GA-based methods take advantage of well known genetic schemes so as to be able to cope with numerous and complex optimization problems in a reasonable time (Shopova and Vaklieva-Bancheva, 2006). Moreover, it will be shown later that near-optimal solutions are also able to produce remarkable and constant improvements on the DDR performance.

4.4.2 A genetic algorithm based searching approach

In this work, each chromosome is a delay vector (**DV**) which is defined as it is shown in Eq. 4.18 (see Genetic Algorithm basis on subsection 4.2.1.1). It contains the delay of each process signal (d_i) regarding a reference signal. This reference signal can be the most or the least delayed process signal but any other one can be also selected.

$$\mathbf{DV} = [d_1, d_2, ..., d_n] \tag{4.18}$$

The fitness of each individual **DV** is evaluated considering the correlation among the process variables. Let us consider a data matrix:

$$\mathbf{Y} = [\mathbf{y}_{1,t_k}, \mathbf{y}_{2,t_k}, ..., \mathbf{y}_{n,t_k}] \quad t_k = 1, ..., m \tag{4.19}$$

containing data corresponding to m time samplings of n process variables. $\mathbf{y}_{i,t_k}$ represents a sampling of process variable i at time t_k. The error in $\mathbf{Y}$ is supposed to follow a normal probability distribution and must be normalized with zero mean and unit variance in each data vector (Narasimhan and Jordache, 2000). Let $\overline{\mathbf{Y}}$ be the corresponding normalized data matrix. Introducing **DV** into $\overline{\mathbf{Y}}$ allows to obtain a new delay adjusted data matrix (Y_{DA}):

$$\mathbf{Y}_{DA} = [\overline{\mathbf{y}}_{1,t_k-d_1}, \overline{\mathbf{y}}_{2,t_k-d_2}, ..., \overline{\mathbf{y}}_{n,t_k-d_n}] \tag{4.20}$$

The correlation matrix **R** for two random variables is formally defined as:

$$\mathbf{R}_{\mathbf{x},\mathbf{y}} = \frac{\mathbf{cov}(\mathbf{x},\mathbf{y})}{\sigma_x \sigma_y} \tag{4.21}$$

where $cov(x, y)$ is the covariance of both variables and σ_x and σ_x are the respective standard deviation of each variable. Then, extending that formulation to $\mathbf{Y_{DA}}$, the correlation matrix of the adjusted variables matrix can be easily evaluated by:

$$\mathbf{R}_{\mathbf{Y_{DA}}} = \frac{\mathbf{cov}(\mathbf{Y_{DA}})}{\prod_{i=1}^{n} \sigma_{\overline{y}_{i,t_k-d_i}}} \tag{4.22}$$

The norm of **R**, that give us quantitative information about the correlation of matrix $\mathbf{Y_{DA}}$, is calculated as its maximum singular value, $maxSV(\mathbf{R})$. The fitness of **DV** is considered to be proportional to $maxSV(\mathbf{R})$. The complete procedure to evaluate the signals correlation is summarized next:

- Calculate the correlation matrix **R** of $\mathbf{Y_{DA}}$
- Calculate the norm of **R** as its maximum singular value $maxSV(\mathbf{R})$
- Assign to each individual a fitness proportional to $maxSV(\mathbf{R})$

To implement this algorithm, the Matlab genetic algorithm toolbox developed by of Sheffield (2003) was used.

4.4.3 Illustrative example

To demonstrate the performance of the proposed algorithm, a simple jacketed Continuous Stirred Tank Reactor (CSTR) is used as an illustrative example (Figure 4.2). An step rise on the tank temperature (T on Figure 4.3a) was forced by a control system command to appreciate intrinsic delays between process variables.

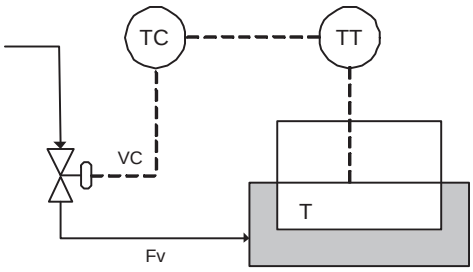


Figure 4.2: Illustrative example: jacketed CSTR.

Figure 4.3a shows the evolution of the monitoring measurements of the three main process variables: the valve command (V_c), the energetic agent flow through the valve (F_v) and the tank temperature (T).

Two types of delays are present:

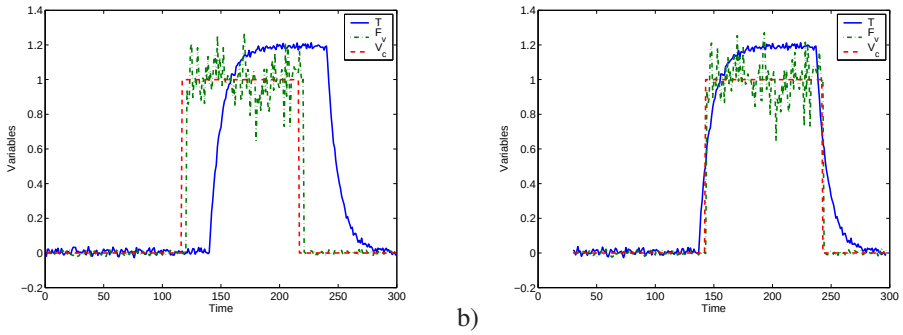


Figure 4.3: Illustrative example: a) Raw process measured data with actual delay, b) adjusted process data by using the genetic algorithm approach.

- The valve response is delayed from the command signal.
- The controlled variable T is delayed from the manipulated variable F_v.

The most delayed variable (T) was selected as the reference signal from a physical sense point of view, as the effect on the variable is the last appreciated on-line, however, any other could have been also chosen. The GA-based optimization method was run with the following parameters: 50 individuals, 10 generations, 0.7 crossover probability and 0.07 mutation probability. Also, a maximum delay of 32 sampling times was considered. In most of simple cases, maximum delay can be easily estimated by plotting the process variables, as shown in Figure 4.3a.

The algorithm converges to the best individual, $\mathbf{DV}_b = [0, 26, 29]$ (in sampling time units), in three generations. This is the same vector as the one obtained by means of an exhaustive search. Nevertheless, a slight difference with the real delay, $\mathbf{DV}_b = [0, 20, 24]$, was observed due to the inherent process dynamics. Adjusting delays by using the obtained delay vector ($\mathbf{DV}_b$) led to more consistent data (compare Figures 4.3a and 4.3b).

Figure 4.4 shows eigenvectors of the correlation data matrix corresponding to the illustrative example. It can be appreciated that the system variance is concentrated in the first eigenvector after the delay adjustment, that is, the process variance is explained in a unique direction. Therefore, by maximizing the variance explained by the first eigenvector, the process correlation is also maximized. Thus, delays have been correctly identified and their negative effect has been properly eliminated. It is important to note that in the proposed approach the combined effect of all process delays is globally compensated.

4.5 Integrating TDI into DDR

Integration of TDI and DDR can be achieved by considering the identified time delays within the measurement model (Eq. 4.5) as follows:

$$\mathbf{y}_{i,t_k} = \mathbf{y}^*_{i,t_k-d_i} + \varepsilon_{i,t_k-d_i}, \quad \mathbf{y}_{t_k} \in \mathbb{R}^n \tag{4.23}$$

In the case of sensor related delays, DDR can be performed simply using the known sensor delay specifications and modifying the measurement model according to Eq. 4.23. In

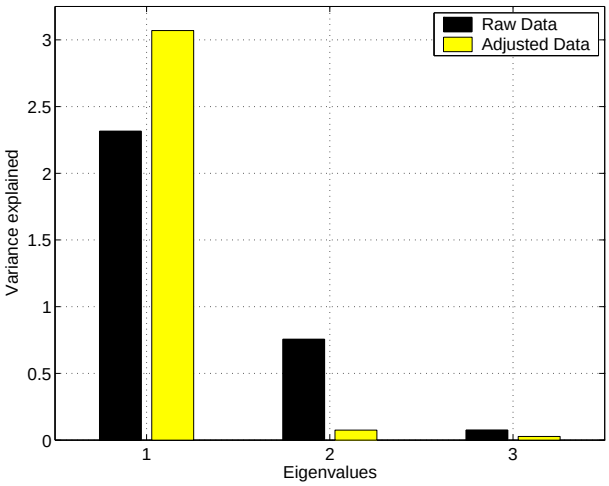


Figure 4.4: Variance explained in the eigenvectors of the correlation matrix. Illustrative example.

the case of process related delays or any unknown delay in the process (as those caused by the sum of SRD and PRD), TDI can be efficiently performed by using the approach proposed in the previous section. Although, the effect of delays in the process may not be completely compensated, the performance of the system can be highly enhanced as it will be shown in next section.

Time delays (**DV**) can dynamically change throughout time, so an on-line adjustment must be performed regularly in order to maintain the DR performance. A quick delay identification is crucial because of the on-line process demands. Therefore, the proposed GA searching procedure becomes a very attractive application for real-world industrial environments due to its quick performance. In that sense, **DV** is estimated periodically using the GA based procedure (Figure 4.5).

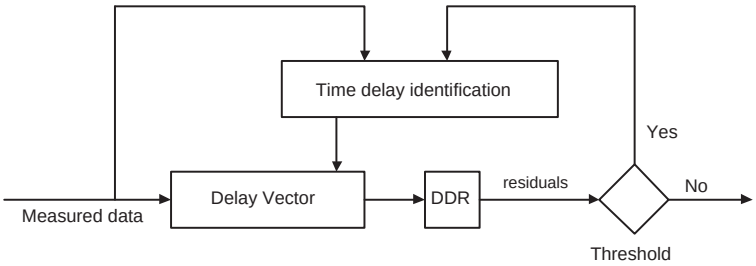


Figure 4.5: Adaptive time delay identification.

If the square sum of the residuals is less than a threshold value, the same **DV** is used to perform DDR but if the threshold is exceeded, TDI is activated in order to re-calculate **DV**. Residuals are evaluated based on the differences between process measurements and reconciled data. The evaluation is carried out periodically, in a pre-established frequency based ona process dynamic analysis.

Seeking to reduce computation time of TDI, current **DV** is included in the initial population of the new delay estimation. Although current **DV** may be outdated or erroneous, the proposed GA-based approach can quickly improve it through the mutation and crossover operations.

4.6 Case studies

4.6.1 Academic case study

The proposed methodology can be applied both to PRD and SRD. However, a case study presenting PRDs is first introduced to highlight the complexity of delay propagation effects. Figure 4.6 shows a flowsheet of the process considered to evaluate the proposed DDR methodology.

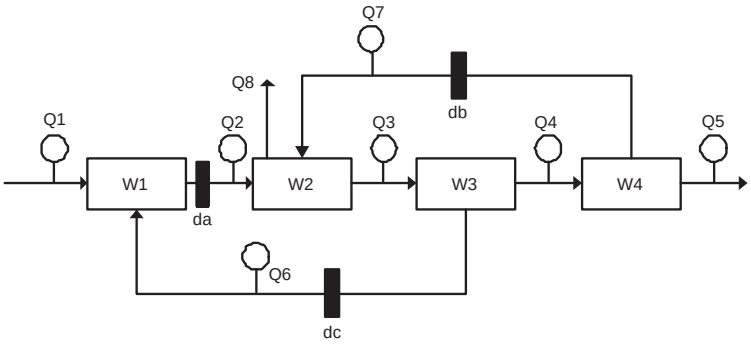


Figure 4.6: Academic case study with 3 added delays in positions indicated by the black boxes.

It was taken from the paper by Darouach and Zasadzinski (1991) and consists of eight streams and four storage tanks. In this example, the flow rates (Q) and mass hold-ups (W) are the process variables to be monitored. Because of the simplicity and linear relationships nature existing among monitored variables, the simple Kalman filter was enough to obtain an accurate and reliable dynamic data reconciliation. The good performance of the proposed DDR methodology is demonstrated next.

The original example (Darouach and Zasadzinski, 1991) is modified by adding three process delays ($[d_a, d_b, d_c] = [200, 100, 10]$, in sampling time units) at the positions shown in Figure 4.6. The combined delay effect leads to a considerable discrepancy between the mass balance (model) around tank 1 and the measured values. Therefore, a bias in the

DDR was expected.

In addition to the predefined delays, a disturbance due to a sudden but lasting change causes a modification in the operating conditions: the flow Q_1 rises off suddenly at sampling time equals 250 to reach the value of 34.5 and it is maintained during 250 samplings. Then, it returns to its original value of 29.5 (see Figure 4.7).

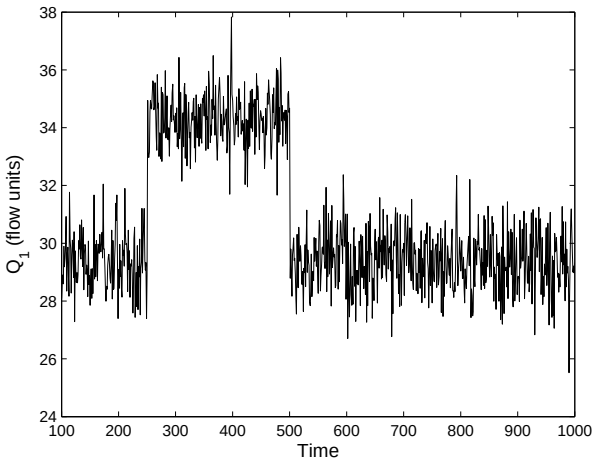


Figure 4.7: Academic case study: Q_1 measured data (disturbed during 250 samplings).

First, DDR was applied to the case study without any delay. The performance of such application can be seen in Figure 4.8, showing some ability to reduce the measurement noise and a notable estimation accuracy in variable Q_3 reconciliation.

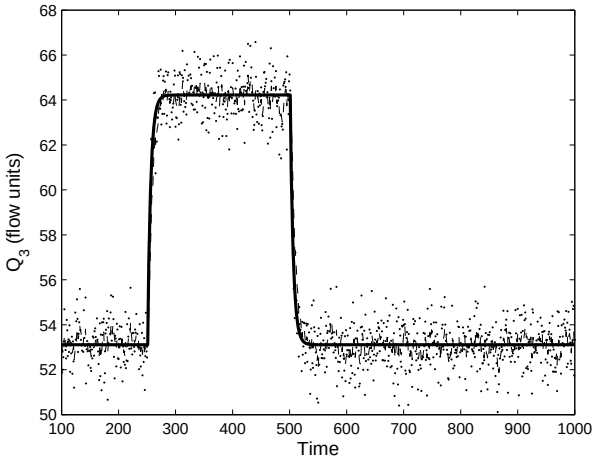


Figure 4.8: Academic case study: Reconcilied Q_3 values without delays.

In this figure the continuous line is the true value behavior, the discontinuous line is the estimated value and the points represent the measurements. In the case of incorporating the PRD, DDR can still perform correctly in some variables (Figure 4.9a) (due to the fact that KF can manage modeling errors) even when the nonlinearities introduced by the delay effect are present. Nevertheless, it shows unacceptable results in other variables where the combined delay effect is more critical (Figure 4.9b). Note that the inlet flow (Q_1) disturbance is now simulated at sampling time equals 500.

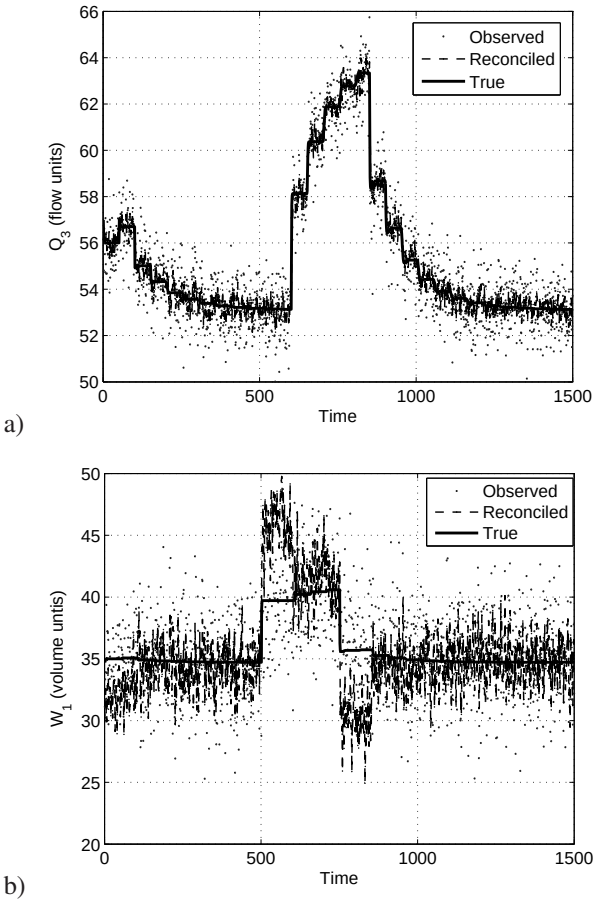
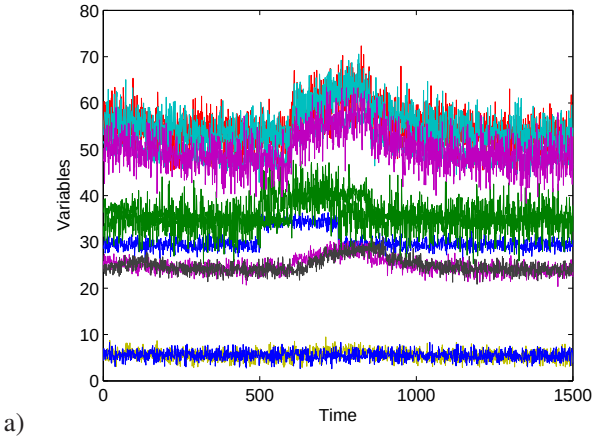


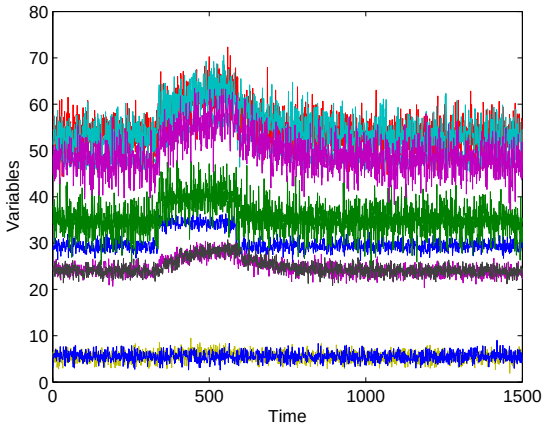
Figure 4.9: Academic case study: a) reconcilied Q_3 values with delays and b) reconcilied W1 values with delays.

By using the proposed combined methodology (DDR-TDI), a delay vector was obtained. The GA-based optimization method was run with the following parameters: 1000 individuals, 100 generations, 0.7 crossover probability, 0.0007 mutation probability and a maximum delay of 300 sampling times. This delay vector was used to adjust the measurements signals. Big differences between delayed and adjusted variables can be easily observed in Figure 4.10. Note that adjusted and not adjusted variables are not given in the same temporal moment. That is, each figure represents different time windows of 1500

samplings.



a)



b)

Figure 4.10: Academic case study: Process variables a) with delays and b) with adjusted delays.

The best correlation found for the adjusted signals shown in Figure 4.10b, can be quantitatively evaluated by calculating the variance distribution along the extracted eigenvectors, as it was done for the illustrative example in section 4.4.

By this way, variance explained by first singular value was approximately 55% when no adjustments were implemented. When variable delays were adjusted such variance increased up to 65% approximately. This fact can be graphically checked by revising the variance distribution throughout the eigenvectors depicted on Figure 4.11.

In order to quantitatively compare DDR performance before and after delays adjustment, a quadratic error function for each variable was proposed:

$$\mathbf{VE_i} = \sum_{t_k=1}^{m} (x_{i,t_k} - \widehat{x}_{i,t_k})^2 \tag{4.24}$$

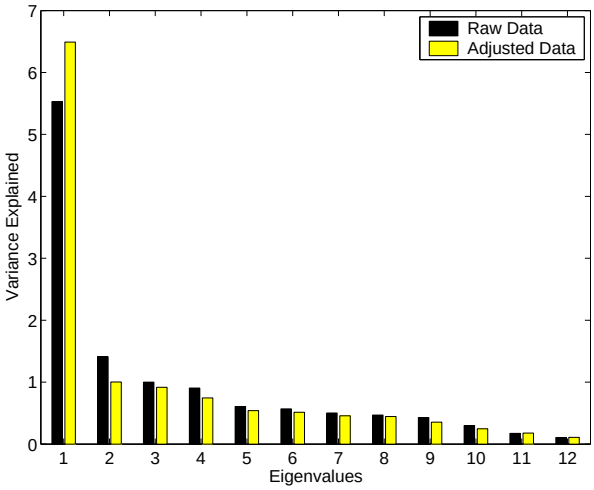


Figure 4.11: Academic case study: Variance explained in the eigenvectors of the correlation matrix.

$\mathbf{VE_i}$ represents the variable i estimation error throughout time (m are the considered sampling times), $\mathbf{x_{i,t_k}}$ is the actual true value of variable i at sampling time t_k and $\widehat{\mathbf{x}}_{\mathbf{i,t_k}}$ represents the variable i estimated value from DDR at the same sampling time. Global error (**GE**) can be evaluated considering all the studied variables (n):

$$\mathbf{GE} = \sum_{i=1}^{n} (VE_i) \tag{4.25}$$

Two additional optimization experiments were carried out to demonstrate the robustness of the methodology. Parameters of the optimization algorithm and the corresponding achieved delay vectors are summarized on Table 4.7. Note that the reference signal selected was the most delayed signal (Q_7) following the same criteria discussed on the illustrative case discussion.

Table 4.7: Academic case study: optimization parameters and obtained **DV**.

Experiments	Optimization parameters					Delay vectors (sampling time units)
	N_{ind}	N_G	P_c	P_m	d_{max}	[$Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, W1, W2, W3, W4$]
Adjustment 1 (AD1)	1000	100	0.7	0.0007	300	[270, 90, 70, 80, 70, 60, 0, 220, 270, 70, 70, 70]
Adjustment 2 (AD2)	250	100	0.7	0.0028	300	[270, 100, 80, 70, 70, 60, 0, 210, 270, 70, 60, 80]
Adjustment 3 (AD3)	1000	10	0.7	0.0007	300	[270, 90, 80, 50, 60, 100, 0, 270, 270, 90, 110, 90]

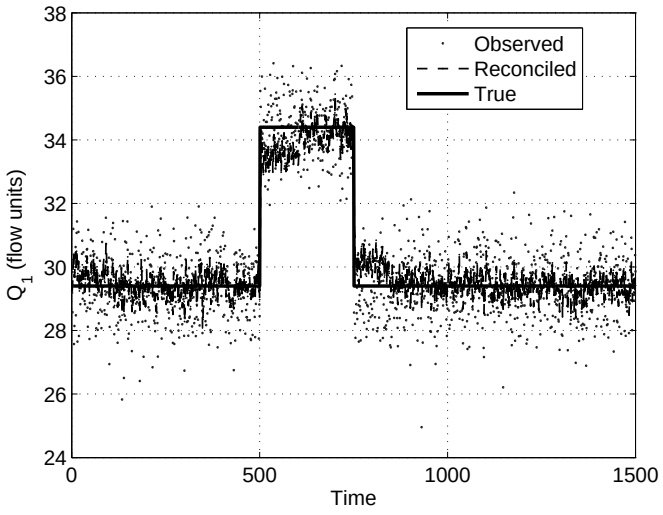
DDR results obtained without delay adjustments are compared with these obtained adjusting delays (Table 4.8) by using DVs from Table 4.7. These results show an important decrease of estimation errors when delay vectors are used. Furthermore, it should be noted that all three experiments highly increase DDR performance without appreciable

differences between them, even in the case that sub-optimal solutions are generated.

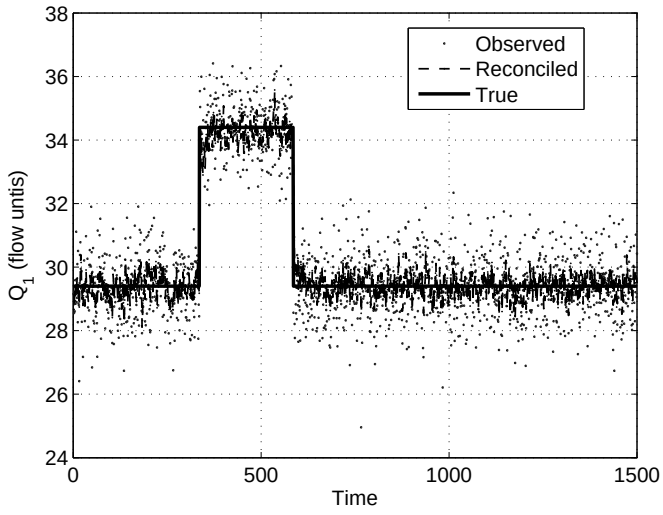
Table 4.8: Academic case study: DDR errors (VE_i and GE).

Variables	Without delays	With delays	With Adjusted Delays		
-	-	-	AD1	AD2	AD3
Q1	0.172	0.249	0.164	0.165	0.157
Q2	0.169	0.198	0.160	0.158	0.146
Q3	0.210	0.137	0.153	0.140	0.145
Q4	0.189	0.147	0.127	0.139	0.153
Q5	0.185	0.142	0.130	0.126	0.130
Q6	0.124	0.172	0.115	0.127	0.129
Q7	0.133	0.127	0.130	0.132	0.135
Q8	0.137	0.129	0.129	0.135	0.134
W1	3.268	8.395	3.722	3.780	3.562
W2	3.425	3.451	3.907	3.882	3.837
W3	3.053	3.389	3.219	2.975	3.406
W4	3.267	3.628	3.260	3.246	3.416
Total (GE)	14.332	20.164	15.216	15.007	15.352

This DDR performance improvement can be visually confirmed comparing Figures 4.12, 4.13, 4.14 and 4.15 that depict observed, true and reconciled values (with and without delays adjustment) of flow and mass hold-ups variables. They show the DDR results obtained with the best delay vector from Table 4.8 (AD2) and the DDR results obtained without removing the existing delays. DDR performance is considerably enhanced applying any of the obtained delay vectors. Note again that both charts in each figure are given in different times. They are different time windows of 1500 samplings each.

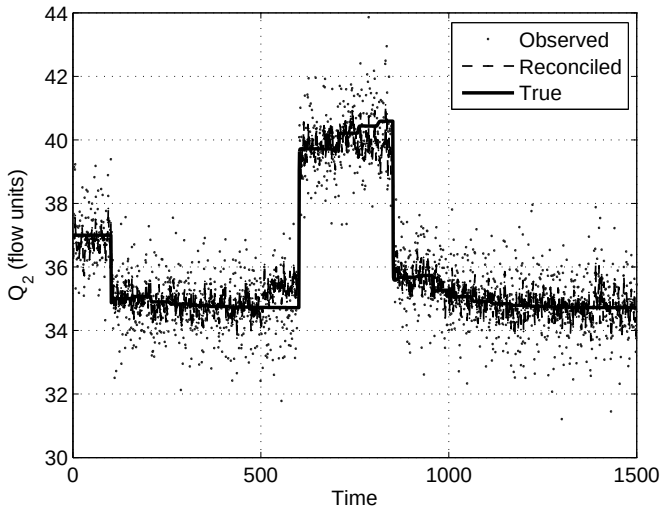


a)

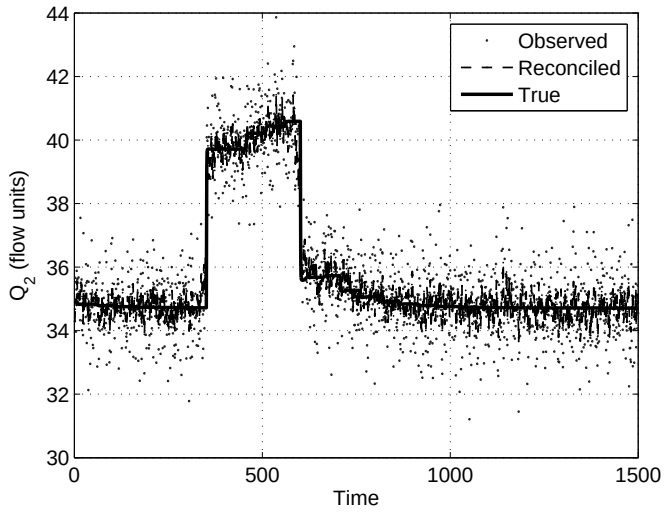


b)

Figure 4.12: Academic case study: Reconcilied Q1 values a) with delays and b) with adjusted delays.

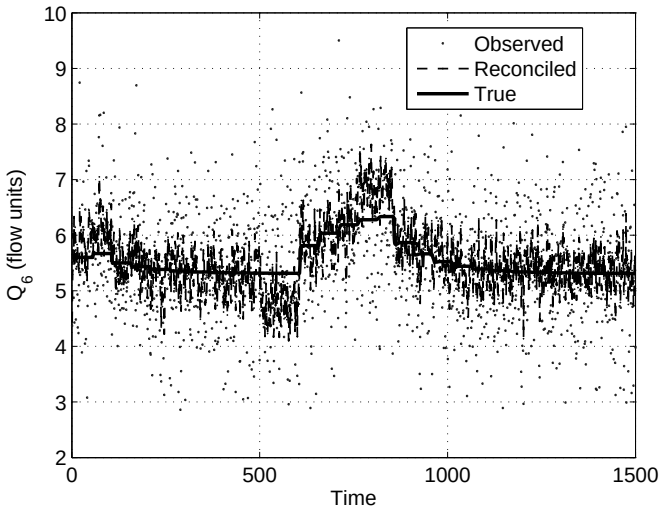


a)

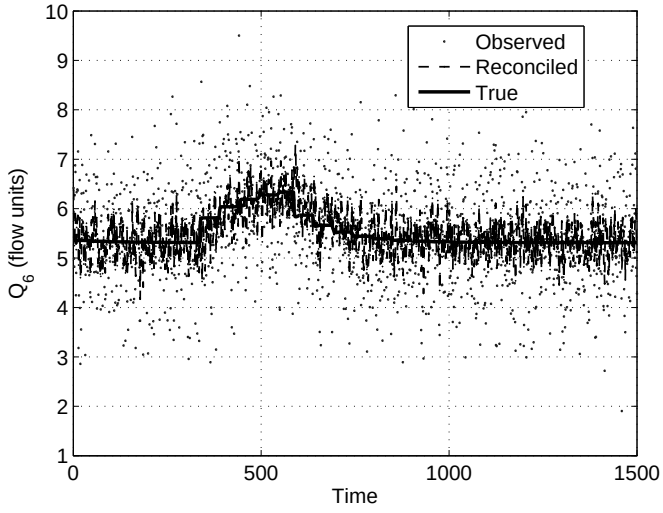


b)

Figure 4.13: Academic case study: Reconcilied Q2 values a) with delays and b) with adjusted delays.

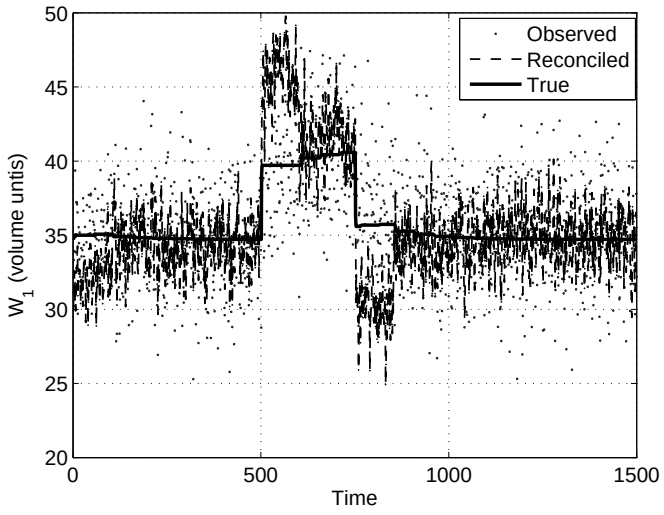


a)

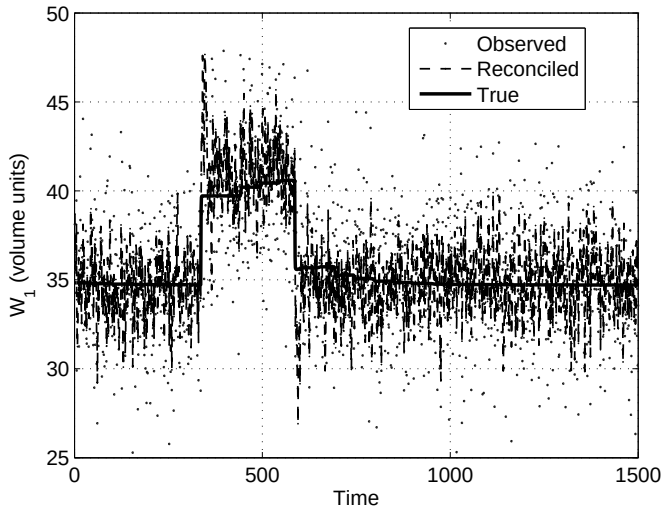


b)

Figure 4.14: Academic case study: Reconcilied Q6 values a) with delays and b) with adjusted delays.



a)



b)

Figure 4.15: Academic case study: Reconcilied W1 values a) with delays and b) with adjusted delays.

4.6.2 Industrial case study

The proposed DDR methodology was also tested on the TE benchmark (Downs and Vogel, 1993) in order to check the actual scope and robustness of the technique in industrial problems.

The case involves the production of two products (G and H) from four reactants (A,C,E and E). There are 41 measured process variables and 12 manipulated variables. The process (Figure 4.16) is composed of five main unit operations: an exothermic 2-phase reactor, a product condenser, a flash separator, an stripper and a recycle compressor. Such process gathers the required non-linearities and recycles to definitely check the robustness of the proposed DDR methodology.

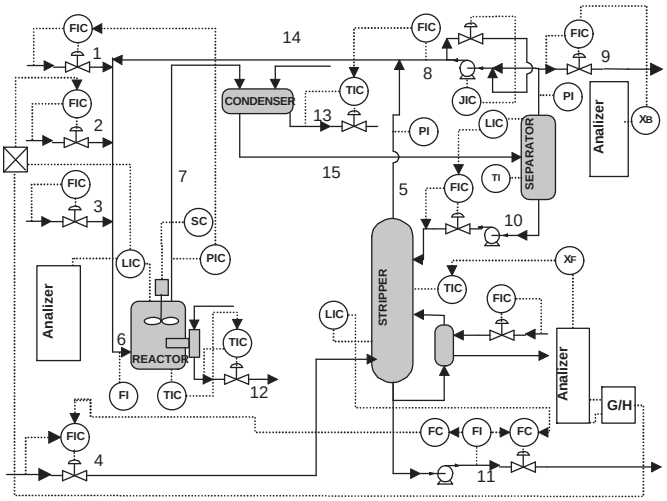


Figure 4.16: Tennessee Eastman process flowsheet.

The case study was simulated in Matlab using the original Fortran code given by Downs and Vogel (1993). The implemented control strategy is that one proposed by McAvoy and Ye (1994). Besides the highly coupled variables and the non-linearities existing, different variables were deliberately delayed to confront a more challenging case study. The EKF was applied to deal better with such non-linear complex problem.

The state space model used to apply the methodology is based in the TE model proposed by Ricker and Lee (1995a). It is a dynamic model which covers the material balance of the process, consisting of 26 states and 15 parameters. The model satisfactorily demonstrated its good performance by giving accurate estimation of the 26 considered states variables and the 23 involved output variables. For deeper details on the applied dynamic model, interested readers should look over articles: Ricker and Lee (1995a) and Ricker and Lee (1995b). In order to apply the EKF, the model was linearized as it is described by Ricker and Lee (1995b).

Data were collected from a 35 h process simulation run by doing samplings every 2 minutes. 23 output variables considered correspond to different XMEAS given in Downs and

Vogel (1993). Table 4.9 shows the relationship among 23 model output variables and original process variables and also depicts forced delays applied to each one. All forced delays are applied to composition measurements and are equals 36 minutes.

Table 4.9: TE benchmark: Elements of the output vector and associated delays.

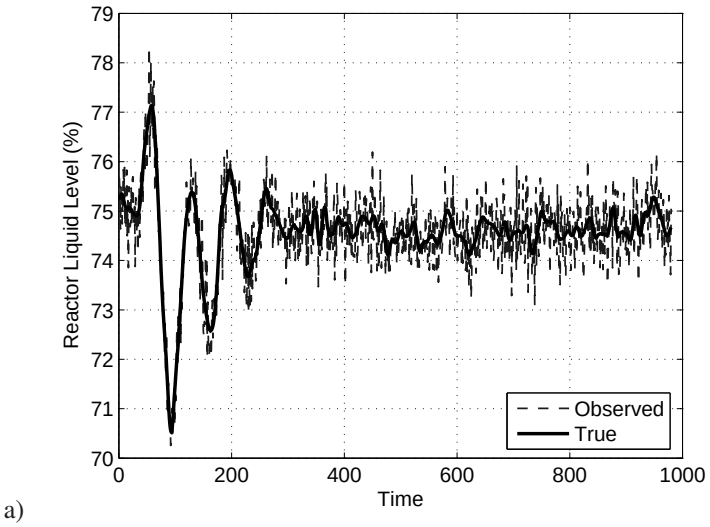
V. Number	Description	Downs Vogel index	Delay
1	Reactor pressure	7	0
2	Reactor liquid level	8	0
3	Separator pressure	13	0
4	Separator liquid level	12	0
5	Stripper bottoms level	15	0
6	Stripper pressure	16	0
7	Reactor feed flow rate	6	0
8	A in reactor feed	23	36
9	B in reactor feed	24	36
10	C in reactor feed	25	36
11	D in reactor feed	26	36
12	E in reactor feed	27	36
13	F in reactor feed	28	36
14	A in purge	29	36
15	B in purge	30	36
16	C in purge	31	36
17	D in purge	32	36
18	E in purge	33	36
19	F in purge	34	36
20	G in purge	35	36
21	H in purge	36	36
22	G in product	40	36
23	H in product	41	36

As TE process simulator is ruled by unknown basic principles, no previous knowledge is available to determine actual process variable values. Thus, comparison between DDR estimates and true variables values becomes impossible and approximation of true values is required. With that purpose a wavelet filtering was applied on the process signals to extract the essential features of process measurements. These filter signals were considered as an accurate approximation of the true variable values. In order to state that fact visual inspections of wavelets filtering results were carried out.

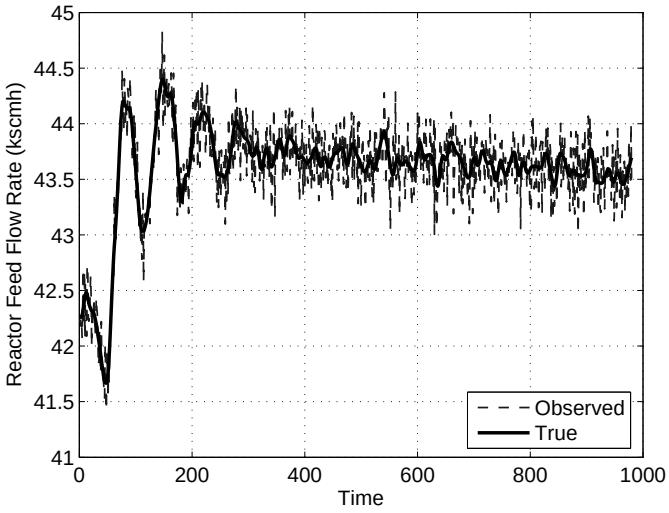
The "daubechies 6" wavelet and a 3 level decomposition were applied in a "soft thresholding" procedure for de-noising process signals (Tona et al., 2005). Examples of the filtering results are given in Figure 4.17, where process measurements are represented by dots (observed) and wavelets filtering signals are given by bold lines (true).

Results obtained under several original disturbances (see Table 4.10), for example IDV(1) (Figure 4.17), show how wavelet filtering signal produces good approximations to the expected true variable values. They show a very good de-noising capabilities, which make us to adopt them as reliable true process variables values.

With the goal to make evident DDR methodology effects, several disturbances were introduced into the process under the conditions described so far. Such disturbances correspond to the first five originally given by the TE original paper (Downs and Vogel, 1993).



a)



b)

Figure 4.17: TE case: Wavelets filtering results under IDV(1). Bold line represents the wavelet filtered signal and the dotted line represents the variable measurement. a) Reactor liquid level, b) Reactor feed flow rate.

The description and correspondence with those disturbances from the original paper are shown in Table 4.10, next to the results obtained with the tested methodology. Disturbances were introduced at the very early moments of simulation and were maintained until the end of the experiment.

The GA was applied to any of the previous five mentioned cases, with the same parameters definition: 100 individuals, 50 generations, 0.7 crossover probability, 0.0007 mutation probability and a maximum delay of 40 minutes. It was required approximately 10 minutes of CPU to obtain the final solution, which was considered a reasonable time for the

Table 4.10: DDR results in the TE case under 5 different disturbances.

Disturbance	Description	GE with delays	GE with adjusted delays	Calc. time (min)
IDV(1)	A/C Feed Ratio step	5.2748	5.0106	10.97
IDV(2)	B Composition step	2.2920	2.2825	10.91
IDV(3)	D Feed Temp. step	1.0731	1.0564	11.63
IDV(4)	React. Cool. Water Inlet Temp. step	1.0898	1.0652	10.89
IDV(5)	Cond. Cool. Water Inlet Temp. step	1.0802	1.0555	10.95

on-line application of the presented technique.

Firstly, a sudden change in A/C feed ratio composition (IDV(1) in Downs and Vogel (1993)) was tested causing significant changes from normal variables behavior. Under the disturbance, different reconciliation results were obtained in case the delay adjustment procedure was working or not. Then, the rest of considered disturbances were tested in a similar way. Table 4.10 shows the tested disturbances and the global estimation error before and after the delays adjustment. The CPU time required to obtain the **DV** in each case is also given in the last column of the table. The short calculation times required allow to generate a new **DV** each 10 minutes approximately, which makes DDR flexible and quite adaptable to the process dynamics.

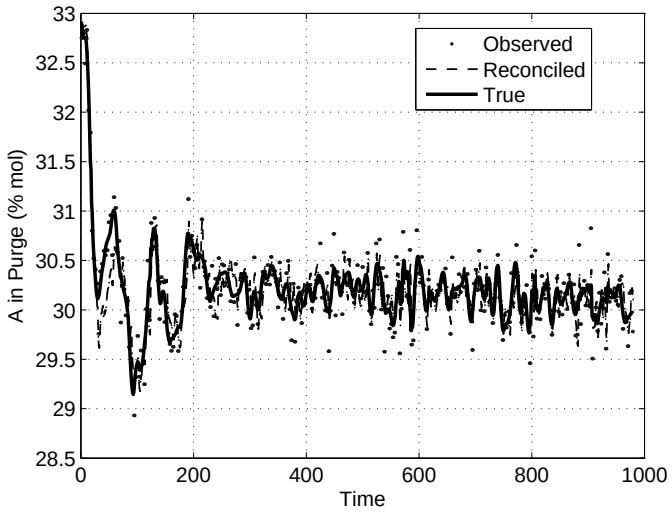
Significant and general reductions in GE from DDR were obtained for all the tested cases. The proposed TDI approach provided good and fast TDI that clearly improved the DDR performance. Reduction of estimation errors can be also checked graphically looking at Figures 4.18 and 4.19.

These figures show reconciliation of variables 14 and 7 from Table 4.9 under disturbance IDV(1) arising at the early moments of simulation. In both figures process measurements and true and reconciled variables values are depicted together. Each sampling time correspond to 2 simulation minutes. Figure 4.18 shows how the model estimates better variable 14 than variable 7 given at Figure 4.19. Better estimations are reached after delays adjustment reconciling variable 14 (Figure 4.18), thus avoiding some measurement noise. Note that, although estimations of variable 7 (Figure 4.19) are biased after the occurrence of the disturbance, also better reconciled values can be appreciated at the early simulation moments by adjusting the existing process variable delays.

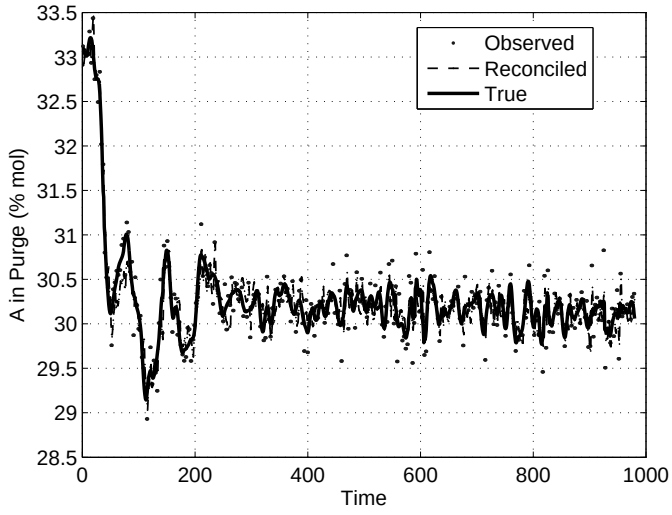
4.7 Conclusions

Delays between process event causes and effects are present in most of dynamic processes. Thus, DDR performance can be highly improved by adjusting the delays between these process variables. One important advantage of the presented TDI method is that no a priori process knowledge is needed to define the number and location of time delays as well as the characteristics of the process variables, i.e. if they correspond to input or output variables. Also, the effectiveness of the proposed TDI procedure makes the approach suitable when delays adjustment is regularly performed to maintain the on-line plant performance.

Firstly, a complex dynamic case study was successfully addressed, highlighting the performance of the proposed TDI approach. Besides that, the procedure was also tested in an industrial case study (TE case) demonstrating promising robustness and applicability



a)

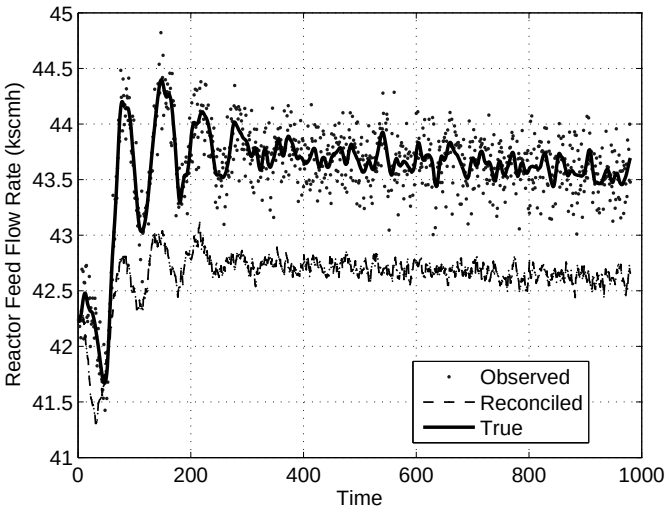


b)

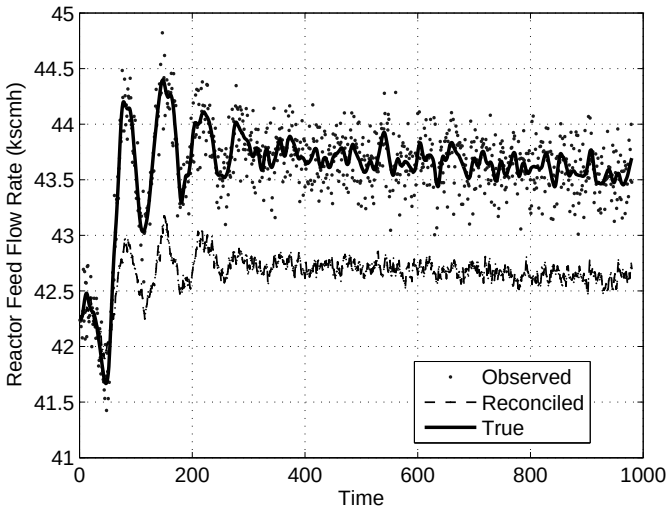
Figure 4.18: TE case: Reconcilied variable 14 from Table 4.9 under IDV(1), a) with delays and b) with adjusted delays.

in highly non-linear and coupled variables processes.

Quantitative and qualitative evidences were included to show the significant improvement obtained by the use of the proposed GA-based optimization method for TDI. An extension of this approach to time-varying delays has been also sketched.



a)



b)

Figure 4.19: TE case: Reconcilied variable 7 from Table 4.9 under IDV(1), a) with delays and b) with adjusted delays.

Nomenclature

d_{max}	Maximum delay
d_i	Delay element
DV	Delay vector
f	Set of differential equation constraints
g	Set of algebraic equality constraints
GE	Global estimation error
h	Set of inequality constraints
m	Number of samplings
$maxSV$	Maximum singular value
n	Number of process variables
N_{ind}	Number of individuals in each generation
N_G	Number of generations
OF	Objective function
P_c	Crossover probability
P_m	Mutation probability
R	Correlation matrix
$\widehat{u}$	Estimated unmeasured variables vector
VE	Variable estimation error
$\mathbf{x}$	State variables vector
$\widehat{\mathbf{x}}$	Estimated state variables vector
$\mathbf{y}$	Measurement variables vector
$\widehat{\mathbf{y}}$	Estimated measurement variables vector
$\mathbf{y}^*$	Unknown true process variables values
$\mathbf{Y}$	Process data matrix
$\overline{\mathbf{Y}}$	Normalized process data matrix
$\mathbf{Y_{DA}}$	Delay adjusted data matrix

Subscripts

i	Process variables
k	Sensors
t_k	Sampling time
z	Individual on the key diagnosis variables search

Operators

σ	Variable standard deviation

Greek Letters

$\widehat{\delta}$	Estimated unmeasured variables vector
γ	Function for predicting current state in EKF
λ	Function for predicting the current measurements in EKF
ε	Random measurement errors
Σ	Variance matrix
$\widehat{\theta}$	Vector of adjusted model parameters
Φ	Fitness function

Acronyms

AC	Acquisition Cost
CPU	Central Processing Unit
DC	De-installation Cost
DR	Data Reconciliation
DDR	Dynamic Data Reconciliation
EKF	Extended Kalman Filter
FDS	Fault Diagnosis System
FM	Flow Meter
IC	Installation Cost
GA	Genetic Algorithm
KF	Kalman Filter
LM	Level Meter
MINLP	Mixed Integer Non-Linear Programming
PCA	Principal Component Analysis
PRD	Process Related Delay
PSE	Process System Engineering
SN	Sensor Network
SR	Sensor Reliability
SRD	Sensor Related Delay
TDI	Time Delay Identification
TE	Tennessee Eastman process

CHAPTER 5

Monitoring and Diagnosis Basis

Once accuracy, reliability and consistency of process measurements is assured, this amount of quantitative information must be transformed into useful knowledge for process operators. Thus, monitoring and diagnosis systems become crucial for extracting knowledge about current plant state from process data. This chapter focuses on describing the basis of monitoring and diagnosis.

5.1 Introduction

AEM is in charge of detecting and diagnosing any possible process deviation (abnormal events) that could lead to dangerous consequences on safety or economy of the plant. Then, an AEM system must transform all the plant available information into useful knowledge to take the most appropriated corrective decisions. Although proper data acquisition and rectification systems as well as effective decision-making supporting tools are important for achieving a good overall performance, the core of AEM consists of the monitoring and fault diagnosis systems. In that sense, an early detection and diagnosis when the plant is still under control is always critical in avoiding harmful consequences.

As it was mentioned in section 3, this thesis focuses on the data driven fault diagnosis systems due to the difficulties encountered on chemical processes modeling and the usual availability of wide process data bases. As remarked in section 2 there are several clear shortcomings/opportunities to improve the current process fault diagnosis. These are remarked again below:

- Lack of standards to compare the diagnosis systems
- Poor use of machine learning algorithms
- Ignoring integration possibilities

This section goes deeper in solving the first point of the list by strictly formulating the process diagnosis as a classification problem. Also, it introduces and establishes the basis

for future diagnosis results comparisons.

Regarding the other mentioned limitations, a machine learning based and a rules based complementary FDSs are proposed to take advantage of the machine learning field as well as the synergy (Venkatasubramanian et al., 2003a) created by FDSs integration. Details on both systems are given in chapters 6, 7 and 8.

5.2 Problem formulation

In order to formalize the fault diagnosis problem, it may be seen as a general classification problem from which the structure and results evaluation can be better arranged. Following that assumption, the detection problem is just a binary classification problem (BC), in which two excluding classes are considered, the normal and the abnormal plant behavior. Hence, the global fault diagnosis problem is a multi-class classification (MC) problem, since many classes are involved.

Furthermore, fault diagnosis problem can be considered as monoLabel (mL, when only a fault can happen at a time) or MultiLabel (ML, when more than a fault may happen at a time). Adopting a mL or a ML view depends on whether or not the assignment of a class (faults) excludes the assignment of other classes. In this sense, the fault diagnosis regarded as MultiLabel shares the characteristics of the Text Classification problem in Computational Linguistics (Joachims, 2002; Schapire and Singer, 2000). However, despite being a clear ML problem (simultaneous faults can always occur), the general fault diagnosis of chemical processes has frequently been addressed under an implicit mL view by constraining the global problem to the diagnosis of single faults.

In the recent literature, pattern recognition techniques have continued with that implicit view using the mL approach for solving the general fault diagnosis problem (Raich and Çinar, 1996, 1997). Nevertheless, the classification of simultaneous faults requires either the simultaneous class assignment, which is the natural ML view, either the use of more classes representing the occurrence of simultaneous faults (class A+B instead of classes A and B), which is the solution adopted when trying to preserve the mL view.

Besides its conceptual and methodological suitability to the problem, the ML approach also offers practical benefits. The basic advantages of the ML approach presented stem from the ability of simply training single classes, being later able to classify the individual classes that compose the current simultaneous faults (combination of single classes). In that way, no artificial classes resulting of single faults combination must be built to address simultaneous faults, thus saving computational cost and increasing learning performance by just training single classes.

To clarify that concept the ML and mL views can be confronted when facing the global diagnosis problem. For instance, for a problem with 20 (n) pre-defined faults, a mL approach should consider the combinatorial number of possible simultaneous faults. Assuming the least possible simultaneity (p = 2, which means only 2 faults arising simultaneously) up to 190 artificial fault classes should be contemplated (see Eq. 5.1).

$$P_n^p = \frac{n!}{(n-p)! \cdot p!} = \frac{20 \cdot 19}{2} = 190 \tag{5.1}$$

Since the identification capabilities of classification techniques strongly decreases with the number of classes considered, using such an exponentially growing number of artificial classes makes this approach computationally un-affordable as well as inefficient for industrial problems (see discussion about Figure 6.7 in chapter 6).

5.3 Evaluation indices

In order to evaluate the performance of a FDS, this work rigorously formulates the diagnosis classification problem and proposes general evaluation indices.

Process data can be characterized by a matrix X of values x_{vs} for each process variable v and for each sampling s, as well as by a given set of faults $f = 1, 2, \ldots, F$ that may be happening (or not) and that may be eventually diagnosed (or not). Hence, a binary matrix H may be used to indicate which of these faults (in columns) is happening at each sampling time (in rows). Accordingly, another binary matrix D may be used to indicate the diagnosis made at each sampling time. Both matrices are shown next:

$$H = \begin{pmatrix} h_{11} & h_{12} & \ldots & h_{1F} \\ h_{21} & h_{22} & \ldots & h_{2F} \\ \vdots & \ddots & \ddots & \vdots \\ h_{s1} & h_{s2} & \ldots & h_{SF} \end{pmatrix}, h_{sf} \quad \epsilon \quad \{0,1\}, \quad \forall s, f : 1 \leq s \leq S, 1 \leq f \leq F \tag{5.2}$$

$$D = \begin{pmatrix} d_{11} & d_{12} & \ldots & d_{1F} \\ d_{21} & d_{22} & \ldots & d_{2F} \\ \vdots & \ddots & \ddots & \vdots \\ d_{s1} & d_{s2} & \ldots & d_{SF} \end{pmatrix}, d_{sf} \quad \epsilon \quad \{0,1\}, \quad \forall s, f : 1 \leq s \leq S, 1 \leq f \leq F \tag{5.3}$$

Either a fault f is happening ($h_{sf} = 1$) or not ($h_{sf} = 0$), a good diagnostic is given by $d_{sf} = h_{sf}$. Therefore, the ML classification problem seeks for the matching of matrix D to matrix H when any distribution of binary values is allowed, including null rows for the normal case (no faults) and rows including several non-null values (simultaneous faults).

For determining the matching degree of these matrices, thus the actual FDS performance, two essential measurements are given in the machine learning literature, Precision and Recall. Precision for fault f (Eq. 5.4) can be defined as the conditioned probability of happening fault f, conditioned to fault f has been diagnosed

$$\text{Prec}(f) = \Pr[h_{ij} = d_{ij} \,|\, d_{if} = 1 : 1 \leq i \leq S, j = f\ ] \tag{5.4}$$

and Recall for fault f (Eq. 5.5) can be defined as the conditioned probability of the FDS of predicting fault f conditioned to the sampling being fault f.

$$\text{Rec}(f) = \Pr[h_{ij} = d_{ij} \,|\, h_{if} = 1 : 1 \leq i \leq S, j = f\ ] \tag{5.5}$$

The $F1$ index (widely used in the Machine Learning area) is a measure that combines both Precision and Recall. It is used in this thesis as a measure for evaluating the general performance of any FDS. It is evaluated by the next equation:

$$\mathrm{F1} = \frac{2 \times \mathrm{Prec} \times \mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}} \tag{5.6}$$

Accuracy, Error and Global measurements are additional and complementary indices to globally evaluate the system classification general performance. Accuracy represents the percentage of right assignments, not only the correctly diagnosed but also the correctly not happening and not diagnosed samplings. Error is the percentage of wrong assignments and is the Accuracy supplementary measure. Next, they are formally defined

$$Acc(f) = \Pr[d_{ij} = h_{ij} \,|\, i : 1 \leq i \leq S, j = f\ ] \tag{5.7}$$

$$Error(f) = \Pr[d_{ij} \neq h_{ij} \,|\, i : 1 \leq i \leq S\ \ ,j = f] \tag{5.8}$$

Global index shows how many samplings have been completely well diagnosed through all the existing classes, hence giving a very valuable measurement about the system global diagnosis correctness. High values in that measurement are difficult to achieve (as they require a perfect diagnosis for each sampling) and not only indicate a good performance diagnosing the isolated classes but also assure good MultiLabel characteristics that are crucial facing simultaneous faults (a complete well diagnosed system also implies no misdiagnosis, which is essential in the ML classification approach). Following the definition procedure it can be expressed as it is shown in Eq. 5.9:

$$Global = \Pr[d_{ij} = h_{ij} \,|\, i : 1 \leq i \leq S, \forall\ j\ ] \tag{5.9}$$

Three last measurements (Eq. 5.7 to 5.9) only have sense when considering the ML problem because of its particular samplings assignments nature. All mentioned performance measurements can be calculated considering the FDS responses or votes after testing each of the trained faults. The confusion matrix (Table 5.1) gathers such information for each fault, showing the occurred fault in columns and the diagnosed fault in rows.

Table 5.1: Contingency or Confusion Matrix.

		Happening fault (true class)	
		f	$\neg f$
Diagnosed fault (predicted class)	f	a	b
	$\neg f$	c	d

where a are samplings happened and diagnosed, b are samplings diagnosed but not happened, c are samplings happened and not diagnosed, and d are samplings not happened and not diagnosed.

Hence, the following equations are proposed for directly evaluating the classification performance indices from the confusion matrix (Manning and Schütze, 1999):

$$\text{Prec}(f) = \frac{a}{a+b} \quad \text{Rec}(f) = \frac{a}{a+c} \tag{5.10}$$

$$Acc(f) = \frac{a+d}{a+b+c+d} \quad Error(f) = \frac{b+c}{a+b+c+d} = 1 - \text{Acc}(f) \tag{5.11}$$

As a clarifying example, a fault happening matrix (H) and the FDS corresponding diagnosing matrix (D) are next presented. 3 faults and 12 samplings (represented in columns and rows respectively) are assumed.

$$H = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad D = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

In H matrix, first two samplings correspond to the normal state; second, third and fourth pairs of samplings correspond to faults 1, 2 and 3 whereas the last four samplings correspond to a simultaneous fault composed of faults 2 and 3. Matrix D represents the FDS results facing samplings on matrix H. For example, the first two rows in matrix D show that both samplings were classified as fault number 3, as well as the samplings in the last four rows. The performance indices can be calculated by comparing the matching of matrix D to matrix H by using the details of the individual and global contingency matrices. Next, these matrices for normal state, faults 1, 2, 3 and the general results are depicted. Note that class 0 means normal state.

CM	0	¬0
0	0	0
¬0	2	10

CM	1	¬1
1	2	0
¬1	0	10

CM	2	¬2
2	2	2
¬2	4	4

CM	3	¬3
3	4	2
¬3	2	4

ΣCM	yes	no
yes	8	4
no	6	18

Therefore, performance indices can be easily evaluated by replacing the values from these matrices (a, b, c and d in Table 5.1) in equations 5.6, 5.10 and 5.11. The results (Table 5.2) offer complete and detailed information about the FDS performance. Precision gives information about the diagnosed samplings, Recall gathers the information about the happened faults and $F1$ index gives a single normalized measurement for the evaluation of

the general FDS performance. Note that even though the Normal state was not trained it was later tested and its performance was also evaluated in the same way. Accuracy and Error are supplementary measurements about the correct diagnosed and not diagnosed samplings whereas the Global measurement is lower just because it focuses on the totally well-diagnosed samplings throughout all the existing classes, 4 over the original 12 (check H and D matrices).

Table 5.2: Indices evaluation for the proposed example. General does not include Normal class.

Index*100	Normal	Fault 1	Fault 2	Fault 3	General
Prec.	0/(0+0) = -	2/(2+0) = 100	2/(2+2) = 50.0	4/(4+2) = 66.7	8/12 = 66.7
Recall	0/(0+2) = 0	2/(2+0) = 100	2/(2+4) = 33.3	4/(4+2) = 66.7	8/14= 57.1
F1	-	100	40	66.7	61.5
Acc.	10/12=83.3	12/12 =100	6/12 = 50.0	8/12 = 66.7	26/36= 72.2
Error	17.7	0	50	33.3	27.8
Global			4/12 = 33.3		

In case a mL view is adopted, the problem formulation can be easily simplified by constraining the simultaneity of faults. Now, the sum of elements by row is equal 1 for H and D matrices.

$$\sum_{f=1}^{F} h_{sf} = 1, \forall s : 1 \leq s \leq S \tag{5.12}$$

$$\sum_{f=1}^{F} d_{sf} = 1, \forall s : 1 \leq s \leq S \tag{5.13}$$

Thus, H and D matrices can be reduced by using new integer variables h_s and d_s which define the fault happening and diagnosed at each sampling s, instead of the binary variables h_{sf} and d_{sf} used in the general formulation:

$$H = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_S \end{pmatrix}, h_s \in \{0, 1, \ldots, F\}, \forall s : 1 \leq s \leq S \tag{5.14}$$

$$D = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_S \end{pmatrix}, d_s \in \{0, 1, \ldots, F\}, \forall s : 1 \leq s \leq S \tag{5.15}$$

Under that view, Precision and Recall are the only quantitative required indices to address the classification performance, as just one assignment is allowed for each sampling. Both measures can be formally defined compressing the nomenclature as done in equations 5.16 and 5.17. Their practical values are attained by using equations 5.10.

$$\text{Prec}(f) = \Pr[h_i = f \,|d_i = f : 1 \leq i \leq S\ ] \tag{5.16}$$

$$\mathrm{Rec}(f) = \Pr[d_i = f \,|h_i = f : 1 \leq i \leq S\ ] \tag{5.17}$$

As a clarifying example, a fault occurrence matrix (H) and the FDS corresponding matrix response (D) are next presented and the mentioned measures are calculated.

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad D = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

For the mL case H and D may be simplified into:

$$H = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \end{pmatrix}^T \qquad D = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 3 \end{pmatrix}$$

The confusion matrices for this example are:

CM	1	¬1
1	3	1
¬1	0	5

CM	2	¬2
2	3	1
¬2	0	5

CM	3	¬3
3	1	0
¬3	2	6

Σ CM	yes	no
yes	7	2
no	2	16

The diagnosis performace results for that mL example are summarized in Table 5.3,

Table 5.3: Indices evaluation for the proposed mL example.

Index(%)	**Fault 1**	**Fault 2**	**Fault 3**	**mA**	**MA**
Precision	3/4 = 75.0	3/4 = 75.0	1/1 = 100	77.8	83.3
Recall	3/3 = 100	3/3 = 100	1/3= 33.3	77.8	77.8

where mA and MA are the micro and macro average values respectively that provide average FDS performance measurements under the mL approach.

Macro-average refers to the simple average for all the considered faults. It gives a sense of the quality of classification across all faults by means of a global evaluation measure. Another option is micro-averaging which is calculated summing the scores (a, b, c, and d) for all faults. This measure gives equal weight to each sampling whereas macro-averaging weights equal to each fault.

By this way, Recall and Precision have equal micro-averages but not necessarily equal macro-averages. Both measurements can be practically evaluated using the confusion matrices by equations 5.18 and 5.19.

$$\text{mAPrec} = \text{mARec} = \frac{\sum_{f=1}^{F} a_f}{\sum_{f=1}^{F} a_f + \sum_{f=1}^{F} b_f} = \frac{\sum_{f=1}^{F} a_f}{\sum_{f=1}^{F} a_f + \sum_{f=1}^{F} c_f} \tag{5.18}$$

$$\text{MA}(X) = \frac{\sum_{f=1}^{F} x_f}{F} \tag{5.19}$$

where X is overall system Recall or Precision and x_f represents Recall and Precisions for fault f. F is the total number of considered faults.

Nomenclature

a	Samplings happened and diagnosed
b	Samplings diagnosed but not happened
c	Sampling happened and not diagnosed
d	Samplings not happened and not diagnosed
Acc	Accuracy index
d_{sf}	Individual diagnosis for a time s and a fault f
D	Diagnosis matrix
h_{sf}	Individual occurrence for a time s and a fault f
H	Happening matrix
$Prec$	Precision index
Rec	Recall index
x	Process variable
X	Data matrix

Subscripts

f	Fault
s	Sampling
v	Variable

Acronyms

AEM	Abnormal Events Management
CM	Confusion or Contingency Matrix
mA	Micro-Average
mL	Mono-Label
MA	Macro-Average
ML	Multi-Label

CHAPTER 6

A Machine Learning based FDS

In this section, fault diagnosis is approached from the machine learning point of view. A recent developed algorithm, Support Vector Machines (SVM), widely used in other technical areas, was chosen to check their capabilities facing chemical process data. In that sense, this study tries to accomplish with one of the stressed opportunities on fault diagnosis (section 2.3), the use of novel machine learning algorithms.

Within this chapter not just SVM classification capabilities are tested but also our view to face the general diagnosis approach is put forward. First, SVM diagnosis abilities were tested and compared against other previous applied diagnosis techniques on the same basis. This comparative attempt led to uncover some basic strategic deficiencies at the time of facing a general diagnosis problem. To solve these limitations a ML approach considering SVM was adopted to illustrate the formal way to address the problem.

6.1 Introduction

Methods of classification and artificial intelligence areas have become some of the most extended fault diagnosis systems in the process industry diagnosis literature. They are based on mapping input data from process plant to different process states. The classification methodologies have succeeded because of their implementation simplicity, so they do not require of process operators experience or process first principles knowledge. Among these classification methodologies, data-based rules systems and machine learning are the most commonly applied methodologies. However, machine learning has not been properly exploited. Only neural networks has been widely applied as a pattern learning technique. This huge knowledge area developed in parallel to chemical fault diagnosis during last years, does not only consider classical neural networks, but more recent methodologies that have succeeded in other technical areas, obtaining promising results. Thus, considering the fault diagnosis issue as a classification problem would allow taking advantage of the benefits gained by the learning community throughout last years. Some remarkable algorithms can be outlined among the most widely tested machine learning techniques:

- Naive Bayes (Duda and Hart, 1973) which is the simplest representative of probabilistic learning methods.
- Decision list, applied by Yarowsky (1995).
- AdaBoost. Abney et al. (1999) gave a generalized version of that technique.
- Support Vector Machines (SVM) introduced by Boser et al. (1992). It has been slightly tested in fault diagnosis of chemical process.

This chapter focuses on the application of Support Vector Machines, a recently developed algorithm that has proven to be very powerful in many pattern and image recognition, text categorization and different Non Linear Programming (NLP) problems (Escudero (2005)).

6.2 Support Vector Machines

Support Vector Machines (SVM) were introduced by Boser et al. (1992), but they have only been gaining popularity in the learning community from 1996. Nowadays, SVM have been successfully applied to a number of problems related to pattern recognition in bio-informatics and image recognition. Regarding text processing, SVM have obtained the best results to date in text categorization (Joachims, 1998) and they are being used in more and more Natural Language Processing (NLP) related problems, e.g., chunking (Kudo and Matsumoto, 2001), parsing (Collins, 2002), or Word Sense Disambiguation (Murata et al., 2001; Escudero et al., 2004).

SVM are based on the Structural Risk Minimization principle from the Statistical Learning Theory (Vapnik, 1998) and, in their basic form, they learn a linear hyperplane that separates a set of positive examples from a set of negative examples with maximum margin (the margin is defined by the distance of the hyperplane to the nearest of the positive and negative examples). This learning bias has proved to have good properties in terms of generalization bounds for the induced classifiers. The linear classifier is defined by two elements: a weight vector w (with one component for each feature), and a bias b which stands for the distance of the hyperplane to the origin (Figure 6.1). The classification rule assigns +1, to a new example x, when $f(x) =< x, w > +b \geq 0$, and -1 otherwise. The positive and negative examples closest to the hyperplane (w,b) are called support vectors (Figure 6.2).

Learning the maximal margin hyperplane (w,b) can be simply stated as a convex quadratic optimization problem with a unique solution, consisting of (primal formulation):

$$Minimize \quad \|w\| \tag{6.1}$$

subject to the constraint (one for each training example):

$$y_i \, (< w, xi > +b) \geq 1.$$

In case the training examples are not linearly separable or, simply, it is not desirable to obtain a perfect hyperplane, it is preferable to allow some errors in the training set so as to maintain a "better" solution hyperplane (see Figure 6.3). This is achieved by a variant of the optimization problem, referred to as soft margin, in which the contribution to the

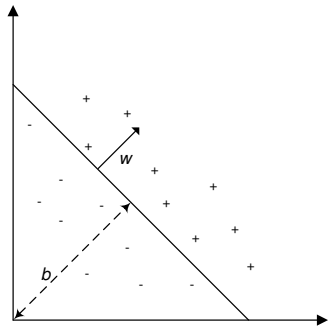


Figure 6.1: Hyperplane definition.

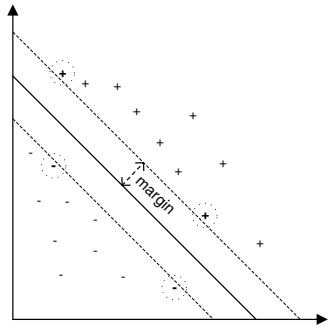


Figure 6.2: Support vectors and margin.

objective function of the margin maximization and the training errors can be balanced through the use of a parameter called C.

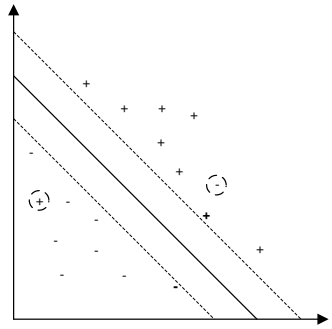


Figure 6.3: Non linearly separable data. Soft margin optimization.

Despite the linearity of the basic algorithm, SVM can be transformed into a dual form (in which the training examples only appear inside dot product operations), allowing the use of kernel functions to produce non-linear classifiers. The kernel functions make SVM

to work efficiently and implicitly in very high dimensional feature spaces, where new features can be expressed as combinations of many basic features. See Christianini and Shawe-Taylor (2000) for details about soft margin and non-linear SVM.

By means of the "one vs. all" binarization (check subsection 6.4.1.1), SVM which is a binary classifier (+1,-1) is also able to deal properly with multiclass problems, implementing so many classifiers as process states considered in the diagnosis problem.

SVMlight software (Boser et al., 1992), an open source implementation of the SVM original mechanism (Joachims, 1999) was used in the thesis experiments. That implementation adopts some practical simplifications in order to make applicable the theoretical SVM formulation (available at http://svmlight.joachims.org).

6.3 Particular diagnosis problem: single faults

In order to estimate the SVM diagnosis capabilities, a first study on the constrained single fault diagnosis problem was done. References were searched for doing formal comparison and quantification of the SVM diagnosis performance regarding other approaches. In that way, the TE benchmark was used again because its complexity and popularity in the fault diagnosis community.

6.3.1 Evaluation of reported works

A clear lack of quantitative results can be appreciated from a general analysis of chemical process fault diagnosis literature (subsection 2.3.4). This means an important difficulty at the time to compare new and former diagnostic techniques. Different works (Vedam and Venkatasubramanian, 1999; Dash et al., 2003; Leung and Romagnoli, 2002) have addressed different case studies and those addressing an standard benchmark deal partially with the problem, focusing only in some of the complete set of faults or using particular performance measurements (Kulkarni et al., 2005; Gertler et al., 1999; Wachs and Lewin, 1999). An example is given by the results obtained by the diagnosis community for the well known Tennessee Eastman (TE) process.

The most complete fault diagnosis results on the TE process were given by Raich and Çinar, who used angular relations between data clusters (Raich and Çinar, 1997) and techniques for PCA model overlapping quantification (Raich and Çinar, 1996) as the way to manage the fault diagnosis problem. The success of their techniques were reported fault by fault, however, they faced the diagnosis problem from a single-fault view, which led them to neglect the actual problem nature. Thus, their results are partial and hardly assessable, as it will be demonstrated on next subsections.

On the other hand, most of the results on the TE process diagnosis do not consider the problem globally and the partial information reported (such as not considering the original number of faults proposed in the paper by Downs and Vogel (1993)) or the lack of quantitative results does not allow a rigorous comparison between such methodologies. For instance, Maurya et al. (2004) propose a hierarchical signed directed graph (SDG) based on qualitative knowledge requiring further quantitative information to reduce the inaccurate fault candidate sets diagnosed. In spite of the presented methodology benefits

there are deficiencies on the basis for further comparisons.

Attempts to apply SVM to the fault diagnosis problem have been also reported on that benchmark case study by Chiang et al. (2003) and Kulkarni et al. (2005) but no comparable results were shown because they did not deal with the global problem and did not show diagnosis performance indices fault by fault. Besides, misclassification index given is not enough to evaluate the SVM global diagnosis performance.

Chiang et al. (2000) shows that Fisher Discriminant Analysis (FDA) gives a good low dimensional representation in terms of maximizing the separation between different data clusters, but the results on misclassification are not easily comparable because such misclassification is given as a general system diagnosis response, losing information of each fault diagnosis. Furthermore, this index is not explicitly defined, hence not being useful for later comparisons. Another similar case is the work proposed by Lin et al. (2000) which uses a non-linear dynamic PCA (NDPCA) for monitoring and a feed-forward neural network for diagnosing the TE process. In this work, the problem is highly reduced just to two simple faults from those proposed in the original paper by Downs and Vogel (1993). Wachs and Lewin (1999) presented a time-dependent variable deviation analysis including PCA capabilities, which proved proper isolation capabilities. However, among the 20 faults proposed by the original TE case, they only addressed the first seven and did not provide comparative results regarding the global diagnosis problem.

Chen and Liao (2002) addressed the detection partial problem, but without extracting the abnormal events causes. The detection issue may also play a significant role in the evaluation of the performance of a FDS. When the diagnosis system includes a previous detection module, it may wrongly lead to evaluate the performance regarding only the detected samplings instead of referring the actual raw input data (Raich and Çinar (1997) using "Serial" approach represented on Figure 6.4). In these cases, such performance indices do not admit to be compared with straight diagnosis techniques which does not include a detection step, as they are dealing with all raw input data.

From a practical point of view, a detection step makes sense as a decomposition strategy for solving the global problem or as a redundant system for enhancing FDS reliability (Figure 6.4). Nevertheless, this should be taken into account when establishing references. Since direct classification methods may be conceived without intermediate detection steps, it seems more advisable to refer the success of a FDS to the raw input data not only to these samplings out of control. Furthermore, the performance of a diagnosis system could be mistakenly improved by a very robust detection module that only gives pass these samplings certainly out of control.

Besides all the preceding works focused on TE diagnosis, other authors have applied their fault diagnosis methodologies to other case studies (Dash et al., 2003; Genovesi et al., 1999), but once again it is evident that most of them report results that are highly limited to make rigorous comparisons. Thus, the literature analysis on the TE process diagnosis shows that:

- Quantitative results reported are limited for rigorously testing new diagnosis methodologies and no common performance indices have been used with that purpose.
- Not clear distinction between the detection and diagnosis phases makes difficult a correct and systematic comparison.

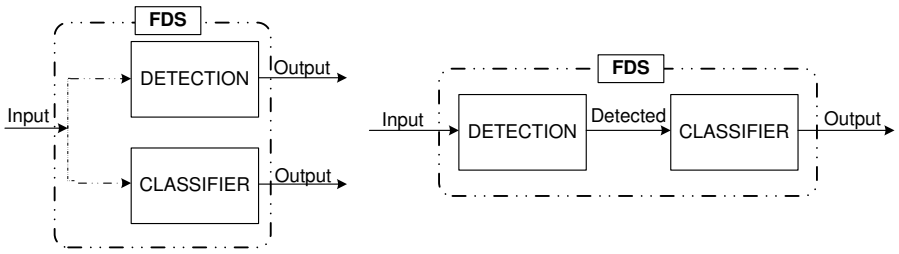


Figure 6.4: "Parallel" and "Serial" FDS approaches.

- Since many simulation runs may be obtained from the same case study (different ways of sampling, several initial and final conditions, simulation span, etc.), complete data sets are required to be agreed on the same basis.

To overcome the first point, a general diagnosis formulation and evaluation procedure were established on chapter 5. The second and third points are tackled along this chapter.

6.3.2 mL-SVM: Results and discussion

In order to compare the SVM performance against other diagnosis techniques, works offering most complete quantitative results on the TE case were considered, showing the lack of comparable data. A first work by Chen and Liao (Chen and Liao (2002), [1] in Table 6.3) only presents detection results for the 1 to 15 TE faults. A second one by Raich and Çinar (Raich and Çinar (1996), [2] in Table 6.3), allows to evaluate Precision and Recall indices as they were defined at chapter 5, but the values cannot be compared as they are referred to previously detected faults instead of actual faults happening ("serial" configuration in Figure 6.4). Not taking into account any of the non-detected samplings, overall Recall and Precision are increased by removing possible wrong diagnosis (subsection 6.3.1). The last work analyzed (Raich and Çinar (1997) [3] in Table 6.3), also refers results to previously detected faults and, despite the better Recall obtained, offers less quantitative information, so that is not possible to evaluate the global diagnosis Precision, as the number of diagnostics for each fault are not known.

6.3.2.1 Data analysis

An extensive analysis to contrast the SVM diagnosis performance with previously reported data, including complete quantitative information, was done. Results are obtained after SVM learning with treated data sets from TE process simulations. Raw data are collected at second intervals during 24-hour runs. Next, aimed at enhancing performance, different tests were carried out in order to determine the data treatment which results on a better tuned SVM. In that sense, several data training sets size, data filtering techniques and parameter values were tested.

In order to select the best strategy, a comparison is made based on global Recall (micro-averaging) described in section 5.3. The 20 faulty classes proposed in the TE case are used in these experiments. It must be noted that, even for testing the same case study, the best SVM performance strongly depends on parameter selection, as the training and

testing sets sampling frequency or the considered classes in the problem. Firstly, data are treated by means of different filtering techniques to reduce useless data noise. For that purpose, different kinds of techniques were tested, including PCA and additional popular filters used in industry as Exponential Weighted Moving Average (EWMA) and more recent wavelet based approaches (Tona et al., 2005). The obtained results corresponding to each data treatment are compared to assess the ability of each of these approaches to improve the SVM classification performance.

Several experiments taking into account these filtering techniques were carried out in order to choose the best data management. In these experiments, a set of 300 statistical samplings for each fault were used in the training stage, fixing a value of 0.1 to the parameter C of SVM. A data filtering by means of EWMA and a later data PCA resulted as the most effective procedure to remove noise from plant data (Table 6.1).

Table 6.1: Data pre-processing methodologies to improve FDS performance.

Data treatment	Simple Sensor	Dealt with PCA	**EWMA + PCA**	Wavelet + PCA
Recall (%)	40,24	23,14	**45,76**	26

Secondly, a SVM tuning using the C parameter is done to allow more or less training errors. The right selection of that parameter supposes a significant influence in the diagnosis results as it can be checked revising Table 6.2. The combined EWMA-PCA was used to continue increasing the SVM performance. Besides the 300-sampled training data set was maintained.

Table 6.2: C parameter tuning to increase the SVM performance.

C	0.1	0.05	0.01	0.005	0.001	**0.0005**	0.0001	0.00005
Recall (%)	45,76	50,05	58,05	57,67	57,76	**58,38**	57,29	56,19

A quite smaller value in the C parameter (C = 0.0005) than the initial one, achieved a great increase on the classification performance (Table 6.2).

This tuning study shows that the use of a soft margin and the allowance of certain training inaccuracies increase the system performance (Figure 6.5). Accordingly, an unnecessary relaxation drops this performance. Although an optimum value seems imprecise, for practical purposes a range of C values is revealed reliable enough for maximum performance.

Finally, a search of the required data training sets size was done, considering both parameter values previously selected (data training size and parameter C). Four different data training sets with 100, 200, 300 and 400 samplings respectively were used to check the SVM classification response. As expected, the best training set resulted to be the largest, which retained the basic data characteristics while not introducing superfluous noise in the training that could have misled the SVM classification. On the other hand, the performance response is an asymptotic function of the training set size (Figure 6.6), which poses a trade-off between the size of the data and the performance gain. However, the 100, 200, 300 and 400 samplings verified a trend far from saturation. Being all computationally affordable, the largest set was selected (Recall = 64.45 %). Deeper analysis

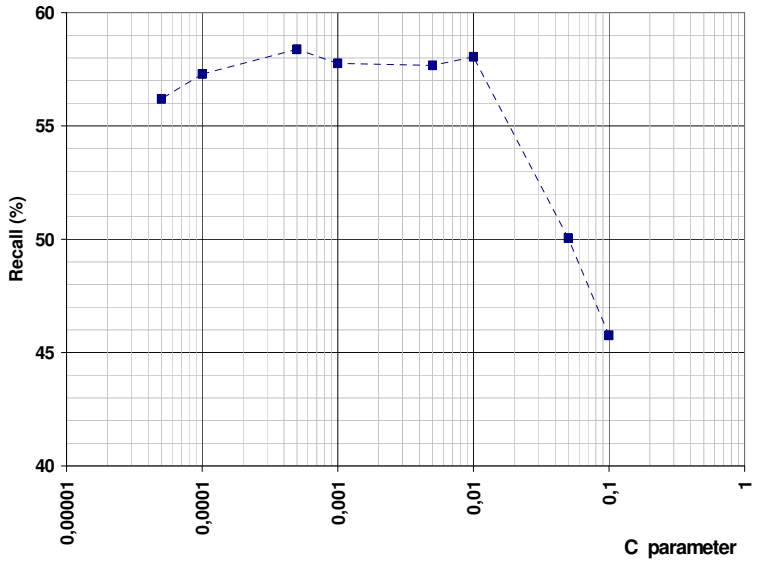


Figure 6.5: C parameter tuning.

would be required for ascertaining the optimum size for the training set.

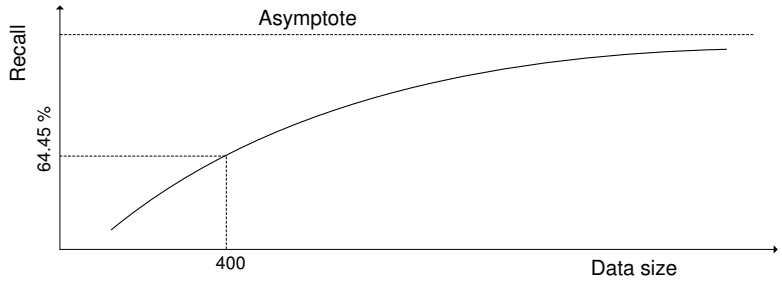


Figure 6.6: Qualitative recall trend when data sets size increases.

Therefore the combination of a 400 sample-training data size, an EWMA-PCA data pre-treatment and a value of 0.0005 in the *C* parameter, resulted in an improved combination in order to achieve higher SVM performance.

6.3.2.2 Effect of the number of classes considered on the classification performance

In order to get a deeper knowledge of the SVM performance solving the TE classification problem, SVM was tested using a series of cases for which the number of samplings was kept constant while increasing progressively the number of states (faults) considered. The results are depicted in Figures 6.7 and 6.8.

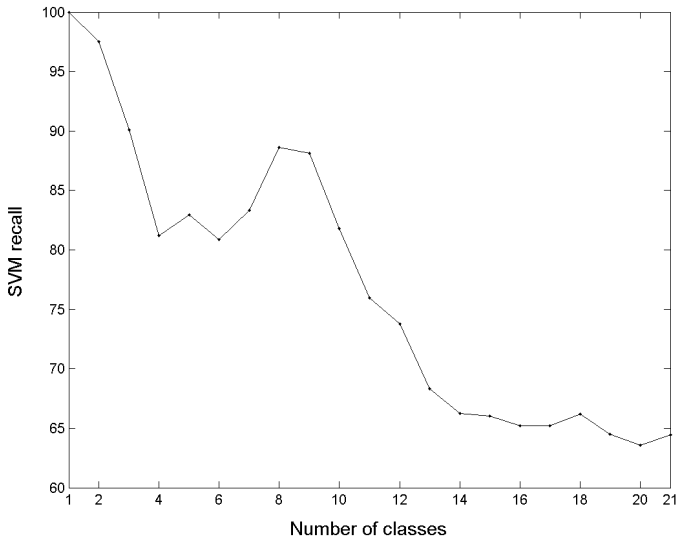


Figure 6.7: SVM micro-averaging recall regarding the number of considered faults. Number of classes indicates that classes $1 + 2 + \ldots + n$ are considered.

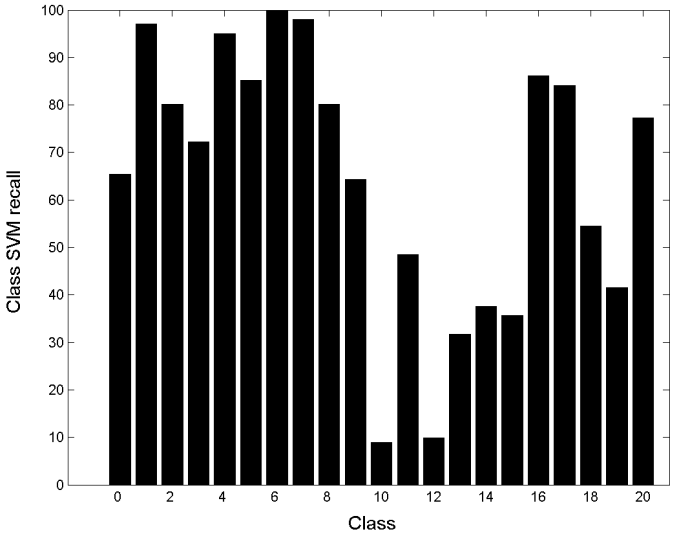


Figure 6.8: SVM faults Recall. Fault 10, and 12 are badly identified.

Figure 6.7 shows how Recall globally decreases as more classes are considered, as mentioned in subsection 5.2. The peak between 6 and 10 (Figure 6.8) is a deviation from the expected behavior caused by accounting faults (6, 7 and 8) with higher Recall than aver-

age. In spite of this effect, the general decreasing trend is very clear along the complete set of tested classes.

6.3.2.3 Comparative results

Results obtained using SVM were then compared with other referenced results in Table 6.3. For each different work, Table 6.3 shows in columns for each fault (rows) the total number of samplings in which fault f happens (S_f), the number of detected samplings for each class (D_f), the times the system diagnoses each class ($V_f, \sum_{i=1}^{S} (a_f + b_f)$ described in section 5.3) and the Precision (Prec) and Recall (Rec) for each fault. Mean values for Precision and Recall are introduced at final rows as comparative global measures. These measures are the Macro-average (MA) and the micro-average related to the number of actual samplings in each case (mA). The micro-average related to the detected samplings (mA*, weighted by the number of detected samplings) is also evaluated to make possible the comparison with some reported results ([2] in Table 6.3).

For comparative purposes also, two different computational experiments were carried out: SSVM ("serial" configuration of the SVM) and PSVM (parallel configuration of the SVM). SSVM addresses the 21 faults and implements fault detection before the SVM classification ("serial" configuration shown in Figure 6.4), which allows comparing results with [2] and [3] (in Table 6.3). Besides, an extra calculation considering all input data was done to show the actual diagnosis Recall (mA), as we know the total happening samplings in the system.

PSVM does not include information on detected faults in order to assess performance just regarding FDS input and output ("Parallel" configuration in Figure 6.4). It only requires three columns for the complete analysis of its diagnosis performance. Moreover, the 21st TE fault is not analyzed in PSVM, since it is a simultaneous fault that should not be treated as an additional and new fault but under the general outlook of a ML approach.

Computational experiments summarized in Table 6.3 have been carried out using the exponential weighted moving average (EWMA) for removing useless noise. PCA was applied as a fault detection technique and a noise removal methodology, which also allowed taking advantage of the meaningful latent scores and contributions to SPE for fault diagnosis.

The number of samplings defining each faulty data set used for training PSVM was set to 400 (subsection 6.3.2.1) and to 300 for the SSVM (was trained with 300 samplings resulted from a similar study that this carried out in subsection 6.3.2.1). Each faulty testing data set consisted of 101 different observations in PSVM and 100 in SSVM. The trained algorithm was implemented considering $C = 0.0005$ in both cases.
Results analysis (Table 6.3) shows that SSVM not only produces better average Recall (MA), but the detection weighted micro-average (mA*), which is much more significant, widely improves the Precision and Recall of previous works. Besides, it offers a general measure of micro-average Recall (mA) that is not known in [2], and [3], as they do not offer information about FDS input data. PSVM gives consistent and promising results on the single fault constrained problem, being the only one that is able to give a global FDS Precision value as it considers all the input data in the diagnosis module.

Table 6.3: Comparison of SVM results with previously reported works. N: Normal class. V_f: Votes to the class f ($a+b$ in contingency matrix); D_f: Detected samplings of class f; S_f: Total number of tested samplings in class f ($a+c$ in contingency matrix); R_f: Recall (%); P_f: Precision (%); ; mA*: mA related to D_f; mA: mA related to S_f. *:Considerated state. **: Variable.

Work	**[1]**					**[2]**					**[3]**					**SSVM**					**PSVM**		
/*	V_f	D_f	S_f	R_f	P_f	V_f	D_f	S_f	R_f	P_f	V_f	D_f	S_f	R_f	P_f	V_f	D_f	S_f	R_f	P_f	V_f	R_f	P_f
N																					393	65	17
1	-	119	120	-	-	-	-	-	100	-	301	238	-	97	76	99	99	100	95	95	98	97	100
2	-	117	120	-	-	-	-	-	93	-	185	241	-	77	100	86	97	100	89	100	81	80	100
3	-	115	120	-	-	-	-	-	29	-	0	55	-	0	0	26	53	100	0	0	87	72	84
4	-	119	120	-	-	-	-	-	74	-	57	234	-	24	100	167	99	100	96	57	128	95	75
5	-	120	120	-	-	-	-	-	98	-	68	240	-	28	100	138	89	100	100	64	158	85	54
6	-	120	120	-	-	-	-	-	93	-	241	241	-	100	100	106	99	100	100	93	101	100	100
7	-	117	120	-	-	-	-	-	62	-	194	194	-	100	100	94	99	100	95	100	99	98	100
8	-	111	120	-	-	-	-	-	99	-	115	216	-	52	98	51	97	100	53	100	81	80	100
9	-	94	120	-	-	-	-	-	12	-	1	78	-	1.3	100	0	47	100	0	0	80	64	81
10	-	66	120	-	-	-	-	-	88	-	166	165	-	76	76	62	96	100	34	53	34	8.9	26
11	-	109	120	-	-	-	-	-	31	-	260	104	-	74	30	60	77	100	3	21	92	48	53
12	-	111	120	-	-	-	-	-	0	-	218	222	-	20	21	0	54	100	0	0	44	10	23
13	-	74	120	-	-	-	-	-	87	-	497	222	-	88	85	68	88	100	70	91	87	32	37
14	-	118	120	-	-	-	-	-	0	-	53	136	-	39	100	37	51	100	18	0	106	38	36
15	-	118	120	-	-	-	-	-	44	-	113	16	-	25	4	0	52	100	0	0	58	36	62
16	-	-	-	-	-	-	-	-	47	-	149	15	-	40	4	98	94	100	91	88	104	86	84
17	-	-	-	-	-	-	-	-	97	-	178	214	-	71	85	135	94	100	98	68	85	84	100
18	-	-	-	-	-	-	-	-	98	-	192	219	-	84	96	87	73	100	97	82	65	54	85
19	-	-	-	-	-	-	-	-	44	-	10	90	-	10	90	200	65	100	49	16	62	42	68
20	-	-	-	-	-	-	-	-	32	-	149	15	-	40	4	86	86	100	100	100	78	77	100
21	-	-	-	-	-	-	-	-	0.9	-	227	219	-	76	34	89	80	100	100	90			
MA	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**58.5**	**-**	**-**	**-**	**-**	**53.5**	**66.7**	**-**	**-**	**-**	**61.3**	**58.0**	**-**	**64.45**	**70.69**
mA*	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**62.6**	**62.6**	**-**	**-**	**-**	**68.7**	**68.7**	**-**	**-**	**-**
mA	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**55.3**	**-**	**-**	**64.45**	**64.45**

The details of all samplings diagnosis distribution for PSVM are given in Table 6.4.

Table 6.4: Summary of confusion matrices obtained after classifying the TE benchmark by PSVM.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
a_f	98	81	73	96	86	101	99	81	65	9	
b_f	0	0	14	32	72	0	0	0	15	25	
c_f	3	20	28	5	15	0	2	20	36	92	
d_f	2020	2020	2006	1988	1948	2020	2020	2020	2005	1995	
S_f	2121	2121	2121	2121	2121	2121	2121	2121	2121	2121	
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**N**
a_f	49	10	32	38	36	87	85	55	42	78	66
b_f	43	34	55	68	22	17	0	10	20	0	327
c_f	52	91	69	63	65	14	16	46	59	23	35
d_f	1977	1986	1965	1952	1998	2003	2020	2010	2000	2020	1693
S_f	2121	2121	2121	2121	2121	2121	2121	2121	2121	2121	2121

Note that these values allow to calculate any of the PSVM results columns (Table 6.3).

6.3.3 Assessment

Interesting results have been obtained by comparing SVM with previous reported works. On the hand, SVM diagnosis capabilities were tested on a well known benchmark case study. A competitive diagnosis response was achieved when comparing it with previous quantitative diagnosis reported results. On the other hand, this comparison also revealed critical deficiencies in the way the general diagnosis problem has been addressed. Aside from the limited data usually available in the literature, performance indices utilized to formally compare diagnosis results are often selected specifically to the case. A rigorous revision of performance criteria has identified past deficiencies and helped setting the general diagnosis problem.

Such aspects have arisen when trying to extend the single fault constrained problem to the actual combinatorial diagnosis problem. In that sense, simultaneous faults have been tackled as new faults by considering a mL view. Much has been learned from the literature review and the SVM application.

1. SVM is a flexible an efficient algorithm to deal with complex chemical process data.
2. ML nature of the general fault diagnosis problem of chemical processes must be managed by using a ML approach.
3. mL view always constraints the general fault diagnosis problem to only take into account individual faults.

Next section describes the ML approach using SVM as the learning algorithm.

6.4 General diagnosis problem: A ML approach

The ML proposed methodology and its results establish the basis for future comparison as well as show the way of rigorously approaching the general diagnosis problem.

6.4.1 Methodology

The proposed FDS was set to face the diagnosis problem from the ML approach. In that sense, the binarization step was arranged to organize the labeled process information under the ML scheme. Then, learning and classification stages generate the required classifiers and evaluate their performances respectively. Finally, a de-binarization and evaluation steps were included to decode and show the FDS performance. The procedure to train, test and evaluate the ML-FDS is shown on Figure 6.9.

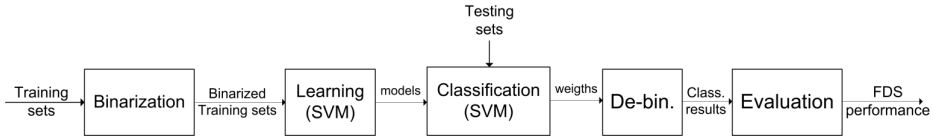


Figure 6.9: ML-FDS performing scheme.

Training and testing sets were random and independently selected to avoid biased results. Thus, each class was first randomly built by sampling original representative data. Then, these new sets were split in two different sets from which training and testing sets may be respectively selected with the desired set size. The complete ML-FDS scheme consists of SVM as the core learning algorithm (see section 6.2), plus a wrapper implemented in Borland Delphi, whereas Matlab was used to make valuable data analysis. The computer used to obtain all the results presented in the chapter was a Pentium IV (2.6 GHz) operating under Windows 2000.

Next subsections describe the methodology details sketched in Figure 6.9 and outline the main differences with the mL approach. The methodology description is illustrated through an example (Figure 6.10) consisting of two variables (x1 and x2), three single classes (including normal class) and a composed class (1-2), and two samplings per class.

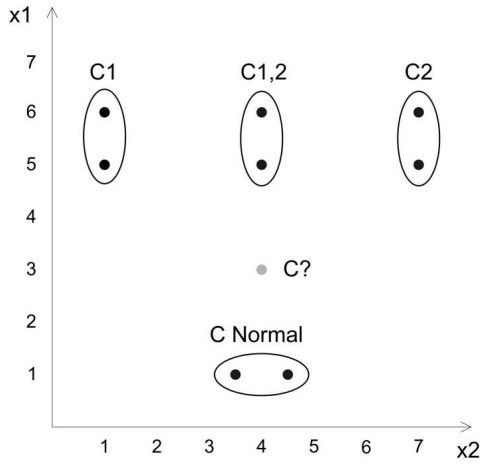


Figure 6.10: Training and testing examples. Black dots are labeled samplings, while the grey dot is the testing sampling.

Training and testing samplings coordinates are given in two first rows of Table 6.5.

Table 6.5: Problem Class Samplings and Binarized Sets Codification (C? refers to the unlabeled sampling).

	C1	C2	C1,2	C Normal	C?
Training (X1, X2)	(5,1), (6,1)	(5,7), (6,7)	(5,4), (6,4)	(1,3.5), (1,4.5)	-
Testing (X1,X2)	-	-	-	-	(3,4)
1st ML binarized set	+1,+1	-1,-1	+1,+1	-1,-1	-
2nd ML binarized set	-1,-1	+1,+1	+1,+1	-1,-1	-
1st mL binarized set	+1,+1	-1,-1	-1,-1	-1,-1	-
2nd mL binarized set	-1,-1	+1,+1	-1,-1	-1,-1	-
1+2 mL binarized set	-1,-1	-1,-1	+1,+1	-1,-1	-
Normal mL binarz. set	-1,-1	-1,-1	-1,-1	+1,+1	-

6.4.1.1 Binarization

The ML based binarization for each class consisted of labeling as positive all the samplings belonging to the class while assigning negative values to all other samplings ("one vs. all" (Allwein et al., 2000)). In the example, two training binarized sets were built, as just two classes must be trained under the ML approach. Labels for each training set are shown in third and fourth rows of Table 6.5. In case a mL approach had been used, two additional classes (Normal class and 1+2 class) should have been trained, thus complicating the later learning step (6.3.2.2). The four last rows in Table 6.5 show the mL particular binarization.

For each fault f, the binarized data set is given by the following matrix:

$$B_f = \left\{ x_{1s}, x_{2s}, \ \ldots \ , x_{vs}, h_{sf} \mid s \in \ \ learning \right\}, \ \forall \, f \tag{6.2}$$

being learning the training subset of samplings.

6.4.1.2 Learning

These training sets previously binarized (one per fault; two in the example) are used to generate the different models during the learning stage (also one per fault; two in the example, instead of the four that would require the mL approach). Then, SVM are used as the binary learning algorithm which establishes the hyperplanes to separate positive (belonging to the class) from negatives samplings (not belonging to the considered class). The result of this step, using SVM, is a model for each class f given by a vector w_f (with one component for each considered variable) that characterizes the shape of the hyperplane and an offset parameter b_f which stands for the distance of the hyperplane to the origin (Figure 6.1).

6.4.1.3 Classification

During the classification or testing step, new and not labeled samplomgs (testing samplings in Figure 6.10 and Table 6.5) are "weighted" using each classifier model (w_f and b_f). The weight for each sampling is the result of the sum of b_f, and the inner product of the model vector (w_f) and the testing sampling vector. These weights (y_{sf} in Eq. 6.4) are

the basis for the sampling classification on the evaluation step.

Being the testing subset of samplings:

$$x_s = \{x_{1s}, x_{2s}, \ \dots \ , x_{vs} \mid s \in \ \ classification\} \tag{6.3}$$

weights are evaluated by:

$$y_{sf} =< w_f, x_s > +b_f \tag{6.4}$$

where "$< w_f, x_s >$" represents the inner product of the w_f vector and the tested sampling vector (x_s). In that sense, quantitative information for classifying any tested sampling is available, and will be used later to make the diagnosis.

Following the comparison with the mL view, no difference can be found on the learning or classification steps but that of working with more classes (four, in mL approach) than in the ML approach (just two classes) due to the dragged differences from the binarization.

6.4.1.4 De-binarization

By this step the system decodes weights given in the classification y_{sf} into diagnosis responses d_{sf}, thus assigning a class to each treated sampling. It may be done by simply identifying the positive weights with the Heaviside step function (Eq. 6.5).

$$d_{sf} = \theta\left\{y_{sf}\right\} \tag{6.5}$$

In our case, the testing sampling (3,4) obtained the same negative weights for each classifier (-0.57), hence being classified as normal class (d_{s1}, d_{s2} and $d_{s3} = 0$) which includes those samplings not diagnosed as any of the learned classes.

Under the mL approach a significant difference can be outlined at that point. As just one class is diagnosed for each sampling, the diagnosis is made by the "winner takes all" methodology for which the sampling is classified as that class with the highest weight. Thus, correlation between y_{sf} and d_s would be expressed by equation 6.6 (note that under the mL view, subindex f is not required).

$$d_s = \arg\max_f \left\{y_{sf}\right\} = f \quad \mid y_{sf} \geq y_{sk}, \quad \forall\ k = 1, \dots, F \tag{6.6}$$

The tested sampling (Figure 6.10) was also identified as normal class by the mL approach because of its highest weight, 0.0 versus -1.14, -1.14, -1.0 of classes 1, 2 and 1+2 respectively. By the de-binarization step, the ML approach allows training each class with specific and differentiated variables, which is a great advantage in terms of computational efficiency and independent class analysis.

6.4.1.5 Evaluation

Finally, it is checked the agreement between the actual sampling class membership and the class diagnosed, in order to evaluate the ML-FDS performance. This matching degree from all the existing point of views can be analyzed by the indices introduced on section 5.3. It is in that stage where the ML approach requires of a more complex implementation. Under such approach, several classes can be correctly diagnosed for the same sampling, which allows diagnosing simultaneous faults from single faults learning at the expense of

solving a more complex evaluation. In that sense, under the ML approach, the sampling correct diagnosis not only requires of the right sampling classification (mL) but of the correct non assignment of the sampling in any of the remaining classes.

6.4.2 Simultaneous faults diagnosis results

Thus, the FDS was trained and tested again with the original 20 faults proposed by Downs and Vogel (1993) but without any additional combination of faults. No preliminary filtering or data pre-treatments were applied to the raw plant data and no tuning was applied to the SVM (default soft margin and linear kernel were utilized). Despite taking no advantage of these improving opportunities, the approach outlined the crucial points of the multiple fault diagnosis analysis and showed very promising performance, as it will be shown in the case studies that follow.

In order to check the differences with respect to the mL approach the constrained single fault diagnosis problem was addressed by both approaches.

6.4.2.1 Case study

6.4.2.2 Single faults diagnosis problem

Firstly, the no combinatorial diagnosis problem is approached from the mL and ML viewpoints. The same training and testing data sets of 351 and 1000 respective samplings per single fault from TE process simulations were used to check SVM classification capabilities under mL and ML approaches. Additionally, 112 samplings under normal operating conditions (no faults) were included in all test data sets in order to include the normal state in the analysis. Master original data sets were extracted from the simulation of a 20 hour operation horizon.

Sampling every 10 seconds resulted on data matrices having 7200 rows (samplings) per 52 columns (variables) for each single fault. The mentioned training and testing sets were built by a random sampling of the considered variables for each existing fault, first splitting the simulation samplings in training and testing sets and then fixing the percentage of required samplings in each set. The 52 variables considered correspond to the 41 TE process measured variables (XMEAS in the original paper) plus the first 11 manipulated variables (XMV in the original paper). The 12th manipulated variable (XMV12: Agitator speed) was not used because it remains constant under any tested operating condition, thus not being relevant from the fault diagnosis viewpoint. The data sets building procedure is shown in Figure 6.11.

The diagnosis results obtained for these sets are given in Table 6.6, which shows exactly the same values of the F1 index for both classification methodologies except for the normal class (named as 0).

Strictly speaking, the normal case should not be considered as a class under the ML approach, since it is an artificial class that excludes all other classes. Therefore, this class (no faults) is better diagnosed by the mL approach because of the advantage of training this class as an additional state and because of the ML approach efficiency on the normal class critically depends on the achievement of very high classification performance in the rest of classes. Table 6.6 also shows a similar classification capability for those cases

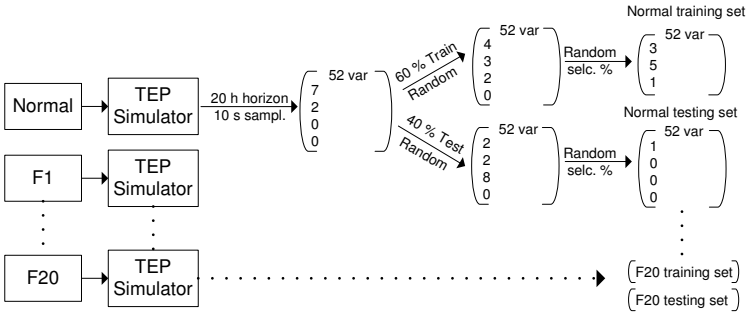


Figure 6.11: Training and testing data sets building procedure.

Table 6.6: Single-fault diagnosis. F1 index (%) for the monoLabel (mL) and MultiLabel (ML) approaches.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
mL	96.2	96.7	93.3	8.4	89.3	3.7	100	100	41.8	0	37.0
ML	13.5	96.7	93.3	8.4	89.3	3.7	100	100	41.8	0	37.0
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**Average**
mL	2.8	0	0	10.2	0	16.1	94.8	78.2	10.2	85.9	**45.9**
ML	2.8	0	0	10.2	0	16.1	94.8	78.2	10.2	85.9	**42.0**

revealed more difficult (faults 3, 5, 9, etc.).

Once validated the matching of the mL and ML approaches for the single fault case, the ML approach was checked regarding the sample influence. With that purpose, a 50 hour operation horizon was simulated collecting data every second (sample 2, S2). Then, after splitting simulation data on training and testing sets, a random selection allowed building equal-sized but differently sampled training and testing sets (351 and 1000 samplings respectively) than those from first sample (S1). The overall FDS performance obtained from each sample is compared in Figure 6.12.

Although some differences can be appreciated between performances obtained from different samples, F1 index shows a similar trend for each class, thus validating the original performance result on S1. Particularly, faults 1, 2, 4, 6, 7, 17, 18 and 20 are very efficiently classified in both cases, while faults, 3, 5, 9, 11, 12, 13, 14, 15, 16 and 19 are badly classified despite the sample. Performance differences in faults 8 or 10, which are also misclassified, are due to differences on specific dynamic features captured by each sample (that comes from the random nature of these faults).

Quantitative results regarding the normal class should be considered merely illustrative and not relevant. The identification of the normal state (a case excluding the rest of the fault classes in the ML approach considered) is a detection problem that can be more easily identified by using a parallel detection system that is simpler and normally more robust than a diagnosis focused tool. The results on that normal class may be given as additional information to check that the FDS does not diagnose any of the trained faults. Good results in normal class under the ML approach could be just expected in case the

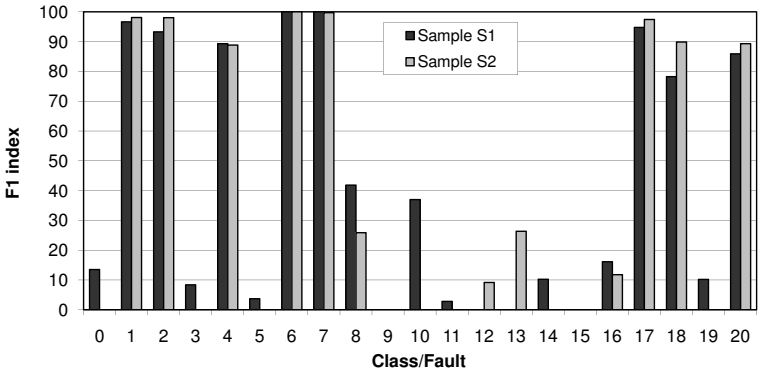


Figure 6.12: ML approach: Comparison of F1 index fault by fault using two different data samples.

remaining states were properly addressed.

6.4.2.3 Simultaneous faults results

Considering the equivalent results produced (Table 6.6) and the evident formulation advantages for the diagnosis of simultaneous faults (check section 5.2), it is clear that the ML view must be adopted in a classification problem such as fault diagnosis in chemical processes. Thus, different simultaneous diagnosis tests were arranged. Combinations of two, three and up to four simultaneous faults were simulated and results obtained when diagnosing them are shown in Table 6.7. Just faults that were individually well diagnosed (Table 6.6) were selected to build simultaneous faults because they cannot be identified whether composing faults are not properly recognized.

Table 6.7: Multiple-fault diagnosis. F1 index (%) under MultiLabel (ML) approach.

	1	**2**	**6**	**7**	**17**	**18**
2 Simultaneous faults (1,2)	97.2	96.1	-	-	-	-
2 Simultaneous faults (6,7)	-	-	96.6	99.9	-	-
2 Simultaneous faults (17,18)	-	-	-	-	90.5	64.6
3 Simultaneous faults (2,7,17)	-	82.8	-	68.7	91.8	-
4 Simultaneous faults (2,7,17, 18)	-	80	-	58.8	86.6	17.8

The first three tested pairs were successfully diagnosed as both single faults were identified when happening simultaneously with very high performance (F1). Very good performance is also obtained when diagnosing three and four simultaneous faults, although it is not surprising to get lower F1 values when more faults arise simultaneously. When faults 2, 7 and 17 are simulated at the same time, the FDS is able to identify correctly the isolated faults with which the system has been trained. In the four simultaneous faults case, the F1 index just suffers a significant decrease for the 18th fault, while faults 2, 7

and 17 are being correctly diagnosed by the system.

Table 6.8 shows the performance details for the case of four simultaneous faults, including Precision, Recall, and the individual class assignment, allowing thorough analysis about its goodness.

Table 6.8: Details for the classification of a case consisting of four simultaneous faults (2, 7, 17, 18).

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
a	0	667	0	0	0	0	416	0	0	0
b	0	0	147	0	1	114	0	143	99	0
c	0	333	0	0	0	0	584	0	0	0
d	1112	112	965	1112	1111	998	112	969	1013	1112
Prec	-	100	0	-	0	0	100	0	0	-
Rec	-	66.7	-	-	-	-	41.6	-	-	-
F1 (%)	-	**80**	0	0	0	0	**58.8**	0	0	0
	11	12	13	14	15	16	17	18	19	20
a	0	0	0	0	0	0	763	98	0	0
b	1	0	0	116	0	168	0	0	75	16
c	0	0	0	0	0	0	237	902	0	0
d	1111	1112	1112	996	1112	944	112	112	1037	1096
Prec	0	-	-	0	-	0	100	100	0	0
Rec	-	-	-	-	-	-	76.3	9.8	-	-
F1 (%)	-	-	-	-	-	-	**86.6**	**17.8**	-	-

One of the most significant points to be highlighted from these detailed results is the low mis-diagnosis rate (*b* in Table 6.8) of the FDS. Note that, the sum of samplings in any column of Table 6.8 is 1112, which corresponds to 1000 samplings belonging to that fault, plus 112 samplings belonging to the normal class. For comparative purposes again, simultaneous fault results in Table 6.7 are confronted with those obtained using the second sample described in the previous subsection (Figure 6.13). Equivalent results are achieved again, following the expected predictions.

The results presented show a very high diagnosis performance facing two, three and up to four simultaneous faults, given that the classes are those correctly trained and diagnosed (Table 6.6). Since it is not previously reported, they are very remarkable from different viewpoints:

- The entire set of faults (20) in the original TE process is addressed considering non-excluding classes.
- The results include the complete and required information to evaluate quantitatively the FDS performance in all senses (accuracy, reliability, misdiagnosis, etc).
- They show correct two, three and up to four simultaneous faults diagnosis.
- Moreover these results are obtained under a flexible methodology that only requires training single classes, thus:
  - Without any need for defining artificial classes for multiple faults, thus considering any possible faults combination.
  - Without any need for training these extra classes, which results in higher computational efficiency and an increase of the overall classification performance.

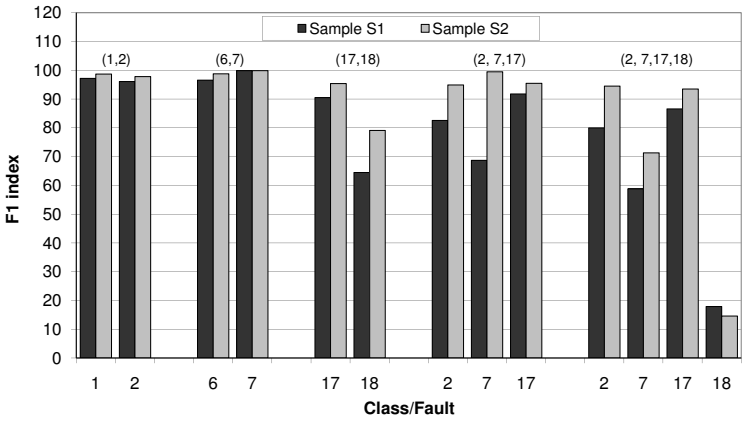


Figure 6.13: Comparison of F1 index diagnosing simultaneous faults using two different data samples.

Nevertheless and despite of these excellent results coming from the SVM classification capabilities and from the ML approach considered, certain faults (3, 9, 10, 11, 12, etc.) are badly identified (Table 6.6) and their possible simultaneous combinations will not be either properly diagnosed. For that reason, and considering the high dependence of machine learning results on the information provided, a data analysis was made to improve the information representation utilized as input data by the system.

6.4.2.4 Performance analysis

The appropriate sizes for training and testing data sets were selected from a preliminary study on the F1 index response (similarly as it was done on section 6.3.2.1). The learning curve in Figure 6.14 plots F1 index averaged for all the considered faults versus different training data set sizes. All these training sets were extracted from the master split training set (subsection 6.4.2.2) in such a way that each greater set includes samplings of smaller sets. As a result, the set containing 351 samplings for each single fault was selected as the trade-off between the system classification performance and the computational cost (training set: (351x20)). The 1000-sampling testing set previously used was employed for obtaining these results (files are available at http://ciao.euetib.upc.es[1]).

Splitting the global learning curve into its individual classes (Figure 6.15) permits focusing on diverse individual classification behaviors. In that sense, two different groups can be identified in Figure 6.15.

A first group consists of certain faults properly classified for any training set size, increasing their performance by asymptotes which offer an adequate training size (faults 1, 2, 7 ...). The second group is conformed by those faults badly classified by the system (8, 9, etc.). For these faults, performance classification seems not to depend on the training set size. Decrease on normal class performance as well as random F1 index variation of these second group faults come from the respective classifier deficiencies that make the

[1] Also at http://webon.euetib.upc.es/ciao/recursos.html

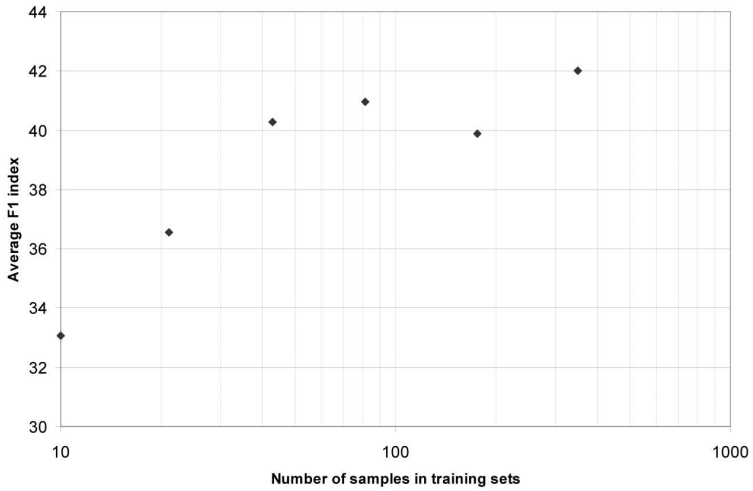


Figure 6.14: F1 average learning curve in logarithmic scale.

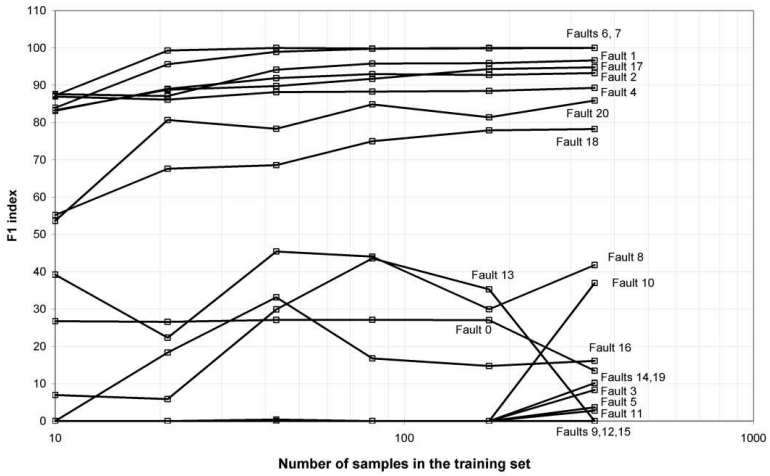


Figure 6.15: Individual learning curves in logarithmic scale.

performance independent of the training set sizes.

Figures 6.14 and 6.15 cannot be considered as true learning curves since there is no asymptotic behavior for some faults, thus being impossible to determine the best training set size. This is the reason for adopting the largest of the set sizes evaluated (351-samplings) for the subsequent experiments. However, both resulted very useful for comparing different fault behaviors. For example, to identify the random nature of the classification performance at some faults.

6.4.2.5 Enhancing information representation

In order to increase the classification performance of this second group, the original data and their characteristics were studied. The measurements utilized to characterize these faults, process variables and manipulated variables, are not significant enough and cannot differentiate some faults from the rest. However, the variances of these process variables were able to capture some representative features of these faults. For example, original variables XMEAS21 and XMV10 (Downs and Vogel, 1993) were randomly modified when fault 14 occurs, whereas other original variables were not altered. Although biases on these original variables were subtle and only were appreciated after variables normalization, they revealed an interesting point. There was a clear opportunity to better collect extra valuable process information, by creating new variables which account for the change of variance of the process variables over time. This new information representation could enhance the classification performance, at least for these faults with a random nature. The procedure followed to take advantage of that characteristic is detailed next.

Figure 6.16 shows the 41 process variables and 11 first manipulated variables given in the TE process (Downs and Vogel, 1993) after simulating a 20 hours operation horizon under the control strategy given by McAvoy and Ye (1994) previously specified (subsection 6.4.2.1).

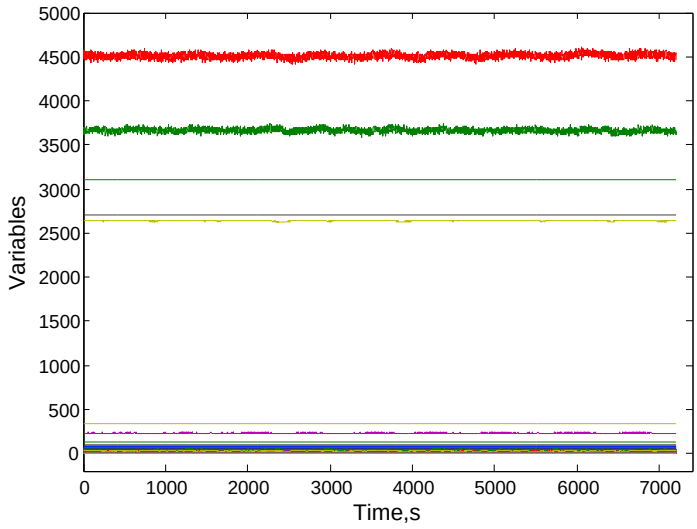


Figure 6.16: TE process measured and manipulated variables.

Fault 14 was introduced at second operating hour and the examples shown are those sampled each 10 seconds. Note that in Figure 6.16, any abnormal situation may be clearly observed from the view of variables XMEAS 21 and XMV10. Figure 6.17 shows the same variables normalized to 0 mean and unit variance in order to adjust all the variables at the same scale and facilitate the analysis.

Figure 6.16 does not show evidences of deviations from the normal operating regime while Figure 6.17 shows a variance increase at least in two of the original variables at 720

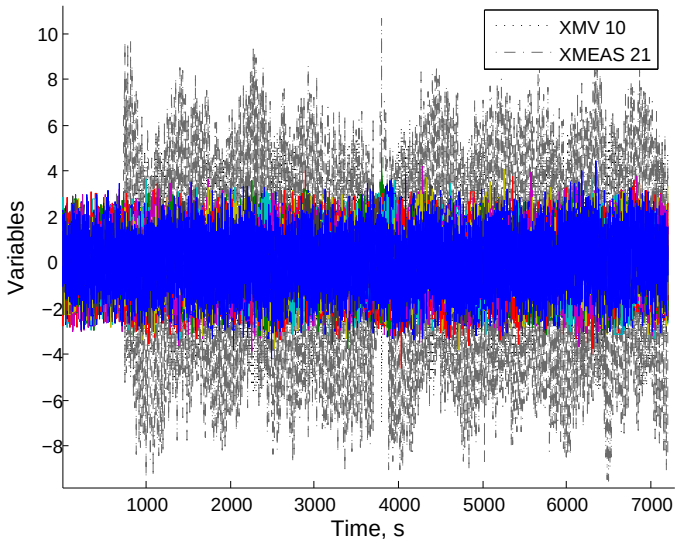


Figure 6.17: TE process measured and manipulated variables, normalized to zero mean and unit variance.

sampling time (fault arising time). This variance increase is not properly characterized by the depicted variables as it just shows random variable variations around the normalized variable mean, which is not significant enough to describe the fault from a machine learning algorithm point of view.

In order to measure the data variance, the standard deviation of each variable was evaluated for a given moving window. 20 samplings were selected to configure the window length, which demonstrated to accurately capture the process variance changes. The result was an additional set of 52 variables capturing the changes of standard deviation in any of the original variables. Figure 6.18 shows how two of these new variables (StdXMEAS21 and StdXMV10) suddenly went off the normal operating regime at sampling 720.

By a later filtering procedure a more homogeneous behavior was achieved in these new variables, facilitating the fault feature extraction and then improving the machine learning algorithm results (Figure 6.19). The filter utilized was the standard Exponential Weighted Moving Average (EWMA) with $\alpha = 0.01$.

6.4.2.6 Results including new information representation: Single faults

Without applying any SVM tuning procedure and including as new variables the standard deviation on those faults badly classified (3, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 19) the F1 average raised from 42.0 % to 76.2 %. The use of specific variables to train each fault represents an additional advantage of the ML regarding the mL approach. It allows the use of different variables to deal with challenging states, hence, avoiding to useless increase the input information to the already solved classes.

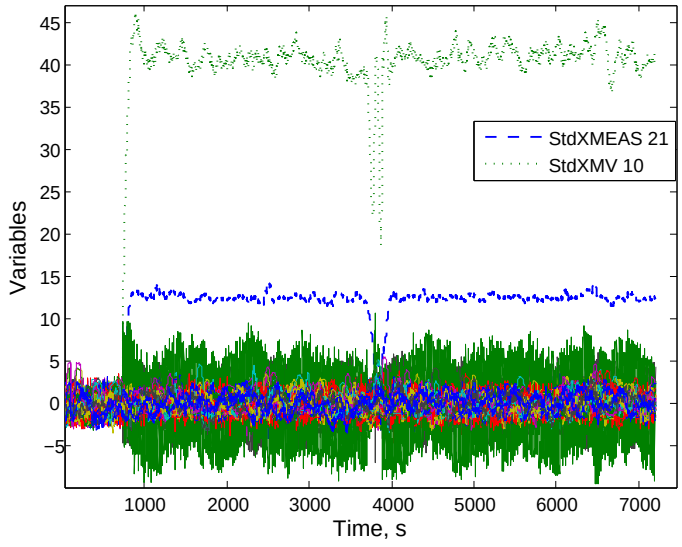


Figure 6.18: TE process normalized variables plus their normalized standard deviation built variables.

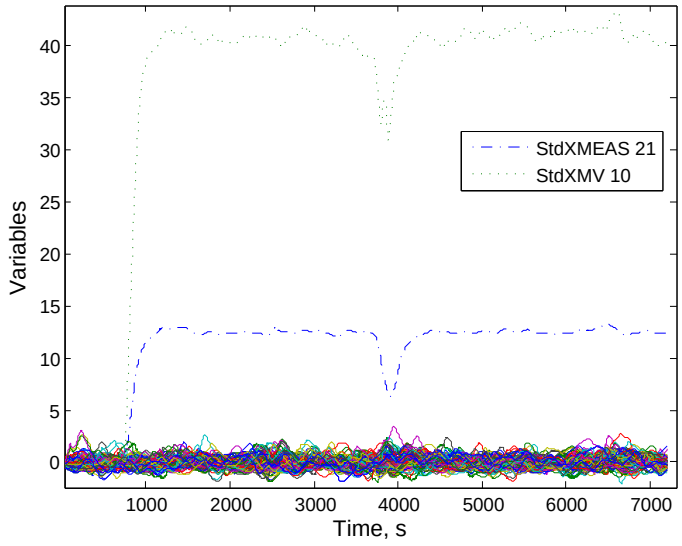


Figure 6.19: TE process normalized and filtered variables plus their normalized standard deviation built variables.

Figure 6.20 compares for each fault/class the individual F1 indices obtained when using raw data with those attained when incorporating standard deviation variables. Faults correctly diagnosed (F1 > 50 %) originally (Faults 1, 2 6, 7, 17, 18 and 20) were trained without including standard deviation variables to avoid increasing unnecessarily data processing. Figure 6.20 shows the same results for these classes. The complete training and

testing data sets used to obtain these results are available at http://ciao.euetib.upc.es[2].

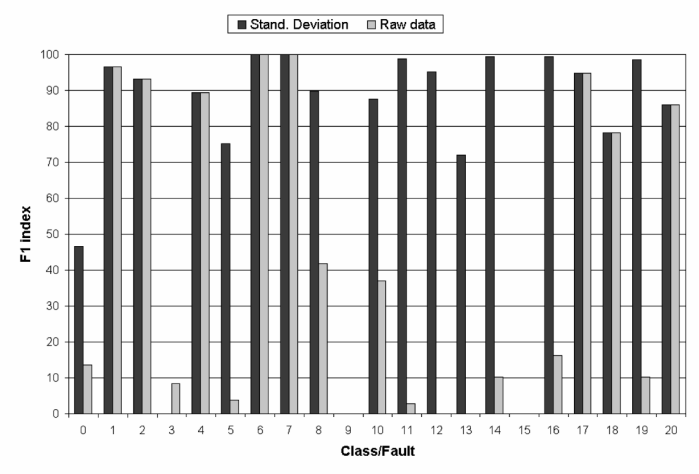


Figure 6.20: F1 indices obtained using original process variables including standard deviation.

Once again, the mL and ML approaches are compared. The results achieved including standard deviation are detailed in Table 6.9 for both cases.

Table 6.9: Single-fault diagnosis. F1 index (%) for the monoLabel (mL) and MultiLabel (ML) approaches.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
mL	96.2	96.7	93.3	0	89.3	75.2	100	100	89.8	0	87.7
ML	46.6	96.7	93.3	0	89.3	75.2	100	100	89.8	0	87.7
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**Average**
mL	98.8	95.3	72	99.3	0	99.4	94.8	78.2	98.6	85.9	**78.6**
ML	98.8	95.3	72	99.3	0	99.4	94.8	78.2	98.6	85.9	**76.2**

Similarly as demonstrated in subsection 6.4.2.2, the difference between both approaches comes from the attempt to infer the diagnosis of the normal state, which would only achieve good quantitative results when the rest of states are perfectly addressed. The normal class difficulties could be managed using a reliable fault detection system from the wide variety of reliable systems available in the literature (Yoon and MacGregor, 2000; Bakshi, 1998; Gertler, 1998). Note that the inference of the normal class is now enhanced (from F1 = 13.5 in Table 6.6 to F1 = 46.6 in Table 6.9) due to the improvements attained in the classification of the rest of considered classes. Learning curves for these novel solutions are depicted on Figure 6.21.

[2] Also at http://webon.euetib.upc.es/ciao/recursos.html

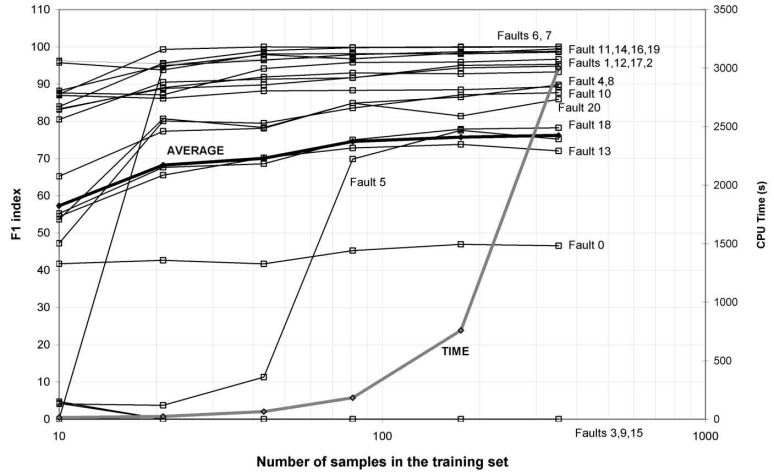


Figure 6.21: Individual and average learning curves in logarithmic scale join to their calculation times.

Comparing the classification performance versus that shown in Figure 6.14 an evident change of behavior is observed. Many of the faults badly classified with the original process variables (faults 4, 10, 11, etc) now belong to these faults following asymptotic learning curves. Nevertheless, several classes (3, 9, and 15) remain poorly identified thus requiring of a different information representation able to gather their specific features by means of alternative variables.

The CPU time required for training each data set size is also depicted in Figure 6.21 right hand vertical axis. It exponentially increases with doubling number of samplings in training sets. Comparing the F1 asymptotes vs. the CPU time, an intelligent trade-off decision between training calculation time and classification performance can be made. In that case, the 351-sampling training set is selected because of its better performance and attainable training calculation time.

6.4.2.7 Results including new information representation: Simultaneous faults

Including the new derived variables, most of the faults are correctly diagnosed, while just three classes remain being poorly identified (fault 3, 9 and 15). Therefore, the new information representation allows extending the simultaneous fault diagnosis to other combinations. Table 6.10 summarizes some two, three and four simultaneous faults diagnosed by the ML based FDS when including the new data representation.

Not only double and triple simultaneous faults shown in Table 6.10 are identified with a high performance, but even combinations of more faults are well diagnosed. Four simultaneous faults were also considered. The first example (faults 2, 7, 17, 18) consists of those faults previously tested. Same results were obtained since the same variables are considered (check Table 6.7). The second and third set of four simultaneous faults correspond to novel fault combinations involving faults that were not individually identified initially but which classification performance was notably increased later by the new information

Table 6.10: Multiple-fault diagnosis. F1 index for different simultaneous faults.

	1	**2**	**6**	**7**	**11**	**14**	**17**	**18**
2 Simultaneous faults (1,2)	97.2	96.1	-	-	-	-	-	-
2 Simultaneous faults (2,7)	-	76.8	-	82.1	-	-	-	-
2 Simultaneous faults (6,7)	-	-	96.6	99.9	-	-	-	-
2 Simultaneous faults (11,14)	-	-	-	-	97.2	75.8	-	-
2 Simultaneous faults (17,18)	-	-	-	-	-	-	90.5	64.6
3 Simultaneous faults (2,7,17)	-	82.8	-	68.7	-	-	91.8	-
3 Simultaneous faults (2,11,14)	-	94.2	-	-	94.9	76.9	-	-
3 Simultaneous faults (7,11,14)	-	-	-	99.9	34	57.8	-	-
4 Simultaneous faults (2,7,17,18)	-	80	-	58.8	-	-	86.6	17.8
4 Simultaneous faults (2,7,11,14)	-	77.1	-	84.9	30.5	59.6	-	-
4 Simultaneous faults (2,11,14,17)	-	94.2	-	-	20.8	9.2	94.7	-
4 Simultaneous faults (6,7,1,2)	0	0	100	100	-	-	-	-

representation (subsection 6.4.2.5). Now, these faults are not just identified when occur individually but when they arise simultaneously with others. Note that the simultaneous testing data sets used to obtain these results are also available at http://ciao.euetib.upc.es[3].

However, some faults cannot be simultaneously identified in the presence of other faults due to the overlapping of fault contributions. Such effect becomes more evident when fault simultaneity increases. A clear example is given by the fourth four-simultaneous fault shown in Table 6.10. It includes fault 6, which biases more heavily variables 1, 7 or 16 of the TE process than the other faults. These variables are key variables that the classification system uses to identify faults 1 or 2, and the contribution of these faults to the simultaneous fault is unobserved by the effect of fault 6. Identification of faults 18, 11 and 14 are strongly penalized in first three four-simultaneous faults respectively because of that effect.

Such overlapping deficiencies can be viewed as a problem or as a natural and appropriate FDS response. In that sense, when a more serious fault (biases deeper the same variables) is happening, other faults which affect much less the same variables may be considered negligible and should not necessarily be identified. Next, details of the same tested four simultaneous fault diagnosed initially (2, 7, 17, 18) are shown in Table 6.11.

Although the same F1 indices are obtained for each fault (see Table 6.8), since the same variables are considered for these classes, however, a new samplings distribution is now attained. By using the additional information representation for those faults poorly classified (Table 6.9), a much lower false diagnosis occurs, thus increasing the operators' trust on the FDS response. It can be easily verified by comparison of row "b" contents on Tables 6.8 and 6.11. The 880 wrongly diagnosed samplings (Table 6.8) have come down to 138, thus revealing this additional advantage on simultaneous fault diagnosis by increasing the diagnosis performance of the remaining faults.

[3] Also at http://webon.euetib.upc.es/ciao/recursos.html

Table 6.11: Details for the classification of the improved four simultaneous faults (2, 7, 17, 18) case.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
a	0	667	0	0	0	0	416	0	0	0
b	0	0	0	0	0	114	0	0	0	8
c	0	333	0	0	0	0	584	0	0	0
d	1112	112	1112	1112	1112	998	112	1112	1112	1104
Prec	-	100	-	-	-	0	100	-	-	0
Rec	-	66.7	-	-	-	-	41.6	-	-	-
F1	0	**80**	0	0	0	0	**58.8**	0	0	0
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
a	0	0	0	0	0	0	763	98	0	0
b	0	0	0	0	0	0	0	0	0	16
c	0	0	0	0	0	0	237	902	0	0
d	1112	1112	1112	1112	1112	1112	112	112	1112	1096
Prec	-	-	-	-	-	-	100	100	-	0
Rec	-	-	-	-	-	-	76.3	9.8	-	-
F1	0	0	0	0	0	0	**86.6**	**17.8**	0	0

6.5 Conclusions

This chapter deals with the single and simultaneous fault diagnosis problem from the machine-learning viewpoint by using and comparing the mL and ML approaches. Benefits of ML approach were demonstrated theoretically and practically using a flexible and robust machine-learning algorithm called SVM.

Firstly, a mL view was adopted to compare SVM with previous diagnosis systems on the same basis. A competitive diagnosis response was achieved when comparing it with previous quantitative diagnosis reported results on a challenging benchmark case study, the Tennessee Eastman process (TEP). In addition to the obtained promising results, this comparison revealed critical deficiencies in the way that the general diagnosis problem had been previously addressed. With the purpose to solve this weaknesses a ML approach was adopted.

Then, single and simultaneous faults generated from the TEP were tested and diagnosed by the ML-SVM system offering very good results even when dealing with four simultaneous faults. By means of improving the fault information representation, a process deeper knowledge was achieved resulting on a remarkable FDS performance increase. Note that, diagnosis results obtained under this ML-SVM system are the best ever reported on the TEP. In addition to reach the highest quantitative diagnosis performance indices ever reported, the ML-SVM system results are also noticeable because of they were achieved without the need of training simultaneous faults, thus enabling any possible fault combination. In contrast with the mL view, the ML approach allowed to train any class with specific variables, which contributed to reduce computational cost and to focus efforts on challenging faults.

Nomenclature

b	Hyperplane bias
C	Class or state
C	SVM training parameter
mA	Micro-Average
mA*	Detection weighted Micro-Average
MA	Macro-Average
S	Total number of considered samplings
V	Votes of the FDS
w	Hyperplane weight vector

Operators

θ	Heaviside step function

Subscripts

f	Fault
s	Sampling

Acronyms

PCA	Principal Component Anlysis
PSVM	Parallel Support Vector Machines
SSVM	Serial Support Vector Machines
SVM	Support Vector Machines
XMEAS	Measured Variables in the TE process
XMV	Manipulated Variables in the TE process

CHAPTER 7

An Expert System based FDS

In addition to the presented ML-SVM diagnosis system, a second data driven FDS was developed to take advantage of different complementary diagnosis abilities. Thus, the black-box nature of the ML-SVM based approach is complemented by using a more understandable FDS which can be interacted and modified directly by the operators in charge of the plant control.

Therefore, it is proposed a two-module system consisting of a PCA model which operates as the fault detection system and a set of fuzzy rules grouped as the diagnosis module. The system is fully comprehensible for operators to use and also provides additional abilities which can complement the FDS described first. By this way, integration of two diagnosis techniques is considered in order to address and to go further on the diagnosis research.

7.1 Introduction

Since in general, normal plant operation data is available, fault detection can be easily implemented using univariate statistical process monitoring techniques. Nevertheless, they lead to an important number of false alarms during process state transitions (set point changes). This is mainly because most of the modern chemical processes are multivariate in nature.

Multivariate Statistical Process Monitoring (MSPM) methods as PCA and PLS (MacGregor et al., 1994; Nomikos and MacGregor, 1994; Musulin et al., 2004) are robust methods that can help to reduce the number of false alarms as they monitor the variables correlation in addition to variables themselves. Besides, these methods inform about the contribution of the different variables to changes in the operation conditions. Such information can be used to support operators in the diagnosis task. However, when the number of process variables is high and the dynamics of the process are complex, the interpretation and association of the contribution of the variables with the root causes of faults is not trivial. Moreover, manual interpretation of process deviations makes difficult the supervisory task

of taking on line corrective measures.

Some fault diagnosis algorithms (Vedam and Venkatasubramanian, 1999; Leung and Romagnoli, 2002) based on qualitative modeling have been integrated with PCA to automate the analysis of process variables contributions to perform fault isolation. Although qualitative models guarantee that the root cause is a subset of the proposed set of root causes, they generate several spurious solutions. The filtering of these solutions continues being a field of research. In addition, the development of qualitative models can be a highly time consuming task that requires a considerable level of expertise that is not always available.

Norvilas et al. (2000) presented a methodology to interface PCA with Knowledge-Based (KB) expert systems for monitoring multivariate processes. After the PCA detection (considered as $T^2 > 95\%$ in two consecutive times) they implement *if-then* rules on the variable contributions to T^2 deviations. Although the methodology showed promising results and interesting features (fast and reliable fault diagnosis), it presents important drawbacks that should be considered:

Since the KB is based on standard crisp logic, it can not deal with process uncertainty. As the variables uncertainty causes that events are not exactly reproduced, this diagnosing proposal will lead to large rule bases and misclassification problems. Furthermore, changes in the variables contributions during the fault propagation are not taken into account, which added to the fact that the rules are not introduced by an automatic procedure, makes the system impractical in most complex chemical processes.

Extracting knowledge from data is a very important task in process engineering. Several approaches exist based on the so called intelligent techniques (e.g. ANN or fuzzy systems) to overcome the limitations of classical methodologies. In this context fuzzy systems have important properties:

1. Possibility of linguistic interpretation
2. Universal approximation capabilities
3. Transparency (comprehensibility)

The first two properties are also (hardly) attained by multilayer perceptron neural networks, Support Vector Machines and other machine learning techniques, but all of them have the limitation of being black box systems. On the other hand, fuzzy systems are transparent, specially due to its linguistically interpretable rules. Specifically, in fault diagnosis applications the reasoning transparency is as important as the classification performance in order to obtain a deeper process understanding, useful to increase the diagnosis system operator confidence.

Although diagnosis error minimization has been widely addressed in the literature, the transparency has been hardly considered. Recently, the trade-off between these two objectives has been discussed by Ishibuchi and Yamamoto (2004). When fuzzy systems are applied to high-dimensional problems, their transparency are seriously degraded by the high number of rules and the number of antecedents and linguistic values in each rule.

In this section a Fault Diagnosis System (FDS) is developed integrating fuzzy logic knowledge-based (FLKB) systems with Principal Component Analysis (PCA). Additionally, a simple methodology to automatically extract rules from data is presented. Special

interest has been dedicated to maximize classification transparency. Finally, a genetic algorithm based optimization procedure is applied to optimize the overall system performance, maintaining the comprehensive structure.

PCA model is built only using process normal state data. It evaluates the process variables values and their correlation allowing fast and reliable fault detection. Once detection is performed, a FLKB system is used to evaluate the contributions of each process variable to changes in the process, finding the root causes of the abnormal event detected.

The chapter is arranged as described next. PCA is presented in section 7.2. Section 7.3 presents some desirable characteristics of FLKB. Section 7.4 introduces the PCA and FLKB integrated configuration and section 7.5 shows its automatic design. Section 7.6 describes the optimization algorithm implemented to improve the FDS performance. Finally, the methodology is demonstrated in two case studies.

7.2 Principal Components Analysis

Consider a matrix $\mathbf{X}$ (of dimension $m \times n$) containing data corresponding to m samplings of n process variables. Each column of $\mathbf{X}$ is supposed to follow a normal probability distribution and is normalized with zero mean and unit variance. Let $\overline{\mathbf{X}}$ be this normalized data matrix, and $\mathbf{R}$ its corresponding correlation matrix:

$$\mathbf{R} = \frac{\overline{\mathbf{X}} \cdot \overline{\mathbf{X}}^T}{m-1} \tag{7.1}$$

Then, performing singular value decomposition (SVD) on $\mathbf{R}$,

$$\overline{X} = \mathbf{U}^T \mathbf{D}_\lambda \mathbf{U} \tag{7.2}$$

a diagonal matrix $\mathbf{D}_\lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, where λ_i are the eigenvalues of $\mathbf{R}$ sorted in decreasing order, is obtained. Note that $\mathbf{R}$ is square and symmetric, so the generalized SVD, which is applied to nonsymmetrical matrices, reduces to the above presented case. The corresponding eigenvectors $\mathbf{p}_i$ (columns of $\mathbf{U}$) are the principal components and form an orthonormal base in $\Re^n$.

It is possible to divide the principal components in two orthogonal sets, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_A]$ and $\overline{\mathbf{P}} = [\mathbf{p}_{A+1}, \mathbf{p}_{A+2}, \ldots, \mathbf{p}_n]$. The first containing most of the process variance (called the Dominant Variation Subspace (DVS)) and the second describing the variance due to the noise (called the residual subspace). Methods to select the proper number of PCs are reviewed in the literature (Jackson, 1991; Himes et al., 1994). A reduction of dimensionality is made by projecting every normalized sampling vector $\mathbf{x}'$ in the common variance subspace generated by $\mathbf{P}$ (loading matrix), obtaining $\mathbf{t} = \mathbf{P}^T\mathbf{x}'$, which is called the principal score vector. Because of the normalization of $\mathbf{X}$, when the process is in control, $\mathbf{x}'$ is projected at the origin or in its proximity. Then, the state of the process can be monitored using the Hotelling's statistic, which is a measure of the Mahalanobis distance to the origin of the new space,

$$T^2 = \left\| \mathbf{D}_\lambda^{-\frac{1}{2}} \mathbf{P}^T \mathbf{x}' \right\|^2 \tag{7.3}$$

The test consists of declaring normal operation if $T^2 \leq \delta_T^2$, where δ_T^2 is the control confidence limit of T^2. However, monitoring based only on this statistic is not sufficient because it only shows the deviations in the principal components subspace. If new data are not contained in the reference set of data, the projection of the new observation will move to the residual space, and it is possible that the anomaly is not detected by the T^2 test. To detect this kind of deviation the Squared Prediction Error statistic (SPE) is usually used. SPE is defined as the sum of the quadratic error between $\mathbf{x}'$ and the reconstructed signal after the dimensionality reduction,

$$\mathbf{\Delta x}' = \mathbf{x}' - \mathbf{P}\mathbf{P}^T\mathbf{x}' = (\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x}' = \widetilde{\mathbf{C}}\mathbf{x}' \tag{7.4}$$

Thus,

$$SPE = \left\|\widetilde{\mathbf{C}}\mathbf{x}'\right\|^2 \tag{7.5}$$

If $SPE \leq \delta_{SPE}^2$, where δ_{SPE}^2 is the control limit for SPE, the data vector is declared normal. Different approaches to obtain the threshold limits are presented in Jackson (1991). In this work thresholds are fixed in such a way that a certain percentage (typically 1% and 5%) of the statistics values are under them.

T^2 and SPE contributions. First attempts to diagnose chemical processes based on PCA were done by Wise et al. (1989). They presented a fault diagnosis method that consists of defining different zones on the DVS, each one corresponding to a different process state. This approach is useful in simple processes when the number of signals and faults is not large.

Later on, MacGregor et al. (1994) proposed the idea of the contribution plots to identify the process faults. Contribution plots give very valuable information about the group of deviated process variables that contribute more to deviations in the SPE and T^2 statistical plots. Variable's contributions to SPE deviations can be evaluated directly as $\Delta x'$ (see Eq. 7.4). Thus, if the variation in the model space (T^2) becomes large, the contribution of each individual variable (x_j) to a large value of the $i-th$ score (t_i) can be evaluated as,

$$Cont(\mathbf{x}(j)) = p_{ij}\,\Delta\mathbf{x}(j) \tag{7.6}$$

where p_{ij} is the weight of the $j-th$ variable x_j in the $i-th$ latent dimension (i.e. $p_{ij} = \mathbf{p}_i(j)$) and $\Delta\mathbf{x}(j)$ is the error in the estimation of x_j evaluated in the current time. More details are given by Kourti and MacGregor (1996).

In this work, diagnosis is implemented analyzing the latent scores and the variable's contributions to SPE. The inclusion of latent scores as inputs makes the FDS sensitive to disturbances that causes the plant to run away from the normal region described by the model (i.e., deviations into the DVS). Besides, the contribution of each variable to the SPE allows to diagnose abnormal events which projections go outside of the DVS.

Additionally, PCA shows which variables have caused the abnormal situation and facilitates the identification of the incidence root causes. Finally, note that PCA allows to know the set of variables more influenced by the abnormal event, even in those cases in which no previous information is stored in the KB (novel faults).

7.3 Fuzzy logic

Fuzzy inference (Jang and Sun, 1997) applied to fault diagnosis is the process of formulating the mapping from a given input (i.e. process measurements) to an output (i.e. process state) using fuzzy logic. Then, the mapping provides a basis from which decisions can be made, or patterns discerned.

The basic structure of a fuzzy inference system consists of three conceptual components: a rule base, which contains a selection of *if-then* fuzzy rules, a database or dictionary, which defines the membership functions used in the fuzzy rules, and a reasoning mechanism, which performs the inference procedure upon the rules and a given condition to derive a reasonable output or conclusion. In this work, the Mamdani fuzzy model (Mamdani and Assilian, 1975) was used.

7.3.1 Interpretability of fuzzy systems

A good method for constructing fault diagnosis fuzzy classificators should not only aim to find the best classification performance, but also to extract knowledge in form of rules easily understandable by human operators. Interpretability has not taken much attention in the recent years due to the erroneous concept that the rules are always easy to understand by humans. This assumption is not necessarily true, specially for complex systems. When fuzzy systems are obtained through data, there are several important properties that should be evaluated to assure interpretability (Wang et al., 2005): completeness, distinguishability, non-redundancy, compactness and utility.

Completeness and distinguishability:

For each input variable x_i (an element of the input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$), there exists M_i fuzzy sets represented by $A_1(x_i), A_2(x_i), \ldots, A_{M_i}(x_i)$. Then the partition of the fuzzy sets is complete if the following conditions are satisfied:

$$\forall x_i \in U_i,\ i \in [1, \ldots, n],\ \exists\, A_j(x_i) > 0.\ \ j \in [1, \ldots, M_i] \tag{7.7}$$

where U_i is the universe of x_i and n is the dimension of the input vector.

The concepts of completeness and distinguishability of fuzzy systems are usually expressed in the literature through a fuzzy similarity measure. When selecting the membership function parameters it should be taken into account the overlapping between them that is directly related with their similarity. Let's assume A and B are two fuzzy sets, therefore their similarity, $\mathbf{\Psi}(A, B)$, can be defined as the result of the division of the area of their intersection by the area of their union (Paiva and Dourado, 2004):

$$\mathbf{\Psi}(A, B) = \frac{\| A \cap B \|}{\| A \cup B \|} \tag{7.8}$$

where the fuzzy intersection and union are performed, respectively, by the minimum and maximum operators. If the fuzzy sets have an acceptable value of similarity, and are not completely overlapped, completeness and distinguishability are fulfilled.

Non-redundancy:

The redundancy is a concept related with the rule base. A rule is redundant if the information that it adds is included on other rules of the rule base. Therefore, the rule should be eliminated to improve computational performance.

Compactness:

A compact fuzzy system is one that has the minimum number of fuzzy sets, fuzzy rules, and fuzzy variables. Compactness is specially worth when several variables are analyzed.

Utility:

The fuzzy system utility is assured when all the fuzzy sets, in which the fuzzy variables space has been defined, are used in at least one rule.

7.4 Integrated configuration

Figure 7.1 presents the integrated PCA - FLKB configuration to perform on-line fault diagnosis. First, a PCA model is created, using process normal state data. Later, this model is used in the detection and symptom generation stage. It allows to perform fast fault detection using the T^2 and SPE statistics. When one of the statistics exceeds the detection limits for more than two consecutive samplings, an abnormal state is detected and the isolation stage starts. The variables contribution to the SPE (changes in the process correlation) and the values of the first A latent variables (changes in the operation point) are evaluated by the fuzzy system to infer the current process status (i.e. fault isolation stage).

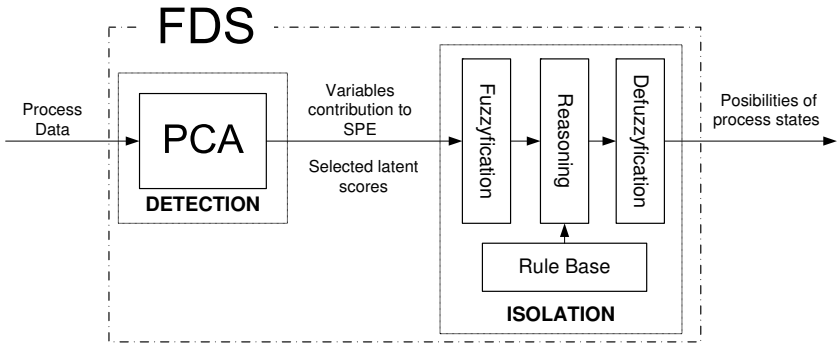


Figure 7.1: Integrated configuration of PCA and the FLKB for fault diagnosis.

7.5 Fuzzy system automatic design

7.5.1 Fuzzy space partitioning: Membership functions

In order to implement the FLKB, the fuzzy system inputs (i.e. latent variables ($\mathbf{t}$) and variables contributions to the SPE ($\Delta\mathbf{x}'$, see Eq. 7.4)), must be normalized to take zero

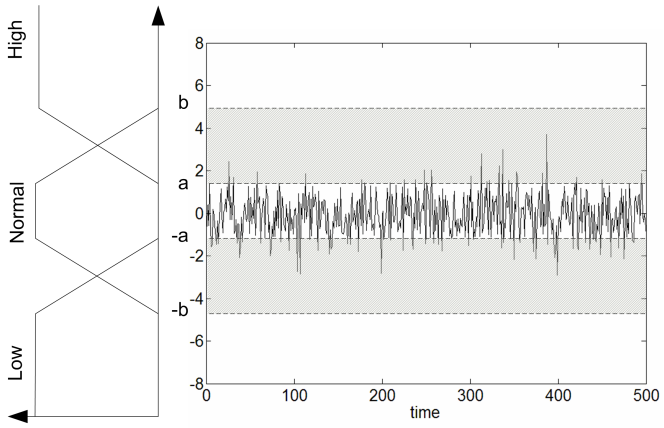


Figure 7.2: Fuzzy space partitioning.

mean and unit variance during the normal operating conditions. The design of this fuzzy system, including the automatic rule extraction is presented in the following subsections.

7.5.1.1 Input space partitioning

In this work, standard trapezoidal membership functions, were used,

$$A(x) = \text{trap}(x; d, c, a, b) = \begin{cases} 0, & x \leq d \\ \frac{x-a}{b-a}, & d \leq x \leq c \\ 1, & c \leq x \leq a \\ \frac{d-x}{d-c}, & a \leq x \leq b \\ 0, & b \leq x \end{cases} \tag{7.9}$$

The input-space partitioning is made after the normalization of the inputs, therefore, all the variables can be treated in the same manner. For each input variable, three membership functions are defined, Normal, Low and High (Figure 7.2).

Normal membership functions are considered symmetric,

$$A_2(x) = \text{Normal} = \text{trap}(x; -b, -a, a, b) \tag{7.10}$$

and the High and Low membership functions are defined as,

$$A_1(x) = \text{Low} = \text{trap}(x; -c, -c, -b, -a) \tag{7.11}$$

$$A_3(x) = \text{High} = \text{trap}(x; a, b, c, c) \tag{7.12}$$

where c is the maximum range considered for the normalized inputs. Therefore, the completeness of the partitioning depends of two parameters, a and b that have an important influence on the fault diagnosis classification performance and in the interpretability of the results.

Here some guidelines for the selection of the parameters are given. When $a \to 0$, the membership function correspondent to the normal operation zone tends to be triangular

(see Figure 7.2), therefore any deviation from the normal operation state, including those corresponding to process or measurement noise are considered as less normal than the zero measurement. On the other hand, in the limit when $a \to c$ the process is considered as normal in any condition. As a consequence, a trade-off should be solved to make the system insensitive to noise, but sensitive to process disturbances. When $b >> a$, the decision area is very fuzzy and the system will consider more uncertainty in the process measurements. On the contrary, when $b \to a$, the fuzzy system tends to a standard crisp expert system, without uncertainty, which is more sensible to detect deviations, but can not deal with actual measurements and process uncertainty.

The selection of the membership function parameters, a and b, can generate changes in the membership function similarity. Because of the definition of the membership functions, the similarity between High and Low will be always null, showing now fuzzyness between the elements of this sets. On the other hand, assuming that the range of the normalized variable is bounded ($c < \infty$), the degree of similarity between Normal and High results (see Appendix B),

$$\mathbf{\Psi}(A_1, A_2) = \mathbf{\Psi}(A_2, A_3) = \frac{\frac{\Delta}{c}}{\frac{\Delta}{c} + 4\frac{a}{c} + 4} \tag{7.13}$$

That has a maximum of $\mathbf{\Psi}(A_1, A_2) = 0.2$ when $a = 0$ and $\Delta = 10$, and a minimum of $\mathbf{\Psi}(A_1, A_2) = 0$ when $\Delta = 0$, $\forall\ a < c$. This last case is equivalent to the non-fuzzy scenario. Note that with this similarity values, distinguishability is fulfilled for any pair (a, b).

7.5.1.2 Output space partitioning

The outputs of the system are the possibilities of the plant of being in a given predefined state S_j (including the normal state). The outputs are set between zero and one, which respectively represent null and total possibility of the occurrence of state S_j. Two membership functions are defined for each output,

$$B_1 = \text{Low} = \text{trap}(0, 0, a_0, b_0) \tag{7.14}$$

$$B_2 = \text{High} = \text{trap}(a_0, b_0, 1, 1) \tag{7.15}$$

where $a_0 = 1 - b_0$ and $0 < a_0 < 0.5 < b_0 < 1$.

7.5.2 Automatic rule extraction

The algorithm to automatically extract rules from process data is depicted in Table 7.1.
The association of process measurements with a process state S_j is made trough an *if-then* rule. For each state, it is attempted to extract one or more *if-then* rules with the form:

$$R_k : \text{If } x_1 \text{ is } A_{k1} \text{ and } x_2 \text{ is } A_{k2} \text{ and } \cdots x_n \text{ is } A_{kn} \text{ then } S_j \textit{ is } B_{kj} \tag{7.16}$$

where the membership functions $A_{kj} \in [A_1, A_2, A_3]$ and $B_{kj} \in [B_1, B_2]$ are those defined above and $x_1, \ldots, x_n$ are the FLS inputs.

Table 7.1: Rule extraction algorithm.

1. For each sampling interval z
2. For each state S_j
3. project data using PCA built with normal state data
4. For each input x_i to the fuzzy system (i.e. contributions and latent variables)
5. normalize
6. assign a linguistic value to the variable x_i in the form: x_i is A_k
7. End
8. Merge the antecedents obtained for each variable to extract the rule R_j that associate the linguistic values of the input variables with S_j, R_j: If x_1 is A_k and $\ldots x_n$ is A_k then S_j is B_2
9. If R_j is not in the rule base, add the rule to the rule base
10. End
11. End
12. Add two rules for fault detection
    R_{n1}: If x_1 is A_2 and $\ldots x_n$ is A_2 then S_J is B_2 and S_j is B_1
    for all $j = 1, \ldots, J-1$
    R_{n2}: If x_1 is not A_2 or $\ldots x_n$ is not A_2 then S_J is B_1

7.5.2.1 Assignment of linguistic values

The assignment of a linguistic value A_k to each input variable x_i during a given state S_j is presented on the example depicted in Figure 7.3. After the PCA detection (indicated as a vertical line at time 250), a set of samplings is taken on the signal (e.g. at time = 50, 100, and 150 after detection). The membership degree of each sampling to the sets Low, High and Normal is evaluated using the predefined membership functions.

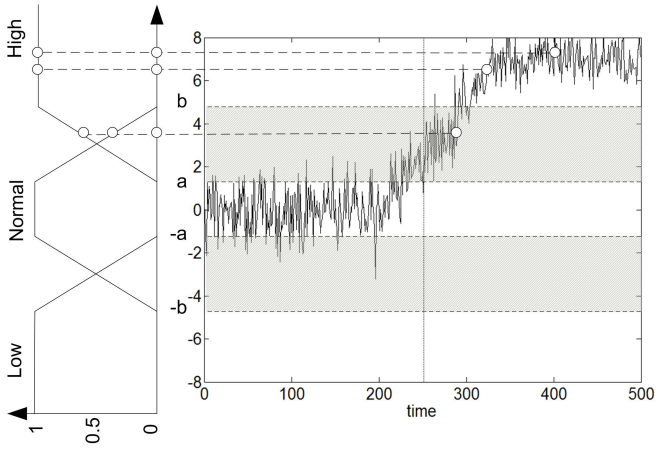


Figure 7.3: Assignment of a linguistic value for a variable during an abnormal state.

Next, the sampling values are grouped in vectors corresponding to each membership function,

$$\begin{aligned} V_{Low} &= [0, 0, 0] \\ V_{Normal} &= [0.4, 0, 0] \\ V_{High} &= [0.6, 1, 1] \end{aligned} \tag{7.17}$$

Then, the average of each of these vectors is evaluated as,

$$\begin{aligned} \mu_{Low} &= \text{avg}(V_{Low}) = 0 \\ \mu_{Normal} &= \text{avg}(V_{Normal}) = 0.13 \\ \mu_{High} &= \text{avg}(V_{High}) = 0.87 \end{aligned} \tag{7.18}$$

Finally, if one of the averages exceeds a predefined limit (say $L_{min} = 0.8$), the variable is assigned to the corresponding set, in this case: x_i is High. If the condition is not reached for any average, no value is assigned for this variable during this state, which is equivalent to consider that this variable (x_i) will not be used to infer the presence of the state S_j.

7.5.2.2 Merging antecedents

The second part of the algorithm consists of merging the extracted antecedents (i.e. x_i is A_k) to build a complete rule (R_j) that will be used to identify the state S_j,

$$R_j: \text{ If } x_1 \text{ is } A_k \text{ and } \ldots \; x_n \text{ is } A_k \text{ then } S_j \text{ is } B_2 \tag{7.19}$$

This algorithm is used to extract one rule to identify each abnormal process state. Finally, two rules are added to perform fault detection. The first one states that if all the variables are normal then process is in normal state and the second says that if one of the variables is not normal, then process is in an abnormal state (see Table 7.1).

7.5.2.3 Setting the number of sampling intervals

To account for temporal changes in the variables trends during process faulty behaviors, the automatic rule extraction algorithm allows defining several sampling intervals (Z) and different number of samplings in each interval. The number of sampling intervals divides the signals domain in different temporal regions whereas the number of samplings in each interval define the number of evaluations required to assign a linguistic value to each variable in each interval. The obtained rules are highly dependent on the selected parameters values (the number of intervals and the number of samplings in each interval), therefore, both will have a great influence on the diagnosis performance. Several rules will be extracted from each interval, by this way, these new rules are only introduced at the rule-base when they are not redundant.

7.6 Performance optimization

7.6.1 Genetic algorithm approach

A genetic algorithm (GA) was used to optimize the FLS performance. In this optimization problem there is not specific knowledge domain, so an evolutionary algorithm was adopted. The GA was chosen as a simple and straightforward efficient searching approach. In any case, several evolutionary algorithms such as Evolutionary programming or genetic programming could have been also adopted to solve this problem. Further analysis might be done in that direction to check the best option in each specific case.

GA does not guarantee the convergence to global optima. However, it searches a population of points in parallel, not a single point, and therefore has a reduced chance of converging to local optima. It should be pointed out that the optimization can be in detriment of the classification interpretability. Therefore, to maintain the transparency, the structure of the system is fixed; the number of inputs, outputs, the number and name of the membership functions remain unchanged.

The parameters a_i and b_i which define the shape, range and limits (thresholds) of trapezoidal fuzzy sets for each variable i were chosen as the decision variables. Thus, after optimization, each variable has two associated parameters (a_i and b_i) that define the new associated set of optimized membership functions (i.e.High_i, Normal_i and Low_i).

7.6.1.1 Coding

One of the main steps in genetic algorithm optimization is the coding of the solution. In this work, each individual represents a specific partitioning of the input space, in other words, each individual ($\mathbf{q}$) includes the parameters to define the membership functions correspondent to each input.

As explained in section 7.5, two parameters (a_i and b_i) are enough to define the membership functions of each input. The individual, is therefore a double-parameter string, the first representing the membership function *shoulder* (a_i) and the second representing the *foot* (called $\Delta = b_i - a_i$),

$$\mathbf{q} = a_1,\ a_2, \ldots, a_i, \ldots, a_n,\ \Delta_1,\ \Delta_2, \ldots, \Delta_i, \ldots, \Delta_n \tag{7.20}$$

where the parameters have been arranged to facilitate crossover.

7.6.1.2 Fitness function

The fitness of each individual musts represent the goodness and effectiveness of the obtained fuzzy system to perform fault diagnosis.

The performance to detect and isolate the fault j is calculated by means of the following objective function,

$$\phi_j = \int_{t_{d_j}}^{t_f} F_j(t) \cdot \sum_{\substack{i=1 \\ i \neq j}}^{J+1} (1 - F_i(t)) \cdot w_j(t)\, dt \tag{7.21}$$

where F_j and F_i are the outputs of the FLS corresponding to the possibility of occurrence of the state S_j and S_i respectively. The first factor within the integral considers the successful diagnosis of the actual state while the second factor evaluates the absence of erroneous diagnosis. Finally, $w_j(t)$ (Eq. 7.22) is a weight function that contains two parameters, α_j that allows to introduce relative weights to give priority to the isolation of one fault over others, and τ_j that is introduced as an exponential *urgency* in the detection of the state S_j. Introducing this exponential weight function into the general objective function the importance of an early diagnosis is taken into account.

$$w_j(t) = \alpha_j \cdot e^{\frac{-(t - t_{d_j})}{\tau_j}} \tag{7.22}$$

α is the y-axis value in origin and τ defines the exponential function tangent slope in x-axis origin. In that sense, as higher the τ value, higher the importance of an early diagnosis.

Finally, the general fitness function(Φ) that quantitatively shows the global diagnosis system performance, is obtained as the sum of all the considered states performances,

$$\Phi = \sum_{j=1}^{J+1} \phi_j \tag{7.23}$$

7.6.1.3 Crossover

The recombination methodology chosen to generate a new population performs intermediate recombination between pairs of individuals at the current population. Then, it returns a new one, after ordered mating each row with the next even row. The last row is not mated and added at the end of the new population.

This methodology combines parent values using the following rule:

$$\text{Offspring} = \text{Parent1} + \alpha \cdot (\text{Parent2} - \text{Parent1}) \tag{7.24}$$

where α is a scaling factor chosen uniformly at random in the interval $[-0.25, 1.25]$. A new α is generated for each pair of parents to be combined. Intermediate recombination can generate any point within an hypercube slightly larger than that defined by the parents.

7.7 Case Studies

Two significant case studies were considered to demonstrate the proposed methodology. They were solved using an AMD XP2500 processor with 512MB RAM running on Microsoft Windows XP.

7.7.1 Debutanizer column

The debutanizer column (Figure 7.4) provided by Aspentech at its World Wide Site documentation (http://support.aspentech.com) was simulated in Hysys.Plant to show the proposed methodology in a valuable illustrative example. It is a 15 stage-multicomponent distillation column fed by two input streams consisting of a mixture of light hydrocarbons. The process includes 10 measured variables ($n = 10$). Six step and a slow drift faults were simulated (see Table 7.2) to check the FDS capabilities.

Most difficulties were expected at the isolation of faults F_3 and F_4, since the same variables are biased in both cases. The whole set of data, collected from simulation of each fault (Table 7.2) throughout 10 hours, was randomly sampled and split in training and testing sets. Training data sets (F_1 to F_6) were used to develop the FDS. Testing sets were kept back for the later FDS testing and performance evaluation.

Additionally, F_7 was tested to check the FDS response when diagnosing new faults that have not been considered during the FDS design. In that sense, a subtle step fall in Feed 2 flow-rate was chosen to check the sensibility of the FDS, and the additional information

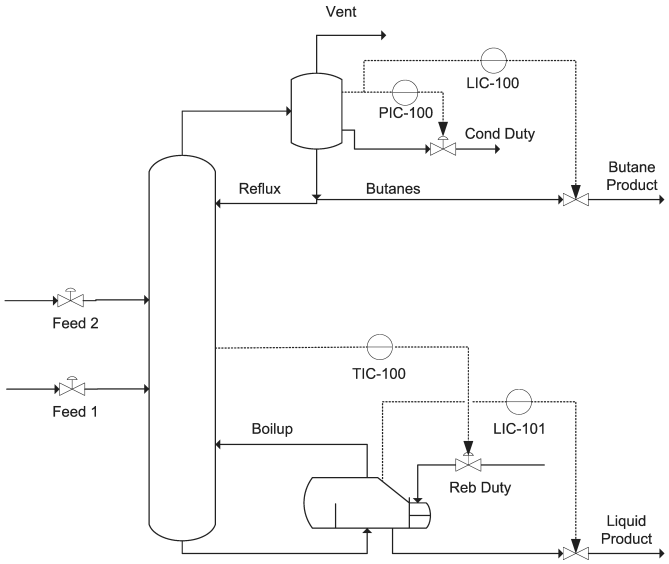


Figure 7.4: Case Study I: Debutanizer column flowsheet.

Table 7.2: Case Study I: Fault states definition.

Fault	Main variable involved	Deviation	Type
F1	Feed 1 flowrate	−20%	step
F2	Feed 1 flowrate	+20%	step
F3	Feed 1 temperature	+5%	step
F4	Feed 1 temperature	$+0.1^oC/min$	drift
F5	Feed 2 temperature	+5%	step
F6	Feed 2 flowrate	+20%	step
F7	Feed 2 flowrate	−5%	step

that the FDS is able to offer to operators when it faces a new abnormal event.

7.7.1.1 Monitoring and diagnosis without optimization

Firstly, a PCA model was developed using normal state data (see Figure 7.1). Seven principal components were selected by parallel analysis (see reference Himes et al. (1994)). The later on-line use of this module allowed to detect abnormal operation under the occurrence of all considered faults (Table 7.2) by monitoring SPE and T^2 statistics from the PCA detection module.

Afterwards, the isolation module was designed without applying the optimization methodology, choosing $a_i = 2$ and $b_i = 5$ for all the input variables, and $a_0 = 0.2$ and $b_0 = 0.8$ for the output variables. Then, contribution of each variable to SPE and latent scores variables from PCA were used off-line as input variables to characterize each fault in form of rules. The set of rules was obtained using the presented rule extraction algorithm (section 7.5). Three intervals (Table 7.3) with 10 samplings for each interval were considered to check the basic FDS performance.

Table 7.3: Sampling intervals. The initial time of the first interval corresponds to the detection time.

interval	initial time [min]	end time [min]
1	50	133
2	134	216
3	217	300

An objective function value of $\Phi = 7.185 \cdot 10^4$ (without considering $w_j(t)$) was obtained. Once the set of rules is completed it is included on the isolation module for its on-line use to diagnose the process satate. The results of the complete FDS diagnosing those faults previously trained are presented in Figure 7.5. Each figure depicts the possibility of occurrence of any of the considered fault at each sampling time, including normal state which is a fault detection measurement that complements the PCA statisticals. Trace line, corresponding to the process normal state, falls when the abnormal state arises at time 50 (each sampling corresponds to 2 simulation minutes). Thus, all the considered faults were satisfactorily detected not only by PCA statistics but also by the isolation module itself. Notice how bold line, corresponding to the actual occurring fault rises at the same time that trace line falls in each figure, outlining the most probable plant state. Note that the number of lines representing the rest of considered states in each figure, appears to be less than the number of predefined states (6 faults plus the normal state) since some of them are overlapped.

As a first conclusion, it can be stated that despite the overall FDS performance is acceptable, there were misclassification problems in faults F_4 (Figure 7.5d) and F_5 (Figure 7.5e). To overcome this diagnosis difficulties, the optimization algorithm was applied to better adjust the membership functions.

7.7.1.2 Monitoring and diagnosis with optimization

Details of the optimization parameters are given in Table 7.4. The algorithm was run several times using a different number of sampling intervals from 1 to 10, selecting in each case the best results from different GA runs. It allows to check the sampling influence on the extracted rules and the diagnosis performance removing the GA performance variability. The required time linearly increased with regard to the number of intervals considered.

Table 7.4: Case Study I: Optimization algorithm parameters.

Number of intervals tested	from 1 to 10
N_{ind}	20
N_G	200
G	15
selection	Roulette wheel
recombination	Intermediate recombination
P_m	0.035
Range of a	$[0, 5]$
Range of Δ	$[0, 8]$
individual length	$n + A = 17$

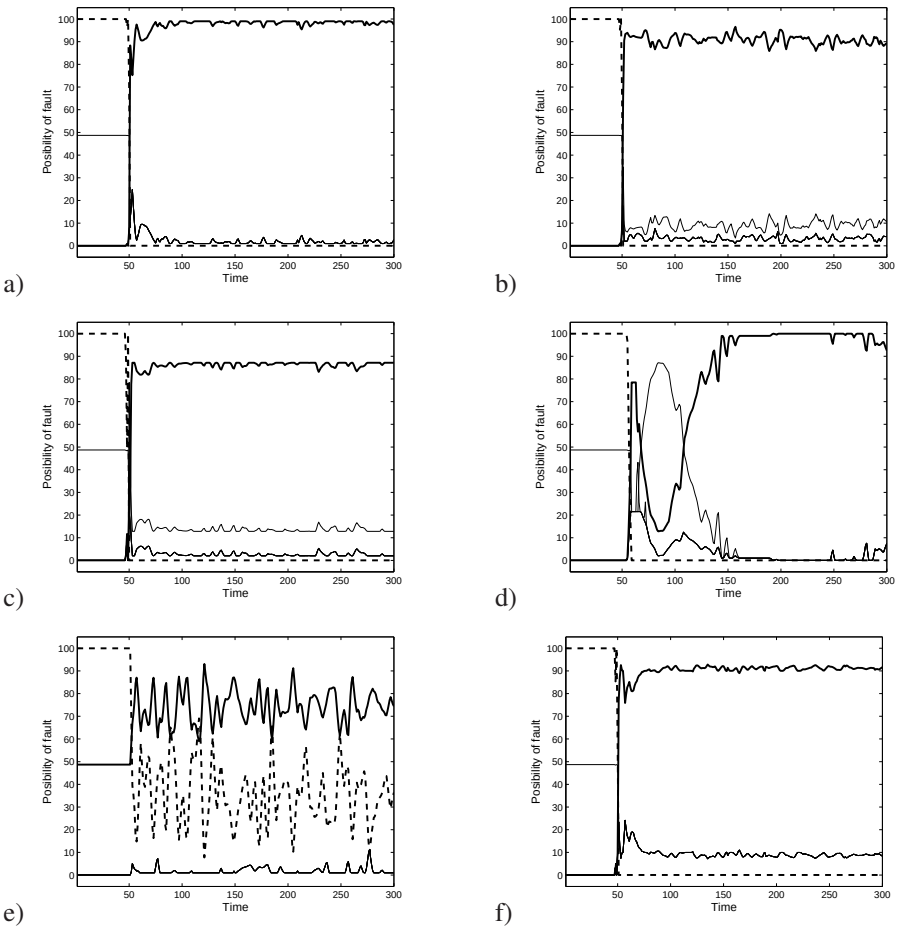


Figure 7.5: Case Study I: Fault isolation results: Process states classification a) F_1, b) F_2, c) F_3, d) F_4, e) F_5 and f) F_6, with $a_i = 2$ and $b_i = 5$. The trace line corresponds to the normal state identification. Bold lines correspond to the actual abnormal state (F_1, F_2, F_3, F_4, F_5 and F_6 respectively). The remaining lines correspond to the other considered states.

Figure 7.6 shows that the number of rules extracted increases as the number of sampling intervals does it. However, if more than 8 intervals are used, new rules are redundant and therefore are removed by the rule extraction algorithm. As a consequence, the number of rules stops increasing.

Moreover, it should be stressed that the isolation performance (objective function represented by the trace line in Figure 7.6) does not proportionally increase with regard to the number of rules. In that sense, as more sampling intervals are considered more details are captured from signals. By this manner, noise is being captured when the selected sampling intervals are more than three, so they are harming the FDS performance. Therefore, isolation performance reaches the maximum when three intervals are sampled and nine rules are used.

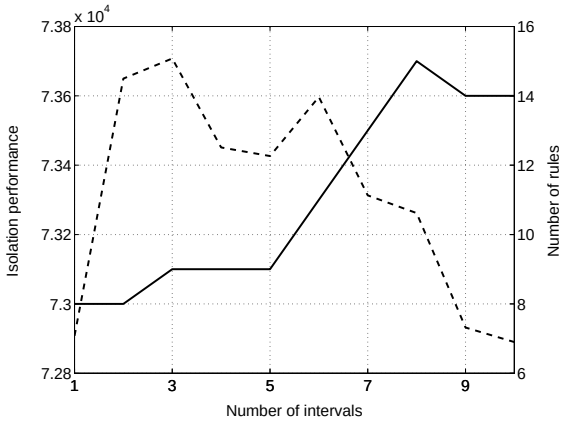


Figure 7.6: Case Study I: Optimized performance (trace line) and number of rules (bold line) vs. the number of intervals used to obtain the rules.

The algorithm had been executed six times at each sampling interval allowing to check the optimization goodness. The evolution of the fitness function (Φ) along the generations is depicted in Figure 7.7 showing good convergence for any of the tested runs. The best individual (i.e., who achieves the maximum fitness, $\Phi = 7.371 \cdot 10^4$) is presented in Table 7.5. Note that this value corresponds with the highest peak observed on the Isolation Performance curve at Figure 7.6.

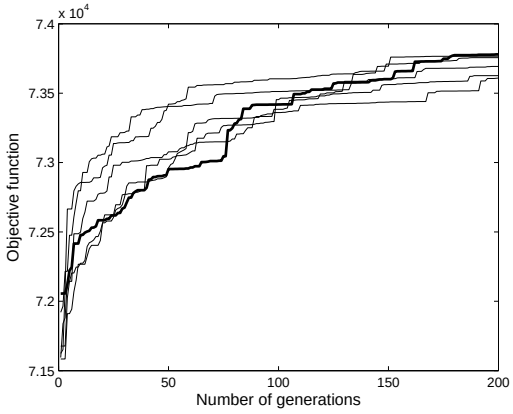


Figure 7.7: Case Study I: Optimization profiles for a fixed number of intervals. The bold line reaches the highest Objective Function value.

Table 7.5: Case Study I: Optimized fuzzy parameters.

	Latent Variables							Residuals								
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	Δx_1	Δx_2	Δx_3	Δx_4	Δx_5	Δx_6	Δx_7	Δx_8	Δx_9	Δx_{10}
a	2.77	1.75	5.00	2.07	2.22	4.30	3.17	3.61	4.36	2.08	3.40	3.24	4.10	4.84	2.72	2.67	3.03
Δ	0.00	0.86	4.12	0.00	0.23	3.32	0.99	0.46	2.13	0.81	6.71	0.37	0.28	4.17	0.29	0.17	0.46

Analyzing these individual variables it is possible to know which one behaves in a more or less fuzzy way (i.e., high Δ values implies high uncertainty in variable values), hence obtaining additional process knowledge. Results corresponding to this individual are presented in Figure 7.8.

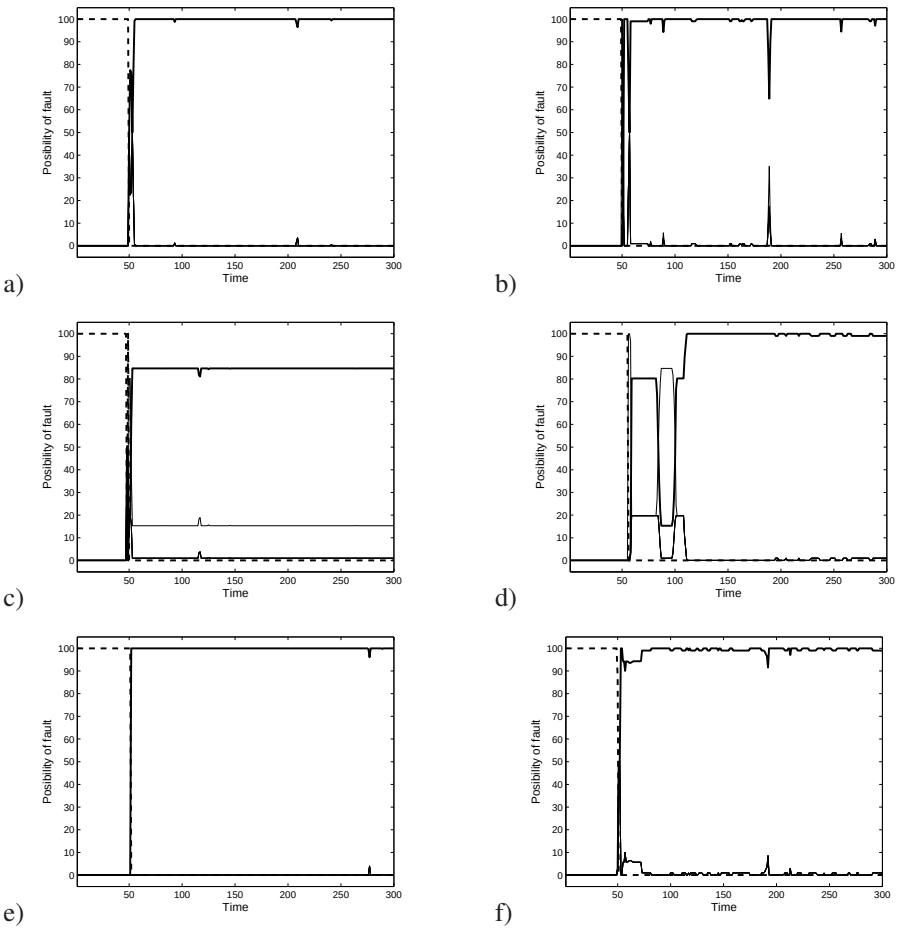


Figure 7.8: Case Study I: Fault isolation results after optimization: Process states classification a) F_1, b) F_2, c) F_3, d) F_4, e) F_5 and f) F_6, using the best solution. The trace line corresponds to the normal state identification. Bold lines correspond to the actual abnormal state (F_1, F_2, F_3, F_4, F_5 and F_6 respectively). The remaining lines correspond to the other considered states.

Isolation performance after optimization of the membership functions is better than the initial one, where no optimization was applied (compare Figure 7.8 and Figure 7.5). Noteworthy improvements were specially attained isolating F_5, which resolution problems were completely solved.

The FDS still presents misclassification problems diagnosing F_4. F_3 is still diagnosed under the occurrence of F_4 during several samplings (7.8d). Nevertheless, F_4 early fault diagnosis was solved (check Figure 7.8d) and misclassification was clearly reduced. More-

over, concerning on that fault was relative, as F_3 and F_4 faults have the same origin and even they can be considered as the same fault. Note that early diagnosis improvements were partly achieved by the inclusion of the exponential term in the objective function (subsection 7.6.1.2).

Transparency of the proposed FDS can be checked by reviewing the explanation facilities offered by the rules base. Table 7.6 shows the set of rules which conforms the FDS for that case study. Table 7.6 illustrates how each fault diagnosed is supported by a small number of comprehensive rules which offer a highly appreciated linguistic information to operators about variables behavior. Note that, L, N, H, and I correspond to abbreviations of Low, Normal, High and Irrelevant.

Table 7.6: Case Study I: Rules extracted by the automatic procedure (described in section 7.5) for 3 sampling intervals, corresponding to the best individual obtained after optimization.

Rule	Latent Variables								Residuals										States							Type
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	9	10	F1	F2	F3	F4	F5	F6	N	
1	L	L	L	H	N	H	N	L	N	N	N	N	N	N	N	N	N	N	H	L	L	L	L	L	I	and
2	H	H	H	L	N	L	L	H	N	N	N	N	N	N	N	N	N	N	L	H	L	L	L	L	I	and
3	H	N	H	L	L	L	L	H	L	H	H	N	N	L	I	H	N	N	L	L	H	L	L	L	I	and
4	H	I	H	L	L	L	L	H	L	H	H	N	H	L	L	H	H	H	L	L	L	H	L	L	I	and
5	N	N	N	L	N	N	N	N	N	N	N	N	N	N	N	N	N	N	L	L	L	L	H	L	I	and
6	N	N	N	H	H	H	H	N	H	N	L	N	N	H	N	L	N	N	L	L	L	L	L	H	I	and
7	N	I	N	H	H	H	H	N	H	N	L	N	N	H	N	L	N	N	L	L	L	L	L	H	I	and
8	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	L	L	L	L	L	L	H	and
9	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	Ñ	I	I	I	I	I	I	L	or

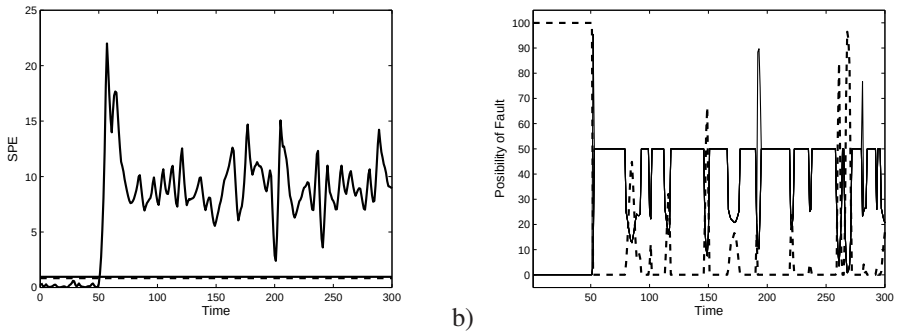
Moreover, this FDS architecture allows the operators to introduce new rules manually from their experience making possible the human-system interactivity. This human-machine link also supports the system comprehensibility and transparency which positively influence the FDS operator's trust and the consequent application of the technique.

Finally, the FDS behavior was tested under a novel fault that was not included during the rule extraction or FDS design stage (F_7 in Table 7.2). Once the fault was simulated it was rapidly detected by the SPE (see Figure 7.9a) and the T^2 plots. Also, it was clearly detected by checking the sudden fall of the normal state occurrence possibility on its corresponding diagnosis plot (see Figure 7.9b). Evidently, any of the known states were clearly identified, showing that the process was under a new unknown state. However, some invaluable information can be also collected from this wide and flexible FDS. PCA is not just able to detect the fault but also to provide information about which are the most biased variables (Vedam and Venkatasubramanian, 1999), thus helping operators to know the source causes of the occurring fault.

7.7.2 Tennessee Eastman process

The proposed methodology was also validated in the Tennessee Eastman process (Downs and Vogel (1993)), described at subsection 4.6.2. The process flow-sheet is depicted in Figure 4.16.

The 15 first faults from those originally proposed by Downs and Vogel (1993) (see Table 7.7) were considered to assess the FDS performance. Training and testing data sets were randomly and independently collected from 20 h simulations for each considered fault.



a) b)

Figure 7.9: Case Study I: a) SPE plot under simulation of fault 7, b) diagnosis response under fault 7. Trace line represent normal state.

All those standard deviation variables that showed improvements on the problem representation (sub-section 6.4.2.5) were included as input variables to the FDS.

Table 7.7: Process faults given at the Tennessee Eastman original paper (Downs and Vogel, 1993).

Fault	Process variable	Fault	Process variable
IDV(1)	A/C Feed ratio	IDV(11)	Reactor cooling water inlet temperature
IDV(2)	B Composition	IDV(12)	Condenser cooling water inlet temperature
IDV(3)	D Feed temperature	IDV(13)	Reaction kinetics
IDV(4)	Reactor cooling water inlet temperature	IDV(14)	Reactor cooling water valve
IDV(5)	Condenser cooling water inlet temperature	IDV(15)	Condenser cooling water valve
IDV(6)	A feed loss	IDV(16)	Unknown
IDV(7)	C header pressure loss	IDV(17)	Unknown
IDV(8)	A, B, C feed composition	IDV(18)	Unknown
IDV(9)	D feed temperature step	IDV(19)	Unknown
IDV(10)	D feed temperature	IDV(20)	Unknown

The sampling space, corresponding to 20 h of process simulation, was divided in five different ways. At more considered intervals more samplings made to extract the set of rules. Number of samplings used for each space division are shown together with other interesting results on Table 7.10.

In this industrial benchmark case, the algorithm was applied in two different ways: 1) Firstly, no optimization was carried out in order to show how the FDS offers a good initial diagnosis and detection performance. 2) Then, the FLS was optimized following the methodology described in this chapter in the same way as it was shown in the first illustrative case study.

7.7.2.1 Without optimization

Parameters selected for the input and output membership functions were fixed to $a_i = 3$ and $b_i = 5$ for all the input variables, and $a_0 = 0.2$ and $b_0 = 0.8$ for the output variables. The limit for the average in the set assignment was set up at $L_{min} = 0.9$.

Firstly, standard PCA was applied to perform fault detection (see Table 7.8). Most of simulated faults were clearly detected by both statistics.

Table 7.8: PCA detection reliability for the Tennessee Eastman disturbances (in percentage).

IDV	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SPE	98.9	98.0	3.7	99.9	10.6	100	99.9	97.5	17.3	96.6	99.2	96.8	83.6	99.3	1.8
T^2	98.8	97.0	0.0	63.9	3.0	100	99.8	97.0	22.1	29.1	99.1	87.8	82.8	98.2	0.5

Then, the FDS was firstly applied to diagnose the most clearly detected disturbances (i.e. IDV(1), IDV(2), IDV(4) or IDV(6)-IDV(7)) and the normal state. FDS results obtained diagnosing some of these well detected states are depicted in Figure 7.10. Afterwards, the FDS was tested diagnosing the rest of faults, that is, those that were not clearly detected by PCA (IDV(3), IDV(5), IDV(9) and IDV(15)) (see Table 7.8).

Good performance was obtained when diagnosing any of the well detected faults and confuse diagnosis responses were given when IDV(3), IDV(5), IDV(9) and IDV(15) were simulated. However, some additional detection capabilities can be appreciated when the completed FDS faces these difficult faults. For instance, although IDV(5) caused very subtle deviations in monitored variables, its detection was supported by the diagnosis module response (see Figure 7.11). The trace line on Figure 7.11 (corresponding to the normal state) deeply decreased several times during the fault simulation. These results demonstrate how the isolation module may also support fault detection in addition to its basic diagnosis capabilities.

7.7.2.2 With optimization

In order to improve the diagnosis reliability, particularly for IDV(3), IDV(5), IDV(9) and IDV(15), the optimization procedure was applied. The optimization parameters for the new case are summarized in Table 7.9.

Table 7.9: Case Study II: Optimization algorithm parameters.

Number of intervals tested	from 1 to 5
N_{ind}	10
N_G	200
G	50
selection	Roulette wheel
recombination	Intermediate recombination
P_m	0.031
Range of a	$[0, 3]$
Range of Δ	$[0, 8]$
individual length	$n + A = 110$

The overall fitness function obtained for each sampling interval, their corresponding calculation time and the number of extracted rules for each interval are presented in Table 7.10. Observe the linear increment of samplings used at each interval. As more divided is the sample space more samplings were accounted and more time was required to design

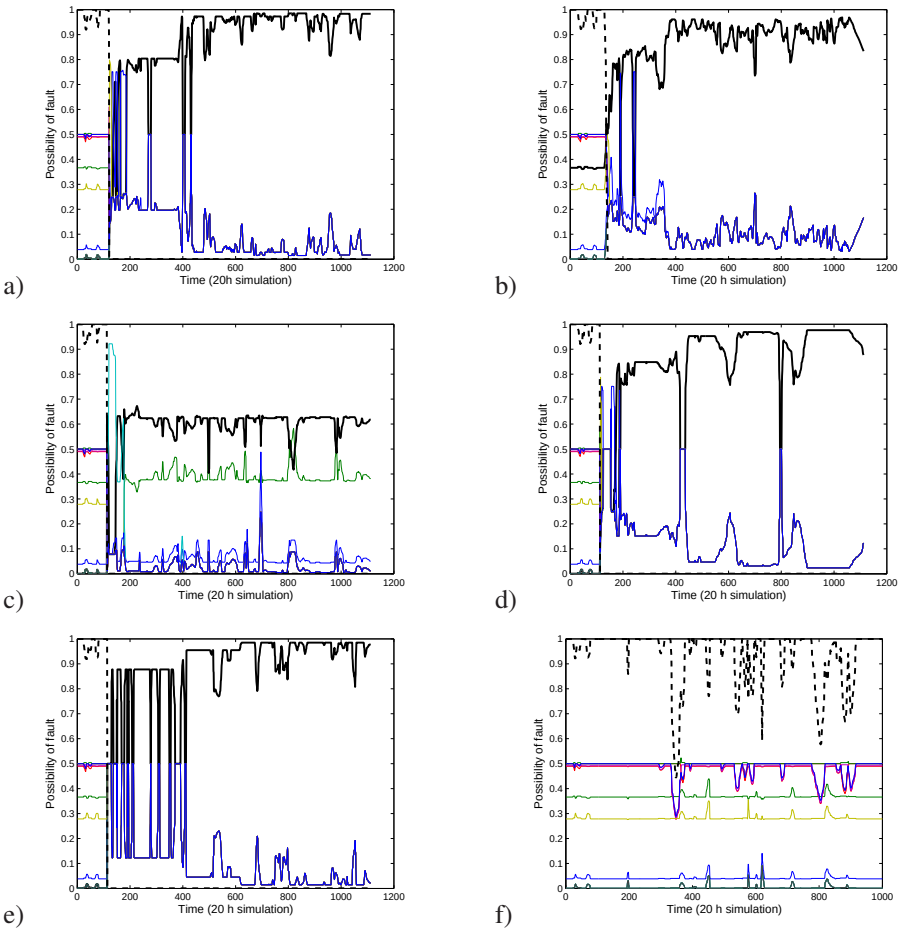


Figure 7.10: Case Study II. Fault isolation results. Disturbances and normal state classification a) IDV(1), b) IDV(2), c) IDV(4), d) IDV(6), e) IDV(7) and f) normal state, with $a_i = 3$ and $b_i = 5$. The trace line corresponds to the normal state identification. Bold lines correspond to the actual abnormal state (IDV(1), IDV(2), IDV(4), IDV(6) and IDV(7) respectively). The remaining lines correspond to the other considered states.

and evaluate the FDS.

Table 7.10: Sampling intervals selection in Teneessee Eastman Optimization.

Number of Intervals (Z)	1	2	3	4	5
Objective Function (OF)	3262.6	3683.2	4137.3	4840.3	4679.9
Calculation Time, h	22.3	28.6	36.1	46.5	57.2
Number of samplings	10	20	30	40	50
Number of rules	17	29	42	56	69

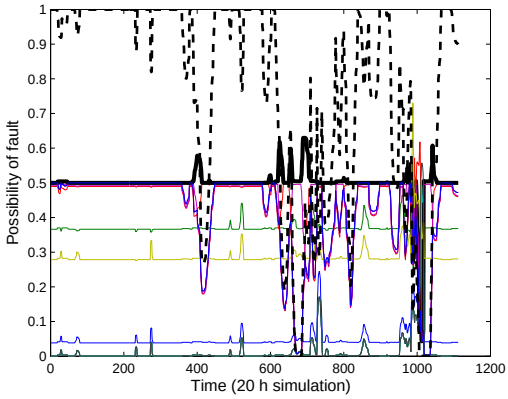


Figure 7.11: Case Study II. Fault diagnosis signals. The trace line corresponds to the normal state detection and the bold line corresponds to IDV(5) fault possibilities. The remaining lines correspond to the other considered states.

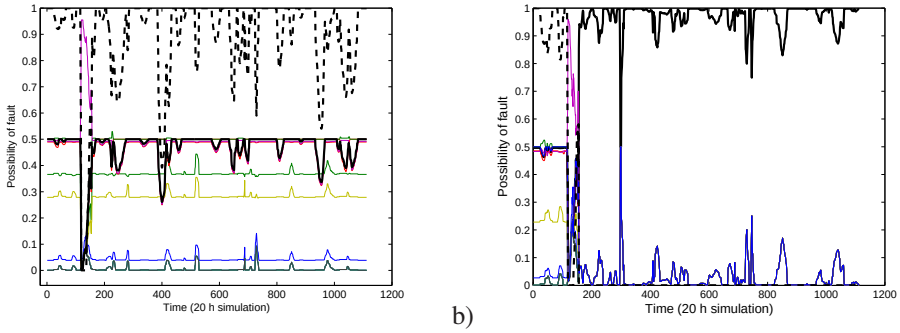
The objective function allowed to choose the best way of sampling, which in this case was Z = 4. For comparison purposes, the fitness function was calculated for the initial (a_i = 3 and b_i = 5) and the new optimized individual using the best sampling intervals (Z = 4). Table 7.11 shows that most of the states are better isolated by the new optimized FDS. Note that, higher fitness function values were achieved in the most conflictive cases. The best diagnosis performance was confirmed by including more reliable comparative measurements as Recall, Precision or the index $F1$ given at section 5.3 (see Table 7.11).

Table 7.11: Individual and global fitness function before and after Tennessee Eastman optimization (Z=4).

IDV	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Normal	Avg.
Initial																	
OF	176	259	115	192	65	169	291	134	115	481	464	435	43.7	194	114	656	3906
Recall	94.6	96.5	2.0	93.8	0.0	90.8	90.9	72.7	11.9	97.6	92.4	86.1	61.1	65.0	0.0	98.9	65.9
Precision	66.1	97.5	14.1	97.3	0.0	95.8	100	73.4	100	98.3	70.0	95.8	89.3	98.8	0.0	19.9	69.8
$F1$	77.8	97.0	3.5	95.5	0.0	93.2	95.2	73.1	21.3	97.9	80.3	90.7	72.6	78.4	0.0	33.1	63.1
Optimized																	
OF	232	301	125	394	408	117	303	183	125	502	421	442	36.5	487	127	638	4840
Recall	96.9	88.6	5.7	95.8	95.4	84.0	85.5	68.1	17.2	91.8	83.9	84.3	52.8	96.8	0.0	91.9	71.2
Precision	52.8	100	20.3	96.7	100	99.6	100	59.6	70.2	100	96.3	91.3	87.8	99.0	0.0	25.0	74.9
$F1$	68.4	93.9	8.9	96.2	97.6	91.2	92.2	63.5	27.7	95.7	89.7	87.7	66.0	97.9	0.0	39.3	69.7

Note that optimization of parameters a and Δ allowed an important improvement on all the tested comparative performance indices. Note also that Recall or Precision values for two different faults can be very similar under very different OF values. For example IDV(6) Recall is practically the same than IDV(7) Recall (90.8 and 90.9 respectively on Table 7.11). This is due to Recall and the employed OF are measuring different quantitative details. OF focus on quantitative distances between actual and the rest of fault possibilities while the other indices are evaluated on the basis of correct or wrong diagnosis by a less detailed quantitative analysis.

Furthermore, a key improvement was achieved on the detection and diagnosis of IDV(5). While the system was unable to detect (Table 7.8) or diagnose the fault using the original membership functions, the optimized individual completely addressed the diagnostics and also the detection of the event (Figure 7.12).



a) b)

Figure 7.12: Case Study II: Diagnosis of IDV5 a) without optimization b) and using the final optimized individual (Table 7.10).

7.8 Conclusions

An integrated fault detection and isolation framework was developed as an additional and complementary FDS. It uses PCA to perform fault detection, and to pre-process plant measurements. Then a fuzzy logic expert system which extract relevant diagnosing rules is used to perform fault isolation and to support PCA fault detection.

An automatic procedure was developed to create the set of diagnosing rules used by the fuzzy logic system. Moreover, an optimization method was proposed to improve the isolation performance without losing comprehensiveness. It focuses on fixing the best sampling way as well as the membership functions shapes. The framework was tested in an academic case study and in the TEP showing very promising and effective diagnosis results. Moreover, the system showed very efficient interaction and transparency capabilities with the process operator, which may mean a key point at the time the operators make practical use of the system.

Nomenclature

a	*Shoulder* of the trapezoidal membership function
A	Input fuzzy sets
b	Trapezoidal membership function parameter
B	Output fuzzy sets
c	Normalized FLKB bound value
D_λ	Eigenvalues diagonal matrix
F	Fault
G	Number of generations to stop the GA optimization when no improvements are appreciated
J	Number of process states
I	Number of variables taken into account
L_{ind}	Individual length
LS	Latent scores
N_G	Number of generations
N_{ind}	Number of individuals in each generation
p	Eigenvectors
$\mathbf{P}$	Loading matrix
P_m	Mutation probability
q	Individual in the optimization procedure
$\mathbf{R}$	Data correlation matrix
R_k	*If-then* rules
S	Plant state
SPE	Squared predicted error statistic
t	Principal score vector
T_i	Control time constant
T^2	Hotteling's statistic
$\mathbf{x}'$	Normalized sampling vector
$\mathbf{X}$	Data matrix
$\overline{\mathbf{X}}$	Normalized data matrix
Z	Number of sampling intervals

Greek Characters

α	Priority in the weight function
Δ	*Foot* of the trapezoidal membership function
λ	Eigenvalues
μ	Membership function
ϕ	Objective function for each fault
Φ	General fitness function
Ψ	Similarity of fuzzy sets
τ	Urgency in the weight function

Subscripts

i	Variables
j	States
z	Sampling interval
k	Latent score

Acronyms

ANN	Artificial Neural Network
DVS	Dominant Variation Subspace
EWMA	Exponentially Weighted Moving Average
FDS	Fault Diagnosis System
FLS	Fuzzy Logic System
FLKB	Fuzzy Logic Knowledge-Based
KB	Knowledge-Based
MSPM	Multivariate Statistical Process Monitoring
OF	Objective Function
PCA	Principal Component Analysis
PCs	Principal Components
PLS	Principal Least Square
SVD	Singular Value Decomposition

CHAPTER 8

Diagnosis Conflict Solver Module

As it has been remarked in this thesis, different strategies and knowledge can be used to develop a FDS. These differences confer each system specific and unique abilities over the rest that can be exploited on the overall diagnosis benefit. Thus, FDSs complementarity may represent an important opportunity to obtain relevant synergies from an intelligent FDSs integration. The idea of improving the diagnosis performance by complementing individual systems was already pointed out by Venkatasubramanian et al. (2003a), or Mylaraswamy and Venkatasubramanian (1997). Why do we have to adopt a single FDS for the diagnosis of a complex problem? Why to resign ourselves to partially limited systems? For instance, lack of detection robustness of a FDS (e.g. ML-SVM system on the TE process) might be compensated by a supplementary and independent detection module (e.g. PCA module included on the PCA-FLS).

In this chapter, an integration of the ML-SVM diagnosis system and the PCA-FLS expert system is presented. Such strategy tries to take advantage of their complementarity to reach a more robust and reliable FDS. The underlying idea is that the weaknesses of one method (e.g. the lack of interactivity of ML-SVM diagnosis system) can be offset by the strength of the other (the interactivity of PCA-FLS). A practical implementation of both systems integration is given. This chapter shows complete design details and provides quantitative proofs of the theoretical and practical benefits obtained by the proposed integration methodology.

8.1 Introduction

Although the same knowledge basis (historical data) was used in the deployment of both FDSs proposed in this thesis (chapters 6 and 7), different properties of the applied strategies offered appropriate complementarity from which integration profits can be clearly attained. Such complementarity is evident when checking (Table 8.1) how both systems meet the main characteristics a FDS should offer for its success in an industrial environment (subsection 2.3.4).

Table 8.1: Qualitative comparison of ML-SVM and PCA-FLS diagnostic methods. *V*, *M* and *X* represent good, medium and bad capacities of the system respectively on the topics at each row.

Theoretical Diagnosis Criteria	**PCA-FLS**	**ML-SVM**
Explanation and reasoning facilities	*V*	*X*
Quick detection	*V*	*X*
Human-Machine Interaction	*V*	*X*
Modeling requirements	*V*	*V*
Multiple Faults	*X*	*V*
Required operators know how	*X*	*V*
Storage and Computational Requirements	*M*	*M*
Adaptability	*M*	*M*
Novelty identifiability	*V*	*X*
Classification error[1]	*V*	*V*
Robustness[2]	*V*	*M*

SVM has been reported as a very good classification technique in terms of generalization (Vapnik, 1998), one of the main problems when addressing "Multiple Faults" in complex diagnosis cases. These good generalization properties were also demonstrated at chapter 7 by means of training individual faults, then being able to properly identify complex faults consisting of simple faults combination. PCA-FLS is not as good at "Multiple Faults" since its current design based on a unique most probable state only allow to discern simultaneous faults from operator visual inferences that may lead to significant diagnosis errors.

The PCA-FLS modular frame provides good "Detection" given by the PCA module in one hand, while offer understandable information in form of rules to process operators by the diagnosis module. Human-system interactivity is favored by the chance of adding manual rules into the automatically-extracted rules set, which also contributes to the good comprehensibility and related system operator's trust. Nevertheless, "Comprehensibility" and "Quick Detection" are not so well accomplished by the ML-SVM system, since it is a black-box system which does not provide insights for diagnosis, thus harming the diagnosis system "Comprehensibility". Moreover, ML-SVM system requires of a complete good diagnosis performance for each considered fault in order to be able to assure reliable process fault detection.

Regarding "Robustness" and "Novelty Identifiability", PCA-FLS also offers better characteristics as it includes explicitly fuzzy logic to deal with the process variables uncertainty. Additionally, it incorporates basic analysis from PCA model (for instance, the variables contribution to the SPE) to predict most probable causes of novel abnormal situations. Although ML-SVM system also allows dealing with the uncertainty coming from noisy data by balancing parameter C, it does not address robustness in such a straight way. Neither is easy for this system to identify untrained faults, which would be classified as normal state by the system, that is, not belonging to any of the trained states. Thus, novel identification

[1] Associated confidence of the system to the given diagnosis response.
[2] Capability of reasoning under uncertainty and ability to justify that reasoning

by that system will require the training of new labeled data to predict the same state in the future.

"Adaptability", one the most important difficulties found at the time to extend academic diagnosis methodologies to industrial problems, is an equally challenging issue for both systems. Intense effort must be made to keep the system robustness when the plant is running away from its initial design conditions. In that sense, ML-SVM system may provide some advantages since its simple implementation allow easily adapting and training the classifiers with new available data, as well as easily training novel known states. On the other hand, PCA-FLS is flexible enough to allow the inclusion of new rules from human experience which may help to accomplish with the system updating requirements.

"Modeling Requirements" are low for both systems because of the automatic building procedures utilized to extract knowledge from historical data bases. This makes both systems easily implementable in new cases from the modeling viewpoint. Additionally, as both systems are optimized in some sense, intensive calculus is required for their designs, then medium "Storage and Computational Requirements" are needed in both cases. However, once they have being designed and installed the on-line diagnosis will be instantaneous, which make both systems applicable on real time.

Table 8.1 allows concluding that there are several points in which complementarity of both systems is promising. This promptly led to good expectations on both system integration success. In that sense, both diagnosis systems combination may, for example, take profit of the higher flexibility (by favoring the human interaction), comprehensibility and operator trust of the PCA-FLS and the better isolation capabilities (coming from a more rigorous quadratic optimization) or multiple faults management of the ML-SVM approach. Furthermore, diagnosis redundancy at each time will allow a more ample range from which to make diagnosis decisions, thus improving the overall diagnosis performance. By this manner, not only qualitative enhancements can be obtained from the use of complementary FDSs but also an increase on the quantitative isolation results can be expected. In order to check that point, quantitative analysis on the integrated FDS results are given at section 8.3.

Redundancy on the diagnosis response is not easy to manage and, although it supposes a great opportunity, must be dealt carefully to maintain clear diagnostics. Thus, the system must assure at each moment univocal diagnostics while increasing the accuracy and reliability of the overall system regarding their individual systems. At that point, the need of a diagnosis conflict solver (DCS) which prioritizes some responses over others (for instance, using relevant knowledge from each available FDS) becomes crucial. Therefore, a conflict solver module will be considered on the integration strategy to manage different diagnostics coming from each FDS.

8.2 Integration methodology

In order to take advantage of the diagnosis redundancy, a simple and fast DCS was considered. This system was designed to decide in each moment which of both diagnosis responses is the most reliable. The DCS was built by using a simple rule-based expert system which relying on each system diagnosis performance is able to offer a unique and

final diagnosis response.

The knowledge gathered from the testing and evaluation of each available FDS is the basis for the DCS decision-making. After checking the FDSs behavior, the reliability and accuracy of each FDS diagnosing each considered fault will be available. Such information will be invaluable at the time of deciding which diagnostic is more strongly supported. Reliability and accuracy are acquired from testing and evaluation of each FDS by means of $F1$ index introduced and used at chapters 5, 6 and 7 respectively. In this sense, the expert system will prioritize at each moment the diagnostic provided with a higher confidence. This is managed by checking which is the reliability of each FDS on the fault diagnosed. Figure 8.1 shows the general DCS information flow as well as its basic tools.

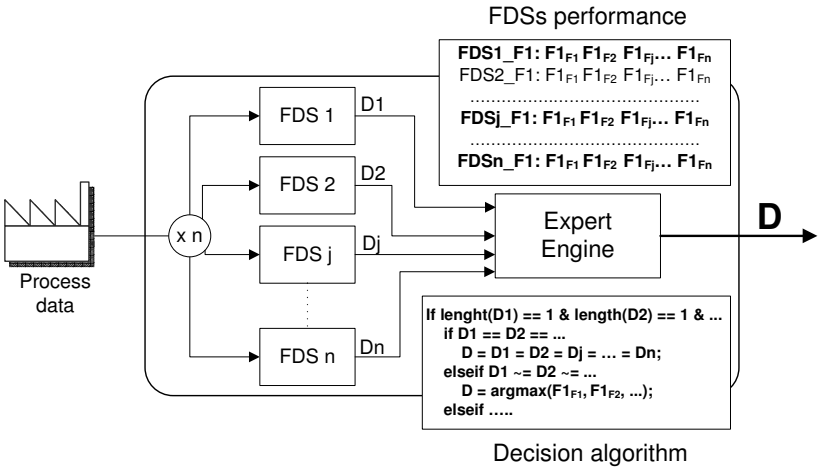


Figure 8.1: Diagnosis Conflict Solver (DCS) performance and information flow.

Firstly, process variable measurements are supplied to each available FDS by means of splitting the original signal. Thus, each FDS provides a diagnostic of the process state (D_1, D_j,..., D_n). Several FDSs can be considered within the global diagnosis framework. This work particularly focuses on the use of both FDSs presented in the thesis (described in chapters 6 and 7) but more FDSs can be considered extending the presented methodology (Figure 8.1). As shown on Figure 8.1, all diagnosis responses (two in our example) will be reduced by the expert system to a unique diagnosis response (**D**). Thus, the expert system consist of two main elements:

- The knowledge about each FDS performance ($F1$ indices table).
- The algorithm which infers the final diagnostic.

The FDSs performance can be obtained by testing and evaluating each FDS. The algorithm which decides a unique diagnostic at each sampling time (***D***) is conceived to manage equal or different diagnosis responses from each FDS. Thus, straight diagnostics will be inferred when individual diagnosis responses are equal and a more detailed reasoning will be used in case different diagnostics are provided by each sub-system. Besides, not only a diagnostic will be provided but also the reasoning and the information from which the

decision was taken will be made available to operators. These reliabilities (***R***) from which the diagnostics are reached will be mainly worthy in those cases whose diagnostics are not very clear.

First of all, $F1$ tables must be made available to the expert engine. Then, the expert system analyzes the diagnostic of each contemplated fault diagnosis sub-system. If these responses are univocal in each case, that is, both just give one clear most probable fault, the engine takes the most reliable based on the $F1$ tables available (check point 1 in Table 8.2). In case the sub-systems give several probable faults for the same sampling, then, each one of these faults are part of the multiple fault diagnostic. Thus, all possible combinations between faults sets diagnosis by each sub-system are accounted. To make the algorithm clearer, the rules previously described are depicted in Table 8.2. This set of rules contemplate the sub-system single and multiple diagnosis for two operating FDS to produce a final diagnosis response. In case more FDSs were included within the overall diagnosis frame, these if-then rules should be extended.

D represents the final diagnostic set given by the overall system, whereas **R** corresponds to the reliability or trust on the fault diagnosed. $F1_{D_i}$ and $F1_{D_j}$ are $F1$ indices corresponding to sub-systems i and j respectively on the current sampling diagnosing the respective state (D_i or D_j). Note that, n and m are the number of simultaneous faults diagnosed by the different FDSs.

8.3 PCA-FLS and ML-SVM integration results

In order to check the integration benefits in quantitative terms, ML-SVM and PCA-FLS were combined in a practical example. Once again the Tennessee Eastman Process (TEP) was selected to check the diagnosis performance. Assuming that only the ML-SVM system was tested under the occurrence of multiple faults, the integration was checked under the occurrence of single faults. However, multiple faults could be also considered by using parts 2, 3 and 4 of the DCS decision system (Table 8.2).

First the complementarity of both systems was verified. Such requirement consists of having systems that offer complementary results on different considered faults. This is, that a system is more efficient than the other diagnosing some of the tested faults and vice-versa. In order to check this requirement both systems were trained and tested under the same conditions. The TEP was tested using the same simulator under the occurrence of the same original 20 faults (Table 7.7 on chapter 7). Training and testing sets were created by random sampling of a large super set obtained by simulation. The initial super set includes simulation data of each fault plus data from the normal state during 20 h process operation by sampling every second. Hence, the super set consists of 21 sub-sets having 72000 samplings per each considered variable. A set of 104 variables was used in the design of each FDS. This set includes the 41 measured and 11 first manipulated TEP variables (XMEAS and XMV respectively) given at the original paper (Downs and Vogel, 1993), plus their corresponding standard deviation variables, built as shown in subsection 6.4.2.5. The training and testing data sets are available at http://ciao.euetib.upc.es[3]. They are used throughout the chapter to check the diagnosis performance of the PCA-FLS, the

[3] Also at http://webon.euetib.upc.es/ciao/recursos.html

Table 8.2: Expert engine conflicts resolution algorithm. Two FDSs and single and multiple sub-system diagnostics are contemplated.

1. If $m == 1$ and $n == 1$
   - If $D_i = D_j$
     - $\mathbf{D} = D_i = D_j$
     - $\mathbf{R} = max[F1_{D_i}, F1_{D_j}]$
   - Elseif $D_i \neq D_j$
     - $\mathbf{D} = argmax[F1_{D_i}, F1_{D_j}]$
     - $\mathbf{R} = max[F1_{D_i}, F1_{D_j}]$
   - end
2. Elseif $m > 1$ and $n == 1$
   - for $k = 1{:}m$
     - If $D_{k,i} = D_j$
       - $D_k = D_i = D_j$
       - $R_k = max[F1_{D_i}, F1_{D_j}]$
     - Elseif $D_{k,i} \neq D_j$
       - $D_k = [D_i, D_j]$
       - $R_k = [F1_{D_i}, F1_{D_j}]$
     - end
       - $\mathbf{D} = All \quad D_k, \quad k = 1, \ldots, m$
       - $\mathbf{R} = All \quad R_k, \quad k = 1, \ldots, m$
   - end
3. Elseif $m == 1$ and $n > 1$
   - for $p = 1{:}n$
     - If $D_i = D_{p,j}$
       - $D_p = D_i = D_j$
       - $R_p = max[F1_{D_i}, F1_{D_j}]$
     - Elseif $D_i \neq D_{p,j}$
       - $D_p = [D_i, D_j]$
       - $R_p = [F1_{D_i}, F1_{D_j}]$
     - end
       - $\mathbf{D} = All \quad D_p, \quad p = 1, \ldots, n$
       - $\mathbf{R} = All \quad R_p, \quad p = 1, \ldots, n$
   - end
4. Elseif $m > 1$ and $n > 1$
   - for $k = 1{:}m$
     - for $p = 1{:}n$
       - If $D_{k,i} = D_{p,j}$
         - $D_{k,p} = D_i = D_j$
         - $R_{k,p} = max[F1_{D_i}, F1_{D_j}]$
       - Elseif $D_{k,i} \neq D_{p,j}$
         - $D_{k,p} = [D_i, D_j]$
         - $R_{k,p} = [F1_{D_i}, F1_{D_j}]$
       - end
       - $\mathbf{D} = All \quad D_{k,p}, \quad k = 1, \ldots, m, \quad p = 1, \ldots, n$
       - $\mathbf{R} = All \quad R_{k,p}, \quad k = 1, \ldots, m, \quad p = 1, \ldots, n$
     - end
   - end

end

ML-SVM system and the integrated approach.

The first step on the integration was to obtain the performance indices for both FDSs, upon which the expert engine can make final diagnostics. With that purpose both system were trained, tested and evaluated. As PCA-FLS was not tested when occurring simultaneous faults, the results presented next are tested under the occurrence of single faults. Thus, just part 1 from the DCS rules is used in that example (1 in Table 8.2).

8.3.1 Results using the PCA-FLS

Firstly, PCA-FLS was trained and tested by using the aforementioned new data sets. Its design was done similarly as demonstrated in chapter 7 but modifying the optimization to be led by $F1$ index instead of the objective function presented there. This variation was adopted to eliminate some irrelevant quantitative details included in the originally proposed objective function (sub-section 7.6.1.2), thus now driving the optimization by the index upon which comparison will be based.

Table 8.3 shows the first optimization results, from which the way of sampling can be selected. The sampling space was divided in five different ways. Z is the times sampling space is divided. Thus, the higher is Z the more space division and more samplings are evaluated to extract meaningful rules, which results in heavier computation. Space division trade-off is reached when most of fault details are captured while leaving out process noise.

Table 8.3: TE case study: Sampling intervals selection for the FLS.

Number of Intervals (Z)	1	2	3	4	5
Objective Function (OF)	60.7	62.4	68.4	73.2	67.6
Calculation Time, h	27.6	48.6	67.3	86.3	103.3
Number of samplings	10	20	30	40	50
Number of rules	21	42	61	81	97

Table 8.4 shows Recall, Precision and $F1$ indices (notice that OF is now $F1$ index) using the best way of sampling ($Z = 4$), before and after optimization, under parameters shown in Table 8.5. Noticeable improvements were achieved by adjusting the membership functions by the GA optimization.

Table 8.4: TE case study: Individual and global fitness function before and after FLS optimization (Z=4).

Individual	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	N.	Avg.
Initial																						
OF(F1)	80.1	96.5	3.5	95.5	0.0	93.5	95.2	67.1	20.6	98.5	80.0	90.7	70.3	78.4	0.0	93.6	53.9	22.7	94.8	38.2	49.7	63.0
Recall	93.3	95.4	2.0	93.8	0.0	90.8	90.9	66.1	11.6	97.6	91.9	86.1	60.4	65.0	0.0	88.0	79.6	14.7	90.1	29.6	100	64.1
Precision	71.3	97.6	14.9	97.3	0.0	96.3	100	68.1	93.5	99.4	70.9	95.9	84.1	98.8	0.0	100	40.7	50.2	100	53.7	33.0	69.8
Optimized																						
OF(F1)	90.3	95.4	72.5	96.8	55.0	95.2	95.1	68.9	27.1	98.4	89.7	87.7	65.5	97.5	33.8	92.2	50.5	27.9	91.2	43.0	63.8	73.2
Recall	94.8	91.5	60.5	96.7	52.2	90.8	90.7	68.9	21.3	97.0	83.8	78.4	64.4	96.2	25.8	85.6	81.2	18.5	87.0	34.1	92.9	72.0
Precision	86.2	99.6	90.4	96.9	58.1	100	99.9	68.8	37.2	99.8	96.4	99.6	66.6	98.9	49.1	100	36.6	56.4	95.8	58.4	48.6	78.3
($\Delta F1$)	10.2	−1.1	69.0	1.3	55.0	1.7	−0.1	1.8	6.5	−0.1	9.7	−3.0	−4.8	19.1	33.8	−1.4	−3.4	5.2	−3.6	4.8	14.1	10.2

Table 8.6 shows quantitative details from the diagnosis system performance that can be used for further analysis. The square matrix (Normal*Normal) summarizes the system diagnosis response for each tested sampling. Thus, samplings under the occurrence of each fault, indicated by columns, are diagnosed as belonging to faults indicated by rows, being false diagnosis those samplings standing off the matrix diagonal. a, b, c and d are the respective contingency matrix values for each tested fault (section 5.3). Additionally, last three rows summarize the performance indices corresponding to the optimized indi-

Table 8.5: Optimization algorithm parameters.

Number of intervals tested	from 1 to 5
N_{ind}	10
N_G	200
G	50
selection	Roulette wheel
recombination	Intermediate recombination
P_m	0.031
Range of a	$[0, 3]$
Range of Δ	$[0, 8]$
individual length	$n + A = 110$

vidual (Table 8.4).

8.3.2 Results using the ML-SVM system

Afterwards, the same analysis was done for the ML-SVM diagnosis system. Default soft margin and lineal kernel were utilized to build the hyperplanes. Table 8.7 shows all the diagnosis results details. ML-SVM attained higher performance than PCA-FLS in most of the tested faults. However, better PCA-FLS behavior in those faults poorly addressed by ML-SVM (3, 9, 15) lead to high expectations on the integration success. Note that the sum of samplings for each column of Tables 8.6 and 8.7 is equal to 1000, that is, the number of samplings tested for each fault. Normal columns at these tables sum 2240 ($112 * 20$) that comes from the testing of 112 samplings per fault (20), which represent a 2 hours random sampling under normal conditions simulated at the beginning of each tested fault. Note also that normal state was just tested not trained under ML-SVM system.

Faults 3 and mainly 9 and 15 are poorly classified by both systems. It is due to their effects on the process are hardly noticeable because of their slow dynamics as well as their low bias intensities. Fault 3 is better addressed because of its effects becomes evident after some time. It is after this time when PCA-FLS rules, extracted for that specific time intervals, are able to diagnose the fault. Further analysis on these faults would be required to increase their diagnosis performances.

However, these further analysis could be avoided by just looking at some other AEM system modules, as the SN design (see section 2.1 or last paragraph of chapter 3). Thus, focusing on the measurements of variables involved on these faults would allow easier isolations. This fact outlines how interactivity over the global AEM chain modules can enhance and lead to faster solutions of an specific module's problem.

Apart of this noticeable fact that helps to keep in mind the global AEM view of the thesis, integration of both FDSs was carried out to check the benefits of an integrated diagnosis scheme (global FDS). Thus, both systems were tested together diagnosing the same sets used to evaluate each independently. Analytical complete results of the integrated system are depicted on Table 8.8. Figure 8.2 summarizes the integrated and individual systems results for comparative purposes.

Note that, by the synergy of both systems integration, average $F1$ increased almost 17 % and 8 % regarding PCA-FLS and ML-SVM individual results respectively. Overall $F1$ increase was motivated by the general effect fault by fault. Performance on well addressed faults was kept by the integrated approach while those poorly classified were enhanced by the effect of the diagnosis selection algorithm and the FDS complementarity. Thus, in the integrated solution, each fault diagnosis performance reaches (or almost reaches) the highest of the individually attained performances (e.g. faults 3, 6 or 16). Such performance is even higher for some cases (e.g. faults 9 or 15) because of the positive effect caused by increasing the rest of faults performance.

In that way, both systems integration becomes a very effective way to improve qualitatively and quantitatively fault diagnosis performance in the tested case study. From the quantitative point of view, important performance increases can be obtained by a careful integration of FDSs (check Figure 8.2). From the qualitative point of view, the integrated scheme grants more resources and flexibility than any of the isolated systems by themselves. Thus, the resulting system combines, for example, good novelty identification properties of PCA-FLS and multiple fault diagnosis abilities given by the ML-SVM system.

Table 8.6: PCA-FLS analytical performance results. Square matrix (Normal*Normal) shows tested samplings under the occurrence of fault indicated by column which was diagnosed as belonging to fault indicated by row: samplings off the diagonal are false diagnosis. Last column shows items summation of each row and average for last three rows. Last seven rows show contingency matrix items for each fault (*a*, *b*, *c*, *d*) and the system performance indices.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**Normal**	**Sum/Avg.**
1	948	0	0	0	0	11	0	11	0	0	0	0	33	0	0	0	0	83	0	14	0	1100
2	0	915	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	919
3	0	0	605	0	6	0	0	0	24	1	0	0	4	1	12	0	0	11	0	5	0	669
4	0	0	0	967	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	998
5	2	4	10	0	522	0	0	1	143	0	0	77	0	0	128	0	0	11	0	0	0	898
6	0	0	0	0	0	908	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	908
7	0	0	0	0	0	0	907	0	0	0	0	0	0	0	0	0	0	1	0	0	0	908
8	15	7	0	3	0	21	0	689	0	0	52	81	27	26	0	48	20	5	2	5	0	1001
9	0	0	11	0	61	0	0	0	213	1	0	3	0	0	115	0	0	10	0	0	159	573
10	0	0	0	0	0	0	0	0	0	970	0	0	0	0	0	2	0	0	0	0	0	972
11	0	0	0	30	0	0	0	0	0	0	838	0	0	1	0	0	0	0	0	0	0	869
12	0	0	0	0	0	0	0	0	2	1	0	784	0	0	0	0	0	0	0	0	0	787
13	10	0	9	0	33	2	0	15	39	0	1	32	644	5	22	0	9	42	103	0	0	966
14	0	0	0	0	0	0	0	0	11	0	0	0	0	962	0	0	0	0	0	0	0	973
15	0	0	6	0	75	0	0	0	159	0	0	0	2	0	258	0	0	25	0	0	0	525
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	856	0	0	0	0	0	856
17	19	55	0	0	0	56	85	253	0	0	62	0	81	0	0	76	812	302	0	415	0	2216
18	0	0	0	0	0	0	0	9	0	0	6	0	41	0	0	5	7	185	0	75	0	328
19	0	0	0	0	0	0	0	0	30	0	0	0	0	0	8	0	0	0	870	0	0	908
20	0	5	6	0	2	2	7	0	34	0	0	0	1	0	6	0	74	105	1	341	0	584
Normal	6	14	353	0	301	0	1	22	345	23	10	23	167	5	451	13	78	220	24	145	2081	4282
a	948	915	605	967	522	908	907	689	213	970	838	784	644	962	258	856	812	185	870	341	2081	16275
b	152	4	64	31	376	0	1	312	360	2	31	3	322	11	267	0	1404	143	38	243	2201	5965
c	52	85	395	33	478	92	93	311	787	30	162	216	356	38	742	144	188	815	130	659	159	5965
d	21088	21236	21176	21209	20864	21240	21239	20928	20880	21238	21209	21237	20918	21229	20973	21240	19836	21097	21202	20997	17799	438835
Recall	**94.8**	**91.5**	**60.5**	**96.7**	**52.2**	**90.8**	**90.7**	**68.9**	**21.3**	**97.0**	**83.8**	**78.4**	**64.4**	**96.2**	**25.8**	**85.6**	**81.2**	**18.5**	**87.0**	**34.1**	**92.9**	**72.0**
Precision	**86.2**	**99.6**	**90.4**	**96.9**	**58.1**	**100**	**99.9**	**68.8**	**37.2**	**99.8**	**96.4**	**99.6**	**66.7**	**98.9**	**49.1**	**100**	**36.6**	**56.4**	**95.8**	**58.4**	**48.6**	**78.3**
F1	**90.3**	**95.4**	**72.5**	**96.8**	**55.0**	**95.2**	**95.1**	**68.9**	**27.1**	**98.4**	**89.7**	**87.7**	**65.5**	**97.5**	**33.8**	**92.2**	**50.5**	**27.9**	**91.2**	**43.1**	**63.8**	**73.2**

Table 8.7: ML-SVM analytical performance results. Square matrix (Normal*Normal) shows tested samplings under the occurrence of fault indicated by column which was diagnosed as belonging to fault indicated by row: samplings off the diagonal are false diagnosis. Last column shows items summation of each row and average for last three rows. Last seven rows show contingency matrix items for each fault (*a*, *b*, *c*, *d*) and the system performance indices.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Normal	Sum/Avg.
1	988	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	988
2	0	980	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	980
3	0	0	5	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	13
4	0	0	0	970	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	977
5	0	0	8	0	710	0	0	0	7	4	0	0	0	0	1	0	0	9	0	0	0	739
6	0	0	0	0	0	994	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	994
7	0	0	0	0	0	0	999	0	0	0	0	0	0	0	0	0	0	0	0	0	0	999
8	0	0	0	0	0	0	0	957	0	0	0	0	0	0	0	0	0	0	0	0	0	957
9	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	5
10	0	0	0	0	0	0	0	0	0	840	0	0	0	0	0	0	0	0	0	0	0	840
11	0	0	0	10	0	0	0	0	0	0	978	0	0	0	0	0	0	0	0	0	0	988
12	0	0	0	0	23	0	0	0	0	0	0	957	0	0	0	0	0	0	0	0	0	980
13	0	0	0	0	0	0	0	14	1	0	0	0	774	0	0	0	0	0	0	0	0	789
14	0	0	0	0	0	6	0	0	0	0	0	0	0	992	0	0	0	0	0	0	0	998
15	0	0	0	0	0	0	0	0	3	0	0	0	0	0	3	0	0	2	0	0	0	8
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	989	0	0	0	0	0	989
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	921	0	0	0	0	921
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	695	0	2	0	697
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	957	0	0	957
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	857	0	857
Normal	12	20	987	20	267	0	1	29	984	156	15	43	226	8	988	11	79	294	43	141	2240	6564
a	988	980	5	970	710	994	999	957	5	840	978	957	774	992	3	989	921	695	957	857	2240	17811
b	0	0	8	7	29	0	0	0	0	0	10	23	15	6	5	0	0	2	0	0	4324	4429
c	12	20	995	30	290	6	1	43	995	160	22	43	226	8	997	11	79	305	43	143	0	4429
d	21240	21240	21232	21233	21211	21240	21240	21240	21240	21240	21230	21217	21225	21234	21235	21240	21240	21238	21240	21240	15676	440371
Recall	**98.8**	**98.0**	**0.5**	**97.0**	**71.0**	**99.4**	**99.9**	**95.7**	**0.5**	**84.0**	**97.8**	**95.7**	**77.4**	**99.2**	**0.3**	**98.9**	**92.1**	**69.5**	**95.7**	**85.7**	**100**	**78.9**
Precision	**100**	**100**	**38.5**	**99.3**	**96.1**	**100**	**100**	**100**	**100**	**100**	**99.0**	**97.7**	**98.1**	**99.4**	**37.5**	**100**	**100**	**99.7**	**100**	**100**	**34.1**	**90.4**
F1	**99.4**	**99.0**	**1.0**	**98.1**	**81.7**	**99.7**	**99.9**	**97.8**	**1.0**	**91.3**	**98.4**	**96.7**	**86.5**	**99.3**	**0.6**	**99.4**	**95.9**	**81.9**	**97.8**	**92.3**	**50.9**	**79.5**

Table 8.8: Integration analytical performance results. Square matrix (Normal*Normal) shows tested samplings under the occurrence of fault indicated by column which was diagnosed as belonging to fault indicated by row: samplings off the diagonal are false diagnosis. Last column shows items summation of each row and average for last three rows. Last seven rows show contingency matrix items for each fault (*a*, *b*, *c*, *d*) and the system performance indices.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**Normal**	**Sum/Avg.**
1	988	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	83	0	0	0	1104
2	0	980	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	984
3	0	0	600	0	1	0	0	0	24	0	0	0	4	1	12	0	0	11	0	3	0	656
4	0	0	0	971	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	983
5	2	4	16	0	842	0	0	1	150	0	0	17	0	0	129	0	0	14	0	0	0	1175
6	0	0	0	0	0	994	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	994
7	0	0	0	0	0	0	999	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1000
8	0	0	0	0	0	0	0	958	0	0	0	0	0	0	0	0	0	0	0	0	0	958
9	0	0	11	0	20	0	0	0	212	0	0	0	0	0	115	0	0	9	0	0	159	526
10	0	0	0	0	0	0	0	0	0	974	0	0	0	0	0	0	0	0	0	0	0	974
11	0	0	0	29	0	0	0	0	0	0	978	0	0	0	0	0	0	0	0	0	0	1007
12	0	0	0	0	23	0	0	0	2	0	0	960	0	0	0	0	0	0	0	0	0	985
13	4	0	9	0	8	0	0	14	40	0	0	0	789	2	22	0	1	18	0	0	0	907
14	0	0	0	0	0	6	0	0	11	0	0	0	0	992	0	0	0	0	0	0	0	1009
15	0	0	6	0	34	0	0	0	157	0	0	0	2	0	260	0	0	26	0	0	0	485
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	989	0	0	0	0	0	989
17	0	0	0	0	0	0	0	5	0	0	0	0	4	0	0	0	921	0	0	0	0	930
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	611	0	3	0	614
19	0	0	0	0	0	0	0	0	30	0	0	0	0	0	8	0	0	0	976	0	0	1014
20	0	2	6	0	0	0	0	0	34	0	0	0	1	0	6	0	0	10	0	857	0	916
Normal	6	14	352	0	72	0	1	22	340	22	10	23	167	5	448	11	78	217	24	137	2081	4030
a	988	980	600	971	842	994	999	958	212	974	978	960	789	992	260	989	921	611	976	857	2081	18932
b	116	4	56	12	333	0	1	0	314	0	29	25	118	17	225	0	9	3	38	59	1949	3308
c	12	20	400	29	158	6	1	42	788	26	22	40	211	8	740	11	79	389	24	143	159	3308
d	21124	21236	21184	21228	20907	21240	21239	21240	20926	21240	21211	21215	21122	21223	21015	21240	21231	21237	21202	21181	18051	441492
Recall	**98.8**	**98.0**	**60.0**	**97.1**	**84.2**	**99.4**	**99.9**	**95.8**	**21.2**	**97.4**	**97.8**	**96.0**	**78.9**	**99.2**	**26.0**	**98.9**	**92.1**	**61.1**	**97.6**	**85.7**	**92.9**	**84.7**
Precision	**89.5**	**99.6**	**91.5**	**98.8**	**71.7**	**100**	**99.9**	**100**	**40.3**	**100**	**97.1**	**97.5**	**87.0**	**98.3**	**53.6**	**100**	**99.0**	**99.5**	**96.3**	**93.6**	**51.6**	**88.8**
F1	**93.9**	**98.8**	**72.5**	**97.9**	**77.4**	**99.7**	**99.9**	**97.9**	**27.8**	**98.7**	**97.5**	**96.7**	**82.7**	**98.8**	**35.0**	**99.4**	**95.4**	**75.7**	**96.9**	**89.5**	**66.4**	**85.6**

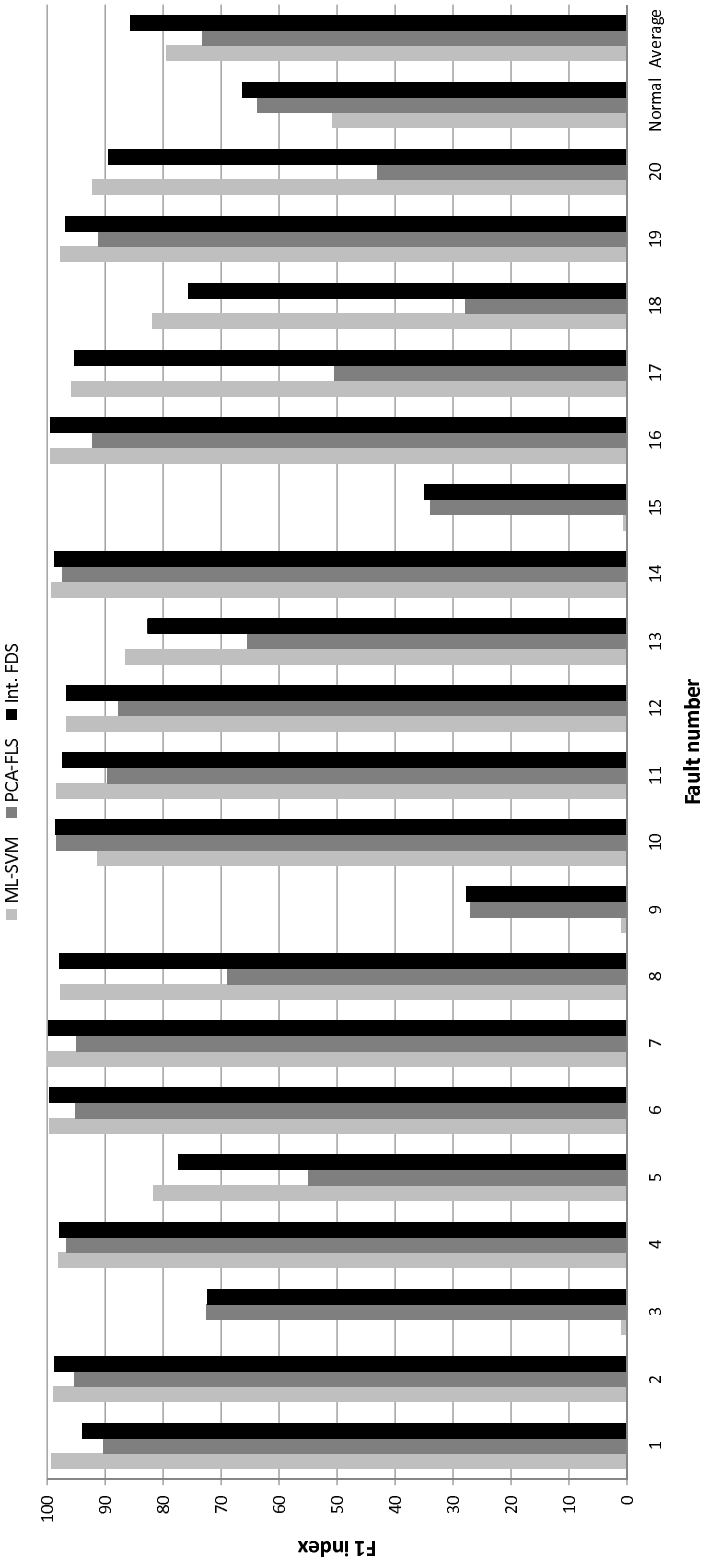


Figure 8.2: Comparison of performance results for the ML-SVM, the PCA-FLS and the integrated super FDS.

8.4 Conclusions

In this chapter both diagnosis systems proposed in the thesis are compared in the same basis obtaining very good quantitative diagnosis results. Complementarity of both systems was demonstrated theoretically as well as practically by reviewing the system characteristics and their results diagnosing the TEP. Three faults were not properly addressed revealing a very interesting point when considering the thesis general approach. The necessity of diagnosing correctly these three faults could lead to invest high human resources on the further development of both FDS by means of case dependent detailed procedures. However, more simple solutions might be found when looking at the problem from the global AEM view of this thesis. Thus, big efforts required to solve this specific and difficult faults by focusing in both diagnosis techniques, might be easily avoided by looking at the SN design. By this manner, a new SN that focuses on the variables involved on these difficult faults in an independent could easily solve the diagnosis problem with lower cost.

Apart of this interesting conclusion, the goal of developing an effective and robust super FDS was attained. A rule based expert system was proposed as a DCS to generate univocal diagnostics from both FDSs. The system was developed in a very intuitive manner being tested quantitatively under the occurrence of single faults. Nevertheless, the response of the DCS under multiple faults was also detailed theoretically (Table 8.2). The system final diagnostics were supported at any sampling time by additional information (reliability of diagnostics R) to the process operator.

The proposed diagnosis integrated framework generated important synergies from quantitative and qualitative points of view. High diagnosis performance was obtained in a very challenging diagnosis problem, offering quantitative complete results for comparison with present and future approaches. These presented results are the best ever reported addressing the complete set of faults of the TE benchmark. Although even higher performance indices could have been attained by a more detailed adjustment of the DCS set of rules, or by means of fine tuning of each FDS, main FDSs deployment details were shown throughout the chapter.

Nomenclature

a	Samplings happened and diagnosed
b	Samplings diagnosed but not happened
c	Samplings happened and not diagnosed
d	Samplings not happened and not diagnosed
$F1_{D_i}$	F1 index diagnosing fault D by sub-system i
$F1_{D_j}$	F1 index diagnosing fault D by sub-system j
G	Generations criteria for stopping GA search
m	Number of faults diagnosed by the first FDS
n	Number of faults diagnosed by the second FDS
N_G	Number of generations
N_{ind}	Number of individuals in each generation
OF	Objective function
P_m	Mutation probability
D	Final diagnostic
R	Reliability of the diagnostic
Z	Number of sampling intervals

Subscripts

i	i-th diagnosis system
j	j-th diagnosis system
k	Number of fault diagnosed by the first FDS
p	Number of fault diagnosed by the second FDS

Acronyms

DCS	Decision Conflict Solver
FDS	Fault Diagnosis System
FLS	Fuzzy Logic System
GA	Genetic Algorithms
PCA	Principal Component Analysis
ML-SVM	MultiLabel based Support Vector Machines
TE	Tennessee Eastman

CHAPTER 9

Decision Support System

At that point, fault sources are assumed to be known. However, such information may not be necessarily enough. When a fault is diagnosed there are new questions that require of effective and fast responses. Which are the bias consequences? or what measures should I take? and even further, what should be done first? Thus, actions derived from response to these questions must be carefully tackled, as they can enhance or put at risk the safety and economy of the plant.

In this chapter, an effective and dynamic integration of AEM and Process Hazards Analysis (PHA) is carried out to better deal with these new issues. In the integrated system, HAZOP analysis original version was extended to generate quantitative information about thresholds for key variables, as well as a set of corrective actions (CAs) to choose from. Dynamic integration is achieved by combining the estimation of future variables values given by a plant dynamic model with the obtained CA list. The outcome of such combination will be a protocol of preferred CAs depending on the current plant conditions. The system is developed in a Matlab and Aspen Dynamics integrated framework. An industrial sour water stripping plant was used as case study.

9.1 Introduction

The same level of automation and robustness achieved by the current regulatory control systems in chemical plants is also expected for AEM to increase plant safety and efficiency. As the first step in achieving this goal, monitoring and diagnosis systems were developed over the last twenty years for the on-line detection and identification of abnormal situations. Nevertheless, the automation of AEM not only requires accurate fault diagnosis but also a complete system response with proposed corrective actions (CAs) based on the knowledge of causes and consequences of the diagnosed faults.

Process hazard analysis (PHA) can generate information for an AEM system to deal with on-line plant safety issues. Although PHA is done off-line during the process design step, it can also provide qualitative answers about causes and consequences of process

deviations (Vaidhyanathan and Venkatasubramanian, 1995) which can be used to support AEM. In the literature, some works have dealt with both off-line and on-line safety issues separately (Ruiz, 2001) or have dealt with the safety problem from the opposite viewpoint by using AEM tools for improving PHA (Eizenber et al., 2006). However, few and only qualitative contributions have appeared so far on the AEM-PHA integration showing its potential as a right way to enhance the supervisory system automation and robustness (Dash and Venkatasubramanian, 2003).

This chapter presents the development of a methodology that dynamically integrates a data driven FDS (chapters 6, 7 and 8), a process dynamic model and HAZOP analysis (a popular PHA technique). The goal is to improve the AEM performance by integrating aspects of the off-line safety analysis into the on-line AEM (AEM-PHA integration). To achieve this goal, the HAZOP analysis was extended to provide quantitative information about thresholds for key variables, as well as corrective actions (CA) to face consequences of abnormal events. The integrated approach also uses a dynamic model of the plant to simulate the effects of the diagnosed fault on the future plant performance. Dynamic simulation is crucial in identifying the variable(s) that will reach the specified safety threshold(s) the earliest, thereby helping the system to prioritize CAs. The system is developed in a Matlab and Aspen Dynamics integrated framework and an industrial sour water stripping plant was used to show the benefits of the decision supporting system.

The chapter is arranged as follows: section 9.2 outlines off-line and on-line aspects of plant safety. Section 9.3 presents the off-line and on-line components used in the proposed methodology. The AEM-PHA integrated configuration is discussed in section 9.4 while results and benefits of such integration are presented in section 9.5. A summary of the proposed methodology is given in section 9.6.

9.2 Plant safety: PHA-AEM

As it has been already mentioned in this thesis, safety management in current chemical plants is a complex issue that has attracted great attention from the industry throughout last decades. Two different approaches to plant safety have evolved during these years. One approach corresponds to the safety related criteria used in the process design stages (Process Hazard Analysis). The other approach is the main subject of this thesis and focuses on plant safety when the process is running (Abnormal Events Management). A brief description of both approaches is given next to summarize and recall basic points and major differences.

9.2.1 Process Hazards Analysis

PHA is a critical activity of Process Safety Management (PSM) and it can be defined as the "systematic and proactive identification, mitigation, and assessment of potential process hazards which could endanger the health and safety of humans and cause serious economic losses" (Dash and Venkatasubramanian, 2003). It has become a crucial issue in the process industry since very restrictive regulations have been introduced to ensure safety of large facilities. Regulations in most of the industrialized countries highlight the importance of PHA, for example, the Operational Safety and Health Administration (OSHA) of the U.S government (OSH94, OSHA PSM standard title 29 CFR 1910) requires a wide safety analysis prior to building major chemical plants, or the Technical

Notes of Prevention ("Notas técnicas de prevención") edited by the Spanish National institute of Safety and Hygiene at Work ("Instituto Nacional de Higiene y Seguridad en el Trabajo") standardizes the techniques to identify potential accidents in the process industry.

Among several available techniques, HAZOP analysis (Lawley, 1974; Kletz, 1986) has become the most popular and preferred methodology to develop the PHA, although other techniques such as Fault Tree Analysis, Checklist or Failure Modes and Effects Analysis can be also found in the related literature.

9.2.2 Abnormal Events Management

AEM is the main object of study of this thesis. As it has been remarked before it is the on-line system which diagnoses process deviations that could lead to dangerous consequences (abnormal events) and transforms them into useful knowledge to take corrective actions. The core of AEM consists of monitoring and fault diagnosis systems, but proper data acquisition and rectification systems as well as effective decision-making supporting tools are also important for achieving a good overall performance. In that sense, an early diagnosis when the plant is not yet out of control is always critical in avoiding more dangerous consequences. However, in complex processes, early diagnosis is not enough because very often corrective actions must be quickly adopted without a thorough problem comprehension. For that reason, tools that can help in the problem understanding and in organizing corrective measures are important. Considerable amount of work has been reported in the literature regarding data rectification (Yélamos et al., 2007) and fault detection and diagnosis (Venkatasubramanian et al., 2003b) but few works have practically faced the final supporting step of providing help in the decision-making phase (Chen and Modarres, 1992; Weidl et al., 2005).

9.3 Decision supporting tools

In this section, basic tools of the proposed AEM framework are described. They are classified as on-line and off-line operation systems based on their use. In that sense, they are real-time supporting tools (FDS, dynamic model) or process knowledge from off-line analysis (extended HAZOP analysis) made available for helping AEM.

9.3.1 On-line operating systems

9.3.1.1 Fault diagnosis system

Data driven fault diagnosis systems (FDSs) (Venkatasubramanian et al., 2003a) have been gaining popularity in process industry during the past decades due to their ease of implementation and robust design. The main requirement to use these efficient FDSs is the availability of broad and representative data that capture the process dynamics, which in turn helps the diagnostics.

In this chapter faults considered on the design of the FDS are only those identified from the off-line knowledge extracted from a prior HAZOP analysis. The representative data used in the design of the FDS may be generated by a dynamic model. The on-line FDS

used to show the AEM-PHA integration benefits is that one resulting from the combination of the PCA-FLS and the ML-SVM systems described in chapter 8.

The PCA model was built by using process data at normal-state, and was used for fault detection. The ability of PCA to build low dimensional data driven models for the monitoring of process operations is well known (Kourti and MacGregor, 1995) and industrial applications have been presented in the literature (Neogi and Schlags, 1998). It enables rapid fault detection using T^2 and SPE (squared prediction error) (Yoon and MacGregor, 2000) when one of them exceeds the detection limits for more than two consecutive samplings. Later on, the FLS and the ML-SVM systems infer the most probable process state from the process signal analysis. The redundancy increase the diagnosis robustness (see chapter 8).

9.3.1.2 Dynamic model

The plant dynamic model is a crucial tool in the proposed methodology. In the one hand, characteristic data sets can be generated by simulating the original faults considered in the HAZOP analysis under specific plant deviations. These data sets are invaluable to train the FDS. On the other hand, the dynamic model allows us to reproduce the current plant conditions and forecast the future plant state, thus predicting the consequences of deviations. Thus, the effects of consequences and the CAs to solve them can be quantified and ranked respectively, thereby helping in the selection of the best protocol. In this sense, the dynamic model becomes a vital link in the AEM-PHA integration.

9.3.2 Off-line operating systems: Extended HAZOP analysis

HAZOP analysis was chosen as the technique for making PHA because of its wide acceptance in the chemical process industry. HAZOP is typically carried out by a team of experts specialized in the design, operation and maintenance of facilities. During HAZOP analysis, disturbances are applied to a process using systematic and methodical procedures (Gillet, 1997) to identify potential problems. The system is studied one element at a time in a sequential way. Each system element is questioned using "guide-words" (for example, "More of ", "Less of", "More than", etc.) to produce deviations on it and evaluate their causes, consequences and possible safeguards. HAZOP analysis consists of the analysis of the consequences of these deviations and the conclusions reached by the expert team.

This analysis requires substantial amount of time, effort and money, and the results are invaluable from the safety design viewpoint. However, the original HAZOP analysis cannot be used as a supporting tool for the on-line decision-making because of two main drawbacks:

- Only qualitative results are available.
- The information is isolated and cannot be efficiently integrated on the process operation control.

Extended HAZOP Thus, HAZOP analysis results from mandatory safety regulations could be used effectively for making decisions in plant operation. With that purpose, the

traditional qualitative HAZOP analysis should be quantified and integrated on an automatic framework to generate advice and recommendations to plant operators. The proposed "extended HAZOP" analysis consists of the following four main steps:

1. Key variables identification.
2. Quantification of the key variables safety thresholds.
3. Proposal of the set of corrective actions.
4. Grouping of CAs based on the seriousness of problems to be solved.

To illustrate these four steps, an example of a typical HAZOP analysis is described next. The example consist of the consequence *high pressure in pipe-12 releasing flammable components into plant area and causing fire hazard* resulting from the deviation *high flow in pipe-12*.

9.3.2.1 Key variables identification

First of all, the critical variables appearing in the cause and consequence lists from the HAZOP analysis must be identified. These variables contents are the origin of the adverse problems occurring in the plant and the possible system response should be focused on them. In this example, *PRESSURE* would be identified as the critical variable of the consequence described above.

9.3.2.2 Quantification

Once the critical variables have been identified, the quantification of the variable safety threshold is essential in order to know which limits cannot be exceeded when the plant is running. For the key variable identified above, such quantification would be in the form of *Pressure for release of flammable components in pipe-12 = 5 atm*.

9.3.2.3 Corrective actions

The third step consists of proposing CAs to fix the abnormal event causes and abort or mitigate their consequences. In this sense, from a deep knowledge of the plant, several actions can be implemented to avoid critical plant situations. For the example given above the following CAs, based on causes and consequences of the studied deviation, could be proposed: 1) *shut down upstream flow* or 2) *by-pass flow partially to secondary tank*.

9.3.2.4 Grouping of corrective actions

The final step involves the classification of CAs by the consequence seriousness "degree" that they are addressing. Such "degree" indicates the relative CA/consequence importance regarding the rest of CAs/consequences. In that sense, three or more groups, depending on the process complexity can be established. As an example three groups could be arranged: critical, medium and low seriousness. This classification will allow assigning weights to CAs belonging to each group in order to determine the CAs application order. The more serious consequences the more weighted CAs. These heavier weighted CAs will have priority over those that are weighted less. This information together with the time to reach the safety thresholds will provide the quantitative basis for ordering the CAs in a protocol adapted for the prevailing conditions. The quantitative criteria established

to order CAs from information of time to thresholds and CAs weights will come from expertise and process knowledge in each case.

In the example described above, the first CA proposed (*shut down upstream flow*) would belong to the critical group and would be highly weighted because it potentially rectifies a critical consequence, namely *breakage of a pipe resulting in the release of flammable material*. The second CA proposed (*by-pass partial flow to secondary tank*) would belong to the medium group and would be weighted less, because it rectifies a consequence which is not so critical, namely *the release of secondary facility gas to plant*. If the model predicts that pipe breakage and pipe gas release would occur at the same time in the future, *shut down upstream flow* would be implemented first because it is a more weighted CA, otherwise time to safety thresholds will lead to the most urgent CA.

9.3.2.5 Extended HAZOP analysis configuration

The flowchart of the proposed algorithm to extend the standard qualitative HAZOP analysis is shown in Figure 9.1. It generates a set of quantified and weighted CAs that can be later used for on-line decision-making during AEM.

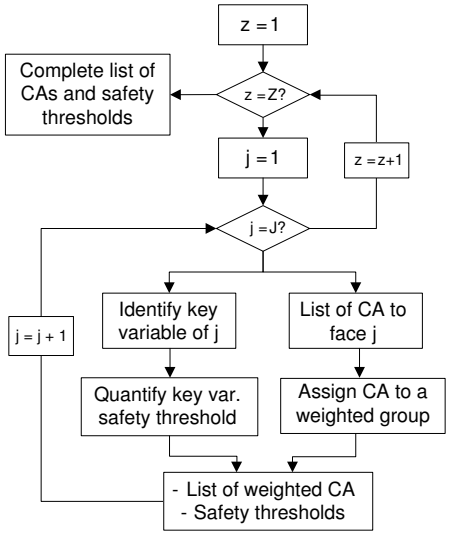


Figure 9.1: Flowchart of the extended HAZOP analysis algorithm. z represents deviations and j causes-consequences in the original HAZOP analysis.

9.4 AEM-PHA dynamic integration

In this methodology, the dynamic model and the extended HAZOP analysis are crucial and complementary techniques which allow integration of future and past safety analyses respectively in order to take the appropriate present CAs. The system architecture, involving the plant and all the supporting tools required to generate CAs, is shown in Figure 9.2. The physical components of this scheme, the plant and process operator, are connected to the software modules via interfaces represented by interlined blocks. The goal of all

supporting tools is to extract information contained in process signals and transform it into useful supervisory knowledge and plant safeguards to keep the plant under control.

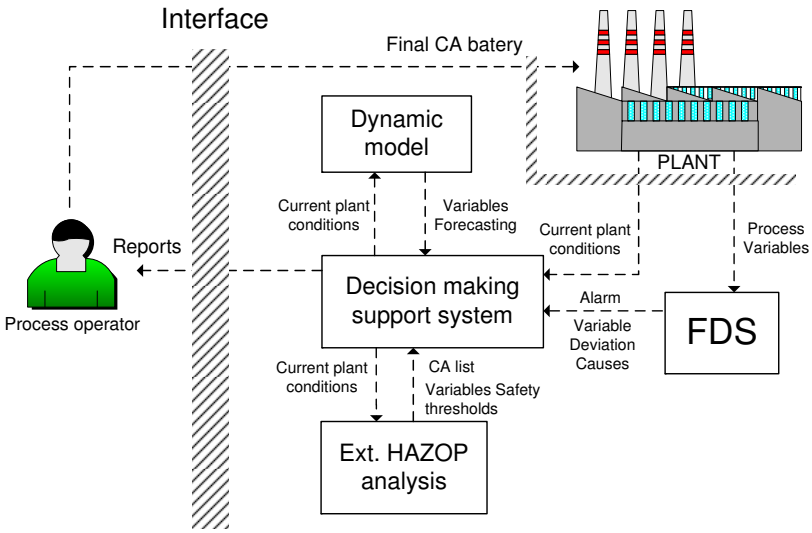


Figure 9.2: System architecture and information path.

The information collected from plant sensors in form of variables measurements goes directly to the trained FDS. The FDS generates alarms and provides source causes of deviations in case the process biases from its normal operating conditions. This information together with current plant status is used by the decision support system as straight supporting indications. Whenever the FDS detects a deviation, the current plant conditions and deviation causes are simulated using the dynamic plant model. At the same time, information on deviation causes is used to collect the extended HAZOP matching weighted CA list along with safety thresholds for the corresponding variables.

Using outputs from all components, the decision support system generates reports for the process operator. Using the profiles predicted by the dynamic model, it ranks the variables based on how close they are to the safety thresholds (urgencies). The CAs for all causes and consequences are then collectively ranked using this urgency and the weight/seriousness criteria. The ranked CAs are also supplemented with secondary information such as the most contributing variables to the fault, as well as the fault causes and consequences. With all this knowledge, the process operator will be able to make the best decisions. Details of the information flows through the decision support system are given in Figure 9.3.

9.5 Results and discussion

9.5.1 Case study

A sour water stripping plant was selected as a case study to demonstrate the benefits of the proposed framework. It is both small in size to be manageable and rich in operational

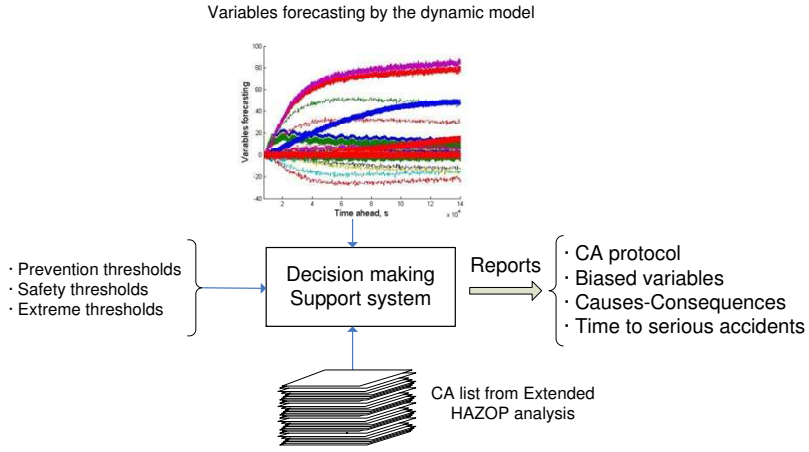


Figure 9.3: Decision-making support system: knowledge sources.

complexities. Additionally, a detailed HAZOP analysis is available in the literature. A qualitative HAZOP analysis was carried out by a team of experts from Arthur D. Little, Inc, Cambridge, MA (USA). Later, the analysis was compared with the results of a HAZOP analysis software developed by Vaidhyanathan (1995) called HAZOPExpert.

The sour water stripping (SWS) plant is fed with several sour water streams generated by sulfide removal operations in a refinery. As a final result, the SWS plant generates a water stream that is pure enough for reuse or safe disposal to the environment. The flowsheet of the SWS plant is given in Figure 9.4. First, the sour water is separated in a surge drum decanter to remove slop oil. Then, the outlet stream is pumped into a storage tank where any remaining slop oil is removed. From this storage tank, sour water is pumped through a heat exchanger to a sour water stripper where ammonia and hydrogen sulfide dissolved in sour water are stripped off.

The specific plant operating conditions, sour water impurities, feed composition, equipment operation conditions, design sizes and so on are defined by considering different and common refinery sour water characteristics and stripping plant details available in the literature (Thiele et al., 2003; Chevron, 1998). Table 9.1 summarizes key streams compositions of the SWS plant.

In terms of safety, it is important to focus on the main risks that the system has to face. The hydrocarbon streams (slop oil) are flammable whereas streams containing hydrogen sulfide and ammonia are highly toxic. Therefore, the release of any of these streams due to any malfunction is a safety concern in the plant.

The process consists of 26 pipes, 5 flow control valves (FCV), 5 non return valves (NRV), 5 pumps, 1 surge drum, 1 storage tank, 1 stripper, 1 condenser, 1 stripper overhead accumulator and 6 controllers. In each of the processing units (total of 5 units) 3 biases for flow (high, low, zero), 2 biases for temperature (high, low), 2 biases for pressure (high, low), 3 biases for level (high, low, zero), and 3 biases for each concentration (total of 8 species) were considered in the original HAZOP analysis. 32 causes and 42 consequences were

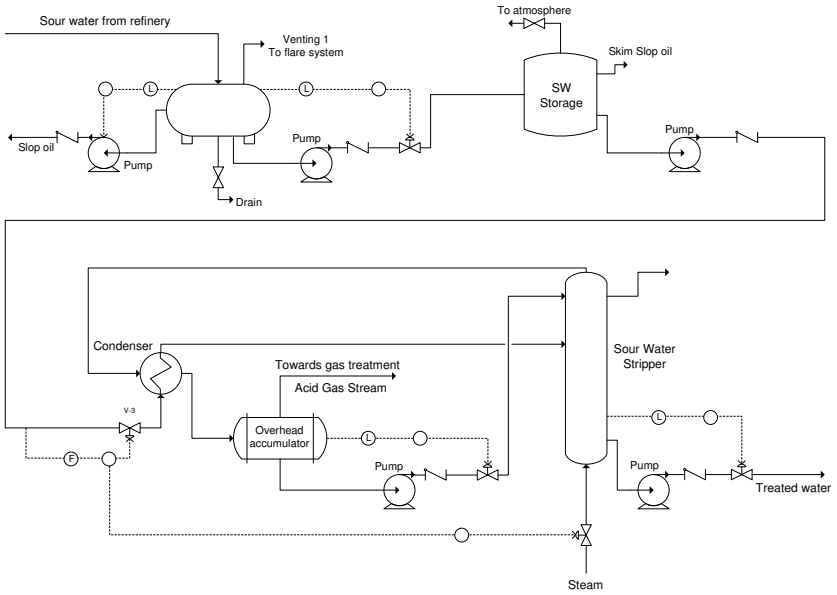


Figure 9.4: Sour water stripper plant flowsheet.

Table 9.1: Sour water plant steady state main streams details.

	From refinery*	Slop oil**	Venting 1**	Acid gas**	Treated water **	Steam*
H_2O flow	28.2	0.02	0.11	21.2	21.9	15
NH_3 flow	1.2	0.02	0.08	1.1	2.2 E-04	0
CO_2 flow	6.0 E-04	4.90E-05	5.4 E-04	1.57 E-05	2.0 E-18	0
H_2S flow	0.6	0.12	0.44	0.04	6.9 E-13	0
n-Octane flow	7.5	7.4	0.1	6.7 E-03	2.2 E-17	0
n-Nonane flow	7.5	7.3	0.04	0.16	1.0 E-08	0
n-Decane flow	7.5	7.5	0.02	3.5 E-03	5.1 E-21	0
n-Undecane flow	7.5	7.5	7.9 E-03	4.7 E-04	3.5 E-28	0
Total flow	60	29.8	0.8	22.5	21.9	15
Temperature	100	76.2	75	120	134.2	180
Liquid fraction	95	100	0	0	100	0

Flows are given in kg/h, Temperature in ^{o}C and Liquid fraction in %.

extracted by the Arthur D. Little team summarizing the most important plant deviations. Additional details of the complete plant HAZOP analysis can be found in Vaidhyanathan (1995).

The system was initially simulated at steady-state using Aspen Plus to test the model sensitivity to big changes in flow rates and other variables (temperatures, pressures, concentrations, etc.), and to check thermodynamic consistency. The NRTL thermodynamic model was used for calculating the V-L-L equilibrium in the oil decanter and for coping with the non-ideality of water solutions present in the stripper. NRTL is used for calculating liquid phase properties while vapor phase properties are calculated from Redlich-Kwong equation of state. H_2S, NH_3 and CO_2 are considered as Henry components (Lee et al., 2004). The thermodynamic data from Aspen Properties databanks was used. The steady-state simulation model was later converted into Aspen Dynamics for predicting plant dynamics. Plant controllers were added to imitate the control system reported in the original plant (Vaidhyanathan, 1995) in order to mimic the process response to dynamic changes.

The steady-state simulation was built by selecting suitable models from the Aspen Plus model library. The decanter was modeled using a L-L equilibrium model, assuming perfect phase separation. Pump and valve models were added to the flowsheet in order to model pressure changes. The stripper was modeled using a rigorous multistage vapor-liquid equilibrium model (RADFRAC, AspenTech (2006)) containing 4 equilibrium stages. Sour water and condenser liquid recycle streams enter the stripper at top above stage 1, while high pressure steam enters at the bottom. Condenser pressure is set at 3 bars and treated water leaves the column at 134^{o}C. Vapor outlet from the column is flashed in a separate cooled vessel that acts as a partial condenser, allowing separate control of this flow. No liquid entrainment was considered in the modeling of the L-V phase separator. The condenser heat flow is used to pre-heat the sour water inlet to the column.

The steady-state model was exported to Aspen Dynamics by using an expert system that translates the steady-state information into initial values and model parameters in the dynamic simulator. To do this translation, information about equipment geometry, valve specifications and pump curves are required. The expert system generates curves of head *vs.* suction volumetric flow rate and efficiency *vs.* suction volumetric flow rate for centrifugal pumps to give realistic behavior during the dynamic simulation. These typical curves are fitted around the steady-state conditions. The operating head, suction volumetric flow rate and efficiency are used in generating typical performance curves (AspenTech, 2006). The pressure drops across valves were set greater than 0.1 bar to attain good control. Nominal steady-state flow was set at 50% of valve opening.

After creating the dynamic model, controllers were added according to the HAZOP analysis outline available. The controller on the decanter consists of two level controls, one measuring the oil-water interface and the other measuring the total liquid level. The controller measuring slop oil level acts directly on a downstream pump, while the controller measuring the interface level acts on a valve that controls the flowrate out of the decanter. In addition, two similar level controllers were added to the control scheme, one to the stripper which measures the sump liquid level and controls output flowrate of the treated water, and the other one to the overhead accumulator, which measuring the liquid level controls the liquid recycle flowrate. Finally, a flow controller and a proportional dependent ratio controller were added to control sour water and vapor inlet flows to the stripper. These controllers adjust and measure sour water flow to the column and set the vapor flow by a proportional constant. All controllers added to the dynamic model were considered

to have proportional and integral control actions.

Details of input and output streams from steady state simulation are shown in Table 9.1. Inlet and outlet process streams are marked with one and two asterisks, respectively. Treated water contains only traces of impurities whereas *Venting 1* and *Acid-gas* streams retain most of NH_3 and H_2S, which is gathered for a subsequent stream treatment. Slop oil is supposed to contain a mixture of high molecular weight hydrocarbons (C8 to C11). Given that no petroleum analysis was available, it was assumed that highly soluble hydrocarbons were completely removed from the incoming water stream.

Figure 9.5 shows the software implementation of this methodology. Due to the lack of real plant data, process dynamic data was generated from simulations of the plant in Aspen Dynamics. Gaussian noise was added to variables from simulation in order to emulate real plant variable measurements. Although it is an evident problem simplification, as only a quasi perfect model is available, it does not affect the demonstration of the proposed approach.

The connection between Aspen Dynamics and Matlab was implemented using a Simulink unit model that allows the retrieval and setting of variable values from and to an Aspen Dynamics model by using the Windows COM interface. This provides a robust approach by combining the modeling power of Aspen Dynamics with the flexibility of calculus embedded in Simulink.

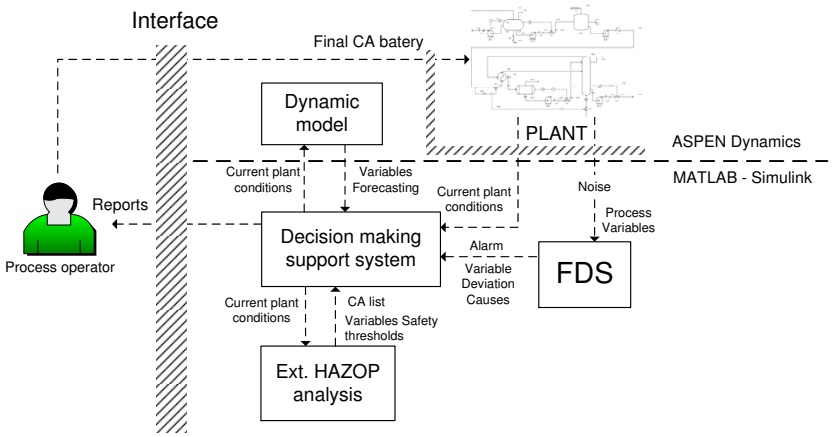


Figure 9.5: Software details of the system architecture.

9.5.2 Experimental arrangements

To illustrate the overall support system performance, two deviations from the original HAZOP analysis were tested: a *High* deviation in pipe's *flow* coming from refinery (*Deviation 1.1* on Table 5.2 at Vaidhyanathan (1995)), and another *High* deviation in accumulator's *level* (*Deviation 10.1* on Table 5.2 at Vaidhyanathan (1995)). Details of both deviations are given in Table 9.2.

Table 9.2: Details of simulated faults.

	Biased variable	Bias	Type	Details	Cause
Fault 1	Flow in pipe from refinery	High	Ramp	Slope = 1.72 [kg/h/h]	Increase of upstream flow
Fault 2	Level in accumulator	High	Ramp	Slope = 0.00524 [m/h]	Failure on level controller. Ramp closing outlet valve

56 variables were selected to collect real-time information about the process state. The variables consisted of temperatures, pressures, levels, species compositions and flowrates of main streams and equipments. The FDS was first trained with plant normal data (from the dynamic model running in steady-state) in order to create a PCA model that could detect any deviation from the normal operating regime. A new subspace was created explaining 80% of the process variance while reducing original problem to a lower dimensional subspace. Thresholds for SPE and T^2 were calculated as the 99 percentile of each statistical normal distribution and were later used as detection limits in monitoring charts.

Fault data sets extracted from simulation of previously described faults were used to train both diagnosis systems. Thus, a set of rules was created as the basis of the PCA-FLS. At the same time, these faulty sets were used to create the ML-SVM classifiers. Both systems extract the main features of each fault in form of rules and faulty set boundaries (hyperplanes) that link variations in the measured variables with the corresponding fault (chapter 7).

In order to illustrate some characteristics of the resulting FDS, some details of the PCA-FLS subsystem are presented throughout the results discussion. For example, 12 different rules were extracted by the PCA-FLS to diagnose normal state as well as Fault 1 and 2. A summary of these rules is given in Table 9.3. Only the states of some of the 56 variables contributing to *SPE* (*CSPE*) are shown due to space constraints. L (low), N (normal), H (high) and I (indifferent) represent the values that *CSPE* variables adopted under the rules extraction from the data base for each fault. L, N, and H values for a *CSPE* variable are later used to infer the occurrence of the corresponding fault. The value I means that this variable is not used to infer the associated fault.

Table 9.3: Rules extracted by the PCA-FLS.

Rule	CSPE 1	CSPE 2	CSPE 3	CSPE 4	CSPE 5	...	CSPE 56	Fault 1	Fault 2	Normal
1	I	I	I	I	I	...	I	H	L	I
2	N	N	N	N	N	...	I	L	H	I
3	N	N	N	N	N	...	N	L	L	H
...	...	...	...	...	...	...	...	...	...	...
7	H	H	H	H	L	...	N	H	L	I
8	N	L	I	H	L	...	L	L	H	I
...	...	...	...	...	...	...	...	...	...	...
12	I	L	H	H	I	...	I	L	H	I

CSPE in Table 9.3 represents variables used in diagnosis, corresponding to the contribution to the *SPE* of each process variable. For example, when all variables are normal (N) as shown in rule 3, "Normal" state will be valued as high (H) whereas Fault 1 and Fault 2 will be valued low (L). The result of the inference of plant data with all these rules will be a final diagnostic. In next subsection the response of the AEM-PHA integrated framework facing Faults 1 and 2 under different operating conditions is described to illustrate the integration benefits.

9.5.3 Deviation 1: High plant inlet flow

From the original qualitative HAZOP analysis performed by HAZOPExpert (Vaidhyanathan, 1995), an extended quantitative scheme to support decision-making can be derived following the procedure explained in subsection 9.3.2. Details of the original HAZOP analysis results are given in Appendix C.

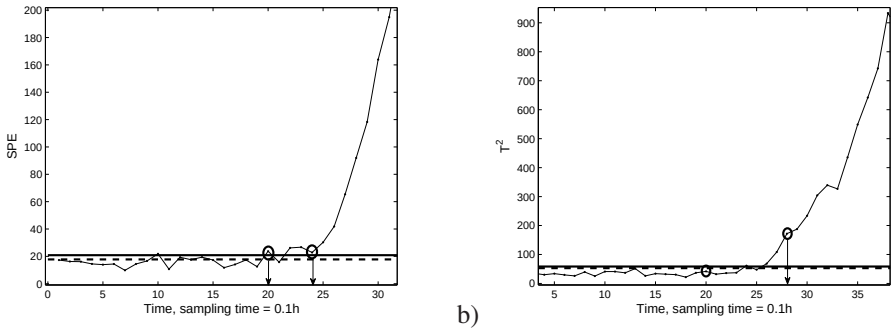
Table 9.4: Extended HAZOP analysis for deviation 1.

Key Variable	Threshold	Corrective actions	Seriousness
Inlet pipe rupture pressure	1.75* Pipe design Pr. = 2.62 bar	Bypass inlet flow to safety storage tank	3
Inlet pipe gas leak pressure	1.3* Pipe design Pr. = 1.95 bar	Pressure relief: switch on safety ventilation	2
Interface level for components concentration bias	0.165 m	Alarm to slop oil plant and downstream installations. Increase stripper reflux rate and steam flow.	2
Interface level for too high water flow	0.14 m	Re-arrange storage settings. Alarm to control level in intermediate storage tank	2
Drum level for high lights flow	0.26 m	Alarm to slop oil plant. Flow increase	2
Drum level for liquid venting and drum overflow	0.489 m	Shut down inlet flow to plant. Re-drive flow.	3
Drum leak pressure	1.4* Drum design Pr. = 2.1 bar	Break of pressure safe disc. Switch on safety ventilation	2
Cause	-	Solve upstream problems	1

Quantitative thresholds in Table 9.4 were established from equipment characteristics and operating specifications required in each stream. For example, the drum leakage pressure was set considering an increase of 40% on drum pressure from its nominal designed pressure (design pressure = 1.5 bar) whereas interface level (IL) for high concentration of heavy components in slop oil is fixed at the level at which such concentration would exceed 15% of the expected nominal concentration. The criteria for establishing these thresholds are often subjective and depend strongly on the knowledge of the equipments and the process (as in HAZOP analysis). In this case, thresholds were set by using general

engineering heuristics and known imposed process constraints.

Such quantitative analysis provides crucial information to propose an action protocol when deviation 1 is diagnosed. For instance, Fault 1 (see Table 9.2), corresponding to deviation 1, was introduced two hours after the plant was started up to check the response of the support system. The fault was tested by ramping the inlet mass flowrate as described in Table 9.2. The fault was detected by the PCA charts (SPE and T^2) when three consecutive samplings went off the established detection thresholds from the fault occurrence (sampling 20 in Figures 9.6a and 9.6b). Detection of an abnormal event was clear after 24 and 48 minutes (4 and 8 samplings after sampling number 20 in Figures 9.6a and 9.6b). In both figures the occurrence and detection times are remarked by circles (sampling interval corresponds to 0.1 h).



a) b)

Figure 9.6: SPE and T^2 plots under simulation of Fault 1. First circled sampling correspond to the fault occurrence whereas second one is the time at which the fault was detected. a) SPE plot, b) T^2 plot .

The rules base incorporated on the overall FDS allowed diagnosing univocally Fault 1 at the 32 sampling time (Figure 9.7). This diagnosis agreed with the ML-SVM diagnostic, thus providing a robust diagnosis performance. Note that, the possibility of Fault 1 occurrence suddenly increased at sampling time equals 31. Therefore, 72 minutes (12 samplings) after the Fault 1 occurrence, the unequivocal cause of the deviation is clear to the system. In parallel to the PCA model, the FLS diagnosis module was also able to detect some anomalies by drastically reducing the normal state occurrence possibility after 36 minutes from the occurrence of Fault 1 at the second operating hour.

Once the fault is diagnosed, most contributing process variables to SPE are clear (Vedam and Venkatasubramanian, 1999). Thus, the fault in its current magnitude as well as under the existing plant conditions can be simulated on the dynamic model to predict the future plant state. This simulation provides of valuable information to rank the CAs obtained in the extended HAZOP analysis.

CAs were ranked in the protocol based on the ratio of seriousness of each consequence/CA (last column of Table 9.4) and the time to reach its safety threshold. The higher seriousness to time to threshold ratio, the higher priority of the corresponding CA. These ratio results are shown for each CA in Table 9.5. Note that the time to the safety threshold of source fault causes was fixed at 2 hours. It allows to include source causes in the analysis

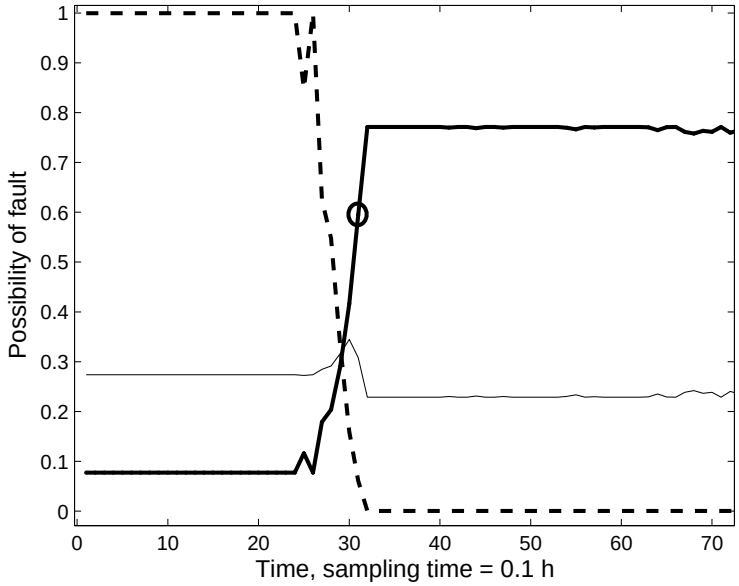


Figure 9.7: Fault 1 diagnosis results. The dashed line corresponds to the normal state. Bold line corresponds to Fault 1 occurrence possibility and the remaining line is the possibility of occurring Fault 2.

as an urgent problem to be solved but with a low seriousness (Table 9.4). This criterion allowed assigning high priority to solve causes when the consequences are far away in time whereas relegate it when seriousness and urgency of consequences are more relevant.

The protocol generated by the system is also shown in Table 9.5. The ratio of seriousness to the time to the associated threshold is a quantitative measure for sorting CAs. It is also worth noting that time to threshold given in Table 9.5 is the real time available for acting, that is, time from diagnosis to reach safety thresholds. It can be observed from the same table that any of the harmful consequences are far away in time. So, according to the seriousness/TT ratio, the system decides first to revise upstream problem in case it can be rectified in time to avoid any of the damaging consequences. In case upstream problems cannot be solved before next threshold is achieved, the next CA would be taken.

Figure 9.8 shows the key variables estimated by simulation of Fault 1. These variables forecasting were matched with their respective "Time to Thresholds" by the decision support system to decide (also considering the consequences severities) the most appropriated CAs protocol, which is given at Table 9.5. As shown in Table 9.5, there is a margin of more than 7 hours until any of the variables reach the earliest safety thresholds. This makes the system recommend first to solve fault source problems.

In order to further test the actual AEM-PHA integration and its benefits, the same fault occurring in other range of severity was applied to check the system flexibility. A more abrupt ramp was applied to increase inlet flow from 58.5 kg/h to 183 kg/h (slope = 12.45

Table 9.5: Protocol advised by the system facing Fault 1 under conditions 1.

CA protocol	Time to safety thresholds (TT)	Seriousness/TT	Consequences/ Causes
1. Revise and solve upstream problems	56.8 h	0.053	Inlet Pipe breakage
2. Alarm to slop oil plant	25.9 h	0.077	Inlet Pipe leakage
3. Alarm in storage tank. Revise tank settings	12.1 h	0.165	High decanter interface level (IL): products out of specification
4. Bypass inlet flow	8.0 h	0.25	High decanter IL: high water flow
5. Increase stripper reflux rate and steam flow	7.4 h	0.27	High decanter level (HDL): high impurities flow
6. Pressure relief: switch on safety ventilation	16.6 h	0.18	HDL: decanter overflow
7. Shut down inlet to plant. Re-drive flow	32.8 h	0.061	Decanter leakage
	2 h	0.5	Source causes

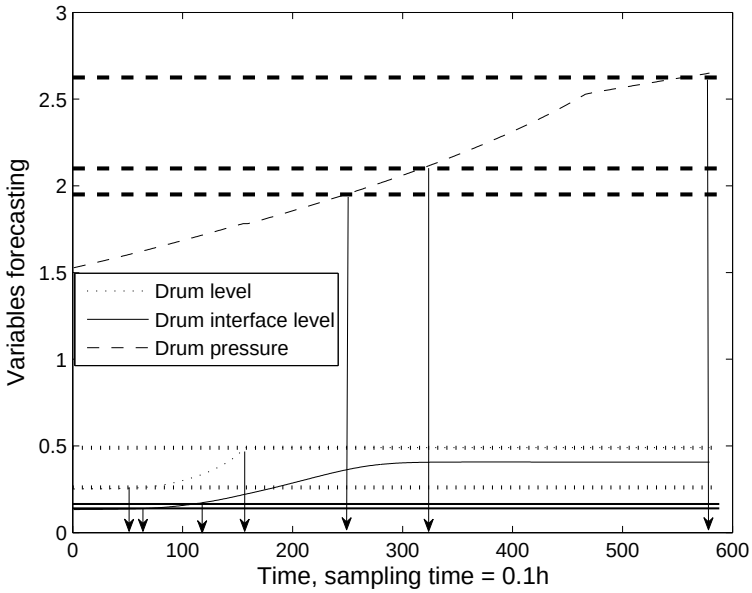


Figure 9.8: Key drum variable profiles under Fault 1. Bold straight lines represent threshold values.

kg/h^2). The abnormal event occurred at the second plant running hour and it was detected and diagnosed at the same time, 24 minutes after the fault occurrence. Note that under this new conditions Fault 1 was diagnosed 8 samplings before that in the first case (24 *vs.* 32 samplings). Figure 9.9 shows the univocal diagnosis of Fault 1 by the PCA-FLS at the

24^{th} sampling time.

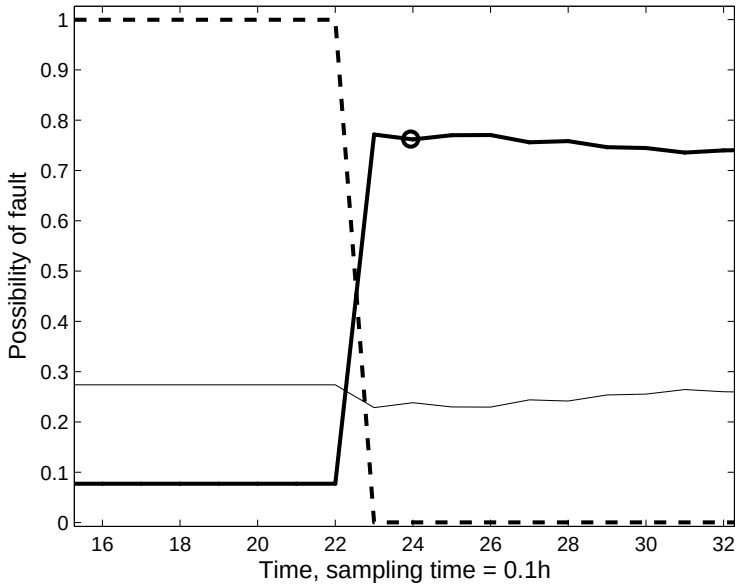


Figure 9.9: Fault 1 diagnosis results for a more abrupt ramp. The dashed line corresponds to the normal state. Bold line corresponds to Fault 1 occurrence possibility and the remaining line is the possibility of occurrence of Fault 2.

Under the same plant conditions but different fault intensity, the new protocol now faces first imminent plant harmful consequences due to the combination of urgency (thresholds occurring earlier) and seriousness, resulting in a different ordered protocol. In that way, the process operator is receiving different advices depending on the current plant conditions and diagnosed fault intensity. See Table 9.6 to check the new complete report as well as Figure 9.10 to see the expected key variables values on the following hours.

Now safety thresholds are much closer in time and the protocol recommends first to take straight CAs against forthcoming harmful consequences (compare the time scale of Figure 9.8 to the one in Figure 9.10). Thus, CAs are now firstly focused on solving urgent plant safety and economic problems in detriment of repairing the fault causes.

It should be also outlined the different dynamic repercussions of a more intense fault due to the dynamic equipment characteristics. In contrast with Table 9.5, now (Table 9.6) *Decanter leakage* will occur before that *Drum overflow* (4.3 *vs*. 4.8 h. instead of 32.8 *vs.* 16.6 in Table 9.5), however seriousness differences of both consequences, 2 and 3 respectively, make the system first try to solve problems related to "Drum overflow", thus placing *By-pass inlet flow* before than *Pressure relief: switch on safety ventilation* in the final CAs protocol. It is also worth to remark that new inverse order of CAs *Increase stripper reflux rate and steam flow* and *By-pass inlet flow* (CAs 5 and 4 in Table 9.5 and 3 and 4 in Table 9.6) is due to the plant dynamics and the explained sorting criteria.

Table 9.6: Protocol facing Fault 1 under second fault conditions.

CA protocol	Time to safety threshold	Seriousness/TT	Consequences/ Causes
1. Alarm to slop oil plant	9.5 h	0.316	Inlet Pipe breakage
2. Alarm in storage tank. Revise tank settings	5.6 h	0.357	Inlet Pipe leakage
3. Increase stripper reflux rate and steam flow	3.1 h	0.645	High decanter interface level (IL): products out of specification
4. By-pass inlet flow.	1.4 h	1.428	High decanter IL: high water flow
5. Revise and solve upstream problems	1.1 h	1.818	High decanter level (HDL): high impurities flow
6. Pressure relief: switch on safety ventilation.	4.8 h	0.625	HDL: decanter overflow
7. Shut down inlet to plant. Re-drive flow.	4.3 h	0.465	Decanter leakage
	2 h	0.5	Source causes

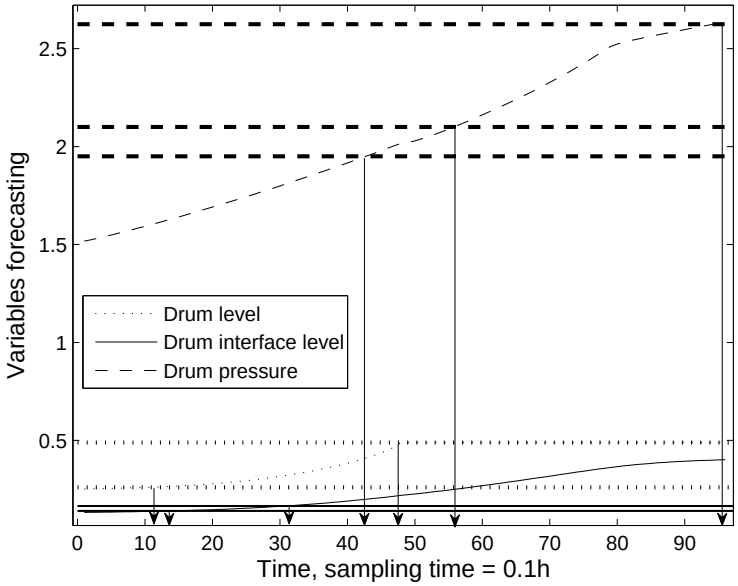


Figure 9.10: Key drum variable profiles under Fault 1 and more intense severity. Bold straight lines represent threshold values.

9.5.4 Deviation 2: High accumulator level

A second deviation was tested in another crucial point of the plant to extend and generalize the obtained results. In this case, an increase on overhead accumulator's level due to a level controller malfunction (Fault 2 on Table 9.2) was simulated to test the decision support system. The extended quantitative HAZOP analysis regarding this deviation is

summarized on Table 9.7.

Table 9.7: Extended HAZOP analysis for deviation 2.

Key Variable	Threshold	Corrective actions	Seriousness
Overacum leak pressure	1.4 *Design pressure = 2.91 bar	Pressure relief: switch on safety ventilation	3
Level for liquid venting	0.40 m	By pass vapor flow by secondary condenser	3
Overacum level to overflow	0.47 m	Open drain to secondary storage tank	3
Flow in pump to cavitation	< 1.8 kg/h	Close stripper reflux and maintain stripper on basis conditions	3
Cause	-	Solve problems related to causes	1

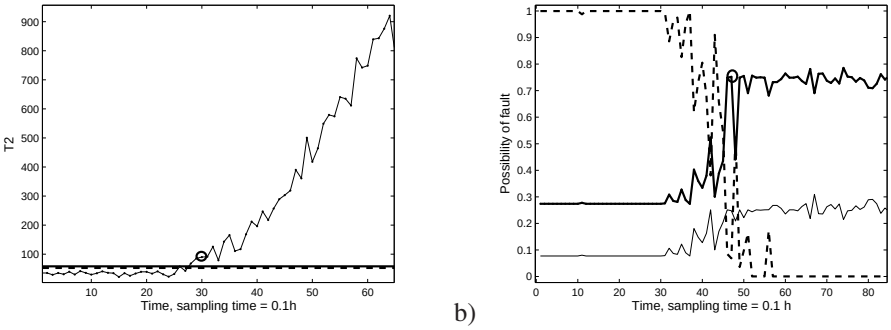
Note that not all the consequences present in the original analysis were considered. The main reason for doing this is that depending on the fault cause, some consequences can come true and some others not. A malfunction in the controller resulting in a progressive closing of the output valve, might lead to cavitation problems at the downstream pump due to lack of flow, but this malfunction will never produce damages on the pump because of high pressures, given that it induces lower pressures over the downstream line (see consequences of original HAZOP analysis for deviation 2 in Appendix C). The design pressure of the overhead accumulator tank was 2.08 bars.

It is also important to point out that all consequences are assumed to be highly severe in that case what will result on protocols just depending on the time to reach each safety threshold. This knowledge is made available to the decision support system. Then, the fault was simulated and quickly detected and diagnosed. Figure 9.11a and 9.11b shows T^2 and the diagnosis charts respectively when Fault 2 occurs at the second operating hour. Fault 2 was firstly detected and then diagnosed after 1 hour and 2.7 hours from the fault occurrence, respectively.

After a robust diagnosis (two consecutives samplings by the global FDS), the decision system determines fault's magnitude from past data and forecasts a faulty ramp on the accumulator's level with a slope of 0.00524 m/h. Then predictions of the plant dynamic model allow the system to generate the CAs protocol written on Table 9.8.

The CAs are ranked in a protocol (as shown above) by checking the seriousness/TT ratio. The dynamic model key variables forecasting are depicted in Figure 9.12. This figure shows how the overhead accumulator's level is the first variable that crosses the permitted thresholds by reaching the venting liquid threshold, then pressure exceeds the limit causing leakage in the tank, and so on.

In order to check the dynamic capabilities of the proposed framework, two additional runs of Fault 2 were tested. The same fault was applied but under more severe condi-



a) b)

Figure 9.11: T^2 and diagnosis plots under simulation of Fault 2. Circled samplings correspond to the detection and diagnosis times respectively. a) T^2 plot, b) diagnosis plot.

Table 9.8: Protocol advised by the system facing Fault 2 under conditions 1.

CA protocol	Time to safety thresholds	Seriousness/TT	Consequences/Causes
1. Solve problems related to causes	34.6 h	0.0867	Venting of liquid into gases outlet
2. By pass vapor flow to secondary condenser	39.2 h	0.076	Overacum leakage
3. Pressure relief: Switch on safety ventilation	41.1 h	0.0729	Overacum overflow
4. Open drain to secondary storage tank	> 42.3 h	< 0.0709	Pump Cavitation
5. Close stripper reflux and maintain stripper on basis conditions	2 h	0.5	Cause

tions. Thus, faster increase on accumulator's level showed different system advices to the process operator. These new fault variations were equally diagnosed at 1.1 h and 1.7 h from the fault occurrence respectively, and caused two different sorted CAs protocols. It is important to point out that Fault 2 was identified now much sooner; 1 h before than the initial Fault 2 test for this new first case; and 1.6 h before in the most intense case. Protocols generated in each case are shown in Tables 9.9 and 9.10, being the variables forecasting depicted in Figures 9.13 and 9.14 respectively.

Notice that firstly, new abnormal event gives enough time margins (Figure 9.13 and Table 9.9) to recommend first solving source causes, whereas new second case, under the most severe faulty conditions, recommend first solve cavitation problems (Table 9.10). Different protocols are given in each case based on the future evolution of variables due to the process dynamics. Thus, a more abrupt ramp (Figure 9.14 and Table 9.10) closing the outlet overhead accumulator's valve implies having less flow than the required at the pump before having leak problems because of high pressure in the overhead accumulator. On the other hand, a less abrupt ramp causes an alteration in this order (Figure 9.13 and Table 9.9).

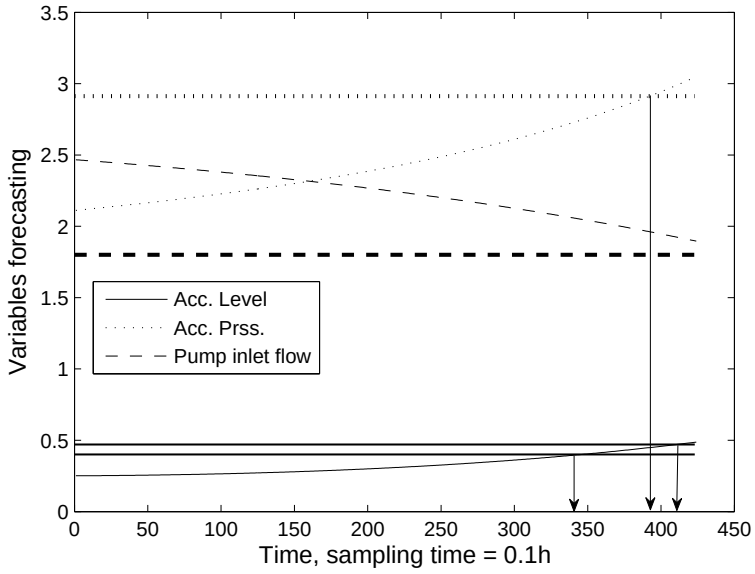


Figure 9.12: Accumulator key variables forecasting under Fault 2. Bold straight lines represent variables threshold values.

Table 9.9: Protocol advised by the system facing Fault 2 under more severe conditions.

CA protocol	Time to thresholds	Seriousness/TT	Causes/Consequences
1. Solve problems related to causes	25.6 h	0.117	Venting of liquid into gases outlet
2. Pressure relief: Switch on safety ventilation	23.4 h	0.128	Overacum leakage
3. By pass vapor flow to secondary condenser	26.0 h >	0.115 <	Overacum overflow
4. Close stripper reflux and maintain stripper on basis conditions	26.0 h >	0.115 <	Pump Cavitation
5. Open drain to secondary storage tank	2 h	0.5	Source cause

9.6 Conclusions

The chapter presents a framework for the integration of Abnormal Events Management (AEM) and Process Hazard Analysis (PHA) which enhances on-line plant safety. The framework consists of a decision-making module which uses an off-line HAZOP analysis extended in quantitative terms, a fault diagnosis system (FDS) and a plant dynamic model, to generate a dynamic corrective actions (CA) protocol. The protocol is based on those CAs extracted by the HAZOP analysis which are on line sorted by the supporting

Table 9.10: Protocol advised by the system facing Fault 2 under the most severe conditions.

CA protocol	Time to thresholds	Seriousness/TT	Causes/Consequences
1. Close stripper reflux and maintain stripper on basis conditions	> 6 h	< 0.5	Venting of liquid into gases outlet
2. Pressure relief: Switch on safety ventilation	5.8 h	0.517	Overacum leakage
3. Solve problems related to causes	> 6 h	< 0.5	Overacum overflow
4. By pass vapor flow to secondary condenser	5.6 h	0.535	Pump Cavitation
5. Open drain to secondary storage tank	2 h	0.5	Source cause

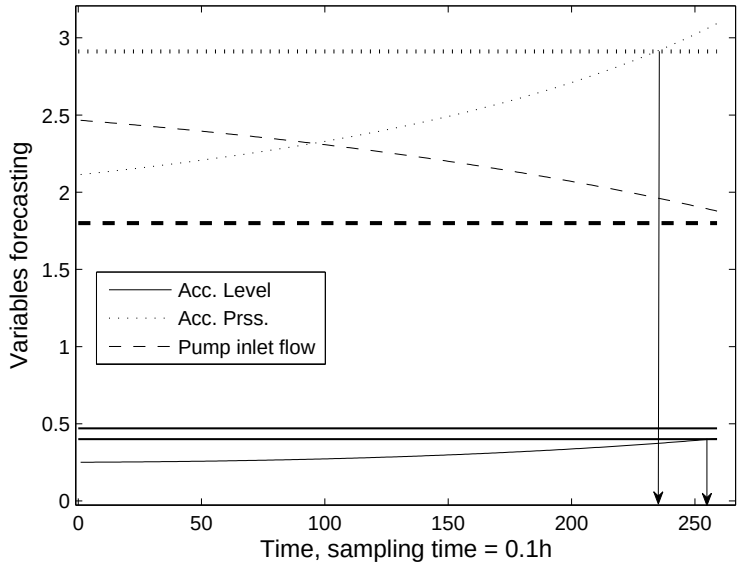


Figure 9.13: Accumulator key variables forecasting under Fault 2 and conditions 2. Bold straight lines represent variables threshold values.

system depending on the current diagnosed fault and on the current plant conditions.

A sour water stripper plant was used to show the supporting system performance. An extended HAZOP analysis was carried out based on an original HAZOP qualitative analysis provided by academy and industry (Vaidhyanathan, 1995). The case study was simulated to allow the decision support system to have knowledge about future variables values under different conditions. The decision support system response was tested facing two faults under different bias magnitudes. The proposed methodology, after detecting and di-

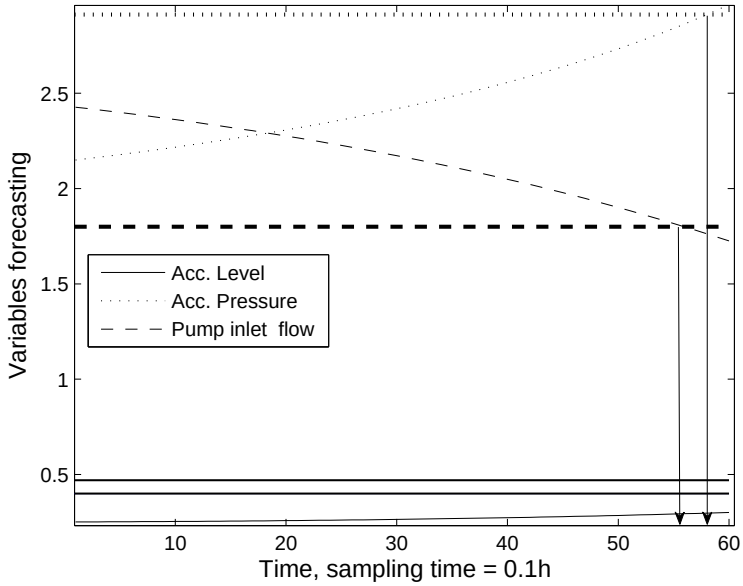


Figure 9.14: Accumulator key variables forecasting under Fault 2 and conditions 3. Bold straight lines represent variables threshold values.

agnosing each fault, proved to be able to efficiently suggest different CA protocols adapted to the most urgent necessities in each case, thus on-line supporting the process operator.

Nomenclature

$CSPE$	Contribution to Square Prediction Error
SPE	Square Prediction Error

Subscripts

j	Causes-Consequences of the HAZOP analysis
z	Deviations of the HAZOP analysis

Acronyms

AEM	Abnormal Events Management
CA	Corrective Actions
COM	Component Object Model
FDS	Fault Diagnosis System
FLS	Fuzzy Logic System
HAZOP	Hazard and Operability
ML-SVM	MultiLabel Suppport Vector Machines
PHA	Process Hazards Analysis
PCA	Principal Component Analysis
PSM	Process Safety Management
SWS	Sour Water Stripper

CHAPTER 10

An Integrated Framework for On-Line Supervised Optimization

The AEM proposal described until now can be integrated with other plant information modules to obtain benefits on the overall plant control. In this chapter the AEM supports the application of an on-line plant improvement algorithm (Real Time Evolution). By this support a higher decision-making level is highlighted. Thus, different decisions are taken depending on the disturbances nature, that is, the seriousness of the plant disturbance. In this sense, not only operative and safety constraints are taken into account in the decision-making process but other crucial issues as the plant economy are also considered.

10.1 Introduction

Current situation of the process industry with growing competitiveness, increase of costs and stricter environmental constraints, makes optimization a crucial issue to obtain the maximum profitability with available resources. Furthermore, process abnormalities have significant impact, in such a way that hundreds of billion of dollars are lost by the industry due to poor abnormal situation management (Huang et al., 2000). Therefore, production efficiency and consistency become essential to success.

An optimal off-line design will not lead the plant to operate at its optimum yield permanently. Most of processes usually work under continuously changing conditions over time, being the operation at optimal condition a continuous challenge in the everyday plant work. Real time optimization has attracted much attention in the past because of its capacity for improving plants operating profit beyond conventional process control (Cutler and Perry, 1983; Marlin and Hrymak, 1996; White, 1998). As Marlin and Hrymak (1996) reported, on-line optimization is specially suitable for processes that are frequently disturbed, where adjustable variables remain being significant for the process economy after addressing higher priority objectives.

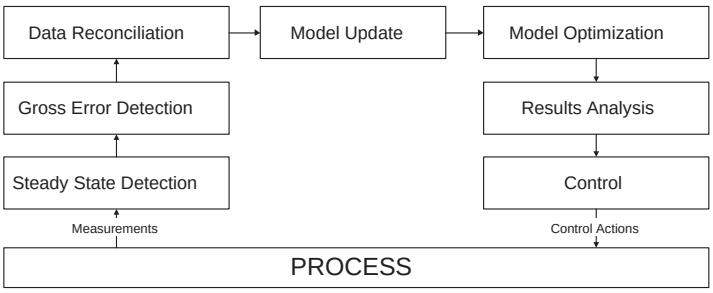


Figure 10.1: Real time optimization closed loop.

There are two typical approaches for on-line optimization: direct search methods and model-based methods. Direct search methods (García and Morari, 1981; McFarlane and Bacon, 1989; Xiong and Jutan, 2003) are recommended when difficulties arise in obtaining either first principle or empirical process models. These methods track the optimum trajectory from plant measurements (Pinto, 1998). In this chapter, attention is focused on the model based optimization methods since they correspond to the most complex but powerful approaches. Accurate models will produce better performance but requiring longer computation times (Yip and Marlin, 2004).

The closed loop for the real time optimization shown in Figure 10.1 starts when process measurements are collected and validated. The validation of the measurements usually includes gross error detection and data reconciliation (see chapters 2 and 4). Validated measurements are employed to update the model parameters reducing the structural mismatch between the model and the plant. Since data reconciliation and parameter estimation are often based on a steady state model, it is important to ensure that the plant is already in or close to the steady state. Then, the on-line optimizer calculates the optimum operating policy. Optimization is always affected by measurements noise and measured or unmeasured plant disturbances, so an on-line statistical analysis (Miletic and Marlin, 2004) will be required to decrease the frequency of unnecessary set-points changes, hence increasing the plant performance. Finally, the optimal set points are implemented by the control system.

Real Time Evolution (RTE) introduced by Sequeira et al. (2002) establishes a new approach for on-line optimization that overcomes the drawbacks reported for the classical RTO approach. According to Friedman (1995), the steady state data required in a common dynamic environment needs heavy filtering and leads to long waiting. Furthermore, detailed process models are demanded whereas a proper trajectory for the implementation of the manipulated variables to reach the optimum is usually not provided. On the other hand, RTE responds immediately to disturbances without waiting for steady state by a continuous adjustment of the set-point values based on a steady state model.

Nevertheless, this approach might be quite unrealistic and not worthy in a real scenario, because this improvement algorithm cannot be launched directly when a fault or disturbance is detected. An incorrect or null identification of the root causes of process upsets will lead this method to inappropriate response and the subsequent incorrect performance of the plant. Therefore, the implementation of a robust and reliable AEM system that

manage the plant response against any process upset is of crucial importance.

The aim of this work is to demonstrate the efficiency and robustness of a supervision module that combines optimization and AEM concepts leading to a realistic scheme that copes with the RTE previous reported limitations. It provides relevant coupled and timely information for plant managers related to safety and economical issues to improve their decision-making. This new knowledge can be also used to the better design or correction of new or already installed processes.

The general optimization methodology applied in this work (RTE) is revised in section 10.2. Section 10.3 focuses on the advantages of integrating an AEM system with the RTE approach. The proposed system framework is described in this part of the work. A case study is presented in section 10.4 and the results obtained applying the proposed methodology are listed in section 10.5. Finally, the chapter ends up at section 10.6 concluding and remarking possible future work.

10.2 Real Time Evolution (RTE)

Real Time Evolution (Sequeira et al., 2002) is an improvement rather than an optimization algorithm. It responds quickly to disturbances by a continuous adjustment of the decision variables according to the current plant conditions. RTE is supported by a steady state model which successively identifies the improvement direction, once given an objective function or performance criterion. Faster and repeated executions of RTE, which may deal with continuous disturbances (since it does not need to reach the steady state), results in an very adequate way of improving process operations. Data reconciliation and model updating are only carried out when the steady state is reached.

RTE relies on four main aspects: the improvement algorithm, the neighborhood, the time between successive executions and the steady state process model. The improvement algorithm selects the best operating conditions in the neighborhood of the current operating point. The neighborhood is defined by the maximum permitted change on the decision variables. Limits for such changes are given by the distributed control system.

In contrast with RTO that deal with a normally very complex formal optimization problem, RTE easily discards infeasible solutions within the improvement algorithm reducing the computational demands. Besides, RTE is much simpler since it uses steady state model composed by algebraic equations as a tendency model. It is worth remarking that, under a similar disturbance pattern, different RTE frequencies can be applied resulting in different decision variables profiles. Therefore, the correct application of RTE requires of an appropriate parameter tuning (Sequeira et al., 2002)

10.3 Integrated supervised optimization

RTE may be incorporated into the general AEM system described along this thesis. By this way, RTE is an additional supporting tool of the plant AEM system. On the other hand, such combination makes the technique applicable in real scenarios, since it is launched intelligently by a decision-making system giving rise to a novel concept: "Su-

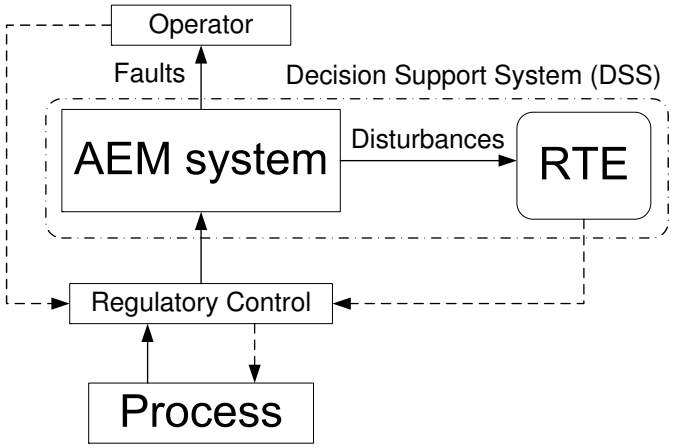


Figure 10.2: Decision Support System (DSS) information flow.

pervised Real Time Evolution" (SRTE). In this way, the methodology practically demonstrates how AEM can be integrated with other information plant modules to obtain common benefits for the global plant performance (see framework outline on Figure 10.2).

The resulting AEM system (including RTE) is the core of the on-line response system against any disturbance or deviation from the normal plant behavior. It does not only favor the overall process safety and reliability but also the process economy. The system organizes the global response of the plant classifying incidences into two main groups:

Disturbances: Incidences affecting the profit of the plant and susceptible to operate under on-line optimization by the RTE or another optimization algorithm. A disturbance, as is it considered here, alters the normal performance of a process without affecting its safety. In this sense, it establishes a particular case from the definition given at section 2.3.

Faults Incidences that affect the safety and good performance of the plant. This description is also a particular view of the general definition introduced at section 2.3. These faults are established in a previous analysis of probable and dangerous possible plant incidences. The AEM system may generate different responses depending on the fault diagnosed (see chapter 9). For instance, a set of corrective actions (CAs) may be activated, to prevent more critical situations, by launching different alarms and advising operators in the way they should act against this abnormal events.

In that sense, the AEM system advise the most convenient action at each moment. If deviations from normal operating regime are not classified as one of the known studied faults, the system will trigger the optimization procedure. Otherwise, a CAs protocol based on off-line and on-line process knowledge (chapter 9) will be proposed as advice to the operator. Figure 10.2 shows the Decision Support System (DSS) information flow.

Incidence detection is carried out by means of the principal component analysis (PCA) module included on the overall FDS (chapter 8). Incidence symptoms coming from process variables are classified by the ML-SVM and PCA-FLS integrated system described at chapters 6, 7 and 8. Data sets corresponding to the most probable faults occurring in

the plant, are tested off-line or taken directly from historical data bases to extract enough information for training the global FDS. Therefore, by the continuous on-line AEM, the system classifies regularly whether data received from plant corresponds to normal behavior or an abnormal situation. When an abnormal event arises, the system determines if this incidence is critical for the good performance of the plant (fault) or if the plant is susceptible to be optimized (disturbance).

In case a disturbance is detected and not recognized as dangerous in safety terms (fault), RTE is automatically activated by the AEM system. The elapsed time since the detection and the activation of the RTE must be optimized as a trade-off between the benefits gained by an early reaction of RTE and the confidence on a reliable detection and diagnosis. Then, current conditions from plant are collected and applied to the steady state model. With this updated model, the improvement algorithm explores the surroundings of the current decision variables values evaluating the resulting objective function. The best objective function value is chosen and the corresponding set points are immediately applied to the plant through the control system. Then, this mechanism is repeated every sampling time. When the plant reaches a new steady state with new optimum operating conditions, and no disturbances occur, the value of the proposed decision variables values will not change and no action will be taken. At that moment, the FDSs must be adapted to operate under new normal operating conditions to continue supervising the plant robustly.

In case a fault is diagnosed, the AEM system warns the operator, as well as suggests proper course of action. Depending on the nature of the abnormal situation, it proposes an on-line adjusted protocol consisting of corrective actions to be applied to the process (review chapter 9).

10.4 Case study: Debutanizer column

The framework described above was implemented using Matlab and HYSYS.Plant. A Matlab script acts as a data manager establishing the communication between the plant (HYSYS.Plant in dynamic mode), the model (HYSYS.Plant in steady state mode), the optimization algorithm (Matlab), and the AEM decision system (Matlab). Communication between Matlab and HYSYS.Plant is executed by means of COM technology. Matlab creates a COM automatic server which interacts with the objects exposed by HYSYS.Plant.

The debutanizer column provided by Aspentech at its web-site documentation (AspenTech, 2004) was chosen to make the results easily reproducible. This multi-component distillation column has fifteen stages and is fed by two streams consisting of a mixture of light hydrocarbons. The flowsheet for this case study is depicted in Figure 10.3. The operational objective of this unit is the recovering of the butane and lighter hydrocarbons from these feed streams.

An off-line analysis of the distillation column allowed to determine the most probable incidences while the plant is running at nominal operating conditions. Some of these possible abnormal situations are listed in Table 10.1 together with a brief description of each one.

Firstly, the column was simulated in HYSYS.Plant at its steady state mode. That is the steady state model used by the RTE algorithm. Secondly, several modifications were ap-

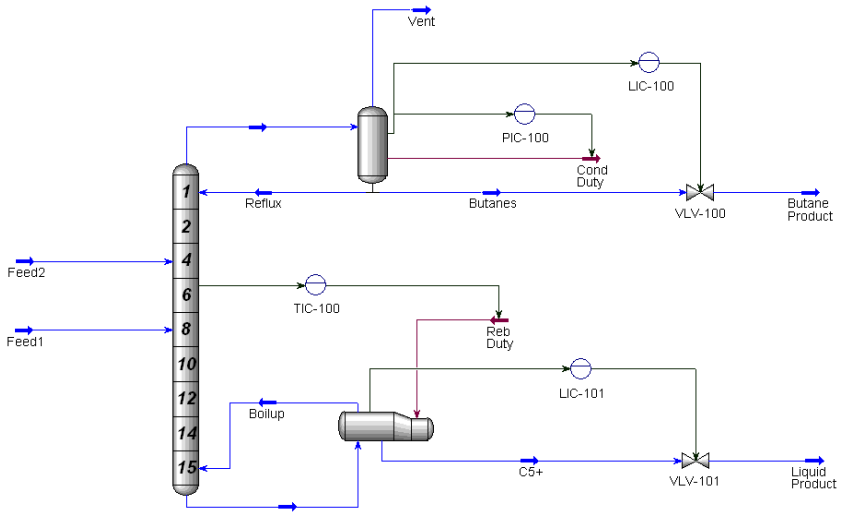


Figure 10.3: Debutanizer column flowsheet.

Table 10.1: Off-line analysis of the incidences.

Incidence	Description	Type
I1	Feed1 Flowrate Step -20% Fall	Disturbance
I2	Feed1 Flowrate Step +20% Rise	Disturbance
I3	Feed1 Temperature +5% Step	Disturbance
I4	Feed1 Flowrate Ramp -0.0072% Fall	Disturbance
I5	Feed2 Flowrate Step +20% Rise	Fault

plied to the model to build the dynamic model version including the control mechanism which consisted of two level controllers for condenser and reboiler, and a column temperature controller which measures temperature of the sixth plate. The control strategy was taken directly from the original case study while the controllers tuning was made using the PID auto-tuning function of HYSYS.Plant. These parameters and their respective tuned values are shown in Table 10.2. This dynamic model was used as the real plant process.

Table 10.2: Parameters for the PI controller.

Controller	LIC-100	LIC-101	PIC-100	PIC-101
Kc	1.80	2.00	2.00	5.00
Ti, min	10.00	10.00	2.00	20.00

The following instant objective function (equation 10.1) was proposed in order to evaluate the plant performance at each sampling time. It accounts for the difference between products sales, and raw material and energy requirement costs:

$$\begin{aligned} &Instant\ Objective\ Function(t) = \\ &+ (PartialFlowrate\ of\ C4\ \&\ lighter\ in\ Butane\ Product) \cdot (Price\ of\ Butane\ Product) \\ &+ (PartialFlowrate\ of\ C5+\ in\ Liquid\ Product) \cdot (Price\ of\ Liquid\ Product) \\ &- (PartialFlowrate\ of\ Feed1) \cdot (Price\ of\ Feed1) \\ &- (PartialFlowrate\ of\ Feed2) \cdot (Price\ of\ Feed2) \\ &- (Condenser\ Heat\ Duty) \cdot (Price\ of\ Condenser\ Heat) \\ &- (Reboiler\ Heat\ Duty) \cdot (Price\ of\ Reboiler\ Heat),\ \ m.u./time \end{aligned} \tag{10.1}$$

Raw material prices and product selling costs were taken to formulate a representative objective function. Prices and costs are shown in Table 10.3.

Table 10.3: Raw material, products and energy prices.

Feed1, m.u./kg	Feed2, m.u./kg	Butane P., m.u./kg	Liquid P., m.u./kg	Cond. Heat, m.u./kJ	Reb.Heat, m.u./kJ
2.00	2.00	3.00	3.00	0.00009	0.0005

Different on-line procedures cannot be compared in terms of an instant objective because the value of an instant benefit does not show which one produces better overall results. So, a mean objective function, taking into account the accumulative produced profit, was formulated (equation 10.2). This objective function was used to check and compare:

- RTE working independently without any kind of supervision.
- The novel supervised RTE and the classical RTO actions.
- The case in which no action is considered.

$$Mean\ Objective\ Function(t) = \frac{\int_{t_o}^{t} IOF(t)dt}{t - t_o} \tag{10.2}$$

10.5 Results and discussion

RTE was applied to the described distillation column to maximize its operating profits by means of adjusting two decision variables that impact the process economy. These adjustable variables are:

1. The set point of the 6^{th} plate temperature controller.
2. The reflux rate of the column

RTE requires a parameter tuning depending on the studied scenario. In this case, after a tuning analysis, a maximum allowed change in set points of 0.3% around the old value and 50 seconds between consecutive executions were considered demonstrating good results. Prior to the RTE activation, correct fault diagnosis must be assured. Data from plant are collected each 150 seconds (supervision sampling time). By analyzing the SPE and T^2 statistics an abnormal behavior is detected when three consecutive samplings are above the detection threshold. This threshold were calculated from PCA model and represents the percentile 99 of the statistics normal data distributions. Once the incidence is detected, data are processed by the global FDS in order to isolate the current situation. Two equal consecutive diagnostics were required for stating a correct diagnosis response, thus, adding a new sampling until the incidence is finally diagnosed. Through this methodology, all incidences shown in Table 10.1 were identified as fault or disturbances (see section 10.3).

Furthermore, for comparison purposes, the classical RTO algorithm was also implemented. The steady state detection was done in a simple way by reviewing the variables variance. The model optimization was carried out solving a successive quadratic programming problem. This method reproduces Newton's method for constrained optimization. At each iteration, an approximation of the Hessian of the Lagrangian function using a quasi-Newton updating method is made. This is used to generate a quadratic programming subproblem which solution is used to lead the search by a lineal searching procedure. Next, different cases are presented to show the proposed framework performance.

10.5.1 Incidence 1: Step fall in Feed 1 stream flow

Firstly, a step fall (-20%) in Feed1 mass flow was simulated and later diagnosed. This incidence is caused by an upstream problem and cannot be immediately addressed. This fact encouraged the system to launch the on-line optimization until normal conditions are restored.

Once the disturbance was diagnosed SRTE was automatically activated. While SRTE early reacts against the disturbance, RTO waits until steady state is reached to act. This SRTE faster response regarding classical RTO is translated into a better mean objective function. Figure 10.4 shows these mentioned responses. Note that RTO reacted much later (at a simulation time of 13200 seconds approximately) than SRTE, hence, losing benefits during the transition to the new steady state.

Focusing on the bottom graph at Figure 10.4, it can be easily appreciated how SPE sharply increases when the disturbance appears. After the fourth consecutive sampling upper the detection threshold, SRTE was correctly activated. The horizontal solid line stands for the detection threshold while the dashed line corresponds to the percentile 95 of the normal operating data. SPE after fault detection, shows a decreasing trend up to getting to a new steady state. It was originated by getting process variables correlation back by the achievement of the new steady state.

Furthermore, two different RTO responses were tested and shown in Figure 10.5 in order to check an interesting stability consideration. In the first case, a trajectory for the optimal set points implementation is not provided, so the new set-points are directly applied to the controllers. This direct set points implementation shows the way real time optimizers implement new set points by neglecting the optimal trajectory that should be

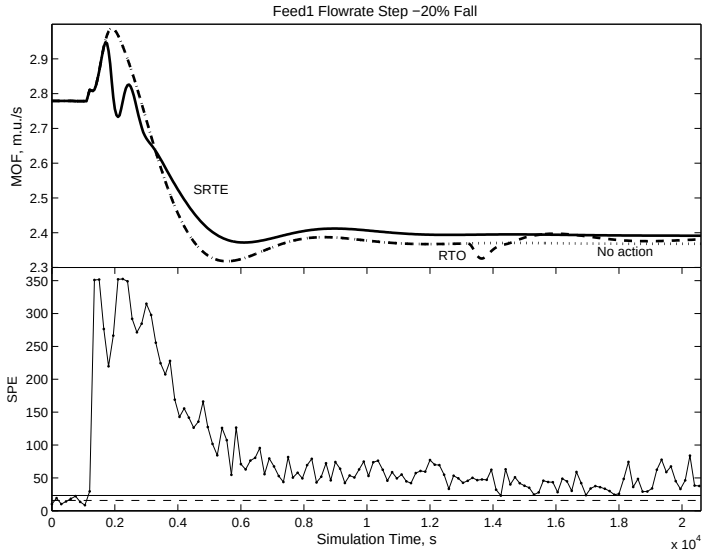


Figure 10.4: Incidence 1 - MOF and SPE representation.

implemented. This sudden change causes a high disturbance in the plant which also alters the value of the instant objective function (peaks on Figure 10.5). In the second case, optimal set-points were progressively implemented by a continuous ramp. This strategy was automatically translated into a softer instant objective function rise and consequently into a more stable plant behavior.

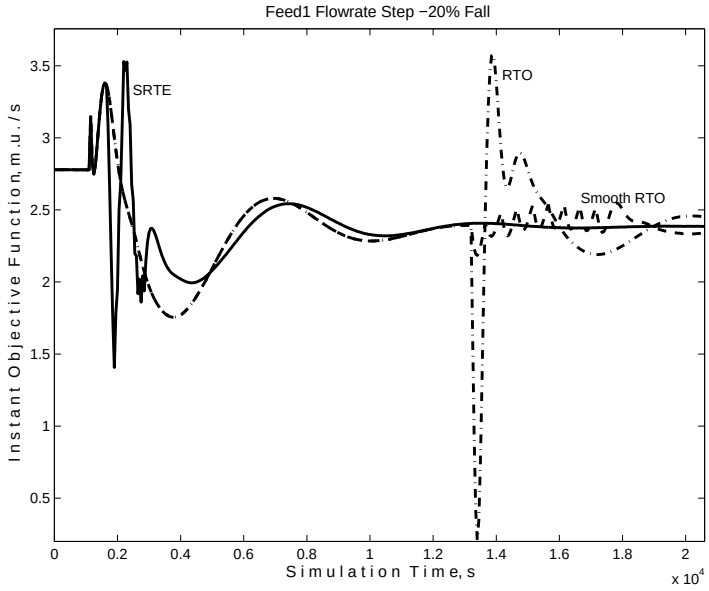


Figure 10.5: Incidence 1 - RTO's comparison (IOF).

10.5.2 Incidence 2: Step rise in Feed 1 stream flow

In this case, a step rise (+20%) in Feed1 flow was detected by the FDS not being recognized as a dangerous incidence. Unexpectedly, the disturbance produced by itself an improvement in the process objective function (Figure 10.6). The SRTE response against the disturbance was similar as that shown on the previous one. It prepared the system for the future, even though losing benefits at the beginning. This is because the SRTE algorithm is supported by a steady state model which provides the best decision variable search direction at the steady state.

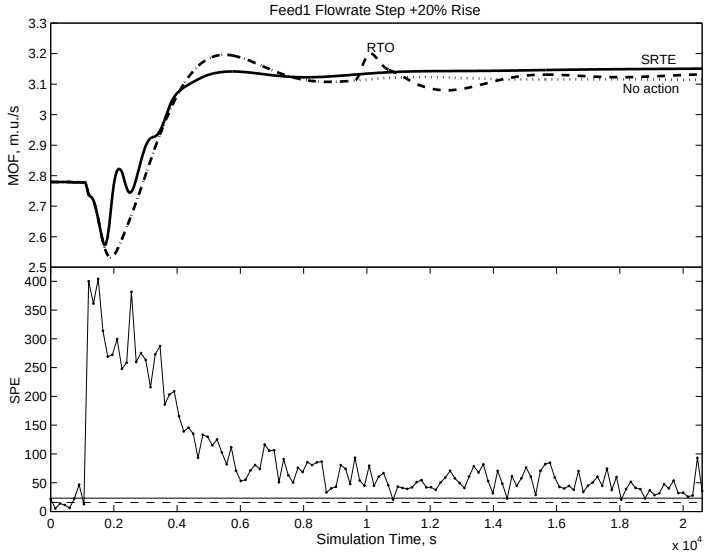


Figure 10.6: Incidence 2 - MOF and SPE representation.

Figure 10.6 demonstrates SRTE obtained better mean objective function values. Figure 10.7 shows how SRTE progressively modifies both decision variables in contrast with RTO performance. As depicted in Figure 10.7, inertia is more noticeable when tracking the temperature set point on the 6^{th} column tray.

10.5.3 Incidence 3: Step rise in Feed 1 stream temperature

This disturbance allows a smaller chance for optimization, since the variation in the objective function is less remarkable, so the on-line optimization finds less chance to take advantage of the new conditions. This is because the temperature of the feed is not directly related with the objective function. Figure 10.8 depicts the mean objective function trend when applying the SRTE, the classical RTO and the result when no action is taken. Classical RTO had to wait to the achievement of the new steady state. Then, the system directly applied the calculated set-points to the plant controllers. That explains the peak on the RTO curve.

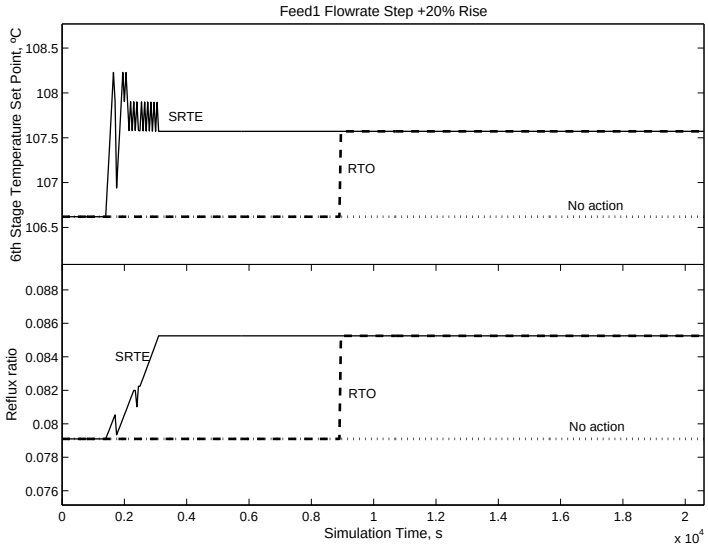


Figure 10.7: Incidence 2 - Decision variables representation.

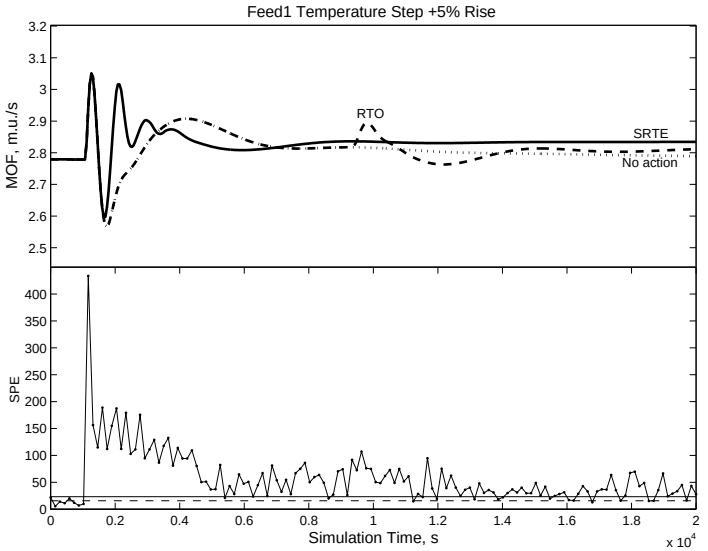


Figure 10.8: Incidence 3 - MOF and SPE representation.

10.5.4 Incidence 4: Continuous ramp fall in Feed 1 stream flow

In case a continuous disturbance arises on Feed1, the application of classical RTO becomes infeasible since the plant does not reach a steady state during the studied period of time. This weakness outlines one of the main SRTE strengths, since SRTE is able to operate under conditions in which a steady state is not achieved quickly.

In order to check that issue, a ramp fall in Feed 1 (-0.0002 kg/s) was simulated. Therefore, the obtained SRTE results under the incidence could only be compared to those when no action was carried out. Figure 10.9 represents the mean objective function for the considered simulation time. It can be seen how SRTE is triggered when the diagnosis is completely confirmed by the fourth consecutive sampling over the detection limit (solid horizontal line at the bottom graph of Figure 10.9). Figure 10.10 shows the evolution of both instant objective functions.

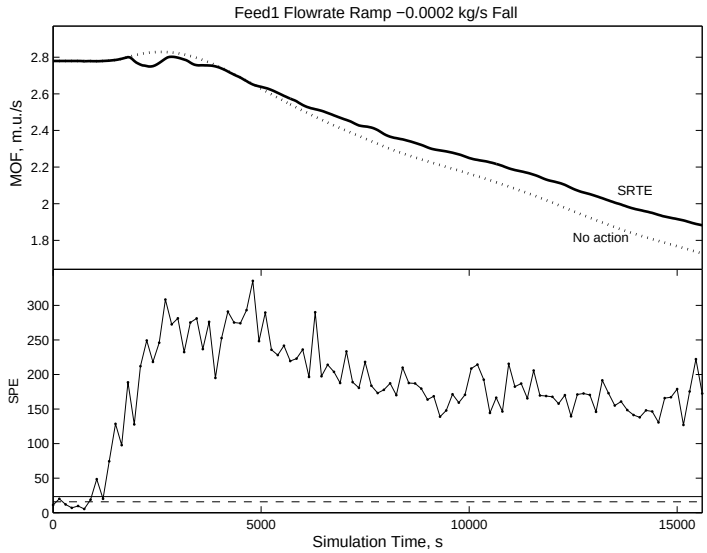


Figure 10.9: Incidence 4 - MOF and *SPE* representation for a continuous disturbance.

Despite instant objective function decreases in both cases, the SRTE curve is permanently over the "no action" curve since RTE continuously rectifies the plant performance. Figure 10.11 reflects how SRTE tracks the decision variables in order to accommodate the continuous disturbance.

10.5.5 Incidence 5: Step rise in Feed 2 stream flow (Fault)

In the considered distillation column, a rise of 20% in the Feed2 mass flow was considered as a dangerous incidence, this is, a fault. When the AEM system isolates the incidence, it proposes a CAs protocol which first measure advises to by-pass the feed flows at the time that changing the operational regime to column total reflux. A message advises the operator to follow the mentioned CAs until the break-down is mended. Figure 10.12 shows the evolution of the outlet and inlet streams in the column for this incidence.

When the increase on Feed2 flowrate was diagnosed, CAs were automatically proposed to the process operator who puts them into practice. Thus, column inlet flows were bypassed to another section of the process and total reflux was imposed in the column. These actions were translated to the instant objective function as can be seen in Figure 10.13.

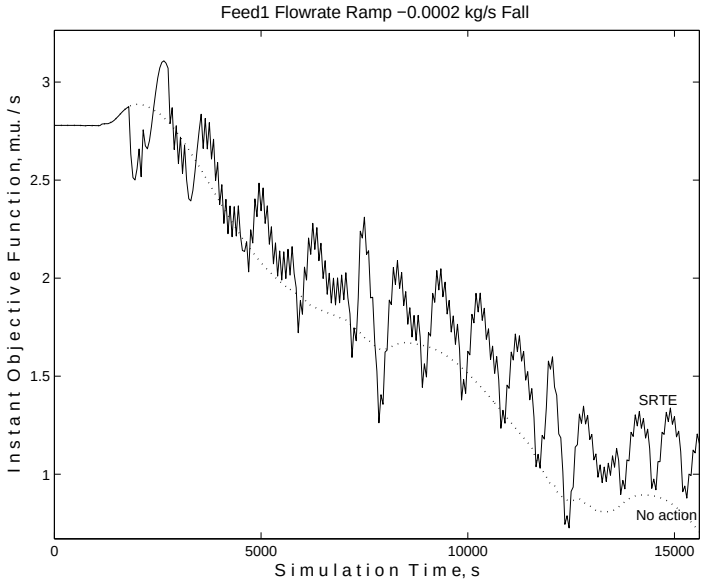


Figure 10.10: Incidence 4- IOF for a continuous disturbance.

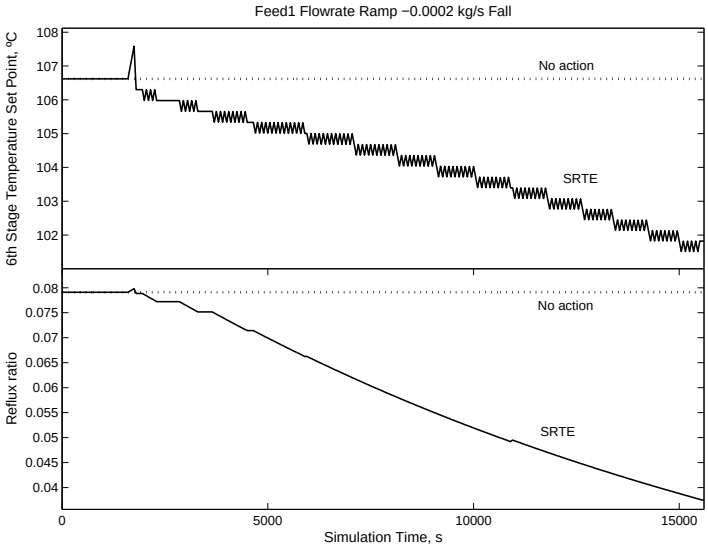


Figure 10.11: Incidence 4 - Decision variables representation.

The interaction between real time optimization and the fault diagnosis system under the presented supervised framework is here quite relevant. The AEM system showed normal conditions until the appearance of the fault. Time between the fault occurrence and its detection can be followed by the first wave at the instant objective function curve (Figure 10.13).

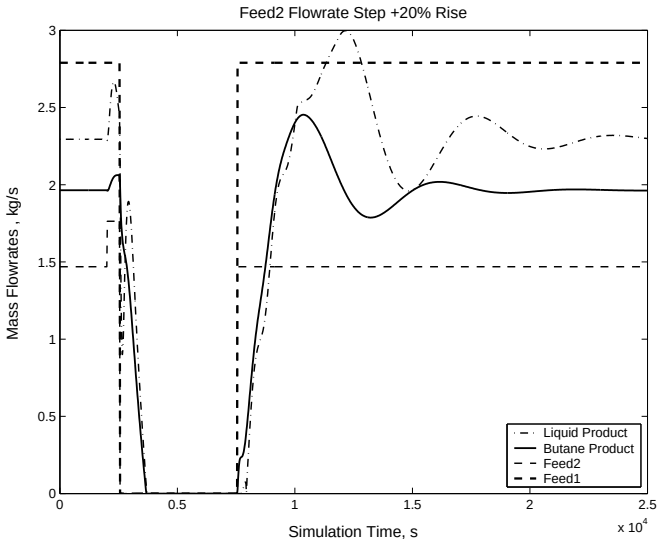


Figure 10.12: Incidence 5 - Evolution of the flow rate for the main streams.

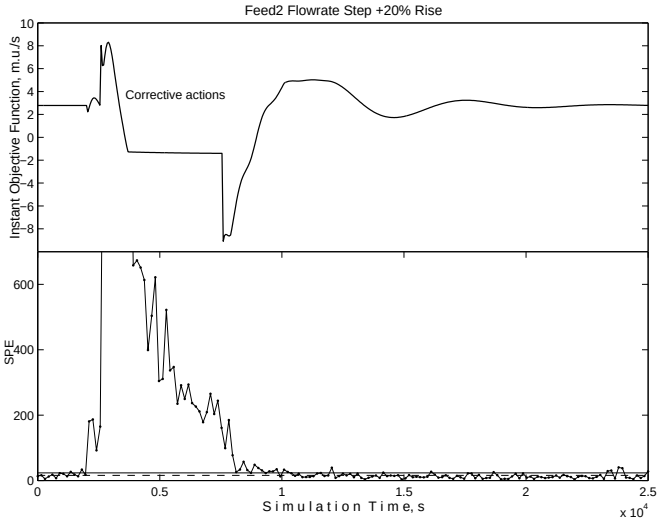


Figure 10.13: Incidence 5 - IOF and SPE representation.

Looking at the first four consecutive samplings above the detection threshold of the SPE plot, it can be seen how the fault was detected. Then, the AEM system relates this incidence with its source causes proposing a CAs protocol. In this case, the protocol was immediately applied by the process operator. It involved the shut down of the column inlet flow and the preparation to operate under total reflux. Total reflux led the plant to stationary conditions fairly far from the normal plant running conditions.

After the repairing time (5000 seconds approximately), feeds are connected back to the column and normal operating conditions are progressively restored. It can be stated by

looking at SPE (bottom graph in Figure 10.13), which returns below its statistical thresholds after the plant comes back to the normal operating regime. After the incidence occurrence, a first profit increase (instant objective function curve on Figure 10.13) is observed due to the plant feed shutdown. This initial profit increase is caused by the gradual increase on the reflux rate. This is given by simultaneity of two different situations affecting the profit function: 1) the gradual production decrease and 2) the sudden disappearance of raw material costs.

10.6 Conclusions

In this chapter, the AEM system proposed in this thesis incorporates a real time optimization algorithm, namely, Real Time Evolution (RTE). As a result, the on-line plant optimization and the appearance of abnormal events in plant are jointly managed by a common decision support system providing a global response under any plant operating condition. Moreover, the module combination makes RTE a useful improvement technique on real problems, since its actions are now supervised.

The extended AEM system, which incorporates the RTE module, demonstrated higher performance than classical RTO in terms of economical profit during process transitions. Additionally, both improved the global process response with respect to the natural process evolution under disturbances when no action is carried out. The methodology demonstrated high success when integrating AEM system with other information plant modules (real time optimization) in different operating scenarios. Regarding chapter 9, the decision-making support to process operator was here extended to keep process safety and to include process economy.

Nomenclature

SPE	Square Predicted Error
T^2	Hotteling's statistic

Acronyms

AEM	Abnormal Events Management
CAs	Corrective Actions
DSS	Decision Support System
FDS	Fault Diagnosis System
FLS	Fuzzy Logic System
PCA	Principal Component Analysis
PID	Proportional Integral Derivative (controller)
RTE	Real Time Evolution
RTO	Real Time Optimization
SRTE	Supervised Real Time Optimization
ML-SVM	MultiLabel based Support Vector Machines

CHAPTER 11

Final Conclusions

Chemical industry has evolved from being a manually and intuitively operated process industry to become a very complex an automated system involving hundreds of operations and requiring of new supervision strategies. Old human supervision procedures based on intuition, experience and the common sense of process operators is not possible anymore because of the current automatic control of chemical plants.

In contrast, new and more structured information is available from on-line measurements. New plant wide installed data acquisition systems allow a better tracking of processes, thus providing the basis upon which to attain better control policies and process competitiveness. Process industry has made a great effort on acquiring much information to turn processes into knowledgeable systems for process operators. Such effort has generally succeeded and precise measuring systems are already installed on most of chemical plants. These acquisition systems are new process operators' eyes and ears. However, this substantial amount of information requires of a deep analysis to extract practical knowledge of the plant state. The interpretation of hundreds of variables measured continuously demands sophisticated tools for decoding data into valuable knowledge upon which make decisions. Although much has been done also in this regard, current supervision systems can take more profit from this available information and go further in the safe and optimal process operability. Related literature reveals wide opportunities by a better information management.

Within this context, this thesis provides novel solutions and practical implementations to address the Abnormal Events Management (AEM) of chemical industry by taking profit of process data bases and data acquisition systems. This thesis adopts a divide-and-conquer conception and focuses on each link of the AEM chain. Firstly, the thesis outlines the general AEM view by presenting the global approach to deal with abnormal events. Then, it centers on methodologies proposed to solve each sub-problem by addressing existing limitations at each area contemplated in the AEM chain. The result is an AEM system proposal and a compendium of novel and practical techniques that over-

come some of the weaknesses reported on the components that conform the entire AEM.

Thus, chapter 2 reviews the approaches and methodologies proposed in the literature to deal with components involved on the process AEM strategy. Based on a strengths and weaknesses analysis of the related literature, a global AEM approach is proposed in chapter 3. It is worth remarking that the proposed AEM contemplates globally the complete information flow required to support process operators faced with any abnormal event, that is, from the process data acquisition to the proposal of corrective action protocols. Note that, the AEM approach incorporates a super Fault Diagnosis System (FDS) as a core module which consists of a set of independent FDSs. This configuration brings in important benefits coming from systems complementarity that makes the strategy more flexible and the final diagnosis more reliable and robust. It is also important to notice that the AEM approach not only uses on-line process information, but takes profit of the previous process off-line safety analysis to support process operators decision-making.

Chapter 4 starts presenting the first chain module: the data acquisition and ratification systems. Highlighting the importance of observability, reliability and consistency of process data from which decisions will be made, this chapter provides two novel solutions at weak points of Sensor Networks (SNs) design and on-line Data Reconciliation (DR). The chapter firstly proposes an optimized SN taking into account observability, reliability and diagnosis performance as optimization objectives. In this sense, the proposal includes the diagnosis performance as a key point on the SN design. Regarding DR, this chapter demonstrates relevant DR improvements by the adjustment of process variables delays. By optimizing the variables correlation, variables delays were corrected enhancing DR results.

Chapters 5, 6, 7 and 8 focus on the FDS, the core module of the AEM system proposal. Chapter 5 formulates and standardizes some performance indices. By this way, the main fault diagnosis aspects are reviewed and the problem is formally formulated from a classification viewpoint. MultiLabel (ML) and monoLabel (mL) approaches are introduced to address the diagnosis problem and some rigorous comparison indices are provided to check FDSs performance in all senses. These indices deal with the performance of a FDS from all points of view and allow to know how good our FDS is.

Chapter 6 presents the first FDS proposed in this thesis. The first part of chapter 6 introduces Support Vector Machines (SVM), a recently developed machine learning algorithm, and outlines its possible success in chemical process fault diagnosis. Then, the mL and ML approaches are both used and assessed to face the single and the simultaneous fault diagnosis problems. The chapter outlines limitations of the mL approach and proposes a FDSs formulation that is able to solve the simultaneous fault diagnosis problem in a simple and general way. The system was tested facing single and simultaneous faults in the challenging Tennessee Eastman process obtaining the best diagnosis results reported to date.

Chapter 7 presents another FDS developed in this thesis. It is a two-module system which separates detection and diagnosis functions. The detection is performed by a Principal Component Analysis (PCA) module whereas diagnosis is carried out by a set of rules automatically extracted by a novel algorithm which applies fuzzy logic. The system is tuned by optimizing the parameters defining the membership functions. It is carried out

by applying a genetic algorithm driven by an original objective function proposed in the chapter. Very good results were obtained, showing high complementarity with those obtained by the first FDS.

Finally, chapter 8 outlines in a practical way, one of the key points of this thesis, the FDSs integration. Complementarity of both FDSs developed in the thesis was stated theoretically as well as quantitatively throughout the chapter showing the high success of both systems combination. By means of integrating both FDSs, synergies were found and high performance enhancements were attained with respect to the systems acting separately.

Furthermore, this chapter reveals another crucial aspect of this thesis, the importance of adopting an AEM general view. Thus, from those faults not properly addressed, a clear example of attainable benefits due to AEM module's interactivity is outlined. After a deep analysis of these faults, it was easy to conclude that sometimes efforts invested on solving problems from a very constrained approach could be avoided by adopting a more general perspective. By this manner, including diagnosis criteria in the SN retrofitting would lead to measure variables involved on these difficult faults, hence, easily solving the diagnosis problem with a lower cost. Therefore, this simple example points out the benefits of the global AEM view and the necessity of establishing links between the AEM modules for an optimal global design.

Chapters 9 and 10 are devoted to the final AEM chain's link, the decision-making support. In this sense, chapter 9 integrates process safety off-line analysis, namely Process Hazards Analysis (PHA), into the AEM to enhance the interpretation of process information. Note that it is a dynamic integration that allows providing different corrective action (CA) protocols depending on the actual plant conditions. Chapter 10 goes further in the support to process operator decision-making. Thus, this chapter incorporates economical aspects to the decision system by a real time optimization module, namely, Real Time Evolution (RTE). Moreover, the combination of AEM with that tool not only extends the capabilities of the decision-making system, but makes the optimization technique more robust and applicable.

Some interesting issues were not addressed because they required an analysis beyond the scope of this thesis. However, they constitute open fields of research that are being studied. Regarding chapter 4, an attempt was done to uncover process variables delays by an MINLP optimization problem. Nevertheless, the resulting formulation has too many constraints based in matricial functions too difficult to be solved by means of non-linear equations. An algebraic optimization approach was used but the big amount on non-linear equations as well as the high combinatorial nature of the problem made infeasible to obtain a rigorous solution with current available optimization software. The main problem came from the correlation matrix (objective function) evaluation, that required of a singular value decomposition of the process data matrix. Attempts to rigorously find out process delays should focus on solving problems related to this objective function evaluation.

Regarding fault diagnosis, further efforts will be oriented to three main directions. Firstly, the global FDS should be extended to deal with the general diagnosis problem. Thus, PCA-FLS performance must be checked facing multiple faults to later test the integrated FDS performance. A second key issue, regarding fault diagnosis, is the test of the general FDS in a changing environment to check the system adaptability, that is crucial in a real

industrial context. In this sense, the optimization module described in chapter 10 can be useful to achieve new operating plant conditions from which the FDS can be tested and modified to accomplish the robustness demands. Also, it would be worth to devote more efforts at the Diagnosis Conflict Solver (DCS) in order to polish the proposed rules and extend them to more than only two FDSs.

At last, further and interesting contributions can be made on the Decision Support System (DSS). The practical and theoretical contributions given at chapter 9 related to process safety can be extended to tackle environmental issues. Off-line environmental analysis could be incorporated into the analysis, thus influencing the final corrective actions protocol. By this manner, the plant can react in a more profitable way by considering more key aspects of the global plant performance. Furthermore, additional integrations into the AEM system, beyond of the optimization module, can be attempted to enhance the general plant performance. Thus, the AEM system will be able to operate on-line with plant scheduling and higher information levels of the plant. Additional synergies can be obtained from considering these modules performing together.

APPENDIX A

Work done

This is a list of the works carried out into the scope of this thesis, in reversed chronological order.

A.1 Journal articles

A.1.1 Published

- E. Musulin, **I. Yélamos**, and L. Puigjaner. Integration of Principal Component Analysis and Fuzzy Logic Systems for Comprehensive Fault Detection and Diagnosis. *Industrial and Engineering Chemistry Research*. Ed. American Chemical Society (ACS), 45, p. 1739-1750, 2006.

- S. Ferrer-Nadal, **I. Yélamos-Ruiz**, M. Graells, and L. Puigjaner. An Integrated Framework for On-Line Supervised Optimization. *Computers and Chemical Engineering*, 31 (5-6), p. 401-409, 2007.

- **I. Yélamos**, C. Méndez, and L. Puigjaner. Enhancing Data Reconciliation Performance Through Time Delays Identification. *Chemical Engineering and Processing*, 46 (12), p. 1251-1263, 2007.

- **I. Yélamos**, G. Escudero, M. Graells, and L. Puigjaner. Simultaneous Fault Diagnosis in Chemcial Plants using a MultiLabel Approach. *AIChE Journal*, 53 (11), p. 2871-2884, 2007.

- **I. Yélamos**, S. Ferrer, M. Graells, and L. Puigjaner. Sistema de Optimización en Línea y Diagnosis de Fallos para Procesos Químicos. *Información Tecnológica*, 18 (2), p. 87-92, 2007.

A.1.2 Under evaluation

- **I. Yélamos**, G. Escudero, M. Graells, and L. Puigjaner. Performance Assessment of a Novel Fault Diagnosis System based on Support Vector Machines. *Computers and Chemical Engineering*, 2006.

- **I. Yélamos**, A. Bojarski, G. Joglekar, V. Venkatasubramanian, and L. Puigjaner. Enhancing Abnormal Events Management by the use of Quantitative Process Hazards Analysis Results. *Industrial and Engineering Chemistry Research*, 2007.

A.2 Articles in conference proceedings

- S. Ferrer, **I. Yélamos**, M. Graells, and L. Puigjaner. Application of Real Time Evolution to continuous Multi-Component distillation columns. In *2004 AIChE Annual Meeting Conference Proceedings*. American Institute of Chemical Engineers, 2004, p. 265-269.

- S. Ferrer, **I. Yélamos**, M. Graells, and L. Puigjaner. On-line Fault Diagnosis Support for Real Time Evolution applied to Multi-Component Distillation. In *15th European Symposium on Computer Aided Process Engineering, ESCAPE 15*, Barcelona, Spain. Elsevier, 2005, p. 961-966.

- **I. Yélamos**, R.V. Tona, M. Graells, and L. Puigjaner. Benefits of Data Filtering on Fault Diagnosis System Resolution. In *10th Congreso Mediterraneo de Ingeniería Química*. Sociedad Española de Química Industrial en Ingeniería Química (SEQUI), 2005, p. 567.

- **I. Yélamos**, G. Escudero, M. Graells, and L. Puigjaner. Application of Support Vector Machines to Enhance Reliable and Accurate Fault Diagnosis in Chemical Processes. In *10th Congreso Mediterraneo de Ingeniería Química*. Sociedad Española de Química Industrial en Ingeniería Química (SEQUI), 2005, p. 568.

- **I. Yélamos**, E. Musulin, and L. Puigjaner. An Hybrid System for Robust and Transparent Process Fault Diagnosis. In *2005 AIChE Annual Meeting Conference Proceedings*. American Institute of Chemical Engineers. Omnipress, 2005, p. 356-360.

- **I. Yélamos**, G. Escudero, M. Graells, and L. Puigjaner. Fault Diagnosis Based on Support Vector Machines and Systematic Comparison to Existing Approaches. In *16th European Symposium on Computer Aided Process Engineering, ESCAPE 16 and 9th International Symposium on Process System Engineering*. Garmisch Partenkirchen, Germany. Elsevier, Amsterdam (Ed. W. Marquardt and C. Pantelides), 2006, p. 1209-1214.

- **I. Yélamos**, M. Graells, and L. Puigjaner. Enhancing Fault Diagnosis by Incorporation of Intelligent Filtering Knowledge. In *2006 AIChE Annual Meeting Conference Proceedings*. American Institute of Chemical Engineers. Omnipress, 2006, p. 302.

- R. Angelini, **I. Yélamos**, and L. Puigjaner. Design of a Diagnosis-Based Sensor Network. In *2006 AIChE Annual Meeting Conference Proceedings*. American Institute of Chemical Engineers. Omnipress, 2006, p. 542.

- **I. Yélamos**, G. Escudero, M. Graells, and L. Puigjaner. Simultaneous Fault Diagnosis in Chemical Plants using Support Vector Machines. In *17th European Symposium on Computer Aided Process Engineering, ESCAPE 17*, Bucharest, Romania. Elsevier, Amsterdam (Ed. W. V. Plesu and P.S. Agachi), 2007, p. 1253-1258.

- M. Graells, **I. Yélamos**, S. Ferrer-Nadal, M. Pérez-Moya. Gestión de recursos docentes para la adaptación al espacio europeo de educación superior. In *15 Congreso Universitario de Innovación Educativa en las Enseñanzas Técnicas (XV-CUIEET)*, Valladolid, España, 18-20, julio 2007. (ISBN: 978-84-690-7547-0)

- R. Angelini, **I. Yélamos**, and L. Puigjaner. Diagnosis-Based Sensor Network, Design and Refrofitting. In *6th European Congress of Chemical Engineering (ECCE 6)*, Copenhagen, Denmark, 16- 20 September 2007, Vol. 1, p. 465-466.

- **I. Yélamos**, A. Bojarski, G. Joglekar, V. Venkatasubramanian, and L. Puigjaner. Enhancing Abnormal Events Management by the use of Quantitative Process Hazard Analsysis Results. In *6th European Congress of Chemical Engineering (ECCE 6)*, Copenhagen, Denmark, 16- 20 September 2007, Vol. 1, p. 517-518.

A.3 Participation in research projects

CHEM, *Advanced Decision Support System for Chemical/Petrochemical Manufacturing Processes-CHEM*, supported by the European Community (G1RD-CT-2001-00466), 2001-2004.

AGAPUTE, *Advanced GAs PUrification TEchnologies for co-gasification of coal, refinery by-products, biomass and waste, targeted to clean power produced from gas and steam turbine -generator sets and fuel cells*, supported by the European Community (Agapute-RFS-CR-04006), 2005-2008.

PRISM, *Towards Knowledge-Based Processing Systems-PRISM*, supported by the European Community (MRTN-CT-2004-512233), 2005-2008.

EMU *Estació de Muntatge Universal*, supported by Generalitat de Catalunya (I00898), 2006-2007.

APPENDIX B

Similarity calculation

To calculate the degree of similarity between two trapezoidal membership functions designed following the specifications of section 7.5, first note that because of symmetry,

$$\mathbf{\Psi}(A_1, A_2) = \mathbf{\Psi}(A_2, A_3) \tag{B.1}$$

to calculate $\mathbf{\Psi}(A_2, A_3)$ following Eq. (7.13), lets start evaluating $Area_1 = \| A_2 \bigcap A_3 \|$, that results (see Figure B.1),

$$Area_1 = \frac{b-a}{4} = \frac{\Delta}{4} \tag{B.2}$$

where Δ is defined as $\Delta = (b-a)$. Following the same procedure, the union can be calculated as (see Figure B.1),

$$Area_2 = \frac{b-a}{2} + 2a + \frac{3}{4}(b-a) + c = \frac{1}{4}\Delta + a + c \tag{B.3}$$

Then, replacing Eq. (B.2) and (B.3) in Eq. (7.13),

$$\mathbf{\Psi}(A_2, A_3) = \frac{\frac{\Delta}{4}}{\frac{1}{4}\Delta + a + c} = \frac{\Delta}{\Delta + 4a + 4c} \tag{B.4}$$

Finally, the similarity can be expressed as,

$$\mathbf{\Psi}(A_2, A_3) = \frac{\frac{\Delta}{c}}{\frac{\Delta}{c} + 4\frac{a}{c} + 4} \tag{B.5}$$

This function have a maximum when the *shoulder* of the function is zero ($a = 0$) and the *foot* stretches from zero to c (see Fig. B.2).

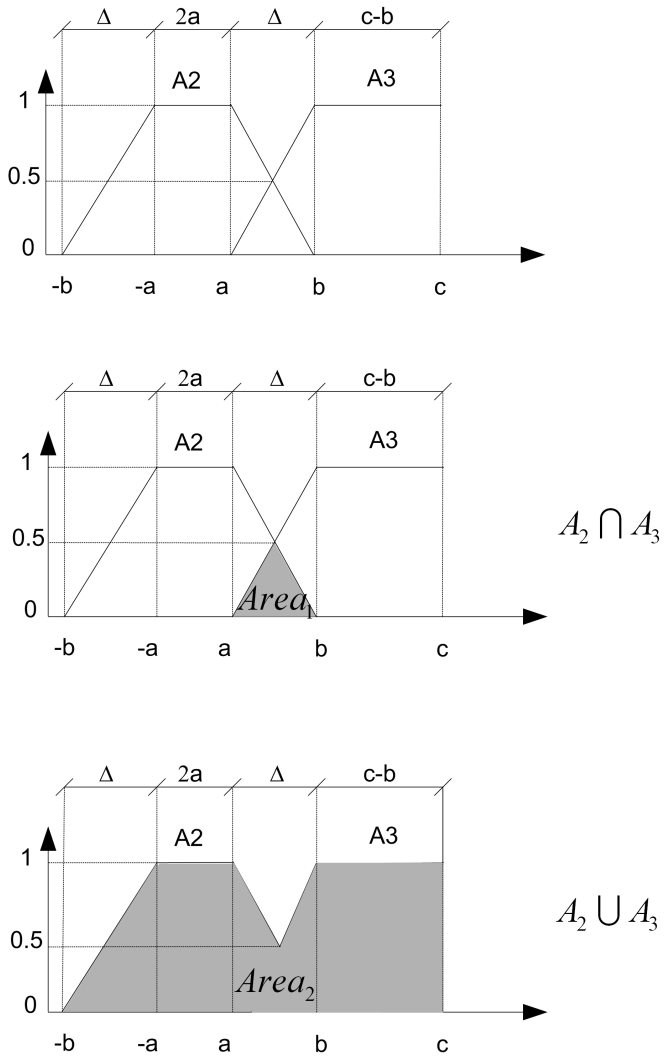


Figure B.1: Similarity calculation.

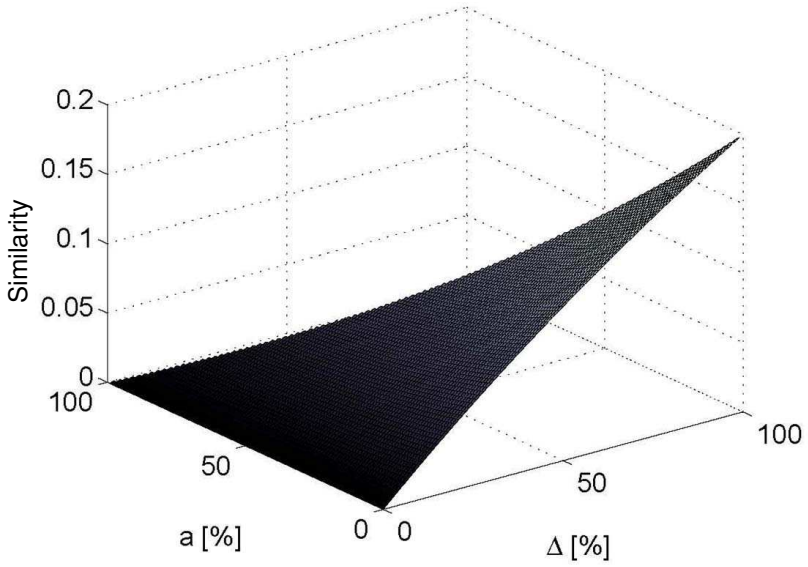


Figure B.2: Similarity surface, where Δ and a are expressed as percentage of the variable bounds c.

APPENDIX C

Original HAZOP analysis

Table C.1: Deviation 1. Original HAZOP analysis.

Causes	The cause for HIGH FLOW in refinery-units is HIGH FLOW of process-materials from upstream units into refinery units.
Consequences	The consequence for HIGH PRESSURE in SW from refinery pipe is PIPE SUBJECTED TO SURGE PRESSURE, FLANGE LEAK, POSSIBLE PIPE RUPTURE AND LOSS OF CONTAINMENT. The consequence for HIGH PRESSURE in SW from refinery pipe is release of flammable H-C-OIL (impurity) into plant area due to LEAK CAUSING FIRE HAZARD. The consequence for HIGH INLET FLOW in SURGE DRUM is HIGH INTERFACE LEVEL in surge drum. The consequence for HIGH INTERFACE LEVEL in SURGE DRUM is LOW HEAVY OUTLET PROCESS MATERIALS CONCENTRATION of the IMPURITY. H-C-OIL in surge Drum. The consequence for HIGH INTERFACE LEVEL in SURGE DRUM is HIGH LIGHTS OUTLET PROCESS MATERIALS CONCENTRATION of the RAW MATERIAL. SOUR WATER in surge drum. The consequence for HIGH INTERFACE LEVEL in SURGE DRUM is HIGH HEAVY OUTLET FLOW in surge drum. The consequence for HIGH INLET FLOW in SURGE DRUM is HIGH LEVEL in SURGE DRUM. The consequence for HIGH LEVEL in SURGE DRUM is HIGH LIGHTS OUTLET FLOW in surge drum. The consequence for HIGH PRESSURE in SURGE DRUM is RELEASE OF FLAMABLE H-C-OIL into plant area due to LEAK, CAUSING FIRE HAZARD. The consequence for HIGH LEVEL in SURGE DRUM is FILLING UP OF SURGE DRUM, possibility of LIQUID ENTERING VENT. The consequence for HIGH LEVEL in SURGE DRUM is RELEASE OF FLAMMABLE H-C-OIL into plant area due to FILLING UP and OVERFLOW of SURGE DRUM causing FIRE HAZARD.

Table C.2: Deviation 1. Key variables identification.

Key variable	Consequence
Pressure of pipe breakage	Loss of containment: Leak of sour water into plant area
Pressure of pipe leakage	Release of flammable oil: fire hazard
Drum interface level for streams out of specification	Process performance decrease
Drum interface level for high water flow	Problems in intermediate storage tank
Drum level for high slop oil flow	Alarm to slop oil: increase of flow
Drum pressure for leakage	Release of flammable oil: fire hazard
Drum level for vent entering	System performance decrease. Equipments wear
Drum level for overflow	Overflow causing fire hazard

Table C.3: Deviation 2. Original HAZOP analysis.

Causes	The cause for HIGH LEVEL in SW-OVHD-ACCUMULATOR is HIGH INLET-FLOW in SWS-OVHD-ACCUMULATOR. The cause for HIGH LEVEL in SW-OVHD-ACCUMULATOR is LOW LIQUID-OUTLET-FLOW in SWS-OVHD-ACCUMULATOR. The cause for HIGH LEVEL in SW-OVHD-ACCUMULATOR is the level sensor fails. The cause for HIGH LEVEL in SW-OVHD-ACCUMULATOR is the level controller fails.
Consequences	The consequence for HIGH LEVEL in SW-OVHD-ACCUMULATOR is filling-up of the drum, possibility of liquid entering vapor flow. The consequence for HIGH LEVEL in SW-OVHD-ACCUMULATOR is release of flammable HYDROGEN-SULFIDE, AMMONIA, into plant area due to filling-up and overflow of SW-OVHD-ACCUMULATOR causing fire hazard in the plant. The consequence for HIGH LEVEL in SW-OVHD-ACCUMULATOR is release of toxic HYDROGEN-SULFIDE, AMMONIA, into plant area due to leak, causing health hazard. The consequence for HIGH PRESSURE in SW-OVHD-ACCUMULATOR is release of flammable HYDROGEN-SULFIDE, AMMONIA, into plant area due leak, causing fire hazard in the plant. The consequence for HIGH PRESSURE in SW-OVHD-ACCUMULATOR is release of toxic HYDROGEN-SULFIDE, AMMONIA, into plant area due to leak, causing health hazard. The consequence for LOW INLET-FLOW in REFLUX-PUMP is loss of NPSH, cavitation problem, damage to the pump The consequence for HIGH INLET-PRESSURE in REFLUX-PUM is pump motor trips out on overload, process upset.

Table C.4: Deviation 1. Key variables identification.

Key variable	Consequence
Overacum level for venting into vapor outlet	Venting liquid into vapor. Equipment wear and bias from design conditions
Overacum level for overflow	Release of toxic and flammable components into plant area. Health and fire hazards.
Overacum pressure for leakage	Release of toxic and flammable components into plant area. Health and fire hazards.
Pump inlet flow	Pump cavitation problems
Pump inlet pressure	Motor trips out on overload

List of Figures

List of Tables

Bibliography

S. Abney, R.E. Schapire, and Y. Singer. Boosting applied to tagging and PP - attachment. In *SIGDAT Conference on Empirical Methods and Very Large Corpora, EMNLP/VLC*, 1999.

Z.H. Abu-el Zeet, P.D. Roberts, and V.M. Becerra. Enhancing model predictive control using dynamic data reconciliation. *AIChE J.*, 48(2):324–333, 2002.

P.S. Addison. *The Illustrated Wavelet Transform Handbook: Applications in Science, Engineering, Medicine and Finance.* Institute of Physics Publishing, Bristol, UK, 2002.

J.S. Albuquerque and L.T. Biegler. Data reconciliation and gross-error detection for dynamic systems. *AIChE J.*, 42 (10):2841, 1996.

Y. Ali and S. Narasimhan. Redundant sensor network design for linear processes. *AIChE J.*, 41:2237–2249, 1995.

Y. Ali and S. Narasimhan. Sensor network design for maximizing reliability of bilinear processes. *AIChE J.*, 42:2563–2575, 1996.

E.L. Allwein, R.E. Shapire, and Y. Singer. Reducing multiclass to binary: A unifying approach of margin classifiers. *Journal of Machine learning research*, 1:113–141, 2000.

R. Angelini, C.A. Méndez, E. Musulin, and L. Puigjaner. An optimization framework to computer-aided design and upgrade of measurement systems. In *16th ESCAPE - 9th PSE, Paper 1464, Garmisch-Partenkirchen, Germany*, pages 1293–1298, 2006a.

R. Angelini, I. Yélamos, and L. Puigjaner. Design of a diagnosis-based sensor network. In *Aiche Meeting, 2006. San Francisco, CA (USA).*, 2006b.

AspenTech. http://support.aspentech.com. Technical report, 2004.

AspenTech. Aspen dynamics version 2006 reference guide. 2006.

M. Ayoubi and R. Isermann. Neuro-fuzzy systems for diagnosis. *Fuzzy Sets and Systems*, 89:289–307, 1997.

M. Bagajewicz. Design and retrofit of sensor networks for linear processes. *AIChE J.*, 41: 2300–2306, 1997.

M. Bagajewicz. A review of techniques for instrumentation design and upgrade in process plants. *Canadian journal of chemical engineering.*, 80 (1):3–16, 2002.

M.J. Bagajewicz and Q. Jiang. Integral comparison of steady state and integral dynamic data reconciliation. *Computers Chem. Engng.*, 24:2367–2383, 2000.

M.J. Bagajewicz and M.C. Sánchez. Cost-optimal design of reliable sensor networks. *Computers Chem. Engng.*, 23:1757–1762, 2000.

B.R. Bakshi. Multiscale PCA with application to multivariate statistical process monitoring. *American Institute of Chemical Engineering Journal*, 44 (7):1596–1610, 1998.

B.R. Bakshi and G. Stephanopoulos. Temporal representation of process trends for diagnosis and control. In *IFAC Symposium on Online Fault Detection and Supervision in the Chemical Process Industry, Newark, DE, USA, 69-74*, 1992.

M. Basseville and A. Benveniste. Detection of abrupt changes in signals and dynamic systems. In *Lecture Notes in Control and Information Sciences: 77. Berlin: Springer - Verlag*, 1986.

C. Benquilou, M. Graells, E. Musulin, and L. Puigjaner. Design and retrofit of reliable sensor networks. *Ind. Eng. Chem. Res.*, 43:8026–8036, 2004.

M. Bhushan and R. Rengaswamy. Comprehensive design of a sensor network for chemical plants based on various diagnosability and reliability criteria. 1. framework. *Ind. Eng. Chem. Res.*, 41 (7):1826–1839, 2002a.

M. Bhushan and R. Rengaswamy. Comprehensive design of a sensor network for chemical plants based on various diagnosability and reliability criteria. 2. applications. *Ind. Eng. Chem. Res.*, 41 (7):1840–1860, 2002b.

B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classiffiers. In *Worshop on Computational Learning Theory, COLT*, 1992.

M. Carnero, J.L. Hernández, M. Sánchez, and A. Bandoni. On the solution of the instrumentation selection problem. *Ind. Eng. Chem. Res.*, 44:358–367, 2005.

G. Castellano, C. Catiello, A. M. Fanelli, and C. Mencar. Knowledge discovery by a neuro-fuzzy modeling framework. *2005*, 149:187–207, 2005.

M. Cerrada, J. Cardillo, J. Aguilar, and R. Faneite. Agents-based design for fault management systems in industrial processes. *Computers in Industry.*, pages accepted (Article in Press), pp. 16, 2006.

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

L.W. Chen and M. Modarres. Hierarchical decision process for fault administration. *Computers Chem. Engng.*, 16 (5):425–448, 1992.

J.T. Cheung and G. Stephanopoulos. Representation of process trends part i. a formal representation rramework. *Computers Chem. Engng.*, 14 (4-5):495–510, 1990.

Company Chevron. *WWT Two-Stage Sour Water Stripping. Improved Performance of Sulfur Recovery Units. [on-Line] Site Web, www.chevron.com/products/prodserv/refiningtechnology/documents/www.pdf*. 1998.

L.H. Chiang, M.E. Kotancheck, and A.K. Kordon. Fault diagnosis based on fisher discriminant analysis and support vector machines. *Computers Chem. Engng.*, 28:1389–1401, 2003.

L.H. Chiang, E.L. Russell, and R.D. Braatz. Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 50:243, 2000.

N. Christianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge university press, 2000.

M. Collins. *Parameter Estimation for Statistical Parsing Models: Theory and Practice of Distribution-Free Methods*. In to appear as a book chapter. Available at the web address: www.ai.mit.edu/people/mcollins, 2002. In to appear as a book chapter. Available at the web address: www.ai.mit.edu/people/mcollins.

C.M. Crowe. Data reconciliation-progress and challenges. *Journal of Process Control*, 6: 89–98, 1996.

C.M. Crowe, Y.A.G. Campos, and A. Hrymak. Reconciliation of process flow rates by matrix projection. i: Linear case. *AIChE J.*, 29:881–888, 1983.

C.R. Cutler and R.T. Perry. Real time optimization with multivariable control is required to maximize profits. *Computers Chem. Engng.*, 7:663–667, 1983.

M. Darouach and M. Zasadzinski. Data reconciliation in generalized linear dynamic systems. *AIChE J.*, 37:193, 1991.

S. Dash, R. Rengaswamy, and V. Venkatasubramanian. Fuzzy-logic based trend classification for fault diagnosis of chemical processes. *Computers Chem. Engng.*, 27:347–362, 2003.

S. Dash and V. Venkatasubramanian. Integrated framework for abnormal event management and process hazards analysis. *AIChE J.*, 49(1):124–139, 2003.

I. Daubechies. *Ten Lectures on Wavelets*. ISBN 0-89871-274-2. CBMS-NSF Regional Conference Series in Applied Mathematics, 1992.

J. de Kleer and S. Brown. A qualitative physics based on confluences. *Artificial Intelligence*, 24(1-3):7–83, 1984.

G. Delmaire, J.P. Cassar, and M. Staroswiecki. Comparison of identification and parity space approaches for failure detection in single input single output systems. In *Conference Control Applications, Glasgow*, 1994.

D. Dong and T.J. McAvoy. Nonlinear principal component analysis - based on principal curves and neural networks. *Computers Chem. Engng.*, 20 (1):65–78, 1996.

J.J. Downs and E.F. Vogel. A plant-wide industrial process control problem. *Computers Chem. Engng.*, 17(3):245–255, 1993.

R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. 1973.

S. Eizenber, M. Shacham, and N. Brauner. Combining HAZOP with dynamic simulation-applications for safety education. *Journal of Loss Prevention in the Process Industries*, 19:754–761, 2006.

G. Escudero. Machine learning techniques for word sense disambiguation. In *International Workshop an Evaluating Word Sense Disambiguation Systems, Senseval-3*, 2005.

G. Escudero, L. Màrquez, and G. Rigau. TALP system for the english lexiccal sample task. In *International Workshop an Evaluating Word Sense Disambiguation Systems, Senseval-3*, 2004.

Z. Fathi, W.F. Ramirez, and J. Korbiez. Analytical and knowledge-based redundancy for fault diagnosis in process plants. *AIChE J.*, pages 42–56, 1993.

P.M. Frank. New developments using AI in fault diagnosis. *Engng. Applic. Artif. Intell.*, 10 No 1:3–14, 1997.

P.M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 6:403–424, 1997.

P.M. Frank, M. Guay, and J.F. Forbes. Model-based fault diagnosis in technical processes. In *Transactions of the Institute of Measurement and Control*, volume 22, 2000.

Y. Friedman. What's wrong with unit closed loop optimization? *Hydrocarbon Processing*, 107, 1995.

P. Fröhllich and W. Nejdll. Resolving conflicts in distributed diagnosis. In W. Wahllster, editor, *12th European Conference on Artificial Intelligence*. John Willey and Sons, Ltd., 1996.

M. Gala and M. Bagajewicz. Rigorous methodology for the design and upgrade of sensor networks using cutsets. *Ind. Eng. Chem. Res.*, 45:6679–6686, 2006.

C.E. García and M. Morari. Optimal operation of integrated processing systems. part i: Open-loop online optimizing control. *AIChE J.*, 27:960–968, 1981.

A.E. García and P.M Frank. Deterministic nonlinear observer-based approaches to fault diagnosis: A survey. *Control Engineering Practice*, 5 (5):663–670, 1997.

A. Genovesi, J. Harmand, and J-P. Steyer. A fuzzy logic based diagnosis systems for the on-line supervision of an anaerobic digestor pilot plant. *Biochemical engineering journal*, 3:171–183, 1999.

J. Gertler. Analytical redundancy methods in fault detection and isolation. survey and synthesis. In *IFAC Fault Detection Supervision and Safety for Electrical Processes*, 1991.

J. Gertler. Diagnosing parametric faults: From parameter estimation to parity relations. In *American Control Conference, Seattle*, 1995.

J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York, 1998. Number ISBN: 0-8247-9427-3.

J. Gertler, L. Weihua, Y. Huang, and T. McAvoy. Isolation enhanced principal component analysis. *AIChE J.*, 45 (2):323–334, 1999.

A.V. Gheorghe, J. Birchmeier, D. Vamanu, I. Papazoglou, and W. Kröger. Comprehensive risk assessment for rail transportation of dangerous goods: A validated platform for decision support. *Reliability Engineering and System Safety*, 88:247–272, 2005.

J.E. Gillet. *Hazard Study and Risk Assesment in the Pharmaceutical Industry*. 1997.

D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69: 331–371, 1910.

F. Hamelin, N. Hassan, and D. Sauter. Fault detection method of uncertain system using interval model. In *ECC'01 Portugal*, 2001.

D. Harrold. How to avoid abnormal situations. *Control Engineering*, 53 (4-6):259–271, 1998.

E.J. Henley. Application of expert systems to fault diagnosis. In *AICHE Annual Meeting, San Francisco, CA.*, 1984.

J.A.R. Hernández, D.A. Suárez, and E.N. Sánchez. Fault diagnosis for a steam generator via recurrent neural networks. *Fault Detection, Supervision and Safety of Technical Processes*, pages 396–401, 2006.

K. Hestetun and M. Hovd. Detection of abnormal alumina feed rate in aluminium electrolysis cells using state and parameter estimation. *Computer Aided Chemical Engineering*, 21(2):1557–1562, 2006.

D.M. Himes, R.H. Storer, and C. Georgakis. Determination of the number of principal components for disturbance detection and isolation. In *Proceedings of the Americal Control Conference*, Piscataway, NJ, June 1994. IEEE Press.

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

T. Höfling and R. Isermann. Fault detection based on adaptive parity equations and single-parameter tracking. *Control Eng. Practice*, 4 No 10:1361–1369, 1996.

J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

D.T. Horak. Failure detection in dynamic systems with modelling errors. *J. Guidance, Control and Dynamics*, 11 (6):508–516, 1988.

J.C. Hoskins, K.M Kaliyur, and D.M. Himmelblau. Fault diganosis in complex chemical plants using artificial neural networks. 37:137–142, 1991.

H J. Hotelling. Analysis of a complex of statistical variables into principal components. *Educ. Psychol.*, 24:417–441, 498–520, 1933.

Y. Huang, G.V. Reklaitis, and V. Venkatasubramanian. Dynamic optimization based fault accommodation. *Computers Chem. Engng.*, 24:439–444, 2000.

R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing. *Automatica*, 29:815–836, 1993.

R. Isermann. Model-based fault detection and diagnosis - status and applications. *Annual Reviews in Control*, 29 (1):71–85, 2005.

H. Ishibuchi and T. Yamamoto. Comparison of heuristic criteria for fuzzy rule selection in classification problems. *Fuzzy optimization and decision making*, 3(2):119–139, 2004.

J.E. Jackson. Principal components and factor analysis: Part I principal components. *Journal of Quality Technology*, 12:201–213, 1980.

J.E. Jackson. *A user's guide to principal components*. John Wiley, New York, 1991.

S.R. Jang and C.T. Sun. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1997.

T.W. Jiang, B.Z. Chen, He X.R., and P Stuart. Application of steady-state detection method based on wavelet transform. *Computers Chem. Engng.*, 27(4):569–578, 2003.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning, ECML*, 1998.

T. Joachims. *Making Large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*. MIT-Press, 1999.

T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, 2002.

C. Karr. Fuzzy control of pH using genetic algorithms. *IEEE Trans. Fuzzy Syst.*, pages 146–153, 1993.

J.D. Kelly. Techniques for solving industrial nonlinear data reconciliation problems. *Computers Chem. Engng.*, 28:2837–2843, 2004.

T.A. Kletz. *In: HAZOP and HAZAN Notes on the Identification and Assesment of Hazards*. 1986.

T. Kohonen. *Self-organization and associative memory*. Springer, New York, 1984.

T. Kourti and J.F. MacGregor. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28: 3–21, 1995.

T. Kourti and J.F. MacGregor. Multivariate statistical process control methods for monitoring and diagnosing process and product performance. *J. Qual. Tech.*, 28:409–428, 1996.

J.V. Kresta, J.F. MacGregor, and T.E. Marlin. Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering*, 69:35–47, 1991.

A. Kretsovalis and R.S.H. Mah. Effect of redundancy on estimation accuracy in process data reconciliation. *Chem. Eng. Sci*, 42:2115–2121, 1987.

W. Ku, R.H. Storer, and C. Georgakis. Distubance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30: 179–196, 1995.

T. Kudo and Y. Matsumoto. Chunking with support vector machines. In *Proceedings of the Meeting of the North American Chapter of the Association for Computational Linguistics, NAACL*, 2001.

D.R. Kuehn and H. Davidson. Computer control. II. mathematics of control. *Chem. Eng. Progress*, 57:44–47, 1961.

A. Kulkarni, V.K. Jayaraman, and B.D. Kulkarni. Knowledge incorporated support vector machines to detect faults in tennessee eastman process. *Computers Chem. Engng.*, 29: 2128–2133, 2005.

M. Laser. Recent safety and environmental legislation. 2002.

H.G. Lawley. Operability studies and hazard analysis. *Chemical Engineering Progress*, 70:105–116, 1974.

J.H. Lee and N.L. Ricker. Extended kalman filter based nonlinear model predictive control. *Ind. Eng. Chem. Res.*, 33:1530–1541, 1994.

S.Y. Lee, J.M. Lee, D. Lee, and I. Lee. Improvement in steam stripping of sour water through and industrial-scale simulation. *Korean Journal of Chemical Engineering*, 21 (3):549–555, 2004.

D. Leung and J. Romagnoli. An integration mechanism for multivariate knowledge-based fault diagnosis. *Journal of Process Control*, 12:15–26, 2002.

F.I. Lewis. *Optimal Estimation with an Introduction to Stochastic Control Theory*. John Wiley and Sons. Inc., 1986.

L. Leyval, J. Montmain, and S. Gentil. Qualitative analysis for decision-making in supervision of industrial continuous processes. *Mathematics and Computers in Simulation.*, 36(2):149–163, 1994.

M.J. Liebman, T.F. Edgar, and L.S. Ladson. Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers Chem. Engng.*, 16:963, 1992.

W. Lin, Y. Qian, and X. Li. Nonlinear dynamic principal component analysis for on-line process monitoring and diagnosis. *Computers Chem. Engng.*, 24:423–429, 2000.

W. Liu. An extended kalman filter and neural network cascade fault diagnosis strategy for the glutamic acid ferementation process. *Artificial Intelligence in Engineering*, 13 (2):131–140, 1999.

L.L. Ljung. *System Identification: Theory for the User*. 1999.

W.L. Luyben. *Process Modeling, Simulation and Control for Chemical Engineers.* New York: McGraw Hill, 1990.

J.F. MacGregor, C. Jaeckle, C. Kiparissides, and M. Koutoudi. Process monitoring and diagnosis by multi-block pls methods. *AIChE J.*, 40(5):826–838, 1994.

F. Madron and V. Veverka. Optimal selection of measuring points in complex plants by linear models. *AIChE J.*, 38:227–236, 1992.

R.S.H. Mah, G.M. Stanley, and D.M. Downing. Reconciliation and rectification of process flow and inventory data. *Ind. and Eng. Chemistry, Proc. Des. Dev.*, 15:175–183, 1976.

S. Mallat. *A Wavelet Tour of Signal Processing*. ISBN: 0-12-466606-X. Academic Press, 1999.

E.H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Machine Studies*, 7(1):1–13, 1975.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

T. Marlin and A.N. Hrymak. Real-time operations optimization of continuous processes. In *AIChE Symposium Series Fifth International Conference on Chemical Process Control*, volume 93, pages 156–164, 1996.

M.R. Maurya, R. Rengaswamy, and V. Venkatasubramanian. Application of signed digraphs based analysis for fault diagnosis of chemical flowsheets. *Engng. Applications of Artificial Intelligence*, 17:501–518, 2004.

T.J. McAvoy and N. Ye. Base control for the tennessee eastman problem. *Computers Chem. Engng.*, 18:383, 1994.

R.C. McFarlane and D.W. Bacon. Adaptative optimizing control of multivariable constrained chemical processes. *Ind. Eng. Chem. Res.*, 28:1828–1834, 1989.

I.P. Miletic and T.E. Marlin. On-line statistical results analysis in real-time operations optimization. *Ind. Eng. Chem. Res.*, 37:3670–3684, 2004.

K. Mingfang, C. Bingzhen, and L. Bo. An integral approach to dynamic data rectification. *Computers Chem. Engng.*, 20:749, 2000.

M. Misiti, Y. Misiti, G. Oppenheim, and J.M. Poggi. *Wavelet Toolbox. User's Guide. 3rd Edition.* Natick, MA (USA), 2004.

L.F.L. Moro. Process technology in the petroleum refining industry - current situation and future trends. *Computers Chem. Engng.*, 27 (8-9):1303–1305, 2003.

M. Murata, M. Utiyama, K. Uchimoto, Q. Ma, and H. Isahara. Japanese work sense disambiguation using the simple bayes and support vector machine methods. In *Proceedings of the International Workshop an Evaluating Word Sense Disambiguation Systems, Senseval-2*, 2001.

E. Musulin, M. Bagajewicz, J.M. Nougués, and L. Puigjaner. Design and upgrade of sensor networks for principal component analysis monitoring. *Ind. Eng. Chem. Res.*, 43:2150–2159, 2004.

E. Musulin, C. Benqlilou, M.J. Bagajewicz, and L. Puigjaner. Instrumentation design and upgrade for optimal kalman filtering. *Journal of Process Control*, 15(6):629–638, 2005.

E. Musulin, I. Yélamos, and L. Puigjaner. Integration of principal component analysis and fuzzy logic systems for comprhensive process fault detection and diagnosis. *Ind. Eng. Chem. Res.*, 45:1739–1750, 2006.

D. Mylaraswamy and V. Venkatasubramanian. A hybrid framework for large scale process fault diagnosis. *Computers Chem. Engng.*, 21:935–940, 1997.

S. Narasimhan and C. Jordache. *Data Reconciliation and Gross Error Detection, an Intelligent Use of Process Data.* Gulf Publishing Company, Houston, TX (USA), 2000.

D. Neogi and C.E. Schlags. Multivariate statistical analysis of an emulsion batch process. *Ind. Eng. Chem. Res.*, 37:3971–3979, 1998.

P. Nomikos and J.F. MacGregor. Monitoring batch processes using multiway principal component analysis. *American Institute of Chemical Engineering Journal*, 40 (8): 1361–1375, 1994.

A. Norvilas, A. Negiz, J. DeCicco, and A. Çinar. Intelligent process monitoring by interfacing knowledge-based systems and multivariate statistical monitoring. *Journal of Process Control*, 10:341–350, 2000.

M.N. Nounou and B.R. Bakshi. On-line multiscale filtering of random and gross errors without process models. *AIChE J.*, 45 No. 5:1041–1058, 1999.

University of Sheffield. *Dept. Automatic Control and Systems engineering. Genetic Algorithm Toolbox. [on-Line] Site Web, URL<http://Www.Shef.Ac.Uk/Uni/Projects/Gaipp/Ga-Toolbox/>.* 2003.

P.R. Paiva and A. Dourado. Interpretability and learning in neuro-fuzzy systems. *Fuzzy sets and systems*, 147:17–38, 2004.

R.J. Patton and J. Chen. A review of parity space approaches to fault diagnosis. *Proc. IFAC SAFEPROCESS Symp.*, pages 65–81, 1991.

R.J. Patton, P.M. Frank, and R.N. Clark. *Fault Diagnosis in Dynamic Systems, Theory and Application.* Prentice Hall, (Eds), 1989.

K. Pearson. On lines and planes of closest fit to systems of point in space. *Phil. Mag.*, 6 (2):559–572, 1901.

J.C. Pinto. On the costs of parameters uncertainties: Effect of parameter uncertainties during optimization and design of experiments. *Chemical Engineering Science*, 53: 2029–2040, 1998.

V. Puig and J. Quevedo. Passive robust fault detection using fuzzy parity equations. *Mathematics and Computers in Simulation*, 60 (3,5):193–207, 2002.

V. Puig, J. Quevedo, T. Escobet, and S. de Las Heras. Robust fault detection approaches using interval models. In *IFAC W. C. Spain*, 2002.

R. Raghuraj, M. Bhushan, and R. Rengaswamy. Location of sensors in complex chemical plants based on fault diagnostic observability criteria. *1999*, 45 (2):310–322, 1999.

J. Ragot, D. Maquin, and G. Bloch. Sensor positioning for processes described by bilinear equations. *Diagnostic et surete de fonctionment.*, 2:115–132, 1992.

A. Raich and A. Çinar. Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *AIChE J.*, 42:995–1009, 1996.

A. Raich and A. Çinar. Diagnosis of process disturbances by statistical distance and angle measures. *Computers Chem. Engng.*, 21(6):661–673, 1997.

J. Rasmussen. *Information processing and human-machine interaction*. North Holland, New York, 1986.

R. Rengaswamy. *A Theory for Integrating Process Monitoring, Fault Diagnosis and Supervisory Control*. PhD thesis, Purdue University, 1995.

N.L. Ricker and L.H. Lee. Nonlinear modeling and state estimation for the tennessee eastman challenge process. *Computers Chem. Engng.*, 19 (9):983–1005, 1995a.

N.L. Ricker and L.H. Lee. Nonlinear predictive control of the tennessee eastman challenge process. *Computers Chem. Engng.*, 19 (9):961–981, 1995b.

J.A. Romagnoli and G. Stephanopoulos. Rectification of process measurement data in the presence of gross erros. *Chem. Eng. Science*, 36:1849–1863, 1981.

D. Ruiz. *Fault Diagnosis in Chemical Plants Integrated to the Information System*. PhD thesis, Chemical Engineering Department, Polythecnic University of Catalonia, 2001.

M. Sánchez and J. Romagnoli. Use of orthogonal transformation in data classification-reconciliation. *Computers Chem. Engng.*, 20:483–493, 1996.

Y. Schapire and R.E. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39, 2/3:135–168, 2000.

S. Sequeira, M. Graells, and L. Puigjaner. Real-time evolution for on-line optimization of continuous processes. *Ind. Eng. Chem. Res.*, 41:1815–1825, 2002.

E.G. Shopova and N.G. Vaklieva-Bancheva. A genetic algorithm for engineering problems solutions. *Computers Chem. Engng.*, 30 (8):1293–1309, 2006.

T. Sorsa and H.N. Koivo. Application of artificial neural networks in process fault diagnosis. In *IFAC/IMACS - Symposium on Fault Detection Supervision and Safety for Technical Processes Safeprocess'91. Baden-Baden*, 1991.

O.A.Z. Sotomayor and D. Odloak. Observer-based fault diagnosis in chemical plants. *Chemical Engineering Journal*, 112:93–108, 2005.

M. Staroswiecki, J.P. Cassar, and V. Cocquempot. Generation of optimal structured residuals in the parity space. *Proc. IFAC 12th World Congress*, 5:535–542, 1993.

G. Stephanopoulos and C. Han. Intelligent systems in process engineering: A review. *Computers Chem. Engng.*, 20 (6-7):743–791, 1996.

S. Tanaka. Diagnosability of systems and optimal sensor location. NY: Prentice Hall, 1989.

E.E. Tarifa and N.J. Scenna. A methodology for fault diagosis in large chemical processes and an application to a multistage flash desaliantion process: Part i. *Reliability Engineering and System Safety*, 60 (1):29–40, 1998.

R. Thiele, O. Brettschneider, J.H. Repke, H. Thielert, and G. Wozny. Experimental investigations of foaming in a packed tower for sour water stripping. *Ind. Eng. Chem. Res.*, 42:1426–1432, 2003.

I.B. Tjao and L.T. Biegler. Simultaneous strategies for data reconciliation and gross error detection of nonlinear systems. *Computers Chem. Engng.*, 15, 1:679–690, 1991.

R.V. Tona, C. Benqlilou, A. Espuña, and L. Puigjaner. Dynamic data reconciliation based on wavelet trend analysis. *Ind. Eng. Chem. Res.*, 44:4323, 2005.

P. Vachhani, R. Rengaswamy, and V. Venkatasubramanian. A framework for integrating diagnostic knowledge with nonlinear optimization for data reconciliation and parameter estimation in dynamic systems. *Chemical Engineering Science*, 56 (6):2133–2148, 2001.

V. Vaclavek. Studies on system engineering-III. optimal choice of the balance measurements in complicated chemical engineering systems. *Chemical Engineering Science*, 24:947–955, 1969.

V. Vaclavek and M. Loucka. Selection of measurements necessary to achieve multicomponent mass balances in chemical plants. *Chemical Engineering Science*, 31:1199–1205, 1976.

R. Vaidhyanathan. *A Model-Based Framework for Automating HAZOP Analysis of Continuous Process Plants*. PhD thesis, Purdue University, Indiana (USA), 1995.

R. Vaidhyanathan and V. Venkatasubramanian. Digraph-based models for automated HAZOP analysis. *Journal of Reliability Engineering and System Safety*, 50 (1):33–49, 1995.

R. Vaidhyanathan and V. Venkatasubramanian. A semi-quantitative reasoning methodology for filtering and ranking HAZOP results in HAZOPExpert. *Reliability Engineering and System Safety*, 53:185–203, 1996.

V.N. Vapnik. *Statistical Learning Theory*. John Wiley, 1998.

H. Vedam and V. Venkatasubramanian. PCA-SDG based process monitoring and fault diagnosis. *Control Engineering Practice*, pages 903–917, 1999.

V. Venkatasubramanian. Inexact reasoning in expert systems: a stochastic parallel network approach. In *Proceedings of the second conference on artificial intelligence applications*, pages 191–195, Miami, FL., USA, 1985.

V. Venkatasubramanian. Abnormal events managements in complex process plants: Challenges and opportunities in intelligent supervisory control. In *4th International Conference on Foundation of Computer-Aided Process Operations, FOCAPO 2003, Coral Springs-FL. USA.*, 2003.

V. Venkatasubramanian, R. Rengaswamy, S.N. Kavury, and K. Yin. A review of process fault detection and diagnosis part III: Process history based methods. *Computers Chem. Engng.*, 27:327–346, 2003a.

V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis. Part I: Quantitative model-based methods. *Computers Chem. Engng.*, 27:293–311, 2003b.

V. Venkatasubramanian, R. Vaidyanathan, and Y. Yamamoto. Process fault detection and diagnosis using neural networks i: steady state processes. *Computers Chem. Engng.*, 14(7):699–712, 1990.

A. Wachs and D.R. Lewin. Improved PCA methods for process disturbance and failure identification. *AIChE J.*, 45(8):1688–1700, 1999.

D. Wang and J.A Romagnoli. Robust multi-scale principal component analysis with applications to process monitoring. *Journal of Process Control*, 15:869–882, 2005.

H. Wang, S. Kwong, Y. Jin, W. Wei, and K.F. Man. Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction. *Fuzzy sets and systems*, 149:149–186, 2005.

G. Weidl, A.L. Madsen, and Israelson. Applications of object-oriented bayesian networks for condition monitoring, root cause analysis and decision support operation of complex continuous processes. *Computers Chem. Engng.*, 29:1999–2009, 2005.

D.C. White. On line optimization: What have we learned? *Hydrocarbon Processing.*, 77 (6):55, 1998.

A.S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.

B. M. Wise, L. Ricker, and D. J. Veltkamp. Upset and sensor failure detection in multivariate process. Technical report, Eigenvector Research, Manson, WA, 1989.

S. Wold, N. Kettaneh, and K. Thessem. Hierarchical multiblock PLS and PC models for easier model interpretation and as an alternative to variable selection. *Journal of Chemometric*, 10:436–473, 1996.

H. Wörn, T. Längle, and M. Albert. Multi-agent architecture for monitoring and diagnosing complex systems. In *The 4th International Workshop on Computer Science and Information Technologies (CSIT) University of Patras/Greece*, 2002.

Q. Xiong and A. Jutan. Continuous optimization using dynamic simplex method. *Chemical Engineering Science*, 58:3817–3828, 2003.

D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Annual Meeting of the Association for Computational Linguistics, ACL*, 1995.

I. Yélamos, C. Méndez, and L. Puigjaner. Enhancing data reconciliation by time delays identification. *Chemical Engineering and Processing*, 46 (12):1251–1263, 2007.

W.S. Yip and T.E. Marlin. The effect of model fidelity on real-time optimization performance. *Computers Chem. Engng.*, 28:267–280, 2004.

S. Yoon and J.F. MacGregor. Statistical and causal model-based approaches to fault detection and isolation. *AIChE J.*, 46 (9):1813–1824, 2000.