Universitat de Girona

CASE-BASED DIAGNOSIS OF BATCH PROCESSES BASED ON LATENT STRUCTURES

Xavier BERJAGA MOLINÉ

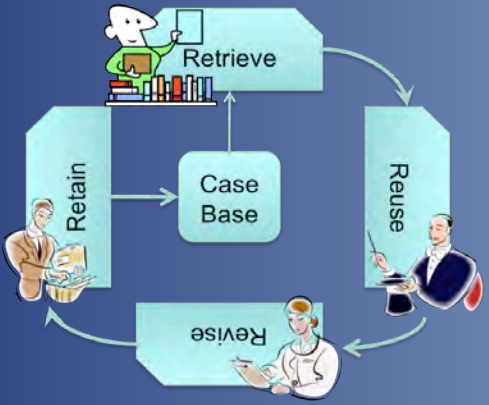
Dipòsit legal: Gi. 1713-2013
http://hdl.handle.net/10803/126303

Universitat de Girona

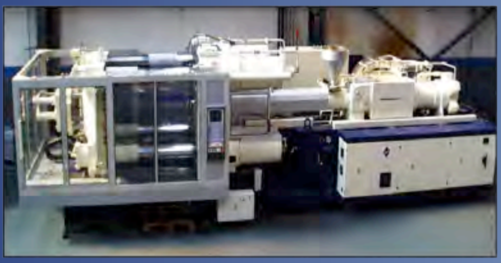
PhD Thesis

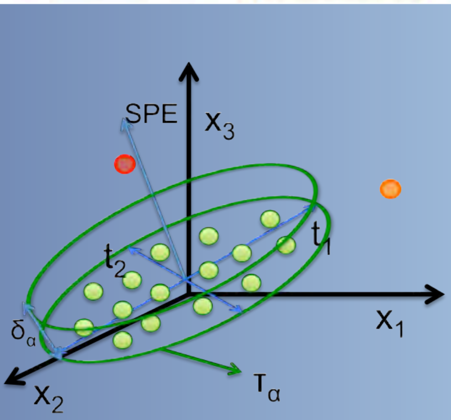
Case-based diagnosis of batch processes based on latent structures

Xavier Berjaga Moliné



ORP
pH-T
DO
EFFLUENT
INFLUENT
AIR





2013

Universitat de Girona

DOCTORAL THESIS

Case-based diagnosis of batch processes based on latent structures

Xavier Berjaga Moliné

2013

Doctoral Programme in Technology

Advisor:
Dr. Joaquim Meléndez i Frigola

Thesis submitted in partial fulfilment of the requirements for the Degree of Doctor in Philosophy at the University of Girona

Case-based diagnosis of batch processes based on latent structures

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

Date of Signature

Xavier Berjaga
Author

Joaquim Meléndez
Thesis Advisor

List of Publications

This is the list of publications in journals and conferences derived from the main contribution of this thesis:

- **Journals**

  1. M. Ruiz, G. Sin, **X. Berjaga**, J. Colprim, S. Puig and J. Colomer, *Multivariate Principal Component Analysis and Case-Based Reasoning for monitoring, fault detection and diagnosis in a WWTP*, Water Science and Technology, vol. 64(8), pp. 1661-1667, 2011, (Ruiz et al., 2011).
  2. **X. Berjaga**, M. Coma, J. Meléndez, S. Puig, J. Colprim and J. Colomer, *Granularity determination of activated sludge through on-line profilles by means of case- based reasoning*, Water Science and Technology (2nd revision), (Berjaga et al., 2013b).

- **Conferences**

  1. **X. Berjaga**, J. Meléndez and A. Pallarés, *Statistical Monitoring of injection moulds*, Congrés Català d'Intelligència Artificial (CCIA), St. Martí d'Empúries-Spain, October 22nd-24th 2008, (Berjaga et al., 2008a).
  2. **X. Berjaga**, A. Pallarés and J. Meléndez, *Case-based diagnosis in the principal component space. Application to injection moulds*, Workshop on Principles of Diagnosis (DX), Stockholm-Sweden, June 14th-17th 2009, (Berjaga et al., 2009c).
  3. **X. Berjaga**, A. Pallarés and J. Meléndez, *A framework for case-based diagnosis of batch processes in the principal component space*, IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Palma de Mallorca-Spain, September 22nd-26th 2009, (Berjaga et al., 2009a).
  4. **X. Berjaga**, A. Pallarés and J. Meléndez, *Quality determination of industrial batch processes in the principal component space*, Seminario de Aplicaciones Industriales de Control Avanzado (SAICA), Madrid-Spain, November 16th-17th 2009, (Berjaga et al., 2009b).
  5. **X. Berjaga**, M. Coma, J. Meléndez, S. Puig, J. Colprim and J. Colomer, *Granularity determination of activated sludge trhough on-line profiles by means of case-based reasoning*, IWA Conference on Instrumentation

Control and Automation (ICA), Narbonne-France, September 17th-20th 2013, (Berjaga et al., 2013a).

And this is the list of publications derived from secondary contributions of this thesis:

- **Journals**

  1. M. Ruiz, L. E. Mujica, **X. Berjaga** and J. Rodellar, *Partial least squares / projection to latent structure (PLS) regression to estimate impact localization in structures*, Smart Material and Structures, vol. 22(2), 02502, 2013, (Ruiz et al., 2013).

- **Conferences**

  1. L. E. Mujica, M. Ruiz, **X. Berjaga** and J. Rodellar, *Multiway partial least squares (MPLS) to estimate input impact localization in structures*, IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS), Barcelona-Spain, June 30th-July 3rd 2009, (Mujica et al., 2009).
  2. **X. Berjaga**, J. Meléndez and C. Barta, *Statistical fault detection and reconstruction of sensors of the Ariane engine*, Mediterranean Conference on Control and Automation (MED), Marrakech-Morrocco, June 23rd-25th 2010,(Berjaga et al., 2010).

This thesis is the result of a continuous research line, started in a Master Thesis at the University of Girona (Berjaga, 2008), which studied the applicability of the combination of multiway principal component analysis to batch and to finite-duration processes, which resulted in the following publications:

1. J. Meléndez, **X. Berjaga**, S. Herraiz, J. Sánchez and M. Castro, *Classification of voltage sags based on k-NN in the principal component space*, International Conference in Renewal Energies and Power Quality (ICREPQ), Santander-Spain, March 12nd-14th 2008, (Meléndez et al., 2008b).
2. J. Meléndez, **X. Berjaga**, S. Herraiz, V. Barrera, J. Sánchez and M. Castro, *Classification of voltage sags according to their origin based on the waveform similarity*, IEEE PES Transmission and Delivery Latin America, Transmission and Distribution Conference and Exposition, Bogotá-Colombia, August 13th-15th 2008, (Meléndez et al., 2008a).
3. V. Barrera, **X. Berjaga**, J. Meléndez and S. Herraiz, *Two new methods for voltage sag source location*, Conference on Harmonics and Quality Control

of Power (ICHQP), Camberra-Australia, September 28th-October 1st 2008, (Barrera et al., 2008).

A lateral publication from the thesis:

1. **X. Berjaga**, J. Meléndez and A. Pallarés, *A case-based centred approach for rapid manufacturing: definitions*, Hybrid Intelligent Systems (HIS), Barcelona-Spain, September 10th-12th 2008, (Berjaga et al., 2008b).

Notation

γ	Number of variables
ι	Number of batches
κ	Number of samples
k-NN	k Nearest Neighbours
ACC	ACCuracy
AI	Artificial Intelligence
AOC	Abnormal Operating Conditions
AS	Auto Scaling
BS	Block Scaling
BWt	Batch-Wise unfolding grouped by Time samples
BWv	Batch-wise unfolding grouped by Variables
CBD	Case-Based Diagnosis
CBR	Case-Based Reasoning
CS	Continuous Scaling
DFW	Distance and class Frequency Weighted voting
DROP4	Decremental Reduction Optimisation Procedure 4
DW	Distance-Weighted voting
eXiT	Enginyeria de Control de Sistemes Intel.ligents
FF	Floccular batch correctly identified as Floccular
FG	Floccular batch misclassified as Granular
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FR	Floccular Rate
GF	Granular batch misclassified as Floccular
GG	Granular batch correctly identified as Granular
GR	Granular Rate
GS	Group Scaling
IB2	Instance-Based learner 2
IB3	Instance-Based learner 3
IBUdG	Instance-Based learner UdG
IMM	Injection Moulding Machine(s)
MPCA	Multiway Principal Component Analysis
MPCA	Multiway Principal Component Analysis
MPLS	Multiway Partial Least Squares
MPLS-DA	Multiway Partial Least Squares with Discriminant Analysis
NF	Number of Floccular batches available
NG	Number of Granular batches available
NOC	Normal Operating Conditions
PCS	Principal Component Subspace
PRE	PREcision
ROC	Receiver Operating Curve
RS	Residual Subspace
SBR	Sequencing Batch Reactor

NOTATION

SBR	Sequencing Batch Reactor
SEN	SENsitivity
SPC	Statistical Process Control
SPE	Square Prediction Error
SPF	SPeciFicity
SV	Simple Voting
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UdG	Universitat de Girona
WWTP	WasteWater Treatment Plants(s)

List of Figures

List of Tables

To my parents.
And to you Aniki, wherever you are.

Acknowledgements

First of all, I would thank Dr. Joaquim Meléndez, my advisor in this Ph. D. Thesis for his ideas and support during this “long” trip that was this thesis.

I would also like to thank the people from Plastiasite S. A., firstly for hiring me during part of the thesis, which enriched me in how research is applied in a daily basis, and secondly for their support and ideas within the different projects I was involved with, especially to Dr. Alvaro Pallarés and Dr. Raquel Ventura, as mentors in both the European project’s “universe” and the doctoral “world”. I also greet the people from ASCAMM Foundation for their ideas with injection moulding machines, with special care to Alfonso Sánchez, whose collaboration was invaluable during all projects I was involved in.

I would also want to greet the financial support from the *Comissionat per a Universitats i Recerca del Departa- ment d’Innovacio, Universitats i Empresa* of the *Generalitat de Catalunya* for granting me the FIE fellowship this thesis was developed under, the European Commission for all the research projects I have been involved (COLL-CT-2006-030339, IP 026599-2) and the *Ministerio de Ciencia e Innovación* from Spain for supporting the research projects CTQ2008-06865-C02-02, CTQ2008-06865-C02-01/PPQ, CONSOLIDER-CSD2007-00055, GRASTAC-CTQ2008-C02-01/PPQ, BEST-ENERGY-CTQ2011-23632, SIRENA-CTQ2011-24114/PPQ and N- OPTIMMOX-IPT-2011-1073-310000.

To all friends in P-IV, and especially, the eXiT research group for all those good times and memories, and among them, I would especially greet the help provided by Llorenç Burgas for helping in the revision and development of the methodology detailed within this thesis.

To Dr. Magda Ruiz from the *Universtiat Politècnica de Catalunya* and Dr. Alberto Ferrer from the *Universidad Politécnica de Valencia* for their insights in the MPCA and CBR methods and their collaborations and courses respectively. They were an invaluable source of inspiration.

I thank the LEQUIA research group for providing the data and knowledge within the wastewater treatment plants, especially Marta Coma, for being so patient with my enquiries.

I also thank TNO and ASCAMM Foundation for providing the data related with the injection moulding machine field.

Contents

Resum

L'objectiu d'aquesta tesi és la de presentar un mètode automàtic per al monitoratge dels processos per lots basat en la combinació de models estadístics i mètodes d'aprenentatge automàtic. El primer s'utilitza per modelar el procés mitjanant les relacions ms significatives entre les variables mesurades al llarg del temps, mentre que el segon s'utilitza per millorar la capacitat de diagnosi del sistema. Els mètodes estadístics no relacionen una observació amb falla amb l'origen d'aquesta (només llisten el subconjunt de variables què han vist alterat el seu comportament) al mateix temps que no tenen capacitat d'aprenentatge. El fet d'utilitzar raonament basat en casos per a la diagnosi permet relacionar les observacions amb falla amb informació més significativa (com seria la causa de la falla). Els models estadístics també proporcionen una nova representació de les observacions, en una base ortogonal, que facilita l'aplicabilitat dels mètodes basats en distàncies del raonament basat en casos, tot millorant-ne els resultats obtinguts.

La informació proveïda de projectar sobre el model estadístic s'ha utilitzat per definir veïnatges significatius per al monitoratge de processos per lots. Tanmateix, s'han definit nous mètodes per minimitzar l'efecte de les instàncies "sorolloses" a l'hora de reutilitzar la informaci recuperada de casos anteriors, així com procediments per minimitzar la base de casos tot mantenint la seva eficàcia. Amb aquest objectiu en ment, tota la informació relativa a un lot (configuració del procés, variables observades, etapes del procés, entre altres), mesures al llarg del temps i la informació relacionada amb el seu diagnosi (origen de la falla, direcció de falla, accions correctives, etc.) s'han fet servir per caracteritzar el lot com a un cas, els quals s'agrupen posteriorment en la base de casos per futures diagnosis.

Per tal de provar la metodologia, s'han desenvolupat dues aplicacions diferents (depuració d'aigües i màquines d'injecció en motlles) basades en la metodologia proposada. La metodologia s’ha aplicat tant “off-line” com dins de la cadena de producció (un cop finalitzat el lot), obtenint molt bons resultats en ambdós casos. Per una banda, la metodologia ha permès identificar la variable i l’etapa del procés en les què s’observa millor la diferència de granularitat en un reactor seqüencial per càrregues en el procés de depuració d’aigües, mentre que per l’altra banda, la metodologia ha permès relacionar cadascuna de les peces injectades amb el seu defecte així com les contramesures per corregir futures aparicions dels defectes.

Paraules clau: monitoratge de processos per lots, anàlisi de components principals, raonament basat en casos, diagnosi basada en casos, plantes de depuració d’aigües, procés d’injecció en motlles.

Resumen

El objetivo de esta tesis es la de presentar un método automático para la monitorización de procesos por lotes basado en la combinación de modelos estadísticos y métodos de aprendizaje automático. El primero se utiliza para modelar el proceso mediante las relaciones más significativas entre las variables medidas a lo largo del tiempo, mientras que el segundo se utiliza para mejorar la capacidad de diagnóstico del sistema. Los métodos estadísticos no relacionan los fallos con su causa raíz (sólo listan el subconjunto de variables cuyo comportamiento se ha visto alterado por la aparición del fallo), así como no poseen capacidad de aprendizaje. El hecho de usar el razonamiento basado en casos para el diagnóstico permite relacionar las observaciones con fallo con información más significativa (como puede ser su causa). Los modelos estadísticos también proporicionan una nueva representación de las observaciones, en una base orotogonal, que facilita la aplicabilidad de los métodos basados en distancia del razonamiento basado en casos, presentando unos mejores resultados.

La información proporcionada por la proyección sobre el modelo estadístico se ha utilizado para definir vecindarios significativos para la monitorización de procesos por lotes. Asimismo, se han definido nuevos métodos para minimizar el impacto de los casos “ruidosos” a la hora de reutilizar la información recuperada de los casos anteriores, así como procedimientos para minimizar la base de casos sin alterar su eficacia. Con este objetivo en mente, toda la información asociada a un lote (configuración del proceso, variables medidas, etapas del proceso, entre otros), tomadas a lo largo del tiempo y la información relacionda con el diagnóstico del lote (causa raíz del fallo, dirección de fallo, acciones correctivas, etc.) se utilizan para su caracterización como caso, que luego se agrupan en la base de casos para futuros diagnósticos.

Con el fin de probar la metodología, se han desarrollado dos aplicaciones diferentes (depuración de aguas y máquinas de inyección en moldes) basadas en la metodología propuesta. La metodología se ha aplicado tanto “off-line” como dentro de la línea de producción (una vez finalizado el lote) obteniendo muy buenos resultados en los dos campos. Por un lado, la metodología ha permitido determinar la variable y etapa del proceso que permite diferenciar mejor entre los dos grados de granularidad de un reactor secuencial por cargas en el campo de depuración de aguas, mientras que por otro lado, la metodología permite relacionar cada una de las piezas inyectadas con su defecto, así como las medidas correctivas para evitar su futura aparición.

Palabras clave: monitorización de procesos por lotes, análisis de componentes principales, razonamiento basado en casos, diagnosis basa en casos, plantas de depuración de aguas, proceso de inyección por moldes.

Abstract

The aim of this thesis is to present a methodological approach for the automatic monitoring of batch processes based on a combination of statistical models and machine learning methods. The former is used to model the process based on the relationships among the different monitored variables throughout time, while the latter is used to improve the diagnosis capabilities of the system. Statistical methods do not relate faulty observations with its root cause (they only list the subset of variables whose behaviour has been altered) and they lack of learning capabilities. By using case-based reasoning (CBR) for the diagnosis, faulty observations can be associated with more significant information (like causes). Statistical models also provide a new representation of the observations, on an orthogonal basis, that improves the use of the distance-based approaches of the CBR, giving a better performance.

The information provided by projecting on the statistical model has been used to define neighbourhoods meaningful for batch process monitoring. Additionally, new methods for reusing the information retrieved from previous experiences were defined such that the influence of "noisy" instances was minimised, as well as a new procedure to maintain the case pool to a minimum, while keeping its performance. In order to do so, all the information related to a batch (process configuration, measured variables, stages of the process, among others), measurements along time and information related to its diagnosis (root cause of the fault, fault direction, actions to correct the misbehaviour, etc.) are used to characterise the batch as a case, and grouped in a case base to diagnose future observations.

In order to test the methodology, two different application examples (wastewater treatment plants and injection moulding machines) have been developed based on the proposed methodology. Both off-line and real-time release (at the end of the batch) applications for both fields have been carried out, obtaining good results for both approaches. On the one hand, the methodology permitted the identification of the variables and stages for which the granulation change in a sequencing batch reactor was most observable for wastewater treatment plants; while on the other hand, the application permitted the traceability of all injected parts, while indicating the defect found on the piece (fault-free, sink marks, flashes or locally oversize) and indicate possible countermeasures to correct these defects.

Keywords: batch process monitoring, principal component analysis, case-based reasoning, case-based diagnosis, wastewater treatment plants, injection moulding process.

1

Introduction

This thesis summarizes the effort of combining statistical and case based reasoning methods for fault diagnosis purposes of batch processes. Both fields are mature areas with large number of applications and methodological refinements to deal with fault detection and diagnosis under different assumptions. Thus, main contribution reported in the thesis is on defining a formal framework to build hybrid solutions that exploit modeling capabilities of multivariate statistical methods and at the same time the representativeness of isolated observations collected during abnormal, faulty or simply significant situations useful for supervision purposes. The thesis aims to provide a formal methodology but at the same time it is strongly oriented to real world applications.

This chapter introduces the main motivations, objectives and background of the work thesis, as well as it outlines the main contributions of the chapters.

1.1 Introduction: General approach for data driven process supervision

Process supervision is the set of tasks oriented to guarantee the operation of processes, even under the presence of faults (fault tolerant). This means that, in a general senses, a supervision system is capable to assess the process behaviour, detect faults, diagnose them and propose reconfiguration actions. This thesis focuses on fault diagnosis in a general sense. Formally, fault diagnosis is the ask that follows fault detection and consists in inferring information useful to isolate (locate) and identify (quantify) faults using some logical procedure to analyse discrepancies between observed data and the process model. In this thesis, a data driven approach is followed, so process models are not describing physical laws but data relations, but at the same time we can exploit additional information contained in the historical registers, useful to classify or differentiate faults according to some simple description related to causes or origin for instance. Thus, we consider diagnosis tasks as the capability to infer information related to fault location (affected component/s, fault direction, fault isolation), identification (magnitude or quantification of the fault) and origin (identification of possible causes).

Fault detection and diagnosis is strongly dependent on the *a priori* knowledge available (Venkatasubramanian et al., 2003c). This usually consists of a description of the normal operating conditions (NOC) and some additional information useful to guide analysis of inconsistent observations given by abnormal operating conditions (AOC) . Depending on the available knowledge to describe the normal operating conditions, the fault diagnosis techniques can be divided into two groups: model-based (physical models, from first principles) methodologies and process history, also known as data-driven methods or model-free methods (Milne, 1987). At the same time, the model-based approach can be divided into qualitative and quantitative. Qualitative models are high level models that describe the influence among variables (causality for example). For example, they can describe functions and/or structural properties of the systems. On the other hand, the quantitative models rely on mathematical relationships, typically mathematical equations describing the process with ordinary differential equations (ODE) and algebraic relationships representing system dynamics, physical behaviours or simply mass or energy balances. Figure 1.1 represents this classification, and derived approaches. For a brief introduction to each method and further references follow Venkatasubramanian et al. (2003c), Venkatasubramanian et al. (2003a) and Venkatasubramanian et al. (2003b).

This work focuses on batch processes; these usually have complex dynamics conducted by a “recipe”that drives the production of a finite quantity of end product by combining and processing finite quantities of raw material. Typically, batch processes operate at several working points and conditions according to the recipe, and the representative variables usually experience characteristic non-linear trajectories whose

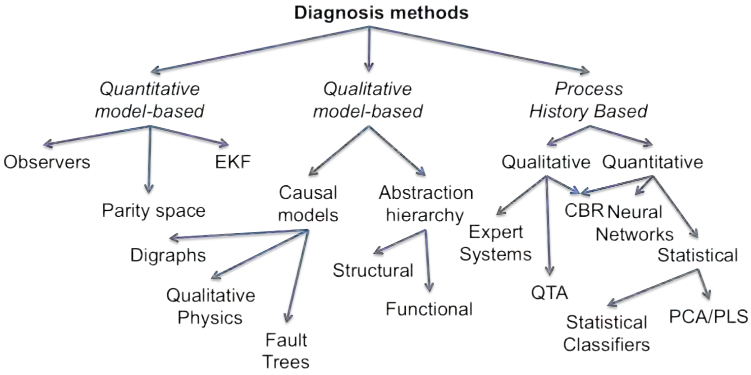


Figure 1.1: Classification of diagnosis methodologies according to the *a priori* knowledge strategy

evolution has a direct cause-effect with the quality of the final product. Batch process are common in the chemical and pharmaceutical industry (autoclaves, reactors) but they can also be found in the manufacturing (e.g. injection, assembling, etc.) and food industry (ovens, breweries, driers, etc.). Usually, different phases can be distinguished in the operation of a batch process and product evolution depends on the final conditions reached at the previous stage and on characteristics of input materials. This dependency on not always observable conditions (input materials and intermediate states) and presence of non linear behaviours makes difficult to obtain analytical models, based on first principles, useful for supervision tasks (Yao and Gao, 2009). Therefore, data-driven methods based on historical data have been used to model their normal operating conditions, especially by means of multiway principal component analysis (MPCA) (Nomikos and MacGregor, 1994, 1995b). The general scheme when using data-driven methods for process monitoring is depicted in Figure 1.2.

As it can be seen in Figure 1.2, process monitoring based on data-driven methods, and consequently also statistical methods, is divided into two main stages: model building, performed off-line using historical data collected during normal operating conditions; and, the on line exploitation of such a model for fault detection and diagnosis purposes. Different approaches can be used to build the model, the goal is to identify and relations among observed variables during normal operating conditions. In this work we selected multivariate statistical methods because they provide a solid theoretical background and at the same time they have been proved to perform quite

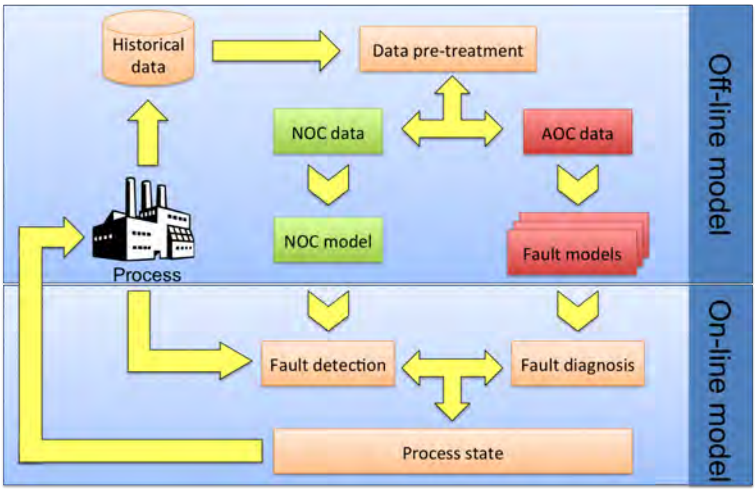


Figure 1.2: General scheme for data-driven process monitoring

well with real processes (statistical process control). *Control charts* based on statistics representing the normal operating conditions are simple plots of some statistics that represents process evolution and allows to detect deviations from the model (fault detection) and to identify the variables responsible for that (fault diagnosis) (Woodall, 2000).

Common tasks when following a data driven strategy follow the *Data Mining* approach. So, the data set used to build or train the model usually known as training set, is submitted to a cleansing process in order to remove outliers, corrupted and blank data or other wrong and non representative observations. Pretreatment also includes data transformation oriented to remove non-linearities and enhance the significance of the features. Multivariate statistical methods usually apply a centring (zero mean) and scaling preprocessing (unit variance) to the observations.

Once the data has been pre-treated, an initial model of the process is built and validated. Usually a testing data set with a known distribution is used for that purpose. The objective is to check both the capabilities to detect faults and recognise normal operating conditions. Performance indices as false alarms/missed detections ratios, accuracy or the receiver operating curve (ROC) are used to evaluate the quality of the model for the pursuit objective. This off-line analysis may pinpoint discrepancies in the observations collected during normal operating conditions due to previous process

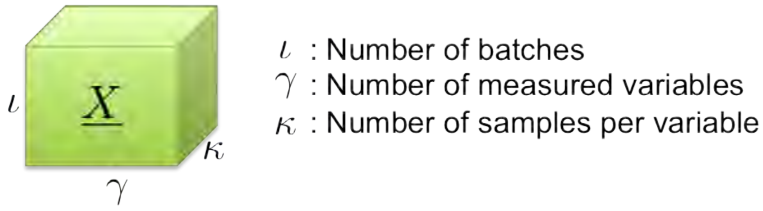

$\underline{X}$
ι
γ
κ
ι : Number of batches
γ : Number of measured variables
κ : Number of samples per variable

Figure 1.3: Three-dimension data matrix associated with historical records of batch processes

misbehaviours (not always known a priori), as well as it improves the process understanding and the inherent relationships between measured variables, causes of faults or the quality of the "product" being produced (Ferrer, 2007).

After the validation of the obtained model (from data), the resulting model can be applied on-line (while the process operates) to detect misbehaviours or deviations on the process operation or in the quality of the product. This new stage is usually referred as on-line monitoring. Discrepancies between the model, obtained with historical data, and the current state of the process alerts of a possible misbehaviour on the process operation due to components (sensors, actuators, devices machines), materials (different quality of raw material, mixing proportions, etc.) or interaction among them.

As stated before, batches vary along a finite time according to the "recipe" driving the process. Therefore, batch monitoring extends for periods of time that depends on the "recipe" along the batch cycle. As a result, historical records of batches are usually represented using three-dimensional matrices as depicted in Figure 1.3.

In this work, we propose the combination of multivariate statistical methods, to perform fault detection, with a case based reasoning strategy to improve diagnosis based on past experiences and give learning capacities to the advanced monitoring system for batch processes.

This thesis formalises in a single framework the results of a research line in the eXiT group at the Universitat de Girona (UdG) and it is funded on previous works developed on the Ph. D. Thesis (Ruiz, 2008) that explored the applicability of the combination of MPCA and case-based reasoning (CBR) to improve diagnosis performance of a sequencing batch reactor (SBR) , and two Master Thesis that further developed this procedure (Berjaga, 2008; Burgas, 2013) and contribute to support the formalism presented in this thesis by means of a Matlab Toolbox implementing the required algorithms. With respect these previous works, this thesis contributes in the following aspects:

- The neighbourhood concept has been formalised using latent structures. This formalisation is consistent with previous works in the field and with Ruiz (2008), as well. Additionally, it has been related to the fault direction concept, allowing to diagnose faults in batch processes based on a similarity concept. The different retrieve criteria and neighbourhoods are formulated in section 3.3 in chapter 3.

- New reuse procedures based on latent structures have been defined with the purpose of overcoming the limitations of the simple voting procedure initially proposed in Ruiz (2008). These methods are detailed in section 3.4 in chapter 3.

- A new update policy to retain cases has been defined. It has a lower computational cost than traditional strategies such as decremental reduction optimisation procedure 4 (DROP4) or instance-based learner 3 (IB3). Further details in all these methods can be found in section 3.6 in chapter 3. Additionally, Appendix B compares the computation time for all methods detailed in this thesis to demonstrate the lower computation time of the proposed algorithm.

- *n*-Fold Cross-Validation is proposed in the validation step in order to take advantage of all available batch processes for training and testing purposes. Additionally, the methodology performance evaluation is now based on the confusion matrix and its associated statistics. This also allows us to identify the best configuration of the CBR using the area under the receiver operating curve (AUC). Details of all these methods and procedures can be found in section 3.5 in chapter 3.

- Different domains for validation purposes have been indtroduced. Injection moulding machines have been used to test the generality of the methodology. Further details of the results in this domain can be found in chapter 5. Additionally, the principles used to build statistical models of batch processes have been applied to finite duration processes (such as voltage sags or to locate the impact location in a commercial flap wing). However, these results have not been included in this thesis is focused in batch processes. Further details regarding this point can be found in section 2.5 in chapter 2.

1.2 Motivation of the work: improving multivariate statistical methods

As stated in the previous section, batch processes present a series of characteristics not found in continuous processes, such as different variable relations depending on the stage the process is in, different duration based on some indicator variable, non-linearities, etc. As a result, conventional data-driven methods cannot be directly applied to historical records of batches. Given the great importance of batch processes in industry, several works discussed the way to adapt traditional multivariate statistical methods to monitor batch processes, such as Lohmöller and Wold (1980), Wold et al. (1987), Geladi (1989) or Smilde and Doornbos (1991). However, the procedure devised by Nomikos and MacGregor in Nomikos and MacGregor (1994) for batch process monitoring was the one that caught the practitioners eye: multiway principal component analysis (MPCA) . The basis of this procedure of this method are detailed in chapter 2, but it presents a series of limitations that are the constant focus of research.

The main problem of using only statistical methods makes difficult to develop efficient strategies for diagnosis. Thus, diagnosis is reduced to strategies oriented to infer the most probable variables that explain a faulty situation (Kourti, 2005). Because in most cases this information is insufficient to determine the fault affecting the process (several faults can affect the same subset of variables), additional procedures are required for this task. On the one hand, clustering techniques such as k-means can be applied to group observations into different clusters (either in the projection or residual subspaces), and then relate the misbehaviours with their root cause like in Ding and He (2004) and Napoleon and Pavalakodi (2011). On the other hand, series of observations with the same misbehaviour can be used to estimate the fault direction inherent to the fault as in Valle et al. (2001). However, both approaches require a sufficiently large and rich set of faulty observations for each fault to diagnose, which is not the usual case when monitoring within the production line.

To overcome this limitation in the number of observations available for any given faulty situation to diagnose, this thesis proposes a combination of multiway principal component analysis (MPCA) and case-based reasoning (CBR) , denoted as case-based diagnosis (CBD) for the automatic diagnosis of batch processes. MPCA is used to model the normal operating conditions of the batch process, while CBR is used to either relate the different faulty observations with their root or to track the operating point of the process. Moreover, the inherent reasoning procedure behind the CBR (new problems are solved based on solutions previously known) facilitates the interaction between the monitoring procedure and the process operator, because it can indicate the steps followed in previous situations to solve the current problem. Note that in order to do so, the aid of process experts is required in order to identify the root cause of the defective product with the batch measurements.

Regarding the interactions between both procedures, note that the information provided by MPCA is used to characterise the cases (representation of a problem and its solution: a diagnosed batch) in the CBR. More specifically, MPCA provides uncorrelated attributes (scores) to the CBR, which is enough to guarantee that adding new attributes adds new information, as well as to grants a simpler implementation and a good performance (Jakulin and Bratko, 2003). On the other hand, the CBR reduces the necessity for a large number of examples since the case base can help identify rare situations (with few examples). Therefore, this methodology first, builds a statistical model of the process studied; and then the CBR is used as the diagnosis tool.

1.3 Objectives

The final objective of this thesis is to develop a general methodology for the diagnosis of batch processes based on the combination of multivariate statistical methods and case based reasoning (CBR). Multivariate statistical model built on the basis of principal component analysis will be used for both, fault detection and as a projection technique to obtain improved case representation. Historical cases (faulty and non-faulty) projected into the PCA space will be used to implement diagnosis tasks following a CBR strategy giving and providing (lazy) learning capacities to the monitoring system (every time a new fault or misbehaviour occurs it can be preserved in the case base, once diagnosed, for further reuse).

In order to obtain this final goal, the following partial (sub)objectives have been envisioned to improve the performance of multivariate statistical methods:

- Formalise a representation of cases in the projection space capable to represent batch process and exploit such representation for diagnosis.
- Define neighbourhoods on the latent structures (principal component space) useful for diagnosis based on similarity principles.
- Reduce the effect of noisy instances when reusing the information extracted from neighbourhoods based on latent structures.
- Specify new case base maintenance and updating policies to minimise the case base, while keeping its diagnosis performance.

The methodology has been tested in different fields to guarantee its generality. Two main domains, wastewater treatment plants (WWTP) and injection moulding machines (IMM) have been selected to illustrate the methodology and quantify performance in this text:

- Waste water treatment plants (WWTP). In this case, the methodology was applied to predict changes in the behavior on the granularity of the active sludge with two different goals. On the one hand, the methodology was used to find the best set-up for differentiating two granulation states (floccular and granular), as well as to identify for which variable(s) and process stage(s) this differentiation was most observable. On the other hand, an on-line application was carried out for which only a subset of the initial observations was available. Therefore, the methodology had to detect and learn about the new situation, in this case, a granulation change. Results presented in this field were partially published in

Berjaga et al. (2013a) and Berjaga et al. (2013b). Additionally, a study indicating the applicability of the methodology in this field was published in Ruiz et al. (2011).

- Injection moulding machines (IMM). Like in the previous field, two different scenarios were considered. The first scenario considers an off-line monitoring application to differentiate between normal operating conditions from four different fault causes (early switch-over, no holding pressure, and late switch-over with or without holding pressure). The objective of the second experiment was oriented to test the methodology in an a production system. In this case the CBR engine was in charge of proposing corrections to overcome the diagnosed problem in further injections. Additionally, mechanisms to automatically update the statistical model and case base were added to track the current state of the process and its performance. Results of the methodology application in this field were partially published in Berjaga et al. (2008a), Berjaga et al. (2009c), Berjaga et al. (2009a) and Berjaga et al. (2009b).

1.4 Statistical methods for batch process monitoring: related work

As stated in section 1.1, data-driven methods are used for batch process monitoring. And from the different strategies detailed in Figure 1.1, the statistical branch is the most used for this purpose. Statistical techniques have been used for process monitoring from the earlies 20's. A set of tools known as Statistical Process Control (SPC) has been developed for this purpose. This methodology was firstly introduced by Walter Shewhart in the 1920s and was based on the usage of *SPC control charts* to adapt the management processes (Hare, 2003). This would allow the creation of profitable situations for both consumers and producers. As time passed, the usage of the SPC was more than only the application of control charts and eventually became used in the manufacturing process. This evolution was also reflected in the change of the original idea of basing the control limits in economic limits to use the process history to compute statistical control limits based on the probability of group variations.

Batch processes present non-linear time variations and working points dependent on the "recipe" governing the process. Because of this, several measurements have to be taken along time on batch processes in order to register the evolution of the process throughout the batch duration. Since the final quality of the product also depends on the reactions taking place, several variables have to be measured for each sample. Finally, in order to find the normal operating conditions of the batch process, a historical pool of batches is required to obtain the expected behaviour of normal operating condition (NOC) processes. As a consequence, batch processes are represented using a three-dimensional matrix (batches × variables × samples). Considering that traditional SPC methods, such as principal component analysis (PCA) or partial least squares (also denoted as projection to latent structures, PLS) were designed to work with two-dimensional matrices, they cannot be directly applied to monitor and diagnose this kind of process. However, Nomikos and MacGregor (1994) and Nomikos and MacGregor (1995a) developed an extension of these methods to deal with the three-dimensional matrices associated with batch processes: multiway principal component analysis (MPCA), also denoted as unfold-PCA, and multiway partial least squares (MPLS), which is also referred as unfold-PLS, which were later on applied on industrial batch and semi-batch polymerisation processes, and the development of new methodologies such as multiblock extensions of both PCA (MB-PCA) and PLS (MB-PLS) like the ones reported in Kosanovich and Piovoso (1994), Kourti et al. (1995), MacGregor and Kourti (1995), Nomikos and MacGregor (1995a) and Nomikos and MacGregor (1995b).

The main points that facilitated the implementation and integration of these methods can be summarised in the following points described in Kourti (2005):

- The operator has to check a reduced set of control charts (based on the latent structure that govern the process).
- These control charts are easily interpretable.
- The utilisation of these methods makes possible the early detection of faults since they look for correlation changes and/or extreme observations that do not follow the process statistical model, which is not possible to do when using univariate statistical control methods.

Because of the information compression inherent to both PCA and PLS (and consequently of their extensions), as well as their visual-oriented process monitoring indices (further detailed in chapter 2), these methods have been applied in several fields in the recent years. Some of the fields these techniques have been directly applied with good results are: wastewater treatment plants (Rosen and Lennox, 2001), (Aguado et al., 2007), autobody assembly process (Ferrer, 2007), voltage sag relative location (Khosravi et al., 2008), ceramic industry (Zahid and Sultana, 2009), electronic devices assembling (Reis and Delgado, 2012), among others.

As the title of this thesis indicates, its main concern is the diagnosis of a batch processes. As has been indicated in previous sections, and further explained in chapter 2, MPCA models only indicate the subset of variables affected by a certain (detected) fault. This means that the root cause that triggered the fault is not pointed out, and thus, additional steps (outside of the model building process) have to be carried out. Within the literature, two main strategies can be identified: 1) build specific statistical models capable of relating variables and root causes; or 2) use the information provided by the MPCA model with other artificial intelligence (AI) methods to relate faults with the variables they affect. Note that both strategies require a rich and sufficiently large set of observations for each fault to diagnose. Moreover, we assume that observations of a given fault misbehave in a similar way (low intra-class variation) and they project in a different subspace than other faults and the normal operating conditions (high inter-class variation).

When building statistical models of the faults, one can take into account the set of faults to build the statistical model. This is the starting point of the variance of the reconstruction error method described in Dunia and Qin (1998). More specifically, this method builds a statistical model capable of detecting and correcting faulty values associated to any of the known fault directions. This work served as basis of the work in Alcala and Qin (2009), which guarantees that the resulting statistical model is tolerant (can operate normally even when certain faults are present) to simple sensor faults (see section 2.1 for an insight in types of faults). The other option when building statistical models of the faults is to compare the faulty observations for each fault with the normal operating conditions of the process. The main idea is to determine which variables present the most important variations for a given fault, and thus, relates root causes (faults) with the variables they affect (the subset of variables whose behaviour has been mostly modified due to the fault). This is the strategy behind works such as Cedeño Viteri et al. (2012), which states a new way to estimate the variables affected by a fault based on the contribution to the T^2 index of the closest in-control neighbour; or Valle et al. (2001), which determines the subset of faulty variables by using a singular value decomposition and compare the resulting model with the normal operating conditions. However, this approach requires the recalibration of the model whenever a new fault is discovered, since the statistical model takes into account either the projection of faulty observations into the statistical model (Cedeño Viteri et al., 2012) or are used to select the variables and number of principal components (Dunia and Qin, 1998).

In cases where new faults can appear throughout the process monitoring, the previous approach is not a viable option. Because of this, the task of relating faults with the faulty variables and simplify the fault diagnosis procedure, a viable alternative is the combination of the information provided by the statistical model with "external" clustering methods. The idea is that the clustering method will look for a combination of the information provided by the statistical model such that the diagnosis task is reduced to a classification problem, and thus, the effects of faults and variables affected are related indirectly (similar misbehaviours are close in the monitoring space, and thus, clustering techniques look for this correlation to group them). Based on this principle, two new methods were devised in order to classify observations in the monitoring space according to their operating conditions: 1) soft independent modelling of class analogy (SIMCA) (Wold et al., 1984); and 2) partial least squares with discriminant analysis (PLS-DA) (Sjöstrom et al., 1985). The former builds a statistical model for each situation in such a way that the NOC region is defined by the modelled situation and the AOC region is related to the rest of observations (other faulty or normal operating conditions). The latter the projection in the monitoring space to guarantee that observations within a certain operating point are close among them (minimises the intra-class variability thanks to the PLS) while at the same time separates each situation from the rest (maximises the inter-class variability due to the discriminant analysis). Note that both approaches use linear combinations to guarantee this, but if this is not the case for the current set of observations, then non-linear techniques have to be applied for this matter. This was the case for example in Guo et al. (2006), where a moving window generalised regression neural network (GRNN) was used over the stage-divide MPCA model to determine the quality at the end of each batch stage in an injection process. Another example is the one presented in Mujica et al. (2008), where MPCA and MPLS where combined with a CBR to locate the impact zone in a commercial flap wing. Hidden Markov models were applied in conjunction with MPCA in order to track the progress and diagnose any misbehaviours in two benchmarks in Chen and Chen (2006); and later on in Chen and Jiang (2011) the procedure was extended for multiphase batch processes.

Since in this thesis the diagnosis relies on the CBR, a hybrid solution capable to integrate statistical and instance based methods, major information can be contained in the case representation for diagnosis purposes. Since CBR is a non-linear classification method, we tackle two problems inherent to fault-dependant statistical models (the ones that use the set of faults to build the statistical model):

- We can relate faults with the variables they affect even in case of non-linear relationships, which is a problem when using the set of known faults to build the statistical model since faults have to be linearly separable in the monitoring space.
- We do not have to rebuild the statistical model when new faults appear in the process, contrarily to the fault-dependant statistical models. Moreover, given the learning capabilities of the CBR, this combination reduces the necessity of large and representative observation pools for new faults, allowing an earlier detection and diagnosis than the fault-dependant statistical models.

1.5 Outline of the thesis

This thesis document is organized in six chapters. This first chapter served as an introduction and contained some of the fundamentals of the thesis.

Chapter 2 introduces the basic notions of principal component analysis (PCA) and its extension to deal with batch processes (MPCA). The chapter serves to introduce the background on multivariate statistical process control and revises the state of the art of batch process monitoring using MPCA. A first approach of using MPCA based solely on the information provided by sensors installed in an injection moulding machine was presented in Berjaga et al. (2008a) and will be used to illustrate the characteristics of batch process and MPCA models. This will be the basis of the diagnosis strategy presented in the following chapter. Additionally, these techniques can be applied to finite-duration events, i.e. to locate the impact in a commercial flap wing as was demonstrated in Mujica et al. (2009) and in Ruiz et al. (2013), provided that the data is organised in a batch-way. Further details on how other processes can be adapted to work with these methods will be detailed in this chapter.

Chapter 3 introduces the concepts related to case-based reasoning (CBR), and how it can be combined with the information provided by statistical methods for the monitoring of batch processes. More specifically, this chapter will present several neighbourhoods based on the latent structures and will relate them with several process monitoring concepts (partially published in Berjaga et al. (2009a) and in Berjaga et al. (2009b)); and two new reuse procedures are defined to take into account the distribution of batches within the monitoring space (partially published in Berjaga et al. (2013a) and Berjaga et al. (2013b)). Additionally, these neighbourhoods were also applied to voltage sags in order to locate their relative origin in Meléndez et al. (2008b), Meléndez et al. (2008a) and Barrera et al. (2008). Note that although voltage sags are not batch processes, they are finite-duration events, and based on the principles depicted in chapter 2, MPCA can be used on them.

Chapter 4 presents the results of applying the CBD methodology to the WWTP field. Both an off-line analysis and on-line application to determine the granulation state (floccular or granular) of the activated sludge of a sequencing batch reactor (SBR) are detailed. Additionally, the off-line application was also used to indicate the variable(s) and stage(s) for which the difference between states was most observables. Results in this field were partially submitted as a journal paper in Berjaga et al. (2013b) and a conference paper in Berjaga et al. (2013a).

Chapter 5 details the results of applying the CBD to the IMM field. Again, an off-line analysis of the process and an on-line application are explained, for which the methodology indicates the final quality of each injected part at the end of the batch. Additionally, the CBD proposes a series of steps to correct the quality defect for future injected parts. Results in this field were partially published in (Berjaga et al., 2009b).

Chapter 6 presents the main conclusions and futures applications of this thesis. A first approximation to the problem of fault reconstruction (one of the future works detailed in this chapter) was presented in Berjaga et al. (2010), which established the basis for the application of the methodology into batch processes. Finally, an appendix explaining the multiway partial least squares with discriminant analysis is included, which can be used to classify observations based on their projection in the principal component subspace (PCS) .

2

Batch process diagnosis based on latent structures

This chapter presents the basic concepts used for monitoring batch processes using latent structures (LS). Firstly, the basic concepts associated with process monitoring are introduced in section 2.1. Then, the fundamentals of principal component analysis (PCA) and its extension to proceed with the monitoring of batch processes (unfold-PCA, also denoted as multiway principal component analysis (MPCA)) are presented in section 2.2. The next two sections in this chapter present the procedures for fault detection (2.3) and diagnosis (2.4) when working with LS. Finally, section 2.5 indicates how MPCA can be applied to finite-duration processes and the information the statistical model provides about the process.

2.1 Concepts

The definitions of the basic concepts related with fault detection and diagnosis of processes are introduced in this section. Since methods considered within this thesis are discrete (only used the values gathered from sensors), all equations within this section are based on the available sample k at a given time instant.

Definition 2.1. Process. A process (or system) is defined as a function that depends on the input vector ($\vec{u}(k)$) at a given time and the current state of the process ($s(k)$), at the same time, the output of the system ($\vec{y}(k)$) at that instant is a function of the

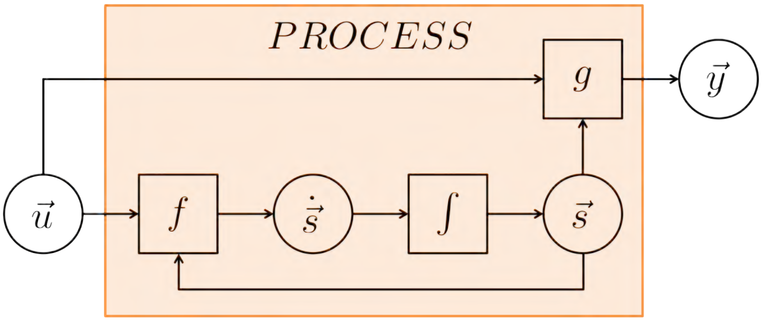


Figure 2.1: Example of a general process relating the input vector ($\vec{u}$), the state of the process ($\vec{s}$) and the output of the system ($\vec{y}$)

input vector and the current state of the process ($s(k)$) (Figure 2.1) expressed as:

$$\dot{\vec{s}}(k) = f\left(\vec{u}(k), \vec{s}(k)\right) \tag{2.1}$$

$$\vec{y}(k) = g(\vec{s}(k), \vec{u}(k)) \tag{2.2}$$

Definition 2.2. Batch process. A batch process is a finite-duration process consisting of three main steps: 1) charge the batch vessel following a specific recipe of materials; 2) process it under controlled conditions during which process variables such as temperatures, pressures, agitation, and feedrates are varied according to specified time trajectories; and 3) discharge of the product (Nomikos and MacGregor, 1995b).

In order to track the variations of the monitored variables, several samples have to be taken for each run, and consequently, a single batch can be described as a series of observation vectors (a two-dimensional matrix), and a historical record of batch process is contained in a three-dimensional matrix, as depicted in Figure 2.2. Further details regarding batch process and these two- and three-dimensional matrices are explained in subsection 2.2.2.

Definition 2.3. Process monitoring. Process monitoring is a continuous real-time task of determining the state of the physical system, by gathering information, and from it, recognise and indicate any anomalies in its behaviour (Isermann and Ballé, 1997).

In order to check the state of the process, process monitoring identifies the relationships between the variables (either measured or estimated) and the respective outputs

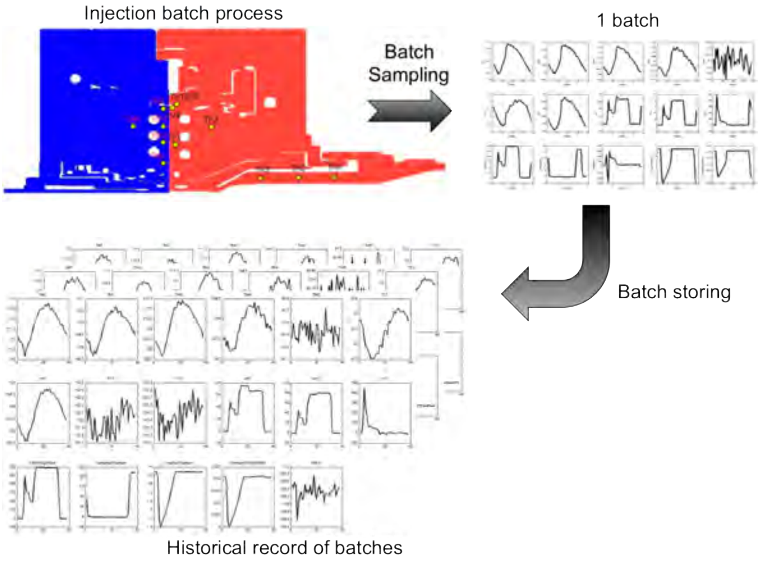


Figure 2.2: Example of an injection batch process: relation between 1 single batch (two-dimensional matrix) and the historical record (three-dimensional matrix)

of the process. Whenever these relationships do not hold, the process is considered to be under abnormal operating conditions (AOC), which is usually associated with the appearance of a fault. Since only the input and output vector of the process are available to check the state of the process, their information is usually grouped for each sample and labelled as observation vector ($\vec{x}(k)$) in this way:

$$\vec{x}(k) = \left[\begin{array}{cc} \vec{u}(k) & \vec{y}(k) \end{array} \right] \tag{2.3}$$

This means, that only those faults that alter the observation vector (denoted as observable faults) can be detected, and consequently, diagnosed.

Definition 2.4. Fault. A fault is a non-allowed deviation of part of the system, which causes the system to not accomplish the function it was originally designed for (Isermann and Ballé, 1997).

One way to classify faults is based on how they affect the process parameters (Gertler, 1998), and results in the following types:

- **Additive faults**. They are unknown sources to the process that cause a variation to either (or both) the input and/or the output vector of the process that cause

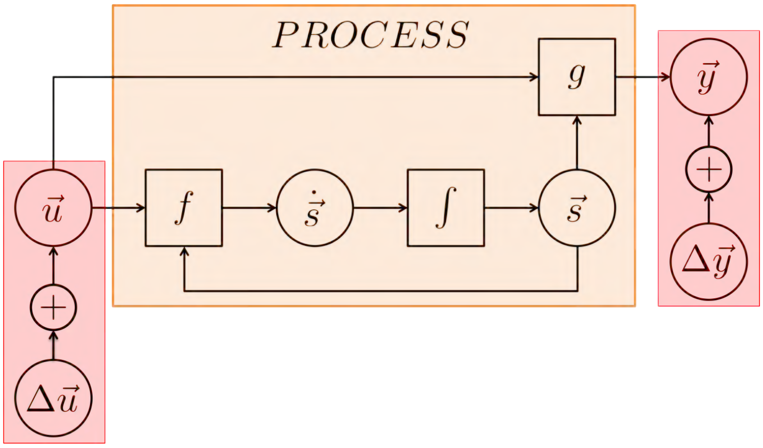


Figure 2.3: Effect of an additive fault (red-shadowed elements) over the general system representation

a variation of the normal behaviour of the process. They are independent of the process, and thus provoke a variation of the outputs, modifying (2.2) in this way:

$$\dot{\vec{s}}_f(k) = f\left(\vec{u}(k) + \Delta\vec{u}, \vec{s}(k)\right) \tag{2.4}$$

$$\vec{y}_f(k) = g(\vec{s}_f(k), \vec{u}(k) + \Delta\vec{\xi}f) \tag{2.5}$$

$$= \vec{y}(k) + \Delta\vec{y}(k) \tag{2.6}$$

were $\vec{s}_f(k)$ and $\vec{y}_f(k)$ are the state and output vectors after the fault appearance, and $\Delta\vec{u}(k)$ and $\Delta\vec{y}(k)$ is the effect the fault has over the input and output vectors respectively. The effect of an additive fault over the system depicted in Figure 2.1 is represented in Figure 2.3.

Finally, from an observation vector point of view, an additive fault can be described in this way:

$$\vec{x}_f(k) = \vec{x}_o(k) + \vec{\xi}_i f_i = \begin{bmatrix} \vec{u}(k) + \Delta\vec{u} & \vec{y}(k) + \Delta\vec{y} \end{bmatrix} \tag{2.7}$$

where $\vec{x}_o$ is the fault-free values of the observation vector; $\vec{\xi}_i$ is the fault direction; and f_i is its magnitude.

Definition 2.5. Fault direction. A fault direction ($\vec{\xi}_i$) is a vector indicating which input and/or output variables are affected by the appearance of a fault. Throughout this thesis, this vector is assumed to be a unit vector (norm equal to one).

Additive faults can be confused with disturbances, which are an unknown and uncontrolled input acting on the system (Isermann and Ballé, 1997), since they cause an alteration of the input vector. Model-based techniques model them specifically (and thus, they can distinguish between them), while data-driven methods include continuous disturbances as part of the model. In case of uncorrelated disturbances (noise), statistical techniques are capable of removing them from the normal operating conditions of the process, as will be described in section 2.2 later on.

- **Parametric faults**. They are an alteration of one or more parameters that drive the process. Consequently, parametric faults modify the relation between the process inputs and outputs, altering the system in this way (and depicted in Figure 2.4):

$$\begin{aligned} \dot{\vec{s}}_f(k) &= f_f\left(\vec{u}(k), \vec{s}_f(k)\right) && (2.8)\\ \vec{y}_f(k) &= g_f\left(\vec{s}_f(k), \vec{u}(k)\right) && (2.9)\\ &= \vec{y}(k) + \Delta\vec{y} && (2.10) \end{aligned}$$

where f_f and g_f are the new relationships between input and output vectors due to the parametric fault appearance with $f_f = f + \Delta f$ and $g_f = g + \Delta g$; $\vec{s}_f$ and $\vec{y}_f$ are respectively the process state and output vector under the parametric fault; and $\Delta\vec{y}$ is the variation of the output vector due to the fault happening.

Finally, from an observation vector point of view, the parametric fault can be described as:

$$\vec{x}_f(k) = \vec{x}_o(k) + \vec{\xi}_i(k)f_i(k) = \left[\begin{array}{cc} \vec{u}(k) & \vec{y}(k) + \Delta\vec{y} \end{array}\right] \qquad (2.11)$$

From the previous expression, it can be observed that the effect of parametric faults is observable as an additive fault at the system output. Consequently, from this point onwards, a faulty observation ($\vec{x}_f$) for a given sample k will be described as an additive fault in this way:

$$\vec{x}_f(k) = \vec{x}_o(k) + \vec{\xi}_i(k)f_i(k) \qquad (2.12)$$

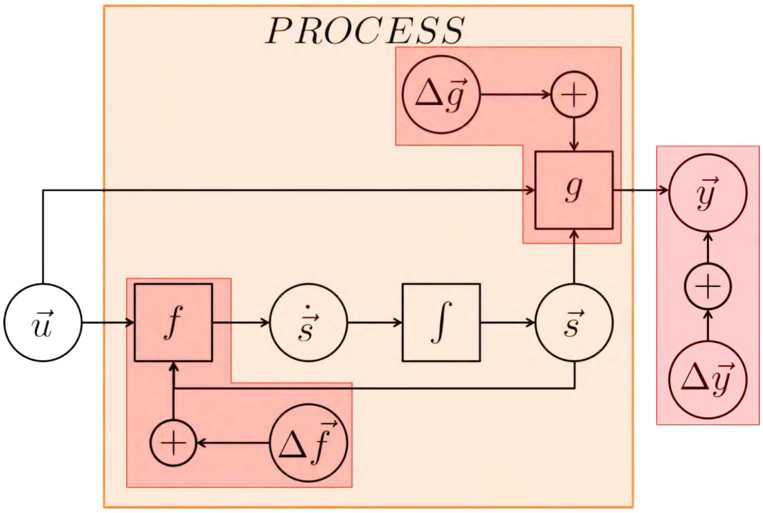


Figure 2.4: Effect of a parametric fault (red-shadowed elements) over the general system representation

where $\vec{x}_o$ refers to the fault-free values of the observation, which are unknown in presence of faults; $\vec{\xi}_i$ is the fault direction affecting the observation and indicates the variables whose behaviour have been modified due to the fault appearance; and f_i is the fault magnitude for the given fault direction. Since batch processes consist of a series of measurements along time $k = [1, \kappa]$, the previous fault definition has to be extended to take into account that faults may occur within a certain time window $k_f = [k_{ini}, k_{end}]$, and assuming that faults are persistent within this time period, in this way:

$$x_f(k) = \begin{cases} x_o(k) & 1 \leq k < k_{ini} \\ x_o(k) + \vec{\xi}_i(k) f_i(k) & k_{ini} \leq k \leq k_{end} \\ x_o(k) & k_{end} < k \leq \kappa \end{cases} \tag{2.13}$$

Based on these definitions, faults can be categorised based on the number variables the fault is observable on in this way:

- **Simple faults**. This kind of faults only alter a single variable in the observation vector. As a result, fault direction vectors ($\vec{\xi}_i$) for simple faults only present a non-zero element (the variable altered) with a value equal to 1. Consequently, only one fault direction ($\vec{\xi}_i$) is detected, and thus (2.12) can be used to define directly this kind of faults.

- **Multiple fault**. Multiple faults are observable on more than one variable, and thus, they can be defined as a subset of simple faults that occur at the same time. Therefore, the fault direction ($\vec{\xi_i}$) and magnitude (f_i) in (2.12) are defined as a sum of NF (orthogonal) simple faults ($\vec{\xi_j}$) in this way:

  $$\vec{\xi_i}(k)f_i(k) = \sum_{j=1}^{NF} \vec{\xi_j}(k)f_j(k) \tag{2.14}$$

  Depending on the physical redundancy of the system, multiple faults can either be related to a single element with physical redundancy, or to several elements misbehaving at the same time with no physical redundancy.

If no information regarding the process structure is available, then additive and parametric faults cannot be distinguished between, since the only indicators of a fault is a series of variables whose behaviour have been altered. This is a common problem with data-driven methods, since structural information is not contained in the historical data of a process. In this thesis, this information can be incorporated as part of the diagnosis structure contained in the CBR, as will be exposed in chapter 3.

2.2 Latent structures: methods and concepts for batch process monitoring

The more sensors are installed in a process, the more redundancy of information is captured. In such situations the use of multivariate statistical techniques, based on the existence of correlations among measured variables, provides a very powerful framework for fault detection based on well proved theoretical background. Principal Component Analysis (PCA) or Partial Least Squares (PLS), also known as Projection to Latent Structures are fundamental techniques that support fault detection and isolation strategies commonly used in chemical industries.

This section introduces the basic idea of PCA (subsection 2.2.1), as well as indicates how this procedure can be extended to work with batch processes (subsection 2.2.2) which defines a series of unfolding procedures (subsection 2.2.3) and scaling methods (2.2.4). Finally, sections 2.3 and 2.4 presents the methods for fault detection and diagnosis based on the PCA information.

2.2.1 Principal component analysis (PCA)

Principal Component Analysis (PCA) is a technique for data compression and information extraction. PCA is used to find combinations of variables or factors that describe major trends in a data set (Wise et al., 1999). This is, PCA is concerned to explain the variance-covariance structure through a few linear combinations of the original variables.

Processes involving a large number of variables can be monitored using this technique. Observations during normal operation conditions (NOC) are used to build a data model. Then, this NOC statistical model is used to assess the behaviour of the process by checking new observations against this model in the principal component subspace (fault detection).

Multivariate data (NOC observations) are expected to be organised in a matrix structure, X, with m variables/columns and n observations/rows. Variables are assumed to be centred (zero mean) and standardised (unit variance).

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix} \tag{2.15}$$

The sample covariance matrix (S) can be computed with the following expression:

$$S = \frac{1}{n-1} X^T X \tag{2.16}$$

And solving an eigenvalue decomposition of the sample covariance matrix S, the loading vectors for this sample can be obtained:

$$S = \frac{1}{n-1} X^T X = V \Lambda V^T \tag{2.17}$$

The orthonormal column vectors in the matrix V are commonly known as *loading vectors*, and the standard deviation of the training set projected along the direction described by the i^{th} column of V , i.e. σ_i, corresponds to the root square of the i^{th} element of the diagonal matrix Λ. That is, the diagonal matrix Λ contains the non-negative real eigenvalues in decreasing magnitude ($\lambda_1 > \lambda_2 > \cdots > \lambda_m \geq 0$) with

$\lambda_i = \sigma_i^2$.

One of the most important characteristic when applying PCA is the dimensionality reduction in the number of variables. This reduction is attained by selecting the first r columns of the loading matrix to build the matrix $P \in R^{m \times r}$; i.e. the loading vectors (eigenvectors) associated with the first r eigenvalues ($r < m$). The projections of the observations in X into the lower dimensional space are contained in the *score matrix*, T, computed as follows:

$$T = XP \tag{2.18}$$

and the score vector for a single auto-scaled observation vector $\vec{x}_i$ can be computed in this way (refer to 2.2.4.2 for further details):

$$\vec{t}_i = \vec{x}_i P \tag{2.19}$$

And the projection of scores, T, back into the m-dimensional observation space ($\hat{X}$) can be computed with:

$$\hat{X} = TP^T \tag{2.20}$$

and for the case of the score vector in (2.19) its projection into the original space is computed in this way:

$$\hat{\vec{x}}_i = \vec{t}P^T \tag{2.21}$$

The difference between X and $\hat{X}$ is the residual matrix $\tilde{X}$ (Russell et al., 2000). It contains a vector for each observation orthogonal to the scores and resumes the variance not captured for the r components selected in the projection subspace (see Equation 2.22). The principal components represent the selection of a new coordinate system obtained by rotating the variables after pre-processing and projecting them into the reduced space defined by the first r few principal components, where the data are described adequately and in a simpler and more meaningful way. The principal components are ordered according to a variance criterion in such a way that the first one describes the largest amount of variation in the data, the second one the second largest amount of variation, and so on (Nomikos and MacGregor, 1994). By retaining

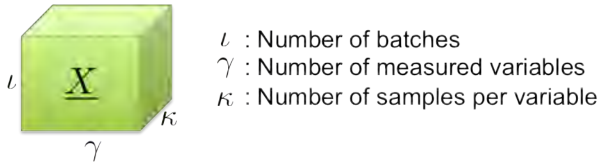

ι
$\underline{X}$
κ
γ
ι : Number of batches
γ : Number of measured variables
κ : Number of samples per variable

Figure 2.5: 3D data matrix associated with finite duration processes

only the first r principal components, the X matrix is decomposed into two parts, the projection ($\hat{X}$) and residual matrix ($\tilde{X}$) according to (2.22) (MacGregor, 2003). Thus, the complete PCA model can be mathematically expressed as follows:

$$X = TP^T + E = \hat{X} + \tilde{X} \tag{2.22}$$

2.2.2 Multiway principal component analysis (MPCA)

The PCA methodology presented before can be directly applied on two-dimensional matrices (*observations* × *variables*). This is the case for example of continuous processes, where observations correspond to time samples. However, batch processes that operates during a finite time to obtain the desired product, require a different representation capable to contain the trajectories of variables during the execution of the process during the whole batch. Usually a three dimensional matrix is used to represent the data (*observations* × *variables* × *time*) as shown in Figure 2.5. This added complexity implies to perform a two steps pre-processing before applying the PCA methodology: unfolding and scaling. In this figure, ι indicates the number of measured batches, γ refers to the number of monitored variables and κ corresponds to the samplings considered for each batch.

2.2.3 Unfolding the data

From the six feasible unfolding directions, only 2 of them are meaningful for monitoring: unfold in the process direction (batch-wise or Nomikos-MacGregor approach (Nomikos and MacGregor, 1994)) and unfold in the variable direction (variable-wise or Wold approach (Wold et al., 1998)). Additionally, a third unfolding procedure was introduced in Chen and Liao (2002): the batch-dynamic unfolding, which can be considered as a

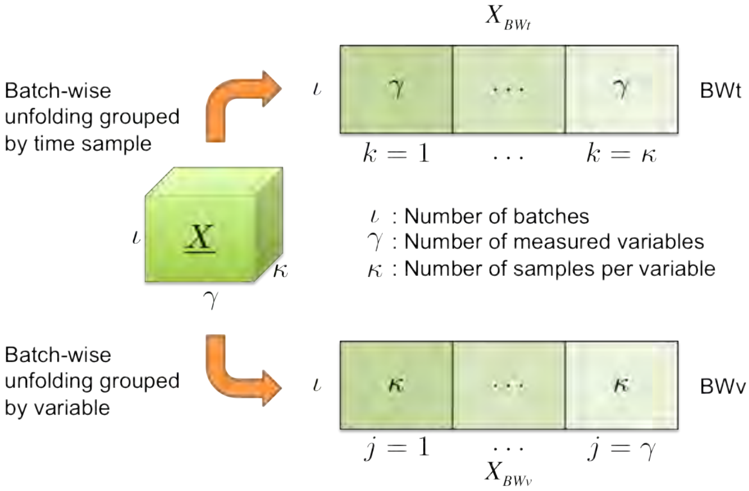

X_{BWt}
Batch-wise unfolding grouped by time sample
ι
γ
$\cdots$
γ
BWt
$k = 1$
$\dots$
$k = \kappa$
ι
$\underline{X}$
κ
γ
ι : Number of batches
γ : Number of measured variables
κ : Number of samples per variable
Batch-wise unfolding grouped by variable
ι
κ
$\cdots$
κ
BWv
$j = 1$
$\dots$
$j = \gamma$
X_{BWv}

Figure 2.6: Batch-wise unfolding of the original 3D matrix, grouping by time samples (upper branch, X_{BWt}) and grouping by variable (lower branch, X_{BWv})

generalised notation for both batch-wise and variable-wise unfoldings.

The batch-wise approach redistributes the time samples of the three-dimensional matrix in the variable axis obtaining a two-dimensional matrix, where each row contains all the samples of every variable (variable × time) for a given batch (Figure 2.6), and a column represents a time instant of a given variable for each process. This approach can only be applied whenever all data of the process is available, this is, when all the samples of the process are available. However, the loading matrix can be used to complete the missing samples as described in García-Munoz et al. (2004) which uses the expected behaviour of the process under normal operating conditions. This means that this unfolding can be applied for monitoring running processes by filling the "missing" values. Since this unfolding procedure represents batches as single points in the projection and residual subspace (as depicted in Figure 2.7), this facilitates the comparison between batches using simple distance and clustering techniques. This property is used in chapter 3 to define neighbourhoods useful for process monitoring, and later on, exploited to differentiate the operating conditions of a sequencing batch reactor (chapter 4) or the final quality of injected parts (chapter 5).

When the standardisation step is carried out with this approach, variations observed

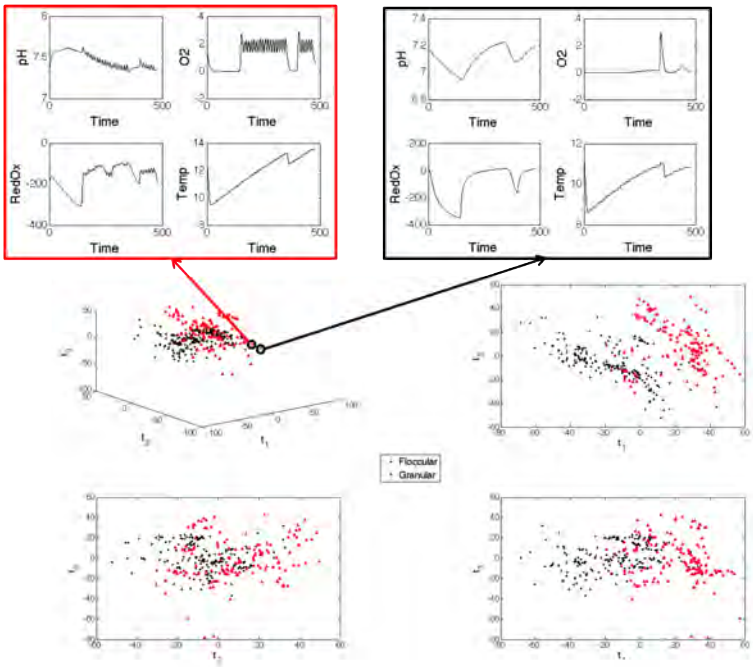


Figure 2.7: Projection of a historical record of batch-wise unfolded batches and relation of two close cases of different operating conditions in a three-dimensional projection subspace

in the unfolded matrix represent variations with respect the mean trajectory (Kourti, 2003). There are two different and complementary unfolded matrices as depicted in Figure 2.6: BWt , which groups variables along time (the first κ columns address the first variable, from $\kappa + 1$ up to 2κ the second variable along time is presented, and so on) and was the one proposed initially in Nomikos and MacGregor (1994); whereas BWv , which groups each measured sample for all variables (the first γ columns indicate the values of all samples at sample $= 1$ for all batches, from $\gamma + 1$ up to 2γ the values of all variables and batches for the second sample are displayed, and so on) and was initially proposed in Westerhuis et al. (1999). Although both matrices contain the same information with the only difference being a different column sorting, BWt directly presents the evolution of each variable throughout time (more easily interpretable by the process operators); and BWv is more appropriate for on-line monitoring, since concatenates all the measurements of all variables at each sample (whenever a new sample is available).

On the other hand, variable-wise approach (Figure 2.8) fixes the variables axis (kept

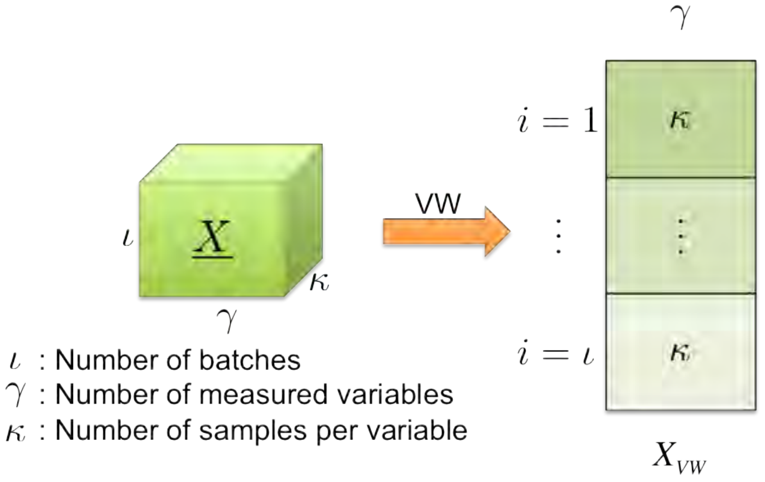

ι
$\underline{X}$
κ
γ
VW
ι : Number of batches
γ : Number of measured variables
κ : Number of samples per variable
γ
$i = 1$
κ
$\vdots$
$i = \iota$
κ
X_{VW}

Figure 2.8: Variable-wise unfolding of the original 3D matrix

as columns) and the product processes × time are the rows. In this case, one row represents a time instant of a given process and the columns are the values of one variable for all processes at every time instant. This approach does not require any prediction to be applied at every time instant, but has the drawback of leaving the non-linear time variations in the standardised data matrix (Kourti, 2003) and assumes that the correlation structure does not change along time, which for most multi-phase batches is not true. Note that the former can be solved by removing the mean trajectory prior to applying this unfolding, but this does not guarantee a constant correlation structure throughout all time samples. Contrarily to the batch-wise unfolding, variable-wise unfolded batches are represented with a series of κ points in the projection and residual subspaces. Consequently, comparing two batches within this unfolding procedure requires the comparison of the dynamic between all κ samples of both batches (this dynamic is associated with the average trajectory of a batch) as time-series. This means that auto- and cross-correlation are not considered when building the model (like in the batch-wise approach) and thus, these relationships have to be identified externally. As a result, points in the residual space are linked to both a batch and a sample within this batch. This circumstance is illustrated in Figure 2.9, where the two batch processes in Figure 2.7 are projected into a variable-wise unfolded model.

Finally, the batch dynamic unfolding (Figure 2.10) only uses a subset of mea-

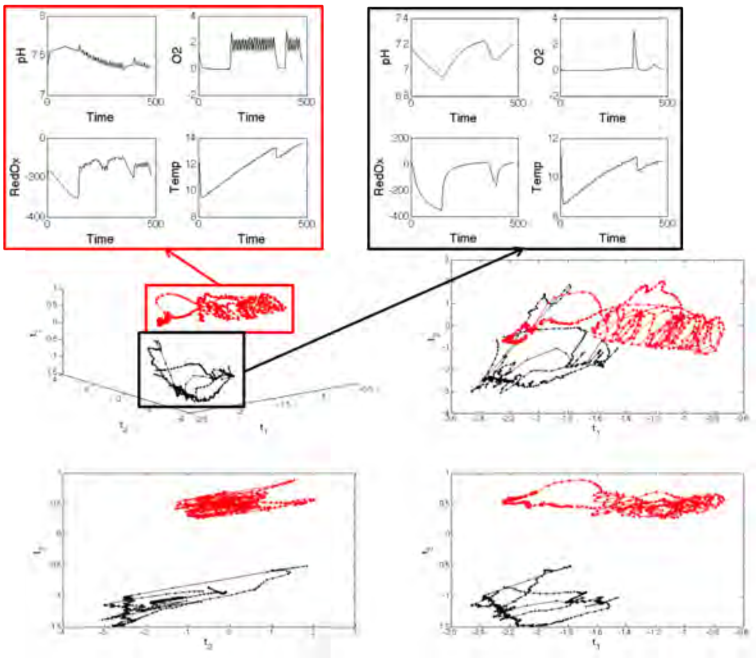


Figure 2.9: Projection of two variable-wise unfolded procedures into the three-dimensional subspace

surement (ℓ, denoted as lagged measured variables (LMV)) to estimate the state of the process. The main idea is that the process dynamic is only contained within the ℓ contiguous samples, which defines a process window for monitoring the process. As a result, the three-dimensional matrix $\underline{X}$ is decomposed in an augmented matrix $X_{i,\ell}\,((\kappa-\ell)\times(\gamma(\ell+1)))$ for each batch process, where κ is the number of samples measured for each batch and γ indicates the number of measured variables for every batch. Finally, this unfolding procedure can be seen as a generalisation of both the variable- and batch-wise unfoldings in the following way:

- If no LVMs are retained ($\ell = 0$), this procedure is equivalent to the variable-wise unfolding.

- If κ - 1 LVMs are retained ($\ell = \kappa - 1$), the resulting augmented matrix is equivalent to the batch-wise unfolded matrix.

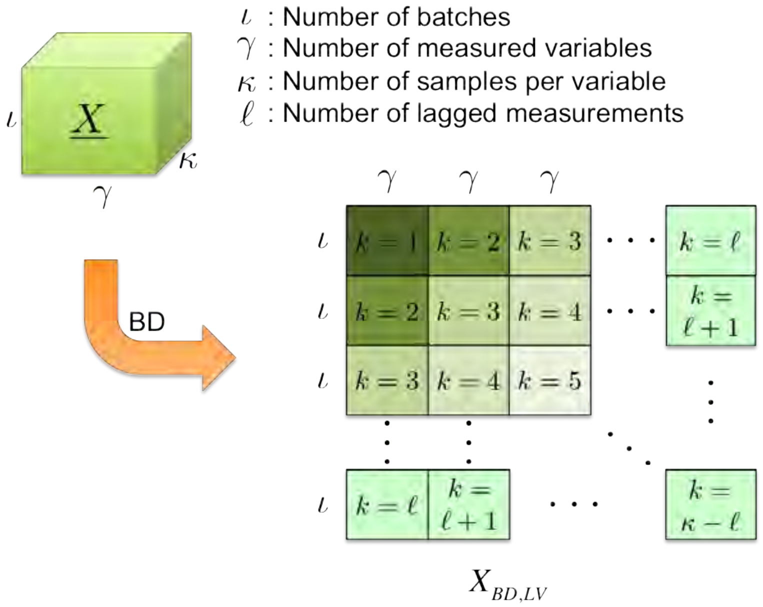

ι : Number of batches
γ : Number of measured variables
κ : Number of samples per variable
ℓ : Number of lagged measurements
$\underline{X}$
ι
κ
γ
BD
γ γ γ
$k=1$ $k=2$ $k=3$ $k=\ell$
$k=2$ $k=3$ $k=4$ $k=\ell+1$
$k=3$ $k=4$ $k=5$
$k=\ell$ $k=\ell+1$ $k=\kappa-\ell$
$X_{BD,LV}$

Figure 2.10: Batch dynamic unfolding of the original 3D matrix using l LVMs

2.2.4 Scaling of the data

The PCA methodology requires the data to be mean-centred, this is, the axis origin of the new projection space will be the mean value. But there are cases where some variables present different value range or deviations. In this case, the data not only has to be centred, but it also has to be scaled. This subsection presents the three main normalisation procedures when dealing with finite duration processes.

2.2.4.1 Continuous scaling (CS)

The continuous scaling procedure assumes that variables in the data matrix share the same distribution. So, it will compute one mean and one standard deviation for each of the original variables (γ) in this way:

$$\mu_j = \frac{\sum_{i=1}^{\iota}\sum_{k=1}^{\kappa} x_{ijk}}{\iota \times \kappa} \qquad \sigma_j = \sqrt{\frac{\sum_{i=1}^{\iota}\sum_{k}^{\kappa}(x_{ijk}-\mu_j)^2}{\iota \times \kappa}} \tag{2.23}$$

This type of normalisation is not usually used because during the process it is not mandatory that variables follow the same distribution, especially if the process can be divided in phases. Another point to note is that this procedure does not remove the mean trajectory of the variables along time (the same drawback of VW), which can lead to a bad performance of the monitoring model.

2.2.4.2 Auto scaling (AS)

AS tries to overcome the previous problem by computing one mean for each variable for all time instants, so the mean trajectory is eliminated. Additionally, AS assumes that the variability changes during the process, and thus, computes one standard deviation at every time instant. This means that AS computes $\gamma \times \kappa$ means and standard deviations, which are computed in this way:

$$\mu_{jk} = \frac{\sum_{i=1}^{\iota} x_i jk}{\iota} \qquad \sigma_{jk} = \sqrt{\frac{\sum_{i=1}^{\iota}(x_{ijk} - \mu_{jk})^2}{\iota}} \tag{2.24}$$

2.2.4.3 Group scaling (GS) or block scaling (BS)

AS gives the same importance to all samples and variables throughout the whole process. However, let us assume that the batch process can be divided into several stages, each having a different duration. Additionally, suppose that one of the divisions lasts two times the durations of the rest stages. Consequently, dominance in this stage implies that small variations in other stages will be masked and the system will only be able to detect large faults occurred in other stages.

In order to solve this, GS/BS gives the same importance (in terms of variance) to all stages (or blocks) the process can be divided into. To do so, GS/BS groups the auto-scaled values (z_{ijk}) into the different stages the process can be divided into, and then rescales each value by the root square of samples within the current stage (k_l) in this way:

$$g_{ijk} = \frac{z_{ijk}}{\sqrt{k_l}} \tag{2.25}$$

This means that although AS and GS/BS have the same mean, they define different projection subspaces since their standard deviations differ.

2.3 Fault detection based on latent structures

Two complementary control charts are usually used for fault detection in multivariate process monitoring using PCA. The purpose is to assess new observations against a PCA model built during normal operating conditions using the T^2 and the square prediction error (SPE) statistics. This section details how they are computed and what information they provide (subsection 2.3.1 for the T^2 index and subsection 2.3.3 for the SPE index), as well as the detectability criteria for both indices (subsection 2.3.2 for the T^2 index and subsection 2.3.4 for the SPE index). Finally, subsection 2.3.5 discusses how false alarms and missed detections can be considered when building the statistical model of a process.

2.3.1 Faults in the projection subspace: T^2

Control charts based on T^2 and the first r principal components can be plotted as follows (MacGregor, 2003):

$$T^2_{\vec{x}} = \sum_{i=1}^{r} \frac{{t_i}^2}{\lambda_i} \tag{2.26}$$

where t_i is the projection of an observation $\vec{x}$ into the i^{th} principal component ($i^t h$ score) computed using (2.18). This expression can be rewritten using the score vector $\vec{t}$, which is a row vector containing the scores for the r retained principal components of $\vec{x}$ in this way:

$$T^2_{\vec{x}} = \vec{t}\Lambda^{-1}\vec{t}^T \tag{2.27}$$

with Λ^{-1} being a diagonal matrix defined as:

$$\Lambda^{-1} = \begin{bmatrix} \frac{1}{\lambda_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\lambda_r} \end{bmatrix} \tag{2.28}$$

and by substituting (2.18) in (2.27) the relation of the original (standardised) observation with the T^2 index can be identified in this way:

$$T^2_{\vec{x}} = \vec{x}P\Lambda^{-1}P^T\vec{x}^T \tag{2.29}$$

$$= \vec{x}D\vec{x}^T \tag{2.30}$$

with $D = P\Lambda{-}1P^T$. If data have been auto-scaled, this is a Mahalanobis distance measure of each observation to the centre of the model. A graph, or control chart, built with such data is useful for detecting variations in the projection space, defined by the r principal components, greater (with respect to the centred data) than common-cause variations, while preserving the structure correlation gathered by the PCA model.

The control limit for the T^2 statistic for a given significance level α (τ_α) can be estimated using the following expression (Anderson, 1984; Johnson and Wichern, 1988):

$$\tau_\alpha = \frac{(n^2-1)r}{n(n-r)}F_\alpha(r, n-r) \tag{2.31}$$

where n is the number of observations/rows of the data matrix X (in this case batches); r is the number of principal components retained; and $F_\alpha(r, n-r)$ is the critical point of the Fisher-Snedecor distribution for r and $n-r$ degrees of freedom. Typical values for the confidence level α are 90%, 95% and 99%, where the closer to 100% the value is, the fewer false alarm rate. So any observation projected on the principal component subspace, $\hat{\vec{x}}$, that surpasses τ_α is labelled as faulty due to the T^2 index. Of course, there is a trade-off between false alarms and missed detections and consequently the selection of such confidence level is a decision to be made.

2.3.2 Fault Detectability in the projection subspace

In order to determine the detectability based on T^2 for a given PCA model, let us define a faulty observation $(\vec{x}_f)$ as an additive simple fault that only affects the i^{th} variable $(\vec{\xi_i})$ with a fault magnitude f_i like previously defined in (2.12):

$$\vec{x}_f = \vec{x}_o + f_i\vec{\xi_i} \tag{2.32}$$

where $\vec{x}_o$ are the fault-free measurement of the observation. As stated in the previous subsection, $T^2_{\vec{x}} > \tau_\alpha$ in order to be detectable due to the T^2 index, so if we substitute the previous equation into (2.30):

$$\left(\vec{x}_o P + f_i \vec{\xi}_i P\right) \Lambda^{-1} \left(P^T \vec{\xi}_i^T f_i + P^T \vec{x}_o^T\right) > \tau_\alpha \tag{2.33}$$

$$\left(\hat{\vec{x}}_o + f_i \hat{\vec{\xi}}_i\right) \Lambda^{-1} \left(\hat{\vec{\xi}}_i^T f_i + \hat{\vec{x}}_o^T\right) > \tau_\alpha \tag{2.34}$$

where $\hat{\vec{\xi}}_i = \vec{\xi}_i P$ and $\hat{\vec{x}}_o = \vec{x}_o P$ are the projections into the PCS of the fault direction and fault-free values respectively. Assuming that the expected value of $\vec{x}_o$ corresponds to the centre of the model ($T^T_{\vec{x}_o} \approx 0$), then detectability on the PCS depends on the following relation:

$$\left\| \vec{\xi}_i P \Lambda^{\frac{-1}{2}} \right\|^2 > \tau_\alpha \tag{2.35}$$

where $\Lambda^{\frac{-1}{2}}$ is a diagonal matrix containing the root square of the eigenvalues associated to the r retained principal components (λ_i) in this way:

$$\Lambda^{\frac{-1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sqrt{\lambda_r}} \end{bmatrix} \tag{2.36}$$

Thus, faults occurring in a direction orthogonal to the projection hyper-plane $\left(\left\|\hat{\vec{\xi}}_i\right\| = 0\right)$ will never be detected with this criterion. This is the case for variables with a low representation in the loading matrix.

2.3.3 Fault in the residual subspace: the SPE index

When a new event in the process, $\vec{x}$, produces a large variation out of the hyperplane described by the r principal components, this means that the data structure has been broken. These types of events are detected in the residual subspace by computing the squared prediction error (SPE) of the residual of each observation according to (Nomikos and MacGregor, 1994):

$$SPE_{\vec{x}} = \sum_{i=1}^{m} (x_i - \hat{x}_i)^2 \tag{2.37}$$

where $\hat{\vec{x}} = (\hat{x}_1, \cdots, \hat{x}_i, \cdots, \hat{x}_m)$ is computed using (2.20). This expression can be also rewritten as a square modulus in this way:

$$SPE_{\vec{x}} = \tilde{\vec{x}}\tilde{\vec{x}}^T = \|\tilde{\vec{x}}\|^2 \tag{2.38}$$

where $\tilde{\vec{x}} = \vec{x}\left(I - PP^T\right) = \vec{x}\tilde{C}$ is the projection into the residual subspace (RS) of the fault.

Assuming that after pre-processing the data (centring, scaling, aligning, etc.), NOC observations can be modelled using a multivariate normal distribution, the SPE will present very small values (typically associated with noise) close to zero, while T^2 presents a larger variation (associated with the variability of the normal operating conditions). Consequently, the SPE will be much more sensitive than T^2 to variations in the process structure, since they alter the correlation structure of the observed data. Furthermore, these changes in the correlation structure are mostly projected in the residual space, and thus, the SPE index is the one that presents the highest increase. The SPE control limit can be computed using the approximation presented in Jackson and Mudholkar (1979) for a given confidence level α (assuming that observations were auto-scaled and the correlation matrix has full rank):

$$\delta_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}\right]^{\frac{1}{h_0}} \tag{2.39}$$

with:

$$\theta_j = \sum_{i=r+1}^{m} \lambda_i^j \quad h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \tag{2.40}$$

where c_α is the value of the normal distribution for a significance level α and m is the number of variables in the original space. Whenever an observation has a value greater than the SPE and/or T^2 control limit, it is labelled as faulty (fault detection).

2.3.4 Fault detectability in the residual subspace

In order to detect a fault in the RS, its projection in this subspace must be greater than δ_α. Using the faulty observation defined in (2.32) and substituting it in (2.38), a detectable fault must hold that:

$$\begin{aligned} \left\|\tilde{\vec{x}}_f\right\|^2 &> \delta_\alpha \\ \left\|\vec{x}_o\tilde{C} + f_i\vec{\xi}_i\tilde{C}\right\|^2 &> \delta_\alpha \end{aligned} \tag{2.41}$$

As stated in subsection 2.3.2, fault-free situations mostly project into the PCS ($\hat{\vec{x}}_o$). Consequently, their projection into the residual subspace ($\tilde{\vec{x}}_o$) is approximately 0, and thus $SPE_{\vec{x}_o} \approx 0$. As a result, the fault appearance is the one responsible for surpassing the SPE limit, which allows us to simplify the previous expression in this way:

$$\|f_i\tilde{\vec{\xi}}_i\|^2 > \delta_\alpha \tag{2.42}$$

Furthermore, from a geometrical point of view, this means that in order to guarantee that a fault will be detected in the residual subspace, its projection in this subspace has to be twofold larger than the interval confidence used to compute the statistical limits, so:

$$\|f_i\tilde{\vec{\xi}}_i\|^2 > 2\delta_\alpha \tag{2.43}$$

Finally, Figure 2.11 presents a graphical representation of the original space and the space decomposition given by PCA, indicating the control limits for $T^2(\tau_\alpha)$ and the $SPE(\delta_\alpha)$ indices and the principal component subspace for all NOC observations (green points within the green ellipsoid delimited by τ_α). Additionally, two observations are presented illustrating a faulty observation (red dot) whose SPE value surpasses δ_α and an unusual (or extreme) observation whose T^2 value surpasses τ_α (orange dot).

2.3.5 False alarms and missed detections when evaluating the model

As stated in section 2.2.1, the statistical model of a process is built up from a series of observations of the process under normal operating conditions (NOC) (denoted as training set). Taking this model as a base, the acceptance limit of all the observations is defined on the basis of the variability observed in the data used to build the model. Subsections 2.3.1 and 2.3.3 presented the corresponding statistical limits for T^2 and the SPE respectively, based on a certain confidence level α. Based on these distributions, the number of false alarms and missed detections found in the training set can be determined. In case that either the number of false alarms or missed detections within

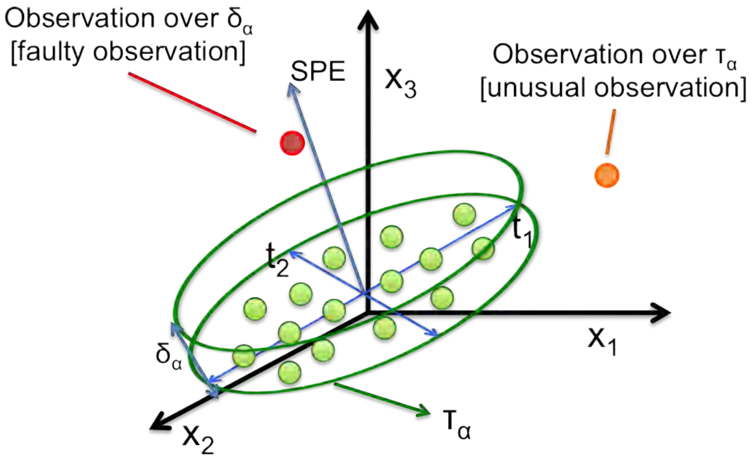


Figure 2.11: Graphical representation of the PCA space decomposition

the training set is significantly different from the expected values, then the statistical limits have to be computed based on the SPE and T^2 values within the training set. Therefore, the first step when monitoring a process is to check the validity of the statistical limits used for fault detection.

Assuming that AOC observations are available, the detection capability of the model can be computed according to the number of missed detections (AOC observations not detected) and false alarms (NOC observations incorrectly labelled as faulty). Taking this into account, the selection of the number of principal components can be defined as a minimisation problem of the number of missed detection and/or false alarms. Since this method is based on the initial set of NOC and AOC observations, whenever the inner distributions of either AOC/NOC set change, the statistical model has to be rebuilt from scratch, since the statistical limits are based on the number of principal component. In order to have more flexible fault detection limits, this thesis proposes the usage of the information related to the nearest neighbours to detect a fault appearance, as will be described in chapter 3.

Now, let us suppose we want to distinguish between two operating conditions within the studied process (C_1 and C_2) and that the correlation structure of both is different. Consequently, when projecting observations from C_2 into C_1, and vice-versa, they will exceed the statistical limits, and thus, identifying the operating conditions of a process can be as simple as checking the statistical limits. Moreover, if both C_1 and C_2 are linearly separable, another viable option is to build a multi-class model with both operating conditions, and then proceed to differentiate between both depending on their location in the projection subspace. This situation is detailed in chapter 4, in which both single-class and multi-class model performances are compared when differentiating between two operating conditions of a wastewater treatment plant.

2.4 Fault diagnosis

Fault isolation on PCA models is usually conducted through the contribution analysis to the statistical indices T^2 and SPE. The idea is to compute the influence of each original variable in the total magnitude of SPE or T^2. Usually a graphical representation is used to help identifying the larger values and consequently the variables that more contribute to the out of control situation. Note that contribution plots only indicate which of the variables are related to the fault, but they do not reveal the actual cause of it (Kourti, 2005). Finally, note that the information contained in the contributions can be used to determine both the fault direction and magnitude affecting an observation, which will be detailed in chapter 3.

2.4.1 Diagnosing in the projection subspace: the T^2 contribution

Since the idea is to relate the influence a faulty observation to the T^2 index, let us relate both the fault detection index and its contribution in this way (Wise et al., 2006):

$$T^2_{\vec{x}} = \vec{c}^{T^2}_{\vec{x}} \left(\vec{c}^{T^2}_{\vec{x}}\right)^T \tag{2.44}$$

$$= \|\vec{c}^{T^2}_{\vec{x}}\|^2 \tag{2.45}$$

This means that the T^2 value is decomposed as the square modulus of the T^2 contribution, and consequently, when relating the contribution to the observation vector $\vec{x}$ we obtain the following expression:

$$\vec{c}^{T^2}_{\vec{x}} = \vec{x}P\Lambda^{\frac{-1}{2}} \tag{2.46}$$

where $\vec{c}_{\vec{x}}^{T^2}$ is composed of the contribution of each variable to the T^2 index, and thus, we can relate the contribution of an observation ($\vec{c}_{\vec{x}}^{T^2}$) to the individual contribution of each variable in the following way:

$$\vec{c}_{\vec{x}}^{T^2} = \begin{bmatrix} c_1^{T^2} & c_2^{T^2} & \cdots & c_m^{T^2} \end{bmatrix} \tag{2.47}$$

where $c_i^{T^2}$ is the contribution to the T^2 index of the i^{th} variable, which can be computed in this way:

$$c_i^{T^2} = \left(\vec{x} P \Lambda^{\frac{-1}{2}} \vec{v}_i^T \right)^2 \tag{2.48}$$

where $\vec{v}_i$ is a row vector with as many elements as measured variables that is used to indicate the variable whose contribution has to be computed. This is done by setting to 1 the position of the observation vector associated with the variable to compute its contribution, and the rest are set to 0. An example of the variable vector for the first variable ($\vec{v}_1$) of a process with three measured variables would be:

$$\vec{v}_1 = [1 \quad 0 \quad 0] \tag{2.49}$$

2.4.2 Isolability in the projection subspace

Let us assume that the we want to correctly diagnose the simple and additive fault in subsection 2.3.2. This means that only variable $\vec{v}_i$ is faulty, and thus, its contribution ($c_i^{T^2}$) has to be the highest among all other measured variables ($c_j^{T^2}$). Therefore, the following condition must hold (assuming the fault is detectable):

$$c_i^{T^2} \geq c_j^{T^2} \tag{2.50}$$

When substituting (2.32) and (2.48) in the previous expression, the isolability condition can be rewritten in this way:

$$\left[\left(\vec{x}_o + f_i \vec{\xi}_i \right) P \Lambda^{\frac{-1}{2}} \vec{v}_i^T \right]^2 \geq \left[\left(\vec{x}_o + f_i \vec{\xi}_i \right) P \Lambda^{\frac{-1}{2}} \vec{v}_j^T \right]^2 \tag{2.51}$$

And assuming that the fault-free values of the observation ($\vec{x}_o$) are close to the centre, then $f_i\vec{\xi_i}$ is the one responsible for surpassing the T^2 limit. Consequently, the previous expression can be simplified in this way:

$$\left(f_i\vec{\xi_i}P\Lambda^{\frac{-1}{2}}\vec{v}_i^T\right)^2 \geq \left(f_i\vec{\xi_i}P\Lambda^{\frac{-1}{2}}\vec{v}_j^T\right)^2 \tag{2.52}$$

and since $\vec{\xi_i}$ is a simple fault, then the expression is left as:

$$\left[P\Lambda^{\frac{-1}{2}}\right]_{ii}^2 f_i^2 \geq \left[P\Lambda^{\frac{-1}{2}}\right]_{ij}^2 f_i^2 \tag{2.53}$$

where the subindex ii indicates the i^{th} element in the diagonal from $P\Lambda^{\frac{-1}{2}}$, and ij refers to the element in the (i^{th}, j^{th}) position in this matrix. This means that any simple fault is diagnosable if:

$$\left[P\Lambda^{\frac{-1}{2}}\right]_{ii}^2 \geq \left[P\Lambda^{\frac{-1}{2}}\right]_{ij}^2 \tag{2.54}$$

This conditions imposes that the maximal value in the product $P\Lambda^{\frac{-1}{2}}$ is found on the diagonal, otherwise, simple faults cannot be detected. Since this product can be computed while building the statistical model, one can select the principal components accordingly such that this condition holds. However, this only grants the isolability of simple faults. Multiple faults add more restrictions to the values of the product $P\Lambda^{\frac{-1}{2}}$, since they affect more than one variable. As a result, the diagnosability of multiple faults reduces (we impose more restrictions for each new variable) the more variables they alter. In order to clarify this point, let us define a multiple fault affecting two different variables ($\vec{v}_i$ and $\vec{v}_j$) in this way:

$$\vec{x}_f = \vec{x}_o + \vec{\xi_i}f_i + \vec{\xi_j}f_j \tag{2.55}$$

This means, that when exploring a variable not associated with the fault, $\vec{v}_k$, and assuming that the fault-free observation originally laid in the centre of the principal component subspace, (2.52) must hold the following conditions when exploring both variables $\vec{v}_i$ and $\vec{v}_j$ respectively:

$$\begin{aligned}\left[P\Lambda^{\frac{-1}{2}}\right]_{ik}^2 + \left[P\Lambda^{\frac{-1}{2}}\right]_{jk}^2 + 2\left[P\Lambda^{\frac{-1}{2}}\right]_{ik}\left[P\Lambda^{\frac{-1}{2}}\right]_{jk} \\ \leq \left[P\Lambda^{\frac{-1}{2}}\right]_{ii}^2 + \left[P\Lambda^{\frac{-1}{2}}\right]_{ij}^2 + 2\left[P\Lambda^{\frac{-1}{2}}\right]_{ii}\left[P\Lambda^{\frac{-1}{2}}\right]_{ij}\end{aligned} \tag{2.56}$$

$$\begin{aligned}\left[P\Lambda^{\frac{-1}{2}}\right]_{ik}^{2}+\left[P\Lambda^{\frac{-1}{2}}\right]_{jk}^{2}+2\left[P\Lambda^{\frac{-1}{2}}\right]_{ik}\left[P\Lambda^{\frac{-1}{2}}\right]_{jk}\\ \leq\left[P\Lambda^{\frac{-1}{2}}\right]_{ij}^{2}+\left[P\Lambda^{\frac{-1}{2}}\right]_{jj}^{2}+2\left[P\Lambda^{\frac{-1}{2}}\right]_{ij}\left[P\Lambda^{\frac{-1}{2}}\right]_{jj}\end{aligned} \tag{2.57}$$

Additionally, and as was stated in section 2.1, multiple faults can happen due to the simultaneous fault of two independent elements, or due to a fault affecting an element with physical redundancy of measurements or due to the fault that affecting one of the latent structures that drive the process. In case of the simple faults affecting independent elements/variables of the process, the correlation between both variables in the principal component subspace, $\left[P\Lambda^{\frac{-1}{2}}\right]_{ij}$, is close to zero $\left(\left[P\Lambda^{\frac{-1}{2}}\right]_{ij}\approx 0\right)$, which means that the previous expressions can be simplified in this way:

$$\left[P\Lambda^{\frac{-1}{2}}\right]_{ik}^{2}+\left[P\Lambda^{\frac{-1}{2}}\right]_{jk}^{2}+2\left[P\Lambda^{\frac{-1}{2}}\right]_{ik}\left[P\Lambda^{\frac{-1}{2}}\right]_{jk}\leq\left[P\Lambda^{\frac{-1}{2}}\right]_{ii}^{2} \tag{2.58}$$

$$\left[P\Lambda^{\frac{-1}{2}}\right]_{ik}^{2}+\left[P\Lambda^{\frac{-1}{2}}\right]_{jk}^{2}+2\left[P\Lambda^{\frac{-1}{2}}\right]_{ik}\left[P\Lambda^{\frac{-1}{2}}\right]_{jk}\leq\left[P\Lambda^{\frac{-1}{2}}\right]_{jj}^{2} \tag{2.59}$$

Taking this into consideration, we can say that in order to correctly diagnose the appearance of a multiple fault, we impose that the values within the diagonal of the projection matrix into the principal component subspace greatly differ from the values outside the diagonal. This situation is usually given for processes with blocks of variables highly correlated among them, and nearly non-correlated with the rest of variables.

Although the problem with the diagnosability of multiple faults was exemplified with a two-variable fault, one can imply that the more variables implied in a multiple fault, the more easy it becomes to confuse this multiple fault with a simple fault affecting a variable not implied in the fault. This happens because for each new variable, we are imposing a higher differentiation between the elements in the diagonal with respect to the elements outside of it.

2.4.3 Diagnosing in the residual subspace: the SPE contribution

Likewise T^2, let us define the SPE as the square modulus of the SPE contributions in this way:

$$SPE = \left\|\vec{c}^{SPE}\right\|^2 \tag{2.60}$$

relating this expression to (2.37), then the contribution to the SPE index is defined as:

$$\vec{c}^{SPE} = \tilde{\vec{x}} = \vec{x}\tilde{C} \tag{2.61}$$

with $\tilde{C} = I - PP^T$ and where I is the identity matrix with as many rows and columns as measured variables. Therefore, the SPE contribution of $\vec{v}_i$ can be defined as:

$$c_i^{SPE} = \left(\vec{x}\tilde{C}\vec{v}_i^T\right)^2 \tag{2.62}$$

2.4.4 Isolability in the residual subspace

In order to identify that $\vec{v}_i$ is the responsible for the simple fault described in (2.32) the following expression must be true:

$$c_i^{SPE} \geq c_j^{SPE} \tag{2.63}$$

for every variable $j \neq i$. By substituting (2.62) and (2.32) in the previous expression it becomes:

$$\left[\left(\vec{x}_o + f_i\vec{\xi}_i\right)\tilde{C}\vec{v}_i\right]^2 \geq \left[\left(\vec{x}_o + f_i\vec{\xi}_i\right)\tilde{C}\vec{v}_j\right]^2 \tag{2.64}$$

And assuming that fault-free observations mostly project into the projection subspace, and thus, $\vec{x}_o\tilde{C} \approx 0$, we can simplify the previous expression to:

$$\left(f_i\vec{\xi}_i\tilde{C}\vec{v}_i\right)^2 \geq \left(f_i\vec{\xi}_i\tilde{C}\vec{v}_j\right)^2 \tag{2.65}$$

Assuming that $\vec{\xi}_i$ is a simple fault, the isolability of the fault is reduced to:

$$\tilde{c}_{ii}^2 \geq \tilde{c}_{ij}^2 \tag{2.66}$$

Like it happened with the diagnosability of faults in the projections subspace (based on the T^2 index), this condition imposes that the maximal value in $\tilde{C}$ has to be located in the diagonal. Consequently, and since $\tilde{C}$ is known when building the statistical model, the number of principal components to retain can be fixed such that simple faults affecting each measuring variable is diagnosable. However, this does not guarantee that complex faults will be diagnosable, since they alter more than one variable, and thus, they add more restrictions to the previous one. As a result, the more variables a multiple fault affects, the lower the probability to correctly diagnose it is. Let us clarify this point with the same example presented in section 2.4.2. This is, a multiple fault that affects two variables, $\vec{v}_i$ and $\vec{v}_j$ like the one described in (2.55):

$$\vec{x}_f = \vec{x}_o + \vec{\xi}_i f_i + \vec{\xi}_j f_j \tag{2.67}$$

In order to correctly identify that only variables $\vec{v}_i$ and $\vec{v}_j$ are responsible for the faulty situation, the following conditions must hold when exploring a variable $\vec{v}_k$ not related with the multiple fault for variables $\vec{v}_i$ and $\vec{v}_j$ respectively:

$$\tilde{c}_{ik}^2 + \tilde{c}_{jk}^2 + 2\tilde{c}_{ik}\tilde{c}_{jk} \leq \tilde{c}_{ii}^2 + \tilde{c}_{ij}^2 + 2\tilde{c}_{ii}\tilde{c}_{ij} \tag{2.68}$$

$$\tilde{c}_{ik}^2 + \tilde{c}_{jk}^2 + 2\tilde{c}_{ik}\tilde{c}_{jk} \leq \tilde{c}_{ij}^2 + \tilde{c}_{jj}^2 + 2\tilde{c}_{ij}\tilde{c}_{jj} \tag{2.69}$$

And in case that the multiple fault is the result of several (in this case two) simultaneous simple faults, the correlation between both variables, $\tilde{c}_{ij}$ is close to zero ($\tilde{c}_{ij} \approx 0$), which allows us to simplify the previous expressions in this way:

$$\tilde{c}_{ik}^2 + \tilde{c}_{jk}^2 + 2\tilde{c}_{ik}\tilde{c}_{jk} \leq \tilde{c}_{ii}^2 \tag{2.70}$$

$$\tilde{c}_{ik}^2 + \tilde{c}_{jk}^2 + 2\tilde{c}_{ik}\tilde{c}_{jk} \leq \tilde{c}_{jj}^2 \tag{2.71}$$

This means that in order to correctly diagnose the appearance of multiple faults, the values within the diagonal matrix must sufficiently differ with respect to the values outside of it (like it happened with the diagnosability in the principal component subspace explained in subsection 2.4.2). Additionally, the more faults implied in the multiple fault, the more easily the multiple fault will be misclassified as a fault affecting a variable not related to the fault appearance (we impose that the values outside of the diagonal are closer to zero). As was also mentioned in subsection 2.4.2, processes whose variables can be divided in groups with high intra-group correlation (the correlation between variables in the same group is close to 1 or -1) and with low inter-group correlations (the correlation between variables of different groups is close to 0).

2.5 Model-free fault diagnosis based on finite duration responses

As was introduced in chapter 1, batch-monitoring techniques such as MPCA (described in the previous section) and multiway partial least squares (MPLS, described in Appendix A) can be applied to finite-duration responses, such as the information gathered from sensors installed into a commercial wing depicted in Mujica et al. (2009) and Ruiz et al. (2013) or voltage sags (a reduction of the nominal voltage of one phase between 10% to 90% and with a duration time from 200 ms to 1 minute (Bollen, 2000)) as described in Meléndez et al. (2008b), Meléndez et al. (2008a) and Barrera et al. (2008) provided the following assumptions:

1. The system when under normal operating conditions does not present a sufficient variation to build a statistical model with these signals. However, events that excite the process (like impacts in a flap wing or voltage sags in the power network), with a perturbation that attenuates its effects along time.

2. Faults affecting these systems are short-term variations of the process (they are not continuous). Consequently, given a sufficiently large process window, when faults cease, their effect disappears, and thus the process returns to the NOC region.

3. This short-time perturbations can be gathered (from the first sample they are detected until the effect disappears) and grouped into a three-dimensional matrix similar to the one used to represent batch processes.

4. The interest of the analysis is to characterise and differentiate the different "faults", and thus, statistical models study variations between the different fault types that affect the process.

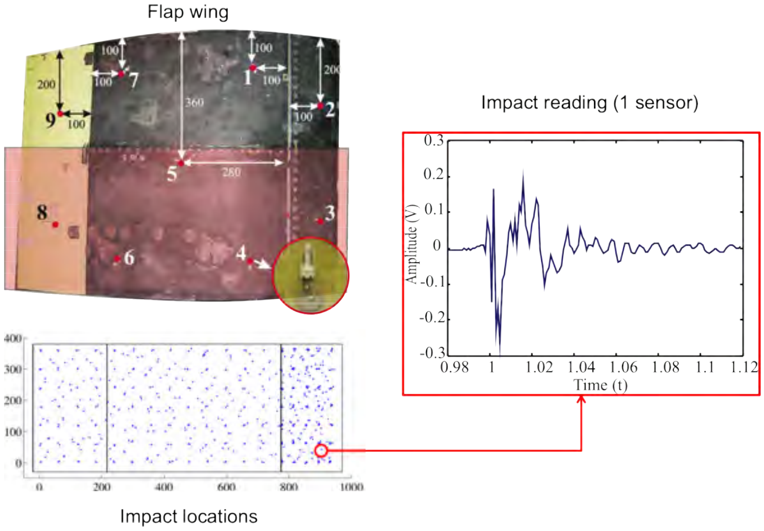


Figure 2.12: Example of an impact reading (for a single sensor) in the commercial flap wing studied in Mujica et al. (2009) and Ruiz et al. (2013)

5. The effect of the fault must not permanently change the relationships between variables, since the initial model is no longer valid, and thus, a new model has to be built with the new relationships.

In order to study these variations using either MPCA or MPLS, one builds a process window to gather the information of the faults that appear within the process. Given that the effect of the fault attenuates through time, the process will return to its normal operating conditions, giving us a maximum number of samples that have to be gathered. Additionally, one can sample NOC values of the process (previous and posterior to the fault appearance) in order to compute the fault magnitude and direction of the fault that appeared. An example of readings of the commercial flap wing and voltage sags gathered in a HV/MV substation are presented in Figure 2.12 and Figure 2.13 respectively.

Regarding Figure 2.12, the impact initially produces an important variation from the normal readings of the sensors. However, this variation reduces after a short period of time, and thus, the sensor readings indicate that the flap wing is within the normal operating conditions previous to the impact. This is the same situation with voltage sags, as depicted in Figure 2.13, where the sag produces an important variation of the

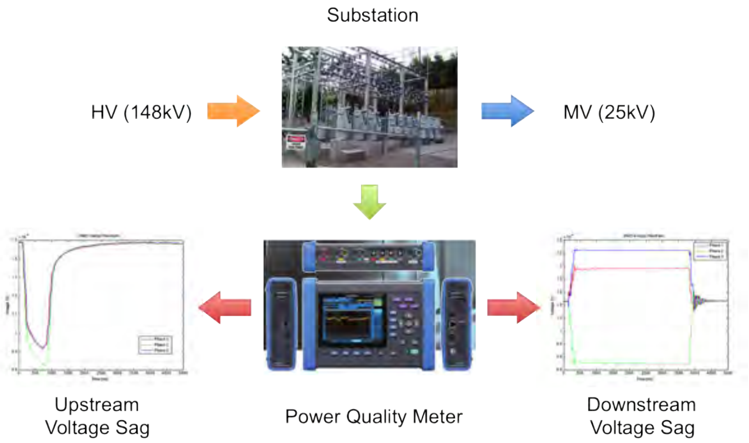


Figure 2.13: Example of upstream and downstream voltage sags gathered in an HV/MV substation studied in Meléndez et al. (2008b), Meléndez et al. (2008a) and Barrera et al. (2008)

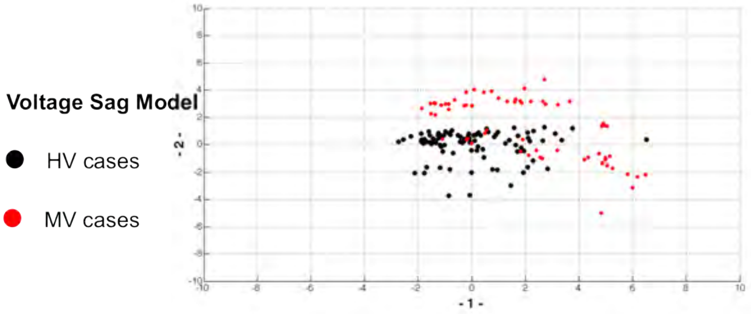


Figure 2.14: Projection of upstream and downstream voltage sags into the two first principal components of the statistical model built solely with upstream voltage sags

voltage value, and whenever the sag dissipates its effect dilutes until the point that the change is no longer visible. This means that the final objective of the statistical model of this kind of processes is the diagnosis of the fault altering the signal instead of modelling the normal operating conditions of the process. The statistical model associated built using only upstream voltage sags (HV sags) is depicted in Figure 2.14, which shows the projection into the two-principal component subspace of both upstream (dark dots) and downstream (red dots).

3

Case-based diagnosis (CBD) based on latent structures (LS)

The use of statistical methods for fault diagnosis is reduced to identify the variables that better explain an out-of-control situation (fault detection). With the aim of improving the diagnosis capabilities by including additional information in the description of faults and, at the same time, reusing occurred faulty experiences; a formalisation of case based reasoning strategies in the principal component space is proposed in this chapter. Previous observations (faulty or not), conveniently diagnosed, projected in the principal component space (defined by NOC observations) is proposed to identify regions assignable to specific misbehaviours or faults.

The chapter is organised as follows. Firstly, the basic methodology of the CBR are explained (section 3.1). Next, cases and case base definition are exposed (section 3.2), followed by the definition of the distance criteria and neighbourhoods based on LS (section 3.3). Next, advantages and drawbacks of combining both approaches will be commented (section 3.7). Finally, this chapter is closed with a discussion of the requirements, contributions and considerations of applying CBD for the monitoring and diagnosis of batch processes when determining the diagnosis scheme (off-line modelling) and when exploiting the resulting model within the production stage of batch processes (real-time release) in section 3.8.

3.1 Case-based reasoning (CBR)

Case-based reasoning (CBR) is a reasoning approach to problem solving capable of using the knowledge acquired by previous experiences (de Mántaras and Plaza, 1997). It has demonstrated to be a good option for solving problems in several domains (diagnosis, prediction, control, planning, etc.) (Aamodt and Plaza, 1994). The basic functions that all CBR present are known as the 4-Rs (Aamodt and Plaza, 1994), and can be organised in a cycle as depicted in Figure 3.1 consisting of the following steps:

1. RETRIEVE the most similar cases of the new case.
2. REUSE the information in these cases to solve the new problem.
3. REVISE the proposed solution.
4. RETAIN the new information of the new experience in order to solve new similar problems.

This is, to solve a new problem, the most similar cases are retrieved from the experiences previously stored. The information contained in these retrieved cases is then reused to propose a possible solution. Once the solution is evaluated, the case is retained, if necessary, for further classifications.

Like in many machine learning algorithms, the independence of attributes involved in the retrieval of cases is usually assumed, i.e. when the Euclidean distance is used to define the neighbourhood of a new observation. As exposed in Jakulin and Bratko (2003), attribute independence also lets a classifier to collect the evidence for a class from individual attributes separately. So, the contribution of an attribute to a class can be determined independently from the other attributes. This requirement, not only simplifies the learning algorithms but it also results in a robust performance and simpler models. Given that LS information is uncorrelated (scores are orthogonal among them, and T^2 and SPE also are orthogonal between them and when combined they describe the whole process space), this partially fulfils this independence, in a sense that adding an additional attribute adds new information.

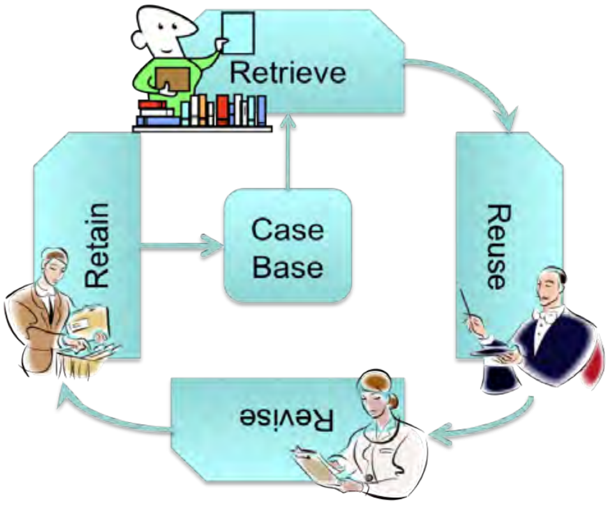


Figure 3.1: The CBR cycle

3.2 Case and case base definition

The basis of case-based reasoning is the case definition. A case (c) is the minimum representation of a past experience and its solution Leake (1996). When several cases are available, they can be grouped in a case base.

Since the final objective of the CBR is the diagnosis of the batch process, a basic structure containing the information associated to each batch observation (Obs) and the information related to the batch diagnosis ($Diag$) is proposed:

$$c = \{Obs, Diag\} \tag{3.1}$$

Regarding the observation structure Obs, it can be divided into two basic subdivision: the batch measurements ($\vec{x}_{BWv}$) and the information associated to their projection into the MPCA model ($Projection$):

$$Obs = \{\vec{x}_{BWv}, Projection\} \tag{3.2}$$

Note that we propose the usage of the batch-wise unfolded observation for the following reasons:

- Qualitative variables from batch processes are usually collected at the end of the batch (when releasing the contents from the batch vessel), and thus, these qualitative variables and batch evaluations are related to the whole process.
- As stated in subsection 2.2.3, auto-scaling batch-wise unfolded batches removes the average trajectory of variables. As a result, the statistical model resulting from these observations takes into account the variation within the different NOC batches used to build the statistical model.
- As indicated in subsection 2.2.3, batch-wise unfolded batches are represented as single points in the projection and residual subspaces, which facilitates the comparison between batches and their clustering.

And between both batch-wise unfold procedures, we propose the usage of the variable-grouped matrix (BWv) since is more comprehensible for process operators. Note that in case of applying this methodology to still running batches, this only requires a change in the sorting order of the unfolded matrix as stated in subsection 2.2.3.

The *Projection* structure is composed of the information related to the projection into the batch-wise unfolded MPCA model of the process as its name indicates. Therefore, the information contained in this structure is the score vector of the observation ($\vec{t}$) and the T^2 ($T^2_{\vec{x}}$) and SPE ($SPE_{\vec{x}}$) indices:

$$Projection = \{\vec{t}, T^2_{\vec{x}}, SPE_{\vec{x}}\} \tag{3.3}$$

Finally, regarding the diagnosis structure ($Diag$), it contains the fault direction ($\vec{\xi_i}$) and magnitude (f_i), and the $Cause$ that originated the fault (when applicable) obtained from an accurate analysis of the batch and the process conditions:

$$Diag = \{\vec{\xi_i}, f_i, Cause\} \tag{3.4}$$

Therefore, the information necessary for each case in the case base is the following:

$$c = \{\vec{x}, \vec{t}, T^2_{\vec{x}}, SPE_{\vec{x}}, \vec{\xi_i}, f_i, Cause\} \tag{3.5}$$

3.3 Similarity and neighbourhood based on latent structures: the retrieve function

According to the CBR methodology, case retrieval is based on a nearest neighbours criterion. Consequently, neighbourhoods based on distance or similarity criteria have to be defined. This section defines and interprets several similarity criteria using the information of LS for monitoring and diagnosis purposes (subsection 3.3.2); and then, in subsection 3.3.3 they are combined to identify appropriate neighbourhoods for diagnosis purposes. Both distance criteria and neighbourhoods were published in Berjaga et al. (2009c).

3.3.1 Retrieval and distances: finding the most similar situations

The first step of the CBR is to find the most similar situation(s) to a new problem in order to provide a plausible solution. This means that the retrieve step has to compare the new problem with each case in the case base in order to find this/these situation/s. In order to do so, the retrieve step of the CBR consists in computing the similarity of cases by comparing the different attributes (information that characterises a case) of cases in the case base and the new problem using a distance criterion. The lower the distance between a stored case and the new problem is, the more similar two observations are. Contrarily, the greater the distance is, the more different the stored case and the new case are.

Once the distance between the new case and cases in the case base is known, then the pruning procedure to finding the most similar situation(s) starts. There are two main approach for determining the set of nearest cases to the new one:

- Fix a distance/similarity threshold (θ). This approach keeps all stored cases (c_i) that satisfy that their distance to the new case (c_{new}) is lower or equal to this maximal distance (or minimum similarity). Therefore, the subset of nearest cases to the new case ($NC(c_{new})$) is defined according to this expression:

$$NC(c_{new}) = \{c_i / d\left(c_{new}, c_i\right) \leq \theta\} \tag{3.6}$$

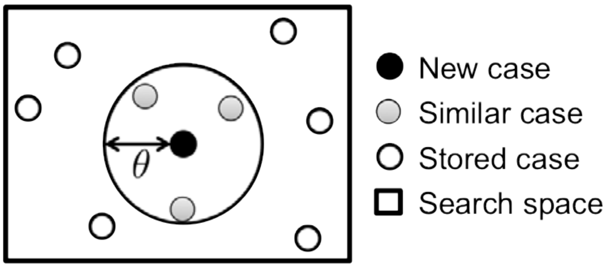


Figure 3.2: Retrieve step limiting the distance between the new case to the stored cases with θ

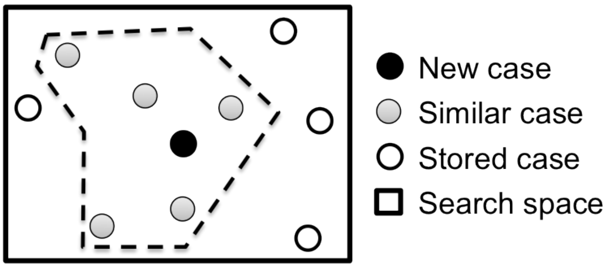


Figure 3.3: Retrieve step forcing the number of similar cases to $k = 5$

Or from a geometrical point of view, limit the radius of the distance circle around of the new observation as depicted in Figure 3.2.

Note that this strategy does not guarantee the retrieval of a similar instance for those situations that the new case is far from the set of stored instances.

- Retrieve a fixed k number of neighbours. This strategy sorts stored cases in the case base according to their distance to the new observation in ascending order (from lowest to largest), and then only keeps the k nearest. Because of this, this strategy is usually known as the k-nearest neighbour (k-NN) approach. Regarding the previous approach, we can say that the maximum distance between the new case (c_{new}) and the furthest case (c_k) is fixed such that:

$$\theta = d(c_{new}, c_k) \tag{3.7}$$

And from a geometrical point of view, this means that the retrieve space is altered such that the number of retrieved cases is equal to k, as depicted in Figure 3.3 (with $k = 5$).

> Contrary to the distance-limited approach, k-NN guarantees that k cases will be retrieved. However, note that the larger the value of retrieved cases k, the greater the distance between new case and the furthest case ($d(c_{new}, c_k)$), and thus, this can eventually cause the retrieval of completely different instances (with maximal distance and minimum similarity).

This thesis uses the k-NN approach for retrieving observations. Consequently, a series of mechanisms to deal with not too similar retrieved cases (due to large k values) have been developed and presented later on in section 3.4. This also means that the different representations shown in the next section fix the distance threshold θ according to (3.7).

3.3.2 Distance criteria and latent structures

Four basic similarity criteria are proposed. The first one is simply an Euclidean distance in the principal component subspace (PCS), whereas the second and third consider the distance to the model in terms of SPE and T^2 respectively. Finally, the fourth method takes into account both SPE and T^2 indices at the same time based on the combined index described in Yoon and MacGregor (2001).

3.3.2.1 Distance between observations in the principal component space

Taking advantage that the application of PCA results in new r uncorrelated components, the space defined by the scores will be appropriate to compute an Euclidean distance between observations projected into it:

$$d_t(c_a, c_b) = \sqrt{\sum_{i=1}^{r} (t_{c_a,i} - t_{c_b,i})^2} \tag{3.8}$$

Nevertheless, remember that principal components are ordered according to the variance captured in each direction (λ_i). Consequently, it is better to weight each score according to its variance. This is known as Mahalanobis distance, which results in the following expression:

$$d_t(c_a, c_b) = \sqrt{\sum_{i=1}^{r} \frac{(t_{c_a,i} - t_{c_b,i})^2}{\lambda_i}} \tag{3.9}$$

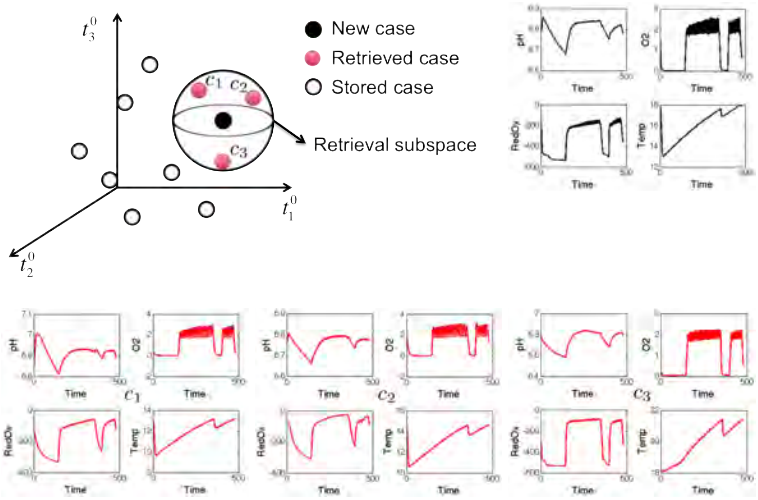


Figure 3.4: Similarity based on the Mahalanobis distance within a three-dimensional PCS (d_t)

Where r stands for the number of retained principal components, $t_{c_a,i}$ is the i^{th} score of a case c_a, for example a new observation, and $t_{c_b,i}$ could represent the same for an observation in the case base. A geometrical interpretation of this distance is shown in Figure 3.4, where t_i^0 refers to the normalised scores (t_i/λ_i).

As stated in 2.3.1, the PCS depicts the normal variation of the process, consequently, two NOC observations close in the PCS present the same traits, as depicted in Figure 3.5a). However, since this distance criterion does not take into account the SPE index, faults mostly projected into the residual subspace, and thus with $\left\|\hat{\vec{\xi}}_i\right\| \approx 0$, can be retrieved as nearest neighbours of NOC observations as depicted in Figure 3.5b).

Regarding faults mostly projected in the projection subspace $(\left\|\tilde{\vec{\xi}}_i \approx 0\right\|)$, nearest neighbours based can be divided into: 1) batches with similar fault-free values $(\vec{x}_o)$, and thus, the distance between cases is due to a small variations in the projection of the fault direction into the PCS (case c_b in Figure 3.6) or due to differences in the fault magnitude (case c_a in Figure 3.6); or 2) different batches, but due to the effect of the fault, they lay close within the PCS (case c_c in Figure 3.6).

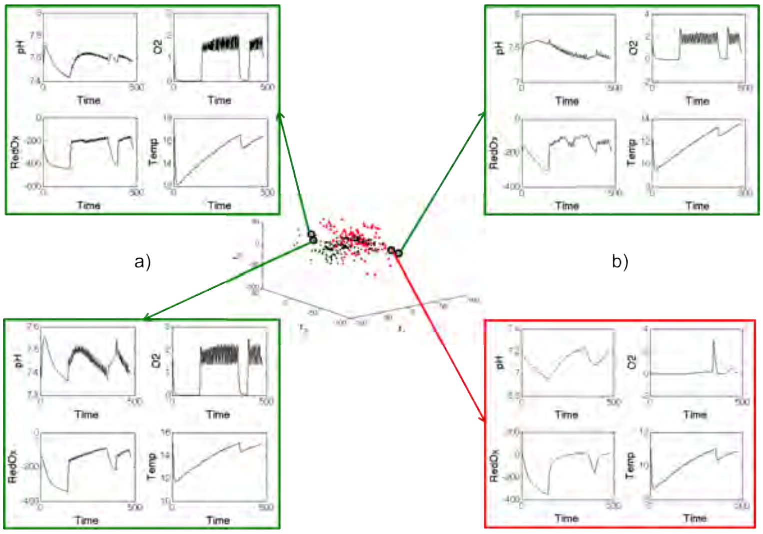


Figure 3.5: Examples of retrieval processes based on d_t in which two NOC observations present similar characteristics (a) and a faulty observation is retrieved as the nearest neighbour of a NOC observation (b)

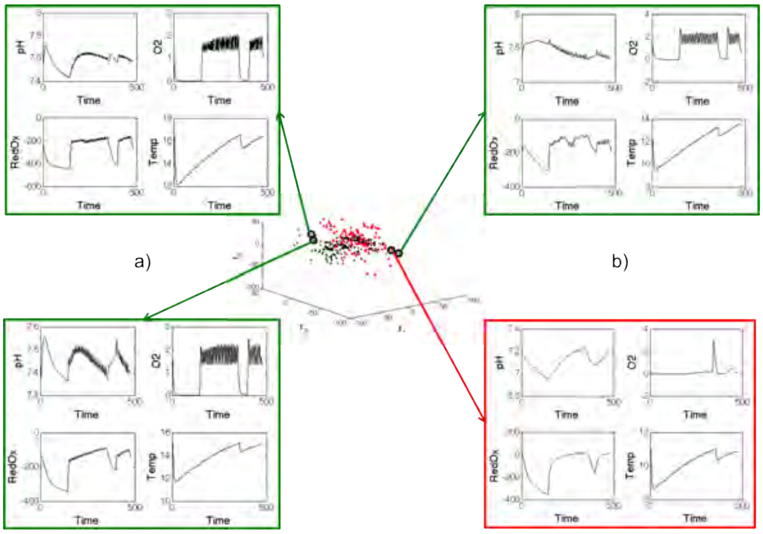


Figure 3.6: Example of relations between nearest neighbours based on d_t in terms of fault direction and magnitude: c_a is a case with the same fault free-values of c_{new} and the distance is due to the differences in fault magnitude; c_b is a batch with small differences in the projection of the fault direction; and c_c is a batch with completely different fault free-values, but due to the pair $\hat{f}_c\hat{\vec{\xi}}_c$ is a neighbour of c_{new}

3.3.2.2 SPE Similarity

As exposed in section 2.3, the SPE index is related to the projection error. Consequently, observations with a low value of SPE are consistent with the projection model (obtained with observations gathered during normal operation conditions) and they are close to the hyperplane defined by the r retained principal components. On the other hand, observations with a large SPE are expected to be inconsistent with the model structure and consequently they are candidates for being faulty situations.

Therefore, observations with a similar SPE can be used to identify similar operating conditions (normal or abnormal). A simple (absolute) difference can be used to compute this similarity:

$$d_{SPE}(c_a, c_b) = |SPE_{c_a} - SPE_{c_b}| \tag{3.10}$$

Since this distance uses the SPE values, differences in the fault directions are omitted (the distance imposes that the fault direction of cases is the same as the SPE an orthonormal vector to the PCS). As a result, differences only take into account variations in the fault magnitude seen in the residual subspace ($\tilde{f}_i$). A possible geometrical interpretation of this distance is showed in Figure 3.7.

As can be seen in Figure 3.7, similar values of SPE can be find far from the new case when looking into the PCS. The main reason is that d_{SPE} only takes into account the projection into the residual subspace, which fixes the same fault direction to all cases. Based on this, we can rewrite d_{SPE} in this way:

$$\begin{aligned} d_{SPE}\left(c_a, c_b\right) &= \left| f_a^2 \left\| \tilde{\vec{\xi}}_a \right\|^2 - f_b^2 \left\| \tilde{\vec{\xi}}_b \right\|^2 \right| & (3.11) \\ &= \left| \tilde{f}_a^2 - \tilde{f}_b^2 \right| & (3.12) \end{aligned}$$

with $\tilde{f}_{new} = f_{new} \left\| \tilde{\vec{\xi}}_{new} \right\|$ and $\tilde{f}_i = f_i \left\| \tilde{\vec{\xi}}_i \right\|$.

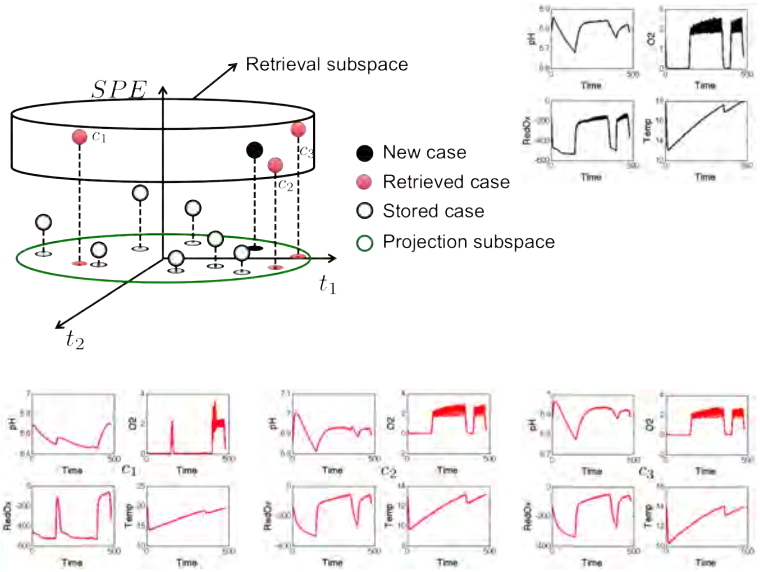


Figure 3.7: Similarity based on the SPE statistic distance (d_{SPE})

3.3.2.3 T^2 Similarity

In section 2.3 the statistic T^2 was presented as a measure of the (Mahalanobis) distance of an observation to the centroid of the model. In fact, it is a square distance and represents the dispersion from the mean of the model since the scores are normalised (unit variance) prior to compute the T^2 index.

Low values of T^2 represent observations close to the mean whereas high values of T^2, over the control limits, are evidences of an abnormal behaviour; although it does not necessary imply that the correlation structure has been broken (this will depend on the SPE).

Similarity between two cases, c_a and c_b, according to the statistic T^2 can be computed as follows:

$$d_{T^2}(c_a, c_b) = \left|T^2_{c_a} - T^2_{c_b}\right| \tag{3.13}$$

Like it happened with the SPE, d_{T^2} fixes the same fault direction to all cases, consequently, the distance between cases depends solely on the difference between the projection into the PCS of the fault magnitude ($\hat{f}_i$). A geometric interpretation of this

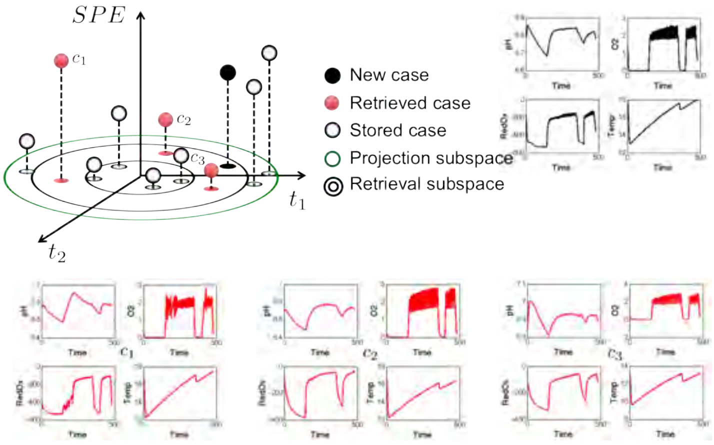


Figure 3.8: Retrieval space using the T^2 statistic (d_{T^2})

distance criterion is depicted in Figure 3.8.

As can be seen in Figure 3.8, observations with a low d_{T^2} may be far away from one another, like is the case of c_1. The main reason is that d_{T^2} fixes the same fault direction to all cases, and as a result, only takes into account the projection of the fault magnitude into the PCS. Additionally, note that this distance does not take into account the SPE of the observations, and thus, nearest neighbours can present completely different behaviours since divergences between fault directions is not taken into account. As a result, d_{T^2} can be rewritten in this way:

$$d_{T^2}\left(c_a, c_b\right) = \left| f_a^2 \left\| \hat{\vec{\xi}}_a \right\|^2 - f_b^2 \left\| \hat{\vec{\xi}}_b \right\|^2 \right| \tag{3.14}$$

$$= \left| \hat{f}_a^2 - \hat{f}_b^2 \right| \tag{3.15}$$

with $\hat{f}_a = f_a \left\| \hat{\vec{\xi}}_a \right\|$ and $\hat{f}_b = f_b \left\| \hat{\vec{\xi}}_b \right\|$.

3.3.2.4 Distance in the $SPE - T^2$ space

The main problem when using either d_{SPE} or d_{T^2} is that they only consider the information contained in the residual subspace and the principal component subspace respectively. As a result, observations with completely different natures, like c_3 in Figure 3.8, which is an observation with a low SPE projection is retrieved as the nearest neighbour of the new case, which presents a large SPE value. In order to avoid these situations, and considering that both the SPE and T^2 indices are complementary (the residual and principal component subspace are orthogonal), we can combine the information of both indices to retrieve cases with similar projections to both spaces at the same time. However, something to take into account is that during normal operating conditions both indices present different variances. On the one hand, NOC, observations are consistent with the statistical model, and thus, their SPE values are close to 0. On the other hand, T^2 presents a larger variance, associated with the working conditions of the NOC region window. As a result, T^2 outweighs SPE during normal operating conditions. However, the inverse situation is given whenever a fault appears. This is, the SPE value of a faulty observation usually presents a larger value than the T^2 index, and thus, in this situation, SPE governs the fault detection procedure.

In order to give the same importance to both indices, the absolute normalised distance (d_ϕ) using their respective statistical control limits is proposed when comparing two cases (c_a and c_b) using this combined index:

$$d_\phi\left(c_a, c_b\right) = \left|\frac{T^2_{c_a} - T^2_{c_b}}{\tau_\alpha}\right| + \left|\frac{SPE_{c_a} - SPE_{c_b}}{\delta_\alpha}\right| \tag{3.16}$$

where $T^2_{c_a}$ and $T^2_{c_b}$ are the T^2 values for case c_a and case c_b respectively; SPE_{c_a} and SPE_{c_b} are the SPE values for c_a and c_b; τ_α is the T^2 statistical limit; and δ_α is the SPE limit. A graphical representation of this criterion is shown in Figure 3.9. Note that for this representation, the SPE and T^2 values have been scaled using the respective control limits δ_α and τ_α $\left(SPE_o = \frac{SPE_{\vec{x}}}{\delta_\alpha}\ and\ T^2_o \frac{T^2_{\vec{x}}}{\tau_\alpha}\right)$.

3.3.3 Neighbourhoods based on latent structures

Definition 3.1. Neighbourhood. The neighbourhood of an observation c_a computed with any of the distance criteria d explained in the previous subsection can be designed by the observations closer than a threshold θ as the following relation suggests:

$$N_d(c_a, \theta) = \{c_i\ /\ d(c_a, c_i) \leq \theta\} \tag{3.17}$$

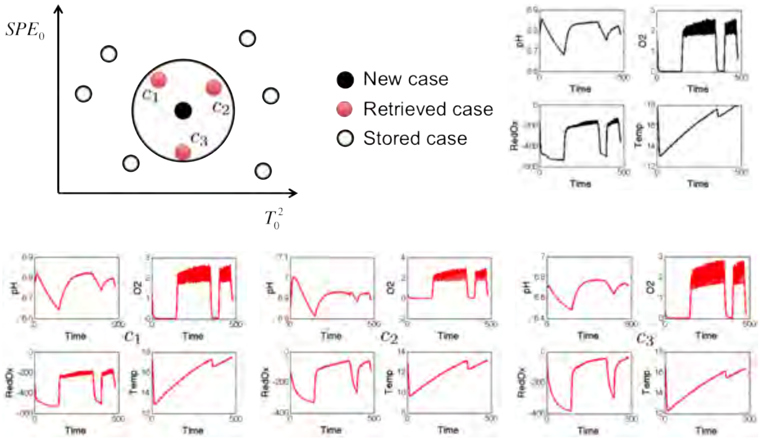


Figure 3.9: Retrieval based on the weighted fault detection index. SPE and T^2 indices shown in this figure have been normalised using the corresponding control limits δ_α and τ_α

Based on this definition, several combinations of the distance criteria presented in the previous subsection can be used to retrieve a set of observations useful for process diagnosis. For example the neighbourhood of NOC observations are expected to be around the origin in the principal component space. Therefore, they would be retrieved as the neighbours of a representative theoretical case located in the origin of coordinates, c_0, with a confidence level α by selecting an appropriate value for the thresholds based on the statistical limits τ_α and δ_α as depicted in Figure 3.10):

$$N_{NOC} = N_{d_{T^2}}(c_0, \tau_\alpha) \cap N_{d_{SPE}}(c_0, \delta_\alpha) \tag{3.18}$$

Note that $\cap$ indicates the intersection of both subsets. Operating in a similar way, it is possible to select the nearest observations with a similar deviation with respect to the projection model, in terms of SPE or T^2 using the following relations. Once an observation c_a has been projected, and the resulting SPE evidences that it is not consistent with the model structure, then a focalised search among neighbours with the same dissimilarity (based on their score projection) can be useful for diagnosis purposes:

$$N_{SPE \wedge t}(c_a) = N_{d_t}(c_a, \theta_t) \cap N_{d_{SPE}}(c_a, \theta_{SPE}) \tag{3.19}$$

A graphical representation of this neighbourhood is depicted in Figure 3.11. Both simple retrieval spaces (d_t and d_{SPE}) are represented using a black circumference and a black cylinder respectively, while the $N_{SPE \wedge t}$ is represented with an orange cylinder.

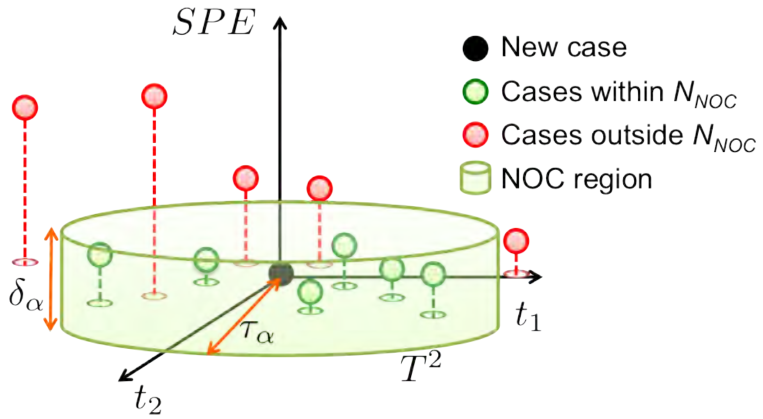


Figure 3.10: NOC neighbourhood (N_{NOC}) of a case placed in the centre of the statistical model (c_0) based on the statistical limits of the model τ_α and δ_α

On the one hand, if d_{SPE} had been used, cases c_1 and c_2 would have been retrieved, although they lay in a different region in the PCS. On the other hand, if only d_t had been used, both c_6 and c_7 would have been retrieved, although they present significant differences with respect the SPE values of c_{new}. Two of these cases are represented in Figure 3.11 in order to exemplify this problem in a graphical way. In case of c_1, it can be seen that its behaviour is significantly different, however, due to the threshold fixed when retrieving cases in the residual subspace (θ_{SPE}), this case would have been retrieved as a similar cases, although variables pH and O2 present completely different behaviours. In the same way, but this time in the principal component subspace, c_7 would have been retrieved as a nearest neighbour due to the threshold retrieval in the PCS (θ_t). Although this cases behaves in a similar way than c_{new}, it presents slight differences in the pH readings, which are projected in the residual subspace, but with a smaller scale than the SPE values of c_{new}. However, by using $N_{SPE \wedge t}$, only cases with similar operating conditions (SPE values) and with the same misbehaviour (t) have been retrieved.

In practical terms, this neighbourhood can be interpreted as a two-stage retrieval procedure that works as described in Algorithm 3.1:

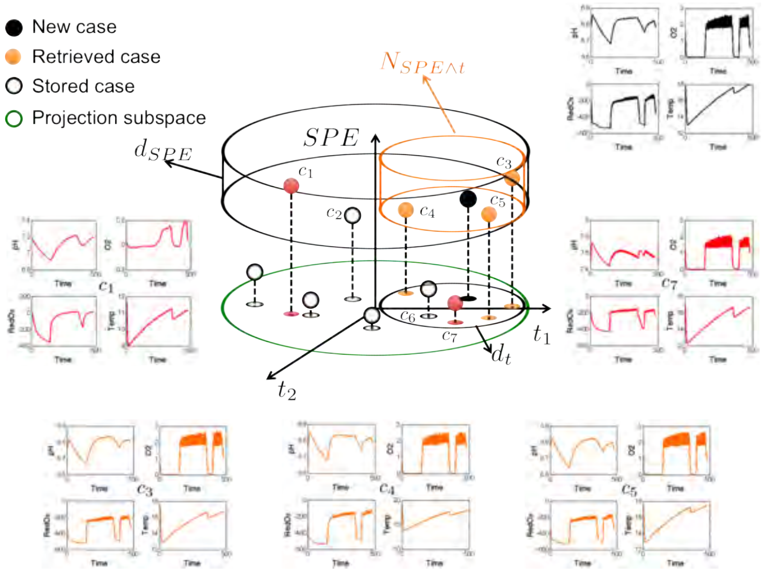


Figure 3.11: Retrieval space based on $N_{SPE \wedge t}$ (orange cylinder) confronted to the single retrieval processes of d_{SPE} (black cylinder) and d_t (black circumference)

Algorithm 3.1 Two-step retrieval for $N_{SPE \wedge t}$

Input: A case base (CB) with all stored cases to compare the new case; the information of the new case to retrieve (c_{new}) described in 3.5; the threshold to retrieve within the RS (θ_{SPE}); and the threshold to retrieve cases based on d_t (θ_t).

Output: The k_t nearest cases to the new retrieved case based on their dissimilarity to the model.

```
function $N_{SPE \wedge t}(CB, c_{new}, \theta_{SPE}, \theta_t)$
    for $i = 1$ to $q_{CB}$ do                    ▷ $q_{CB}$ is the number of cases in the whole case base.
        Compute the distance between $c_{new}$ and $c_i$ using (3.10):
            $d_{SPE,i} = |SPE_{c_i} - SPE_{c_{new}}|$
    end for
    Sort in ascending order (from lowest to highest) the distances in the RS ($d_{SPE}$).
    Keep all cases with $d_{SPE} \leq \theta_{SPE}$, which are the nearest cases to the new one in the residual space, $CB_{SPE}$ ($k_{SPE}$).
    for $i = 1$ to $k_{SPE}$ do
        Compute the distance between $c_i$ to $c_{new}$ using (3.9):        ▷ $c_i$ is the $i$-th case in $CB_{SPE}$
            $d_{t,i} = \sqrt{\sum_{j=1}^{r} \frac{(t_{c_i,j} - t_{c_{new},j})^2}{\lambda_j}}$
    end for
    Sort in ascending order the distances in the PCS ($d_t$).
    Keep only those cases with $d_t \leq \theta_t$ ($k_t$), which are the nearest cases to the new one in the PCS.
    return The $k_t$ nearest cases to $c_{new}$ in the PCS.
end function
```

Note that when computing the different distances, for example, at line 3, this means that the necessary information from the case is retrieved in order to compute this distance. In this example, the SPE values from the new case (c_{new}, which is $SPE_{c_{new}}$) and the i^{th} case retrieved (c_i, which is SPE_{c_i}) are obtained.

In a similar way, a neighbourhood can be restricted to the observations in the hyperplane defined by scores and with a similar value of the T^2 index using the following expression and depicted in Figure 3.12:

$$N_{T^2 \wedge t}(c_a) = N_{d_t}(c_a, \theta_t) \cap N_{d_{T^2}}(c_a, \theta_\alpha) \tag{3.20}$$

Although both retrieval processes combined in this neighbourhood (d_t and d_{T^2}) look within the same subspace (the principal component subspace), d_t is used to focus the search on subtle differences within the PCS. As stated in 3.3.2.3, d_{T^2} compares the projection of faults into the PCS, but only considering the difference between fault magnitudes in this subspace ($\hat{f}$), and consequently, it can confuse faults that alter different (single) score vectors, like c_1 in Figure 3.12 with a different fault direction than the one associated with c_{new}. The same can be applied to c_7, however, in this case happens that the fault direction affects different scores and causes a significant increase in the T^2 value sufficient to surpass θ_{T^2}.

Like it happened with $N_{SPE \wedge t}$, this neighbourhood can be seen as the two-step retrieval process described in Algorithm 3.2.

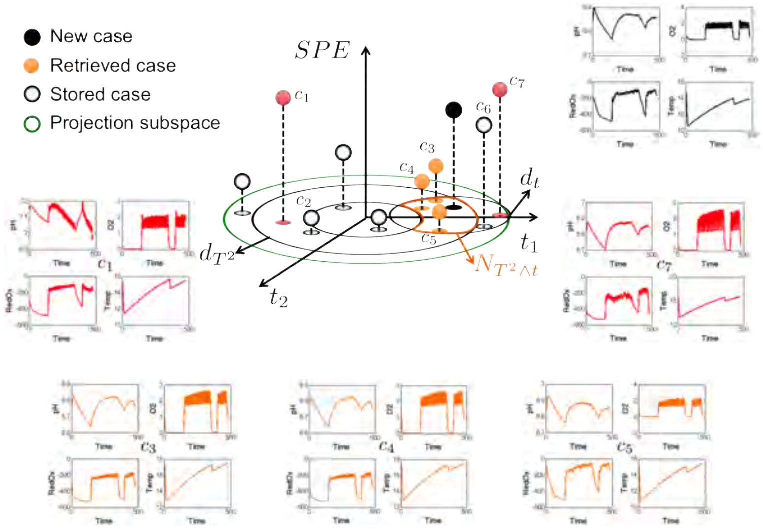


Figure 3.12: Neighbourhood of a new case based on similarity in the T^2 hyperplane

Algorithm 3.2 Two-step retrieval for $N_{T^2 \wedge t}$

```
Input: A case base ($CB$) with all stored cases to compare the new case; the information of the new
    case to retrieve ($c_{new}$) described in 3.5; the threshold to retrieve observation based on the $T^2$ index
    ($\theta_{T^2}$); and the threshold to retrieve the nearest cases based on the Mahalanobis distance in the
    projection subspace ($\theta_t$).
Output: The $k_t$ nearest cases to the new retrieved case based on their similarity to the model.
function $N_{T^2 \wedge t}(CB, c_{new}, k_{T^2}, k_t)$
    for $i = 1$ to $q_{CB}$ do                    ▷ $q_{CB}$ is the number of cases in the whole case base.
        Compute the distance between $c_{new}$ and $c_i$ using (3.13):
            $d_{T^2,i} = \left|T^2_{c_i} - T^2_{c_{new}}\right|$
    end for
    Sort in ascending order (from lowest to highest) the $T^2$ distances for all cases ($d_{T^2}$).
    Keep the all cases with $d_{T^2} < \theta_{T^2}$ (the nearest cases to the new one based on their $T^2$ value,
      $CB_{T^2}$, $k_{T^2}$).
    for $i = 1$ to $k_{T^2}$ do
        Compute the distance between $c_i$ to $c_{new}$ using (3.9):        ▷ $c_i$ is the $i$-th case in $CB_{T^2}$
            $d_{t,i} = \sqrt{\sum_{j=1}^{r} \frac{(t_{c_i,j} - t_{c_{new},j})^2}{\lambda_j}}$
    end for
    Sort in ascending order the distances in the PCS ($d_t$).
    Keep only those cases with $d_t < \theta_t$ (the nearest cases to the new one in the PCS, $k_t$).
    return The $k_t$ nearest cases to $c_{new}$ in the PCS.
end function
```

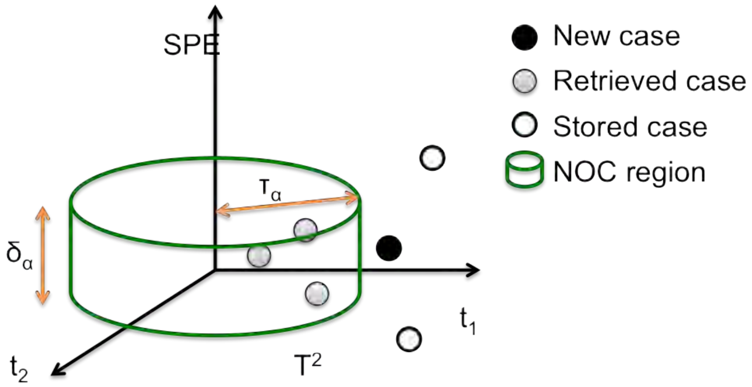


Figure 3.13: Neighbourhood of a new case included in the NOC region

Another possibility when retrieving cases is to locate the set of nearest neighbours within the NOC region. For example, retrieving the nearest subset of in-control neighbours of a faulty observation (c_a) can be useful to determine the set of out-of-control variables, and thus, identify the fault direction and magnitude affecting the observation. Assuming that the new (faulty) observation and the retrieved ones are close enough, the difference in their projection is all due to the effect of the fault. This means that (2.52) and (2.64) can be used to determine the fault directions in the projection and residual subspaces respectively, and when combining both fault direction vectors, obtain the fault direction of the new case. Such neighbourhood can be defined as a refinement of the one in (3.17) by adding the similarity within the PCS in this way: (Figure 3.13).

$$N_{NOC\wedge t}(c_a) = N_{NOC} \cap N_{d_t}(c_a, \theta_t) \tag{3.21}$$

And a graphical representation of the in-control neighbours of a faulty observation is depicted in Figure 3.13.

Or in case of focusing on the observations close to c_a and out of the NOC region:

$$N_{\neg NOC\wedge t}(c_a) = N_{d_t}(c_a, \theta_t) - N_{NOC\wedge t}(c_a) \tag{3.22}$$

Note that in this case the set difference (cases retrieved that are not included in the NOC region) "-" is proposed.

3.4 Classifying observations with latent structures: the reuse step

As introduced in section 3.1, the retain stage of the CBR proposes a solution for the new problem. Moreover, in section 3.2 it was indicated that this stage for diagnosis applications usually is reduced to a classification problem where a class assignment corresponds with a fault type. Therefore, all methods described in this section are focused on determining the class of a new observation based on the classes of the k-NN retrieved using a distance criterion. Three different procedures have been used throughout this thesis to determine the class (fault diagnose) of a detected fault based on the classes of retrieved neighbours: simple voting, a distance-weighted voting, and a weighted voting based on both the distance and class frequency. These three methods are described in more detail in the following three subsections.

3.4.1 Simple voting (SV)

The simple voting is the most simple and common method used for determining the class of a new observation/case. In order to determine the new class of an observation, the method groups the different retrieved cases based on their classes. Then, the class for the new observation is the one with the highest frequency (number of cases within a class). Bringing this approach to fault diagnosis as a classification problem, the different classes will be linked to the different faults we want diagnose. Let us exemplify this concept in a graphical way. Let us suppose that our process only presents two different types of faults we want to diagnose: squares (fault type 1, $\vec{\xi_1}$) and circles (fault type 2, $\vec{\xi_2}$). Now, suppose that the information retrieved for the five nearest cases ($k = 5$, $c_1, .., c_5$) to a new case (c_{new}) and their distribution is the one shown in Figure 3.14.

The simple voting determines that the fault affecting c_{new} is $\vec{\xi_1}$ because this is the most common fault among the five nearest neighbours, even though two out of the three $\vec{\xi_1}$ observations are quite far from the observation, since simple voting does not take into account the distance of the retrieved cases with respect the new case. In order to avoid this problem, one can reduce the number of nearest neighbours retained (k), or add a threshold to limit the distance of the retrieved cases.

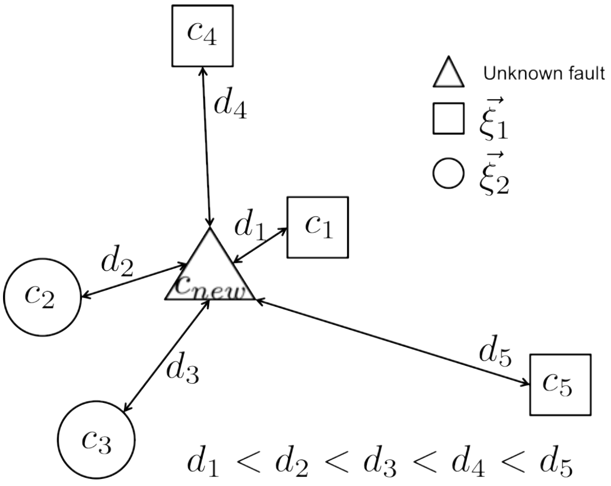


Figure 3.14: Example of a retrieval procedure for a new case (c_{new}) and its five nearest neighbours ($k = 5$)

3.4.2 Distance-weighted voting (DW)

The weighted voting uses the distance of each retrieved observation (c_i) to the new observation case (c_i) in order to weight its influence in the determination of the final class. This means that each case is given a voting weight (w_i) computed as:

$$w_i = \frac{1}{d_i} \tag{3.23}$$

where d_i is the distance of the i^{th} retrieved case to the new observation using a certain distance criterion. Then, these values are accumulated and grouped according to the different classes the problem can be divided into, and denoted as similarity ratio (s_j):

$$s_j = \sum_{l=1}^{q_j} w_l \tag{3.24}$$

where j indicates the class for which the similarity ratio is being computed; and q_j is the number of cases retrieved for this class. The class with the highest similitude ratio is the class assigned to the new observation.

Continuing with the example depicted in Figure 3.14, the distance values (d_i) for each retrieved case c_i and the weighted distance for each observation w_i are presented

Table 3.1: Reuse example: distance (d_i) and weighted voting (w_i) between the new case c_{new} and each retrieved case c_i

d_1	d_2	d_3	d_4	d_5
0.10	0.12	0.15	0.50	0.75
w_1	w_2	w_3	w_4	w_5
10	8.33	6.66	2	1.33

Table 3.2: Reuse example: similarity ratio for both squares (class 1) and circles (class 2)

s_1	s_2
13.33	14.99

in Table 3.1. From these values, the reuse step indicates that the new observation is a circle because it has the highest similarity ratio ($s_2 > s_1$). Consequently, this reuse procedure favours groups of observations close to the new observation.

3.4.3 Distance and class frequency weighted voting (DFW)

A single retrieved case can bias the class prediction given that the rest of observations are far enough from a new observation when using DW. To solve this problem, the class frequency (the number of cases for each class among the retrieved cases) is taken into account for this method. And thus, (3.24) becomes the following expression to compute the similarity ratio (s_j) for the j^{th} class:

$$s_j = q_j \sum_{l=1}^{q_j} w_l \tag{3.25}$$

where q_j is the number of cases retrieved from class j and w_l is the weighted voting for the l^{th} retrieved observations from class j, computed using (3.23).

Finally, note that the three reuse methods presented in this section are sensitive to the presence of clustered “noisy” instances, which are explained in section 3.6 since they affect the case base maintenance policies. Whenever this happens, one has to study the optimal number of retrieved neighbours to minimise this.

3.5 Validation procedure using latent structures: the revise step

Once a solution for a new case has been proposed, the CBR has to evaluate the goodness of the proposal. This section presents the different methods used for determining the best retrieve and reuse strategies during the off-line training procedure. In this case, the validation procedure affects both the case base constitution (to guarantee that all available observations are used) and the study of how the reuse process misbehaves (which classes are confused among them). More specifically, the case based division is carried out using the n-fold cross-validation procedure (detailed in subsection 3.5.1), the classification performance is checked using the confusion matrix and its associated statistics (presented in subsection 3.5.2) and the best classification scheme is found based on the area under the receiver operating curve (AUC), which is explained in subsection 3.5.4).

3.5.1 Cross-validation and case base building

In n-fold cross-validation, the available data is divided into n folders containing approximately the same number of examples. The stratified version of this technique takes into account the several ratios among classes present in the original set. Once the data is divided, one of the n folds of samples is retained for validation of the model formed by the remaining $n-1$ data fold. This process is repeated n times (once for each fold) (Kohavi, 1995) and the average results indicate the performance of the method. Finally, Figure 3.15 presents this methodology in a graphical way.

3.5.2 Confusion matrix and its associated statistics

The classification performance will be evaluated based on the confusion matrix. A confusion matrix is a form of contingency table showing the differences between the true and predicted classes for a set of labelled examples, as shown in Table 3.3 (Bradley, 1997).

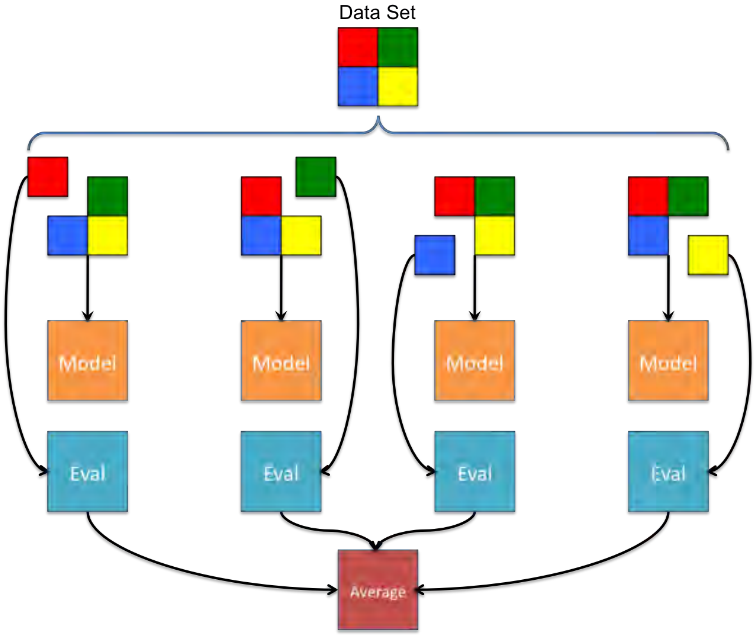


Figure 3.15: n-Fold Cross Validation graphical procedure

Table 3.3: Confusion Matrix elements

		Real Class	
		Ref	**No Ref**
Predicted Class	**Ref**	TP	FP
	No Ref	FN	TN

Table 3.4: Extended Confusion Matrix elements

		Real Class			
		Class 1	Class 2	$\dots$	Class N
Predicted Class	Class 1	$TP(1,1)$	$FP(1,2)$	$\dots$	$FP(1,N)$
	Class 2	$FP(2,1)$	$TP(2,2)$	$\dots$	$FP(2,N)$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	Class N	$FP(N,1)$	$FP(N,2)$	$\dots$	$TP(N,N)$

These labels stand for:

- TP: refers to true positives and are cases correctly predicted that belong to the reference class.
- FP: refers to false positives, which are observations predicted as from the reference when they are not.
- FN: refers to non-reference cases that are classified as from the reference class.
- TN: refers to non-reference instances correctly identified as such.

Note that fault detection and diagnosis fixes the reference class to NOC observations, false negative are associated to false alarms (NOC cases labelled as AOC) and false positive are equivalent to missed detections (AOC instances not detected, or labelled as NOC).

In case of diagnosis, it is a multi-class problems (with more than two classes), the confusion matrix has to be extended as shown in Table 3.4. In this case, $TP(i,i)$ stands for cases of class i correctly classified as i; while $FP(i,j)$ refers to cases that were classified as class i but their real class was j (false positive); and N is the number of diferent classes.

Because the extended confusion matrix does not have all elements found in the binary confusion matrix (TN and FN), the different statistics associated with the confusion matrix, which are explained afterwards, cannot be computed directly. The easiest way to tackle this problem is to binarise the extended confusion matrix (Sokolova and Lapalme, 2009), which consists in fixing one class as the reference class, and aggregate the other $N-1$ classes as non-reference, grouping observations in the four cells TP, FN, FP and TN. This procedure is repeated N times (once for each class), while keeping each of the binary matrices. When all matrices are available, there are two different approaches to measure the classifier performance:

- Macro-averaging. This procedure computes the performance indices for each binary matrix and then averages these values among all classes to indicate the global classifier performance.
- Micro-averaging. This procedure accumulates all binary matrices into one, and then computes the performance indices from this accumulated table.

Note that macro-averaging treats all classes equally (by averaging the different performance indices), while micro-averaging benefits larger classes (by accumulating the values and then computing the indices).

Other statistics commonly used to evaluate performance of classifiers based on the confusion matrix are the followings: accuracy (subsubsection 3.5.2.1), precision (subsubsection 3.5.2.2), sensitivity (subsubsection 3.5.2.3) and specificity (3.5.2.4).

3.5.2.1 Accuracy (ACC)

Accuracy measure the proportion of correctly classified cases among all the cases used for testing, and is computed as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.26}$$

3.5.2.2 Precision (PRE)

Precision measure the proportion of correctly classified cases from the reference class from all cases that were predicted as the reference one, and is computed by:

$$PRE = \frac{TP}{TP + FP} \tag{3.27}$$

3.5.2.3 Sensitivity (SEN) or true positive rate (TPR)

Sensitivity measures the ratio between cases correctly classified as the reference class among all cases that have been predicted as the reference class, and is computed by:

$$SEN/TPR = \frac{TP}{TP + FN} \tag{3.28}$$

3.5.2.4 Specificity (SPF)

Specificity measures the ratio between cases correctly classified as non-reference class among all cases that have been predicted as non reference class, and is computed by:

$$SPF = \frac{TN}{TN + FP} \tag{3.29}$$

3.5.3 Receiver operating characteristic (ROC) curve

The ROC curve representation is a two-dimensional graph where the y-axis represents TPR and the x-axis indicates the false positive rate (FPR) , or what is the same, 1-specificity (Fawcett, 2006). Based on this, there are four main points in the ROC space, which are:

- (0,0): this point represents a classifier that always misclassifies cases from the reference class, but never misclassifies non-reference instances. This is equivalent to classify all observation as non-reference instances.
- (0, 1): this point is related with the perfect classifier. This is, any classifier within this location never misclassifies an observation, and thus, this is the ideal scenario.
- (1, 0): any classifier that lies in this zone always misclassifies, and thus, this is the worst case scenario.
- (1, 1): this point refers to classifiers that cannot identify observations outside of the reference class, which is equivalent to always classify new observations as from the reference class.

Therefore, the closer a point is to (0, 1), the better the classifier is. Contrarily, the closer a classifier is to (1, 0), the worse the classifier is. Finally, the line that

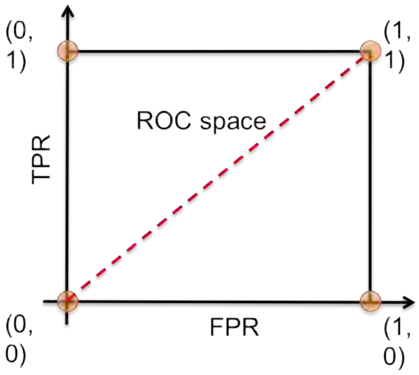


Figure 3.16: Example of ROC curve

links points (0, 0) to (1, 1) represents the random classifier performance. Any classifier has to lie in the upper triangular space above this line, and thus, this line serves as a minimum performance reference. Finally, this information is presented in a graphical way in Figure 3.16.

In order to build the ROC curve, several tests are carried out altering the parameter(s) of a classifier, and the TPR and FPR pairs for each test are represented in this space. Alternatively, the method devised in Fawcett (2006) can be used to represent the performance of a single classifier. This method is based on the positive probability (probability of correctly classifying each reference class instance) for that classifier.

3.5.4 Area Under the ROC Curve (AUC)

Because in some operating points sensitivity can be increased with a minor loses in specificity and in others this is not possible, a non-ambiguous possible comparison of performance can be achieved by computing the area under the ROC curve (AUC). A simple way of computing this value is using the trapezoidal integration method described in Bradley (1997).

$$AUC = \sum_i \left\{ (FPR \times \Delta TPR) + \frac{1}{2} \left[\Delta FPR \times \Delta TPR \right] \right\} \tag{3.30}$$

$$\Delta FPR = FPR_i - FPR_{i-1} \tag{3.31}$$

$$\Delta TPR = TPR_i - TPR_{i-1} \tag{3.32}$$

Where FPR_i represents the FPR value of the actual point (i), FPR_{i-1} represents the FPR value of the previous point ($i-1$), TPR_i represents the TPR value of the actual point (i) and TPR_{i-1} represents the TPR value of the previous point ($i-1$).

The AUC ranges between 0 and 1, where 1 indicates a perfect classifier (classifiers in the (0, 1) ROC region); 0 is related to a classifier that always misclassifies (classifiers in the (1, 0) ROC region); and 0.5 is associated with the random classifier performance. Therefore, the closer the AUC is to 1, the better the classifier is.

3.6 Case base maintenance: the retain step

The retain step is the last stage of a CBR and is the one that gives it the learning capability. The main idea of the retain stage is to determine the set of new and stored observations to carry out for further classifications. Two basic approaches can be used for this task: 1) store every new observation; and 2) store each misclassified case. The former has the drawback that all observations are stored, even if they were correctly classified using the original/previous case base. The latter only focuses on storing the wrongly classified observations to classify new observations. However, this second approach does not take into account that keeping new observations can worse the classification ratio of the original or previous case base. Moreover, the larger the case base is, the more time it takes to retrieve cases (there are more instances to compare new observations with), and thus, the computational cost of the CBD constantly increases. Since in our case, the CBR is used for diagnosing the cause of the fault (as will be exemplified in chapter 5) or to determine the operating point of the process (depicted in chapter 4), any reference to classes within this section refers to either of both situations.

In order to maintain the minimum case base possible while also accounting the overall case base performance, three methods have been considered in this thesis: the decremental reduction optimisation procedure (DROP) 4 (Wilson and Martinez, 2000), the instance-based learner 3 (IB3) (Aha et al., 1991), and the instance-based learner UdG (IBUdG) (Burgas, 2013). Both DROP4 and IB3 were identified in Ruiz (2008) as the best methods for the case base maintenance when applied to WWTP, and thus initially considered in this thesis, while IBUdG is a new maintenance policy that reduces

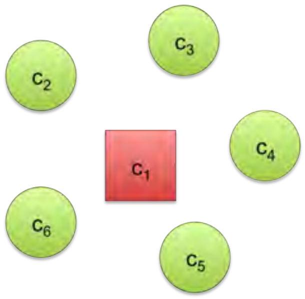


Figure 3.17: Example of a noisy instance. c_1 is considered as a noisy instance because it is a single square instance surrounded by circle class cases (from c_2 up to c_6)

the computation cost when building the best case base for future classifications. Each of them are detailed in the next three subsections.

3.6.1 Decremental reduction procedure (DROP) 4

The family of decremental reduction procedures are focused on reducing noisy instances from a case base (Wilson and Martinez, 2000).

Definition 3.2. Noisy instance. A noisy instance is an observation surrounded by cases from a different class, like it happens in Figure 3.17 with instance c_1 (a square class instance), which is surrounded by circle class instances (from c_2 up to c_6).

Based on this definition, DROP4 focuses on keeping the border points (the closest instances of different classes) of the different class clusters. This means that DROP4 removes instances within the central cluster of a class cluster, since they are not critical points when deciding the classification of a new instance. Let us explain this idea in a more graphical way. Imagine that the cluster of observations in Figure 3.18 is a subspace of the classification problem. All observations in this cluster are from the same case (circles), and thus, the central instances c_1, c_2 and c_3 can be correctly predicted using the border points c_4, c_5, c_6, c_7 and c_8. As a result, DROP4 removes these instances because they are not necessary to classify new observations of the same class that lie within the border cases.

However, now let us suppose that the situation depicted in Figure 3.19 is another subspace within the problem subspace. Both c_1 and c_2 (central cluster instances) are from a different class (squares) than the class of the surrounding or border instances

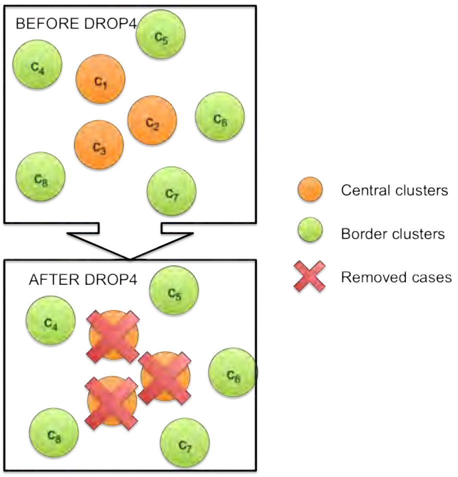


Figure 3.18: Example of DROP4 removing a set of central cases (c_1, c_2 and c_3) since the border cases (c_4, c_5, c_6, c_7 and c_8) are capable of predicting any future new instance within the cluster

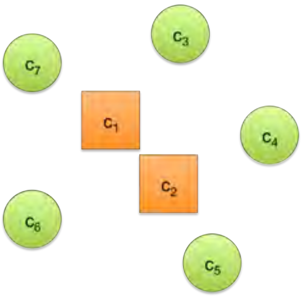


Figure 3.19: Example of DROP4 keeping all instances since the central cluster (c_1 and c_2) are from a different class (square) than the class from the border cluster (c_3, c_4, c_5, c_6 and c_7, which are circles)

(c_3, c_4, c_5, c_6 and c_7, which are circles). Since removing both central cases would wrongly classify any new square instance within the circle's border, DROP4 keeps both instances for future classifications.

With respect to other versions of the algorithm, DROP4 incorporated a noise filter in order to remove noisy instances before looking for the central and border clusters. The main problem with noisy instances is that they initially misclassify the nearest neighbours, and thus, keeping them only produces a lower performance. Considering all of this, the pseudo-code for DROP4 is presented in Algorithm 3.3.

Algorithm 3.3 DROP4 pseudo-code

```
Input: A case base ($CB$) with all stored cases to compare the new case; and the number of neighbours
    to retrieve ($k$) when checking for noisy instances.
Output: The reduced case base $RCB$ without noisy instances and the minimum number of instances.
function DROP4($CB$, $k$)
    Initialise the reduced case base using the initial case base: $RCB = CB$.
    Find all associates to each case $c_i$ in $RCB$ ($A$).    ▷ Associates are all cases that have $c_i$ as a nearest neighbour
Careful noise filtering
    for $i = 1$ to $q_{CB}$ do    ▷ $q_{CB}$ is the number of cases in the whole case base.
        if $c_i$ is misclassified for its nearest neighbours then
            Remove it if not necessary (removeIfNotHelping($c_i$, $RCB$)).
        end if
    end for
More aggressive noise filtering (centroids removal)
    Sort instances based on their distance to the closest enemy (furthest ones first).    ▷ An enemy is a case of a different class than the case we are looking at.
    for $i = 1$ to $q_{RCB}$ do    ▷ $q_{RCB}$ is the number of cases in the reduced case base.
        removeIfNotHelping($c_i$, $RCB$).
    end for
    return The reduced case base $RCB$.
end function
Instance removal function (removeIfNotHelping)
Input: A case instance ($c_i$) whose influence to other observations has to be checked; and the case base
    ($CB$) where the influence of $c_i$ has to be checked.
Output: The resulting case base ($RCB$) does not have $c_i$ if is not necessary to improve the classification
    ratio of $CB$. The initial case base $CB$ otherwise.
function REMOVEIFNOTHELPING($c_i$, $CB$)
    $RCB = CB$.
    Compute the number of associates of $c_i$ ($a_i$) that are correctly classified when $c_i$ is included in $CB$ ($with$).
    Compute the number of associates of $c_i$ ($a_i$) that are wrongly classified when $c_i$ is removed from $CB$ ($without$).
    if $without \geq with$ then
        Remove $c_i$ from $RCB$.
        for $a = 1$ to $q_a$ do    ▷ $q_a$ is the number of associates to $c_i$
            Remove $c_i$ from all its associated cases ($a_i$).
            Get a new nearest neighbour for $a$ in $RCB$.
        end for
    end if
    return The resulting case base $RCB$.
end function
```

3.6.2 Instance-based learner 3 (IB3)

Contrary to the decremental family of algorithms, which start from the whole case base and check which instances can be removed; the instance-based learning procedures start with an empty case base, and the appropriateness of adding a new instance is evaluated at each iteration. IB3 is a noise-tolerant extension of its predecessor IB2, which only adds a new instance c_i to the training set if it is misclassified by its k nearest neighbours. IB3 adds two features to IB2:

- A classification record of each instance in the case base computed as the number of correct classifications divided by the times the instance has been retrieved. This record is used to predict the future performance of each instance.
- Uses the significance test described in (3.33) to determine what instances are good classifiers, which are kept for further classifications; and what cases are noisy, which are discarded from the case base.

$$\rho = \frac{b + \frac{\alpha^2}{2q_{CB}} \pm \alpha\sqrt{b\frac{1-b}{q_{CB}} + \frac{\alpha^2}{4q_{CB}}}}{1 + \frac{\alpha^2}{q_{CB}}} \tag{3.33}$$

where b is the classification performance of the instance computed as successful classifications divided by the number of attempts; α is the confidence interval; and q_{CB} is the number of instances stored in the case base.

The main idea of IB3 is working with acceptable instances when building the reduced case base, while dropping those instances with a poor performance. An acceptable case is an observation whose performance record is significantly above the overall class performance. On the other hand, an observation with a poor classification is an instance with a significantly poor performance record. The usual values for both adding a new acceptable instance is $\alpha = 0.9$, while the confidence limit for dropping a poor instance is set to $\alpha = 0.75$ (values proposed in Aha et al. (1991)). The main idea is to difficult the acceptance of new instances (performance over the 90% of the class record), while the latter value is designed to drop instances with moderate poor classification. The pseudo-code describing the IB3 procedure is presented in Algorithm 3.4.

Algorithm 3.4 IB3 pseudo-code

```
Input: A case base ($CB$) with all stored cases to compare the new case. Note that we use the default
    values confidence level $\alpha$ for determining acceptable and poor instances.
Output: The reduced case base $RCB$ with only acceptable instances.
function IB3($CB$)
    Initialise the reduced case base as an empty set: $RCB = \emptyset$.
    for $i = 1$ to $q_{CB}$ do                ▷ $q_{CB}$ is the number of cases in the whole case base.
        for $j = 1$ to $q_{RCB}$ do       ▷ $q_{RCB}$ is the number of cases in the reduced case base $RCB$.
            Compute the distance between $c_i$ and $c_j$ using a given distance criterion ($d(c_i, c_j)$).
        end for
        Look for the closest acceptable instance ($c_a$) to $c_i$.
        if there are no acceptable cases then
            Initialise $c_a$ with a random instance in $RCB$.
        end if
        if class($c_i$) = class($c_a$) then
            classification = correct
        else
            classification = incorrect
            Add $c_i$ to the reduced case base ($RCB = RCB \cup c_i$).
        end if
        for $j = 1$ to $q_{RCB}$ do
            if $d(c_j, c_i) \leq d(c_a, c_i)$ then          ▷ $c_j$ is at least as close as $c_a$ to $c_i$
                Update the classification record of $c_j$.
                Remove $c_j$ from $RCB$ if its classification record is poor ($\rho < 0.75$).
            end if
        end for
    end for
    Remove all non-acceptable instances in $RCB$ ($\rho \leq 0.9$).
    return the reduced case base $RCB$.
end function
```

3.6.3 Instance-based learner UdG (IBUdG)

Although than in Ruiz (2008) it was demonstrated that a combination of both DROP4 (initially applied to remove the noisy instances in the case-base) and IB3 (applied whenever the retain function was called) gave the best results, this procedure has an important computational cost. Based on the incremental methodology (start with an empty case base and then add new instances), the IBUdG algorithm is proposed. Its main idea is to track the nearest enemies (cases from a different class of the one being studied) in order to keep the cases closest to the border line (like DROP4) but starting with an empty case base (like IB3). By doing so, the method only keeps the minimum number of instances to determine the frontier between the different classes. The pseudo-code for the IBUdG is described in Algorithm 3.5. As a final note, consider that for multi-class (more than two classes) problems, the method looks for the nearest enemy to each of the other classes.

Algorithm 3.5 IBUdG pseudo-code

```
Input: A case base ($CB$) with all stored cases to compare the new case; and the number of nearest
    enemies to retrieve ($k$).
Output: The reduced case base $RCB$ with only acceptable instances.
function IBUdG($CB$, $k$)
    Initialise the reduced case base as an empty set: $RCB = \emptyset$.
    for $i = 1$ to $q_{CB}$ do                    ▷ $q_{CB}$ is the number of cases in the whole case base.
        Find the nearest enemy to $c_i$ ($ne_i$).
        if $ne_i$ is not in $RCB$ then
            Add $ne_i$ into $RCB$ ($RCB = RCB \cup ne_i$).
        end if
    end for
    return the reduced case base $RCB$.
end function
```

A comparison between three retain strategies is presented in chapter 4 in section 4.3.4 in terms of correct classification, as well as in terms of stored cases in the final case base. These results were partially published in Berjaga et al. (2013b). Additionally, Appendix B includes a computational time comparison between the three strategies for a binary classification problem, where both classes completely overlap (the worst-case-scenario for a classifier).

3.7 Benefits, drawbacks and assumptions of CBRs based on LS

This chapter proposed the combination of a CBR methodology using LS in order to complement the limitations of using both methodologies (MPCA and CBR) independently. On the one hand, LS statistical models are capable to determine if a process is under normal operating conditions or not (**fault detection**), and in case of an AOC observation, the statistical method can determine the subset of variables associated with the fault, but the model is not capable of identifying the root cause of the fault. On the other hand, machine learning algorithms like CBR can relate a series of evidences to their root cause, given that the appropriate rules have been defined, or in the CBR case, some previous experiences have been stored to propose a new solution.

In order to have a good statistical model, a large and rich database is required, which sometimes is not the case for certain batch processes. Moreover, the limits for the fault detection indices (T^2 and SPE) are statistical limits, and thus, a series of false alarms, and even missed detections, are expected to happen. When combining the statistical model with a CBR, past situations can be used to reduce this false alarm ratio, and at the same time, reduce the number of observations required to have a representative model for all in-control situations.

Nonetheless, machine learning methods require that their input parameters (attributes) are independent among them in order to guarantee a superior performance and simpler implementation (Jakulin and Bratko, 2003). Although MPCA (or unfold-PCA) does not grant the independence among principal components (they are uncorrelated), it does guarantee that adding a new attribute (for example keeping one more principal component) adds new information for solving a problem. Moreover, if each faulty observation/case in the case base has the information regarding the root cause (indicated by a process expert), then analysing the information of the nearest neighbours can be used to relate how variables misbehave for a given root cause. By combining both methodologies, the operator can interpret and identify the problem in a more natural way, since CBR works with analogies, like the human brain.

Table 3.5: Extended Confusion Matrix elements

		Real Class		
		Class 1	Class 2	Class 3
Predicted Class	Class 1	95	0	5
	Class 2	0	100	0
	Class 3	0	0	100

But the combination of both techniques also presents some drawbacks. The first one, and most significant, is that past situations and its solutions has to be correctly classified to build the initial case base and consequently, to apply the methodology. Also related with the case base, a CBR can only classify observations with the current set of typologies. This means, that when a new problem type appears, the CBR either misclassifies it, or indicates that is not capable to assign a cause/known state for the new case. In order to overcome this problem, novelty discovery techniques could be applied to indicate the user that a new pattern has been discovered. This would lead to assign a new label to these new situations, and consequently, any new observation can be identified. The last problem related with CBD is that is a supervised learning algorithm. This means that in order to check if the proposed solution is correct (the revise step), it requires an external opinion (the operator or the process expert) to validate the appropriateness of the solution. However, the user feedback can be reduced to a subset of observations (or problem typologies) based on the confusion matrix. Let us exemplify how this can be done with the extended confusion matrix shown in Table 3.5.

For the case being, class 2 is not confused with any other class, so the revise step does not require the validation of class 2 instances. The same can be applied to observations classified as class 3, since they are never classified as neither class 1 nor class 2. However, five observations were identified as class 3 situations although they were class 1. Consequently, only observations classified as class 3 have to be checked in order to disregard a misclassified class 1 observation.

Up until now, all drawbacks have been pointed out to the CBR, and thus, problems with the MPCA/unfold-PCA model have been omitted. Two main concerns affect the statistical model of a process: 1) modifying the working point of the process studied; and 2) the deviation of the process throughout time. The former can be corrected, or at least attenuated thanks to the CBR, while the latter is further discussed in the next section (section 3.8). Let us focus on that the process can operate within several working points (like most of batch processes). Additionally, suppose that there are as many statistical models as working points the process can operate into, and thus, there is as many case bases as working points (cases bases use a single statistical model to project observations). Now, let us suppose that the model and case base associated with the i^{th} working point is being employed. A new batch is analysed and when retrieving the k nearest observations, most of them are tied to the j^{th} working point. Because of this, the reuse step determines that the working point has been switched, and thus, the model to use for the next batch release is automatically changed. At the same time, the current batch is classified according to the j^{th} case base. Consequently, the CBR can be used to recognise these operating points and switch monitoring or control strategies according to it. Related with this point, and as stated in the previous chapter, data-driven methods do not usually incorporate structural information about the process. Therefore, they are not able to distinguish between process faults or measurement faults. However, the CBR can incorporate this information within the diagnosis structure presented in section 3.2, and thus, based on the information of previous cases, indicate the type of the fault (simple/multiple/complex), and even present the fault direction and magnitude. This information could be used, for example, to reconstruct misreadings in sensors, and thus, avoid stopping the production line.

3.8 Case-based diagnosis of batch processes: discussion

Within this chapter, the basic idea of case-based diagnosis (CBD), benefits and drawbacks have been presented. The main point behind combining both an MPCA/unfold-PCA and CBR is to overcome the individual limitations of they independent use. In fact, as stated in the previous section, the resulting combination gives rise to a simpler diagnosis procedure than the direct application of the contribution plots. Moreover, the learning capabilities of a CBR reduce the necessity of building new models whenever a new problem typology appears, like it happens with PLS-DA (Sjöstrom et al., 1985) or SIMCA (Wold et al., 1984).

Moreover, since the diagnosis structure indicated in section 3.2 contains detailed information from previous experiences/cases, the diagnosis aim can be changed to indicate whatever the process operator needs at a time, which may include different configurations for a better performance. This capacity is further detailed in the next two chapters, which present off-line and on-line application of the CBD in two different fields: chapter 4 presents how the CBD can be used to alert process operators of the change in the granulation degree of a sequencing batch reactor (SBR); while chapter 5 details how the methodology can determine the final quality of injected parts at the end of the batch.

The final objective of any monitoring procedure is its application during the production chain in order to keep the desired product specifications. Nonetheless, and as explained in section 1.2, data-driven methods are divided into two phases or stages: the first one builds the statistical model of the process using historical data (Phase I or off-line analysis); while the second stage applies this model for the on-line monitoring of the process (Phase II or model exploitation). When in this second stage, the validity of the model has to be checked continuously, and based on this performance, indicate the applicability of the model. Usually, the operator output helps to validate the proposed solution, at least, for each dubious observation, and based on this, the degradation of the monitoring procedure can be estimated. One possible index is to define a temporal window (the last k measurements) from within the classification rates must hold over a certain θ threshold (i.e. for the last five batches the performance has to be at least 95%).

Whenever the monitoring scheme does not keep the expected performance, there are two question that have to be answered:

- Is the process model still valid? Let us suppose that the performance constraints were fixed based on the number of false alarms (NOC observations labelled as AOC) or missed detection (AOC observations classified as NOC). Then, whenever this quota is not satisfied, the process model has to be rebuilt. In order to take into account the process drift, the last set of NOC observations have to be considered.

The previous observation and model can be kept, just in case that the process regresses to the previous operating conditions. A plausible cause of this situation is a change in the working point of the process that was not initially considered.

- Is the diagnosis process reliable? This situation can happen whenever a new problem typology appears, or the process model is changed. When this happens, the proposed CBR configuration has to be reevaluated in order to find the best configuration (retrieve, reuse, and retain strategies) to proceed with the diagnosis.

Note that in order to update the process model and/or the diagnosis procedure, the process has to be stopped to avoid releasing out-of-specification products. As a result, the process reconfiguration should be as short as possible in order to reduce the time the production remains stopped. A common procedure is to define a series of control points, where the process validity is checked. Whenever the specifications at the current point are not met, the operator modifies the necessary parameters in order to adjust the batch. This procedure is repeated for each control point until the batch is released and the final quality product is estimated. If the procedures followed by the operator corrected the misbehaviour, this information is stored in the CBR to replicate the steps followed for future events. If the operator was not capable to solve the problem, this knowledge can be used to extrapolate which steps were not sufficient (when comparing with other retrieved observations) in order to guarantee that future interventions will be satisfactory.

Apart from the appropriateness of the model and diagnosis scheme, the other important point is the computation time. This value should be strictly lower than the batch release time, and at the same time, allow the operator to correct the process, if the diagnosis is not part of the control loop. With the current hardware technology, computation time is not as critical as was before, however, and thanks to the improvements in networks, the monitoring process takes place in an external network element (for example a computer connected to the process using a certain connection protocol). This means that the transmission time and missing samples due to network overflow have to be taken into account. Several works address the problem of missing data when working with PCA models such as Nelson et al. (1996), Arteaga and Ferrer (2005) or Sernells and Verdonck (2008) among others; and these solutions can be applied to deal with information dropping due to network problems (like they were missed measurements). In fact, the CBR can be used to find the subset of closest observations to the new batch (with incomplete samples), and based on the information of the nearest neighbours, propose a more accurate approximation.

4

Identification of the granulation state in an SBR process

This chapter present the results of applying the CBD methodology exlained in chapter 3 into the WWTP field. In this case, the process studied presents two main operating points (floccular and granular) when treating wastewater, based on the size of the sludge granules. Since the granulation state is a key parameter for the correct treatment of wastewater, the main objective of the CBD is to inform the operator of a the granulation change. In order to do so, two different strategies will be considered: use the information of both states to build the statistical model of the process, and thus, differences between observations have to be found on the projection subspace (using d_t); or build single class models, and since the correlation structure of both methods differ, look for differences in the residual subspace (using d_{SPE}). Results presented in this chapter were partially published in Berjaga et al. (2013a) and Berjaga et al. (2013b).

In order to do so, this chapter is divided as follows. Firstly, a brief description of the problem and process are carried out (sections 4.1 and 4.2 respectively). Next, an off-line analysis is described (section 4.3) in which the final goal is to find the best configuration to differentiate between both states. To do so, the different approaches for the automatic diagnosis of the granulation state are presented (subsection 4.3.1), and then compared (subsection 4.3.2). Based on these results, the variable(s) and stage(s) for which this difference is most observable are identified and then used to estimate the granulation from a floccular into a granular operating condition within the transition period. After this, an on-line application of the methodology is described in section 4.4 in which only floccular observations were initially available. The main objective in this scenario is to inform of any granularity change in the process.

4.1 Problem description

Nutrient removal in WWTPs is usually accomplished in continuous systems in which the biological treatment happens in a reaction basin and the sludge is separated from the treated water in a settler. With the improvement of biological nutrient removal, WWTP configurations have changed over the years with a consequent increase in the land requirements of treatment facilities.

Sequencing batch reactor (SBR) technology is a fill-and-draw activated sludge system where all reactions and sludge separation can take place in the same basin. An SBR converts the conventional wastewater treatment process from space-course to time-course, which substantially reduces space occupation (Tchobanoglous et al., 2003). SBRs allow greater flexibility for adapting phases required for nutrient removal according to the influent variations (EPA, 1999). However, more sophisticated equipment is necessary for their operation and automation (Corominas, 2006). SBRs operate applying cycles that are repeated over time. A cycle is built with a sequence of phases according to the main objective desired. Given this cyclic nature repeated over time, MPCA can be used to model the variations of the treatment procedure as will be stated later on.

Aerobic granular sludge (AGS) is an alternative technology for wastewater treatment that can handle high loading rates or large volumes of waste in small facilities (Liu and Tay, 2002). The fast settling properties of the granules allow the treatment to be more compact than conventional systems with floccular sludge. However, the major challenge of this technology is the start-up of the process, as granulation conditions are not yet fully understood. Several factors have been described for granular formation such as substrate composition, but the major selection pressures have been identified as short settling times and high volume exchange ratios (VERs) with the purpose of selecting bioparticles according to their settling velocity (Liu et al., 2005). However, a high washout of biomass is usually obtained during the process (McSwain et al., 2004). Furthermore, granulation of systems treating low-strength wastewater with lower loading rates than 2 kg $\mathrm{CODm}^{-3}\mathrm{d}^{-1}$ takes longer to reach the steady state (Tay et al., 2004) and the transition state, where a system becomes fully granulated, is sometimes difficult to detect. Some off-line analysis such as particle distribution may help in the turning point detection, but they are expensive and time consuming.

Activated sludge systems are usually equipped with on-line sensors such as pH, dissolved oxygen and oxidation-reduction potential (ORP) sensors, which have been proven to give ample information regarding biological nutrient removal processes, through their key parameters indicators (KPI) (Puig et al., 2005; Yuan et al., 2008). The information of these sensors has been demonstrated useful for controlling the reaction and improving the process design. In this work, the off line analysis of batch records of this process (following the CBD strategy proposed in the previous chapters) has demonstrated to be useful to identify variations in the process behaviour from floccular to granular.

The objective is to automatically identify the status of the SBR reactor (floccular, transient, granular) by means of the information provided by the sensing devices (pH, DO, ORP and temperature sensors) and the case based diagnosis methodology for batch process introduced in the previous chapters. Two different scenarios will be depicted: one in which both granularities (floccular and granular) are present (section 4.3) and the approach is applied off-line; and another one for which only floccular batches were initially available (section 4.4) and CBD is used to launch a granulation change warning (real-time release monitoring). For both scenarios, the CBD methodology is presented as an alternative to an off-line analysis of the granules.

4.2 Process description

A 30L lab-scale sequencing batch reactor (SBR) located in the Laboratory of Chemical and Environmental Engineering (LEQUIA- UdG) was used for granulation purposes.

The SBR was initially seeded with floccular sludge which was collected from a full-scale domestic wastewater treatment plant (WWTP) located in Sils-Vidreres (Girona, Spain). The SBR also treated 39 L of wastewater per day collected from the Quart WWTP. An 8-h cycle with two feeding steps was implemented, introducing the wastewater under anaerobic or anoxic conditions to enhance phosphorus removal and denitrification as depicted in Figure 4.1. The pilot plant was equipped with a monitoring and control system made of three parts: i) pH, ORP and DO-temperature probes; ii) data acquisition and switch on/off cards; and iii) interfaces developed in LabWindows®. DO setpoint during aerobic phases was fixed at 2 mg O_2 L^{-1} by an air on/off strategy. On-line mean values were obtained every 5 s and stored in a simple ASCII file for further processing giving 4320 samples. Considering that reactions in biological processes are very slow, the number of samples will be reduced to a sample per minute, and thus, 480 samples for each variable and batch will be processed.

A 30L lab-scale sequencing batch reactor (SBR) was used for granulation purposes. The SBR was seeded with floccular sludge from a full-scale domestic wastewater treatment plant (WWTP) located in Sils-Vidreres (Girona, Spain). The SBR also treated 39 L of wastewater per day from the Quart WWTP. An 8-h cycle with two feeding steps was implemented, introducing the wastewater under anaerobic or anoxic conditions to enhance phosphorus removal and denitrification as depicted in Figure 4.1. The pilot plant was equipped with a monitoring and control system made of three parts: i) pH, ORP and DO-temperature probes; ii) data acquisition and switch on/off cards; and iii) interfaces developed in LabWindows®. DO setpoint during aerobic phases was fixed at 2 mg O_2 L^{-1} by an air on/off strategy. On-line mean values were obtained every 5 s and stored in a simple ASCII file for further processing giving 4320 samples. Considering that reactions in biological processes are very slow, the number of samples will be reduced to a sample per minute, and thus, 480 samples for each variable and batch will be processed.

The granular formation is based on a two-step process and similar to crystal growth. Firstly, fine particles are induced; and then, on the surface of these particles, working as carriers, bacteria develop mature granules (Jin et al., 2012). Therefore, a transition state between the floccular and granular states occurs during the granulation process. The detection of this turning point where the fine particles are colonized to form mature

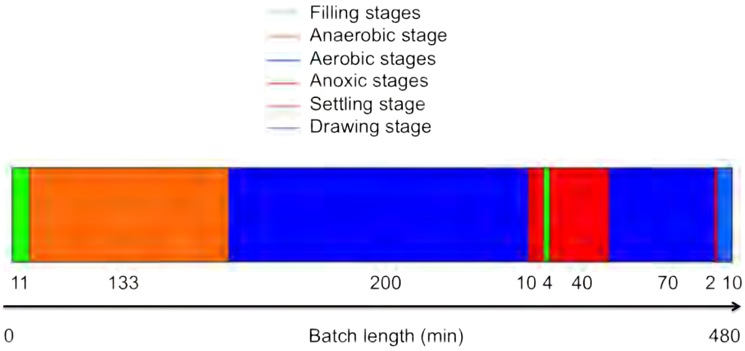


Figure 4.1: Stage configuration of the WWTP lab scale process. Each colour differentiates one of the different actions occurring during the batch. The value below each stage indicates its duration, which add up to 480 minutes (the batch length).

granules is impossible to the naked eye. According to the physical parameters of the sludge, batches between September 24th 2009 and March 24th 2010 were divided into three classes: floccular (1); transition (2); and granular (3). Figure 4.2 presents the mean size of the activated sludge and the classification obtained during the operation and according to experimental observations.

The experimental observations determined that, at least until December 19th, the system was fully floccular (class 1 and 249 batches). This statement was supported as well by a mean size of flocs lower than 200 μm. When the mean size of the sludge rose up to values over 500 m on February 26th, the sludge was classified as granular (class 3 and 184 batches) as depicted in Figure 4.2. However, the transition between floccular and granular state is not instantaneous and it takes several days (more than two months in this experiment and 87 batches), so we assigned "class 2" to these batches in order to represent this transition state. Note that the granule morphology was investigated using a stereomicroscope and to determine the size distribution of the particles in each SBR, 30 mL of mixed liquor were passed through a Beckman-Coulter® laser light scattering instrument, suitable for the measurement of particle sizes in the range of 0.02-2000 μm.

Since the main objective is to determine the granulation state of the process (its operating point) without considering the process state, both NOC and AOC observations will be included when building the respective monitoring models. This means that the

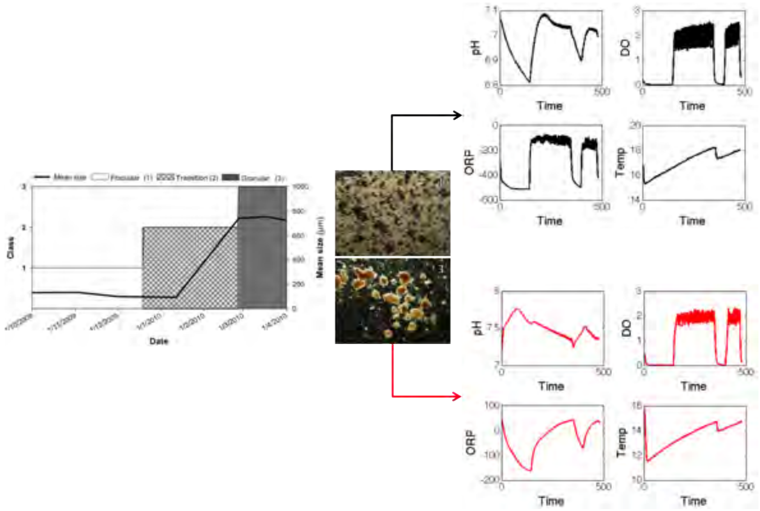


Figure 4.2: Experimental classification and mean size of the sludge (left); and stereomicroscope images from floccular (1) and granular (3) sludge (right) from the SBR

resulting models are not adequate for process monitoring, since they contain all operating conditions within a granulation state. This is a major difference with the traditional MPCA procedure, which uses only NOC observations to define the monitoring model.

4.3 Off-line determination of granulation state

To differentiate between floccular and granular batches, only experimentally well-classified batches will be used. This means that batches within the transition period (class 2) are initially discarded for this stage. Two different strategies will be applied to build the MPCA model: 1) use all available information of both states ($MPCAfg$); or 2) use only one state (either floccular ($MPCAf$) or granular ($MPCAg$)). The former (all information available) represents both states (floccular and granular) in the principal component subspace. Since the variability between the two states (inter-class variability) is greater than the variability within a state (intra-class variability), separability between both states is attained in the projection subspace (based on the values of the scores and T^2), rather than in the residual subspace.

The second approach (only one process state to build the model) intends to capture correlation among variables when the process operates in the floccular (or granular) state and study if this correlation suffers variations large enough to be detected. The idea is to project new batches against this model and observe evolution of both statistics, T^2 and SPE, in order to detect if this correlation change exists and how abrupt it is.

Independently of the strategy, the number of principal components for each model built will be selected based on the scree-plot criterion (Himes et al., 1994). Additionally, and since the granulation is considered once the charge is released (at the end of the batch), observations will be batch-wise unfolded and auto-scaled in order to give the same weight to all variable and samples and remove the average trajectory of all variables. Details in the different models are contained in their respective subsections.

4.3.1 Granularity diagnosis with a multiclass model (*CBDfg*)

As stated before, the multiclass model strategy consists in using all the available batches when building the model and looking for separated distributions in the projection space where different classes could be assigned. This means that MPCA will model the differences between both granulation states instead of the normal operating conditions (as was described in chapter 2). And in order to test with all available observations, floccular and granular batches will be divided in five folds ($n = 5$) using the n-fold cross-validation procedure in chapter 3, which gives us the five different statistical models detailed in Table 4.1, where Fold indicates the set of observations used to build the i^{th} MPCA model, r indicates the number of principal components retained for each model, PVE is the percentage variance explained by the MPCA model, δ_α is the SPE limit and τ_α is the T^2 limit. Both δ_α and τ_α are computed as the percentile 95 of the respective indices, since NOC and AOC observation for each granulation state are included in the model.

As can be seen in Table 4.1, all five models presents different number of principal components, due to the presence of extreme observations. Usually, these observations should be removed, however, in this case, they will be useful when classifying clusters of “noisy” instances within each group (as will be depicted later on in Figure 4.5). Moreover, the fact that both NOC and AOC observations coexist within the model gives rise to greater SPE values than T^2, as can observed by the respective statistical limits.

Table 4.1: Properties of the statistical model of each fold (multi-class approach)

	Fold	r	PVE (%)	δ_α	τ_α
MPCAfg	1	12	91.8116	356.173	23.2873
	2	9	89.3931	467.711	19.8017
	3	11	91.0385	378.379	22.5762
	4	13	92.0008	321.502	28.547
	5	6	81.6925	822.211	16.5421

This means, that differences between floccular and granular observations are more easily found (due to a lower variance) within the projection subspace (as intended) rather than in the residual subspace. Figure 4.3 presents the projection into the first three principal components of all floccular (black dots) and granular batches (red triangles) from the first fold to illustrate this fact.

As can be seen in Figure 4.3, the first and second principal components have the highest discrimination power to differentiate between floccular and granular observations as two main groups can be distinguished within the t_1-t_2 plot, as well as in the 3D representation, while some batches were coincident with the rest of the 2D plots. Consequently, cases (c_{fg}) have been characterised by a labelled vector consisting of the score vector $(t_1, t_2, \cdots, t_r)$ and the granularity class (class 1 (floccular) or class 3 (granular)) expressed as:

$$c_{fg} = \{t_1, t_2, \cdots, t_r, class\} \tag{4.1}$$

The retrieval process for this case base (*CBDfg*) will compare cases using the normalised Euclidean score distance (d_t) explained in 3.3.2.1 and previously presented in (3.9) in this way:

$$d_t\left(c_a, c_b\right) = \sqrt{\sum_{i=1}^{r} \frac{\left(t_{i,a} - t_{i,b}\right)^2}{\lambda_i}} \tag{4.2}$$

where c_a is the new case and c_b is one of the cases retrieved from the case base; $t_{i,a}$ and $t_{i,b}$ are the i^{th} score of case c_a and c_b respectively; and λ_i is the eigenvalue associated to the i^{th} principal component.

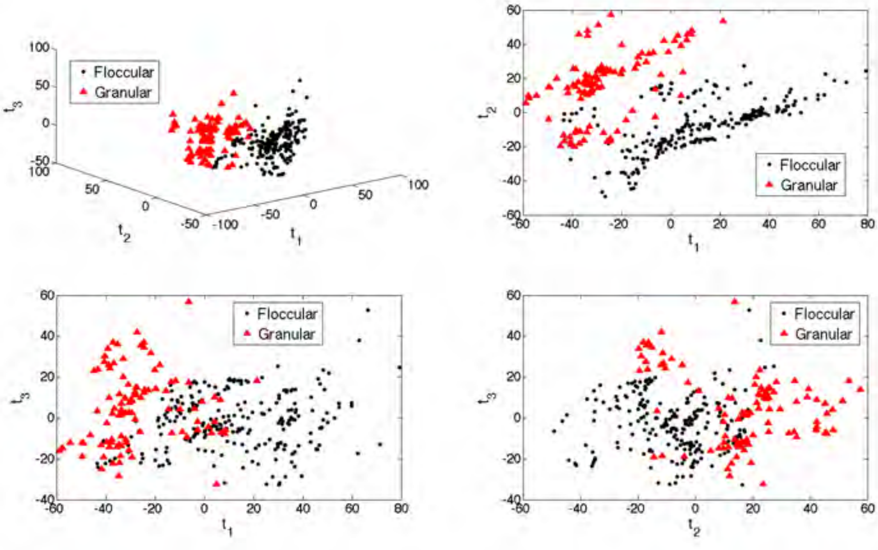


Figure 4.3: Projection into the first three principal components of all floccular (black dots) and granular batches (red triangles)

4.3.2 Granularity differentiation with single class models (*CBDg* and *CBDf*)

As two process states are available, two different single-class MPCA models can be built: one that focuses on floccular batches ($MPCAf$); and another based on granular observations ($MPCAg$). This strategy consists in identifying inconsistencies between actual data and one of he previously computed models. Based on this, the retrieval process is based on the SPE index. Likewise in the previous section, the n-fold cross-validation procedure has been used to build five ($n = 5$) different MPCA models (using the same divisions for all strategies) and presented in Table 4.2, which uses the same notation than Table 4.1. Note that when building the i^{th} $MPCAf$ model, only floccular observations within this fold were used, and vice-versa, when building the i^{th} $MPCAg$ model, only granular batches within this fold were used.

Like it happened in with *MPCAfg*, each statistical model presents different number of retained principal components, especially, *MPCAf*. Moreover, the fact that the first MPCA model based on floccular cases ($MPCAf$) presents the lowest number of principal components retained, while at the same time, the granular model of the first

Table 4.2: Properties of the statistical model of each fold (single-class approach)

	Fold	r	PVE (%)	δ_α	τ_α
MPCAf	1	8	87.593	496.044	18.2729
	2	7	86.7722	529.729	20.1593
	3	9	89.4041	502.494	22.9798
	4	10	90.2813	430.77	20.8701
	5	9	87.5078	509.824	20.5577
MPCAg	1	11	94.0477	296.457	34.6863
	2	8	90.6597	522.052	20.9217
	3	9	91.9452	427.209	22.0003
	4	9	91.9073	460.794	26.2885
	5	9	90.4178	532.782	25.5361

fold (*MPCAg*) presents the greatest value indicates that all extreme observations are related to granular batches, and there are no extreme observations related to floccular batches, or at least, they do not alter the model in a great measure. Aside from this first fold, *MPCAg* presents a lower variation in both the number of principal components and PVE, which means that its intra-class variability is lower to the one found among floccular batches as depicted in Figure 4.4, which shows the SPE values of all batches when projected into *MPCAf* (Figure 4.4A) and into *MPCAg* (Figure 4.4B). Note that floccular batches are representd using black dots, while granular batches are represented using red triangles.

As can be observed from both distributions in Figure 4.4, granular batches present a lower intra-class variance (red triangles in Figure 4.4B were more cluttered than black dots in Figure 4.4A), and thus, the separation using *CBDg* is expected to have a better classification ratio. However, note that in both cases, projecting batches of a different class results in all batches of this new class having a higher SPE value than the respective SPE limit. Therefore, cases for this strategy (c_s) are characterised as a tuple of its SPE value (SPE_i) and its granularity state (class 1 (floccular) or class 3

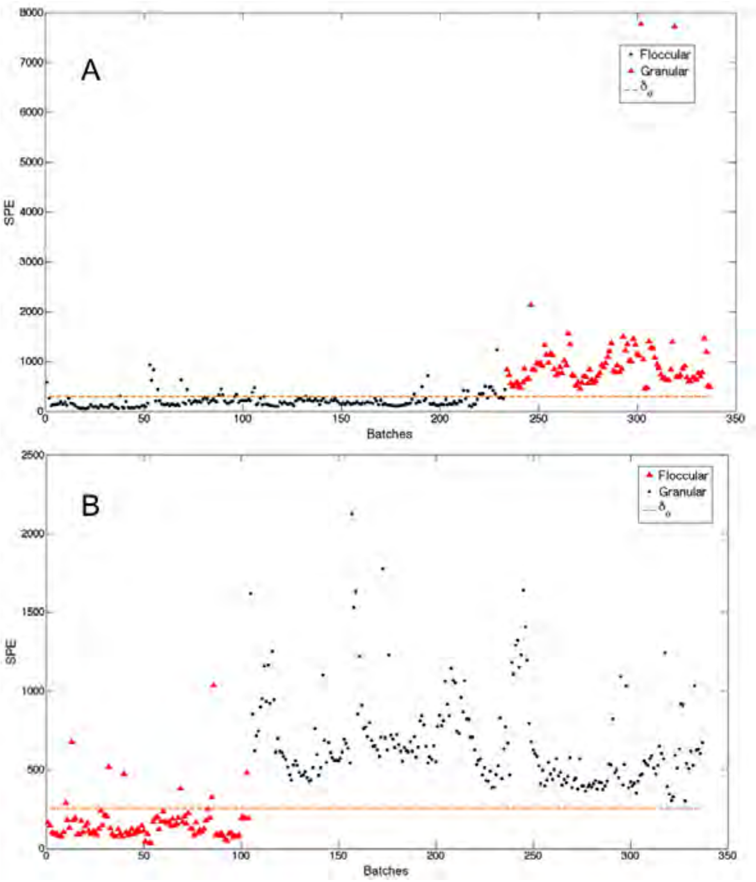


Figure 4.4: *SPE* values for all available batches when projected into a floccular PCA model (A) and when projected into a granular model (B)

(granular)) expressed as:

$$c_s = \{SPE_i, class\} \tag{4.3}$$

And thus, the retrieval process for this strategy will use the SPE criterion presented in 3.3.2.2 (3.10) in this way:

$$d_{SPE}\left(c_a, c_b\right) = \left|SPE_{c_a} - SPE_{c_b}\right| \tag{4.4}$$

where c_a is a new case and c_b is one of the cases retrieved from the case base; and SPE_{c_a} and SPE_{c_b} are SPE values for case c_a and c_b respectively. Finally, note that cases retrieved from *CBDf* will be noted as c_f, while batches retrieved from *CBDg* will be referred as c_g.

4.3.3 Reusing and revising the retrieved information

Once the nearest neighbours have been retrieved (independently of the strategy used), their information is reused to propose the granulation of a new observation. As could be seen in Figure 4.3 and Figure 4.4, there are noisy instances for both states, and thus, both distance-weighted (DW) (firstly introduced in (3.24)) and distance and class frequency weighted reuse (DFW) explained in subsections 3.4.2 (3.24) and 3.4.3 (3.25) will be compared when reusing information.

Regarding the performance evaluation of each strategy, an adapted confusion matrix (shown in in Table 4.3) will be used in order to express the classification rate based on both granulation states.

In Table 4.3 FF indicates floccular batches correctly predicted as floccular; FG stands for floccular batches classified as granular; GF refers granular batches misclassified as floccular; and GG labels granular batches correctly classified as such. NF is the total number of floccular batches, while NG indicates the number of granular batches available. And based on this new notation, two new performance indices can be derived: the floccular ratio (FR) and the granular ratio (GR). The former indicates the ratio of floccular batches correctly identified, while the latter accounts for the ratio

Table 4.3: Confusion Matrix elements within the WWTP field

		Real Class	
		Floccular	**Granular**
Predicted Class	**Floccular**	FF	FG
	Granular	GF	GG
	TOTAL	NF	NG

of granular batches correctly classified. These new indices can be computed using the following expressions:

$$FR = \frac{FF}{FF + FG} = \frac{FF}{NF} \tag{4.5}$$

$$GR = \frac{GG}{GF + GG} = \frac{GG}{NG} \tag{4.6}$$

Note that the closest to 1 these indices are, the better the classifier is in terms of distinguish between floccular and granular observations. The three retain strategies explained in section 3.6 (IB3 (3.4), DROP4 (3.3) and IBUdG (3.5)) will be used to find the method that preserves the best case base in terms of AUC for discerning between floccular and granular batches.

4.3.4 Results and discussion

This subsection presents the different results obtained from the off-line application of the methodology. There are two objectives for this off-line analysis: 1) identify the best configuration to differentiate between both granulation states; and 2) find the variable(s) and stage(s) for which this difference is most observable.

Table 4.4 presents the performance of the three statistical models (subsections 4.3.1 and 4.3.2) and CBD configurations (4.3.3) for the off-line analysis. All batches whose granulation state (class) is known are divided into five divisions using the n-fold cross-validation procedure explained in subsection 3.5.1. **FF**, **FG**, **GF**, **GG**, **NF** and **NG** were introduced in the previous subsection and are related to the different category labels in the adapted confusion matrix. Finally, **FR**, **GR** and **ACC** are related to the performance indices floccular ratio (4.5), granular ratio (4.6) and accuracy (3.26) previously described. **AUCreuse** computes the area under the ROC curve (AUC) using the **FR** (as TPR) and **GR** (as FPR) of the three retain policies used for each reuse procedure in every modelling strategy, while **AUCall** compute the AUC using the six **FR** (as TPR) and **GR** (as FPR) for a given strategy. Finally, numbers in **bold** indicate the best performance for each **DW** and **DWF** boxes for each CBD strategy. Remember that the retrieve procedure for ***CBDfg*** is based on d_t (3.9), while retrieving cases from either ***CBDf*** or ***CBDg*** uses d_{SPE} (3.10).

With respect to floccular (**FR**) and granular (**GR**) ratio performance in Table 4.4, **FR** is consistently higher than **GR**. On the one hand, there are more floccular observations than granular batches (**NF** > **NG**), and thus, misclassifying an observation has a lower influence for floccular batches. On the other hand, there are more noisy instances within granular batches than for the floccular ones. This means that noisy granular observation have a lower influence over floccular batches because of their number. At the same time, these granular noisy instances, when misclassified have a higher toll to **GR**, which worsens due to the presence of floccular noisy instances. Figure 4.5 presents the projection of all floccular and granular cases in one of the ***CBDfg*** folds into the two first principal components. Floccular batches are represented using black dots, while the granular ones are depicted using red triangles. A hypothetical class division between floccular and granular batches is represented using a green line, while floccular noisy instances are circled in blue and granular noisy cases are marked with an orange circle. Note that for both classes, noisy instances are clustered together, which influences the class prediction as will be commented later on.

This noisy instances also affect the results obtained when using **DW** and **DFW**. Because noise instances are grouped, **DFW** amplifies the influence of these noisy instances to the frontier observations (of a different class). Consequently, **AUCreuse** values under the **DFW** reuse procedure are always lower than the ones obtained with **DW**. Among the three modelling strategies, ***CBDfg*** is the most affected, since the noisy cases for each class have to be also modelled in the PCS, while single class MPCA models (used by ***CBDf*** and ***CBDg***) only have to model the ones included in their class (the others only appear in the case base). Consequently, values for those

Table 4.4: Off-line analysis results in the WWTP field

	Reuse	**Retain**	**FF**	**FG**	**GF**	**GG**	**NF**	**NG**	**FR**	**GR**	**ACC**	**AUCreuse**	**AUCall**
CBRfg	**DW**	IB3	230	19	17	167	249	184	0.9237	**0.9076**	0.9169	0.9204	0.9056
		DROP4	233	16	23	161	249	184	0.9357	0.8750	0.9099		
		IBUdG	239	10	18	166	249	184	**0.9598**	0.9022	**0.9353**		
	DFW	IB3	225	24	26	158	249	184	0.9036	0.8587	0.8845	0.9016	
		DROP4	230	19	20	164	249	184	0.9237	0.8913	0.9099		
		IBUdG	238	11	19	165	249	184	**0.9558**	**0.8967**	**0.9307**		
CBRf	**DW**	IB3	229	20	17	167	249	184	0.9197	**0.9076**	0.9145	0.9508	0.9359
		DROP4	234	15	20	164	249	184	0.9398	0.8913	0.9192		
		IBUdG	239	10	18	166	249	184	**0.9598**	0.9022	**0.9353**		
	DFW	IB3	222	27	28	156	249	184	0.8916	0.8478	0.8730	0.9349	
		DROP4	230	19	17	167	249	184	0.9237	**0.9076**	0.9169		
		IBUdG	237	12	21	163	249	184	**0.9518**	0.8859	**0.9238**		
CBRg	**DW**	IB3	229	20	15	169	249	184	0.9197	**0.9185**	0.9192	**0.9525**	**0.9525**
		DROP4	232	17	17	167	249	184	0.9317	0.9076	0.9215		
		IBUdG	239	10	18	166	249	184	**0.9598**	0.9022	**0.9353**		
	DFW	IB3	221	28	27	157	249	184	0.8876	0.8533	0.8730	0.9522	
		DROP4	229	20	15	169	249	184	0.9197	**0.9185**	0.9192		
		IBUdG	237	12	19	165	249	184	**0.9518**	0.8967	**0.9284**		

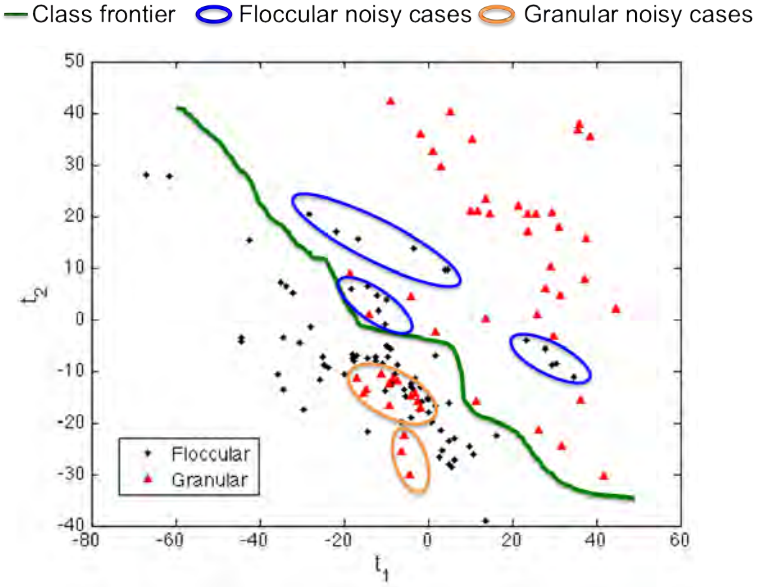


Figure 4.5: Projection into the two first principal components of all floccular and granular batches in one fold of ***CBDfg***

two approaches are higher, especially for ***CBDg***, which is the strategy with the highest performance. Finally, this differences are magnified if all six configuration per modelling strategies (two reuse procedures and three retain methods) to compute the AUC (AUCall), indicating that ***CBDg*** is the most regular strategy.

Finally, regarding the three retain policies, results vary based on the strategy used. IBUdG is the best retain policy for ***CBDfg*** in terms of **FR** and **ACC** (independently of the reuse procedure used), while IB3 (followed by IBUdG) and IBUdG are the best methods in terms of **GR**. With respect to the single class MPCA models (***CBDf*** and ***CBDg***), IBUdG is the best approach again in terms of **FR** and **ACC**, while IB3 is the best for **DW** and DROP4 is the best for **DFW**. And in terms of reduction carried out by each method, the average results obtained with each strategy, reuse and revise procedure are presented in Table 4.5. **NFi** and **NGi** indicate the average values of floccular and granular among the five folds used to build the initial case base. **NFo** and **NGo** refer to the average number of floccular and granular cases after reducing the initial case base using the different retain policies (remember that the retain function uses the same retrieve and reuse procedure of the case base). **NFe** and **NGe** is the average number of floccular and granular cases after revising the proposed solutions for each test set (in case of a misclassified observation, the faulty case is added to the case base, and the maintenance strategy is applied in order to obtain the minimum case base). **NFi vs NFo** and **NGi vs NGo** indicate the percentage of reduction for each case base between the initial case base and the reduced case base. Finally, **NFo vs NFe** and **NGo vs NGe** indicate the percentage of reduction for each case base between the reduced case base and the final case base (once test observations have been revised).

Independently of the strategy and reuse procedure, IB3 is the procedure that reduces the most the case base in both, percentage and number of cases, followed by DROP4 for floccular and granular observations and observables in **NFi vs NFo**, **NGi vs NGo** (first reduction) and in **NFo vs NFe** and **NGo vs NGe** (second reduction). IBUdG keeps the k-nearest "enemies" (case with a different case and close to the current case) for each case in the case base. For the case being, the overlapping region between both operating points is significant (as could be seen in Figure 4.3 for ***CBDfg*** and Figure 4.4 for ***CBDf*** and ***CBDg***), and thus, nearest enemies for each case in the case base are sparse not allowing a significant reduction for this maintenance policy. Additionally, IBUdG is the method that presents the lowest reduction in **NFo vs NFe** and **NGo vs NGe**, which means that the test set requires most of the space covered by the initial reduced case base. Both IB3 and DROP4 present a similar percentage of reduction from the initial reduced case base to the final case base, which means that the first reduced data set covered a much broad space than the required for the case base. And with respect the reduction of cases bases as a function of the reuse procedure used, there are no significant differences when using either IB3 or DROP4. However, this is not the case for IBUdG when using $CBDf$, which presents much less cases when using DFW than when DW. This means that for this case, the nearest "enemies" are more clogged, and thus, the method can greatly reduce the case base.

In order to improve these results, and because $CBRg$ was the best classifier, the contributions to the SPE index (c^{SPE}, which evaluates the projection into the residual subspace) were studied in order to obtain what were the variables and stages that contributed to better differentiate between floccular and granular batches.

Table 4.5: Comparison on the reduction capabilities for the off-line application in WWTP

	Reuse	**Retain**	**NFi**	**NGi**	**NFo**	**NGo**	**NFe**	**NGe**	**NFi vs NFo**	**NGi vs NGo**	**NFo vs NFe**	**NGo vs NGe**
CBRfg	**DW**	IB3	199.2	147.2	21.8	21	8.6	8.6	89.0562	85.7337	60.5505	59.0476
		DROP4	199.2	147.2	26.8	24.2	18	18.6	86.5462	83.5598	32.8358	23.1405
		IBUdG	199.2	147.2	91.2	65.2	75	56	54.2169	55.7065	17.7632	14.1104
	DFW	IB3	199.2	147.2	21.6	19.8	8	9.8	89.1566	86.5489	62.9630	50.5051
		DROP4	199.2	147.2	27.6	25.2	19.2	20.4	86.1446	82.8804	30.4348	19.0476
		IBUdG	199.2	147.2	91.2	65.2	75	56	54.2169	55.7065	17.7632	14.1104
CBRf	**DW**	IB3	199.2	147.2	20.2	16.2	8.4	6	89.8594	88.9946	58.4158	62.9630
		DROP4	199.2	147.2	21.6	17.4	16.4	12.8	89.1566	88.1793	24.0741	26.4368
		IBUdG	199.2	147.2	91.2	65.2	75	56	54.2169	55.7065	17.7632	14.1104
	DFW	IB3	199.2	147.2	20.2	15.4	8.6	5.6	89.8594	89.5380	57.4257	63.6364
		DROP4	199.2	147.2	23	17.8	17.2	13	88.4538	87.9076	25.2174	26.9663
		IBUdG	199.2	147.2	56	34.6	48.6	33.6	71.8876	76.4946	13.2143	2.8902
CBRg	**DW**	IB3	199.2	147.2	17.6	17.2	6.8	7.6	91.1647	88.3152	61.3636	55.8140
		DROP4	199.2	147.2	19.2	19	13.4	13	90.3614	87.0924	30.2083	31.5789
		IBUdG	199.2	147.2	73.4	49.8	60	45.2	63.1526	66.1685	18.2561	9.2369
	DFW	IB3	199.2	147.2	18	17.2	6.2	8.2	90.9639	88.3152	65.5556	52.3256
		DROP4	199.2	147.2	19.2	19.4	14	12.8	90.3614	86.8207	27.0833	34.0206
		IBUdG	199.2	147.2	73.4	49.8	60	45	63.1526	66.1685	18.2561	9.6386

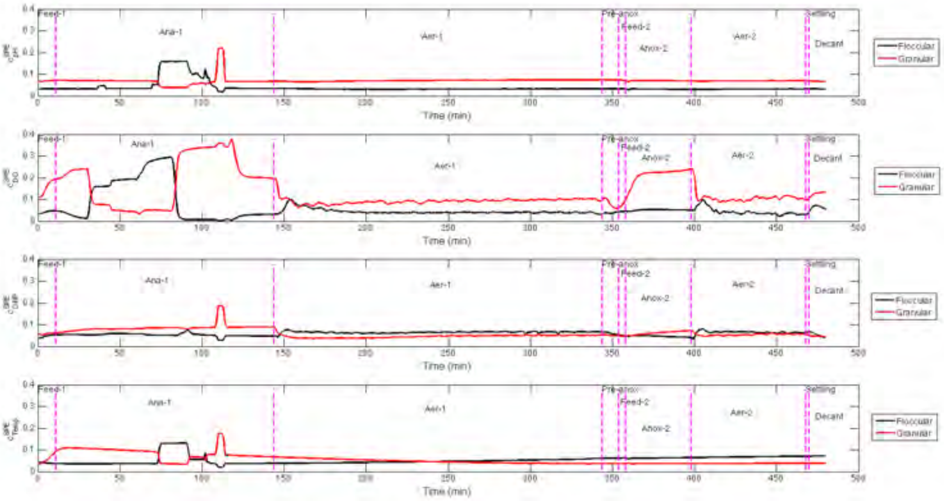


Figure 4.6: Mean SPE contribution for floccular (black) and granular (red) for each on-line variable. Each vertical line indicates the phase division along a cycle

Table 4.6: Properties of the statistical model built using all granular batches

	r	PVE (%)	δ_α	τ_α
$MPCAgAll$	9	91.1391	433.546	20.9828

Figure 4.6 presents the mean SPE contribution for all floccular batches (continuous black line) and all granular batches (continuous red line) when projected into a granular focused MPCA model built using all available granular observations ($MPCAgAll$). The information associated with this new model is presented in Table 4.6 using the same notation than Table 4.1 and Table 4.2. This figure shows the contribution of each on-line variable (pH, O2, ORP and temperature) along time. It also takes into account the mean contribution in each phase of the cycle (vertical dashed lines).

Granular sludge has the ability to diffuse nutrients and compounds such as oxygen and protons inside the particles. Because of that, variations in on-line parameters were expected to be smoother in granular rather than floccular sludge. This fact was reflected in the pH sensor, as it presented the most significant differences between

Table 4.7: Results obtained with the CBD only focused on the anaerobic stage

	FF	FG	GF	GG	NF	NG	FR	GR	ACC
CBDfgAna	243	6	16	168	249	184	0.9759	0.9130	0.9492

floccular and granular batches, i.e. the contributions of granular batches being lower than floccular batches. Additionally, the first anaerobic stage (Ana-1) presented more important differences in all variables compared with the other phases along the cycle. Based on this, only the information related to this stage was retained for all floccular and granular batches to build a new multi-class PCA model and produce a new CBR ($CBRfgAna$). Results obtained with this new configuration are presented in Table 4.7, which uses the same notation than Table 4.4.

Focusing only on the anaerobic phase of the cycle, both floccular and granular classification rates increased when comparing the results obtained using the whole batch (***CBRg***), for all three performance indices: **ACC** = 0.9492, **FR** = 0.9759 and **GR** = 0.9130. This new approach only misclassified 6 floccular batches and 16 granular batches, thus improving the prediction of the model.

4.3.5 Identification of the transition state from floccular to granular

Based on the best classification procedure ($CBRgAna$), batches between December 20^{th} 2009 and February 11^{th} 2010, which belong to the transition state of the reactor (class 2), are projected into the multi-class model to date the transition from floccular to granular. For the evaluation of class 2 batches, all available batches labelled as floccular (class 1) and granular (class 3) are used to build the multi-class model (without using the n-fold cross-validation procedure). Results obtained with these batches are presented in Figure 4.7, where the green stem representation indicates the class predicted for each batch in this period of time. The continuous blue line indicates the mean size of particles found in off-line analysis along the available dates within this period. Note that between measurements, granules mean size is assumed to not significantly vary, and therefore, it is a continuous line.

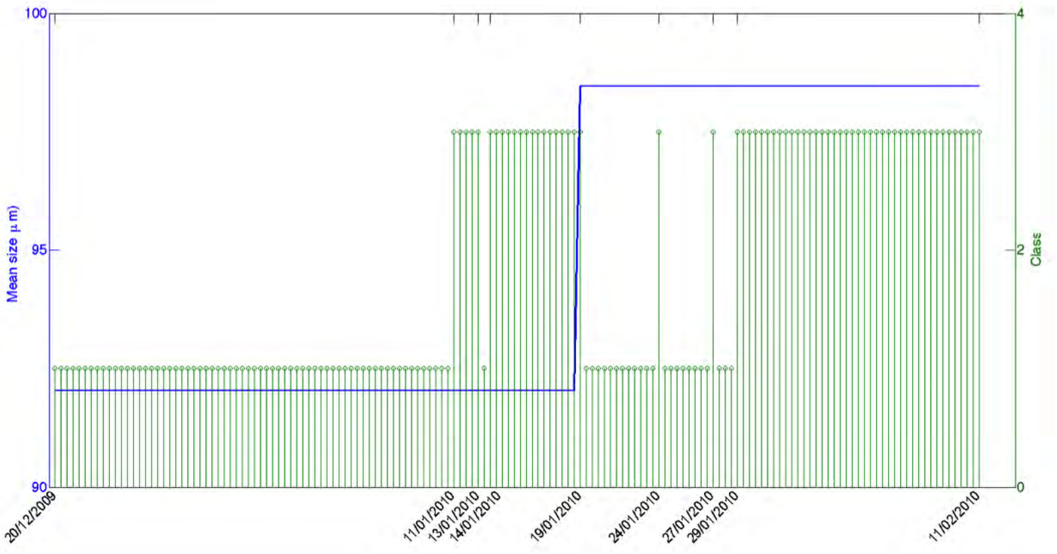


Figure 4.7: Predicted class (green stem representation) versus mean particle size (continuous blue line) for batches in the transition state

As can be seen in Figure 4.7, there were two main transitions from a floccular to a granular state. The first one occurred on January 11^{th} 2010 and lasted until January 19^{th} 2010 (excluding a batch on December 13^{th} classified as floccular). The second transition happened on January 29 th 2010 and lasted for the rest of the batches (again omitting two batches classified as granular on January 24^{th} and 27^{th} 2010). This evaluation did not completely remove the classification of transition state, but it was reduced from 53 to 10 days in which alternation of granular and floccular batches were obtained. This fact was probably due to the formation of small particles that while hardly influencing the performance of the process and the on-line variables, are the starting point for granulation.

4.4 On-line application for granulation prediction and case base evolution

The previous section presented an ideal case where all final classes were initially available (defined classes). However, this is not the most common situation when monitoring a process. New faults appear while monitoring, and these situations have to be taken into account for future events. Based on the learning capabilities of the CBR, this section presents a scenario where a new situation appears, in this case a granulation change.

More specifically, only batches prior to the granulation change are considered to build the initial case base (class 1). Consequently, only information of defined floccular batches was available, and thus, the resulting statistical model only characterises a subspace of the $MPCAfg$ model described in the previous section. Thus, observations of a new class (in this case granular batches) are eager to lie within the statistical limits (SPE and T^2). Therefore, the CBR is proposed as an alternative to the usage of the SPE and T^2 limits (when both classes are available). As a result, cases (c_{diag}) were characterised as a label vector with its SPE (SPE_{c_i}) and T^2 values ($T^2_{c_i}$), with the addition of the granularity state (class 1 (floccular) or class 3 (granular)) expressed as:

$$c_{diag} = \{SPE_{c_i}, T^2_{c_i}, class\} \tag{4.7}$$

The retrieval process will be based on the normalised Euclidean distance d_ϕ explained in 3.3.2.4 and firstly introduced in (3.16) between the SPE values of the new case (SPE_{c_a}) and a retrieved observation (SPE_{c_b}) and their T^2 values ($T^2_{c_a}$ and $T^2_{c_b}$ respectively) in this way:

$$d_\phi = (c_a, c_b) = \left| \frac{SPE_{c_a} - SPE_{c_b}}{\delta_\alpha} \right| + \left| \frac{T^2_{c_a} - T^2_{c_b}}{\tau_\alpha} \right| \tag{4.8}$$

where δ_α and τ_α are the SPE and T^2 statistical limits respectively for a given confidence level α. In this case, both values were fixed as the percentile 95 of the observations used to build the MPCA model.

The reuse procedure was based on the distance-weighted method explained in subsection 3.4.2 (3.23) to compute the similarity to both the floccular (w_f) and granular (w_g) states in this way:

Table 4.8: Classification performance and information related to the MPCA model for the on-line granulation prediction

	r	PVE (%)	FF	FG	GF	GG	NF	NG	FR	GR	ACC
CBDdiag	4	78.17	106	18	23	81	124	104	0.85	0.78	0.82

$$w_f = \sum_{j=1}^{n_f} \frac{1}{d_{\phi_j}} \tag{4.9}$$

$$w_g = \sum_{j=1}^{n_g} \frac{1}{d_{\phi_j}} \tag{4.10}$$

where n_f and n_g are respectively the number of retrieved floccular and granular observations among the k-nearest neighbours; and d_{ϕ_j} is the j^th retrieved floccular or granular observation.

Regarding the revise procedure for this scenario, again the floccular and granular ratio (FR (4.5) and GR (4.6) respectively) and the overall accuracy (ACC (3.26)) are used. Note that batches will be shuffled in order to mimic the transition period of the process. Finally, whenever a batch is misclassified, it is added to the case base in order to help classifying new cases (retain step); and while there are no granular batches in the case base, any observation with either $SPE_{c_i} > \delta_\alpha$ or $T^2_{c_i} > \tau_\alpha$ will be labelled as granular. In case of a false alarm (a floccular observation labelled as granular), the observation is added to the case base to avoid future misclassifications, but the statistical model will not be updated. Whenever a granular batch is available in the case base, then the CBD will be used to diagnose the granulation of each discharged batch. Results obtained within this scenario are presented in Table 4.8, which uses the same notation that in Table 4.7. Note that observations used to build the initial case base were not tested, and thus, the number of floccular observations (NF) in this scenario is lower than in the previous one.

As can be seen, using only the information of floccular classes reduced the performance in comparison with when all information was used to monitor the process

($CBRfg$ and $CBRfgAna$). The main reason is that the initial MPCA model in this scenario only covers a subspace of the whole monitoring space (even from $MPCAf$). In order to solve this problem, the MPCA model could be update whenever a new floccular batch is available, and when enough observations from all classes are available, repeat the training procedure, ending with the same information presented in the previous section.

Despite the stability of granules being essential to maintain biological nutrient removal, most of these reactors suffer efficiency disruptions due to granules losing compactness and breaking into pieces or flocs. As has been demonstrated, the application of the CBD can also be used to detect transition periods from granular to floccular states with the model predicted state variability as on-line values are state dependent. Therefore, disruption from these systems could be detected faster in order to modify the operational conditions before losing nutrient removal. Thus, CBD can be used either as a tool for prediction or prevention.

In terms of practical applicability, the methodology for building both the statistical model and case base can be automated, which means that no specific knowledge about MPCA and CBR are needed to use such a tool. The only requirement is that the relation between sensor readings and typology/classes must be accurate, and thus, the operator must be able to assign accordingly the granulation state of the plant in this case. To simplify this task, a graphical representation of the typical behaviour of the process under the different states can be displayed within the tool to facilitate the task to the operator. This was the idea of the EMOLD project (COLL-CT-2006-030339), which consisted in provide learning capabilities to injection moulds by sensing, and share this knowledge with other network moulds. Results related with this project are presented in the next chapter, and were partially published in Berjaga et al. (2009a), Berjaga et al. (2009b) and Berjaga et al. (2009c).

5

Case-based diagnosis for quality Control of injected parts

This chapter presents an application of the CBD framework within the production stage of injected parts. The main idea is to first build the normal operating conditions of the injection process, and then identify the fault that originated the quality defect found on an injected part. This means that the CBD will relate the type of defect found on the piece (NOC, flashes, sink marks and locally oversized) with the problem with the injection parameters (optimal parameters, switch-over too early, switch-over without holding pressure, switch-over too late without holding pressure and switch-over too late with holding pressure) using only the information provided by the sensors installed in the mould cavity.

In order to do so, this chapter is divided as follows. Firstly, a brief introduction to the problem is carried out (section 5.1). Next, the experimental set-up within this field is detailed (section 5.2), followed by an off-line application of the methodology to predict the final quality of an injected part is detailed and discussed (section 5.3). Finally, a real-time release (when the batch ends) is explained and discussed in section 5.4.

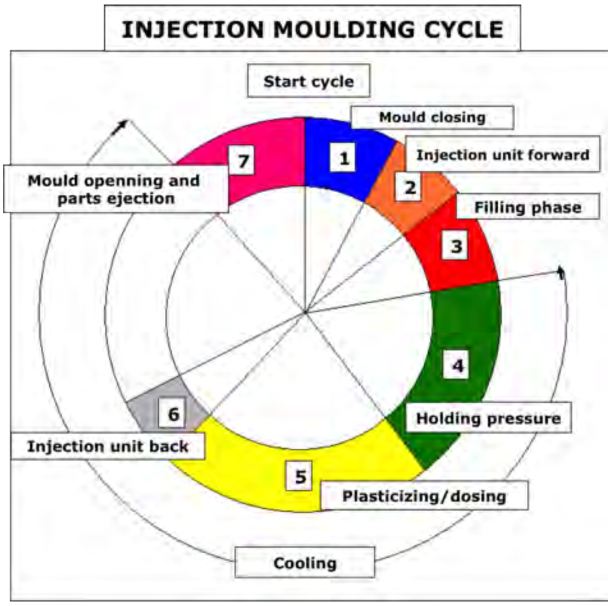


Figure 5.1: Complete injection cycle, including task division for each main phase.

5.1 Problem description

Injection moulding is the most commonly used process to produce plastic pieces due to its low cost, ability to produce complex shapes with good accuracy and short cycle times (Bozdana and Eyercioglu, 2002). It consists in a cyclic process (batch) that can be divided into four main phases: filling, packing, cooling and ejection. The first stage consists in filling the mould with hot polymer melt. In the packing stage, an additional quantity of polymer melt is packed into the mould at a higher pressure in order to compensate the shrinkage produced during solidification. Then, in the cooling stage, the temperature inside the mould is decreased to solidify its content. Finally, in the ejection stage, the mould opens, the product is ejected and the mould closes again until the beginning of the next cycle. Fig. 5.1 presents a more detailed view of the cycle, where the four main phases are divided in the respective tasks that happen in them.

Traditionally, quality control in the injection industry is based on sampling; this is, only some of the injected parts are inspected. The exhaustiveness of the process varies significantly depending on both, the duration of the cycle and the quality control policies. Sampling is a common practice in the manufacturing industry oriented to detect evolving faults and degradation in the process behaviour (wear, ageing, system overheating, etc.). Recently, the moulding industry started to incorporate sensors into the moulds offering the capability to know what is happening inside the mould during the injection. This chapter explores this fact in order to propose a quality control method to automatically evaluate any injected part using the CBD methodology presented in chapter 3.

So, data collected by sensors embedded in the mould, during normal executions of the batch (injection of pieces with good quality) are used to create a statistical model. A single injection is represented by a matrix with as many columns as number of sensors and as many rows as sample times. So, multiple registers (multiple injections or batches) can be organised into a three dimensional matrix (injections $\times$ sensors $\times$ samples) as the one depicted in Figure 2.5 and described by (2.15). This representation is the starting point to develop multiway statistical methods capable to obtain multivariate models representing dependencies among variables at different time instants during the injection. These models are suitable for on-line monitoring of batch process due to their high performance in fault detection, diagnosis or quality control.

Previous works in the literature explored fault detection and diagnosis strategies involving the principal component analysis (PCA) of injection parameters (machine parameters) like in Kazmer and Westerdale (2008). Principal component analysis was also proposed as a dimensionality reduction tool previous to feed a generalised regression neural network with the purpose of modelling dynamics and non-linearities in Guo et al. (2006). Benefits of PCA were also used in Liu and MacGregor (2005) as a preprocessing strategy for dimensionality reduction of texture images obtained from a computer vision system for inspection of injected parts.

Instead of using the injection parameters (like in Kazmer and Westerdale (2008)), the CBD uses the trajectories of the measured variables to build an automatic quality control of the injection process. This makes possible to identify defective pieces in an early stage of the production and also to correct any misbehaviour before the next injection starts. Consequently, losses due to defective parts are significantly reduced. Additionally, such monitoring strategy enables the traceability of all injected parts, adding a signature to every piece just after opening the mould.

This chapter extends the works in Berjaga et al. (2009a) (MPCA and k-means) and Berjaga et al. (2009b) (MPCA and nearest neighbour (NN)), which reported a good performance when identifying different defects on injected parts, and compares the application of the CBD with respect a combination of multiway partial least squares and discriminant analysis (MPLS-DA) . For further details on MPLS-DA, please refer to Appendix A.

5.2 Experimental set-up

The injection mould used to test the methodology is a prototype equipped with nine sensors installed in the mould cavity. These sensors measure temperatures and pressures at different locations in the two sides of the mould and the speed and position of the screw. The main idea is to estimate the quality of the injected parts at the end of the injection cycle using the continuous information registered by the sensors during the injection cycle. Quality of pieces has been divided in four categories according to the most common defects (short shots, sink marks, flashes and locally oversized injections) in the injection process. In order to test the performance of the method, defects have been generated artificially using appropriate injection parameters (see Table 5.1) that induce those defects in the injected pieces. Five different fault root causes were identified that produced these defects, which resulted in the following data sets:

- NOC: 54 injections with optimal parameters (normal operating conditions).
- $\vec{\xi}_1$: 51 injections with switch-over too early. This type of set-up produces short shot injected parts.
- $\vec{\xi}_2$: 52 injections without holding pressure. This configuration produces parts with sink marks.
- $\vec{\xi}_3$: 51 injections with switch-over too late without holding pressure, which causes injections to present flashes and/or sink marks.

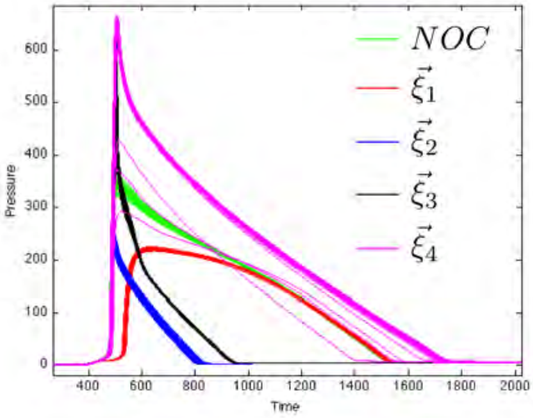


Figure 5.2: Pressure curve for the five typologies: class 1 (normal operating conditions), class 2 (switch-over too early), class 3 (no holding pressure), class 4 (switch-over too late without holding pressure) and class 5 (switch-over too late with holding pressure)

- $\vec{\xi}_4$: 53 injections with switch-over too late with holding pressure. This last configuration provokes locally oversized injected parts.

The switch-over is the injection instant when the injection pressure is increased to compensate the shrinkage of the material due to the cooling effect produced by the lower temperature of the mould. Fig. 5.2 shows the trajectories of one of the monitored pressures for the experiments described before and Table 5.1 details the configuration parameters for the respective data sets collected during October 8^th 2009.

5.3 Off-line quality control of injected parts

This section presents the results obtained when applying CBD and MPLS-DA to automatically determine the quality of injected parts off-line. Firstly, the CBD configuration for this analysis is detailed in the next subsection (MPLS-DA only requires to unfold the three-dimensional data and auto-scale the resulting two-dimensional matrix). Then, both MPCA and MPLS statistical models are compared (subsection 5.3.2); and finally, results for both methodologies are compared and discussed in subsection 5.3.3.

Table 5.1: Configuration parameters for the five data sets.

	Class				
	NOC	$\vec{\xi}_1$	$\vec{\xi}_2$	$\vec{\xi}_3$	$\vec{\xi}_4$
Injection stroke (ccm)	12.5	9.7	12.5	13.3	13.3
Holding pressure (bar)	400	400	400	0	400
Holding pressure time (s)	4	4	0	0	4
Mould temperature (°C)	50	50	50	50	50
Cylinder temperature (°C)	250	250	250	250	250
Screw diameter (mm)	22	22	22	22	22

5.3.1 Final quality prediction using a CBD

In order to differentiate between the different quality degrees, only normal operating conditions (or fault-free observations) are used to build the statistical model. Then, the other faulty observations (bad final quality) are projected into this model and are stored in the case base as examples to predict the final quality of new injections. Fault typologies are expected to present a low intra-class variability (observations within a cluster present small differences among them), and a high inter-class variability (observations of two different clusters behave in a completely different way).

When projecting faulty observations into a NOC-based MPCA model, the SPE index usually presents a higher value than fault-free observations. However, for the case being, this differences were also observable in the PCS (through the score values) as depicted in Figure 5.3, where the projection into the first three principal components of each class are represented using different colours.

Based on this principle, the normalised score distance explained in subsection 3.3.2.1, and detailed in (3.9), is used to retrieve the nearest neighbour ($k = 1$) for a new observation, whose class became the predicted quality for the new observation, and thus, no reuse procedure is needed. Finally, given the large distance between the different quality clusters, DROP4 was used to minimise the case base.

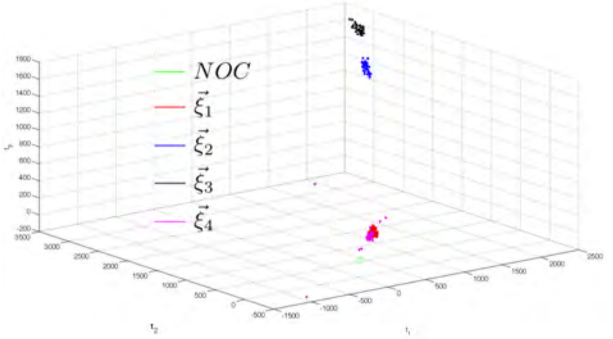


Figure 5.3: Distribution of the five experiment sets in the principal component subspace (first three principal components)

5.3.2 Model building

The original data set has been divided in five subsets ($n = 5$) or folds using the n-fold cross-validation procedure explained in subsection 3.5.1. The MPCA models have been built using only observations (time series of observed variables during injections) of normal pieces (class 1) whereas for the MPLS models all the examples in the training folds are used as quality identifiers to fill the matrix Y. Observations have been autoscaled and cross-validation (Himes et al., 1994) is applied to determine the number of principal components to retain in the models. These results are presented in Table 5.2 where LVs and PCs refer respectively to the number of retained latent variables (for MPLS) and principal components (for MPCA), $PV(X)$ is the percentage of global variance explained for the MPCA/MPLS model over the predictor matrix X, $PV(Y)$ is the percentage explained of global variability for the predicted variable Y and $Fold$ indicates which fold was used to build the MPCA/MPLS model. Finally, N/A stands for not applicable, since MPCA does not take into account the predicted variable Y.

It can be seen that MPLS presents a higher compression rate. This is, it explains more information, in terms of variance, contained in the X matrix (65.54%) than the MPCA model (57%) using less latent variables (3 for MPLS and 5 for MPCA). However, it has to be taken into account that the predictor matrix X and the objectives of both methods are different:

Table 5.2: Relation of principal components (PCs) and latent variables (LVs) retained, as well as the percentages of variance explained for the predictor (X) and predicted (Y) variables for each fold

	Fold	LVs/PCs	PV(X)	PV(Y)
MPCA	1	5	51.07	N/A
	2	6	56.56	N/A
	3	6	56.72	N/A
	4	5	51.45	N/A
	5	6	57.00	N/A
MPLS	1	3	64.19	70.53
	2	3	64.61	71.46
	3	3	65.64	70.55
	4	3	64.70	71.13
	5	3	65.02	72.16

- MPLS uses data from the five experiments described in section 5.2, and then builds a regression model from the observed variables to predict the quality of observations. Consequently, the model maximises the inter-class variance and at the same time that minimises the intra-class variance.

- MPCA builds the statistical model using only normal operating conditions (class 1), and therefore, minimises the intra-class variance. As a result, the isolation has to be carried out by an external procedure, which in this case is attained using the CBD.

The distribution of the training set observations in the latent variable space described for the three first latent variables (MPLS) and principal components (MPCA) are shown respectively in Fig. 5.4 and Fig. 5.3. As can be observed, MPLS select latent variables in order to maximise the differentiation among classes, at the same time that clusters observations of the same class. On the other hand, MPCA focuses on clustering normal operating conditions, and differences with the rest of experiment

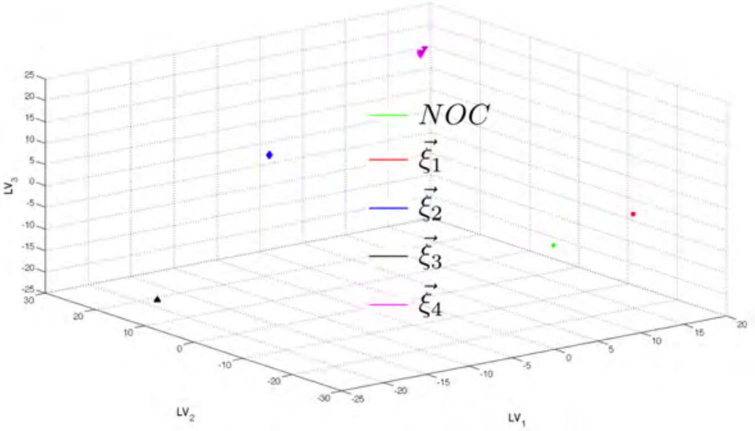


Figure 5.4: Distribution of the five experiment sets in the latent variable subspace (first three latent variables)

sets are due to differences in the correlation structure. All in all, at this stage the visual separability of classes of MPLS outperforms MPCA.

5.3.3 Performance of the quality control methods

The confusion matrix, extended to the five possible types of quality injection, has been used to assess the performance of MPCA/MPLS models. Average values obtained after applying the 5-fold cross-validation method have been used to quantify this performance. True positive rates (in %) are in the diagonal of the confusion matrix, whereas wrong injections classified as good are in the first column (excluding the first element) and wrong classifications of good injections correspond to the first row (with the exception of the first element). The same reasoning can be applied to results related to each typology of injection just addressing rows and columns associated to each class.

Table 5.3 presents this average classification performance over the cross-validation test-datasets for MPLS-DA and CBD respectively. Values from 1 to 5 identify the five injection classes previously defined. It can be observed that the first 4 typologies are perfectly classified for the CBD approach (the values in the diagonal are 100%) while MPLS-DA presents a 2% of wrong classification of class 2 injections that are being confused with normal injections (class 1). Class 5 injections also present some misclassifcations for both approaches being confused with class 1 and 2 for the MPLS-

Table 5.3: Classification results (%) using MPLS-DA and CBD

		Predicted class					
		NOC	$\vec{\xi}_1$	$\vec{\xi}_2$	$\vec{\xi}_3$	$\vec{\xi}_4$	
Real Class	NOC	100	0	0	0	0	**MPLS-DA**
	$\vec{\xi}_1$	2	98	0	0	0	
	$\vec{\xi}_2$	0	0	100	0	0	
	$\vec{\xi}_3$	0	0	0	100	0	
	$\vec{\xi}_4$	4	2	0	0	94	
		Predicted class					
		NOC	$\vec{\xi}_1$	$\vec{\xi}_2$	$\vec{\xi}_3$	$\vec{\xi}_4$	
Real Class	NOC	100	0	0	0	0	**CBD**
	$\vec{\xi}_1$	0	100	0	0	0	
	$\vec{\xi}_2$	0	0	100	0	0	
	$\vec{\xi}_3$	0	0	0	100	0	
	$\vec{\xi}_4$	0	4	0	2	94	

NN model; and with class 4 and 2 when using the CBD approach. Both methods have a very good classification accuracy (MPLS-DA: 98% and CBD: 99%), but it is interesting to remark that CBD never confuses a wrong injection with normal ones (class 1). So, this one seems more robust to isolate good and bad injections and consequently more appropriate for on-line quality control.

5.4 Real-time monitoring of an injection moulding machine

Given the good results presented in the previous section, the CBD was used for the real-time release monitoring of an injection moulding machine within the European project EMOLD (COLL-CT-2006-030339). Its aim was to convert a traditionally passive element like the mould into an active part of the production line with embedded knowledge. This embedded knowledge comes from the CBD and its learning capabilities and thanks to the knowledge stored in other “intelligent” moulds within the “mould network” proposed in this project.

The CBD described in the previous section was used to monitor the Krauss Maffei injection machine depicted in Figure 5.5A using a prototype of the eBox system (Figure 5.5B) developed in this project. One pressure and one temperature sensors were attached to the mould to feed the CBD, which used this information to build the statistical model and case base in this way:

- Injections were batch-wise unfolded and auto-scales in order to build the statistical model of the normal operating conditions of the mould (optimal parameters).
- The number of principal components was selected based on the cross-validation method described in Himes et al. (1994).
- As stated before, faulty situation could be clearly separated within the PCS, and thus, the normalised Euclidean distance d_t described in 3.3.2.1 was used.
- The same defects explained in the previous section (NOC, flashes, sink marks and locally oversized parts) and characterised using the same process configurations (optimal parameters, switch-over too early, no holding pressure, switch-over too late without holding pressure and switch-over too late with holding pressure) were fed to the case base for predicting the final quality of an injected part. Additionally, a series of corrective actions were added for each class in the diagnosis structure to facilitate the defect correction.
- The information of the nearest neighbour was sufficient to indicate the final quality of the currently injected part, and thus, no reuse procedure was necessary.
- DROP4 was applied to minimise the case base. However, this was only applied during the training stage.
- All injected parts received an ID to track the final quality of each injected part and stored in a data base for future reference of other network moulds or to repeat the training process in case of a performance degradation.

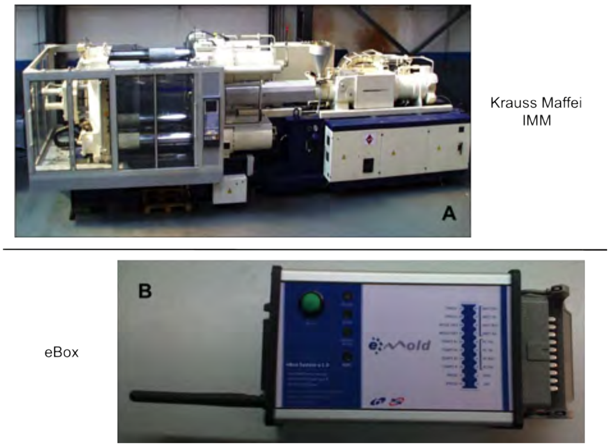


Figure 5.5: Picture of the Krauss Maffei injection moulding machine used during the EMOLD project (A) and the eBox (B) used to collect data from the mould

- Based on the training results, each class was assigned a misclassification margin. In case that any of this margins was surpassed, or the process operator observed a drift from the production objectives (based on the visual tools provided), the training process was applied (based on all stored observations) to update both the statistical model and the case base used for the final quality prediction.

The projection into the two first principal components of all observations in the diagnosis case base (after applying DROP4) are depicted in Figure 5.6 (different colours and shapes are used to represent each class). As can be seen, the first principal component is able to differentiate between the different clusters (gathers the inter-class variability), while the second one shows the (intra-class) variability within the cluster.

In order to test the CBD performance, the parameters of the injection machine were altered in such a way that the resulting part presented one of the defects the CBD recognises. Results obtained are presented in Table 5.4, which presents the extended confusion matrix for this scenario.

As it can be seen, in this case there are no misclassifications, and thus, there are no elements outside the diagonal. The main reason of the perfect classification is that cluster centres are so distant that projections in the first principal component of

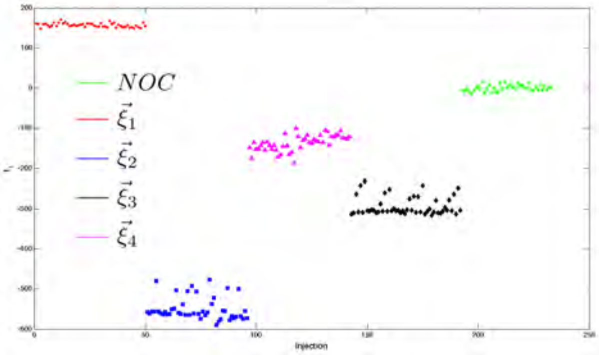


Figure 5.6: Picture of the Krauss Maffei injection moulding machine used during the EMOLD project (A) and the eBox (B) used to collect data from the mould

Table 5.4: Extended confusion matrix for the on-line application

		Real class				
		NOC	$\vec{\xi}_1$	$\vec{\xi}_2$	$\vec{\xi}_3$	$\vec{\xi}_4$
Predicted Class	NOC	41	0	0	0	0
	$\vec{\xi}_1$	0	46	0	0	0
	$\vec{\xi}_2$	0	0	50	0	0
	$\vec{\xi}_3$	0	0	0	50	0
	$\vec{\xi}_4$	0	0	0	0	46

different clusters do not overlap. Additionally, extreme observations for each group are not far enough to be misclassified.

6

Conclusions

This chapter summarizes the conclusions obtained as a result of this research. The relevant conclusions are highlighted and discussed, as well as several ideas for future work are proposed.

6.1 Conclusions

The final objective of this thesis was to develop a general CBD to automatically monitor batch processes. More specifically, the CBD first builds the statistical model of the process, and then feeds the information into a CBR to monitor the process. In order to achieve this final goal, four subobjectives were fixed, which are presented and discussed in the next paragraphs.

Formalise a representation of cases in the projection space capable to represent batch process and exploit such representation for diagnosis and define neighbourhoods on the latent structures (principal component space) useful for diagnosis based on similarity principles. Section 3.3 defined a series distance criteria based on the information provided by the MPCA model of the process (scores, SPE and T^2 indices) to find similar cases. More specifically, three basic distance criteria related to the information in the PCS (scores - d_t - and T^2 - d_{T^2} -) and the RS (SPE index - d_{SPE} -) and a combined SPE and T^2 index (d_ϕ). These distance criteria were later on combined to find neighbourhoods meaningful for process monitoring. On the one hand, some of the distances were defined as two-step retrievals processes to firstly identify batches with the same degree of concordance with the statistical model (based on d_{T^2}) or dissonance (found using d_{SPE}) and then a focalised search for the batches with the same behaviour in the PCS (determined using d_t). On the other hand, some neighbourhoods that restricted the search space to either the normal operating condition (to find the nearest in-control neighbours for faulty observations) or to the abnormal operating conditions. For example, retrieving the nearest in-control observations (within the NOC region) of a faulty observation can be used to determine the fault direction affecting the batch. In case of a sensor fault, this information can be used to correct the sensor misreading without stopping the process, and thus, avoiding the production halt associated costs and the down-time of the process. These definitions were published in a conference paper in Berjaga et al. (2009c), and the relation of these neighbourhood with fault directions and their diagnosability were presented in chapters 2 and 3 from a theoretical point of view.

Reduce the effect of noisy instances when reusing the information extracted from neighbourhoods based on latent structures. Two new reuse methods were defined in section 3.4: a distance-weighted voting (DW) that considered the distance among nearest neighbours to define the likelihood of an observation within all known fault classes; and a distance and class frequency weighted voting (DFW) to overcome the limitation of DW when dealing with single noisy instances. Note that both methods are less sensitive to the number of neighbours retrieved (k) than the traditional simple voting (SV). Both reuse methods were presented in a conference paper (Berjaga et al., 2013a) and a journal paper (Berjaga et al., 2013b).

Specify new case base maintenance and updating policies to minimise the case base, while keeping its diagnosis performance. A new retain method (IBUdG) was detailed in subsection 3.6.3, which is less computationally complex than the well-known methods IB3 and DROP4, especially for the on-line application of the CBD. Its main idea is to retain the case observations that constitute the frontier between the different classes, by keeping the k-nearest enemies (observations from other classes).

To guarantee the generality of the approach, two different application fields were used throughout this thesis: WWTP and IMM. In both cases, the CBD had a great performance on determining the granulation point in an SBR plant (WWTP) and predicting the final quality of injected parts (IMM). Both an off-line analysis (to determine the best statistical model and CBR configuration) and on-line application were stated as well as two preliminary studies showing the applicability of these methods. On the one hand, the CBD facilitated the task of identifying the variable (pH) and stages (first anaerobic stage) for which the differences between two granulation states (floccular and granular) were most observable. Results within this field were partially published in two journal papers (Berjaga et al., 2013b; Ruiz et al., 2011) and in a conference paper (Berjaga et al., 2013a). On the other hand, CBD related the final quality of injected parts (NOC, flashes, sink marks and locally oversized pieces) with its possible root cause (optimal parameters, early switch-over, no holding pressure, switch-over too late with and without holding pressures), and thanks to the experts knowledge, proposed a series of steps to correct the defect found within the production line. Results within this field were published in four conference papers (Berjaga et al., 2008a, 2009a,b,c). Finally, thanks to a series of previous works carried out during the Master thesis (Barrera et al., 2008; Meléndez et al., 2008a,b), it was demonstrated the applicability of MPCA to finite duration processes. Based on these principles, a journal and a conference paper indicating how MPLS could be used to locate impacts in a commercial flap wing was published in Ruiz et al. (2013) and Mujica et al. (2009) respectively.

6.2 Future work

Although the final objective of the thesis was achieved, new situations appeared during this research, which proposed future challenges or topics of interest for the CBD. This section details some future directions and discusses in more detail one application of the methodology for fault sensor detection and reconstruction.

The main limitation of CBRs is that they can only “classify” or reason with the different typologies found on the case base. However, the availability of all possible faulty/problematic situations when training for the first time (to build the statistical model and the case base) is not always an option. For example, let us suppose that no sensor faults have been detected within a process. However, due to external factors, one of the sensors starts to misbehave, since the CBD did not have any previous instance, and this situation was not initially considered, the following observations will be incorrectly classified. However, and since this new case is far from all observations in the case base, this new event can be detected. Some possible approaches to tackle this problem are:

- Compute the average distance among cases and its nearest neighbours in the case base. By doing so, any observation that surpasses this value is either an example of an incipient event, or an extreme observation to any of the current classes. However, note that this approach is sensitive to the inclusion of new observations, since the more cases are in the case base, the lower the influence an observation has to this average. Contrarily, the fewer cases in the case base, the more effect adding a new observation has to this "new situation alarm". Moreover, new trends and performance degradations may not be detected using this approach.
- Use a process window to compute the average distance. This method is an alternative to the previous one, which only uses the last n observations to detect if a new situation, or process drift occurs. Since only the last observations are accounted, one can track how the average distance behaves in order to identify a process drift, which the previous strategy could not detect. However this situation is more sensitive to the appearance of extreme observations because the inclusion of these observation significantly alters the average along time (especially when they appear and while they are kept in the process window). In order to reduce the impact of this kind of observations, statistically robust methods, such as the usage of the median value instead of the average or mean can be used to estimate this threshold.
- Another alternative is to divide or assign subspaces to the different known classes. Let us suppose that for all faulty situation, we know or we can estimate the fault direction vector. Based on this, one can compare the current fault direction with the ones stored in the case base, and that divide the problem space in different and complementary subspaces, then based on the inter- and intra- class variance one can decide if this new observation is an example of a new situation or not.

Another aspect that influences the case base is the number of cases in the case base and how the case base updates. This is the task done during the retain step of the CBR, and depending on the internal distribution of classes on method or the other is more appropriate. However, the retain step is also dependant on the number of observation passed. Moreover, this situation is more critical when a dynamic change happens in the process. Since the process is deviating from the a previously well-known state (the one used to build the statistical model and case base), even by adding only one batch in the retain stage may lead to a case base unable to monitor the current state of the process. This problem can be solved by using any of the previous methods for detecting new trends or situations in the process. For example, if a new trend in the process is detected, the retain step can be paused until n misclassified or new observations are given for this new process state, and then update the case base accordingly.

The last of the future works that the CBD can be oriented towards is the differentiation of process and sensor faults, which statistical models cannot achieve by themselves. As stated in section 2.1, in order to distinguish between additive and multiplicative faults or between simple and multiple faults, one has to take into account the process structure and incorporate this information when building the statistical model of the process. Although multiblock principal component analysis (MB-PCA) can be used to divide the process into the different blocks that it can be divided into, additional rules of criteria have to be placed over each block in order to locate a sensor or block fault (only one of the blocks observes the faults), and complex faults can only be detected using the superscores. On the other hand, when using the CBD, this process is simpler, since the combination of the fault direction vector (even along time) and the fault diagnosis are sufficient to separate sensor faults (simple and multiple faults) from process faults in one single step.

Related with the identification of sensors faults, a first approximation to the problem was presented in Berjaga et al. (2010) applied to the sensor readings of the Arianne engine based on the variance of the reconstruction method devised by Dunia and Qin in Dunia and Qin (1998) (continuous process). In this work, it was proposed the usage of a “global” fault detection method (to detect the sensor fault appearance), while a “local” reconstruction model is used to correct the sensor misbehaviour. Figure 6.1 presents a

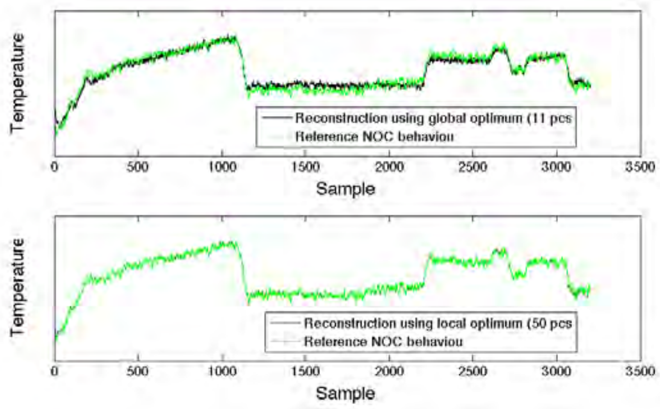


Figure 6.1: Comparison of the reconstruction using the global optimal number of principal components (upper figure dark line) with respect the usage of the local optimum (lower figure dark line). In both cases no normal operating conditions are available for the faulty sensor, and the comparison is made with a redundant sensor placed near the faulty one.

visual comparison of the reconstruction obtained when using the global model (upper figure) with respect the local reconstruction model (lower figure) with respect the NOC expected behaviour of the variable. Additionally, two different approaches were developed in order to deal with multiple sensor faults: parallel and iterative reconstruction. The former reconstruct each faulty sensor building a local model and discarding the information of the other faulty sensors; while the latter add the information reconstructed to correct the misbehaviours in later reconstruction iterations. Future works within this line are oriented to extend these concepts and methods to batch processes.

References

A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7 (1):39–59, 1994. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.9093&amp;rep=rep1&amp;type=pdf. 50

D. Aguado, A. Ferrer, J. Ferrer, and A. Seco. Multivariate SPC of a sequencing batch reactor for wastewater treatment. *Chemometrics and Intelligent Laboratory Systems*, 85(1):82–93, January 2007. ISSN 01697439. doi: 10.1016/j.chemolab.2006.05.003. URL http://linkinghub.elsevier.com/retrieve/pii/S0169743906000840. 12

D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6 (1):37–66, January 1991. ISSN 0885-6125. doi: 10.1007/BF00153759. URL http://link.springer.com/10.1007/BF00153759. 77, 81

C. F. Alcala and S. J. Qin. Reconstruction-based contribution for process monitoring. *Automatica*, 45(7):1593–1600, July 2009. ISSN 00051098. doi: 10.1016/j.automatica.2009.02.027. URL http://linkinghub.elsevier.com/retrieve/pii/S0005109809001277. 12

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, New York, 1984. ISBN 0471360910. 34

F. Arteaga and A. Ferrer. Framework for regression-based missing data imputation methods in on-line MSPC. *Journal of Chemometrics*, 19(8):439–447, August 2005. ISSN 0886-9383. doi: 10.1002/cem.946. URL http://doi.wiley.com/10.1002/cem.946. 88

V. Barrera, X. Berjaga, J. Meléndez, S. Herraiz, J. Sánchez, and M. Castro. Two Methods for Voltage Sag Source Location. In *13th International Conference on Harmonics and Quality of Power*, pages 1–6, 2008. doi: 10.1109/ICHQP.2008.4668849. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4668849. iii, vii, 15, 45, 47, 131

X. Berjaga. *Case-Based Diagnosis in the Principal Components Space. From definition to applications.* PhD thesis, University of Girona, 2008. ii, 6

X. Berjaga, J. Meléndez, and A. Pallarés. Statistical Monitoring of Injection Moulds. In *11th International Conference of the Catalan Association for Artificial Intelligence*, pages 236–243. IOS Press, 2008a. URL http://portal.acm.org/citation.cfm?id=1566933. i, 10, 15, 131

X. Berjaga, A. Pallarés, and J. Meléndez. A Case-Based Centred Approach for Rapid Manufacturing: Definitions. In *Hybrid Intelligent Systems*, 2008b. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4626759. iii

X. Berjaga, A. Pallares, and J. Meléndez. *A framework for Case-Based Diagnosis of batch processes in the principal components space.* IEEE, September 2009a. ISBN 978-1-4244-2727-7. doi: 10.1109/ETFA.2009.5347075. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5347075. i, 10, 15, 114, 118, 131

X. Berjaga, A. Pallarés, and J. Meléndez. Quality Determination of Industrial Batch Processes in the Principal Component Space. In *Seminario de Aplicaciones Industriales de Control Avanzado*, pages 217–224, 2009b. i, 10, 15, 16, 114, 118, 131

X. Berjaga, A. Pallarés, J. Meléndez, and Fco. I. Gamero. Case-Based Diagnosis in the principal component space. Application to injection moulds. In *20th Internation Workshop on Principles of Diagnosis*, pages 195–202, 2009c. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Case-Based+Diagnosis+in+the+principal+component+space+.+Application+to+injection+moulds#1. i, 10, 53, 114, 130, 131

X. Berjaga, J. Meléndez, and C. Barta. Statistical fault detection and reconstruction of sensors of the Ariane engine. In *Control & Automation (MED), 2010 18th Mediterranean Conference on*, pages 1467–1472. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5547841. ii, 16, 133

X. Berjaga, M. Coma, J. Meléndez, S. Puig, J. Colprim, and J. Colomer. Granularity determination of activated sludge trhough on-line profiles by means of case-based reasoning. In *IWA Conference on Instrumentation Control and Automation*, 2013a. ii, 10, 15, 91, 130, 131

X. Berjaga, M. Coma, J. Meléndez, S. Puig, J. Colprim, and J. Colomer. Granularity determination of activated sludge through on-line profiles by means of case-based reasoning. *Water Science & Technology*, 2013b. i, 10, 15, 83, 91, 130, 131

M. H. J. Bollen. *Understanding Power Quality Problems: Voltage Sags and Interruptions*. Wiley, 2000. 45

A. T. Bozdana and O. Eyercioglu. Development of an expert system for the determination of injection moulding parameters of thermoplastic materials: EX-PIMM. *Journal of Materials Processing Technology*, 128:113–122, October 2002. ISSN 09240136. doi: 10.1016/S0924-0136(02)00436-3. URL http://linkinghub.elsevier.com/retrieve/pii/S0924013602004363. 116

A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, July 1997. ISSN 00313203. doi: 10.1016/S0031-3203(96)00142-2. URL http://linkinghub.elsevier.com/retrieve/pii/S0031320396001422. 71, 76

Ll. Burgas. Llibreria pel control estadístic de processos industrials per lots. Aplicació en una planta de depuració daigües. Technical report, Universitat de Girona, Girona, 2013. 6, 77

M. V. Cedeño Viteri, L. P. F. Rodríguez Aguilar, C. R. Alvarez Medina, and M. C. Sánchez. A new approach to estimate variable contributions to Hotelling's statistic. *Chemometrics and Intelligent Laboratory Systems*, 118:120–126, August 2012. ISSN 01697439. doi: 10.1016/j.chemolab.2012.08.004. URL http://linkinghub.elsevier.com/retrieve/pii/S0169743912001591. 13

J. Chen and H.-H. Chen. On-line batch process monitoring using MHMT-based MPCA. *Chemical Engineering Science*, 61(10):3223–3239, May 2006. ISSN 00092509. doi: 10.1016/j.ces.2005.12.006. URL http://linkinghub.elsevier.com/retrieve/pii/S0009250905009358. 14

J. Chen and Y.-C. Jiang. Development of hidden semi-Markov models for diagnosis of multiphase batch operation. *Chemical Engineering Science*, 66(6):1087–1099, March 2011. ISSN 00092509. doi: 10.1016/j.ces.2010.12.009. URL http://linkinghub.elsevier.com/retrieve/pii/S0009250910007220. 14

J. Chen and C. M. Liao. Dynamic process fault monitoring based on neural network and PCA. *Journal of Process control*, 12:277–289, 2002. URL http://linkinghub.elsevier.com/retrieve/pii/S0959152401000270. 26

Ll. Corominas. *Control and optimization of an SBR for nitrogen removal: from model calibration to plant operation*. PhD thesis, Universitat de Girona, 2006. URL http://www.tesisenred.net/handle/10803/7659. 92

R. L. de Mántaras and E. Plaza. Case-based reasoning: an overview. *AI communications*, 10:21–29, 1997. URL http://iospress.metapress.com/index/99el143bfjfv9gr2.pdf. 50

C. Ding and X. He. K -means clustering via principal component analysis. *Twenty-first international conference on Machine learning - ICML '04*, page 29, 2004. doi: 10.1145/1015330.1015408. URL http://portal.acm.org/citation.cfm?doid=1015330.1015408. 7

R. Dunia and S. J. Qin. A unified geometric approach to process and sensor fault identification and reconstruction: the unidimensional fault case. *Computers and Chemical Engineering*, 22(7-8):927–948, 1998. 12, 13, 133

US EPA. Wastewater Technology Fact Sheet-Sequencing Batch Reactors. Technical report, 1999. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Wastewater+Technology+Fact+Sheet+Sequencing+Batch+Reactors#1. 92

T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

ISSN 01678655. doi: 10.1016/j.patrec.2005.10.010. URL http://linkinghub.elsevier.com/retrieve/pii/S016786550500303X. 75, 76

A. Ferrer. Multivariate Statistical Process Control Based on Principal Component Analysis (MSPC-PCA): Some Reflections and a Case Study in an Autobody Assembly Process. *Quality Engineering*, 19(4):311–325, October 2007. ISSN 0898-2112. doi: 10.1080/08982110701621304. URL http://www.informaworld.com/openurl?genre=article&doi=10.1080/08982110701621304&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3. 5, 12

S. García-Munoz, T. Kourti, and J. F. MacGregor. Model Predictive Monitoring for Batch Processes. *Industrial & Engineering Chemistry Research*, 43 (18):5929–5941, September 2004. ISSN 0888-5885. doi: 10.1021/ie034020w. URL http://pubs.acs.org/doi/abs/10.1021/ie034020w. 27

Paul Geladi. Analysis of multi-way (multi-mode) data. *Chemometrics and Intelligent Laboratory Systems*, 7(1-2):11–30, December 1989. ISSN 01697439. doi: 10.1016/0169-7439(89)80108-X. URL http://linkinghub.elsevier.com/retrieve/pii/016974398980108X. 7

J. Gertler. *Fault detection and diagnosis in engineering systems*. New York, New York, USA, 1st edition, 1998. ISBN 0824794273. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Fault+Detection+and+Diagnosis+in+Engineering+Systems#0. 19

X. P. Guo, F. L. Wang, and M. X. Jia. A Substage Moving Window GRNN Quality Prediction Method for Injection Molding Processes. In *Advances in Neural Networks*, volume 3973, pages 1138–1143, 2006. 14, 117

L. Hare. SPC: From Chaos To Wiping the Floor. *Quality Progress*, 36(7):58, 2003. 10

D. M. Himes, R. H. Storer, and C. Georgakis. Determination of the number of principal components for disturbance detection and isolation. In *American Control Conference*, volume 00, pages 2–6, 1994. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=752265. 97, 121, 125

R. Isermann and P. Ballé. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control engineering practice*, 5(5): 709–719, 1997. URL http://www.sciencedirect.com/science/article/pii/S0967066197000531. 18, 19, 21

J. E. Jackson and G. S. Mudholkar. Control Procedures for Residuals Associated with Principal Component Analysis. *Technometrics*, 21(3):341–349, August 1979. doi: 10.2307/1267757. 36

A. Jakulin and I. Bratko. Analyzing attribute dependencies. *Knowledge Discovery in Databases: PKDD 2003*, pages 229–240, 2003. URL http://www.springerlink.com/index/9G11UEGTUT0P72RM.pdf. 8, 50, 84

X. Jin, F. Wang, G. Liu, and N. Yan. A key cultivation technology for denitrifying granular sludge. *Process Biochemistry*, 47(7):1122–1128, July 2012. ISSN 13595113. doi: 10.1016/j.procbio.2012.04.001. URL http://linkinghub.elsevier.com/retrieve/pii/S1359511312001389. 94

R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, NJ, 1988. ISBN 0131877151. 34

D. O. Kazmer and S. Westerdale. A model-based methodology for on-line quality control. *International Journal of Advanced Manufacturing Technology*, 42(3-4):280–292, 2008. ISSN 0268-3768. doi: 10.1007/s00170-008-1592-4. URL http://www.springerlink.com/index/10.1007/s00170-008-1592-4. 117

A. Khosravi, J. Meléndez, J. Colomer, and J. Sánchez. Multiway Principal Component Analysis (MPCA) for Upstream/Downstream Classification of Voltage Sags Gathered in Distribution Substations. *Advances of Computational ...*, 116:297–312, 2008. URL http://www.springerlink.com/index/61266208757N3864.pdf. 12

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International joint Conference on artificial intelligence*, 14:1137–1145, 1995. ISSN 1045-0823. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.529&amp;rep=rep1&amp;type=pdfhttp://frostiebek.free.fr/docs/MachineLearning/validation-1.pdf. 71

K. A. Kosanovich and M. J. Piovoso. Multiway PCA applied to an industrial batch process. *American Control ...*, pages 1294–1298,

1994. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=752268. 11

T. Kourti. Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *Journal of Chemometrics*, 17:93–109, 2003. 28, 29

T. Kourti. Application of latent variable methods to process control and multivariate statistical process control in industry. *International Journal of Adaptive Control and Signal Processing*, 19(4):213–246, May 2005. ISSN 0890-6327. doi: 10.1002/acs.859. URL http://doi.wiley.com/10.1002/acs.859. 7, 11, 39

T. Kourti, P. Nomikos, and J. F. MacGregor. Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *Journal of Process Control*, 5(4):277–284, August 1995. ISSN 09591524. doi: 10.1016/0959-1524(95)00019-M. URL http://linkinghub.elsevier.com/retrieve/pii/095915249500019M. 11

D. B. Leake. *Case-based reasoning: Experiences, lessons and future directions*. 1996. ISBN 0-262-62110-X. 51

J. J. Liu and J. F. MacGregor. Modeling and Optimization of Product Appearance: Application to Injection-Molded Plastic Panels. *Industrial & Engineering Chemistry Research*, 44(13):4687–4696, June 2005. ISSN 0888-5885. doi: 10.1021/ie0492101. URL http://pubs.acs.org/doi/abs/10.1021/ie0492101. 117

Y. Liu and J.-H. Tay. The essential role of hydrodynamic shear force in the formation of biofilm and granular sludge. *Water research*, 36(7):1653–65, April 2002. ISSN 0043-1354. URL http://www.ncbi.nlm.nih.gov/pubmed/12044065. 92

Y. Liu, Z.-W. Wang, L. Qin, Y.-Q. Liu, and J.-H. Tay. Selection pressure-driven aerobic granulation in a sequencing batch reactor. *Applied microbiology and biotechnology*, 67(1):26–32, April 2005. ISSN 0175-7598. doi: 10.1007/s00253-004-1820-2. URL http://www.ncbi.nlm.nih.gov/pubmed/15800730. 92

J. B. Lohmöller and H. Wold. Three-mode path models with latent variables and partial least squares (PLS) parameter estimation. In *European Meeting of Psycomerics Society*, Groningen, 1980. 7

J. F. MacGregor. Multivariate statistical approaches to fault detection and isolation. In *SAFEPROCESS*, 2003. 26, 33

J. F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403–414, 1995. URL http://linkinghub.elsevier.com/retrieve/pii/096706619500014L. 11

B. S. McSwain, R. L. Irvine, and P. A. Wilderer. The influence of settling time on the formation of aerobic granules. *Water Science & Technology*, 50(10):195–220, 2004. 93

J. Meléndez, X. Berjaga, S. Herraiz, V. Barrera, J. Sánchez, and M. Castro. *Classification of sags according to their origin based on the waveform similarity*. IEEE, August 2008a. ISBN 978-1-4244-2217-3. doi: 10.1109/TDC-LA.2008.4641819. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4641819. ii, vii, 15, 45, 47, 131

J. Meléndez, X. Berjaga, S. Herraiz, J. Sánchez, and M. Castro. Classification of Voltage Sags based on k-NN in the Principal Component Space. In *International Conference on Renewable Energies and Power Quality-ICREPQ*, volume 8, pages 1–6, 2008b. URL http://www.icrepq.com/icrepq-08/417_melendez.pdf. ii, vii, 15, 45, 47, 131

R. Milne. Strategies for Diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3):333–339, 1987. ISSN 0018-9472. doi: 10.1109/TSMC.1987.4309050. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4309050. 2

L. E. Mujica, J. Vehi, M. Ruiz, M. Verleysen, W. Staszewski, and K. Worden. Multivariate statistics process control for dimensionality reduction in structural assessment. *Mechanical Systems and Signal Processing*, 22(1):155–171, 2008. ISSN 08883270. doi: 10.1016/j.ymssp.2007.05.001. URL http://linkinghub.elsevier.com/retrieve/pii/S0888327007000751. 14

L. E. Mujica, M. Ruiz, X. Berjaga, and J. Rodellar. Multiway Partial Least Square (MPLS) to Estimate Impact Localization in Structures. In Escobet Teresa, editor, *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 1025–1030, June 2009. doi: 10.3182/20090630-4-ES-2003.00169. URL http://www.ifac-papersonline.net/Detailed/40225.html. ii, vii, 15, 45, 46, 131

D. Napoleon and S. Pavalakodi. A New Method for Dimensionality Reduction using K-Means Clustering Algorithm for High Dimensional Data Set. *International Journal of Computer Applications*, 13(7):41–46, 2011. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+New+Method+for+Dimensionality+Reduction+using+K-+Means+Clustering+Algorithm+for+High+Dimensional+Data+Set#0. 7

P. R. C. Nelson, P. A. Taylor, and J. F. MacGregor. Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and intelligent*, 35:45–65, 1996. URL http://linkinghub.elsevier.com/retrieve/pii/S016974399600007X. 88

P. Nomikos and J. F. MacGregor. Monitoring Batch Process Using Multiway Principal Componet Analysis. *AIChE Journal*, 40(8):1361–1375, 1994. URL http://www3.interscience.wiley.com/journal/109063733/abstract. 3, 7, 11, 25, 26, 28, 35

P. Nomikos and J. F. MacGregor. Multi-way partial least squares in monitoring batch processes. *Chemometrics and intelligent laboratory systems*, 30:97–108, 1995a. URL http://linkinghub.elsevier.com/retrieve/pii/0169743995000437. 11

P. Nomikos and J. F. MacGregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37(1):41–59, 1995b. URL http://www.jstor.org/stable/1269152. 3, 11, 18

S. Puig, Ll. Corominas, M. T. Vives, M. D. Balaguer, J. Colprim, and J. Colomer. Development and Implementation of a Real-Time Control System for Nitrogen Removal Using OUR and ORP as End Points. *Industrial & Engineering Chemistry Research*, 44(9):3367–3373, April 2005. ISSN 0888-5885. doi: 10.1021/ie0488851. URL http://pubs.acs.org/doi/abs/10.1021/ie0488851. 93

M. S. Reis and P. Delgado. A large-scale statistical process control approach for the monitoring of electronic devices assemblage. *Computers & Chemical Engineering*, 39:163–169, April 2012. ISSN 00981354. doi: 10.1016/j.compchemeng.2011.12.008. URL http://linkinghub.elsevier.com/retrieve/pii/S0098135411003450. 12

C. Rosen and J. A. Lennox. Multivariate and multiscale monitoring of wastewater treatment operation. *Water research*, 35(14):3402–3410, October 2001. ISSN 0043-1354. URL http://www.ncbi.nlm.nih.gov/pubmed/11547861. 12

M. Ruiz. *Multivariate Statistical Process Control and Case-Based Reasoning for situation assessment of Sequencing Batch Reactors Magda Liliana Ruiz Ord* . PhD thesis, Universitat de Girona, 2008. 5, 6, 77, 83

M. Ruiz, G. Sin, X. Berjaga, J. Colprim, S. Puig, and J. Colomer. Multivariate Principal Component Analysis and Case-Based Reasoning for monitoring, fault detection and diagnosis in a WWTP. *Water Science & Technology*, 64(8):1661–1667, October 2011. ISSN 0273-1223. doi: 10.2166/wst.2011.517. URL http://www.iwaponline.com/wst/06408/wst064081661.htm. i, 10, 131

M. Ruiz, L. E. Mujica, X. Berjaga, and J. Rodellar. Partial least square/projection to latent structures (PLS) regression to estimate impact localization in structures. *Smart Materials and Structures*, 22(2):025028, February 2013. ISSN 0964-1726. doi: 10.1088/0964-1726/22/2/025028. URL http://iopscience.iop.org/0964-1726/22/2/025028http://stacks.iop.org/0964-1726/22/i=2/a=025028?key=crossref.0de6e92907954b7faeef3e56759d5368. ii, vii, 15, 45, 46, 131

E. L. Russell, L. H. Chiang, and R. D. Braatz. *Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes (Advances in Industrial Control)*. Springer, 1st editio edition, 2000. ISBN 9781852332587. 25

S. Sernells and T. Verdonck. Principal component analysis for data containing outliers and missing elements. *Computational Statistics & Data Analysis*, 52(3):1712–1727, January 2008. ISSN 01679473. doi: 10.1016/j.csda.2007.05.024. URL http://linkinghub.elsevier.com/retrieve/pii/S0167947307002241. 88

M. Sjöstrom, S. Wold, and B. Söderström. PLS Discriminant Plots. In *Proceedings of PARC in Practice*, 1985. 13, 86

A. K. Smilde and D. A. Doornbos. Three-way methods for the calibration of chromatographic systems: Comparing PARAFAC and three-way PLS. *Journal of Chemometrics*, 5(4):345–360, July 1991. ISSN 0886-9383. doi: 10.1002/cem.1180050404. URL `http://doi.wiley.com/10.1002/cem.1180050404`. 7

M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, July 2009. ISSN 03064573. doi: 10.1016/j.ipm.2009.03.002. URL `http://linkinghub.elsevier.com/retrieve/pii/S0306457309000259`. 74

J. Tay, S. Pan, Y. He, and S. Tay. Effect of Organic Loading Rate on Aerobic Granulation. I: Reactor Performance. *Journal of Environmental Engineering*, 130(10):1094–1101, 2004. 93

G. Tchobanoglous, F. L. Burton, and H. D. Stensel. *Wastewater engineering: treatment and reuse*. McGraw- Hill Higher Education, New York, USA, 4th ed. edition, 2003. 92

S. Valle, S. J. Qin, M. J. Piovoso, M. Bachman, and N. Mandakoro. Extracting fault subspaces for fault identification of a polyester film process. In *American Control Conference*, pages 4466–4471. AACC, 2001. ISBN 0-7803-6495-3. doi: 10.1109/ACC.2001.945682. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=945682`. 7, 13

V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri. A review of process fault detection and diagnosis Part II: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27(3):313–326, March 2003a. ISSN 00981354. doi: 10.1016/S0098-1354(02)00161-8. URL `http://linkinghub.elsevier.com/retrieve/pii/S0098135402001618`. 2

V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin. A review of process fault detection and diagnosis Part III: Process history based methods. *Computers & Chemical Engineering*, 27(3):327–346, March 2003b. ISSN 00981354. doi: 10.1016/S0098-1354(02)00162-X. URL `http://linkinghub.elsevier.com/retrieve/pii/S009813540200162X`. 2

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 & Chemical Engineering*, 27(3):293–311, March 2003c. ISSN 00981354. doi: 10.1016/S0098-1354(02)00160-6. URL `http://linkinghub.elsevier.com/retrieve/pii/S0098135402001606`. 2

J. A. Westerhuis, T. Kourti, and J. F. MacGregor. Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics*, 13:397–413, May 1999. ISSN 08869383. doi: 10.1002/(SICI)1099-128X(199905/08)13:3/4⟨397::AID-CEM559⟩3.3.CO;2-9. URL `http://doi.wiley.com/10.1002/(SICI)1099-128X(199905/08)13:3/4<397::AID-CEM559>3.3.CO;2-9`. 28

D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000. URL `http://www.springerlink.com/index/n748p82037443824.pdf`. 77, 78

B. M. Wise, N. B. Gallagher, S. W. Butler, D. D. White, and G. G. Barna. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics*, 13:379–396, May 1999. ISSN 0886-9383. doi: 10.1002/(SICI)1099-128X(199905/08)13:3/4⟨379::AID-CEM556⟩3.0.CO;2-N. URL `http://doi.wiley.com/10.1002/(SICI)1099-128X(199905/08)13:3/4<379::AID-CEM556>3.0.CO;2-N`. 24

B. M. Wise, N. B. Gallagher, R. Bro, J. M. Shaver, W. Windig, and R. S. Koch. *Chemometrics Tutorial for PLS _ Toolbox and Solo*. Eigenvector Research Incorporated, 2006. ISBN 0976118416. 39

S. Wold, C. Albano, W. J. Dunn, U. Edlund, K. Esbensen, P. Geladi, S. Helliberg, E. Johansson, W. Lindeberg, and M. Sjöström. Multivariate Data Analysis in Chemistry. In B. R. Kowalski, editor, *Chemometrics: Mathematics and Statistics in Chemistry*, pages 17–95. 1984. 13, 87

S. Wold, N. Kettaneh, H. Fridén, and A. Holmberg. Modelling and diagnostics of batch processes and analogous kinetic experiments. *Chemometrics and Intelligent Laboratory Systems*, 44:331–340, December 1998. ISSN 01697439. doi: 10.1016/S0169-7439(98)00162-2. URL `http://linkinghub.elsevier.com/retrieve/pii/S0169743998001622`. 26

Svante Wold, Paul Geladi, Kim Esbensen, and Jerker Öhman. Multi-way principal components-and PLS-analysis. *Journal of Chemometrics*, 1(1):41–56, January 1987. ISSN 0886-9383. doi: 10.1002/cem.1180010107. URL `http://onlinelibrary.wiley.com/doi/10.1002/cem.1180010107/abstracthttp://doi.wiley.com/10.1002/cem.1180010107`. 7

W. H. Woodall. Controversies and contradictions in statistical process control. *Journal of Quality Technology*, 32(4), 2000. URL `http://myplace.frontier.com/~stevebrainerd1/STATISTICS/ECE-580-DOEWEEK3_files/WoodallpaperControlcharts.pdf`. 4

Y. Yao and F. Gao. Phase and transition based batch process modeling and online monitoring. *Journal of Process Control*, 19(5):816–826, May 2009. ISSN 09591524. doi: 10.1016/j.jprocont.2008.11.001. URL `http://linkinghub.elsevier.com/retrieve/pii/S0959152408001595`. 3

S. Yoon and J. F. MacGregor. Fault diagnosis with multivariate statistical models part I: using steady state fault signatures. *Journal of Process Control*, 11:387–400, August 2001. ISSN 09591524. doi: 10.1016/S0959-1524(00)00008-1. URL `http://linkinghub.elsevier.com/retrieve/pii/S0959152400000081`. 55

Z. Yuan, A. Oehmen, Y. Peng, Y. Ma, and J. Keller. Sludge population optimisation in biological nutrient removal wastewater treatment systems through on-line process control: a re/view. *Reviews in Environmental Science and Bio/Technology*, 7(3):243–254, April 2008. ISSN 1569-1705. doi: 10.1007/s11157-008-9134-y. URL `http://www.springerlink.com/index/10.1007/s11157-008-9134-y`. 93

M. A. Zahid and A. Sultana. Assessment and comparison of multivariate process capability indices in ceramic industry. *Journal of Mechanical Engineering*, 39(1):18–25, January 2009. ISSN 0379-4318. doi: 10.3329/jme.v39i0.1829. URL `http://journals.sfu.ca/bd/index.php/JME/article/view/1829`. 12

Appendices

Appendix A

Multiway partial least squares with discriminant analysis (MPLS-DA)

A.1 Partial least squares (PLS)

PLS is a linear regression method that is based on identifying the relations between the predictor matrix (X) and the predicted matrix (Y). Like in PCA, the observation matrix X contains a set of n observations (rows) with m measured variables (columns). Y usually contains quality information associated to the observations in X represented by a row vector of k variables for each observation. Consequently, Y has n rows (one for each observation in X) and k columns. The main idea of PLS is to identify a reduced subset of latent variables (r) that explain how changes in the predictor matrix X affect the predicted matrix Y. In order to avoid the problem of having correlated variables in the observation matrix X (linear regression requires the predictor matrix to be non singular), PLS defines the regression from latent variables extracted from X over the latent variables of Y in this way:

$$Y = XB + E \tag{A.1}$$

with B being the regression coefficient matrix and E defined as a noise term. X and Y are assumed to be auto scaled (zero mean and unit variance). The goal is to compute the factor score matrix $T = XW$ for an appropriate weight matrix W and then considers the linear regression model $Y = TQ + E$, where Q is the loading matrix and E is the resulting error matrix for the regression. Once the Q loadings are computed and considering that $B = WQ$, the previous regression model is equivalent to $Y = XB + E$. PLS computes the weight matrix W reflecting the covariance structure between the predictor and response variables instead of gathering the covariance structure between the predictor variables. The standard algorithm for computing PLS regression factors is the algorithm NIPALS (nonlinear iterative partial least squares).

Finally, the estimation of the predicted variable ($\hat{\vec{y}}_{new}$) for a new observation ($\vec{x}_{new}$) is obtained in this way:

$$\hat{\vec{y}}_{new} = \vec{x}_{new} B \tag{A.2}$$

A.2 Discriminant analysis based on latent variables

The main objective of PLS is to identify those variations in the predictor matrix, X, that are responsible of major changes on the predicted variables Y. Consequently, one can use Y in order to obtain a regression model, a PLS model, to distinguish among several categories, or classes, of observations. Since, regression methods are usually associated with continuous variables, PLS is combined with discriminant analysis (DA) to constrain the regression to a finite set of classes in Y. The combination of both techniques provides a sharp separation between groups of observations, by mixing the latent variables in such a way that maximum separation among classes is attained, and based on that observed variables carry the class separating information. Actually, PLS components are built by trying to find a proper compromise between describing the set of explanatory variables and predicting their response.

The main concern when using this approach is that whenever a new category has to be added, a new PLS model has to be built; and considering that discriminant analysis maximises the inter-class variation whilst minimising the intra-class variability, the method requires enough observations to grant a reliable inter- and intra-class variation.

The only requirement to apply this methodology to batch processes is to unfold the three-dimensional matrix using any of the methodologies described in chapter 2 and then scale the variables accordingly.

Appendix B

Comparison of computation time of case base optimisation algorithms

This appendix compares the time consumption for each maintenance policies considered within this thesis (DROP4 (subsection 3.6.1), IB3 (subsection 3.6.2) and IBUdG (subsection 3.6.3)) and adds a new one: store any misclassified observation (also known as Instance-Based learner 2 (IB2)). In order to do so, this appendix has been divided into two main sections: section B.1 details the procedure to generate the data sets used to evaluate the time consumption of each policy; and section B.2 presents the results obtained with each maintenance policy and the conclusions extracted from this comparison.

B.1 Data, model and case base generation for the test

In order to compare the computation time for each of the maintenance policies considered (IB2, IB3, DROP4 and IBUdG), we will generate a series of random batches based on a polynomial function. The main idea is to generate each variable independent from the others by selecting the order of the polynomial function as a random value between 3 and 20. The variability between batches is attained by modifying the independent term of the characteristic equation (variations in the y-axis), as well as modifying some of the variable values with a Gaussian noise generator (the number of points to mod-

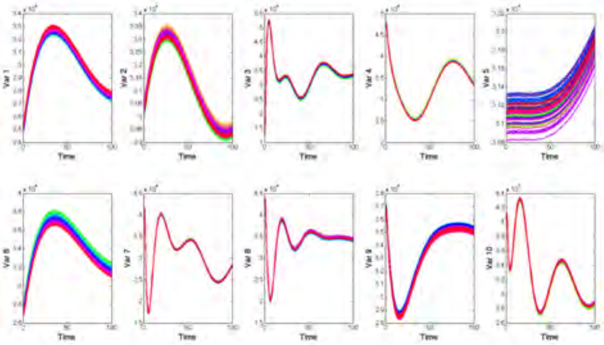


Figure B.1: Trajectory of all variables for all batches used to test the maintenance policy methods (IB2, IB3, DROP4 and IBUdG)

ified is also selected randomly). This means that the pseudo-code for generating a set of ι batches, with γ variables and κ samples is the one presented in Algorithm B.1. Additionally, Figure B.1 presents the trajectory of a random set of 102 batches ($\iota = 102$), with 10 variables ($\gamma = 10$) and with 100 samples for each variable ($\kappa = 100$) to illustrate the type of batch processes generated.

Algorithm B.1 Random generation of batch processes

```
Input: The number of batches ($\iota$) to generate; the number of variables ($\gamma$) and samples ($\kappa$) for each
    batch generated; and the number of classes batches will be divided into ($nClasses$).
Output: The three-dimensional matrix ($\underline{X}$) filled.
function RANDOMFILL($\iota$, $\gamma$, $\kappa$, $nClasses$)
    Create the variable to store the three-dimensional matrix of random batches ($\underline{X}$).
    for $j = 1$ to $\gamma$ do
        Generate the polynomial function of the $j^{th}$ variable ($polF_j$) with a random order $\in [3, 20]$.
        for $i = 1$ to $\iota$ do
            Alter the independent term of the polynomial function to add variability in the y-axis,
    and generate $\kappa$ samples of the $j^{th}$ variable for the $i^{th}$ batch.
            Modify a series of random points of the $j^{th}$ variable in the $i^{th}$ batch to add variability
    in the x-axis.
            Add the $\kappa$ samples for variable $j$ of batch $i$ in the three-dimensional matrix ($\underline{X}(i, j, :)$).
            Randomly assign the class of batch $i$ ($\in [1, nClasses]$).    ▷ The class is only assign once
    for each batch
        end for
    end for
    return The three-dimensional matrix $\underline{X}$ filled.
end function
```

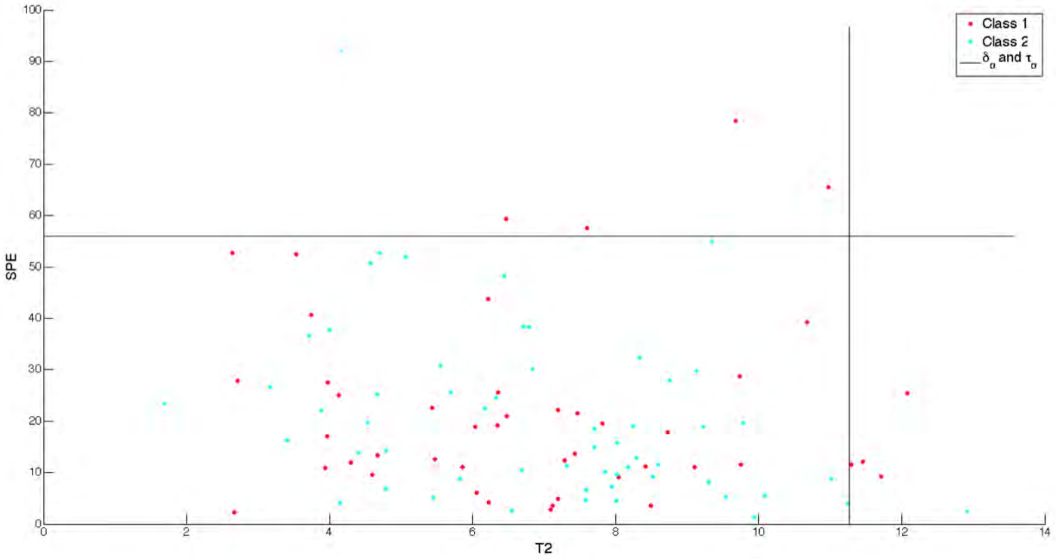


Figure B.2: Projection into the $T^2 - SPE$ space of 102 batches ($\iota = 102$) randomly generated with 10 variables ($\gamma = 10$) and 100 samples for each variable ($\kappa = 100$)

In order to evaluate the time consumption for all maintenance polices, we divided the randomly generated batches into two overlapping classes, which is the worst case scenario for a classifier. The overlapping between the two classes ($nClasses = 2$) is presented in Figure B.2, which presents the pair $SPE - T^2$ values of each batch generated.

B.2 Time comparison of maintenance policies

In order to compare the time consumption of IB2, IB3 (3.4), DROP4 (3.3) and IBUdG (3.5) we will compute the time they destine to update a case base. More specifically, we will count the time each strategy uses to retrieve the nearest cases from the case base, propose a new solution based on this information (reuse), check the validity of the solution (revise) and finally, update the case base whenever necessary (retain). Since the CBR cycle is dependant on the number of cases in the case base, we will compute these times from a case base with 100 ($nIni$) batches up to a case base with 10000 ($nEnd$) cases randomly generated using randomFill (Algorithm B.1). Regarding the retrieve procedure, we will use the combined index criterion (d_ϕ), described in subsec-

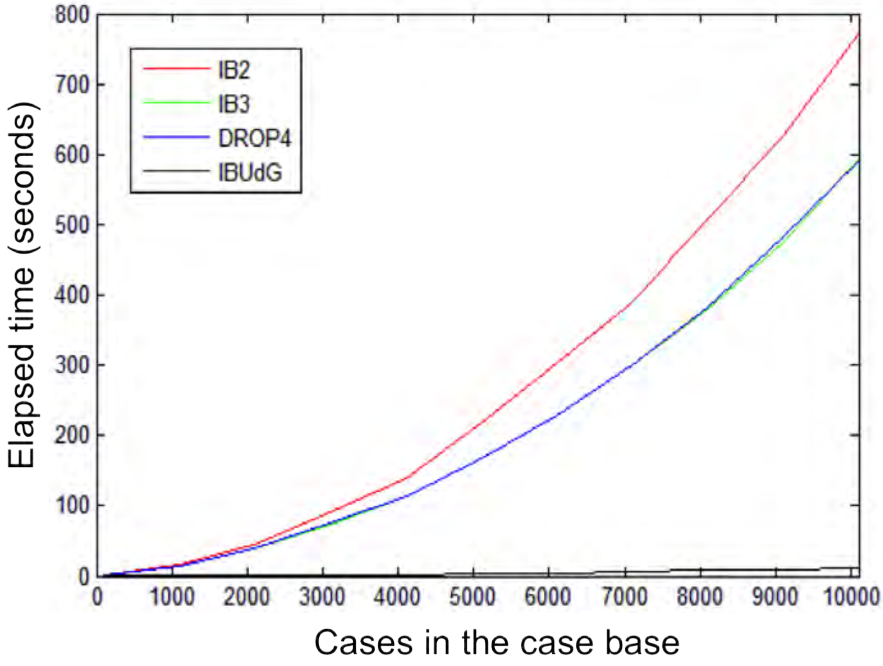


Figure B.3: Elapsed time to retrieve, reuse, revise and update the case base for each of the maintenance policies considered (IB2, IB3, IBUdG and DROP4)

tion 3.3.2.4 (3.16) to consider the information in both the principal component and residual subspace; and the reuse procedure will be the simple voting (SV) presented in subsection 3.4.1, since we are classifying cases within the case base. Figure B.3 presents the elapsed times for each maintenance policy (IB2, IB3, DROP4 and IBUdG). This time includes the time required to retrieve, reuse, revise and update the case base for every generated case base.

From Figure B.3 we can observe that IB2 is the policy that consumes most of the time, followed by DROP4 and IB3, and where IBUdG is the one with the lowest time consumption. Although IB2 is the simplest method (adds any misclassified case into the case base), and thus, it should be the method consuming the lowest time when compared to the other three methods. However, in our data set, both classes are completely overlapped, and consequently, the effect of blindly adding new cases (like IB2 does), causes that cases that were initially correctly classified are now misclassified, and thus, each iteration adds more and more cases. As a result, IB2 ends up being the method that consumes the most time among all four due to the increased time in the retrieve procedure (it is the method that stores more cases).

On the other hand, IBUdG is the method that takes less time among the four because of its simplistic rule (add the nearest enemies to each case if they are not already in the case base) when compared to IB3 and DROP4. In this case, DROP4 is hindered by the initial case base (remember that DROP4 checks if each case in the case base is necessary to classify its "associates"), and thus, most of the elapsed time for DROP4 is destined to check the importance of each individual case. Regarding IB3, the situation is completely opposite to DROP4: we start with an empty case base and we add new cases. However, the problem is to decide whether a new case has to be added, and since we check each of the cases originally in the case base, we have to explore all cases that were originally in the case base. Finally, and as could be observed in Table 4.5 in chapter 4, IBUdG keeps more cases than IB3 and DROP4, however, as can be deduced from the results in Figure B.3, this increase in number of stored cases does not result in a higher computation time.

All in all, we can say that for large case bases, the time elapsed to decide whether a new case has to be added or not becomes more important than the time used to find the nearest neighbours, even for incremental methods such as IB2 and IB3. Consequently, the retain policy is a trade-off between complexity of the inclusion (or removal) criterion and the number of cases that are kept in the reduced case base.

Universitat de Girona
Girona, Catalonia, Spain
http://www.udg.edu

iiia

eXiT
Control Engineering and
Intelligent Systems