Universidad de Valladolid

Escuela de Ingenierías Industriales

Departamento de Ingeniería de Sistemas y Automática

TESIS DOCTORAL:

MONITORING, FAULT DETECTION AND ESTIMATION IN PROCESSES USING MULTIVARIATE STATISTICAL TECHNIQUES

Presentada por Diego García Álvarez para optar al grado de doctor por la Universidad de Valladolid

Dirigida por:
Dra. María Jesús de la Fuente Aparicio

Valladolid, febrero de 2013.

Universidad de Valladolid

School of Industrial Engineering

Department of System Engineering and Automatic Control

PHD THESIS:

MONITORING, FAULT DETECTION AND ESTIMATION IN PROCESSES USING MULTIVARIATE STATISTICAL TECHNIQUES

A Thesis submitted by Diego García Álvarez for the degree of Doctor of Philosophy in the University of Valladolid

Supervised by:
María Jesús de la Fuente Aparicio, PhD

Valladolid, February 2013.

A la memoria de mi abuela Luisa

We can only see a short distance ahead,
but we can see plenty there that needs to be done.

Alan M. Turing (1912 - 1954)

Acknowledgments

Many people have contributed, in one form or another, with the development of this PhD dissertation. I would like to use these words to thank all of them and I hope not to forget anyone.

First of all, I would like to extend my sincerest thanks and appreciation to my supervisor, María Jesús de la Fuente, for guiding me in this way and for believing in me. Thank you very much for your valuable comments, constructive criticism and advice. I would also like to thank you for the hard work of reviewing and correcting this document.

I would also like to express my deepest gratitude to all the people I met in the Department of Applied Statistic, Operations Research and Quality at the Technical University of Valencia, especially to Alberto Ferrer for his professionalism. Your advice has been key to this work.

I want to thank Barry Lennox and all the people I worked with in the Control Systems Group at the University of Manchester.

I owe thanks to my colleagues and friends Alejandro Merino and Luis G. Palacín for providing the evaporating section of the sugar factory and the desalination plant and for their help. I also owe thanks to my friend Anibal Bregón for all his help with the possible conflicts technique and Rubén Martí for his help in the design of the soft sensors.

I want to thank the staff of the Department of Systems Engineering and Automation of the University of Valladolid.

I also want to thank the research grants program (FPU-UVA) of the University of Valladolid for giving me the financial support for this work, that has been co-funded by the Spanish research projects

DPI2006-15716-C02-02, DPI2009-14410-C02-02 and JCyL VA052A07.

I would also like to thank wholeheartedly the great family that I found in the Sugar Technology Centre (CTA) of the University of Valladolid. Thank you for making the workplace such a nice place.

I would like to thank all my dear friends for all the moments that we have lived and for all the moments that we are going to live.

I would also like to thank my brother Ciro for understanding me, for believing in me and for being by my side. Thank you very much.

Por último quisiera dar las gracias a mi querida familia, en particular a mi madre y a mi hermano Miguel por su esfuerzo, cariño y sacrificio. Muchas gracias.

Diego García Álvarez
Valladolid, February 2013

Abstract

Multivariate statistical techniques are one of the most widely used approaches in data driven monitoring and fault detection schemes in industrial processes. Concretely, principal component analysis (PCA) has been applied to many complex systems with good results. Nevertheless, the PCA-based fault detection and isolation approaches present some problems in the monitoring of processes with different operating modes and in the identification of the fault root in the fault isolation phase.

PCA uses historical databases to build empirical models. The models obtained are able to describe the system's trend. PCA models extract useful information from the historical data. This extraction is based on the calculation of the relationship between the measured variables. When a fault appears, it can change the covariance structure captured, and this situation can be detected using different control charts.

Another widely used multivariate statistical technique is partial least squares regression (PLS). PLS has also been applied as a data driven fault detection and isolation method. Moreover, this type of methods have been used as estimation techniques in soft sensor design. PLS is a regression method based on principal components.

The main goal of this Thesis deals with the monitoring, fault detection and isolation and estimation methods in processes based on multivariate statistical techniques such as principal component analysis and partial least squares. The main contributions of this work can be arranged in the three following topics:

- The first topic is related with the monitoring of continuous pro-

cesses. When a process operates in several operating modes, the classical PCA approach is not the most suitable method. In this work, an approach for monitoring the whole behaviour of a process, taking into account the different operating modes and transient states, is presented. The monitoring of transient states and start-ups is studied in detail. Also, the continuous processes which do not operate in a strict steady state are monitored in a similar way to the transient states.

- The second topic is related with the combination of model-based structural model decomposition techniques and principal component analysis. Concretely, the possible conflicts (PCs) approach is applied. PCs compute subsystems within a system model as minimal subsets of equations with an analytical redundancy property to detect and isolate faults. The residuals obtained with this method can be useful to perform a complete fault isolation procedure. These residuals are monitored using a PCA model in order to simplify and improve the fault detection task.

- The third topic addresses the estimation task in soft sensor design. In this case, the soft sensors of a real process are studied and improved using neural networks and multivariate statistical techniques. In this case, a dry substance (DS) content sensor based on indirect measurements is replaced by a neural network-based sensor. This type of sensors take into account more variables of the process and obtain more robust and accurate estimations. Moreover, this sensor can be improved using a PCA layer at the network input in order to reduce the number of inputs in the network. Also, a PLS-based sensor is designed in this topic. It also improves the sensor based on indirect measurements.

Finally, the different approaches developed in this work have been applied to several process plants. Concretely, a two-communicated-tanks system, the evaporation section of a sugar factory and a reverse osmosis desalination plant are the systems used in this dissertation.

Resumen en Español

En este apartado se incluyen el resumen, el índice completo de la tesis, un resumen de cada capítulo y las conclusiones en español.

Resumen

Las técnicas estadísticas multivariantes son probablemente unas de las técnicas basadas en datos más usadas en la detección y diagnóstico de fallos en procesos. Dentro de éstas, el análisis de componentes principales (PCA, por sus siglas en inglés) se ha aplicado en la monitorización de sistemas complejos mostrando resultados muy satisfactorios debido a su efectividad y su simplicidad. Sin embargo, los métodos de detección y diagnóstico basados en PCA pueden presentar problemas a la hora de monitorizar procesos con varios modos de operación, además, la fase de aislamiento de fallos no se realiza de una forma completa.

PCA usa datos históricos del proceso para construir modelos empíricos del mismo que son capaces de extraer las principales tendencias del proceso. Esta extracción se basa en el estudio de las relaciones entre las variables medidas del proceso. De esta forma, cuando ocurre un fallo en el proceso se genera un cambio en estas relaciones que puede ser detectado.

Otra de las técnicas estadísticas multivariantes ampliamente usada en la industria son los mínimos cuadrados parciales (PLS, por sus siglas en inglés), un método de regresión basado en componentes principales. PLS también puede usarse tanto para el diseño de técnicas de detección y diagnóstico de fallos como para la estimación de variables en el diseño de sensores software.

El objetivo principal de esta tesis es la monitorización, detección y diagnóstico de fallos y estimación en procesos continuos mediante el uso del análisis de componentes principales y los mínimos cuadrados parciales. Las principales contribuciones de este trabajo se pueden resumir en los tres puntos siguientes:

- El primero de ellos aborda la monitorización de procesos continuos cuando éstos operan en diferentes modos de operación. Ya que, cuando se da esta situación el enfoque clásico de la detección de fallos usando PCA no es el más adecuado. En este trabajo se presenta un modo de monitorizar este tipo de procesos teniendo en cuenta todos los modos de operación considerados incluyendo tanto los estados transitorios como las zonas del procesos donde no se opere en un estricto estado estacionario.

- En el segundo se contempla la combinación de métodos de detección y diagnóstico de fallos basados en modelos de primeros principios con PCA. La técnica basada en modelos usada es conocida como posibles conflictos (PCs). Los posibles conflictos calculan subsistemas dentro del modelo de ecuaciones con redundancia analítica, la cual, puede usarse para calcular residuos útiles para la detección y diagnóstico de fallos. En este caso, los residuos son monitorizados usando un esquema PCA que permite una etapa de detección mejor y más sencilla.

- En el tercero se estudia la etapa de estimación en el diseño de sensores software. Esta aportación se centra en la mejora de sensores software en procesos reales. Los resultados de una estimación basada en medidas indirectas se han mejorado usando redes neuronales y PCA, por un lado, y PLS, por otro.

Los diferentes métodos presentados en este trabajo se han aplicado a tres plantas de proceso: un sistema de dos tanques comunicantes, la sección de evaporación de una industria azucarera y una planta de desalinización basada en ósmosis inversa.

Índice de la tesis

Resumen de los capítulos

Capítulo 1. Introducción y objetivos

La seguridad en la producción es uno de los principales objetivos de la industria moderna. Las técnicas de control actuales permiten que los procesos funcionen con un alto grado de autonomía, sin embargo, la aparición de causas especiales en los procesos debido, por ejemplo, a fallos pueden provocar que éstos dejen de operar de manera óptima o incluso que la seguridad de los operarios o del medio ambiente se pueda ver comprometida. Por estos motivos, el diseño de sistemas de monitorización, detección y diagnóstico de fallos pueden ser vistos como un paso hacia la fiabilidad y seguridad en la industria.

Otro de los objetivos fundamentales de la industria es la calidad. Normalmente esta medida se evalúa midiendo algunas variables a la salida del proceso, sin embargo, el uso de técnicas de monitorización en línea del procesos y estimación de las variables de calidad pueden ser herramientas útiles para la predicción de la calidad. Ya que, si esta predicción se realiza de manera temprana se pueden ahorrar muchas pérdidas en la producción.

El principal objetivo de esta tesis es el estudio y diseño de técnicas de monitorización, detección y diagnóstico de fallos y estimación en procesos continuos mediante técnicas estadísticas multivariantes principalmente el análisis de componentes principales (PCA).

Los métodos de detección de fallos basados en PCA generan un modelo empírico a partir de datos pasados del proceso en condiciones normales de operación. PCA es capaz de extraer de estos datos, normalmente altamente correlacionados y afectados por ruido, las variables latentes que mejor expliquen el comportamiento del proceso. Mediante la monitorización de estas variables usando estadísticos de control, se puede inferir si alguna causa especial está afectando al proceso.

La detección de fallos en procesos continuos usando PCA se ha conseguido con éxito cuando éstos operan en un único estado estacionario, ya que los cambios en los modos de operación pueden romper el modelo de correlación extraído por el PCA y ser confundidos con la detección de fallos. Además, los estados transitorios por los que el proceso atraviesa en estos cambios tienen una naturaleza no lineal y su monitorización con PCA, basado una transformación lineal, no es la más adecuada.

Por todo ello, esta tesis ahonda en la detección de fallos en procesos continuos que no operan en un único modo de operación o en un estado estrictamente estacionario. También, se han desarrollado diversos sensores software basados en técnicas estadísticas multivariantes para la estimación de variables de calidad en la industria azucarera.

Organización de la tesis

El documento de tesis esta organizado de la siguiente forma. En el capítulo 2 se presenta el estado del arte de dos campos muy relacionados: el control estadístico de procesos y la detección y diagnóstico de fallos. Además, también se describen los métodos estadísticos basados en datos como el análisis de componentes principales (PCA) y los mínimos cuadrados parciales (PLS).

En el capítulo 3 se describen varios métodos basados en PCA. Estos métodos son variantes del enfoque clásico del PCA para la detección de fallos. Las variantes de estos métodos tratan de resolver algunos de los problemas que presenta el enfoque clásico. Todos ellos se han aplicado a una misma planta para poder comparar los resultados. Además, este capítulo deja de manifiesto el problema de la monitorización de estados transitorios.

El capítulo 4 presenta un método para la monitorización de estados estacionarios mediante el uso de una variante del PCA denominada PCA desplegado (UPCA, por sus siglas en inglés). UPCA es un método bien conocido para la monitorización y detección de fallos en procesos *batch* o por lotes. Estos procesos se caracterizan, al igual que los estados transitorios, por su no linealidad.

En el capítulo 5 el método presentado en el capítulo anterior se aplica a una planta desalinizadora de agua mediante ósmosis inversa. Este tipo de plantas, debido a los ciclos de limpieza que deben realizarse en varios de su componentes, no operan en un estado estacionario estricto. Por este motivo el uso del enfoque clásico de PCA no es el más adecuado.

El capítulo 6 plantea la combinación del PCA con un método basado en la descomposición estructural de modelos. Dicho método es conocido como posibles conflictos (PCs). Con este tipo de métodos se pueden generar residuos que permiten el diagnóstico de fallos. En este caso, los residuos son tratados con PCA para simplificar y mejorar la fase de detección.

En el capítulo 7 se aborda el diseño de sensores software. En concreto, se estudia la estimación del contenido en materia seca en una industria azucarera. Los sensores diseñados están basados en redes neuronales, la combinación de éstas con PCA y PLS.

Finalmente, en el capítulo 8 se presentan las principales contribuciones de esta tesis. También se describen las conclusiones y las posibles líneas futuras de investigación que plantea este trabajo.

Capítulo 2. Estado del arte

En este capítulo se presentan dos campos de la ingeniería: la detección y diagnóstico de fallos y el control estadístico de procesos. Ambos campos están muy relacionados cuando la detección y diagnóstico de fallos se enfoca con métodos estadísticos multivariantes como el análisis de componentes principales (PCA) y los mínimos cuadrados parciales (PLS).

Una vez presentada el área de la detección y diagnóstico de fallos, se presenta una clasificación de los diferentes técnicas existentes. Entre las diferentes clases de métodos se hace hincapié en los métodos basados en datos históricos del proceso, y a su vez, dentro de este grupo se hace referencia a los métodos estadísticos para la detección de fallos, ya que tanto PCA como PLS se engloban dentro de este grupo.

Una vez descritos los métodos estadísticos, se detalla el PCA en profundidad para poder entender el resto de la tesis. Se presenta tanto matemáticamente como gráficamente. Además, también se explica desde el punto de vista de la detección y diagnóstico de fallos mediante la descripción de los gráficos de control más usados en la fase de detección de fallos y los gráficos de contribución usados para identificar las variables del proceso relacionadas con el fallo detectado.

Por último, se presenta de una forma más resumida los mínimos cuadrados parciales y como éstos se pueden aplicar para la detección y diagnóstico de fallos en procesos.

Capítulo 3. Detección de fallos en procesos continuos con PCA

En este capítulo se ahonda en la detección y diagnostico de fallos en procesos continuos con métodos basados en PCA. Por un lado se pretende mostrar las debilidades que los métodos clásicos basados en PCA pueden presentar, y por otro, mostrar diferentes métodos propuestos para solventar estas debilidades.

Los métodos descritos en este capítulo son los siguientes: PCA dinámico (DPCA), que se basa una configuración dinámica de la matriz de datos; PCA adaptativo (APCA), que reformula el modelo PCA cuando se detecta un fallo; PCA multi-escala (MSPCA), basado en la descomposición en frecuencias de las señales medidas del proceso; PCA pesado exponencialmente (EWPCA), donde los valores medidos en el proceso se añaden al modelo PCA; PCA con análisis externo (PCAEA), donde a la matriz de datos se le extrae la influencia de las perturbaciones medibles; y por último, una formulación de PCA no lineal basado en redes neuronales autoasociativas.

Cada uno estos métodos se ha aplicado a una misma planta, en este caso, un modelo de simulación de una planta de dos tanques comunicantes y se ha llevado a cabo un estudio comparativo basado en diferentes parámetros como el ratio de falsas alarmas o la adaptación a fallos.

Como conclusión de este capítulo se hace énfasis en la imposibilidad de los métodos presentados para monitorizar los estados transitorios debidos a cambios en el modo de operación.

Capítulo 4. Monitorización de estados transitorios con PCA

Este capítulo enfatiza en la monitorización y diagnóstico de fallos en estados transitorios y puestas en marcha.

Normalmente, cuando se produce un cambio en el modo de operación de un proceso, éste atraviesa un estado transitorio. Esta zona de operación se suele caracterizar por su no linealidad. PCA está basado en una transformación lineal y no presenta buenos resultados cuando se aplica a este tipo de situaciones.

En este capítulo se propone el uso un método aplicado a procesos *batch*, que son también no lineales, para monitorizar las transiciones entre modos de operación con PCA. A este método se le conoce como UPCA.

El método UPCA se describe en profundidad haciendo hincapié en las principales tareas que hay que llevar a cabo para poder diseñar un sistema de este tipo. Éstas son: el despliegue de la matriz de datos, el alineamiento de las señales para poder llevar a cabo dicho despliegue, la imputación de los datos desconocidos en la fase de monitorización y las consideraciones a tener en cuenta para el cálculo de los estadísticos de control y los diagramas de contribución.

El método se aplica a dos procesos: la planta de laboratorio de los dos tanques comunicantes y el modelo en simulación de la sección de evaporación de una industria azucarera. Además, se realiza un estudio de como puede afectar la imputación de los datos en la fase de detección. Para ello se presentan cinco de ellos y se evalúan los resultados.

Capítulo 5. Monitorización y detección de fallos en procesos continuos con un comportamiento no estacionario

En este capítulo se propone el uso del método UPCA, descrito en el capítulo anterior, para monitorizar una planta de desalinización de agua. En este caso, se utiliza un modelo de simulación de una planta de mediana escala de desalinización de agua mediante ósmosis inversa. Debido a la acumulación de depósitos de partículas en varios de sus componentes, como los filtros o las membranas, se requieren ciclos de limpieza para que la planta opere dentro de unos rangos establecidos. Estos ciclos de limpieza provocan que el modo de operación no tenga las mismas características cuando los componentes acaban de ser limpiados que cuando han retenido muchas partículas.

Por estos motivos, la monitorización de este tipo de plantas con un esquema clásico de PCA no es el más adecuado debido al alto ratio de falsas alarmas que aparecen en su monitorización ya que no se observa un estado estacionario estricto en su comportamiento.

La monitorización basada en UPCA muestra mejores resultados en comparación que el enfoque clásico de PCA desde el punto de vista del ratio de alarmas producidas.

Además, debido a que los ciclos de limpieza de los distintos componentes no están sincronizados, no se puede hacer un único modelo de la planta y se recurre a un esquema de detección distribuida con un modelo UPCA para cada uno de los componentes que requieren ciclos de limpieza.

Capítulo 6. Combinación de técnicas de descomposición estructural de modelos y PCA

En este capítulo se propone una combinación de métodos basados en modelos de primeros principios y métodos estadísticos para la detección y diagnóstico de fallos.

La técnica basada en modelo usada es el método de los posibles conflictos. Los posibles conflictos es un método basado en la descomposición del modelo de primeros principios de la planta en submodelos más pequeños que permiten estimar variables del proceso de dos modos diferentes y así poder construir un residuo calculando las discrepancias entre ellos. Estos residuos permiten aislar los fallos de una manera más completa que las técnicas estadísticas por si solas.

Los residuos generados en este caso no se monitorizan de uno en uno, sino que se propone monitorizarlos de una forma global mediante la construcción de un modelo estadístico basado en el enfoque clásico de PCA. Este etapa adicional permite tanto simplificar la etapa de detección como hacerla más robusta.

La propuesta de integración se aplica a la planta de laboratorio de los dos tanques comunicantes presentando buenos resultados tanto en la fase de detección como en la de diagnóstico.

Capítulo 7. Diseño de sensores software mediante el uso de técnicas estadísticas multivariantes

En este capítulo se plantea el diseño de un sensor software para la estimación del contenido de materia seca en la sección de evaporación de una industria azucarera.

En la industria azucarera el contenido de materia seca se suele estimar mediante la utilización de las propiedades coligativas de las disoluciones. Es este caso en particular, la variable estimada es muy sensible a pequeñas variaciones en los valores de presión y temperatura usados para su cálculo.

Como una primera mejora a esta tipo de sensores, se propone usar redes neuronales realimentadas usando, además de los valores de presión y temperatura de la planta, muchas otras variables medidas en el proceso a fin de conseguir modelos más robustos para la estimación.

Para simplificar la red neuronal sin necesidad de eliminar variables de la estimación, se propone entrenar la red neuronal con las variables latentes del sistema de variables del proceso usando PCA. Esto permite reducir drásticamente el número de entradas de la red neuronal conservando todas las medidas del proceso y sin una gran carga computacional añadida.

Por último se propone el uso de PLS para generar un modelo de regresión entre las variables medidas y las variables estimadas. Este método es el más sencillo computacionalmente y además permite analizar cuales son las variables más relacionadas con la variable a estimar. Además, tanto esta solución como la anterior permiten el diseño de un método de monitorización y detección de fallos con un mínimo coste adicional.

Los tres métodos propuestos arrojaron mejores resultados que el uso de medidas indirectas.

Capítulo 8. Conclusiones

En esta tesis se estudian las tareas de monitorización, detección de fallos y estimación en procesos continuos mediante el uso del análisis de componentes principales. En esta última sección se presentan las principales aportaciones de esta tesis, las conclusiones que se derivan del trabajo realizado y las futuras líneas de trabajo.

Resumen de contribuciones

1. Estudio de los métodos de monitorización de procesos basados en PCA [García-Álvarez and Fuente (2008), García-Álvarez (2009) y García-Álvarez and Fuente (2011)]. Esta primera contribución se centra en la detección de fallos basada en PCA. El método clásico de detección de fallos mediante PCA presenta varias limitaciones, como la imposibilidad de detectar varios fallos consecutivos o su baja idoneidad para la detección de fallos en estados transitorios. Por estos motivos existen diferentes variaciones al enfoque clásico del PCA que pretenden solventar estas deficiencias. En este trabajo, varios de estos métodos se han estudiado, implementado y comparado. Este estudio comparativo se ha llevado a cabo usando un modelo de simulación del sistema de los dos tanques comunicantes y puede ser útil para la decisión de qué método usar dependiendo del tipo de sistema. Además, este estudio pone de manifiesto que la monitorización de estados transitorios y puestas en marcha de larga duración, normalmente de naturaleza no lineal, no se pueden monitorizar con los métodos basados en PCA estudiados.

2. Monitorización de estados transitorios mediante el uso de PCA desplegado (UPCA). [García-Álvarez et al. (2012b) y García-Álvarez et al. (2012c)]. Esta segunda contribución se centra en la monitorización de estados transitorios y puestas en marcha. Mientras que estas zonas de operación presentan una naturaleza no lineal, el PCA está basado en una transformación lineal, por este motivo, las técnicas basadas en PCA clásico no son la solución más adecuada para este tipo situaciones. En este trabajo, una técnica basada en PCA pensada para la monitorización de procesos *batch* se aplica a estas zonas de operación. Esta técnica es conocida como UPCA. Para diseñar un método de detección de fallos basado en esta técnica se deben tener en cuenta diversas consideraciones, como el desplegado de la matriz de datos, la

sincronización o la imputación de datos en la monitorización en línea. En este trabajo, se presentan y explican todas estas consideraciones. Esta parte de la tesis se puede concebir como una guía para el diseño de métodos de detección de fallos basados en UPCA para estados transitorios. El método descrito se aplica a la planta de laboratorio de los tanques comunicantes y un modelo de simulación de la sección de evaporación de una industria azucarera. En ambos ejemplos se aplican diferentes métodos de imputación a fin de descubrir si éstos pueden estar relacionados con el retardo en la detección del fallo.

3. Monitorización del comportamiento completo de procesos continuos considerando los diferentes modos de operación y estados transitorios. [García-Álvarez et al. (2009a) y García-Álvarez et al. (2010b)]. En esta contribución el principal objetivo es el diseño de métodos basados en PCA para la monitorización del comportamiento completo de procesos continuos, incluyendo todos los estados estacionarios y transitorios que puedan aparecer durante la operación. Por lo tanto, se identifican los diferentes modos de operación y estados transitorios y se construye un modelo PCA o UPCA para cada uno de ellos. Para poder llevar a cabo esta operación, es necesaria la existencia de una base de datos de las diferentes zonas de operación consideradas. Estas consideraciones se deben implementar fuera de línea. Después, en línea con el proceso, el bloque de detección de fallos debe identificar el estado actual el proceso y seleccionar el modelo PCA o UPCA para monitorizar cada una de las zonas de operación consideradas.

4. Monitorización y detección de fallos en procesos continuos sin un comportamiento estacionario estricto. [García-Álvarez et al. (2011b), García-Álvarez et al. (2011d) y García-Álvarez and Fuente (2013)]. En este caso, el método UPCA es aplicado a procesos continuos que no operan en un estado estacionario estricto. En esta contribución el caso de estudio utilizado es una planta desalinizadora de agua basada en ósmosis inversa. Este tipo de plantas requieren la ejecución de fases de limpieza en varios de

sus componentes debido a la acumulación de partículas en los diferentes filtros y membranas. Las diferencias en la operación del proceso cuando la planta ha acumulado muchas partículas y cuando acaba de ser limpiada hacen que no opere en un estricto estado estacionario, puesto que las variables medidas relacionadas con presiones y temperaturas presentan grandes diferencias. El enfoque clásico del PCA no es el más adecuado en este tipo de procesos, por lo que se aplica UPCA mediante un agrupamiento de los datos basado en los diferentes ciclos de limpieza.

5. Combinación de técnicas de descomposición de modelos y PCA para la detección y el diagnóstico de fallos. [Bregón et al. (2010)]. En esta contribución se plantea la combinación de técnicas basadas en datos y basadas en modelos de primeros principios para la monitorización del comportamiento completo del proceso. Se ha usado un método de descomposición de modelos para generar residuos, en concreto, una técnica basada en el análisis estructural conocida como los posibles conflictos. Usando los residuos calculados con esta técnica se puede diseñar un método completo de diagnóstico de fallos. Además, si el modelo usado es dinámico y preciso, los estados transitorios se pueden modelar y no ser detectados como fallos. Esta propuesta de integración propone monitorizar los residuos usando un esquema basado en PCA. Esta integración permite simplificar y mejorar la etapa de detección de fallos, ya que, la monitorización se reduce a dos gráficos de control que permiten detectar pequeñas desviaciones en los residuos. La propuesta de integración se ha aplicado a la planta de los dos tanques comunicantes.

6. Diseño y mejora de sensores *software* basados en técnicas estadísticas multivariantes y redes neuronales artificiales. [García-Álvarez et al. (2012d) and Martí et al. (2011)]. En este caso, se han diseñado varias propuestas de sensores *software* para la estimación del contenido de materia seca en la sección de evaporación de una industria azucarera. El método más usado en este tipo de medidas se basa en principios físico-químicos, pero en este caso en particular, se obtienen respuestas muy sensibles

a pequeños cambios en las medidas usadas para la estimación. Por ello, una solución adecuada puede ser el uso de redes neuronales, ya que pueden usarse más variables medidas y obtenerse respuestas más robustas. Esta configuración puede simplificarse usando las variables latentes del proceso en lugar de todas ellas mediante el uso de PCA. Otro de los diseños propuestos se basa en el uso de la regresión PLS. En general, los métodos basados en métodos estadísticos multivariantes generan respuestas más robustas y no es muy costoso construir sistemas de supervisión y detección de fallos en los sensores relacionados con la medida.

Otros trabajos publicados durante el desarrollo de esta tesis doctoral se resumen en las siguientes contribuciones:

7. Detección de fallos en estaciones depuradoras de aguas residuales. En este caso se han desarrollado diferentes configuraciones, por un lado, PCA se aplica para la fase de detección de fallos y el discriminante de Fisher para el aislamiento de éstos. [García-Álvarez et al. (2009c) y García-Álvarez et al. (2009b)]. Otra aplicación del discrimante de Fisher en una planta real se ha desarrollado en Fuente et al. (2009). Por otro lado, PCA se ha usado para monitorizar los residuos obtenidos entre el sistema real y un modelo neuronal de la planta depuradora. [Fuente et al. (2012)].

8. Monitorización de controladores predictivos mediante el uso de PCA. En esta contribución se ha desarrollado una herramienta para la monitorización de controladores predictivos usando PCA. El análisis de las contribuciones se ha usado para tratar de identificar las causas del mal funcionamiento del controlador. [García-Álvarez et al. (2012a)].

9. Mejora de la estimación de parámetros mediante el análisis de subsistemas minimales analíticamente redundantes. [García-Álvarez et al. (2010a) y García-Álvarez et al. (2011a)]. Estimación de parámetros biocinéticos usando diferentes enfoques [García-Álvarez et al. (2010c)].

10. Aplicación de métodos de detección y diagnóstico basados en modelos de primeros principios aplicadas a diferentes plantas y otros trabajos relacionados con este área. [Bregón et al. (2007), Bregón and García-Álvarez (2007), García-Álvarez and Bregón (2008) y García-Álvarez et al. (2011c)].

Conclusiones

En base a la información, las discusiones y los resultados de este trabajo de tesis, las principales conclusiones se pueden englobar en los siguientes puntos:

- Los métodos de detección de fallos basados en el enfoque clásico de PCA no son los más adecuados para la monitorización de estados transitorios y puestas en marcha. Para este tipo de zonas de operación son más idóneos los métodos basados en UPCA.

- La monitorización de los procesos continuos que no operan en un estado estacionario estricto obtiene mejores resultados cuando se usan enfoques basados en UPCA en lugar de enfoques basado en PCA.

- La integración de técnicas basadas en datos con técnicas basadas en modelos de primeros principios puede mejorar y simplificar la detección de fallos y permite diseñar esquemas completos de diagnóstico de fallos.

- El uso de técnicas estadísticas multivariantes se puede usar para mejorar la estimación y producir medidas más robustas en el diseño de sensores *software*.

Direcciones de trabajo futuro

En este apartado se citan algunas de las futuras líneas de trabajo que se pueden abordar como continuación al trabajo de tesis descrito en este documento.

- La monitorización de estados transitorios mediante el uso de UPCA se estudia en detalle en este trabajo. Dentro de este campo, se hace hincapié en la relación entre el método de imputación usado y el retardo en la detección. Como línea de trabajo futuro se podría estudiar la relación entre el retardo de detección y los diferentes métodos de despliegue. También se podrían buscar otras variables indicadoras para el alineamiento de estados transitorios de sistemas de segundo orden.

- El análisis de las contribuciones es un primer paso en la fase de diagnóstico de fallos, pero no es un método completo para el aislamiento de fallos. El diagnóstico y el aislamiento del fallo puede ser una línea de trabajo futuro. El comportamiento que los diagramas de contribución experimentan cuando se detecta un fallo puede ser extraído, por ejemplo con una red neuronal.

- Otra de las líneas futuras de trabajo propuestas está relacionada con el control de procesos. La regresión PLS puede usarse para construir modelos estadísticos que pueden aplicarse en el diseño de controladores predictivos. Además, usando este tipo de modelos, pueden diseñarse sin un gran coste adicional herramientas de detección de fallos. Si el modelo PLS se implementa de una forma adaptativa, el controlador predictivo puede adaptarse a los fallos y diseñarse un sistema tolerante a fallos.

- En la combinación de técnicas de detección de fallos basadas en modelo y técnicas estadísticas multivariantes se pueden considerar varias líneas de trabajo. Por un lado, en el ejemplo usado se puede diagnosticar más fallos. También se pueden analizar otras plantas de mayor tamaño. Por otro lado, la estimación usando PLS se podría usar para estimar variables que no se pueden calcular usando las ecuaciones del modelo debido a la causalidad matemática del mismo.

- En este trabajo, la estimación usada en el diseño de sensores *software* usando PLS se ha aplicado de la forma más sencilla. De igual modo que en el PCA, el PLS se puede implementar de una forma dinámica. Esta configuración tiene en cuenta la relación de las variables medidas en la respuesta en el instante de tiempo actual y en instantes pasados. También, el uso de las redes neuronales en este trabajo puede mejorarse usando otras configuraciones de redes realimentadas como las NARX (*Nonlinear AutoRegresive with eXogenous Response*) o las NOE (*Nonlinear Output Error*) en lugar de las redes de Elman.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and objectives

1.1 Motivation

Safe production is one of the most important objectives of modern industry. The control engineering field has successfully resolved many industrial challenges. Advanced control theory has been applied to many processes with promising results. However, special causes can appear during the process running. These special causes can cause out of control behaviour and the operators and the environment may run risks that lead to disastrous consequences. Special causes are usually unpredictable. This unpredictability makes it necessary to develop *on-line* tools to monitor and detect faults in the processes in order to guarantee safe operation.

Another of the principal aims of industry is the production of high quality products. In many cases, the quality is evaluated by measuring some variables in the process outlet. The use of monitoring techniques and quality variable estimation can be a suitable procedure for the prediction of the quality and, therefore, to avoid losing money. Monitoring and estimation techniques can detect special causes which may not present disastrous consequences, but they can present adverse effects in the quality.

The main focus of this dissertation is the study of monitoring, fault detection and estimation approaches in continuous processes using multivariate statistical techniques, concretely, principal components analysis (PCA).

The *on-line* PCA-based methods for fault detection require the use of a statistical model established using data collected from the process in normal operating conditions (NOC). Discrepancies between the process data and the established statistical model can imply the presence of a fault. The *on-line* supervisory task can be compressed into two statistical control charts which give an idea of the discrepancy between the monitored process and the NOC process considered. Also, this type of techniques allow the variables related with the detected special cause to be identified.

There are many fault detection and isolation techniques that have been studied from different scientific fields. These techniques have been classified by several authors in different ways. Venkatasubramanian et al. (2003c,b,a) divide these techniques into three classes: quantitative model-based methods, qualitative models and search strategies, and process history based methods. The PCA-based fault detection method falls into the process history based methods due to the fact that the statistical model performed to capture the NOC process trends is performed using past data under normal conditions.

The proposal of this Thesis deals with the process monitoring of continuous processes using principal components analysis and integration with other fault detection and isolation techniques. Also, the *on-line* estimation of process variables or soft sensor design using multivariate techniques, like PCA or partial least squares (PLS), is studied.

Figure 1.1 shows a scheme of the classical PCA approach for monitoring and fault detection. The PCA model is a statistical model which extracts the principal trends in the process data. The history data measured in the process is usually highly correlated and affected by several sources of noise. The PCA techniques are able to extract these process trends or latent variables from the process data. The PCA model is computed *off-line*. The process can be monitored *on-line* using two control charts based on the established PCA model. Looking at these charts, abnormal behaviour or faults can be detected. Also, using the contribution analysis tools, the process variables related with the fault can be identified. This identification of the variables related with the fault detected can be seen as a first approximation to fault isolation.

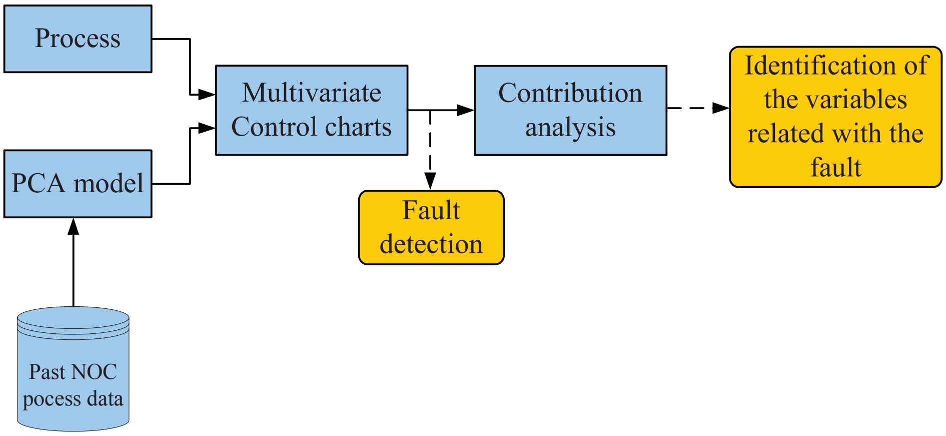

Process
PCA model
Past NOC pocess data
Multivariate Control charts
Contribution analysis
Identification of the variables related with the fault
Fault detection

Figure 1.1: Scheme of fault detection and isolation using PCA.

The PCA approaches have normally been applied to processes running in steady states. The PCA method is based on a linear transformation and it can be a suitable technique for detecting faults in steady states. However, processes sometimes go through several operating modes. The transient states between operating modes are usually non-linear zones. It can be difficult to distinguish between the transient states and the faults. This situation is due to the fact that the non-linearities of the transient states are detected as faults by the linear PCA methods. The situation is similar when the start-ups of processes are monitored.

The main proposal of this PhD Thesis is to deal with the estimation, monitoring and fault detection of whole processes, including the start-ups and the transient states, using principal component analysis.

1.2 Guidelines and main hypotheses

The main guidelines that can be established for this dissertation, taking into account the problems explained in the previous section, are the following:

- Develop techniques based on PCA, or the combination with other fault detection techniques, to monitor and detect faults in continuous processes considering the different steady states, transient states and start-ups.
- Use PCA to monitor continuous processes that do not present a strict steady behaviour.
- Apply the multivariate statistical techniques to estimate quality variables in real processes.

Thus, the main hypotheses of the dissertation are directly related to monitoring and estimation in continuous processes:

> *"The whole behaviour of a continuous process, considering that all its operating points, transient states and non-steady zones can be monitored using the principal component analysis approach."*
>
> *"The use of multivariate statistical techniques to improve and simplify the estimation of quality variables in soft sensor design."*

These hypotheses have several associated objectives. These objectives can be seen as the different stages fulfilled in this dissertation. The next section describes these objectives.

1.3 Main objectives

The main objectives of this work are to:

1. Test different PCA-based methods in continuous processes with several operating modes and transient states.

   Several authors have proposed different variations of the classical PCA approach in order to solve some of the drawbacks of this method; such as the detection of consecutive faults or the monitoring of transient states. The first task related with this objective is to study and implement the most representative methods. As a second task, the implemented methods will be compared. This comparison will be made by applying all the implemented methods to the same plant and analyzing the results obtained.

2. Design and develop methods for monitoring and detecting faults in processes with several operating points and transient states.

   The PCA approach is usually applied to continuous processes in a specific operating point. If the PCA model is established using a specific operating point and, in the monitoring phase, the process operation changes to another operating point, this new situation can be detected as a fault. There are several proposals for dealing with this drawback (Hwang and Han, 1999; Tien et al., 2004).

   Also, the changes between the different steady operating modes cause transient states. These transient states usually present non-linear behaviours which are detected as faults. This situation is due to the fact that the PCA is based on a linear transformation and it is not the best solution for monitoring these parts of the process behaviour.

   Batch processes are usually non-linear processes. Nomikos and MacGregor (1995) propose several modifications to the classical PCA approach to monitor these non-linear processes.

   Within this objective, a method which combines all these considerations can be established to monitor the whole behaviour of a plant, taking into account the different operating points, steady states and transients states.

3. Monitor continuous processes which do not strictly run in a steady state.

   It is normal that continuous processes run in a specific operating point with a steady behaviour. In these cases, the classical PCA approach is a suitable method for plant monitoring and fault detection. However, some continuous processes do not present this steady behaviour throughout the entire running of the process. This type of processes present non-linearities and, therefore, the classical PCA approach is not the best solution for monitoring this type of processes.

In this objective, this type of processes are studied and a PCA-based method similar to the method applied to transient states will be considered.

4. Combine PCA-based methods with model-based methods to improve the diagnosis task.

   The mentioned contribution analysis is a useful tool to identify the variables related to the detected fault. It can be a good first approximation to the isolation procedure. However, the isolation phase goal is not only to identify the variables related to the fault, but also to isolate the root of the fault.

   Diagnosis approaches for *on-line* fault diagnosis based on analytical models (Blanke et al., 2003; Gertler, 1998) provide robust methods to isolate the different faults. These procedures allow the fault root to de identified.

   On the other hand, PCA-based methods allow the process state to be supervised by only looking at two control charts. Also, these charts are based on the PCA model, which extracts the latent variables of the process, and presents good results in fault detection.

   In this objective, methods based on the combination of both techniques are investigated in order to explore the fault isolation task and to simplify and improve the fault detection phase.

5. Design soft sensors for the estimation of quality variables in real processes based on multivariate statistical techniques.

   In this objective, the design of soft sensors is addressed. In this case, the sensors will be designed using multivariate statistical process techniques. The purpose of this design is to estimate a valuable quality variable and, also, to try to detect abnormal behaviour thanks to the above mentioned properties of these statistical techniques.

In this case, partial least squares is a suitable approach due to the fact that it can be applied as a regression method for highly correlated data. Also, the compression feature of the PCA related to the extraction of the latent variables can be investigated to improve other data-based soft sensor design methods.

6. Implement and test all the proposals in simulated and real systems.

   The last objective of this dissertation is to validate and verify the proposed approaches in simulated and real processes. As a first benchmark, a two-communicated-tanks pilot plant can be used. In this laboratory plant, different operating points and transient states can be established. Also, a first principles model of the plant can be used.

   Two complex systems are considered in this dissertation. On the one hand, the simulation model of the evaporation section of a sugar factory, where different operating points and transient states can be considered. Also, real data from this plant are available. These data do not consider different operating points, but they can be useful for designing soft sensors due to the fact that quality variables are available. On the other hand, a simulation model of a water desalination plant is used. This process is a continuous process, but it requires cleaning cycles for some of its components, which means that the process does not run in a strict steady state.

1.4 Organization

This dissertation is organized as follows. Chapter 2 presents the state of the art on statistical process control and the fault detection and diagnosis field. Concretely, the current background of the statistical process history-based method is presented. Then, the principal component analysis approach from the point of view of monitoring and fault detection is explained in detail. Also, the main features of the partial least squares approach are presented.

Chapter 3 presents several PCA-based methods. These methods try to improve the fault detection features of the classical PCA approach. Every method is explained in detail. Then, a comparative analysis of the explained method is performed. This comparison consists of applying the method to the same simulated plant and evaluating several parameters, including the transient states monitoring capacity. The plant is a two-communicated-tanks system.

Chapter 4 addresses the problem of the monitoring process with several operating points and transient states. In this chapter, transient states are monitored using a well-known scheme for batch process monitoring, known as unfolded PCA or UPCA. All the considerations necessary for a fault detection tool using this methodology are explained in detail. Some considerations of this approach, such as the data alignment or the imputation, are undertaken. Two case studies are considered; the two-communicated-tanks real process plant and the simulation model of the evaporation section of a sugar factory.

Chapter 5 proposes the use of the UPCA approach for continuous processes that do not operate in an ordinary steady state. In this chapter, the simulation model of a water desalination plant is used. In this type of plants, the process operation is affected by the cleaning of some of its components. These cleaning phases change the operation conditions in the process, so it the process does not run in a strict steady state.

Chapter 6 presents a monitoring and fault detection method based on the combination of structural model decomposition techniques and PCA. In this case, the PCA is applied to the residuals obtained by structural analysis techniques. Specifically, the possible conflicts (PCs) technique is used. Possible conflicts are able to decompose the system model into the smaller subsets of equations required for fault diagnosis. The combination approach is applied to the real process of the two-communicated-tanks system in order to monitor transient states and improve the fault isolation phase.

Chapter 7 is based on the design of soft sensors for the estimation of quality variables. Concretely, a soft sensor for the estimation of the dry substance (DS %) content in the evaporation section in a sugar factory in designed. In this case, four different sensors have been developed and compared. The first one is based on indirect measurements, the second uses neural networks, the third uses neural networks whose inputs are the scores calculated by means of principal component analysis, and the last is based on the partial least squares regression.

Finally, chapter 8 presents the conclusions and summarizes the contributions of this dissertation. Also, the future directions to improve the current work are addressed.

1.5 Notation

The mathematical notation used in this document is the following: bold capital letters for matrices, bold lower case letters for vectors and cursive letters for scalars.

Chapter 2

State of the art

2.1 Introduction

The safe production of high quality products is one of the main objectives of industry. Control engineering techniques and tools have improved and resolved many of these aims. However, special causes can occur in processes and these processes can present poor quality results, out of control behaviour and even the operators and the environment can be at risk. These unpredictable special causes make it necessary to develop *on-line* tools to detect them and guarantee the quality of the results and safe operation.

Fault detection and isolation is a wide and complex field that has been studied by many authors using several perspectives. Multivariate statistical monitoring methods for the analysis of process data have been applied successfully in fault detection tasks (Venkatasubramanian et al., 2003c,b,a).

2.2 Statistical process control

Quality control and fault detection are closely related concepts. Both tools try to detect malfunctions and the consequential economic damage.

Statistical quality control (SQC) or statistical process control (SPC) is a method of continuous improvement of the quality based on a systematic reduction of the sources of variability which have the most influence on the final product (Vilar, 2005).

In the context of statistical process control, a system can be defined as a set of interactive components with a particular purpose. Systems receive inputs from the environment, transform the inputs into outputs and introduce these outputs into the system.

The transformation task cited above is also called the process. The process is the set of procedures or operations which transform the inputs into the outputs. The main objective of the process is to produce outputs by adding a certain value to the inputs.

Also, in this context, a variation can be defined as the changes in the value of a measured feature, where a feature is the response of a certain process. The value of a feature is considered as the optimal value when its value is within established range of tolerances. The variation in a feature, even within the range or tolerances, is responsible for the economic loss due to low quality products. It is important to emphasize that variability is something which is attached to the process. It is impossible to remove all variability in the process, but, it can be reduced to its minimum expression.

The loss function is also an important concept in the statistical process control field. The loss function is the function that defines the economic loss due to the variations in the optimal value both outside and within the tolerance limits.

With all these concepts established, the main objective of the quality is to make products with the minimum variation and the lowest value of the loss function. The sources of variability in the process must be found and removed and the loss function must be minimized in order to obtain high quality results.

2.2.1 Causes of variation

There are two different sources of variability in processes: the common causes and the special causes.

The common causes are induced by every component in the process. The presence of the common causes must be assumed in all processes. They can be due to several factors, for example, different plant operators' performance (within limits), features of the raw material (within the tolerance limits), the machine tuning or environmental conditions.

The main properties of the common causes are the following: there are many individual causes, they produce low variations, the variation is time constant, it is very difficult to remove and reduce them, and processes under the influence of the common causes are predictable and can be run optimally.

The special causes can be due to several factors, such as machine wear, human faults or materials outside the specification.

The main characteristics of the special causes are the following: they generate high variations, they are individual, the variation is not time constant, the variation disappears when the root cause is removed, processes do not run optimally under their influence and processes are not stable and predictable (Vilar, 2005).

2.2.2 Univariate statistical process control

Modern statistical process control proposes to monitor the behaviour of the plant *on-line* instead of looking only at the final quality of products (Vilar, 2005).

Statistical process control aims to monitor and improve the subsystems of the processes in order to obtain high quality products. The most used tool to monitor the state of processes is the control chart. This graphic shows a statistical signal.

The control charts aim to distinguish between the common causes and the special causes. When a special cause is detected, it must be removed and the monitored process can come back to a behaviour under statistical control, i.e., the process is only affected by the common causes and is in a predictable state.

The control chart assumes that the variability in a process does not change when it is only affected by common causes. This means that statistical propierties such as the mean and the variance are repeatable under the same operating conditions.

Control charts use limits to know if the process is either under or out of control. These limits can detect on-line if a special cause appears in the process operation.

There are several univariate control charts, the most widely used are the Shewhart control chart (Shewhart, 1938), the cumulative sum (CUSUM) control chart (Page, 1954; Woodward and Goldsmith, 1964) and the exponentially weighted moving average (EWMA) (Hunter, 1986).

The univariate control charts allow a unique variable to be monitored without taking into account the rest of the variables in the process. Figure 2.1 shows an example of a Shewhart control chart for a variable x, where its mean is $\bar{x}$ and its standard deviation s. These charts do not take into account the correlation between the measured process variables.

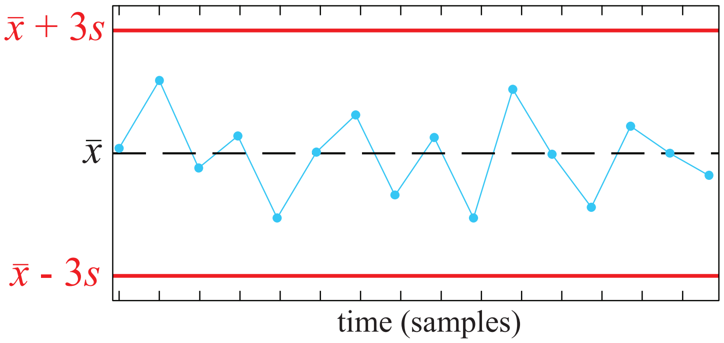


Figure 2.1: Shewhart control chart.

When univariate control charts are applied to multivariate systems, with hundreds of variables, the results are improper because, when there is a fault or an abnormality in the operation, several of these charts set off alarms in a short period of time or simultaneously. This situation is due to the fact that the process variables are correlated, and a special cause can affect more than one variable at the same time. An example of this drawback can be seen in figure 2.2 (Nomikos and MacGregor, 1995), where two variables (x and y) present a normal behaviour if they are monitored using univariate control charts. However, both variables are correlated and the sample shown with a cross breaks the correlation. This breaking of the correlation can be due to a special cause.

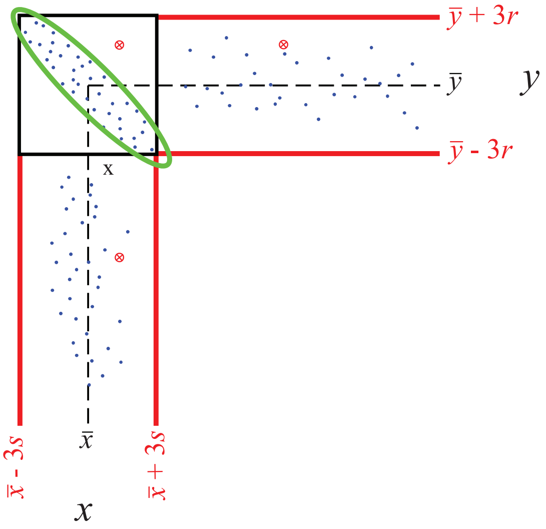


Figure 2.2: Two variables correlated.

For all these reasons, modern industry requires techniques that take into account the correlation between the variables and consider the whole plant in order to build models which estimate the behaviour of the plant. Principal component analysis (PCA) and partial least squares (PLS) are two multivariate statistical process control (MSPC) techniques which deal with these requirements. The main features of the multivariate statistical process control are explained in this chapter, in the section dedicated to principal component analysis.

2.3 Fault detection and isolation

Modern control techniques have solved many problems in the industry. However, the appearance of a special cause in the process can mean that it does not work under control. The special causes in many cases can be due to faults in the process or in the process components. In Blanke et al. (2003), a fault is defined as *an unpermitted deviation of* *at least one characteristic property or parameter of the system from its acceptable/usual/standard condition.* Faults can be due to blockages in pipes, offsets in sensors, decay of catalysts, extreme changes in concentrations or abrupt changes in the ambient temperature, among others. For this reason, a new step in process automation is needed by developing automatic fault detection and isolation methods.

Faults can be classified following different criteria. From the point of view of the process model, the faults can be classified as (Gertler, 1998; Bregón, 2010):

- *Additive faults.* These can be seen as unknown inputs acting in the plant and can be due to system leaks or loads, etc.

- *Multiplicative fault.* These are due to some system parameters. They can be due to clogging or loss of power.

The fault can also be classified, from the point of view of the time dependency (Gertler, 1998; Bregón, 2010), as:

- *Abrupt faults*, where the fault magnitude is constant along time, or *evolutive faults*, where the fault magnitude is time-varying.

- *Incipient faults*, where the faults' effects appear gradually, or *crisp faults*, where the fault effects appear suddenly.

- *Permanent faults*, where the faults' effects are constant along time, or *intermittent faults*, where the faults' effects appear and disappear along time.

Also, from the point of view of the component affected by the fault, the faults can be classified into three classes (Blanke et al., 2003; Bregón, 2010):

- *Plant faults.* These faults are due to changes in the dynamical input/output properties of the system.

- *Sensor faults.* These faults are due to malfunctions in sensors.

- *Actuator faults.* These faults are due to malfunctions in actuators.

The main objective of the fault detection and isolation methods is to guarantee successful operation and to detect possible faults or special causes in an automatic way. Using these techniques, plant operators can have the most information about the process. They can make decisions by looking at the outputs of these techniques in order to restore the normal process behaviour.

The fault detection and isolation tasks can be divided into four phases (Puigjaner et al., 2006; Blanke et al., 2003) that can be performed in an iterative process:

- Detection of the existence of a fault.
- Identification of the fault. In this phase, the variables related with the fault must be identified.
- Fault diagnosis. This task should use a classification method to determine the type and the root of the fault.
- The last phase must recover the process, if this procedure is possible.

Other authors, like Gertler (1998), propose considering only two main tasks in the fault detection and isolation field: fault detection and fault diagnosis.

2.3.1 Design requirements

When a fault detection and isolation system is designed, several design requirements must be taken into account. The main features that a fault detection and isolation system should include are the following (Puigjaner et al., 2006):

- **Fault detection delay:** This is the time between fault occurrence and detection. This parameter must be minimized.

- **Correct detections ratio:** This parameter measures if all the detected faults have actually occurred.
- **False alarms ratio:** This parameter must be minimum. A false alarm is the detection of a fault which has not occurred.
- **Isolation capability:** This is the capability of distinguishing between the different faults.
- **Sensitivity:** This is the capability of detecting low size faults.
- **Robustness:** This parameter is related with the capability of detecting and diagnosing faults under disturbances and errors.

2.3.2 Fault detection and isolation approaches

There are several classifications for fault detection and isolation techniques. Fault detection and isolation is a highly complex task and many solutions have been proposed. These solutions are studied from different fields such as process control, computing science, statistics or electronics. Also, these techniques have been applied to industry, aeronautics, electronic components and image processing, among others.

Several authors have proposed different classifications. Gertler (1998) proposes to divide the fault detection and isolation methods into two groups: the model-free methods and the model-based methods, depending on the necessity of using a mathematical model or not, normally first principles, state space or input-output models.

One of the most recent classifications published is the classification of Venkatasubramanian et al. (2003c,b,a). The authors divide these techniques into three classes:

- *Quantitative model-based methods.* This category groups fault detection and isolation approaches such as parity relations, Kalman filters and parameter estimation. All these methods require quantitative models, normally input/output or state-space models, although first principles models can also be used. The diagnostic procedure is based on finding discrepancies between the real plant and the model.

- *Qualitative models and search strategies.* In this group, two main classes are grouped together. On the one hand, the methods based on qualitative methods such as knowledge-based expert systems. On the other hand, the search strategies such as topographic searches and symptomatic searches. Topographic searches find malfunctions using a template of normal operation and symptomatic searches try to find symptoms to direct the search to the fault location.

- *Process history based methods.* The methods grouped in this category use past data from the process to capture the process trends. This procedure can be performed in a qualitative way, like expert systems, or in a quantitative way, like neural networks, statistical classifiers or the multivariate statistical techniques, like partial least squares (PLS) or principal component analysis (PCA). These techniques compare the past trends of the process under control with the current state of the plant in order to detect and diagnose faults.

  Figure 2.3 shows a diagram with the classification of Venkatasubramanian et al. (2003c,b,a)

  Nevertheless, fault detection and isolation problems are usually very complex and the best solution can be obtained with the combination of several techniques of different categories.

2.4 Statistical process history-based fault detection and isolation methods

When statistical process history-based methods are applied, data is normally collected from the plant in normal operation conditions and arranged into a data matrix $\mathbf{X} \in \Re^{K \times J}$, where K is the number of time observations and J the number of measured variables.

Essentially, the quantitative process history-based methods deal with the fault detection and isolation task as a pattern recognition problem (Venkatasubramanian et al., 2003a). An example is the Fisher

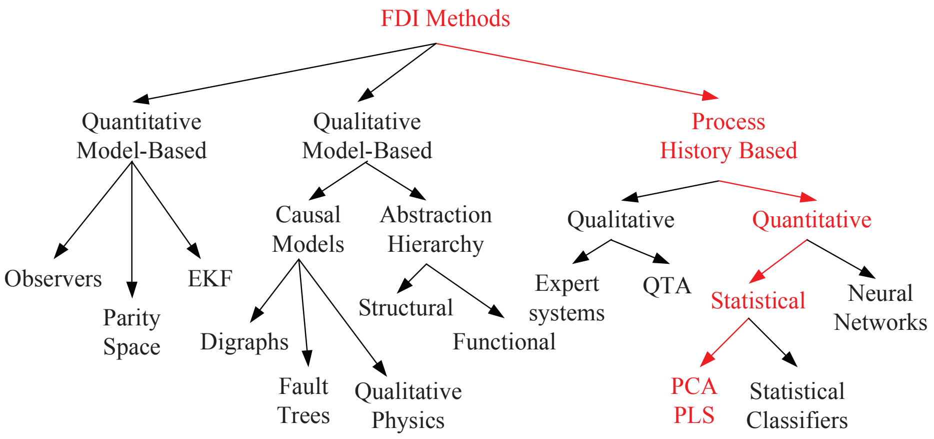


Figure 2.3: Fault detection and isolation methods classification (Venkatasubramanian et al., 2003c,b,a).

discriminant analysis (FDA). Fisher discriminant analysis is a linear pattern classification method used to find the linear combination of features which best separate two or more classes. It is an empirical method based on observed attributes over the collected examples. FDA provides an optimal lower dimensionality representation in terms of a discriminant between classes of data, where, for fault diagnosis, each class corresponds to data collected during a specific known fault. (Duda et al., 2000; Chiang et al., 2001; He et al., 2005; Fuente et al., 2008).

On the other hand, methods, such as the partial least squares (PLS) (Höskuldsson, 1988; Ferrer et al., 2008) and the PCA, extract the principal trends from process history data. When new data from the process do not follow these trends, a faulty situation can be detected. The PLS can also work as a pattern recognition method or as an estimation method.

Recently, other multivariate process history-based methods have been studied and applied, such as independent component analysis (ICA), correspondence analysis (CA) or canonical variate analysis (CVA).

Independent component analysis (ICA) (Hyvärinen et al., 2001; Lee et al., 2003, 2004, 2006) aims to decompose the set of multivariate data into a base of statistically independent components without loss of information. This method takes into account the fact that the variables in many real processes do not follow a Gaussian distribution. The process monitoring based on ICA uses control charts based on the Mahalanobis distance and the residual space in the same way as the PCA monitoring approach. The ICA model can be applied to the augmenting matrix with time-lagged variables in order to capture dynamics in the process. This method is known as dynamic ICA (DICA) (Villegas et al., 2010; Odiowei and Cao, 2010).

Correspondence analysis (CA) is also used as a fault detection and isolation method (Detroja et al., 2007). Correspondence analysis is a dual analysis. It simultaneously analyses dependencies in columns (variables), rows (observations or individuals), and the joint row-column spaces in a dual lower dimensional space. So, dynamic correlations can be represented without having to augment the data matrix with time-lagged variables. The methods based on correspondence analysis are useful when the row and column spaces have certain dependences due to the nature of the system dynamics.

Another technique based on data dimensionality reduction is canonical variate analysis (CVA) (Russell et al., 2000; Odiowei and Cao, 2010). CVA is a dimension reduction technique based on state variables and can be suitable for monitoring processes with auto-correlated and cross-correlated measured data. A state space is estimated when this type of technique is used.

2.5 Principal component analysis (PCA)

As mentioned before, PCA falls into the third class of the Venkatasubramanian et al. (2003c,b,a) classification because it uses process data to build a statistical model. In this classification, PCA is considered to be a statistical quantitative process history-based fault detection and isolation method (red colour path in figure 2.3).

One of the main problems in a fault detection and isolation method based on process history is the great amount of data which must be analysed. For this reason, it can be very useful to capture the process trends with a lower number of variables. This dimensionality reduction must be accurate and the information lost must be minimum.

Principal components analysis (PCA) deals with this problem. Given J measured variables, the PCA represents them using a lower number of variables. These new variables are linear combinations of the original measured process variables.

This technique was described by the American mathematician and economist Harold Hotelling in 1933, but the origin of this technique is the orthogonal adjustment by the least-squares presented by the British mathematician Karl Pearson in 1901.

Principal components analysis has been studied from two perspectives in process control, the cited multivariate statistical process control (MSPC) and the fault detection and isolation field.

Multivariate statistical process control, and principal components analysis particularly, have been studied and researched by several authors from the point of view of statistical process control (SPC) (Jackson and Mudholkar, 1979; Wold et al., 1987; Kourti and MacGregor, 1996; Shlens, 2005; Ferrer, 2007).

The use of principal components analysis in monitoring and fault detection have been applied in several industrial system. The wide use of this technique is due to two main causes (Peña, 2002):

- PCA is able to represent optimally a J-dimensionality space with A dimensions, where $A < J$. This procedure finds the non-observed latent variables which explain the process state.

- PCA transforms the original process variables, normally highly correlated, to new uncorrelated variables. These new uncorrelated variables can make the data interpretation easier.

PCA has been studied in detail as a process monitoring technique. This method has been widely used due to several factors. The reduced dimension representation captures the hidden latent variables, which extract the process state. The model built by the PCA can be used to compare new process observations and to decide if something wrong occurs in the process. The PCA splits the data space into two subspaces. One of them captures the process trends and the other contains the process noise.

2.5.1 Geometric interpretation of the PCA

PCA aims to find a reduced dimension space. When the data are projected onto this new space, the points must preserve the original distribution as much as possible with the minimal distortion.

Given a cloud of points dataset represented on a plane (two dimensions), as shown in figure 2.4, the points can be projected onto a straight line (one dimension) while still maintaining their relative positions. So the line must be as close as possible to the points. The line which fulfils these conditions is a good summary of the original data with a dimensionality reduction. The condition that the line must be as close as possible to every point, can be satisfied by fixing that the distance between the points and their projection must be minimum.

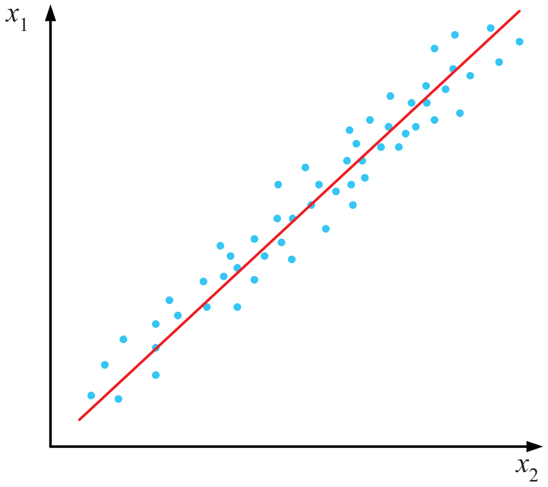


Figure 2.4: Cloud of points approximated by a straight line.

Given a point on the plane $\mathbf{x}_i$, as figure 2.5 shows, it can be localized on the plane by means of the vector $\overrightarrow{\mathbf{x}}_i$. This vector can be decomposed as the vectorial sum of two other vectors:

$$\overrightarrow{\mathbf{x}}_i = \overrightarrow{\hat{\mathbf{x}}}_i + \overrightarrow{\mathbf{e}}_i \tag{2.1}$$

where $\overrightarrow{\hat{\mathbf{x}}}_i$ is the vector that localizes the projection of $\mathbf{x}_i$ onto the straight line and $\overrightarrow{\mathbf{e}}_i$ is the residual vector whose module is the distance between the module and its projection.

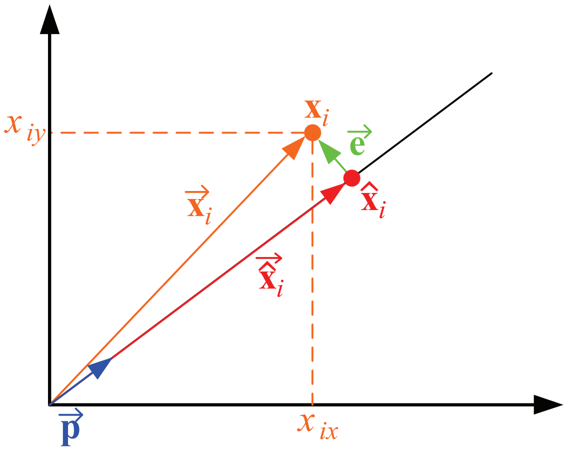

x_{iy}
$\mathbf{x}_i$
$\overrightarrow{\mathbf{e}}$
$\hat{\mathbf{x}}_i$
$\overrightarrow{\mathbf{x}}_i$
$\overrightarrow{\hat{\mathbf{x}}}_i$
$\overrightarrow{\mathbf{p}}$
x_{ix}

Figure 2.5: Geometric interpretation.

The vector $\overrightarrow{\hat{\mathbf{x}}}_i$ can also be expressed as:

$$\overrightarrow{\mathbf{x}}_i = t_i \overrightarrow{\mathbf{p}} \tag{2.2}$$

where $\overrightarrow{\mathbf{p}}$ is a unit vector. This vector is the direction vector of the straight line. t_i is the module $(|\overrightarrow{\mathbf{x}}_i|)$ of the vector $\overrightarrow{\mathbf{x}}_i$. The vector $\overrightarrow{\mathbf{p}}$ is known as the *loading* and t_i as the *score*.

Once the direction of the line is known $(\overrightarrow{\mathbf{p}})$, the points projected onto the line can be defined by the score, which means that every point can be defined by a scalar t instead of a pair $(\hat{x}_x, \hat{x}_y)$.

Figure 2.6 shows the first component extracted. The origin of the new dimension can be fixed to establish the mean of the scores (t_i) to zero value.

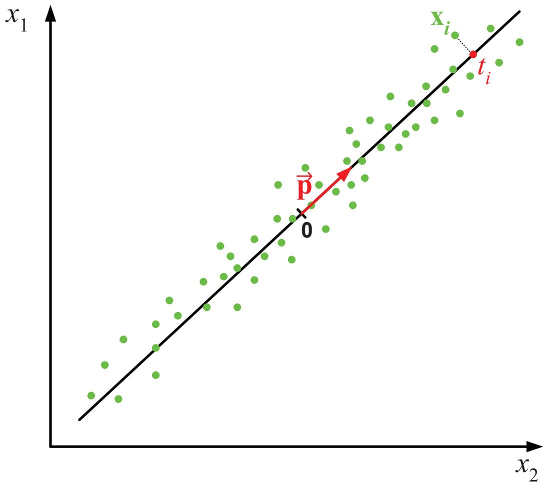


Figure 2.6: First component.

The objective function to ensure that the distance of the points to their projections in the new direction must be minimum can be expressed as:

$$\min \sum_{i=1}^{n} \left|\vec{\mathbf{e}}_i\right|^2 = \min \sum_{i=1}^{n} \left|\vec{\mathbf{x}}_i - \vec{\hat{\mathbf{x}}}_i\right|^2 = \min \sum_{i=1}^{n} \left|\vec{\mathbf{x}}_i - t_i \vec{\mathbf{p}}\right|^2 \tag{2.3}$$

where n is the number of points considered.

Looking at figure 2.5, when a point of the cloud is projected onto the new direction, a right-angled triangle is formed, where the hypotenuse is the distance between the origin of the new direction: $\left|\vec{\mathbf{x}}_i\right|$. The catheti of the triangle are the distance from the origin to the projection (t_i) and the distance from the point to its projection ($\left|\vec{\mathbf{e}}_i\right|$). Applying the Pythagorean theorem, the following expression can be obtained:

$$\left|\vec{\mathbf{x}}_i\right|^2 = t_i^2 + \left|\vec{\mathbf{e}}_i\right|^2 \tag{2.4}$$

If this expression is summed for every point, the following expression can be obtained:

$$\sum_{i=1}^{n} \left|\overrightarrow{\mathbf{x}}_i\right|^2 = \sum_{i=1}^{n} t_i^2 + \sum_{i=1}^{n} \left|\overrightarrow{\mathbf{e}}_i\right|^2 \tag{2.5}$$

where the second addend is the objective function (Eq. 2.3). For the same cloud of points, the term $\sum_{i=1}^{n} \left|\overrightarrow{\mathbf{x}}_i\right|^2$ is constant. For this reason, the minimization of $\sum_{i=1}^{n} \left|\overrightarrow{\mathbf{e}}_i\right|^2$ is equivalent to the maximization of $\sum_{i=1}^{n} t_i^2$. As cited before, the mean of the distribution of the scores (t_i) is zero, which means that the maximization of this term is equivalent to the maximization of the variance of the distribution of the scores. In other words, to find the direction that minimizes the distance between the points and their projection is equivalent to finding the direction where the projection of the point (scores) preserves the maximum variation with respect to the original points.

2.5.2 PCA computation

The measured process variables can be arranged into a matrix $\mathbf{X} \in \Re^{K \times J}$, where K is the number of observations and J is the number of measured variables.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1J} \\ x_{21} & x_{22} & \dots & x_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ x_{K1} & x_{K2} & \dots & x_{KJ} \end{pmatrix} \tag{2.6}$$

The columns of matrix $\mathbf{X}$ are called variables ($\mathbf{x}$) and represent the values of every variable along the time. The rows are called individuals ($\mathbf{z}^T$) and represent the value of all the measured variables at every time stamp.

It is advisable to normalize every variable (column) to zero mean and unit variance by subtracting the variable mean $\bar{x}_j$ and then dividing by the standard deviation s_j. The means and standard deviations can be arranged on vectors $\bar{\mathbf{x}}$ and $\mathbf{s}$ respectively. This pretreatment is necessary because it is suitable that all variables have the same weight in the PCA computation. Also, the outliers must be removed from the original data (MacGregor and Kourti, 1995; Kourti, 2005).

The covariance matrix can be calculated as:

$$\mathbf{S} = \frac{1}{K-1}\mathbf{X}^T\mathbf{X} \tag{2.7}$$

and performing the singular value decomposition (SVD):

$$\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \tag{2.8}$$

where $\mathbf{\Lambda} \in \Re^{J\times J}$ is a diagonal matrix that contains the eigenvalues of $\mathbf{S}$ in its diagonal sorted into decreasing order ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{rank(\mathbf{X})} \geq 0$). The transformation matrix $\mathbf{P}_{1:A} \in \Re^{J\times A}$ is generated by choosing A eigenvectors or columns of the orthogonal matrix $\mathbf{V}$ corresponding to the largest A principal eigenvalues. Matrix $\mathbf{P}_{1:A}$ transforms the space of the measured variables into the reduced dimension space:

$$\mathbf{T} = \mathbf{X}\mathbf{P}_{1:A} \tag{2.9}$$

The matrix $\mathbf{P}_{1:A}$ establishes the directions of the new reduced space and its columns are called *loadings* $\mathbf{p}$. The scores matrix $\mathbf{T}$ can be considered as a set of row vectors $\boldsymbol{\tau}^T$ (scores in the ith observation) or column vectors $\mathbf{t}_a$ (latent variables).

Operating in Eq. 2.9, the scores can be transformed into the original space:

$$\widehat{\mathbf{X}} = \mathbf{T}\mathbf{P}_{1:A}^T \tag{2.10}$$

The residual matrix $\mathbf{E}$ is calculated as the difference between the original variables and the reconstructed variables:

$$\mathbf{E} = \mathbf{X} - \widehat{\mathbf{X}} \tag{2.11}$$

Finally, the original data space can be reconstructed as (Fig. 2.7):

$$\mathbf{X} = \mathbf{T}\mathbf{P}_{1:A}^T + \mathbf{E} \tag{2.12}$$

where the original data space can be shown as the sum of two terms. The first term is explained by the principal components and the second is the noise space that does not explain the process trend.

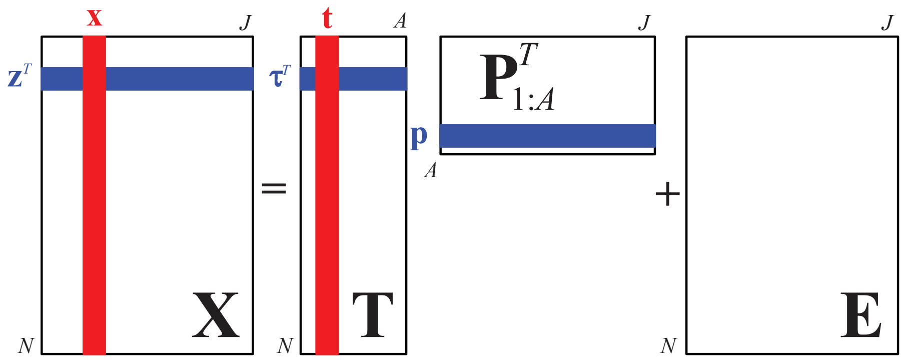


Figure 2.7: Matrices and notation in the PCA formulation.

The PCA model can also be calculated using the non-linear iterative partial least squares (NIPALS) algorithm (Wold et al., 1987). The formulation of both methods is quite similar, but there are variations of the NIPALS algorithm which are able to deal with missing data. The NIPALS algorithm allows only the desired number of components to be calculated. An implementation of this algorithm can be found in appendix A of this document.

The first principal component, which is a linear combination of the original variables, defines the direction of the biggest variability in the process data and it has the biggest eigenvalue in the diagonal of the matrix Λ. The second component has the second largest source of variability in the process data and so on. Normally, in industrial processes, the main sources of variability are captured with only a small number of principal components. When the process variables are more correlated, the number of necessary principal components is lower. The components related to the lower components represents the noise process and the instrument noise and must not be considered.

2.5.3 Selection of the number of principal components

There are several techniques to choose the adequate number of principal components (A). This selection establishes the number of loadings in the matrix $\mathbf{P}_{1:A}^T$ and the components related with the process noise.

There are several methods published for performing this selection (Jackson, 2003; Chiang et al., 2001; Weighell et al., 2001; Peña, 2002). One of these methods consists of drawing a bar plot with the ordered eigenvalues λ_i $i = 1, 2, \ldots, rank(\mathbf{X})$, from highest to lowest. In this bar plot, a knee must be found, which means, to find the lowest and similar bars. The components related with these eigenvalues are not selected. Other methods are based on the cumulative percent variance (CPV). These methods add components until a level of CPV is raised (Normally, $80\% - 90\%$). In other cases, the components related to the eigenvalues of the covariance matrix lower than 1 are removed.

All the mentioned methods have an important heuristic behaviour. For this reason, the cross validation is one of the more extended procedures to choose the number of principal components (Eastment and Krzanowski, 1982; Bro et al., 2008). This procedure selects the number of components that maximizes the goodness of fit and the goodness of prediction (Zarzo, 2004). But the best method to be applied is actually a matter of scientific debate (Camacho, 2007; Camacho et al., 2010).

The goodness (R_x^2) of fit is a measurement of how the PCA model fits the process data and it is calculated using the variance of the components. This measurement raises the maximum value when all the components are considered. For this reason, this goodness is complemented with the goodness of prediction.

The goodness of prediction Q^2 is calculated based on the prediction error sum of squares ($PRESS$). The number of components selected must maximize this measurement. A detailed explanation of these measurements can be found in Zarzo (2004).

2.5.4 Statistics for process monitoring using PCA

The monitoring statistics are used to monitor the state of the process using the established PCA model. These statistics are used to draw control charts which are very useful to supervise the state of the process. The statistics proposed in the principal components analysis allow the whole process to be monitored with two control charts, instead of monitoring every variable as the univariate SPC proposed.

The most used monitoring statistics are (Chiang et al., 2001):

- **Hotelling's statistic (T^2):** Given a new process observation $\mathbf{z}_i^T \in \Re^{1\times J}$ after the normalization using the vectors $\bar{\mathbf{x}}$ and $\mathbf{s}$ of means and standard deviations, this statistic, based on the Mahalanobis distance, can be calculated as:

$$T^2 = \mathbf{z}^T \mathbf{P} \mathbf{\Lambda}_A^{-1} \mathbf{P}^T \mathbf{z} \tag{2.13}$$

This calculation is equivalent to the sum the squared principal scores divided by the related eigenvalue:

$$T^2 = \sum_{i=1}^{A} \frac{\tau_i^2}{\lambda_i} \tag{2.14}$$

where $\mathbf{\Lambda}_A$ is an $A \times A$ diagonal matrix of the higher A eigenvalues of the covariance matrix $\mathbf{S}$ (Eq. 2.7) in decreasing order along the diagonal.

This statistic gives a measurement of the variation in the process captured by the PCA model. It is calculated using the subspace of the selected components which represent the most important sources of variability.

The process is considered normal for a given significance level α if:

$$T^2 \leq T_\alpha^2 = \frac{(K^2-1)A}{K(K-A)} F_\alpha(A, K-A) \tag{2.15}$$

where $F_\alpha(A, K-A)$ is the critical value ($100(1-\alpha)\%$ percentile) of the Fisher-Snedecor of F distribution with A and $N-A$ degrees of freedom and α is the level of significance. α takes values between 5% and 1%.

This threshold can be calculated *off-line* because all the necessary parameters are known. T^2 will only detect an event if the variation in the latent variables is greater than the variation explained by common causes. New events can be detected by calculating the squared prediction error statistics described in the next item.

- **Squared prediction error (SPE) or Q statistic:** The scalar value Q is a measurement of goodness of fit of the sample to the model and is directly associated with the noise.

  This statistic can be calculated for a new process observation $\mathbf{z}^T$:

$$Q = \mathbf{r}^T\mathbf{r} \tag{2.16}$$

  with:

$$\mathbf{r} = (\mathbf{I} - \mathbf{P}_{1:A}\mathbf{P}_{1:A}^T)\mathbf{z}$$

  where $\mathbf{r}$ is the vector of residuals between the observation and its projection onto the reduced space.

  In the same way as with the previous statistic, an upper limit of this statistic can be established as follows:

$$Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{2.17}$$

  with:

$$\theta_i = \sum_{j=a+1}^{m} \lambda_j^i \qquad h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$$

  where c_α is the $100(1-\alpha)$ standardized normal percentile and λ_j are the eigenvalues of the PCA residual covariance matrix $\mathbf{E}^T\mathbf{E}/(K-1)$.

The Q statistic should be checked first, and if no point raises the control limit, the process can be considered to be in-control. If the value of one of the statistics is greater than the upper limit, an alarm can be detected in the process. It is normal in fault detection and isolation procedures that several consecutive alarms must occur for a fault to be detected.

Figure 2.8 shows a geometric interpretation of both explained statistics. In this example, for a geometric interpretation, there are three measured variables (x_1, x_2 and x_3) and, after a PCA computation, two components were chosen (PC_1 and PC_2). The graph of the figure shows that the measurements are placed in a three-way space, but the distribution of these points is better explained by a plane (two dimensions).

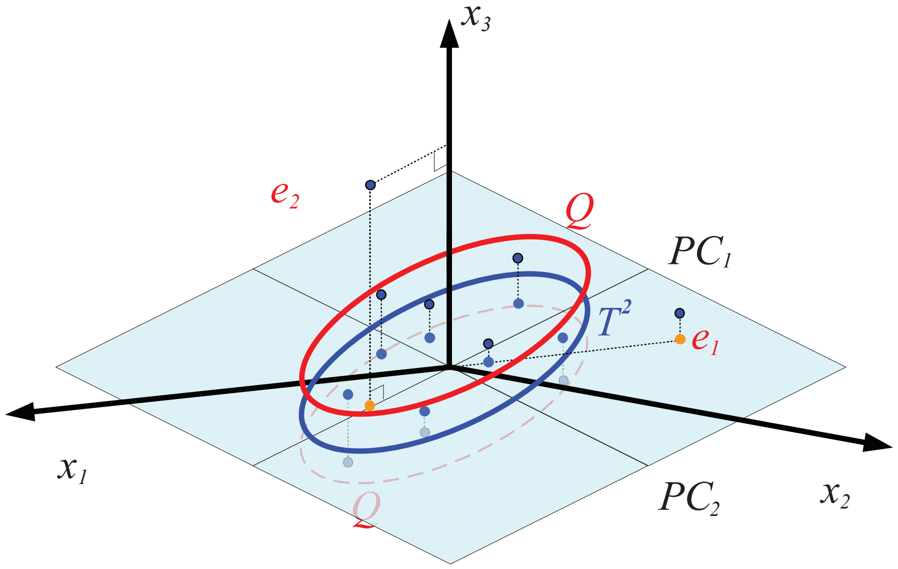


Figure 2.8: Geometric interpretation of the monitoring statistics.

Considering two components, Hotelling's statistic and the upper limit (equations 2.14 and 2.15) can be related as:

$$\frac{\tau_1^2}{\lambda_1} + \frac{\tau_2^2}{\lambda_2} \leq \frac{(K^2-1)A}{K(K-A)} F_\alpha(A, K-A) \tag{2.18}$$

where λ_1 and λ_2 and the right term are constant. So, the expression of equation 2.18 is the geometric place of an ellipse. The observations considered as faulty by the T^2 statistic are those observations whose projection onto the reduced space are outside the blue ellipse in figure 2.8 (observation e_1).

The statistic Q, defined by the equation 2.16, is the squared prediction error $Q = |\mathbf{r}|^2 = \left(\sqrt{\mathbf{r}^T\mathbf{r}}\right)^2 = \mathbf{r}^T\mathbf{r}$, which is the squared module of the residual vector and can be interpreted geometrically as the squared distance between the observations and its projection onto the reduced space. Relating the equations of the statistic and the threshold (equations 2.16 and 2.17):

$$\mathbf{r}^T\mathbf{r} \leq \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{2.19}$$

the statistic can be interpreted as a high boundary value for the distance between the observations and their projection onto the reduced space.

2.5.5 Fault diagnosis

Though PCA has been successfully used for monitoring and fault detection, it provides little information for fault isolation. Contribution analysis (Kourti and MacGregor, 1996; Ferrer, 2007) has been proposed as a first approximation to fault isolation. Contribution analysis automatically computes the influence of each of the system's variable to the value of the Q and T^2 statistics. The most used contributions plots are introduced in this section (Kourti and MacGregor, 1996):

Bar plots of normalized errors of the variables

When an observation $\mathbf{z}^T \in \Re^{1\times J}$ falls outside the limits in the Q statistic, the normalized error for each variable is calculated as:

$$cont_{z_j} = (z_j - \mu_j)/s_j \tag{2.20}$$

and these contributions can be plotted in a bar plot.

The variables with the highest value in this graph can be detected as the variables with the most influence in triggering the statistic.

Bar plots of normalized scores

Once the T^2 statistic has fallen outside the limits, the normalized scores can be plotted in a bar chart.

The normalized scores can be calculated for the scores of an observation outside the limits $\boldsymbol{\tau}^T \in \Re^{1\times A}$ as:

$$cont_{\tau_a} = \tau_a/\lambda_a \tag{2.21}$$

They are plotted in a bar chart on this type of plot. In this case, the scores with the highest value can be identified as the scores which contribute most to triggering the T^2 statistic.

This plot gives information about the scores, but this information can be poor from the point of view of the plant operators. For this reason, the plot explained in the next subsection may be more useful.

Variable contributions to individual scores

In this case, a contribution plot indicates how each variable involved in the calculation of a score contributes to it. Normally, contribution plots are constructed for normalized scores with high values. The variables with a high contribution to the score should be investigated.

The contribution of each variable z_j (normalized) to the score of the principal a-th component is:

$$cont_{z_j,\tau_a} = p_{a,j}x_j \tag{2.22}$$

This contribution analysis allows us to inspect which variables are related to the scores with high values.

Overall Average variable contributions

It is very common that more than one score presents a high value and it can be useful to compute the overall average contribution per variable, instead of drawing a plot for every score with a high value. Further information about this contribution plot can be found in Kourti and MacGregor (1996).

2.6 Fault detection methods based on PCA

This section presents different variations to the classical PCA-based fault detection method. The different methods present different improvements and considerations in order to reduce the number of false alarms, to detect consecutive faults or to detect faults in transient states.

The dynamic PCA method (Ku et al., 1995) is performed in the same way as the DICA method explained previously. This method does not only extract the relation between the variables at the same instant, it also takes into account the relations between the variables at the same time instant and at previous instants in order to extract the dynamics in the process. The data matrix in this case is augmented with time-lagged variables.

When a fault is detected using the classical PCA approach, the monitoring statistics rises above the established control limits and these high values are normally persistent. A new fault cannot be detected in this situation due to the fact that the statistics are over the thresholds. A method proposed for detecting consecutive faults is the adaptive PCA (APCA) (Zumoffen and Basualdo, 2007). In this configuration, when a fault is detected, the PCA model is updated to the new situation and the statistics come back under the limits and new faults can be detected.

Other method based on PCA and related with the APCA is the exponentially weighted PCA (EWPCA) (Lane et al., 2003; Tien et al., 2004). In this case, the PCA model is established in an adaptive and exponentially weighted way *on-line* using a moving window. In the EWPCA configuration, the current samples are considered more valuable for the PCA model than the samples measured some time ago. This configuration allows consecutive faults to be detected and reduces the alarms ratio.

The recursive PCA (RPCA) (Li et al., 2000) is also an adaptive formulation of the PCA. In this type of method, the correlation matrix is recursively updated. In general, these methods deal with slow process changes that can occur in the process operation.

Other extended PCA-based methods are the multiscale PCA approaches (MSPCA) (Misra et al., 2002). The MSPCA inspects low frequency components of the measured variables because they provide more information than the high frequency components. This frequency decomposition can be performed using wavelets.

Changes in the operating modes can be detected as faults using classical PCA approaches. The PCA using external analysis (Kano et al., 2004) proposes removing the external influence of the reference signals (responsible for changes in the operating modes) from the process variables. This configuration aims to build a model without the influence of the operating modes.

The PCA formulation uses parameters like the mean and the variance. The variance is very sensitive to the outliers. If these outliers are not visible, the principal components direction can be affected by them. For this reason, a formulation using robust estimators, like the median centre, has been proposed. This type of method is known as robust PCA (Stanimirova et al., 2004). Another alternative to avoid the influence of outliers is the fuzzification of the matrix data using the fuzzy logic methodology. The methods based on this transformation are called fuzzy PCA (FPCA) methods (Luukka, 2011; Sârbu and Pop, 2005; Heo et al., 2009; Cundari et al., 2002).

2.6.1 Non-linear PCA methods

As mentioned in this chapter, the PCA method is based on a linear transformation. The latent variables extracted, or principal components, are linear combinations of the original measured process variables. But, if these relationships are not linear, then maybe a non-linear technique for finding the reduced space of the latent variables would be a better solution. In these cases, the methods are known as non-linear PCA (NLPCA).

One family of methods that can be included in this category are the kernel PCA approaches. Kernel PCA uses the techniques of kernel methods. Using a kernel, the originally linear operations of PCA are done in a reproduced kernel Hilbert space with a non-linear mapping (Schölkopf et al., 1998; Choi et al., 2005).

Several non-linear PCA methods are based on artificial neural networks. The non-linear PCA model can be obtained by combining a principal curve method and a neural network (Dong and McAvoy, 1996). Also, the non-linear PCA model can be performed using an autoassociative neural network with a bottle-neck layer (Kramer, 1991). Another option is to use input training networks (IT-Nets) instead of autoassociative networks (Tan and Mavrovouniotis, 1995; Fourie and de Vaal, 2000).

Another non-linear approach of PCA can be to use a neural model to identify the plant and compute the residuals between the model and the real plant, using a linear PCA method in order to find dissimilarities between them (Fuente et al., 2012).

2.6.2 Combination of methods

Many alternatives can be considered for designing a fault detection and isolation scheme using statistical process history-based techniques. However, a fault detection and isolation scheme does not necessarily have to be based on a particular method. It can be based on two or more methods in order to satisfy the requirements of the designed approach. Even so, non-statistical methods can be used and combined.

An example of a combination of methods can be found in García-Álvarez et al. (2009c). In this work, a classical PCA approach is used to detect faults in a wastewater treatment plant and the Fisher discriminant analysis (FDA) is applied in the fault isolation task.

Another example of the combinations of these methods can be found in Odiowei and Cao (2010). In this case, the ICA method uses CVA instead of PCA for the dimension reduction task. The method is called state space based ICA (SSICA). The authors applied the method to the Tennessee Eastman process plant.

2.7 Partial least squares (PLS)

Another of the most widely used multivariate statistical techniques in industry, together with the PCA, is the partial least squares (PLS) regression (Höskuldsson, 1988; Helland, 2001; Wold et al., 2001; Geladi and Kowalski, 1986). This tool is widely used, due mainly to its many properties. PLS is able to establish regression models from data with high collinearity, allows data with more variables than individuals to be analysed, provides models with a high rate of stability in the prediction, minimizing the rate of adjustment, can treat information sets with missing data, and is able to detect *outliers* in the data source (Ferrer et al., 2008).

The partial least squares method is closely related with the PCA. In fact, there is a version of the NIPALS algorithm for PLS (Geladi and Kowalski, 1986).

PLS establishes a projection structure that models the relationship between a response matrix $\mathbf{Y} \in \Re^{K \times L}$ and the prediction matrix $\mathbf{X} \in \Re^{K \times J}$.

$$\mathbf{X} = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^T + \mathbf{E} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \tag{2.23}$$

$$\mathbf{Y} = \sum_{a=1}^{A} \mathbf{u}_a \mathbf{c}_a^T + \mathbf{F} = \mathbf{U}\mathbf{C}^T + \mathbf{F} \tag{2.24}$$

where $\mathbf{T}$ and $\mathbf{U}$ are known as *scores* matrices, $\mathbf{P}$ and $\mathbf{C}$ are the *loadings* and $\mathbf{E}$ and $\mathbf{F}$ are the residual matrices of $\mathbf{X}$ and $\mathbf{Y}$ respectively, for a model with A latent variables determined using cross-validation. The *scores* of $\mathbf{X}$ are linear combinations of the $\mathbf{X}$ matrix, for the first latent variable, or a residuals matrix $\mathbf{X}_a$ for the a-th latent variable:

$$\mathbf{t}_a = \mathbf{X}_{a-1}\mathbf{w}_a,\ \mathbf{X}_a = \mathbf{X}_{a-1} - \mathbf{t}_a \mathbf{p}_a^T \tag{2.25}$$

where $\mathbf{w}_a$ is the weight vector of the a-th latent variable.

The relationship that maximizes the covariance between $\mathbf{T}$ and $\mathbf{U}$ is established by an inner relationship as follows:

$$\mathbf{U} = \mathbf{TB} + \mathbf{H} \tag{2.26}$$

where $\mathbf{B}$ is a diagonal matrix and $\mathbf{H}$ is a residual matrix.

Once the PLS model is established from the historical data of the process, it is possible to perform the prediction of new responses or estimates as follows:

$$\hat{\mathbf{Y}} = \mathbf{TC}^T = \mathbf{XW}\left(\mathbf{P}^T\mathbf{W}\right)^{-1}\mathbf{C}^T = \mathbf{XW}^*\mathbf{C}^T \tag{2.27}$$

where the loadings matrix $\mathbf{W}^* = \mathbf{W}\left(\mathbf{P}^T\mathbf{W}\right)^{-1}$ is used to perform the $w^*c[1]/w^*c[2]$ loadings graph. This graph is very useful to show which variables in $\mathbf{X}$ are most related to the model response $\mathbf{Y}$. It provides graphical information about the relationship between the measured variables and the estimated variables (Prats-Montalbán et al., 2006).

The matrix $\mathbf{Y}$ is arranged with variables which are not measured *on-line*, normally related with the process quality. The monitoring statistics presented in subsection 2.5.4 can be applied to both cases (matrices $\mathbf{X}$ and $\mathbf{Y}$) and the state of the measured and quality variables can be monitored. Also, contribution analysis can be used to identify the variables related with a detected fault.

PLS regression can be used as the estimation engine in the design of soft sensors, as in the case study presented in chapter 7. Also, PLS can be used in the isolation phase of a fault detection and isolation scheme. In the same way as with the Fisher discriminant method, if data from faulty situations are available, a PLS regression can be established. The matrix $\mathbf{X}$ is arranged with the variables measured in the normal operation conditions (NOC) and faulty situations. The matrix $\mathbf{Y}$ must have the same number of columns as the faulty situations plus the nominal case. The columns of every situation are filled with ones if the data in the same rows in the matrix $\mathbf{X}$ corresponds to this situation and with zeros otherwise. In the *on-line* monitoring phase, looking at the estimated $\mathbf{y}^T$ vector, the current process situation can be known.

2.8 Discussion

This chapter explains the main characteristics of the statistical process control. The main causes of variation (the common causes and the special causes) in processes are presented. Also, the univariate statistical process control techniques are described along with their main limitations. The multivariate statistical process control techniques, like principal component analysis, are presented as a solution for some of these limitations.

The special causes are normally due to faults in processes. So the fault detection and isolation field is presented and described in detail. A classification of the main fault detection and isolation methods is also presented. The multivariate statistical process control techniques are included in the statistical process history-based methods considered in this classification.

Principal component analysis is described in detail from the point of view of fault detection. Therefore, the control charts for fault detection are presented. Many techniques based on principal component analysis are introduced and discussed. Also, the main features of partial least squares are introduced.

Chapter 3

Fault detection in continuous processes using PCA

3.1 Introduction

This chapter deals with the fault detection task in continuous processes. Concretely, different methods based on PCA are applied to the same case study in order to compare the results under the same conditions.

The considered PCA-based approaches are the classical PCA, the dynamic PCA, the adaptive PCA, the multiscale PCA, the exponentially weighted PCA, the PCA using external analysis and the nonlinear PCA. For this reason, a monitoring scheme has been designed using Matlab© 7.0 for every cited PCA-based method.

The main objective of this chapter is to compare the cited PCA-based methods in the same simulated laboratory plant. The main advantages and improvements of the different methods can be observed and discussed.

Also, in this chapter, the monitoring of systems with several operating modes is presented. The transient states between the different operating modes can be detected as faults and PCA-based methods can lose part of their detectability features.

3.2 Methods based on the PCA

In this section, the PCA-based methods used in this chapter are presented and explained.

3.2.1 Dynamic PCA (DPCA)

Normally, processes cannot be modelled using static models because they are dynamic systems. If a process presents a steady state in a fixed operating point, the classical PCA-based fault detection method may be enough. But if the process changes to different operating modes, a dynamical PCA model could be a better solution. The dynamical PCA (DPCA) (Ku et al., 1995) builds the PCA model, taking into account for every individual $\mathbf{z}^T$, the current and past values of the process variables.

The DPCA model can be stablished in the same way as the classical PCA, but the data matrix $\mathbf{X}$ must include past values or delays of the variables in every row.

$$\mathbf{X} = [\mathbf{X}_t|\mathbf{X}_{t-1}|\dots|\mathbf{X}_{t-h}] \tag{3.1}$$

where the operator | is the matrix concatenation operator. $\mathbf{X}_t$ is the data matrix $\mathbf{X}$ at the time instant t and $\mathbf{X}_{t-h}$ at the time instant $t-h$, that is, with a delay of h time samples. Basically, the data matrix with delay is the original data matrix with a row displacement of h time samples, as shown in figure 3.1.

In this configuration, the PCA model is able to capture the relationships between the variables at the current sample and the relationships between the variables at the current sample and past samples.

When the data matrix is established, the *off-line* and *on-line* computations are similar to the classical PCA computation, taking into account the fact that the number of columns in the data matrix has changed. This change can modify the upper limits of the statistical control charts.

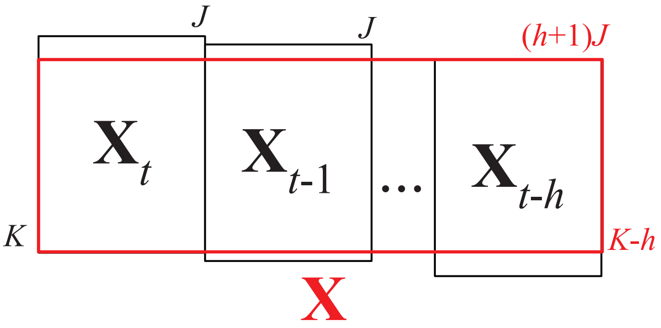


Figure 3.1: Data matrix structure $\mathbf{X}$ for DPCA.

3.2.2 Adaptive PCA (APCA)

The adaptive PCA (APCA) method is also based on the classical PCA method. The formulation of the PCA model can be implemented *off-line* in the same way as the the classical PCA approach. In this method, when a fault is detected, the new observations measured from the plant are not normalized using the means and standard deviations ($\bar{\mathbf{x}}$ and $\mathbf{s}$) calculated *off-line* using historical data. The new samples are normalized using new values of mean and standard deviations, calculated using a moving data window of the new faulty state of the process. The statistic's thresholds must be established using the dimensions of the new matrix. This is why the method is called adaptive. The main advantage of this method is that it is able to adapt to the faulty situation and the monitoring statistics will come back under the upper limits. So, new faults will be detected (Zumoffen and Basualdo, 2007).

The detection algorithm of this method can be implemented with the following steps:

1. The PCA procedure can be implemented in the same way as the classical PCA. This procedure can be carried out *off-line*. The monitoring statistics can also be calculated *off-line* and the size of the window of data N_{aux} can be established.

2. Obtain and normalize the next vector of measurements $\mathbf{z}^T$.

3. Calculate the monitoring statistics T^2 and Q using the current PCA model.

4. Check if Q and T^2 raise the upper limits. If the current measurement is normal, go to step number 2.

5. If the current measurement is not normal, an alarm must be generated. When a consecutive number of alarms have been generated, a fault can be notified.

6. If a fault has been notified, store this measurement and the previous faulty measurements in the matrix $\mathbf{X}_{aux}$. Add new faulty measurements to the matrix $\mathbf{X}_{aux}$ until the size of samples goes above the parameter N_{aux}.

7. When the size of $\mathbf{X}_{aux}$ goes above N_{aux} samples, update the PCA model, the limits of the statistics and the values of the mean and standard deviation vectors using the matrix $\mathbf{X}_{aux}$.

8. Go to step number 2.

Two parameters must be considered in this method and in the following methods: the number of consecutive alarms necessary to detect a fault (this parameter must be considered in all PCA-based methods) and the size of the auxiliary matrix N_{aux}. These parameters can be established taking into account the alarms ratio or the dynamic of the process.

However, when the method is overloading the matrix with faulty measurements, the statistics are over the upper limits and it is not possible to detect a new fault.

3.2.3 Multiscale PCA (MSPCA)

Multiscale principal component analysis (MSPCA) is based on the frequency decomposition of the measured variables' signals (Misra et al., 2002). In this case, this decomposition is performed using wavelets.

Normally, in processes, the low frequency components of the signals provide more information than the high frequency components, which are normally related with the noise in the process and in the instruments.

In the case of wavelets, the different frequency components are separated using filters. Figure 3.2 shows an example of the use of filters in wavelets. In this figure, S is the original signal, A is the signal in the low-pass filter output and D is the signal in the high-pass filter output. The filters have to be designed in a complementary way, i.e., the sum of A and D must be the original signal S.

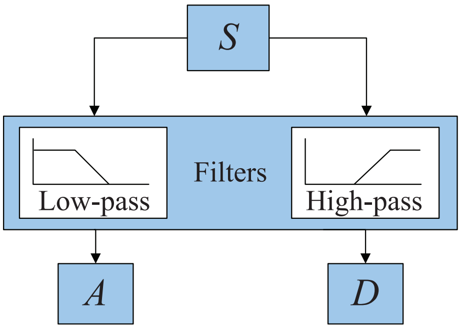


Figure 3.2: Signal decomposition in wavelets.

When the wavelet decomposition is applied to more complex signals, the filtering process can be iterated, that is, to apply the decomposition procedure to the output signals of the first stage and to repeat this procedure until the required precision. This multilevel decomposition is known as the wavelet tree. Figure 3.3 shows an example of a wavelet tree. In this example, D_1 is the highest frequency component and A_2 is the lowest.

The implementation of this method requires the use of a moving window in order to carry out the *on-line* monitoring. This window is necessary for computing the signal decomposition since with only one sample, this task cannot be performed. This method can be summarized in the following algorithm:

- *Off-line*: Data from normal process operation must be recorded. Then, the wavelet frequency decomposition has to be performed for every measured variable. The decomposed variables must be

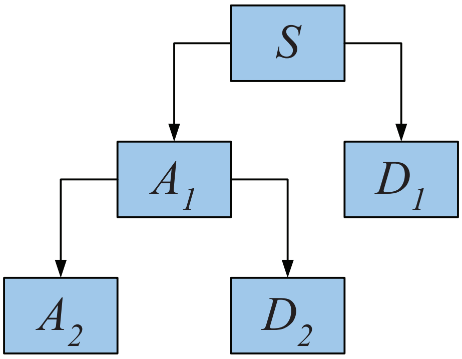


Figure 3.3: Wavelet tree.

arranged in a data matrix for every frequency considered. The PCA model has to be computed for every data matrix. Figure 3.4 shows an example of this procedure. Also, the upper limits of the statistics can be fixed *off-line*. Normally, the Q statistic is the only one used in this method (Misra et al., 2002).

- *On-line*:

  1. Add the following measurements vector to the moving window and calculate the wavelet frequency decomposition using the current moving window and obtain the measurement vector for the different $D_1, \dots, D_l$ and A_l frequencies.
  2. Normalize the current measurement and the frequency measurements $D_1, \dots, D_l$ and A_l with the corresponding means and standard deviations and calculate the monitoring statistic Q.
  3. If the statistic of the component A_l does not go above the upper control limit, consider this measurement as normal and add it to the moving window. The other components can also be inspected.
  4. If a consecutive number of alarms are detected, a fault detection must be notified.
  5. Go to step number 2.

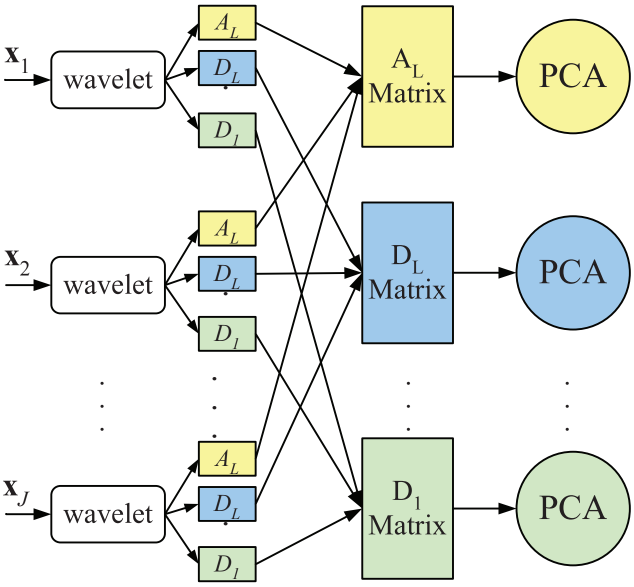


Figure 3.4: Matrices for MSPCA implementation.

3.2.4 Exponentially weighted PCA (EWPCA)

Another method based on PCA is the exponentially weighted PCA (EWPCA) (Lane et al., 2003; Tien et al., 2004). This PCA model is computed *on-line* using a moving window. In this case, the PCA model is established in an adaptive and exponentially weighted way. It is based on a recursive updating of the covariance matrix $\mathbf{S}$. In the EWPCA configuration, the current samples are considered more valuable for the PCA model than the samples measured some time ago. This consideration is established using a weighting factor.

The EWPCA can be implemented in four steps:

- STEP 1: Initialization:

  1. Acquire N_0 samples of the process variables under normal operation conditions and arrange them into the $\mathbf{X}_0$ matrix. The matrix must be normalized to zero mean and unit standard deviation and store the normalization parameters $\bar{\mathbf{x}}_0$ and $\mathbf{s}_0$

2. Calculate the initial covariance matrix:

$$\mathbf{S}_0 = \frac{1}{N_0 - 1}\mathbf{X}_0^T\mathbf{X}_0 \tag{3.2}$$

3. Calculate the initial weighting factor:

$$\beta_0 = (1 - \frac{1}{N_0}) \tag{3.3}$$

- STEP 2: Apply the EWPCA to a new observation:

1. Collect the current process variables vector $\mathbf{z}_t^T$ and normalize it using the normalization parameters of the previous iteration $\bar{\mathbf{x}}_{t-1}$ and $\mathbf{s}_{t-1}$
2. Recalculate the covariance matrix using the weighting factor:

$$\mathbf{S}_t = \beta_{t-1}\mathbf{S}_{t-1} + (1 - \beta_{t-1})(\mathbf{z}_t^T\mathbf{z}_t) \tag{3.4}$$

3. Perform the SVD decomposition to the covariance matrix $\mathbf{S}_t$ and establish the number of principal components.
4. Compute the monitoring statistics T_t^2 and Q_t for the current observation.
5. Update the monitoring thresholds for both statistics T_α^2 and Q_α.
6. If the monitoring statistics are under the upper limits, go to step number 3, otherwise go to step number 4.

- STEP 3: Updating the EWPCA parameters.

1. Update the data matrix $\mathbf{X}_t$:

$$\mathbf{X}_t = \beta_{t-1}\begin{bmatrix} \mathbf{X}_{t-1} \\ \mathbf{z}_t \end{bmatrix} \tag{3.5}$$

2. Update the mean and the standard deviation:

$$\bar{\mathbf{x}}_t = \beta_{t-1}\bar{\mathbf{x}}_{t-1} \tag{3.6}$$

$$\mathbf{s}_t = \lambda_{t-1}\mathbf{s}_{t-1} \tag{3.7}$$

3. Update the weighting factor :

$$\beta_t = 1 - \frac{\frac{(1-T_t^2)}{J}\frac{Q_t}{J}}{\sqrt{N_t - 1}} \tag{3.8}$$

and:

$$N_t = \beta_{t-1}N_{t-1} + 1; \tag{3.9}$$

4. Go to step number 2 with $t = t + 1$.

- STEP 4: When an alarm is detected.

1. Store the observations vector responsible for the alarm in the matrix $\mathbf{X}_{aux}$
2. If the alarm is not persistent (there are no consecutive alarms), go to step number 2.
3. If the alarm is persistent, when the number of time samples in $\mathbf{X}_{aux}$ is higher than N_{aux} go to step number 1 using $\mathbf{X}_{aux}$ as data matrix, otherwise go to step number 3.

3.2.5 PCA using external analysis (PCAEA)

A different implementation for a fault detection and isolation method using PCA is described in Kano et al. (2004). This article proposes splitting the process variables into two groups. The first group consists of the variables which are related with the operating mode. This group is called the group of the external variables. The second group consists of the variables which are affected by the external variables and/or unmeasured disturbances. This group is known as the group of the principal variables. In this document, this method is called PCAEA (PCA using external analysis).

The principal variables are classified into two types. On the one hand, the variables which are affected by the external variables, and on the other hand, the variables which are not affected by the external variables. This method proposes to remove the external variables' influence from the principal variables.

Taking into account these considerations, the data matrix **X** can be partitioned as follows:

$$\mathbf{X} = [\mathbf{H}\mathbf{G}] \tag{3.10}$$

where **G** represents the group of the external variables and **H** represents the group of the principal variables. As mentioned above, **H** has a part affected by **G** and another part which is not affected. So, a multiple linear regression analysis can be established using the external variables as inputs and the principal variables as outputs, and determining a regression coefficient matrix **C** as:

$$\mathbf{C} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{H} \tag{3.11}$$

where the errors matrix can be computed as:

$$\mathbf{F} = \mathbf{H} - \mathbf{G}\mathbf{C} \tag{3.12}$$

The principal variables matrix can be expressed as the sum of two terms, operating in 3.12:

$$\mathbf{H} = \mathbf{G}\mathbf{C} + \mathbf{F} \tag{3.13}$$

where **GC** is the term influenced by the external variables and **F** is the term which is not influenced.

This method can be a solution for removing the influence of the reference signals. The reference signals can be considered external variables.

The matrix **F** (which is not influenced by the external variables **G**) is used to perform the principal components analysis. So the classical PCA is applied to data without the influence of the external variables.

During the *on-line* monitoring task, the measured variables vector must be processed using the **C** matrix in order to remove the influence of the external variables. Then, the monitoring statistics can be computed.

This method can be set out taking into account the dynamics in the process. When a process is dynamical, the influence of the changes in the external variables cannot be ignored. For these cases, a dynamical implementation of this method can be used.

It is usual that the influence of the external variables over the principal variables is defined by a non-linear relationship. For this reason, a linear regression is not the most suitable method to remove the influence of the external variables. The non-linear relationship can be modelled using an artificial neural network (ANN). This variation will be called in this work PCANLEA (PCA using non-linear external analysis) (Kano et al., 2004).

The neural network can be established considering the current and the past values of the external variables $\mathbf{g}_t, \mathbf{g}_{t-1}, \mathbf{g}_{t-2}, \ldots, \mathbf{g}_{t-r}$ as the inputs and the current value of the principal values $\mathbf{h}_t$ as outputs (Figure 3.5). This neural network can be seen as an approximator of $\mathbf{H}$ with the influence of the external variables: $\hat{\mathbf{H}}$ in a similar way to the term $\mathbf{GC}$ in the linear case. The matrix $\mathbf{F}$ (principal variables without the influence of the external variables) can be calculated in this case as:

$$\mathbf{F} = \mathbf{H} - \hat{\mathbf{H}} \tag{3.14}$$

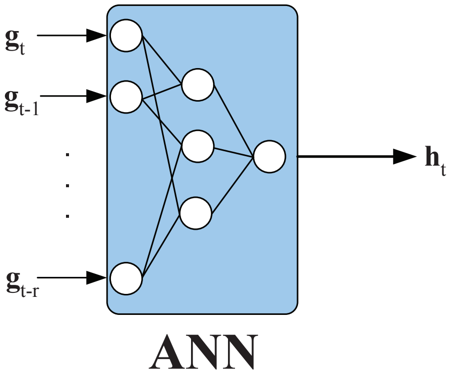


Figure 3.5: Structure of the ANN for PCANLEA.

In the same way as with the linear case, the matrix $\mathbf{F}$ can be used to perform the classical PCA. The monitoring task requires the use of the neural network and the PCA model calculated over **F**.

3.2.6 Non-linear PCA (NLPCA)

As mentioned in subsection 2.6.1, autoassociative neural networks can explain, for example, the non-linear relationships in the process variables (Kramer, 1991, 1992).

The function learned by the autoassociative neural network is the identity function where the network inputs are equal to the network outputs. Normally, there is an inner constraint in the neural network in order to prevent the network from memorising the input data. This constraint consists of a bottle-neck layer. This means that one of the hidden layers must have few neurons than inputs. If the number of neurons in the bottle-neck layer is A, the inequation $A < J$ must be satisfied, where J is the number of inputs or process variables considered.

The structure of an autoassociative neural network for the implementation for the NLPCA can be seen in figure 3.6. This structure usually consists of a feedforward neural network with the following hidden layers:

- The first hidden layer is called the mapped layer. The transfer function of the neurons is usually non-linear, i.e., sigmoid in this layer.

- The second hidden layer is the bottle-neck layer. The nodes of this layer can have a linear or a non-linear transfer function. The number of nodes must be lower than the number of measured variables J.

- The third hidden layer is called the demapped layer. Normally, the transfer function in the neuron outputs is non linear in this layer.

The input layer, the mapped layer and the bottle-neck layer form the non-linear function g. This function transforms the original vari-

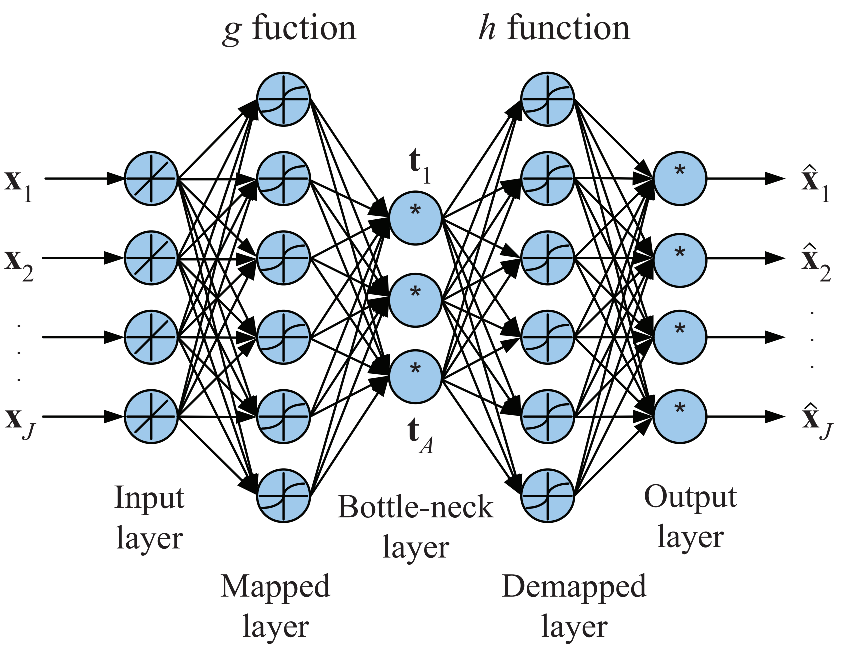


Figure 3.6: Structure of an autoassociative neural network for NLPCA.

ables into the non-linear latent variables. This mapped procedure can be expressed as:

$$\mathbf{t}_a^T = g_a(\mathbf{z}^T), a = 1, ..., A \tag{3.15}$$

where $\mathbf{t}_a^T$ is the a-th output of the bottle-neck layer.

On the other hand, the bottle-neck layer, the demapped layer and output layer form the non-linear function h. This function generates a reproduction of the original variables using the bottle-neck outputs:

$$\hat{\mathbf{z}}_j^T = h_j(\mathbf{t}^T), j = 1, ..., J \tag{3.16}$$

where $\mathbf{t} = [t_1, t_2, ..., t_A]$ are the outputs of the bottle-neck layer, the non-linear latent variables or the non-linear scores.

The mapping and demapping phases are a non-linear generalization of the principal components analysis. The loss of information associated with this process of mapping and demapping can be measured as the sum of the squared errors between the input and output signals during the training stage.

$$e_{SPE} = \sum_{n=1}^{N} \sum_{j=1}^{J} (x_{nj} - \hat{x}_{nj})^2 \tag{3.17}$$

During the training stage, e_{SPE} must be minimized.

There is no stabilised method for choosing the dimension of the mapped and demapped layers. If these layers have too many nodes, overfitting problems can appear. The following inequation gives an idea of the size of these layers with respect to other parameters of the network (Kramer, 1991, 1992):

$$M_1 + M_2 \ll J(N - A)/(J + A + 1) \tag{3.18}$$

where M_1 is the number of nodes in the mapped layer and M_2 is the number of nodes in the demapped layer. Cross validation procedures can be suitable for determining the neural network structure.

There is no method either for establishing the number of nodes in the bottle-neck layer, that is, the number of non-linear scores. An option could be to determine the suitable number of linear components using a cross validation procedure.

3.3 Case study: simulated two-communicated-tanks

The system used to test and analyse all the explained detection method based on the PCA approach is a simulation model of a laboratory plant. This plant, whose scheme is shown in figure 3.7, is formed by two cylindrical tanks T_1 and T_2, both with the same transversal area, connected by a cylindrical pipe p_{12} with a valve. The flow through this pipe is q_{12}.

The control objective of the plant is to maintain the level of the two tanks at a desired reference, which is accomplished with two PI controllers, one for each tank. The levels of the tanks h_1 and h_2

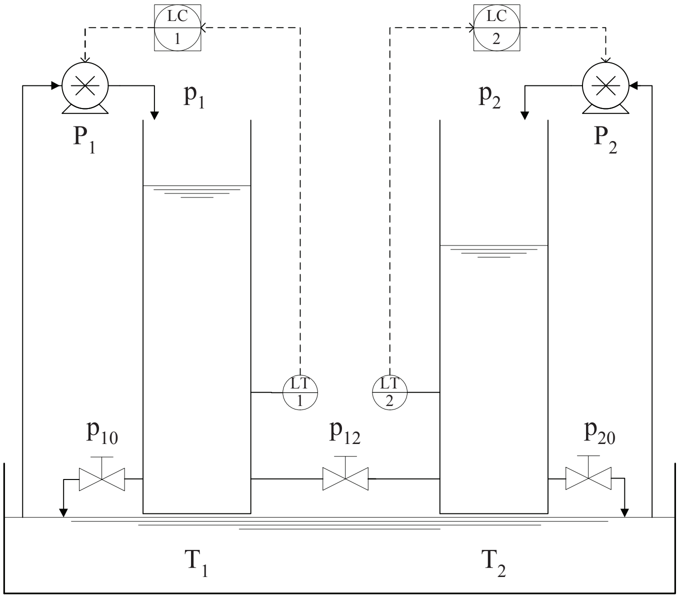


Figure 3.7: Two tank system.

are measured using two level sensors LT_1 and LT_2. Both tanks have been equipped with pumps (p_1 and p_2) for supply flow q_1 and q_2 respectively. The water is drained from tanks T_1 and T_2 by means of two pipes p_{10} and p_{20} respectively. The flows through these pipes are q_{10} and q_{20}. The measured variables are the levels of the tanks T_1 and T_2 and the flows produced by the pumps. A detailed description of the hardware and software of this plant and the model can be found in Fuente et al. (2008).

Three parameters were estimated in order to solve the discrepancies between the real system and the simulation model. Several parameter estimation techniques were performed to set the best parameters. This work is presented in García-Álvarez et al. (2011a).

Two additive faults were considered in this system. Both faults were induced in level sensors LT_1 and LT_2. This type of fault consists of introducing abruptly a bias in the sensor signal.

Three different cases were considered in the behaviour of the plant for the comparison purpose of this chapter:

- **Case 1:** In this case, the level reference is fixed to 15% in tank T_1 and to 23% in tank T_2. So one operating mode is only considered. This experiment is formed by 6001 time samples. A sensor fault of 20% size in sensor LT_1 is induced at the time sample number 3000.

- **Case 2:** This case is similar to the previous one, but in this experiment, a second fault is introduced. The second fault is a sensor fault in level sensor LT_2 with a size of 20% and it is triggered at the time sample number 4500.

- **Case 3:** This experiment presents several changes in the setpoint of both levels h_1 and h_2. Figure 3.8 shows the different steps in both references. In this case, a fault with the same characteristics as in case 1 is induced, i.e., a sensor fault of 20% in sensor LT_1 at the time sample number 3000.

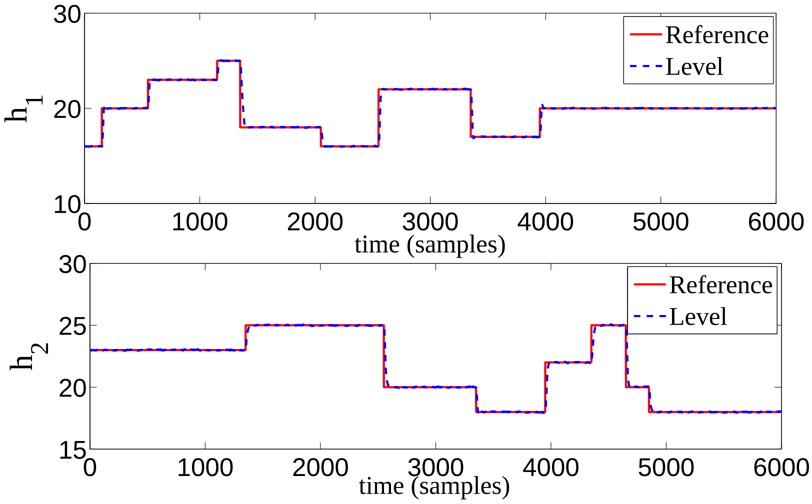


Figure 3.8: References and tank levels for case 3.

The same PCA model can be established for case numbers 1 and 2. In both cases, the PCA model must be established using data from the plant running in a similar operating mode, as changes in the operating mode can be detected as a fault, as the following chapter explains. For case number 3, the PCA model must be established with data that belong to the different operating modes considered in this experiment. This consideration is necessary because if an operating mode is not considered in the PCA model, it can be detected as a fault during the monitoring phase.

The measured variables considered in this example are the two level sensor signals h_1 and h_2, and the two actuator signals of the pumps q_1 and q_2. The sampling time of this example is 1 second.

3.4 Fault detection

In this section, the different monitoring and fault detection schemes tested in the three explained cases are presented. The graphical monitoring using T^2 and Q is shown and discussed. In all methods, the value α in the monitoring statistics limits is 0.01.

3.4.1 Fault detection using PCA

Figure 3.9 shows the monitoring evolution using the T^2 and Q statistics and how both control charts are able to detect the fault sensor LT_1. The fault is clearly detected at the fault occurrence instant. In both cases, a logarithmic scale has been used. The fault detection based on classical PCA can detect faults in cases that are similar to the data used to perform the PCA model and there are no important changes in the operating mode. Looking at the graphic, several alarms can be seen, but they do not last long and cannot be confused with faults.

If case number 2 is monitored (figure 3.10), the second fault cannot be detected using the classical PCA configuration. The first fault causes the T^2 and Q statistics to trigger, and when the second fault occurs, both statistics signals are over the thresholds.

The monitoring statistics evolution, when case number 3 is monitored using the classical PCA approach, can be seen in figure 3.11. In this case, the different transient states between the operating modes generate a sufficient consecutive number of alarms to consider these

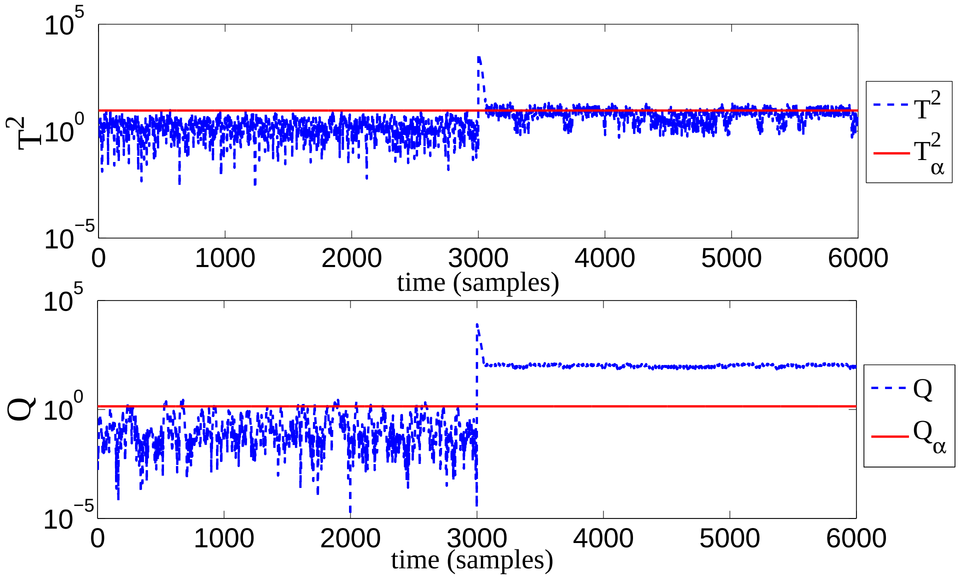


Figure 3.9: Monitoring of case 1 using PCA.

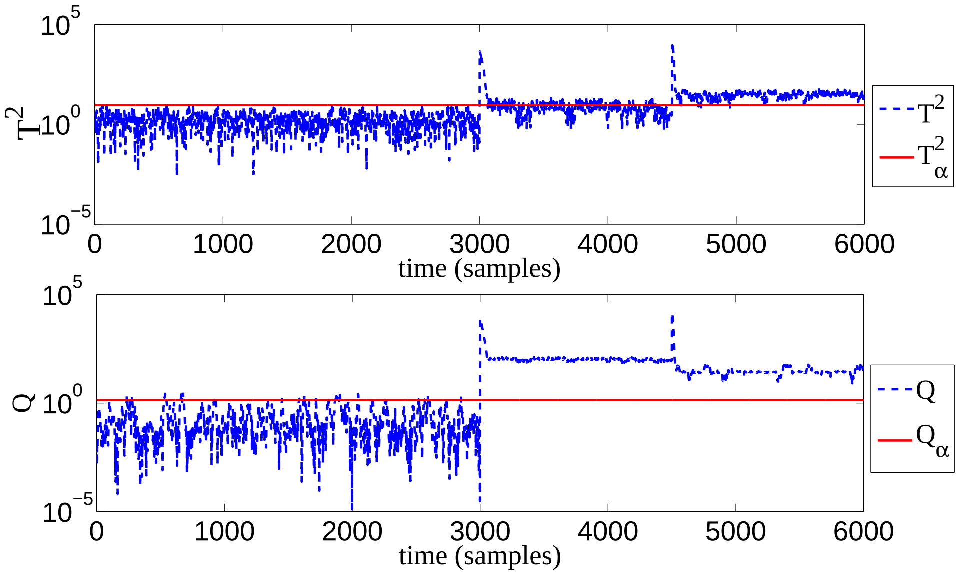


Figure 3.10: Monitoring of case 2 using PCA.

situations as faults. Also, in the T^2 statistic monitoring, it is not possible to detect the induced fault. The Q statistic control chart detects the transient states as faults and it is not easy to know when a fault or a change in the operating mode has occurred. The monitoring of processes with several operating points using classical PCA generates a loss of detectability.

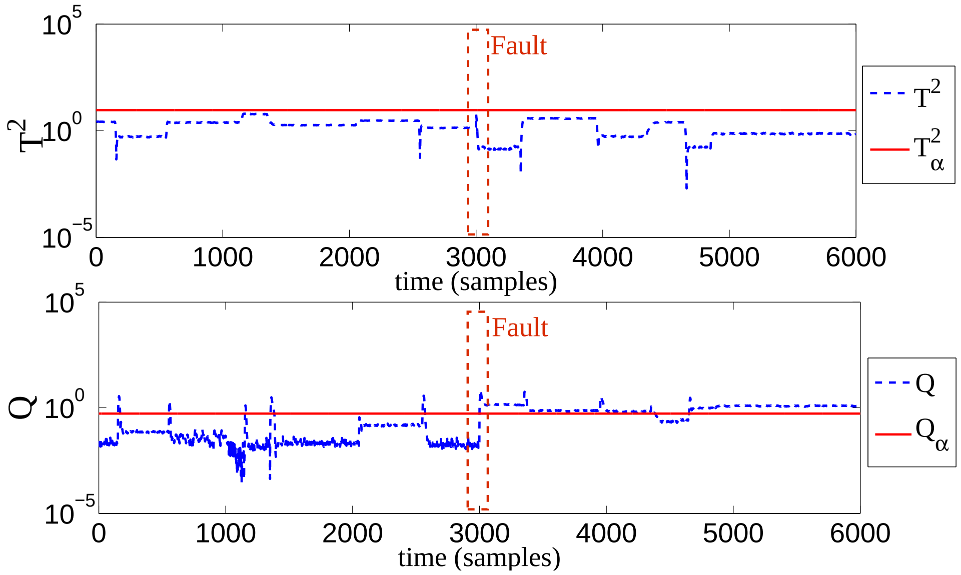


Figure 3.11: Monitoring of case 3 using PCA.

In this method, and the other PCA-based methods considered, all PCA models were arranged with two components.

3.4.2 Fault detection using DPCA

The monitoring evolution using the T^2 and Q statistics with a dynamic PCA model is shown in figure 3.12. In this example, the number of past values considered in the dynamic model was 5. The fault is clearly detected at the fault occurrence time in a similar way to the classical PCA approach.

When the two consecutive faults of case 2 are monitored using the DPCA, the same results as in the use of the classical PCA approach

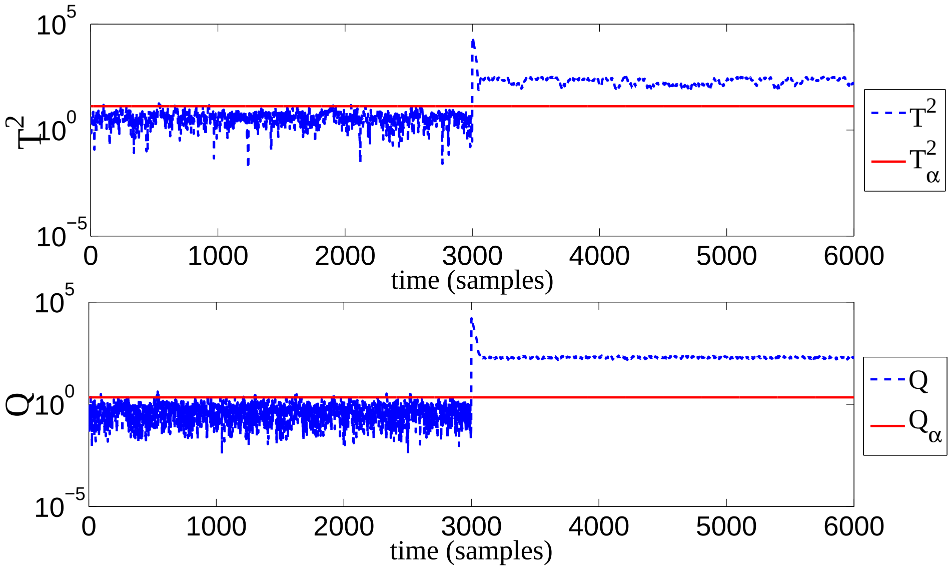


Figure 3.12: Monitoring of case 1 using DPCA.

are obtained, as figure 3.13 shows.

Also, as in the previous two cases, the monitoring of case 3 is quite similar to the monitoring using the classical PCA approach. The T^2 statistic control chart is not able to detect the sensor fault induced, as figure 3.14 shows. Looking at the Q statistic monitoring, it is not easy to distinguish between the transient states of this experiment and the induced fault.

3.4.3 Fault detection using APCA

Figure 3.15 shows the evolution of the monitoring statistics T^2 and Q when the experiment of case number 1 is monitored using the adaptive version of the PCA explained in the previous section. Both control charts clearly detect the fault and then they come back under the threshold when a new PCA model is established.

If the second case considered in this work is monitored, both faults are detected as figure 3.16 shows. Both monitoring statistics T^2 and Q are able to detect the second fault induced in this experiment, due

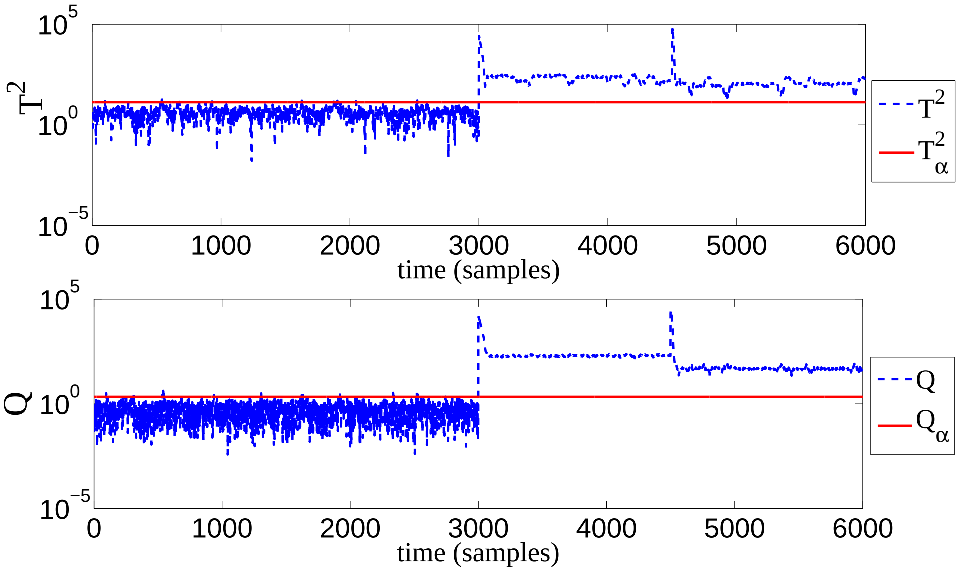


Figure 3.13: Monitoring of case 2 using DPCA.

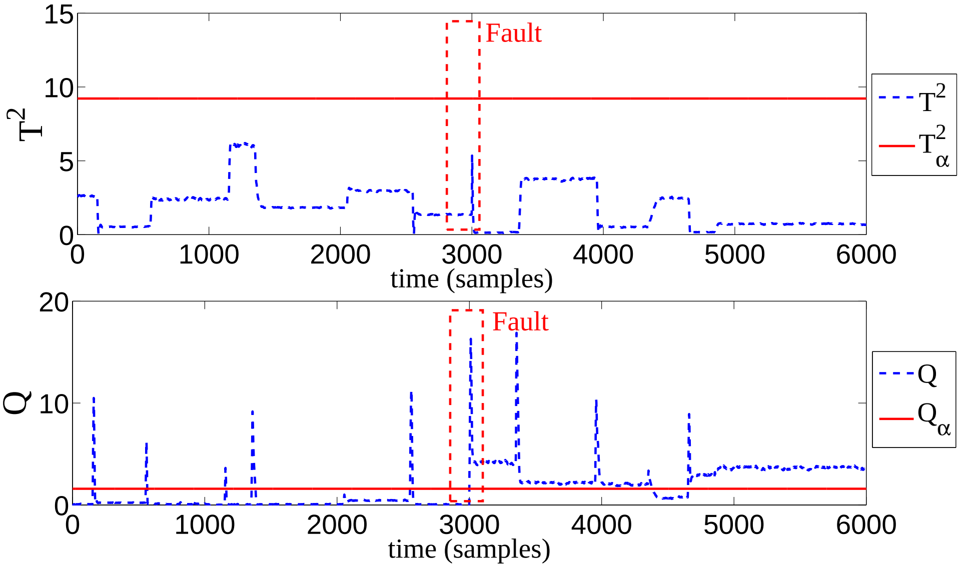


Figure 3.14: Monitoring of case 3 using DPCA.

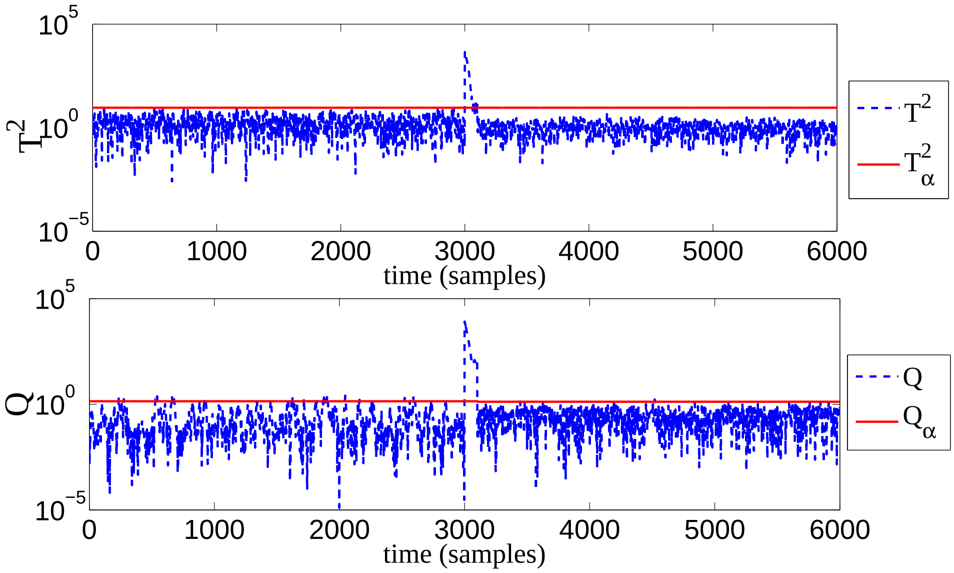


Figure 3.15: Monitoring of case 1 using APCA.

to the calculation of a new PCA model for the new faulty situation. T^2 statistic presents a better behaviour than the Q statistics because, after the second fault, the number of alarms is not very high and a third fault could be detected. In this case, the value of parameter N_{aux} is 100 samples because, with fewer samples, the PCA model arranged is not very accurate and many alarms are detected. A higher value of N_{aux} may produce better PCA models after a fault is detected.

Figure 3.17 shows the process monitoring using the APCA approach in the experiment with several operating modes. In this case, the results are not very successful, as it is not easy to distinguish between the transient states and the induced fault. Before the fault occurrence, the transient states are not detected as faults. However, after the fault detection, the monitoring scheme based on the newly established PCA model is not able to come back under the upper limit and future fault could not be detected.

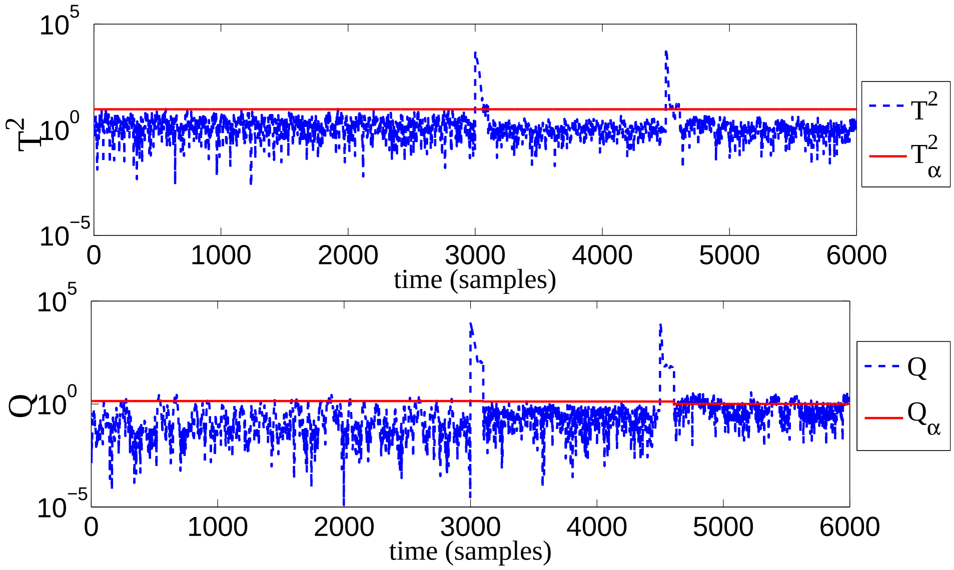


Figure 3.16: Monitoring of case 2 using APCA.

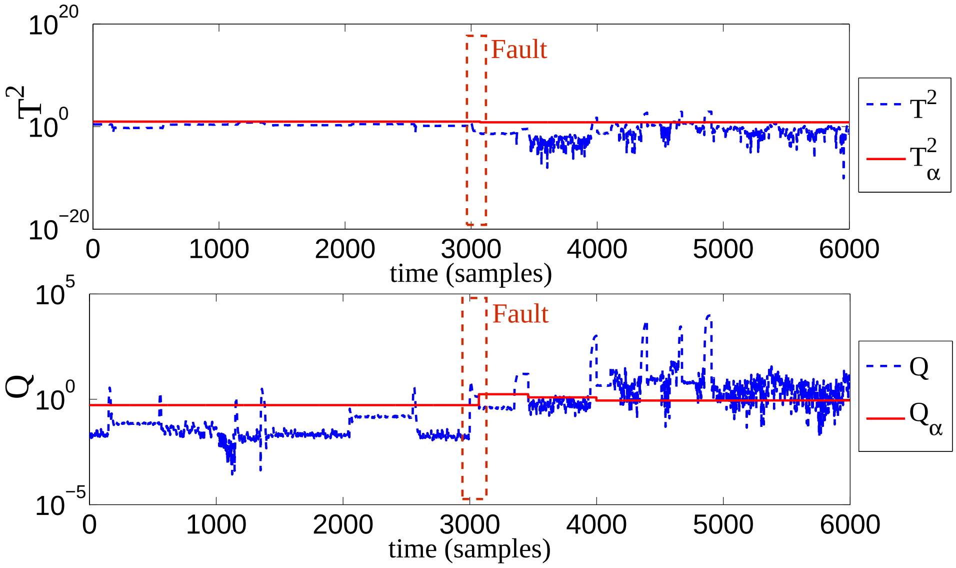


Figure 3.17: Monitoring of case 3 using APCA.

3.4.4 Fault detection using MSPCA

The monitoring of the first case using the multiscale PCA approach explained in section 3.2.3 can be seen in figure 3.18. The evolution of the Q statistics of frequency component A_3 clearly detects the induced fault.

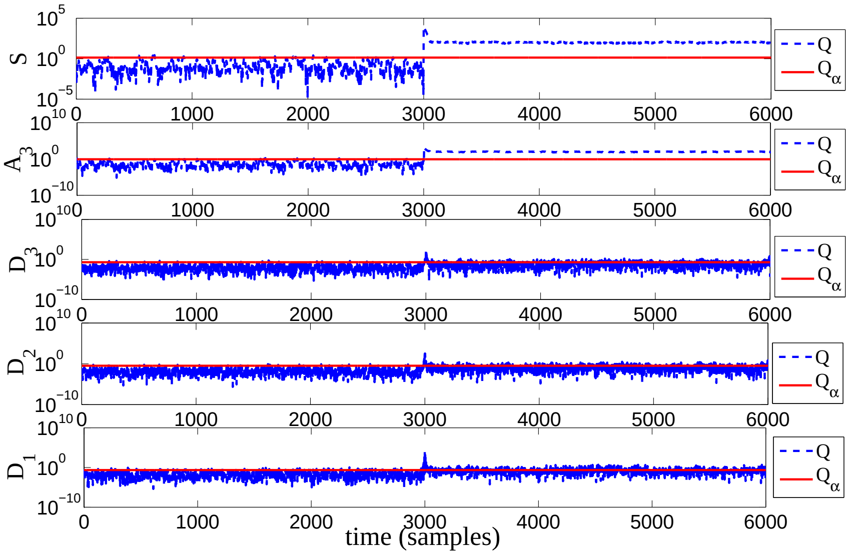


Figure 3.18: Monitoring of case 1 using MSPCA.

The consecutive second fault of the experiment considered in case 2 is not detected in this monitoring approach. The monitoring statistic is triggered when the first fault is detected and so the second fault cannot be detected in the same way as the classical PCA and DPCA schemes.

The process monitoring results of the third considered case are no better than the previous methods evaluated. It is not easy to distinguish between faults and transient states, as figure 3.20 shows.

3.4.5 Fault detection using EWPCA

If the process monitoring scheme is based on the exponentially weighted PCA method, the results are very similar to the APCA approach. Fig-

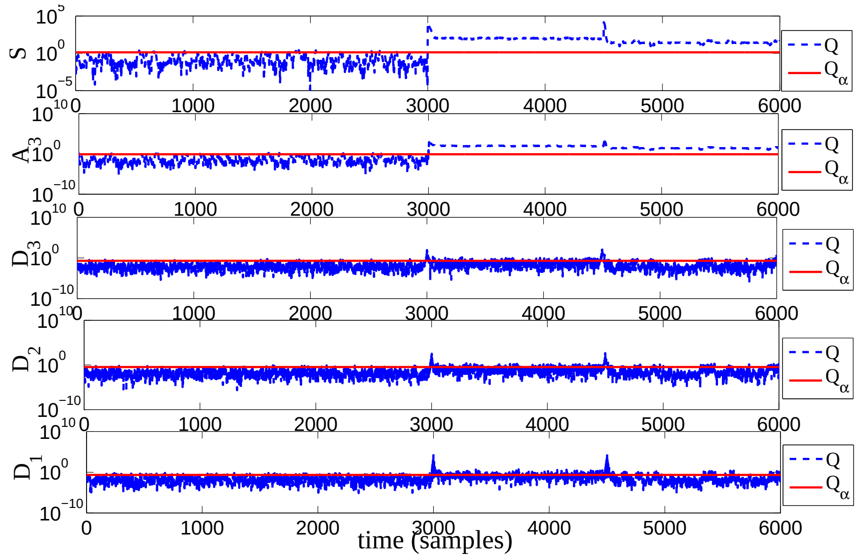


Figure 3.19: Monitoring of case 2 using MSPCA.

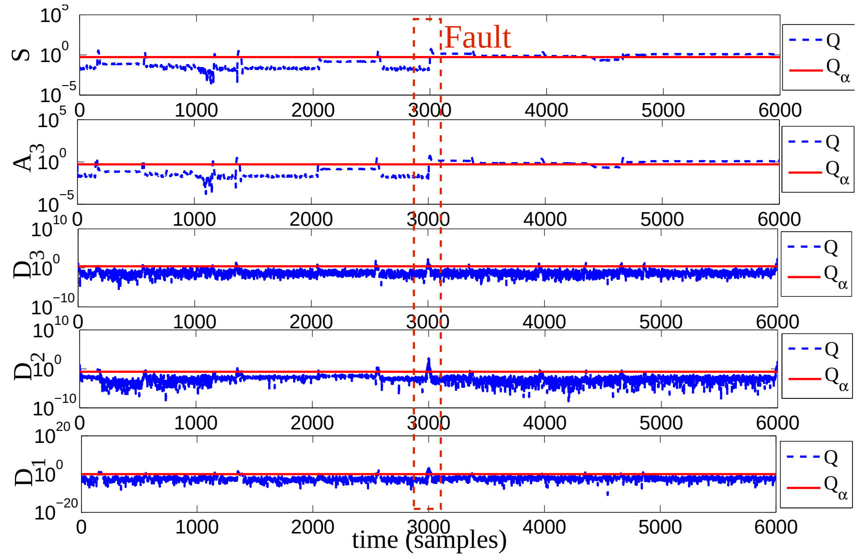


Figure 3.20: Monitoring of case 3 using MSPCA.

ure 3.21 shows the detection of the fault of case number 1. When the fault is detected, both statistics come back under the upper limit due to the adaptive behaviour of this approach.

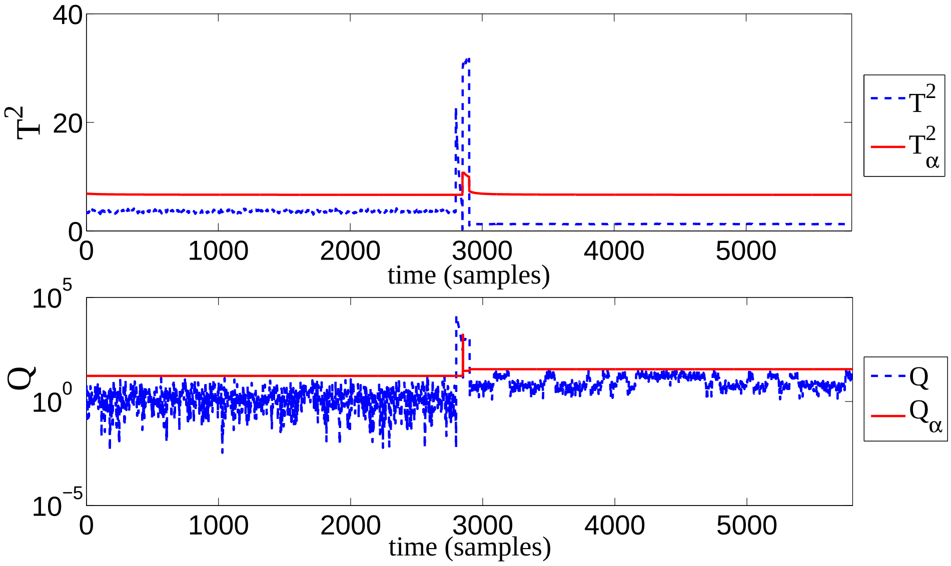


Figure 3.21: Monitoring of case 1 using EWPCA.

Thanks to this cited adaptive behaviour, this method is able to detect the two consecutive faults of the second case considered. Figure 3.22 shows how the T^2 and Q statistics detect the first fault, the adaptation to the new faulty situation and the detection of the second fault, in this case, only by the Q statistic. In the same way as in the APCA case, the value of parameter N_{aux} is 100 samples because, with fewer samples, the PCA model is not very accurate and many alarms are detected. Also, a higher value of N_{aux} could generate more accurate PCA models.

When the experiment with several operating modes and transient states of case 3 is monitored, the results are no better than the previous cases. Figure 3.23 shows that several transient states are detected as faults, but the fault itself is not detected.

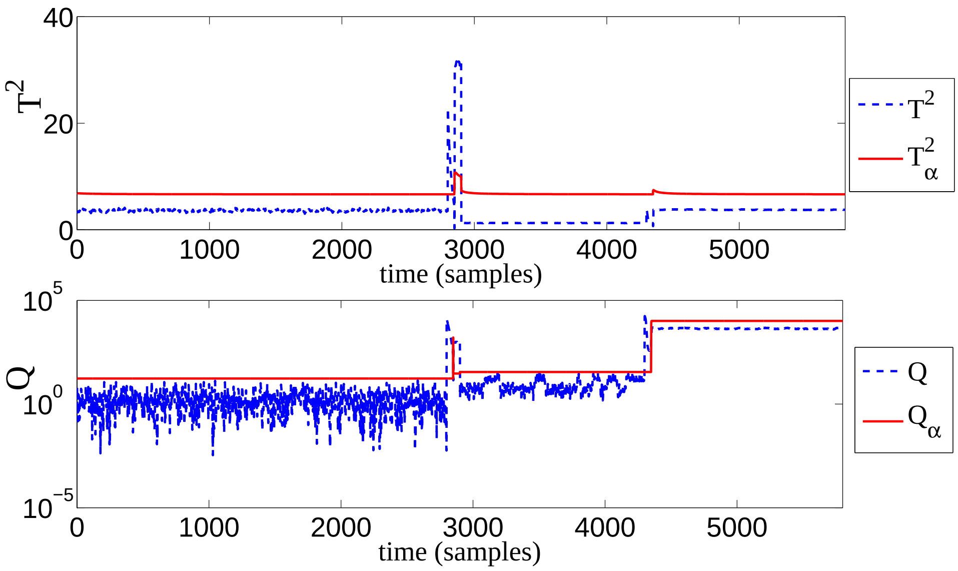


Figure 3.22: Monitoring of case 2 using EWPCA.

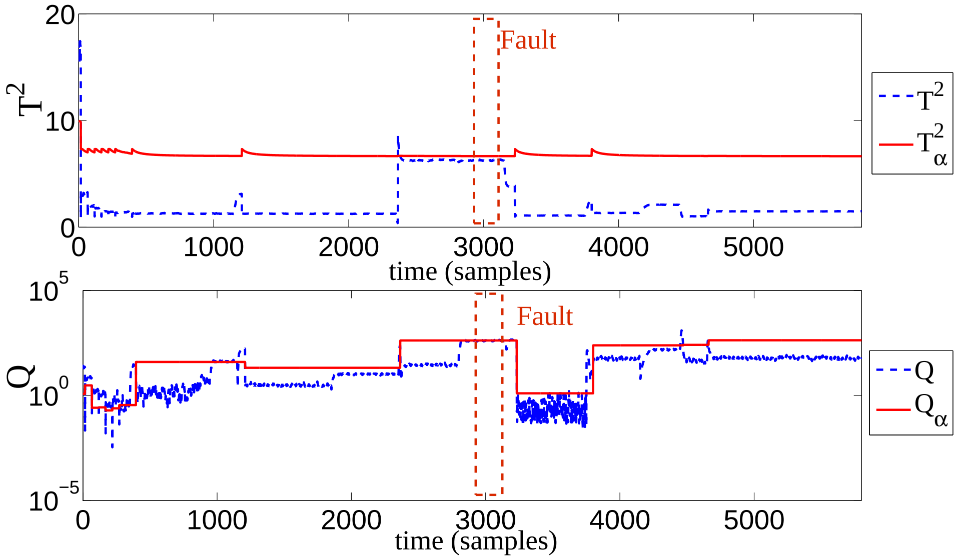


Figure 3.23: Monitoring of case 3 using EWPCA.

3.4.6 Fault detection using PCAEA

The monitoring results using PCA with the external analysis approach is very similar to the PCA or DPCA results for this case study. Figure 3.24 shows the detection of the sensor fault of the first case. In the implementation of this method, the reference signals of the levels of both tanks are considered as external variables.

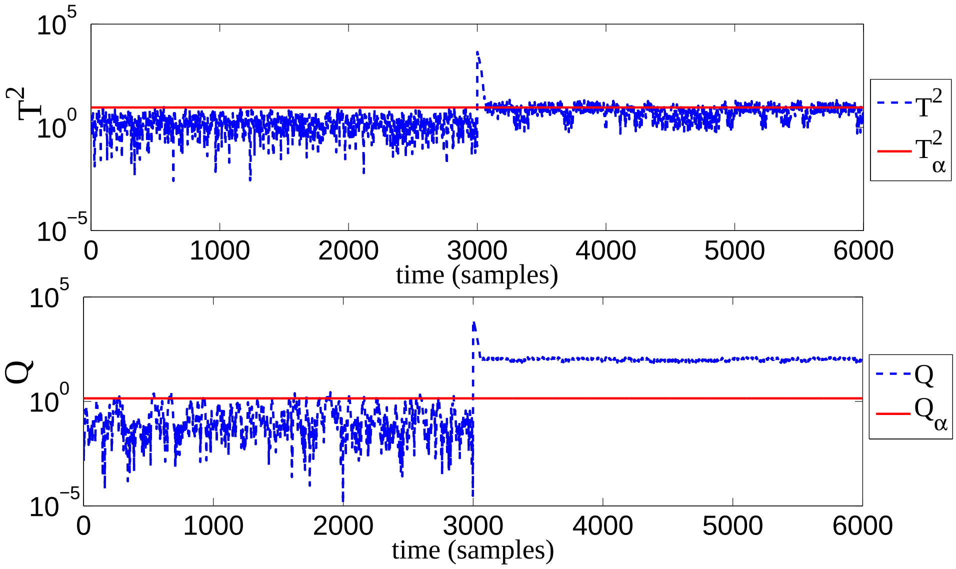


Figure 3.24: Monitoring of case 1 using PCAEA.

When the second case with two consecutive faults is monitoried, figure 3.25 shows how the second fault cannot be clearly detected.

In the monitoring of the third case, as in the previous case, this method does not present any advantage. The transient states are confused with faults and it is not possible to detect the fault, as figure 3.26 shows.

3.4.7 Fault detection using PCANLEA

For the implementation of the monitoring scheme using PCA with non-linear external analysis, a feedforward neural network has been

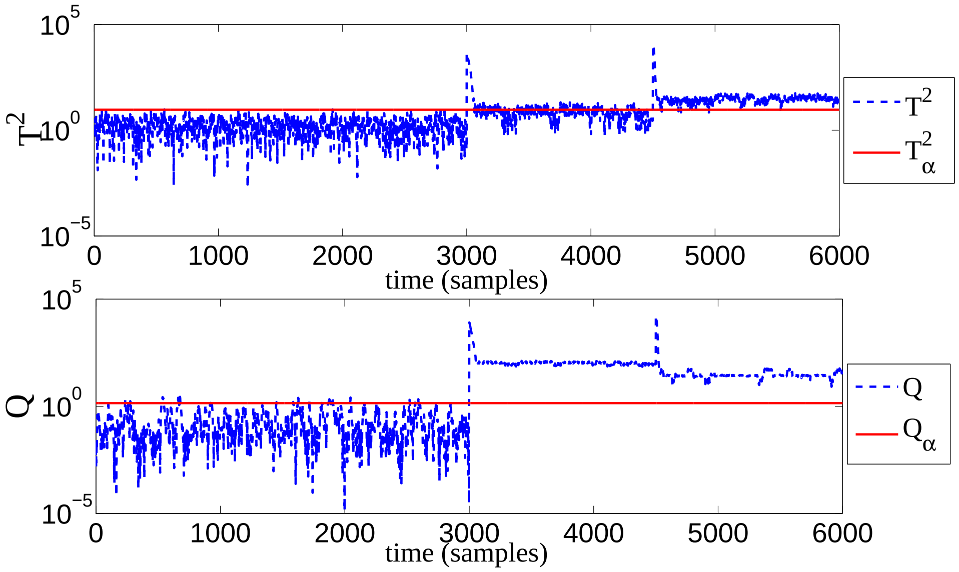


Figure 3.25: Monitoring of case 2 using PCAEA.

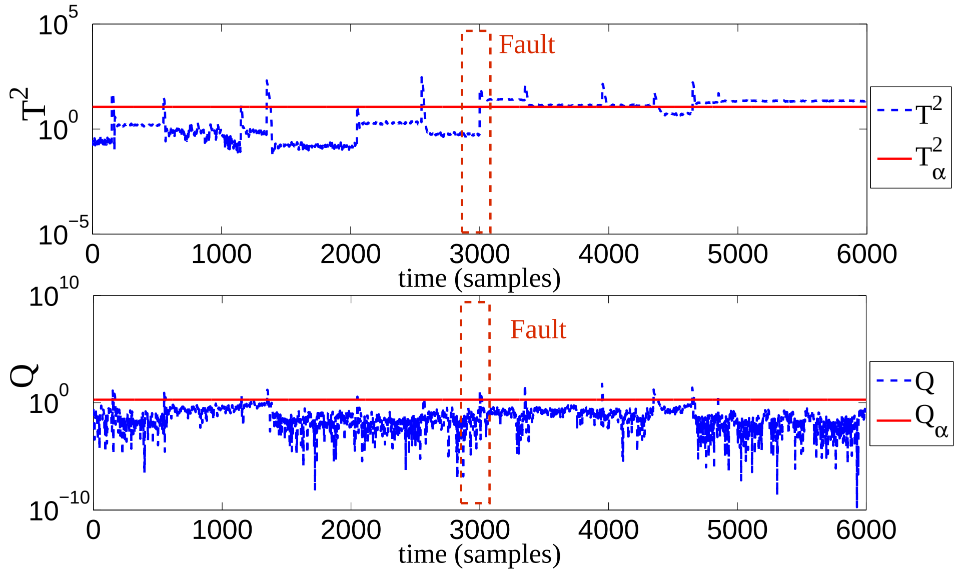


Figure 3.26: Monitoring of case 3 using PCAEA.

used. In this case, the number of past values of the external variables was 6. The ANN had one hidden layer with 16 sigmoid neurons and 4 linear neurons in the output layer. The training task was performed during 500 epochs with an expected error of 0.01.

The results shown in figures 3.27, 3.28 and 3.29 are quite similar to the ones obtained with the previous method.

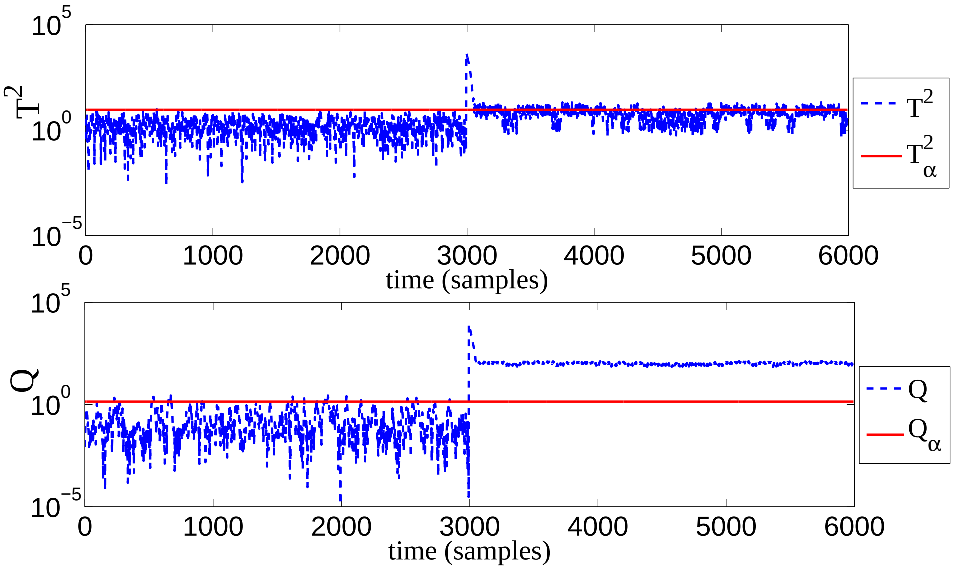


Figure 3.27: Monitoring of case 1 using PCANLEA.

3.4.8 Fault detection using NLPCA

Finally, the case study is monitored using a scheme based on a nonlinear PCA based on autoassociative ANNS. The autoassociative neural network used in this example had the following structure: 8 sigmoid neurons in the mapped layer, 8 sigmoid neurons in the demapped layer and 2 linear neurons in the bottle-neck layer. The training task was performed during 500 epochs with an expected error of 0.0001.

For the two first cases, the ANN was trained using data from the same operating mode. However, for the third case, the ANNs were trained using data from the different operating modes considered.

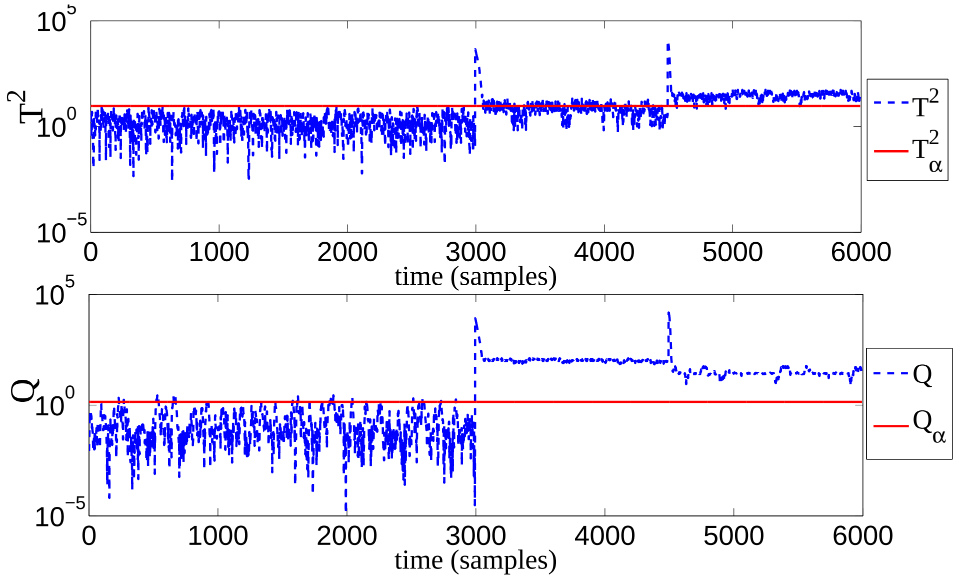


Figure 3.28: Monitoring of case 2 using PCANLEA.

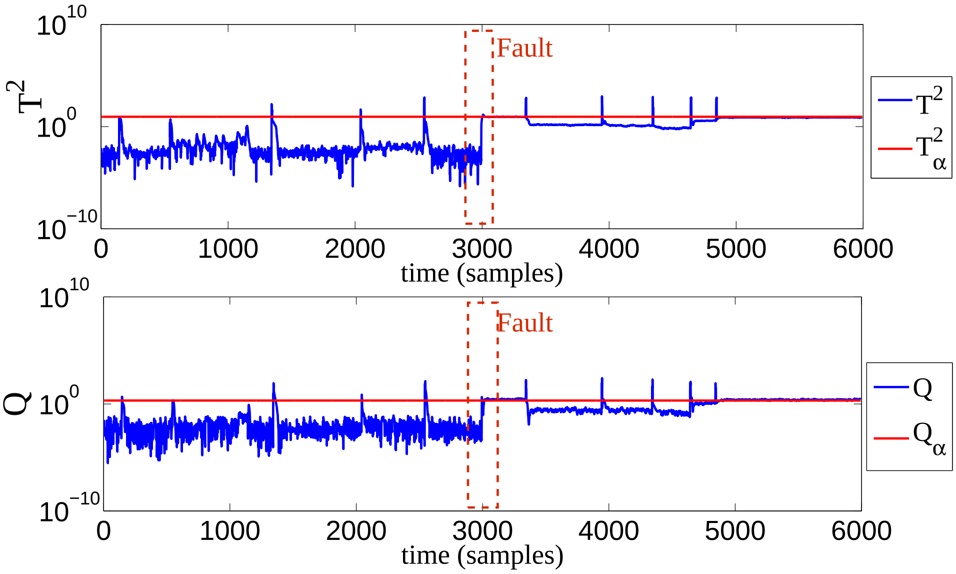


Figure 3.29: Monitoring of case 3 using PCANLEA.

Figure 3.30 shows the detection of the sensor fault of case 1. In this case, the Q statistic detects the fault more clearly.

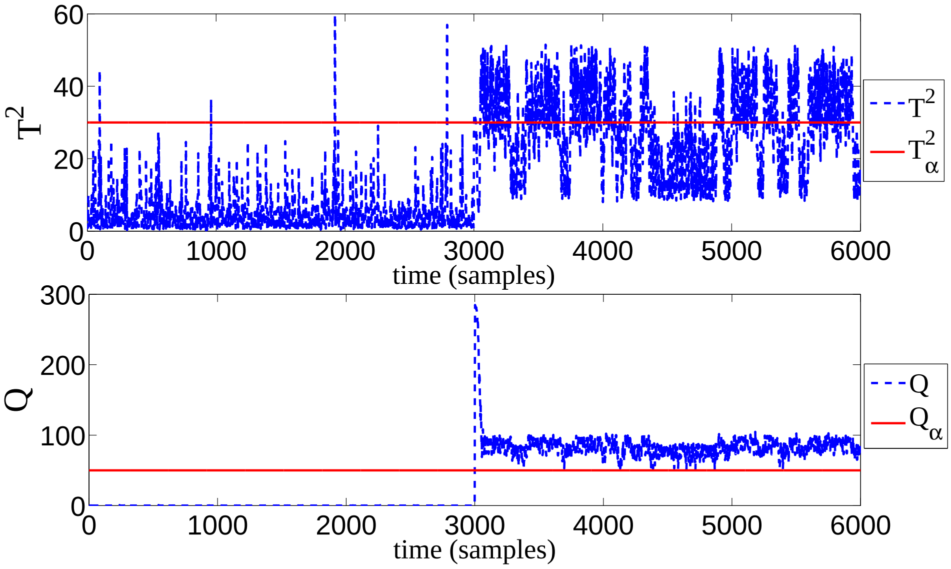


Figure 3.30: Monitoring of case 1 using NLPCA.

As this method does not have an adaptive behaviour, the second fault of case number 2 is not detected, as figure 3.31 shows.

The monitoring of the third case does not present any improvement with respect to the previous methods considered in this work. The monitoring of this case is shown in figure 3.32.

3.5 Results

A comparative analysis of the results of the different methods applied to the case study of this chapter is presented in table 3.1. Several parameters have been considered. These parameters can be divided into two groups: quantitative and qualitative. The quantitative parameters are the type I errors percentage, the mean CPU time, the fault detection delay and the minimum fault size detected. The considered qualitative parameters are the detection of multiple faults and

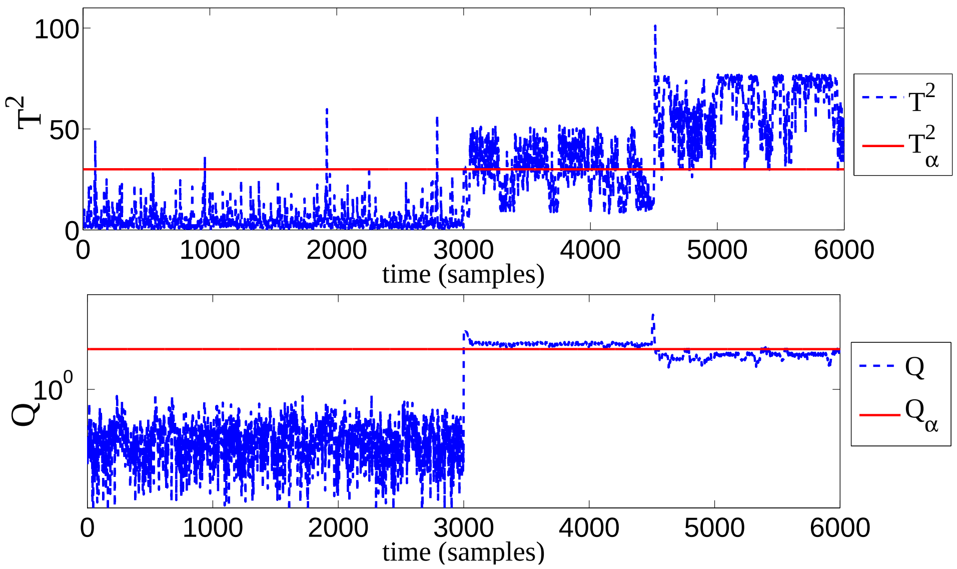


Figure 3.31: Monitoring of case 2 using NLPCA.

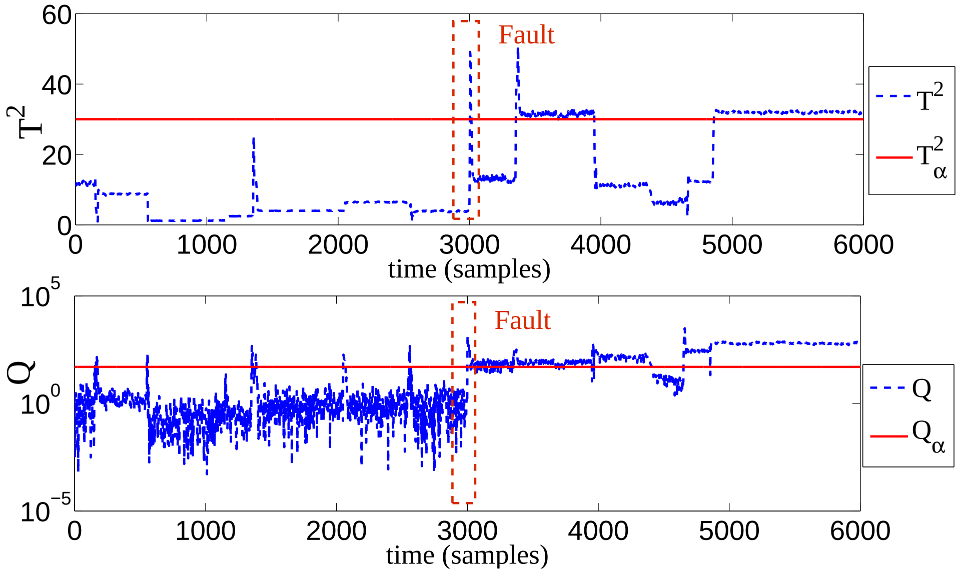


Figure 3.32: Monitoring of case 3 using NLPCA.

the monitoring of several operating modes and transient states.

	TIE	CPUT	FDD	MFSD	DMF	MOMTS
PCA	4.62%	$8.03 \cdot 10^{-5}$	10	10%	No	No
APCA	2.90%	$4.2615 \cdot 10^{-4}$	10	5%	Yes	No
MSPCA	2.82%	$1.8054 \cdot 10^{-4}$	10	3%	No	No
DPCA	1.80%	$8.53 \cdot 10^{-5}$	10	2%	No	No
EWPCA	0%	0.0029	12	3%	Yes	No
PCAEA	2.92%	$9.2721 \cdot 10^{-5}$	12	2%	No	No
PCANLEA	2.83%	0.0027	10	18%	No	No
NLPCA	0	0.0034	10	19%	No	No

Table 3.1: Comparative analysis.

The parameters used in this work are explained next:

- **TIE**: Type I errors or false positives percentage. This measurement is the percentage of samples where the monitoring statistics reaches the thresholds without the appearance of a fault. This percentage is calculated using experiments formed by 6001 samples. All the methods reduce the number of false positives with respect to classical PCA monitoring approach. The EWPCA and NLPCA do not present false positives for this case study.

- **CPUT**: Mean CPU time in seconds. This parameter calculates the mean time to process every sample during the monitoring phase. This task involves scaling the new sample, performing the transformation required by every method, computing the T^2 and Q statistics and testing if a fault has occurred. The CPU time was calculated using a PC with Intel© Core™2 Duo 3GHz processor and a 3Mb memory. The experiments were run using Matlab© 7.0 over the operating system Windows XP©. All methods present similar CPU times except the EWPCA, PCANLEA and NLPCA methods, which present significantly higher CPU times.

- **FDD**: Fault detection delay. This parameter measures the number of samples between the fault occurrence and the fault detection. All methods, except the EWPCA and PCAEA, were able to detect the faults instantaneously since, in this case study, faults are detected when 10 consecutive alarms are detected. So the minimum fault detection delay is 10 seconds. In the case of the EWPCA and PCAEA monitoring approaches, the fault detection delay was 12 seconds.

- **MFSD**: Minimum fault size detected. The minimum size of a fault that every PCA-based method can detect is measured by this parameter. In this case, a fault in the level sensor of tank 1 was induced. The methods based on neural networks show the worst results, while the other methods improve this parameter with respect to the classical approach.

- **DMF**: Detection of multiple faults. This parameter analyses the possibility of detecting consecutive faults, that is, the detection of a second fault after a fault has been detected. As cited before, the APCA and EWPCA methods can detect consecutive faults due to their adaptive procedures, which allow the monitoring statistics to come back under the upper limit when a fault is detected.

- **MOMTS**: Monitoring of several operating modes and transient states. This parameter analyses if the methods are able to distinguish between changes in the operating mode and faults, i.e., if the transient states between the operating modes are detected as faults or not. In general, the methods studied in this work cannot deal with this problem. This situation is studied in the following chapter.

3.6 Discussion

The comparative analysis presented in this chapter allows us to see the different improvements of the main PCA-based methods.

All considered methods reduce or remove the number of false positives, so it can be taken into account when the monitoring scheme produces many false positives , thus causing stress to the plant operators. When there are so many false positives, the plan operators may ignore them and this situation can have disastrous consequences.

In real-time systems with a low sampling time, the CPU time of monitoring can be taken into account. For this reason, methods with low computing complexity can be used in order to satisfy the real-time constraints.

If the fault detection sensibility, i.e., the detection of low size faults, is a constraint in the design of fault detection schemes, a method with a low value of this parameter should be used.

The combination of several of these methods can be a good solution, for example, the adaptive feature can be included in order to detect consecutive faults.

Finally, this work shows that the monitoring of processes with several operating modes or transient states is not very effective, even with adaptive or dynamic approaches. In the next chapter, this drawback will be studied and analysed.

Chapter 4

Monitoring of transient states using PCA

4.1 Introduction

It is very usual that continuous processes go through several operating modes or steady states due to changes in product set-points, feed flowrate or compositions. These situations can produce changes in the covariance structure used by PCA to build the data-driven model. These changes are normally detected as faults by the classical PCA-based monitoring approach, as chapter 3 presents.

The monitoring and fault detection in transient states and start-ups using MSPC techniques, such as PCA, has been discussed by several authors. Duchesne et al. (2002) propose a multivariate SPC scheme to monitor transition trajectories to achieve reduction in transition time and amounts of off-grade materials during transitions. This method was illustrated using a simulated fluidized-bed process to produce linear low-density polyethylene. The authors of this paper take into account the unfolding and alignment problems that appear in this type of problems, but they do not include the imputation problem, i.e., the missing data that appear when this methodology is applied in *on-line* process monitoring.

In Kourti (2003b), an analysis of the problems that appear in batch process monitoring are addressed, examining different methods that have appeared in the literature, studying their assumptions, advantages, disadvantages and their range of applicability. The nature of the transition data (start-ups and grade transitions) is also discussed, and issues relating to aligning, centering, scaling and unfolding such types of data are presented. However, the imputation problem is not explained systematically.

Zhang et al. (2003) and Zhang and Dudzic (2006) also use this type of multivariate statistical approaches to monitor a continuous steel casting process, but the missing data problem is also treated as indicated by Kourti (2003b).

In Zhao et al. (2007), a new method (STMPCA: soft-transition multiple PCA) is presented to solve the problem of batch processes operating in a variety of stages. The method divides the states into stationary and transitional ones, calculates the number of each type of state from historical data using a clustering algorithm and membership grades, and establishes a sub-PCA model for sub-stages and weighted sum models for transition regions. While in this method all the data are from batch processes, in this work the process is a continuous one and only the transitions are treated as batch processes.

In this chapter, a fault detection and isolation technique is proposed for continuous processes with several operating modes. The fault monitoring takes into account all the steady and transient states in the plant, building a classical PCA model for the first case (i.e., a PCA model for each steady state) and a modification of the PCA approach used in batch processes for the transient states.

4.2 Monitoring of processes with several operating modes and transient states

As presented in chapter 3, an important drawback of PCA is that it is not able to deal with processes where the operating conditions can change, i.e., when the current operating mode changes to another correct operating mode. Hwang and Han (1999) and Tien et al. (2004) classify the different solutions for monitoring this type of processes into three different categories:

1. Build a PCA model for each operation mode.

2. Update the model to reflect the changes in the operation modes.

3. Develop a conventional PCA model to account for all such changes.

In the third category, if the number of operating modes is high, the PCA model can record multiple correlation structures and this approach can lose detection power. This situation was seen in the third case considered in chapter 3 for every method considered. Figure 4.1 shows a hypothetical case where a PCA model was built with data from four different operating modes (OM_1, OM_2, OM_3 and OM_4). The diagram shows how a faulty situation falls into the Hotelling's statistics limits because this area is wide, due to the inclusion of different operating modes in the PCA model.

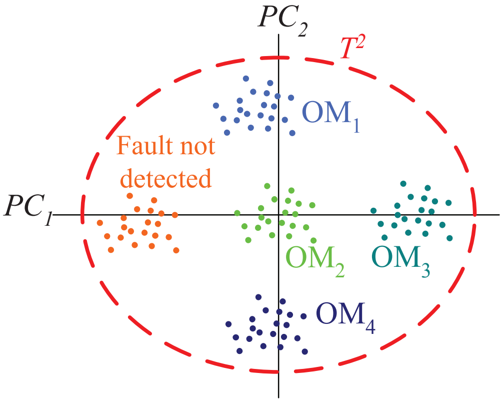


Figure 4.1: PCA model built with data from different operating modes.

The second category based on the updating of the model is similar to the adaptive formulations explained in section 3.2. This formulation is suitable for the detection of consecutive faults. Normally, changes in the operating mode are known, so they can be distinguished from faults. A new PCA model can be performed when a new operating mode is reached. One drawback of this formulation is that no fault can be detected during the *on-line* fault computation of the new PCA model.

The first category can be a better solution. A classical PCA model, or some of the variations cited in chapter 3, can be performed for every operating mode. The loss of detectability of the third case and the *on-line* reformulation of the PCA model with every operating mode detection of the second mode can be avoided. The adaptive formulations can only be used when faults are detected. One drawback of this category can be observed if there are so many operating modes in the process considered, because a PCA model for every case must be performed. In general, the statistical techniques are used in processes without many operating modes.

Another of the most important drawbacks in dealing with processes with multiple operating modes is monitoring transient states. Transient states are the phases between the steady states, due to changes in the references, changes in the inputs or effluents, etc. In transient states of dynamic systems, the relationships between the variables are highly non-linear. When transient states are monitored using a PCA model only, they are detected as special causes or faults as the study presented in chapter 3 shows.

Using a specific PCA model for every operating mode does not solve this problem because the transient state is usually detected as a fault for both PCA models due to this non-linearity, as the diagram of figure 4.2 describes schematically. In the steady state, it is reasonable to perform a linear model, but in the transient phase this is not possible. This scheme is the same for the start-ups of processes, due to the fact that, in this stage, process variables are rising and catching the set-points. One solution applied by some authors is to ignore the alarms produced by the transient stage. This can be possible if the transient stage is not very long and it can be ignored. However, when these transient states are long, it is better to monitor them, because one of the most important objectives of fault detection and isolation schemes is to detect faults as soon as possible.

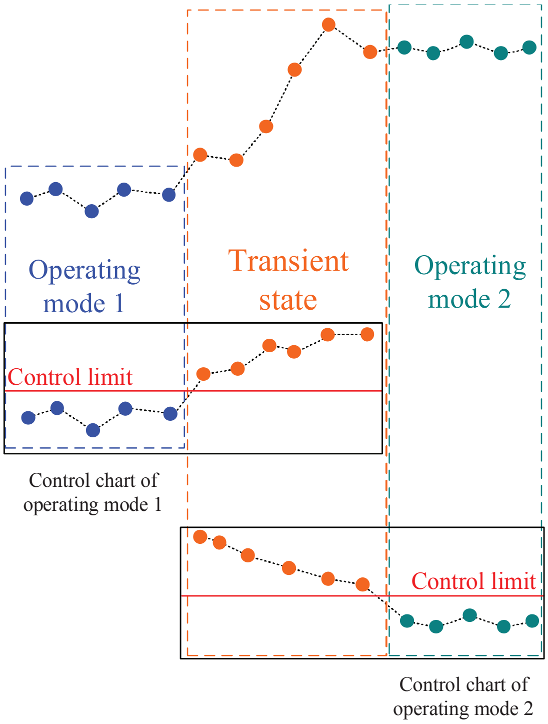


Figure 4.2: Monitoring of transient states.

In spite of this, there are non-linear problems that have been dealt with using PCA. This is the case of batch processes. This type of processes has many measured process variables and a time-varying and highly correlated structure. Dealing with batch processes using multivariate SPC methods is a well-studied matter; Nomikos and MacGregor (1995) employed multiway PCA (MPCA) or unfolded PCA (UPCA); Zarzo and Ferrer (2004) applies statistical process control methods to batch processes. This is possible because the information in the process variable trajectories can be projected into low dimensional latent-variable spaces. Other works of the use of UPCA in batch processes can be found in Nomikos (1996), Aguado et al. (2007) and Villez et al. (2009).

In this work, transitions or transient states between steady states have been treated as batch processes. However, as a batch procedure is applied to a continuous process, some problems such as unfolding, data alignment and imputation must be taken into account.

So, this chapter proposes a fault detection method for processes with multiple operation modes which consists of using different PCA models for each steady operation mode and a modification of the UPCA method to compute batch models for the transient states. The different and necessary tasks to perform this type of techniques are presented and explained systematically, specially the imputation methods, i.e., the missing data problems that appear when the UPCA method is applied to a continuous process and the different solutions are compared to be able to choose the best method in each case.

4.3 Unfolded PCA

To perform the unfolded PCA (UPCA) approach, first of all, a database of past transitions is necessary. In every $i = 1, 2, \ldots, I$ correct past transition, $j = 1, 2, \ldots, J$ variables are measured at $k = 1, 2, \ldots, K$ time intervals. All this data can be arranged into a three-way matrix $\underline{\mathbf{X}}(I \times J \times K)$ (following the notation of Kiers (2000)) as shown in the left hand part of figure 4.3.

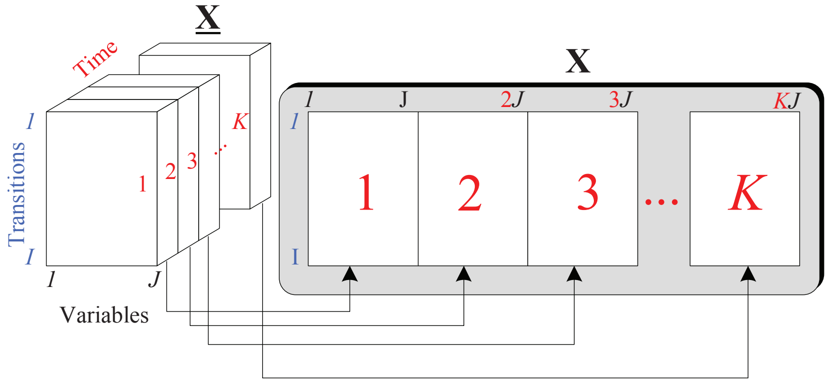


Figure 4.3: Matrix unfolding.

4.3.1 Unfolding

To apply UPCA, the three-way matrix must be unfolded into a two-way matrix structure. When the unfolding is performed, one of the directions remains unaltered while the other is the combination of the other two directions slice by slice. The unfolding can be developed in six different ways. Table 4.1 shows all the possibilities of unfolding based on Zarzo and Ferrer (2004). As the authors explain, possibilities B and C are similar, reordering the rows, while possibilities D and E are also equivalent, reordering the columns. Option F is the transpose of option A.

Unfolding type	**Matrix**	**Unaltered direction**
A	$KI \times J$	variables
B	$JI \times K$	time
C	$IJ \times K$	time
D	$I \times KJ$	batches
E	$I \times JK$	batches
F	$J \times IK$	variables

Table 4.1: Types of unfolding.

As Nomikos and MacGregor (1995) and Kourti (2003b) mentioned, option D is the most suitable when the historical data is to be analysed, but it is also quite applicable to *on-line* monitoring. Option A (Wold et al., 1998) is also used in *on-line* monitoring applications. In this work, option D is used to detect faults in transient states, i.e., the direction of transitions is maintained and the trajectory of all the process variables of the first sample time are arranged into the unfolded matrix, followed by the next sample time, and so on, as shown in figure 4.3.

There are other alternatives to deal with the three-way matrix without any unfolded procedure, such as PARAFAC (parallel factor analysis) or Tucker1 and Tucker3 techniques (Westerhuis et al., 1999).

4.3.2 Data alignment

When all data is collected from the process variables, including transition trajectories, it is not uncommon that the start-up and transitions between different operating modes have a different duration. This can be due to either external conditions of the process, i.e. climatological conditions, or the nature of the process itself, which can mean that the set-points were not achieved at exactly the same time under the same conditions. Figure 4.4 shows schematically a matrix $\underline{\mathbf{X}}$ arranged with transitions with different lengths. This matrix cannot be unfolded. For these reasons, it is necessary to apply an alignment method to correct, synchronize or align the trajectories of the variables in order to handle comparable data sets. However, as commented by González-Martínez et al. (2011) and Kourti (2003a), in more complex cases, the trajectories of the variables have different shapes, even when transient state durations are the same, indicating that the timing for key events during each batch is different. Therefore, equal transition lengths does not mean synchronized transitions. So the alignment of trajectories collected during transitions and/or batch processes is necessary.

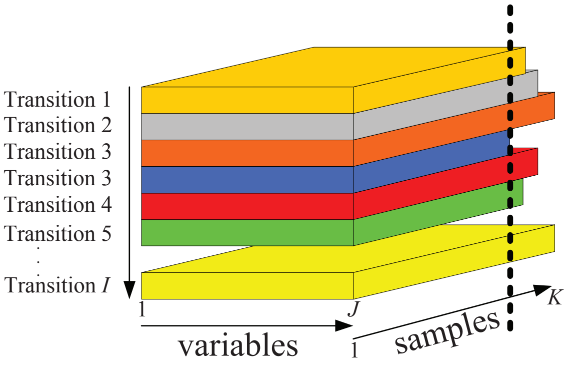


Figure 4.4: Data matrix arranged with transitions with different lengths.

During the steady states, the measured variables are collected every sample time. However, during the transient states, which can have a variable time duration, this is not possible and, because the three-way matrix must have the same number of measurements for every past transition in order to develop the unfolding methods, a method to align the data is necessary.

Indicator variable

A suitable method to align all this data is the indicator variable approach (Nomikos and MacGregor, 1995). When an indicator variable is used, a monotonically increasing or decreasing variable with the same starting and ending value for each batch must be found. This variable will be used to collect the data instead of time.

In this work, taking into account that transient states can be due to a step applied in the reference signal, the proposed indicator variable is the percentage of that reference signal (ΔR) reached by the process variable (y). This means that the process data will be collected when the process variable reaches, for example, $5\%, 10\%, \ldots, 100\%$ of the reference step value, as shown in figure 4.5.

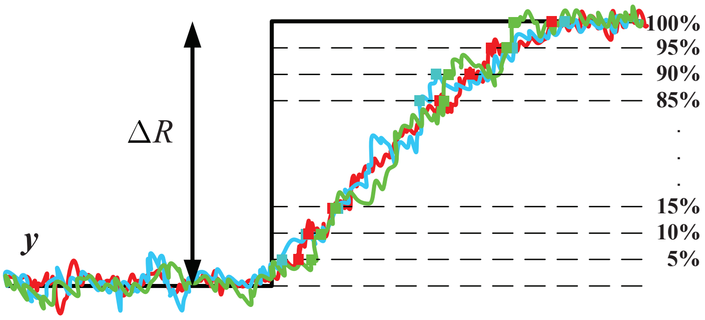


Figure 4.5: Transition trajectory alignment.

The indicator variable (r) takes the values:

$$\begin{aligned} \Delta R &= Ref_e - Ref_0 \\ \Delta r &= \frac{\Delta R}{n_s} \\ r_0 &= Ref_0 \\ r_i &= r_{i-1} + \Delta r,\ \ i = 0, 1, 2, \ldots, n_s \end{aligned} \tag{4.1}$$

where ΔR represents the step in the reference (Ref_0 is the value before the step and Ref_e after the step). Δr is the increase of the indicator variable if the step in the reference is divided into n_s portions.

The process variables during the transition or start-up will be collected when the process variable (y) reaches every value of r, instead of every sample time. During the monitoring phase, the new measured variables will be grouped using this criteria. In both cases, if the value of a particular variable is missing or unknown, it can be interpolated using the same kind of interpolation methods available in the literature.

During the *on-line* monitoring of a new transient state, if the indicator variable approach is applied, when the process variable arises a new stage into the reference step, the values of the process variables are stored in order to compute the monitoring statistics.

DTW

However, in other situations, the indicator variable method is not suitable, for example, when it is not possible to find a monotonically increasing or decreasing variable. A problem similar to this has been dealt with in speech recognition methodologies. In this field, some of these problems have been solved using dynamical time-warping (DTW) when past trajectory data is arranged (Kassidas et al., 1998).

In this work, the iterative method for the synchronization of batch trajectories proposed by Kassidas et al. (1998) has also been implemented and applied. Basically, for two multivariate trajectories of two different transitions **A** and **B**, both matrices of dimensions $K_1 \times J$ and $K_2 \times J$, where the number of samples K_1 and K_2 is not equal, DTW aligns both trajectories to the length of one of these or to a reference trajectory creating or eliminating some points. These processes of compression or expansion of the time scales must be performed by minimizing the dissimilarity between the two trajectories.

During the *on-line* monitoring of a new transient state, using the DTW approach, the *on-line* DTW implementation finds the point on the reference trajectory that best represents the progress of the new transient state at the current time and synchronizes the new trajectory to this point. The future behaviour of the transient state from the current time is unknown but can be predicted. This problem, called imputation, is discussed in the next subsection. The *on-line* DTW has been implemented following the guidelines mentioned in Kassidas et al. (1998).

Using the alignment approach based on the indicator variable described in this section (figure 4.5), where the transition is modelled with 20 samples, this number can be increased if the percentage step is reduced. In some problems, this small number of samples can be insufficient to model the transitions and the DTW approach must be applied. However, in some cases, i.e., fast transitions of a process with hundreds or thousands of measured variables, the indicator variable approach presented can simplify the complex matrix calculus during the alignment phase and, as shown in the next subsection, the imputation procedures.

Recently, a new algorithm for *on-line* DTW implementation has been proposed in González-Martínez et al. (2011). This algorithm is based on the DTW approach described in Kassidas et al. (1998) and includes a new time warping procedure for *on-line* batch synchronization based on relaxed greedy, called Relaxed-Greedy Time Warping (RGTW). Although this algorithm can present better results with the lowest false alarms ratio by reducing the *on-line* computational cost, the classical approach was selected because it has a simpler implementation and does not require all the *off-line* considerations required in GRTW.

4.3.3 Imputation

When transient states and the start-ups are being monitored using the UPCA approach, the following problem appears: the measured variables between the beginning of the transition and the current instant t are available, as figure 4.6 shows, but the measured variables between the current instant and the end of the transition are not available. So, it is necessary to predict this unknown future data, as it is needed to calculate the scores and the monitoring statistics.

There are several methods to deal with these missing data prob-

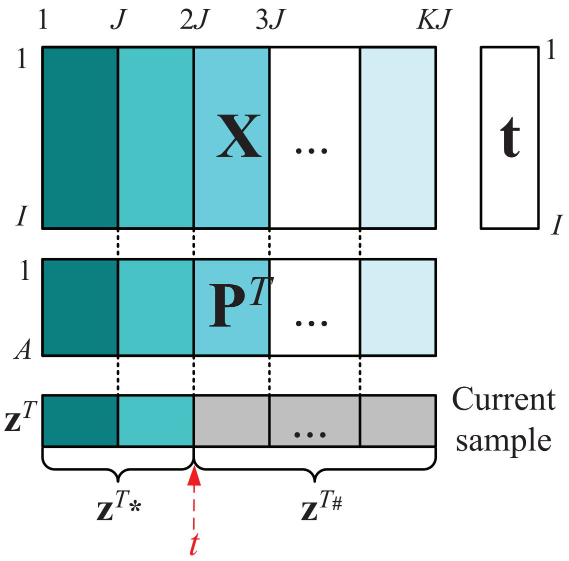


Figure 4.6: Imputation scheme.

lems. In Arteaga and Ferrer (2002) the principal missing data imputation methods are presented, explained and compared. In this chapter, three methods will be used: trimmed score methods (TRI), known data regression method (KDR) and trimmed score regression method (TSR).

Following the notation used in Arteaga and Ferrer (2002), a data matrix $\mathbf{X}$ can be considered as a collection of row vectors $\mathbf{z}_i^T$ (observations) or column vectors $\mathbf{x}_j$ (variables). The columns of the loading matrix $\mathbf{P}$ are $\mathbf{p}_j$. The score matrix will be considered as a set of row vectors $\boldsymbol{\tau}_i^T$ (scores in the ith observation) or column vectors $\mathbf{t}_i$ (latent variables) (figure 2.7).

A new observation $\mathbf{z}$ can be partitioned as:

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}^* \\ \mathbf{z}^{\#} \end{bmatrix} \tag{4.2}$$

where $\mathbf{z}^*(KJ - M \times 1)$ is the known past and current values of the process variables into the transition and $\mathbf{z}^{\#}(M \times 1)$ is the unknown future data values of the transition.

The loadings matrix $\mathbf{P}$ can be partitioned in the same way as equation 4.2 and it can also be partitioned by separating the significant A principal components and leaving $H - A$ components, where $H = rank(\mathbf{X})$:

$$\begin{aligned}\mathbf{P} &= \begin{bmatrix} \mathbf{P}^{*} \\ \mathbf{P}^{\#} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{1:A} & \mathbf{P}_{A+1:H} \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{P}^{*}_{1:A} & \mathbf{P}^{*}_{A+1:H} \\ \mathbf{P}^{\#}_{1:A} & \mathbf{P}^{\#}_{A+1:H} \end{bmatrix}\end{aligned} \tag{4.3}$$

This partition between the known and unknown data can also be applied to the covariance matrix:

$$\begin{aligned}\mathbf{S} &= \frac{1}{KJ-1}\mathbf{X}^{T}\mathbf{X} = \\ &= \frac{1}{KJ-1}\begin{bmatrix} \mathbf{X}^{\#T}\mathbf{X}^{\#} & \mathbf{X}^{\#T}\mathbf{X}^{*} \\ \mathbf{X}^{*T}\mathbf{X}^{\#} & \mathbf{X}^{*T}\mathbf{X}^{*} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^{\#\#} & \mathbf{S}^{\#*} \\ \mathbf{S}^{*\#} & \mathbf{S}^{**} \end{bmatrix}\end{aligned} \tag{4.4}$$

Once the notation is presented the imputation methods can be described:

Trimmed score method (TRI)

This method proposes to substitute every missing value by its mean. The statistics are calculated using centred data, so, the mean value is zero. This method substitutes $\mathbf{z}^{\#} = 0$. The scores can be estimated as:

$$\begin{aligned}\hat{\boldsymbol{\tau}}_{1:A} &= \mathbf{P}^{T}_{1:A}\mathbf{z} = \begin{bmatrix} \mathbf{P}^{*T}_{1:A} & \mathbf{P}^{\#T}_{1:A} \end{bmatrix} \begin{bmatrix} \mathbf{z}^{*} \\ \mathbf{z}^{\#} \end{bmatrix} = \\ &= \mathbf{P}^{*T}_{1:A}\mathbf{z}^{*} + \mathbf{P}^{\#T}_{1:A}\mathbf{z}^{\#} = \mathbf{P}^{*T}_{1:A}\mathbf{z}^{*}\end{aligned} \tag{4.5}$$

The covariance matrix for the score vector estimation error can be calculated as:

$$\begin{aligned} Var(\boldsymbol{\tau}_{1:A} - \hat{\boldsymbol{\tau}}_{1:A}) &= \mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}\boldsymbol{\Lambda}\mathbf{P}^{*T}\mathbf{P}^{*}_{1:A} + \boldsymbol{\Lambda}_{1:A} \\ &\quad -\mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}_{1:A}\boldsymbol{\Lambda}_{1:A} - \boldsymbol{\Lambda}_{1:A}\mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}_{1:A} \end{aligned} \tag{4.6}$$

Known data regression method (KDR)

When the UPCA model has been fitted for a transient state or a start-up, the A significant scores matrix can be expressed as:

$$\mathbf{T}_{1:A} = \mathbf{X}\mathbf{P}_{1:A} = \mathbf{X}^{*}\mathbf{P}^{*}_{1:A} + \mathbf{X}^{\#}\mathbf{P}^{\#}_{1:A} \tag{4.7}$$

where the term $\mathbf{X}^{\#}$ during the *on-line* monitoring is unknown.

This method proposes to estimate the scores of an incomplete observation using the training data set. In particular, the columns of the training data set corresponding to the known variables ($\mathbf{X}^{*}$) in the current observation:

$$\mathbf{T}_{1:A} = \mathbf{X}^{*}\mathbf{B_1} + \mathbf{U_1} \tag{4.8}$$

in this equation, the least squares estimator of $\mathbf{B}$ is:

$$\hat{\mathbf{B}} = (\mathbf{X}^{*T}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\mathbf{T}_{1:A} \tag{4.9}$$

As Arteaga and Ferrer (2002) prove, the significant scores of the incomplete current observation can be calculated as:

$$\hat{\boldsymbol{\tau}}_{1:A} = \boldsymbol{\Lambda}_{1:A}\mathbf{P}^{*T}_{1:A}(\mathbf{P}^{*}\boldsymbol{\Lambda}\mathbf{P}^{*T})^{-1}\mathbf{z}^{*} \tag{4.10}$$

The KDR methods pose a problem due to the fact that the matrix $\mathbf{P}^{*}\boldsymbol{\Lambda}\mathbf{P}^{*T}$ can be a very ill-conditioned matrix and its inverse matrix may not be calculated. To deal with this drawback, Arteaga and Ferrer (2005) propose a framework, where different regression-based estimation methods are presented. Three alternatives can be considered:

KDR method with PCR: In this case, the unknown future data is estimated using the known data too, but using the principal component regression (PCR) approach instead of the least squared regression model used in Eq. 4.8. The regression model can be written as:

$$\mathbf{X}^{\#} = (\mathbf{X}^{*}\mathbf{V}_{1:\rho})\mathbf{B}_2 + \mathbf{U}_2 \tag{4.11}$$

where $\mathbf{V}_{1:\rho}$ is the $(KJ - M) \times \rho$ matrix whose columns are the eigenvectors of the $\mathbf{S}^{**}$ matrix associated with the greatest eigenvalues η and $\rho \leq \eta = rank(\mathbf{S}^{**})$. Then the scores can be estimated using the following expression:

$$\hat{\boldsymbol{\tau}}_{1:A} = \mathbf{P}^{\#T}_{1:A}\mathbf{S}^{\#*}\mathbf{V}_{1:\rho}(\mathbf{V}^{T}_{1:\rho}\mathbf{S}^{**}\mathbf{V}_{1:\rho})^{-1}\mathbf{V}^{T}_{1:\rho}\mathbf{z}^{*} + \mathbf{P}^{*T}_{1:A}\mathbf{z}^{*} \tag{4.12}$$

The calculation of the covariance matrix for the score vector estimation is performed using the equation:

$$\begin{gathered} Var(\boldsymbol{\tau}_{1:A} - \hat{\boldsymbol{\tau}}_{1:A}) = \\ \mathbf{P}^{\#T}_{1:A}\left[\mathbf{S}^{\#\#} - \mathbf{S}^{\#*}\mathbf{V}_{1:\rho}\left(\mathbf{V}^{T}_{1:\rho}\mathbf{S}^{**}\mathbf{V}_{1:\rho}\right)^{-1}\mathbf{V}^{T}_{1:\rho}\mathbf{S}^{*\#}\right]\mathbf{P}^{\#}_{1:A} \end{gathered} \tag{4.13}$$

KDR method with pseudoinverse: Another alternative to impute the value of the future unknown data using the known past data is expressed in the next expression:

$$\mathbf{X}^{\#} = (\mathbf{X}^{*}\mathbf{V}_{1:\eta})\mathbf{B}_3 + \mathbf{U}_3 \tag{4.14}$$

In this case all the eigenvectors associated with the positive eigenvalues of $\mathbf{S}^{**}$ are taken into account. The scores can be estimated as:

$$\hat{\boldsymbol{\tau}}_{1:A} = \mathbf{P}^{\#T}_{1:A}\mathbf{S}^{\#*}\mathbf{V}_{1:\eta}(\mathbf{V}^{T}_{1:\eta}\mathbf{S}^{**}\mathbf{V}_{1:\eta})^{-1}\mathbf{V}^{T}_{1:\eta}\mathbf{z}^{*} + \mathbf{P}^{*T}_{1:A}\mathbf{z}^{*} \tag{4.15}$$

In the KDR method with pseudoinverse, the covariance matrix for the score vector estimation is calculated:

$$Var(\boldsymbol{\tau}_{1:A} - \hat{\boldsymbol{\tau}}_{1:A}) = \tag{4.16}$$
$$\mathbf{P}_{1:A}^{\#T} \left[\mathbf{S}^{\#\#} - \mathbf{S}^{\#*} \mathbf{V}_{1:\eta} \left(\mathbf{V}_{1:\eta}^{T} \mathbf{S}^{**} \mathbf{V}_{1:\eta} \right)^{-1} \mathbf{V}_{1:\eta}^{T} \mathbf{S}^{*\#} \right] \mathbf{P}_{1:A}^{\#}$$

KDR method with PLS: If the regression method used to estimate the unknown future data during the transition is partial least squares (PLS), the regression model can be expressed as:

$$\mathbf{X}^{\#} = (\mathbf{X}^{*}\mathbf{W}^{*})\mathbf{B}_4 + \mathbf{U}_4 \tag{4.17}$$

where $\mathbf{W}^{*}$ is the loading matrix that allows the PLS scores to be calculated as $\mathbf{T}_{PLS} = \mathbf{X}^{*}\mathbf{W}^{*}$ in the estimation model of $\mathbf{X}^{\#}$ from $\mathbf{X}^{*}$.

The scores can be estimated in this case as follows:

$$\hat{\boldsymbol{\tau}}_{1:A} = \mathbf{P}_{1:A}^{\#T} \mathbf{S}^{\#*} \mathbf{W}^{*} (\mathbf{W}^{*T} \mathbf{S}^{**} \mathbf{W}^{*})^{-1} \mathbf{W}^{*T} \mathbf{z}^{*} + \mathbf{P}_{1:A}^{*T} \mathbf{z}^{*} \tag{4.18}$$

In this case, the covariance matrix for the score vector estimation error is computed as:

$$Var(\boldsymbol{\tau}_{1:A} - \hat{\boldsymbol{\tau}}_{1:A}) = \tag{4.19}$$
$$\mathbf{P}_{1:A}^{\#T} \left[\mathbf{S}^{\#\#} - \mathbf{S}^{\#*} \mathbf{W}^{*} \left(\mathbf{W}^{*T} \mathbf{S}^{**} \mathbf{W}^{*} \right)^{-1} \mathbf{W}^{*T} \mathbf{S}^{*\#} \right] \mathbf{P}_{1:A}^{\#}$$

In the three alternatives proposed to deal with the ill-conditioned problems that can appear using the KDR methods, the calculation of the specific matrices $\mathbf{V}_{1:\rho}$, $\mathbf{V}_{1:\eta}$ and $\mathbf{W}^{*}$ requires the singular value decomposition (SVD) or similar approaches in each sampling time. Due to the size, too much time can be spent and the monitoring may not be developed in fast processes. So, these matrices can be calculated and stored *off-line* for every possible size of the matrix $\mathbf{S}^{**}$ and the *on-line* tasks can be executed quickly.

Trimmed score regression method (TSR)

The trimmed score regression method estimates the scores of a new observation using the expression $\mathbf{T}^{*}_{1:A} = \mathbf{X}^{*}\mathbf{P}^{*}_{1:A}$. This method reconstructs $\mathbf{T}_{1:A}$ from the trimmed scores using the regression model $\mathbf{T}_{1:A} = \mathbf{T}^{*}_{1:A}\mathbf{B} + \mathbf{U}$. In this method, the scores can be estimated as follows (Arteaga and Ferrer (2002)):

$$\hat{\boldsymbol{\tau}}_{1:A} = \boldsymbol{\Lambda}_{1:A}\mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}_{1:A}\left(\mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}\boldsymbol{\Lambda}\mathbf{P}^{*T}\mathbf{P}^{*}_{1:A}\right)^{-1}\mathbf{P}^{*T}_{1:A}\mathbf{z}^{*} \tag{4.20}$$

The calculation of the estimated score in this method presents a computational cost higher than the cost using the TRI method. But, in both examples considered in this work, no problems appear during the *on-line* monitoring related with the estimation time and no *off-line* calculation is necessary.

In this case, the covariance matrix for the score vector estimation error is computed as:

$$\begin{gathered} Var(\boldsymbol{\tau}_{1:A} - \hat{\boldsymbol{\tau}}_{1:A}) = \\ \left[\mathbf{I}_{A} - \boldsymbol{\Lambda}_{1:A}\mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}_{1:A}\left(\mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}\boldsymbol{\Lambda}\mathbf{P}^{*T}\mathbf{P}^{*}_{1:A}\right)^{-1}\mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}_{1:A}\right]\boldsymbol{\Lambda}_{1:A} \end{gathered} \tag{4.21}$$

4.3.4 Control limits

As stated before, to develop an MSPC scheme, it is necessary to collect several past transient states or start-ups. All these data must be aligned to the same duration. The three-way matrix $\underline{\mathbf{X}}$ can be built using the aligned data. Finally, the three-way matrix must be unfolded into a two-way matrix. A UPCA model can be established using this unfolded matrix $\mathbf{X}$.

Classical principal component analysis can be performed over this new matrix. As mentioned in subsection 2.5.3, one has to decide how many components to choose, while a new loading matrix ($\mathbf{P}$) and scores matrix ($\mathbf{T}$) will be computed.

In this case, the T^2 monitoring statistic and its upper limit, which are necessary in order to build the control charts, can be calculated using equations 2.13 and 2.15. It is important to take into account that, in this case, the number of individuals in the unfolded matrix $\mathbf{X}$ is the number of past transitions I instead of the number of process samples N.

In the case of the Q statistic, as Nomikos and MacGregor (1995) propose, it is a better option to calculate the squared prediction error Q at every particular instant k:

$$Q_k = \sum_{c=1}^{J} \mathbf{e}(c)^2 \tag{4.22}$$

instead of the squared residuals over all time periods Q, as this measurement does not represent the instantaneous perpendicular distance to the reduced space. Here $\mathbf{e}$ is calculated for the current measures vector $\mathbf{z}^T_{new,t}(1 \times J)$ (scaled) at instant k as:

$$\mathbf{e} = \mathbf{z}^T_{new,t} - \hat{\mathbf{z}}^T((k-1)J + 1 : kJ) \tag{4.23}$$

with

$$\hat{\mathbf{z}} = \hat{\boldsymbol{\tau}}_{1:A}\mathbf{P}^T_{1:A}$$

The upper limit for these statistics can be calculated by approximating the value at every instant to a chi squared distribution, as:

$$Q_{\alpha k} = g_k \chi^2_{h_k,\alpha} \tag{4.24}$$

where $\chi^2_{h_k,\alpha}$ is the limit value of the chi squared variable with h_k degrees of freedom and α level of significance. g_k and h_k can be approximated in different ways as Nomikos and MacGregor (1995) discuss and Lennox et al. (2001) compare. In this work, these parameters are calculated using the variance (v_k) and the mean (m_k) of the Q_k sample over the reference nominal data set used to compute the UPCA model, by means of the following expressions:

$$g_k = \frac{v_k}{2m_k}; h_k = \frac{2m_k^2}{v_k} \tag{4.25}$$

This upper limit at each instant k can be composed using its immediately previous and following time observations $k-2, k-1, k, k+$

$1, k+2$. This means using a moving window for the estimations of the control limit for the instant in the center of the window.

In processes with several operation modes and their corresponding transient states and start-ups, the upper limit will change for every one of these states and, in the case of the Q statistic, the limits change in every sample. To simplify the control charts, the upper limit of both statistics can be unified to the same value. The upper limits can be normalized by setting them to unitary value, $T^2_{(n)\alpha} = 1$ and $Q_{(n)\alpha} = 1$ and computing the normalized statistics for every operating mode, transient state and start-up i as:

$$\begin{aligned} T^2_{(n)i} &= \frac{T^2_i}{T^2_{i\ \alpha}} \\ Q_{(n)i} &= \frac{Q_i}{Q_{i\ \alpha}} \end{aligned} \tag{4.26}$$

Finally, to perform a comparative analysis from the point of view of the best imputation method, the control limits were estimated from theoretical results and then those limits were tuned using a leave-one-out approach (Ramaker et al., 2006; Camacho et al., 2009) for an imposed significance level (ISL) of 5%. This value is the expected percentage of alarms for a transient state under normal operation conditions (NOC). The Overall Type I (OTI) risk is computed to perform the proper tuning of the control limits. For a coherent monitoring system, the real OTI percentage of alarms under NOC should be close to the ISL expected percentage of alarms under NOC.

$$\text{OTI} = 100 \times \frac{nf}{I_{NOC}K}\% \tag{4.27}$$

where nf is the total number of faults and I_{NOC} is the number of NOC transitions considered.

The detection accuracy of the faults is evaluated using the Overall Type II (OTII) risk:

$$\text{OTII} = 100 \times \frac{nnf}{I_F l}\% \tag{4.28}$$

where nnf is the number of non-signaled faults, I_F is the number of faulty transitions considered and l is the length of the faulty interval.

Both parameters can be calculated using the database of NOC and faulty past transient states. Additionally, the Type I (TI) risk measures the percentage of transient states under NOC detected as faulty transient states (three consecutive samples greater than the upper limits in any of the two charts was the condition for a transient to be considered as abnormal) and the Type II (TII) risk measures the percentage of faulty transient states detected as NOC transient states. The different approaches used for imputation can be compared when the TI and OTI present similar values. So this study is performed in this chapter.

4.3.5 Contribution plots

As mentioned in chapter 2, Q statistic charts should be checked first (Ferrer, 2007). The contribution of each variable to the Q statistic applied to transient states can be calculated in the same way as with classical PCA. When the Q statistic raises the upper limit, the errors in this instant k could be plotted in a bar plot and the operator can see the variables involved in the detected fault at this precise time instant.

If the Q chart does not signal but Hotelling's T^2 does, then the contribution plots of the normalized scores, variable contributions to individual score plots and the overall variable contributions plots, should be plotted to identify the variables related with the fault (Kourti and MacGregor, 1996). In the last two plots, the variable contributions to the greatest scores from the beginning of the transition to the current instant can be grouped by variable in different bar plots and the operator can see the evolution of the variables and detect which of them are related with the fault. Figure 4.7 shows a scheme of the preforming of this type of bar plots for UPCA. An example of this bar plot can be seen in figure 5.14.

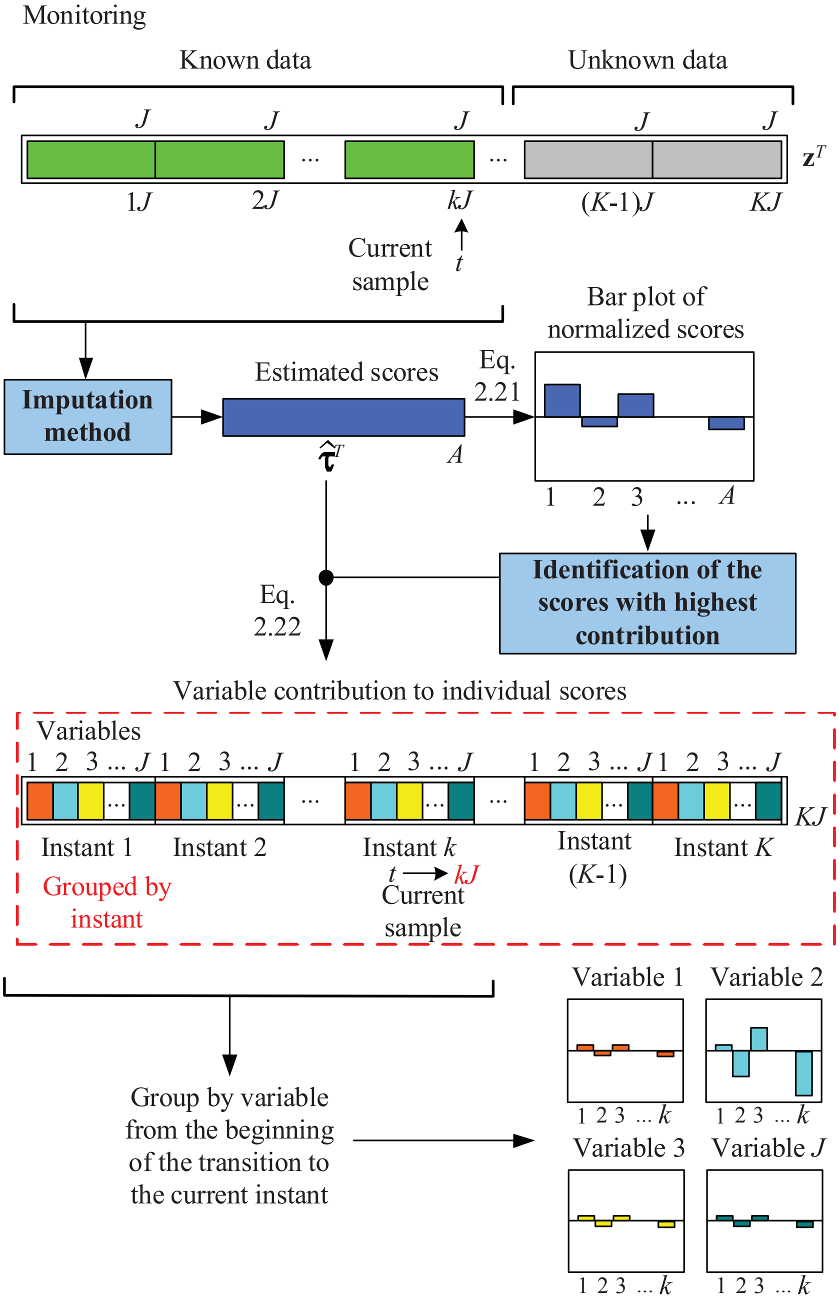


Figure 4.7: Variable contributions plot to individual scores for UPCA

4.4 UPCA and DPCA

The DPCA (subsection 3.4.2) is related with the UPCA method. The DPCA method proposes to establish the PCA model considering the dynamic of the process. So, the data matrix **X** considers the current and past values of the process variables at every sample time (individuals). The number of variables (columns) is increased due to the fact that the past values of the process variables are considered:

$$\mathbf{X} = [\mathbf{X}_t | \mathbf{X}_{t-1} | \ldots | \mathbf{X}_{t-h}] \tag{4.29}$$

where the operator | is the matrix concatenation operator. $\mathbf{X}_t$ is the data matrix **X** at the time instant t and $\mathbf{X}_{t-h}$ at the time instant $t-h$. Basically, the data matrix with delay is the original data matrix with a row displacement of h time samples. The top part of figure 4.8 shows an example of the establishment of the data matrix **X** considering 3 delays.

In this configuration, the PCA model is able to capture the relationships between the variables at the current sample and the relationships between the variables at the current sample and past samples.

If all past variables along the batch or transient state are considered using DPCA, as figure 4.8 shows in the bottom part, all the dynamics along the transient state can be captured. However, the data matrix obtained only have one individual, and the PCA procedure cannot be performed. The UPCA approach can be seen as a DPCA model considering several normal transient states in order to extract the normal operating conditions at every sample time considered.

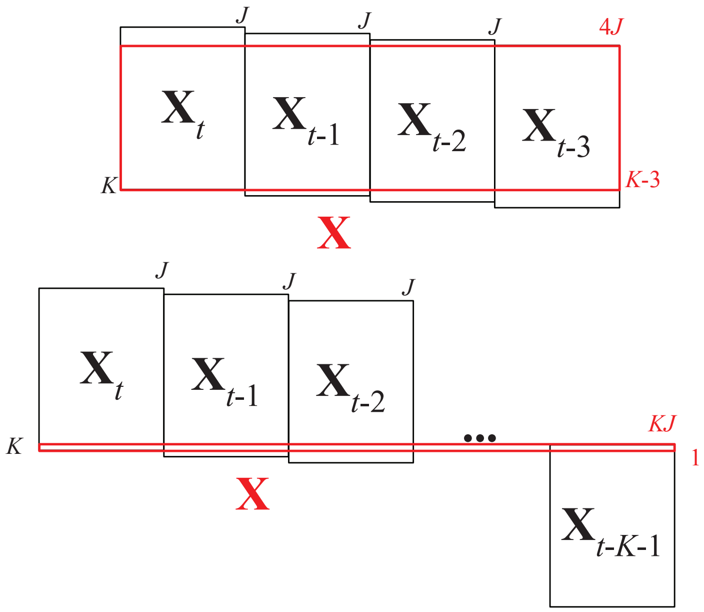


Figure 4.8: Data matrix structure **X** for DPCA

4.5 Examples of transient state monitoring

4.5.1 Case study: simulated evaporation section of a sugar factory

The first example presented in this chapter is a simulated factory, to be precise, the evaporation section of a sugar factory. This section belongs to a distributed simulator of a whole sugar factory (Alves et al., 2004; Alves, 2005; Alves et al., 2008).

In the evaporation stage, the sugar content of a juice rich in saccharose is increased by boiling. The syrup obtained in this phase is used to obtain sugar crystals. The evaporation section of this example is formed by five interconnected effects. Each effect is formed by one or several evaporation units. The steam generated in one effect is used to provide heat to the evaporators of the next effect. The sugar concentration in the juice is increased from one effect to another. More information about the evaporation stage can be found in Merino et al. (2005) and Merino (2008).

Moreover, the simulator used in this work was designed to introduce several types of faults. The evaporation section consists of 2,546 equations and 3,699 variables because the simulation model is a first principles model, so the faulty behaviour can be simulated perfectly. An overview of the process is shown in figure 4.9.

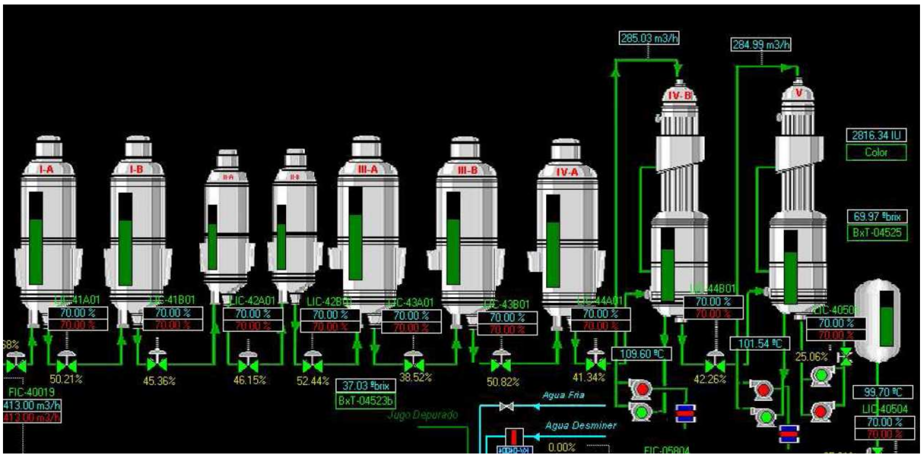


Figure 4.9: Operator interface of the evaporation section of the sugar factory.

The faults are introduced as an abrupt step in the equations that perform the physical or chemical principle, so their behaviour is not abrupt in all the cases due to the different nature of the principle modified. Four typical faults were induced into the plant with sizes of 20%, 40% and 60% for each fault:

- Fault 1 (F1). Decay of the performance in one of the evaporators.
- Fault 2 (F2). Blockage in a valve.
- Fault 3 (F3). Accumulation of non condensing materials in one of the evaporators.

- Fault 4 (F4). Sensor offset.

It is very usual to halve the sugar production in this type of beet sugar production factories when the weather is rainy. This is because the beets are not collected. The evaporation section is a critical part of this type of process and its behaviour should be monitored, including the transient states that appear when the production is halved. The reduction in the production is carried out by reducing the input flow of juice and reducing the steam flow in the evaporators to avoid burning the syrup produced.

The variables collected to perform the UPCA model are 36 signals of the typical sensors, controllers and actuators, such as flow rates, pressures, Brix and levels. The data set of 19 nominal transitions was aligned using the indicator variable approach. In the case of the faulty situations, 5 simulations were running per fault.

In this case, the method used to align the variables was the indicator variable. The values of the measured variables were stored when the process variables rose by every 2.5% increase with respect to the reference step. The control variable selected in this case was the input flow of juice because it has a slow response. The steam flow control loop has a very quick response, so it is not a good representation of this transient state. The number of samples stored along the transition was 40. $\mathbf{V}_{1:\rho}$, $\mathbf{V}_{1:\eta}$ and $\mathbf{W}^*$ were calculated *off-line* for the 40 samples. The PCA model was established with 9 components selected using a cross-validation approach. The length of the transient estates takes values between 82 and 95 samples.

The indicator variable approach can be applied in this example because the control loop selected has a first order response, so the indicator variable is a monotonically decreasing variable. Also, this control loop represents the dynamic of the whole transient state and when this control loop raises the steady state, it is raised in the whole plant too.

The prediction error sum of variances (PRESV) can be calculated as a suitable parameter to select the most correct imputation method. This parameter can be calculated as the trace of covariance matrix for every imputation method (Eqs. 4.6, 4.13, 4.16, 4.19 and 4.21). It is equivalent to calculating the mean square error (MSE) (Arteaga and Ferrer, 2005).

As mentioned before, when the transient state is monitored, the future samples $(K - k)$ at time k $(k < K)$ are unknown. So an imputation method must be selected to predict this unknown data and calculate the monitoring statistics. Figure 4.10 shows the PRESV values at different time points for the different imputation methods considered in this work. Nine different time points along the transient state were considered ($k_1 = 144$, $k_2 = 288,\ldots$, $k_9 = 1296$). As the figure displays, the estimation error decays along the transient state due to the decrease in the unknown data.

Figure 4.10 shows that the methods with the lowest PRESV are KDR methods using PCR or PLS when the maximum number of components are considered (the KDR method using the pseudoinverse presents similar results to the KDR with PCR and PLS when all components are considered). The PRESV values calculated for these two methods are drawn in the figure with the name of the method and in magenta and black, respectively. The number of components considered appears after the name of the method. If the number of components considered is decreased, the PRESV value increases, as the graphic shows. The TSR method does not present such good results as the KDR when all the components are considered, but, as cited before, this method does not require any previous *off-line* calculation and so the score imputation is quick. The evolution of the PRESV values for this imputation method appears in red in the graphic. The PRESV for the TSR method presents acceptable results, especially, if these results are compared with the TRI imputation method, which presents the worst results. The PRESV evolution appears in blue in figure 4.10 for the TRI method.

Tables 4.2 and 4.3 present the results of TI and TII. The different imputation methods present similar TI and OTI values since the theoretical control limits are re-adjusted using a leave-one-out approach. The TI and OTI values are similar for all the imputation methods and the OTI are close to 5% (the ISL of the limits). The OTII results obtained are quite similar for all the imputation methods explained for the first three faults considered. In fault number four, the biggest

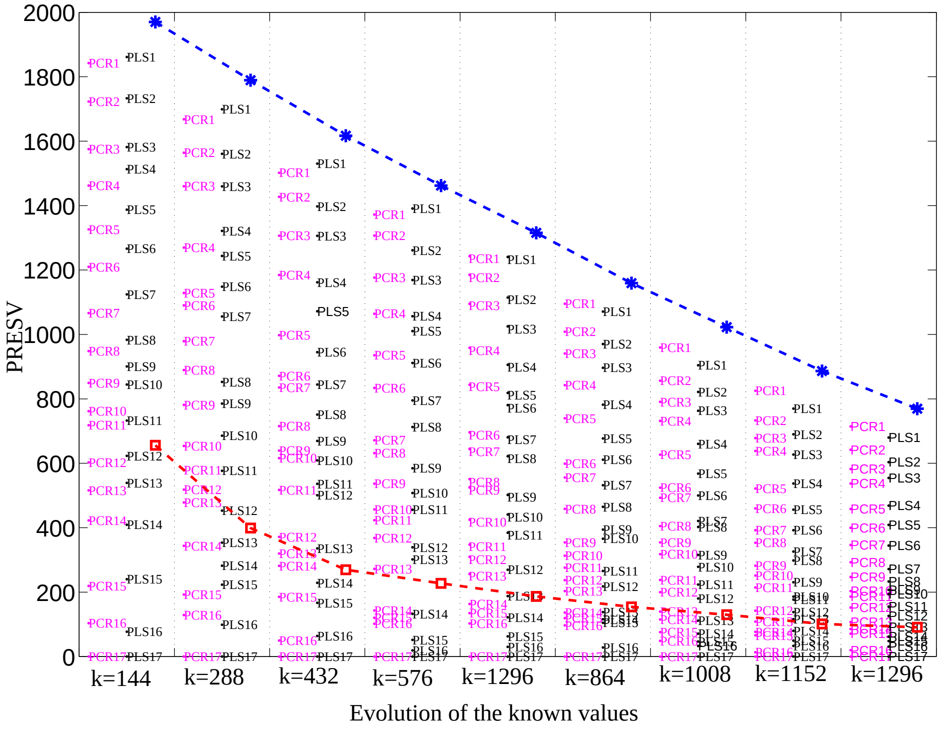


Figure 4.10: PRESV for the different imputation methods considered at different time points k. PCRx means KDR with PCR with x components. PLSx means KDR with PLS with x components. The TSR method results are plotted with red squares and a dashed line and the TRI method results are plotted with blue asterisks and a dashed line.

percentage is obtained by the TSR method.

The first fault considered (F1) in this example consists of a decay of the performance in the first evaporator of the third effect. Sugar layers are accumulated on pipes inside the evaporator and the heat transmission decreases. This fault was detected by the Q statistic with all the fault sizes. Figure 4.11 shows the case of a 20% fault, Q detects the fault clearly. The results related with the fault detection delay are very similar for all the imputation methods considered. Table 4.4 shows the mean (m) and standard deviation (std) of the fault detection

	Test-NOC		
Imputation method	**TI**	**OTI$_{T^2}$**	**OTI$_Q$**
TRI	0% (0/19)	4.04%	5.75%
TSR	0% (0/19)	4.87%	4.21%
KDR PCR	0% (0/19)	4.29%	4.00%
KDR pseudoinverse	0% (0/19)	4.82%	4.26%
KDR PLS	0% (0/19)	4.04%	5.67%

Table 4.2: TI and OTI results calculated for the different imputation methods. Imposed level of significance equal to 5%.

	Test-F	**Test-F1**	**Test-F2**	**Test-F3**	**Test-F4**
Imputation method	**TII**	**OTII**	**OTII**	**OTII**	**OTII**
TRI	0% (0/42)	79.20%	77.10%	66.70%	21.80%
TSR	0% (0/42)	74.00%	75.32%	59.54%	41.76%
KDR PCR	0% (0/42)	77.78%	75.78%	60.28%	18.84%
KDR pseudoinverse	0% (0/42)	77.78%	75.78%	60.28%	29.79%
KDR PLS	0% (0/42)	79.31%	76.84%	61.08%	12.14%

Table 4.3: TII and OTII results calculated for the different imputation methods. Imposed level of significance equal to 5%.

delay for the different faults considered. In the case of the first fault, the standard deviation is bigger than the other cases because this fault is a parametric fault simulated by altering several parameters and all of them are influenced by noise.

Imputation	**F1**		**F2**		**F3**		**F4**	
method	*m*	*std*	*m*	*std*	*m*	*std*	*m*	*std*
TRI	$238,25$	$247,02$	$37,00$	$1,15$	$74,66$	$17,62$	$44,00$	$9,39$
TSR	$187,75$	$152,68$	$37,00$	$1,15$	$74,66$	$17,62$	$44,00$	$9,39$
KDR PCR	$211,25$	$195,75$	$37,00$	$1,15$	$74,66$	$17,62$	$44,00$	$9,39$
KDR pseudoinverse	$187,75$	$152,68$	$37,00$	$1,15$	$74,66$	$17,62$	$44,00$	$9,39$
KDR PLS	$238,25$	$247,02$	$37,00$	$1,15$	$74,66$	$17,62$	$44,00$	$9,39$

Table 4.4: Mean and standard deviation of the fault detection delay (seconds) in Q statistic using different imputation methods and fault sizes.

A blockage in the juice pipe valve that connects the third effect (F2) with the fourth effect is the second fault considered. The Q was able to detect this fault and it presents the same fault detection delay

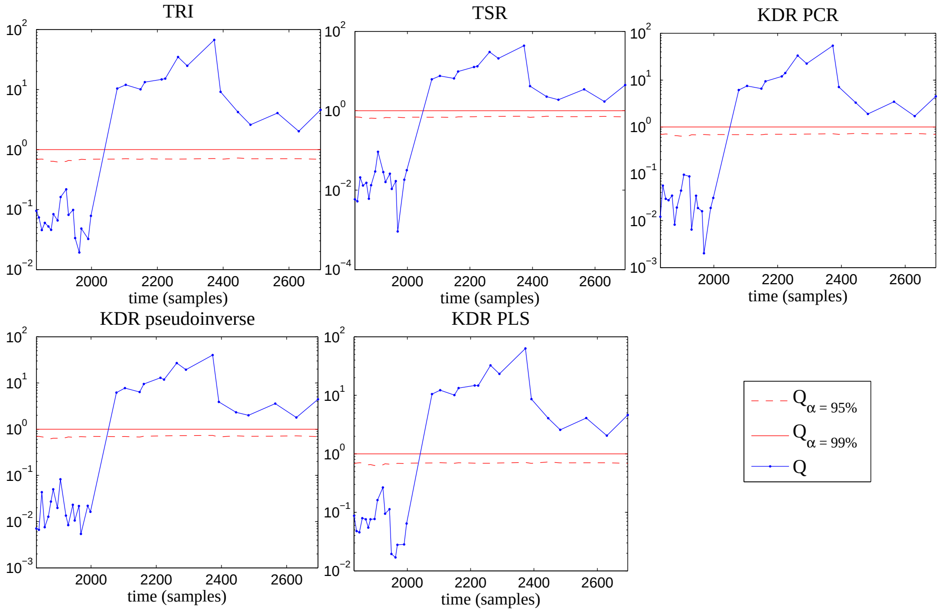


Figure 4.11: Monitoring of the transient state in the sugar factory with Q statistic and a 20% type 1 fault using different imputation methods. (Fault instant: 2000).

for this example with all the imputation methods. Figure 4.12 shows the case of a 20% fault, Q detects the fault clearly.

The third fault considered (F3) is due to the accumulation of incondensable gases in the heating systems of the evaporator of the fourth effect. The Q statistics were able to detect this fault in all the sizes except the smallest one. In all cases, the fault detection delay was similar independently of the imputation method used. Figure 4.13 shows the case of a 20% fault, Q detects the fault clearly.

The last of the faults introduced into the plant is a sensor offset (F4). The sensor chosen is a temperature sensor placed in the output steam pipe of the second evaporator in the second effect. In this type of fault, Q is able to detect the fault in all the cases. Figure 4.14 shows the case of a 20% fault, Q detects the fault clearly.

The Q statistic was more sensitive and faster in fault detection than T^2 in all the cases.

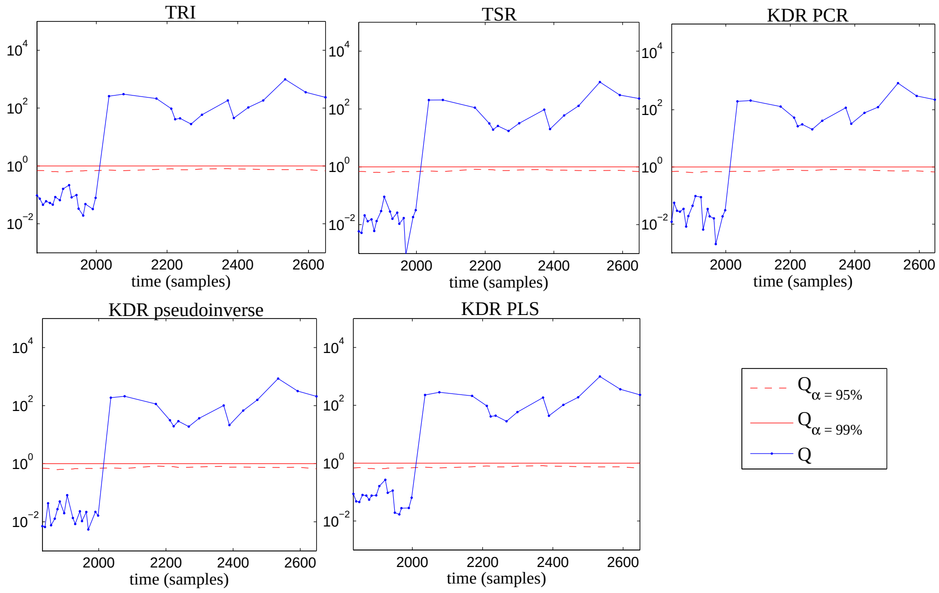


Figure 4.12: Monitoring of the transient state in the sugar factory with Q statistic and a 20% type 2 fault using different imputation methods. (Fault instant: 2000).

Figure 4.15 shows the plot of the normalized error of the variables from TSR at the instant that the fault was detected. A type 2 error was simulated in this case. One can see in the figure that the variable with the biggest bar is LIC-43B01_PV. This variable is the process variable of the control loop of the valve that was blocked. The figure also shows that the variables' residuals of the following effects begin to increase their values, mainly the process variables of the following valves control loops. In this graphic, only the variables with an unusually high value are drawn with their names.

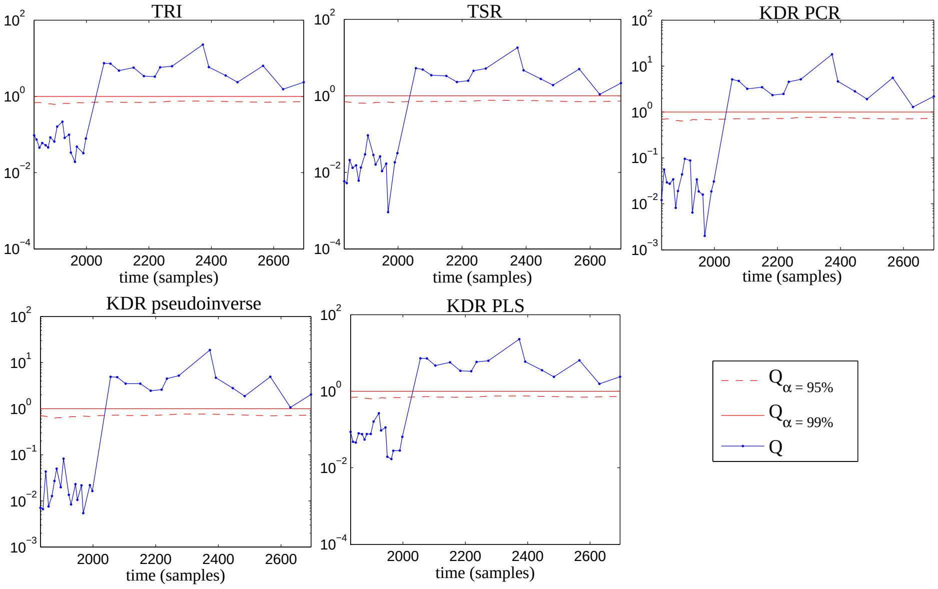


Figure 4.13: Monitoring of the transient state in the sugar factory with Q statistic and a 20% type 3 fault using different imputation methods. (Fault instant: 2000).

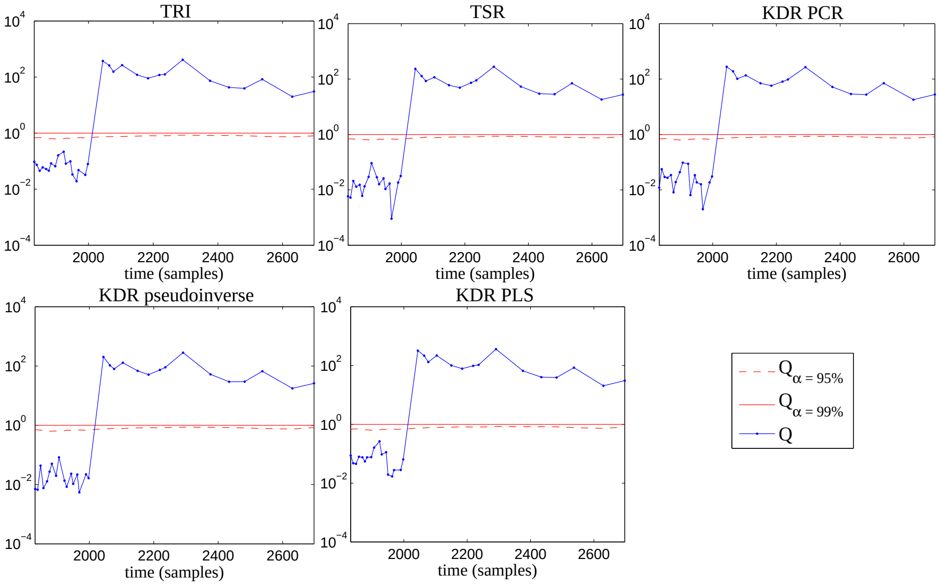


Figure 4.14: Monitoring of the transient state in the sugar factory with Q statistic and a 20% type 4 fault using different imputation methods. (Fault instant: 2000).

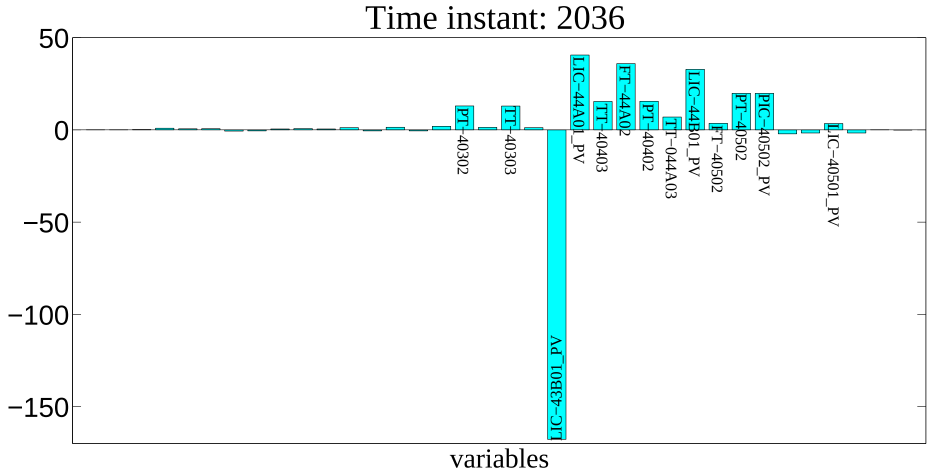


Figure 4.15: Contributions plot of the normalized error of the variables for a 20% type 2 fault along the transitions using TSR imputation method.

4.5.2 Case study: two communicated tanks

Another plant used to perform all the explained methods is a real laboratory plant. This plant is the two tank system used in chapter 3. But in this case, the real plant is used instead of the simulation model. The description of this plant can be found in section 3.3.

Two operation modes have been established in the plant. Operation Mode 1 (OM1), where the levels of tanks T_1 and T_2 are set at 30%; and Operation Mode 2 (OM2), where the level of tank T_1 is set at 30% and that of tank T_2 at 60%.

Nevertheless, two transient states have been identified. One of these is the transient state between the initial point, when the two tanks are empty, and the operation mode (OM1). This is the start-up SU1. The other transient state is the transition between operation modes OM1 and OM2, i.e., when the level reference of the tank T_2 changes its level from 30% to 60% at time t=400s. This is the grade transition GT1.

A PCA model has been developed for each operation mode OM1 and OM2 and for each of the transient states SU1 and GT1. The sampling time used in this example was one second. The PCA models for steady states were made using data samples acquired over 15 minutes. In the case of transient states, 29 normal transitions were stored to perform the UPCA method, the data set of 29 nominal transitions was aligned using the DTW approach. In the case of the faulty situations, 5 experiments was running per fault. In this case, the size of the matrices is not very large and the *on-line* monitoring tasks can be performed in the sampling time of the plant without any delay related with the *on-line* alignment and the scores imputation. The PCA model for the transient states was performed using 9 latent variables, in both cases selected using a cross-validation approach. The length of the transient states takes values between 201 and 254 samples.

Two faults were considered in the plant. Both faults are offsets in the level sensors. F1 is the fault for the sensor LT_1 in the tank T_1 and F2 is the fault for the sensor LT_2 in the tank T_2. The offset is a percentage value of the signal. All faults considered are abrupt.

In the same way as in the previous section, the PRESV were used to compare the performance of the different imputation methods. Figures 4.16 and 4.17 display the evolution of this parameter for the transient states SU1 and GT1 respectively. In this case, five points along the transition were taken into account. The TSR method in this example does not obtain such good results as in the previous example. Maybe KDR with PLS or PCR with 20 or 21 components could be used if the results obtained by TSR method are not considered acceptable by the plant operators.

The control limits for both statistics were calculated theoretically and then adjusted using the leave-one-out approach for an imposed significance level (ISL) in order to compare the results obtained by the different imputation methods considered in this work.

Tables 4.5 and 4.6 present the results of TI and TII. The results obtained in this case study are quite similar to the results obtained and explained in the previous section.

	Test-NOC			
Imputation method	**TI**		**OTI$_{T^2}$**	**OTI$_Q$**
TRI	0%	(0/29)	5.21%	5.97%
TSR	0%	(0/29)	5.78%	5.85%
KDR PCR	0%	(0/29)	5.42%	6.02%
KDR pseudoinverse	0%	(0/29)	5.28%	5.61%
KDR PLS	0%	(0/29)	5.08%	5.78%

Table 4.5: TI and OTI results calculated for the different imputation methods for the transient state SU1. Imposed level of significance equal to 5%.

Another characteristic related with the robustness in a fault detection tool is the delay in fault detection. Three sizes of offset were introduced in the sensor (20%, 40% and 60%) to test this parameter. Table 4.7 shows the mean (m) and standard deviation (std) of the delay values for the five test experiments measured in seconds, taking into account the fact that the sampling time is one second. The Q statistics detects the fault faster than the T^2 statistics in all cases. In this statistic, all methods present very similar results and, for this specific plant, the TSR method has the lowest mean delay, but, after performing an analysis of the variance (ANOVA), the detection delay

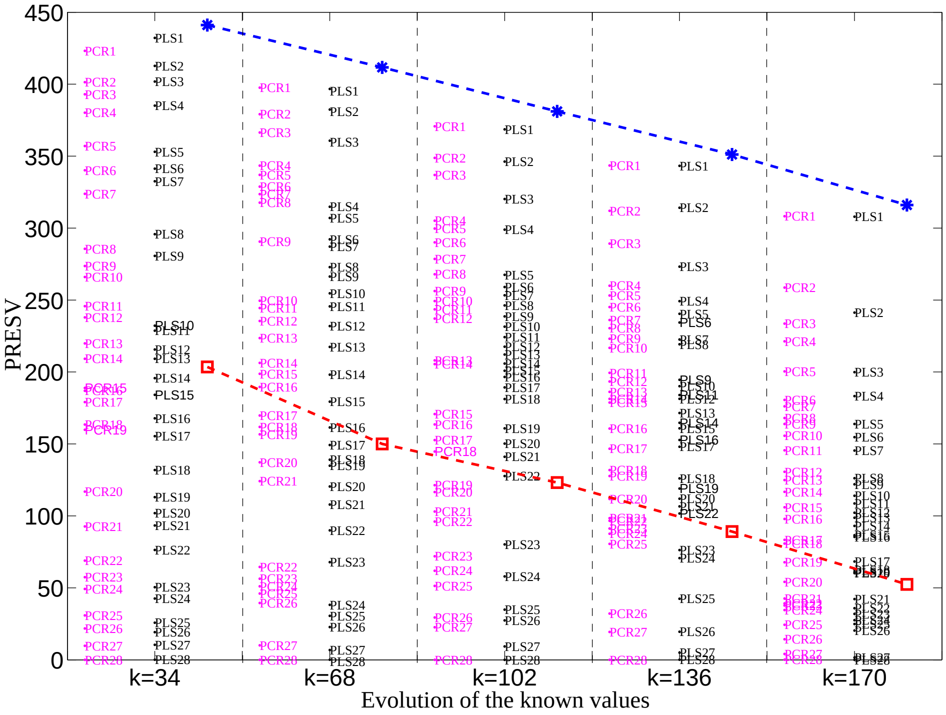


Figure 4.16: PRESV for the different imputation methods considered at different time points k for the transient state SU1. PCRx means KDR with PCR with x components. PLSx means KDR with PLS with x components. The TSR method results are plotted with red squares and a dashed line and the TRI method results are plotted with blue asterisks and a dashed line.

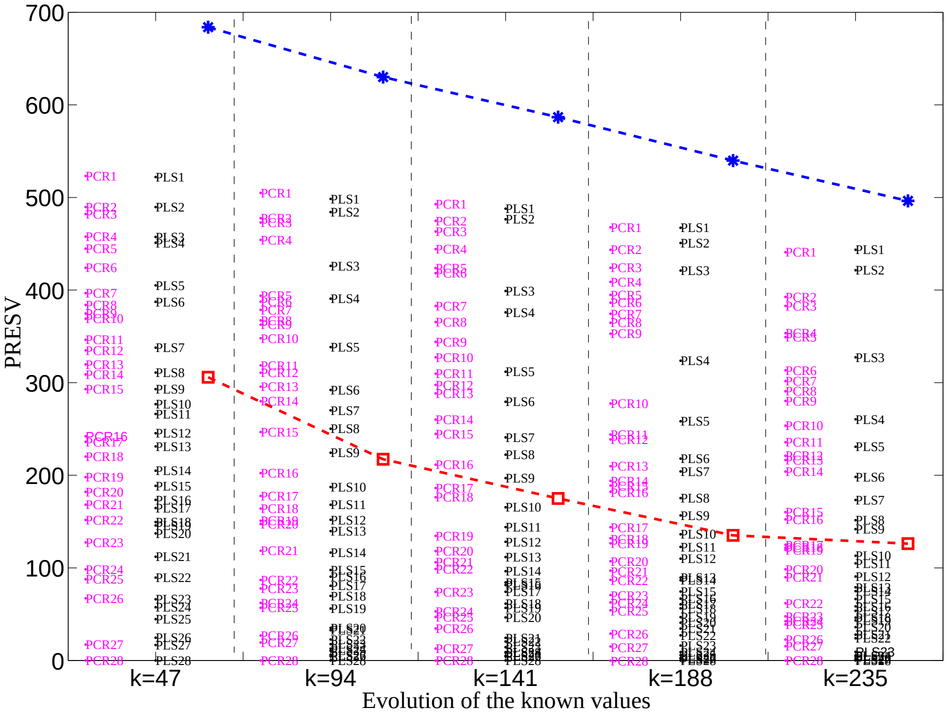


Figure 4.17: PRESV for the different imputation methods considered at different time points k and for the transient state GT1. PCRx means KDR with PCR with x components. PLSx means KDR with PLS with x components. The TSR method results are plotted with red squares and a dashed line and the TRI method results are plotted with blue asterisks and a dashed line.

Imputation method	Test-F TII	Test-F$_1$ OTII	Test-F$_2$ OTII
TRI	0% (0/9)	85.24%	83.18%
TSR	0% (0/9)	81.45%	80.28%
KDR PCR	0% (0/9)	88.58%	86.58%
KDR pseudoinverse	0% (0/9)	89.79%	87.56%
KDR PLS	0% (0/9)	85.83%	86.54%

Table 4.6: TII and OTII results calculated for the different imputation methods for the transient state SU1. Imposed level of significance equal to 5%.

of the different imputation methods is not statistically different (table 4.7). Figure 4.18 shows the monitoring of the start-up SU1 with a 20% offset fault.

Imputation	**F1**		**F2**	
method	*m*	*std*	*m*	*std*
TRI	4.94	1.50	4.54	1.48
TSR	3.93	0.97	3.99	0.91
KDR PCR	4.00	1.23	4.12	1.03
KDR pseudoinverse	4.00	1.45	4,00	1.54
KDR PLS	4.94	1.41	4.75	1.33
ANOVA data				
F-ratio		0.79		0.36
P-value		0.5431		0.8324

Table 4.7: Mean and standard deviation of fault detection delay using Q statistic for the transient state SU1. In both cases, after performing an analysis of variance, the *P-value* of the *F-ratio* is greater than 0.05, so, there does not exist a statistically significant difference between the delay of the five methods at the 95.0% confidence level (results obtained using STATGRAPHICS Centurion XVI ©).

Table 4.7 does not show the result for the T^2 statistical because the Q statistic detects the faults faster.

Similar results were obtained for the transient state GT1. Figure 4.19 shows the monitoring of this operation zone when a 40% fault is induced in the level sensor h_1.

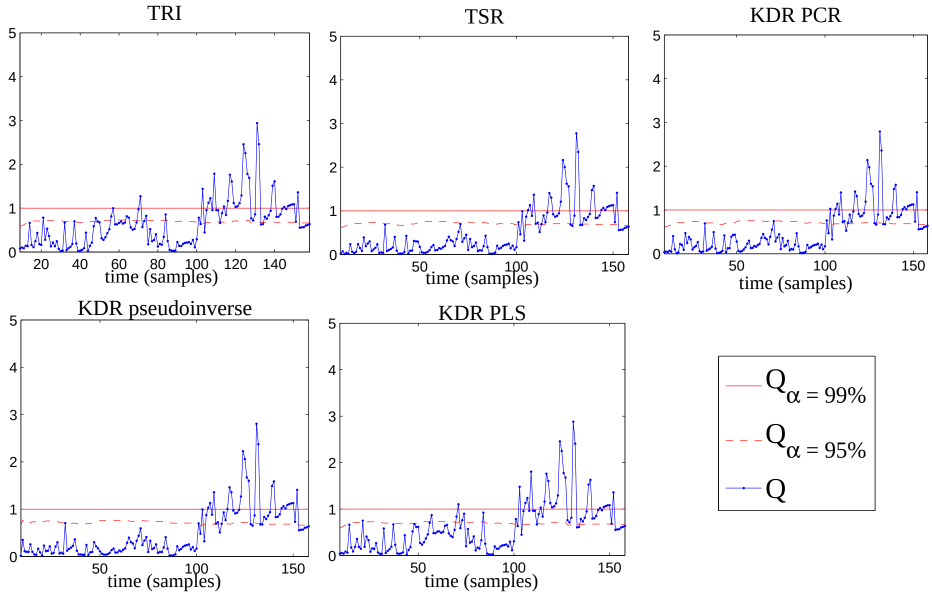


Figure 4.18: Monitoring of the transient state SU1 with Q statistic using different imputation methods with 20% offset in the sensor level h_1. (Fault instant: 100).

Finally, when the fault is detected, the contribution plots for the fault instant can be plotted and the operators, who know the behaviour of the plant, can diagnose the origin of the fault or the possible sources by looking at the variables with high values on these plots. The plant monitored in this example is very easy and, by looking at the contributions of the normalized errors of the variables, as figure 4.20 shows, one can detect that the cause of the fault is a specially high value in the sensor of level h_1 due to the 20% offset fault that was introduced in it.

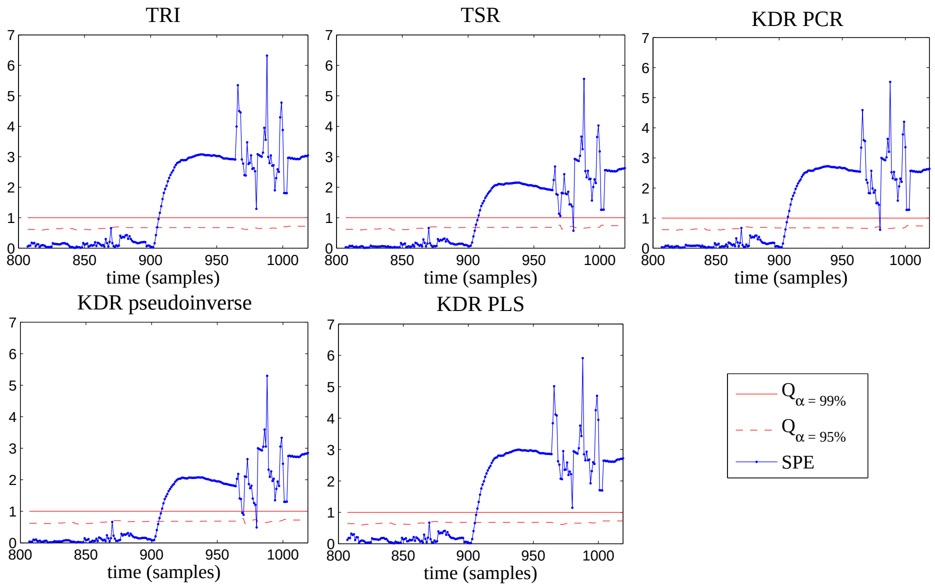


Figure 4.19: Monitoring of the transient state GT1 with Q statistic using different imputation methods with 40% offset in the sensor level h_1. (Fault instant: 900).

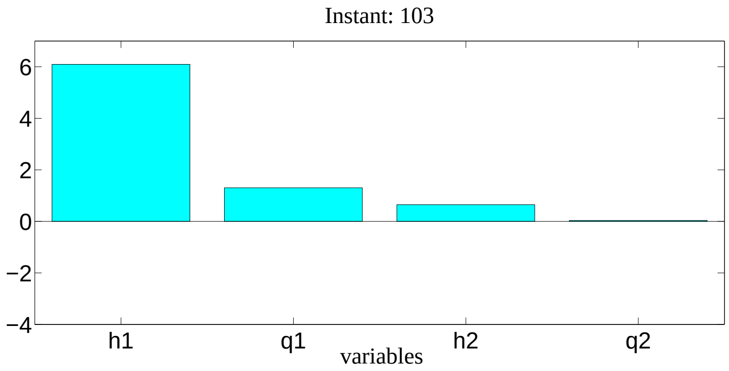


Figure 4.20: Contributions plot of the normalized error of the variables for a 20% offset fault along the start-up SU1 using the TSR imputation method.

4.6 Discussion

This chapter presents an approach to monitor start-ups and transient states in processes emphasizing fault detection and isolation. This approach is based on the adaptation of a PCA approach used in batch processes (UPCA). Start-ups and transitions between different steady states present a high non-linear behaviour and, if they have a long duration, cannot be monitored using classical linear PCA approaches because they can be detected as faults by the monitoring statistics.

The principal aspects of UPCA are presented and the necessary transformations to deal with transient phases are studied in depth. First of all, the data of this type of processes are arranged in three way matrices using past nominal transitions and the PCA model is performed over an unfolded two dimensional matrix. Secondly, different approaches existing in order to align the data of the different past nominal transitions, when their duration is not the same, are presented, implemented and discussed.

Also, the imputation of unknown future data problems that appear along the *on-line* monitoring are studied. In this case, the more common imputation methods are explained and compared in order to choose the best option in a fault detection and isolation scheme.

Finally, all the methods discussed are applied to two different examples. One of these examples consists of the evaporation section of a sugar factory with different faults, obtaining acceptable results in the fault detection and isolation tasks.

On the other hand, the second example studied is a real plant formed by two communicated tanks. In this case, two different faults are studied, also obtaining acceptable results.

Chapter 5

Monitoring and fault detection of continuous processes with non-strict steady behaviour

5.1 Introduction

As discussed in previous chapters, the PCA approach has been widely used in monitoring and fault detention tasks in continuous processes (Kourti and MacGregor, 1996; Ferrer, 2007). The PCA monitoring tools obtain good results when they are used to monitor steady states, since the relationship between the variables in the steady states are linear. However, the classical PCA fault detection schemes are not very suitable for processes that present non-linear behaviour, like batch processes or transient states, because an increment in the false alarms ratio can be observed, due to a change in the correlation structure due to the non-linear relationship between the variables.

In chapter 4, a configuration of the classical PCA approach to monitor normally non-linear processes is presented. These methods are called multi-way PCA (MPCA), unfolded PCA (UPCA) or Batch PCA (Nomikos and MacGregor, 1995; Nomikos, 1996; Aguado et al., 2007; Villez et al., 2009). All of them are equivalent.

This chapter describes how to apply a fault detection and isolation method based on the UPCA scheme in a continuous process. The designed approach is applied to a simulated reverse osmosis desalination plant. This model is based on small and medium real plants placed in remote areas. This remote location requires the use of monitoring tools, since the operator cannot be in the operation room all day. The use of modern technologies can allow the state of the plant to be monitored from a remote centralised operation centre.

The case study of this work is a continuous process, but due to the cleaning phases for the correct plant running, its behaviour is not precisely a steady state. So, the designed monitoring tool is based on the UPCA scheme.

5.2 Case study: simulated desalination plant

The fault detection approach presented in this chapter has been tested in a reverse osmosis desalination plant. This plant has been modelled as a first principles model using the modelling environment EcosimPro©. A complete description of the model can be found in Palacín et al. (2011). The plant has been developed as one of the results of the European project OPEN-GAIN.

The aim of the plant used in this work is to desalinate well water. The brackish water is pumped from the well to a supply tank. The water of the supply tank is pumped through a high pressure pump. The objective of this pump is to increase the pressure to above the osmotic pressure. The pressurized water goes through the reverse osmosis membrane rack. This difference in pressure between the membrane's sides creates a flow of clean water. The clean water is stored in another tank after a purification process. This last tank supplies water to the consumers.

The simulated plant used in this work corresponds to a real plant placed in a remote area in Tunisia, which is why a simulated plant is used. The plant is a prototype and is not running normally yet, so there are not enough data to perform this type of data-driven fault detection and isolation tasks. The aim of the simulated plant is to test different techniques as control or monitoring and fault detection methods, in order to facilitate its autonomous functioning and to reduce the human maintenance and operation due to its location. Another aim of the plant is to be fed by means of renewable energy. An overview of this plant can be seen in figure 5.1.

Another reason to use a simulated plant is because several types of faults can be included by manipulating the equations or parameters of the different physical components. For example, different types of breakage can be simulated in the membrane or different types of blockage in the different filters without taking any risk.

The plant is based on a reverse osmosis separation process. It is necessary to use high pressure to force the water through a semi-permeate membrane. The membrane retains the salt. Two different filters are placed before the membrane: first a sand filter and then a cartridge filter. These filters are required to remove several types of solid particles which can damage the membrane.

The decrease in the performance of membranes and filters during the plant operation is very common in this type of plants. This decrease is due to several types of deposits, such as scale, organic components, silt, etc. Cleaning cycles must be run to clean these deposits in order to obtain an optimal plant operation and avoid possible malfunctions.

The accumulation of deposits in the different filters and membranes and the required cleaning cycles are the reasons why the plant does not strictly run in a steady state. This is due to the noticeable differences in several of the pressure and concentration measurements when the plant has just been cleaned and when the plant was cleaned a long time ago. Figure 5.2 shows this phenomenon in one of the pressures measured in the sand filter output. This functioning mode makes the plant run in a not strictly steady state.

Eleven variables were considered in this work. All variables correspond to variables measured in the real plant in control loops or for supervision tasks. The measured variables set is formed by pressures, flows, total solid and salt concentrations. Table 5.1 shows the name

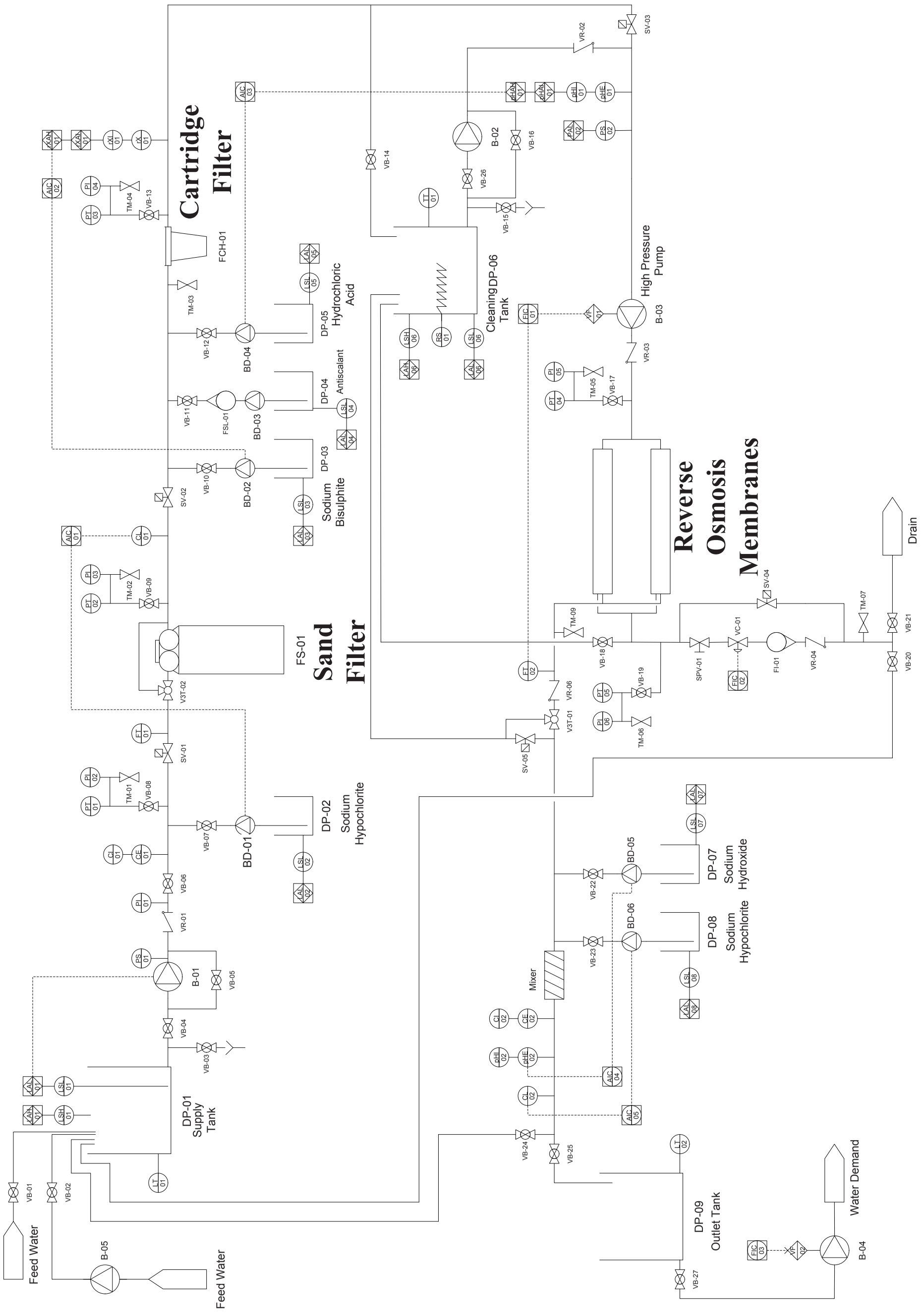


Figure 5.1: Desalination plant scheme.

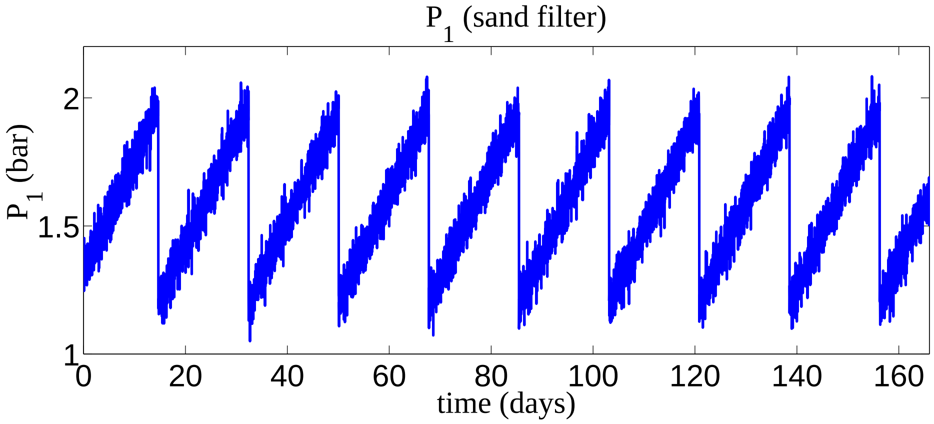


Figure 5.2: Pressure measured in the sand filter input.

and the description of the variables.

#	Name	Description	Units
1	P_3	Pressure in the cartridge filter input	*bar*
2	X_{S1}	Total solid concentration in the sand filter input	kg/m^3
3	X_{S2}	Total solid concentration in the sand filter output (cartridge filter input)	kg/m^3
4	P_1	Pressure in the sand filter input	*bar*
5	P_2	Pressure in the sand filter output (cartridge filter input)	*bar*
6	X_1	Salt concentration in the sand filter input	kg/m^3
7	P_4	Pressure in the membrane input	*bar*
8	Q_1	Flow in the sand filter input	m^3/d
9	X_2	Salt concentration in the membrane output	kg/m^3
10	Q_3	Flow in the membrane output	m^3/d
11	Q_2	Flow in the plant output	m^3/d

Table 5.1: Description of the considered variables.

Three types of faults were considered in the plant. One of them consists of an offset in the pressure sensor in the sand filter input (P_1). The other two faults are related with the membranes, concretely, faults simulated by altering several equations in the membrane model, these being a blockage and a breakage.

5.3 Application of PCA

As cited before, this system does not strictly run in a steady state due to the different particularities mentioned. So, the classical PCA scheme for monitoring and fault detection is not the most suitable solution for this system. If the plant is monitored using the T^2 statistic and the Q statistic with a classical PCA approach, a high number of alarms appears, as figure 5.3 shows. The PCA model built to perform this monitoring task was arranged with nominal data from the eleven variables measured during several running cycles and different cleaning cycles. The monitoring scheme was applied to new data collected from the plant.

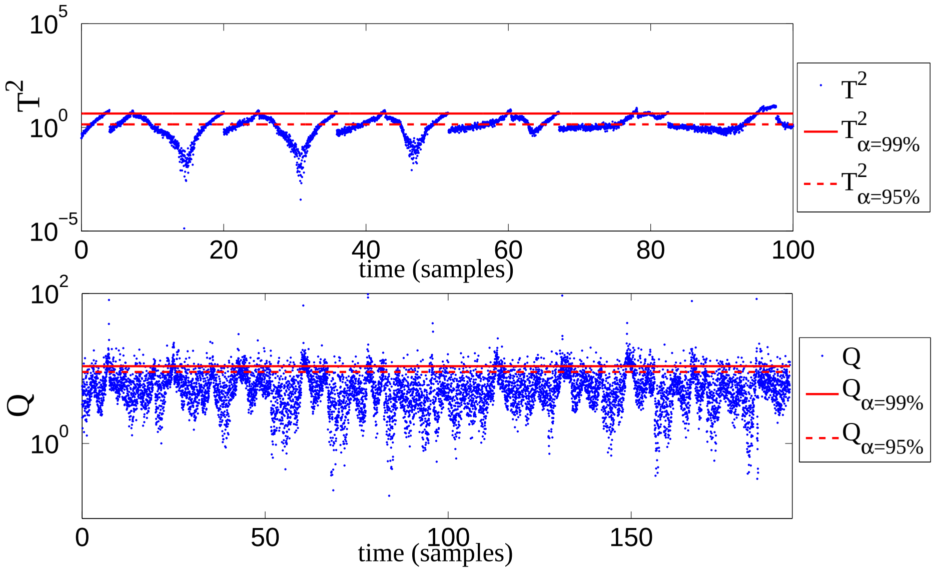


Figure 5.3: Monitoring using the classical PCA approach.

5.4 Application of Unfolded PCA

The case study of this work is a continuous process, but due to the cleaning phases for the correct plant running, its behaviour is not precisely a steady state. Each running between two cleaning cycles can be considered as a batch. Therefore, this type of processes can be treated as a batch process from the point of view of fault detection and isolation tasks.

The database for performing the PCA model can be arranged with data from past NOC (normal operation conditions) phases between cleaning cycles. Following the notation established in section 4.3, for every $i = 1, 2, \ldots, I$ normal past execution between two cleaning cycles $j = 1, 2, \ldots, J$ variables are collected at $k = 1, 2, \ldots, K$ time samples. All these data are ordered into a three-way matrix $\underline{\mathbf{X}}(I \times J \times K)$ as figure 4.3 shows.

When the data is collected from the cycles between cleaning phases, the number of samples of the data sets can have different lengths. The reason for this is that the events that activate the cleaning phases are not time events, they depend on other parameters, such as concentrations or pressures. When this phenomenon occurs, it is not possible to arrange the data matrix as shown in figure 4.3, due to the fact that the number of samples of the different data sets is not the same.

In this case study, it is not easy to find an indicator variable to perform the data alignment. As mentioned in chapter 4, when it is not possible to find an indicator variable, the dynamic time-warping (DTW) approach can be performed.

If the UPCA approach is applied to this case study a rich database must first be arranged. This database is formed by the variables measured in different past implementations between the cleaning cycles under normal operation conditions.

The values of the variables along time could be ordered into the same matrix in order to apply the UPCA approach as a first approximation for each past running cycle considered, following the structure shown in figure 4.3. However, the cleaning phase frequency is not the same, due to the different nature of the filters and the membranes. This phenomenon represents a new problem because, in the same way as applying classical PCA, these changes in the behaviour produced by the different cleaning cycles can be detected by the monitoring statistics as faults. So, there is no single criterion for ordering the data into the three-way matrix because of the different cleaning cycle

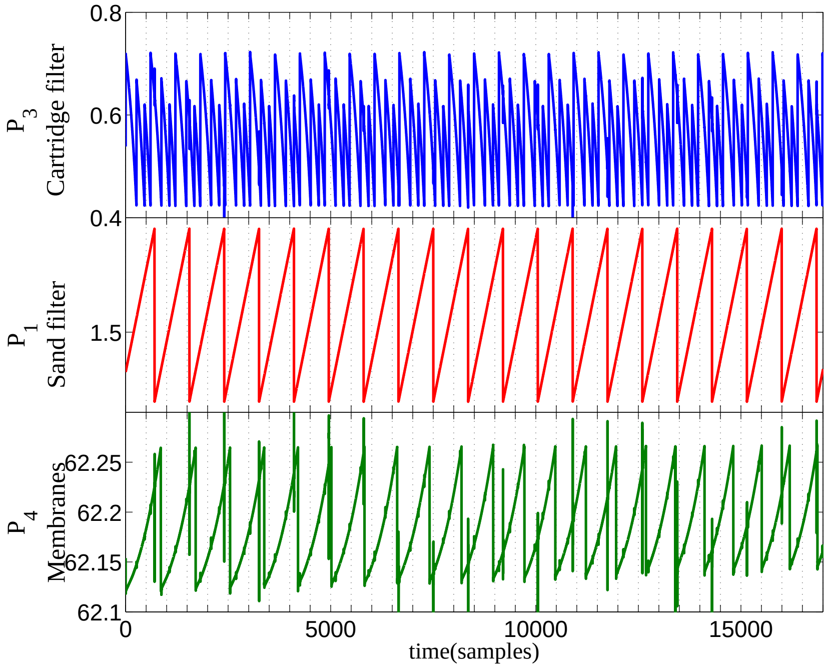


Figure 5.4: Variables affected by the cleaning cycles. The cleaning cycles are not synchronized.

frequencies. The desynchronization between the cleaning cycle events are shown in figure 5.4. In this graphic, a pressure measured in the sand filter, the cartridge filter and the membranes are shown along the same time scale.

The solution proposed to deal with this drawback consists of arranging three different UPCA models. One specific model for variables related with the membrane and another two models for the two filters respectively.

Table 5.2 shows the main characteristics of the three UPCA models considered in this work. The second column of this table shows the variables related with this specific section. The variables that are not affected by any of the cleaning cycles can be considered in several models, if they are related to two or more subsystems. Therefore, the variables affected by a particular cleaning cycle running could only be included in that specific model. The third column shows the number of principal components considered using a cross-validation procedure. The last column shows the interval of the length of the different cycles between cleaning cycles for every subsystem. This variable organization can considerably reduce the number of false alarms and allows the three critical parts of the plant to be monitored separately. Also, this configuration can be seen as a step towards distributed fault detection.

UPCA model	**Variables name**	**Number of components**	**Cycle length**
Membranes	P_2, X_1, P_4, Q_1, X_2, Q_2, Q_3	10	840-860
Sand filter	X_{S1}, X_{S2}, P_1, P_2, X_1, Q_1	8	590-610
Cartridge filter	P_3, P_2, X_1, Q_1	13	675-858

Table 5.2: UPCA models.

The cleaning cycles are not performed with a determined time frequency. The running of these cycles depends on several variables like pressure or concentration. When one or more variables rise above a determined threshold, an event is triggered and the cleaning phase begins. This is the reason why the running cycles considered to arrange the three way matrix do not have the same number of samples and an alignment method must therefore be performed to align all these cycles and to build the UPCA model. The DTW (section 4.3.2) approach was applied to achieve this objective.

Figure 5.5 shows the non-steady behaviour of the variables related to the membranes during different running cycles. The non equal duration of the different running cycles can also be seen in these graphics. The variables, after being synchronized to perform the three-way matrix needed for applying UPCA, are shown in figure 5.6. As this graphic shows, the signals are slightly deformed during the alienation procedure. The variables were synchronized to the mean length, and the added points were principally added at the beginning of the signals.

Figure 5.7 shows the evolution of the weights of the different vari-

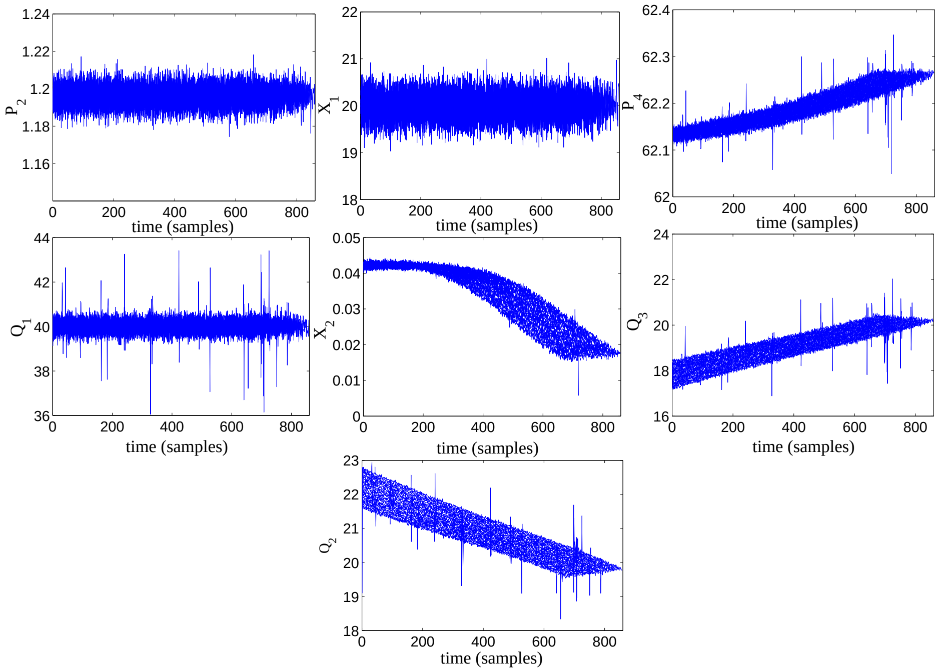


Figure 5.5: Measured variables related with the membrane.

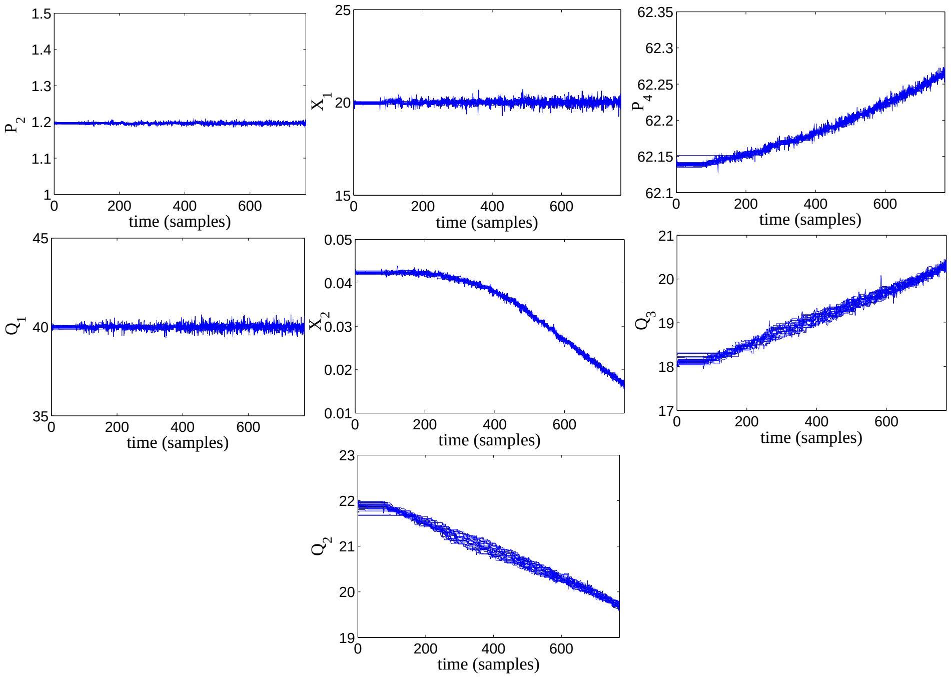


Figure 5.6: Membrane variables synchronized.

ables related to the membranes along the iterations of the DTW algorithm until convergence. The variable with the greatest importance to lead the synchronization is clearly the concentration X_2. Using an indicator variable approach, this variable would lead all the alienation, but in this case, the pressure P_4 is not so important, but still has considerable weight. The flows Q_3 and Q_2 also have considerable weight, but little importance. The rest of the variables are not taken into account to lead the alienation by the algorithm.

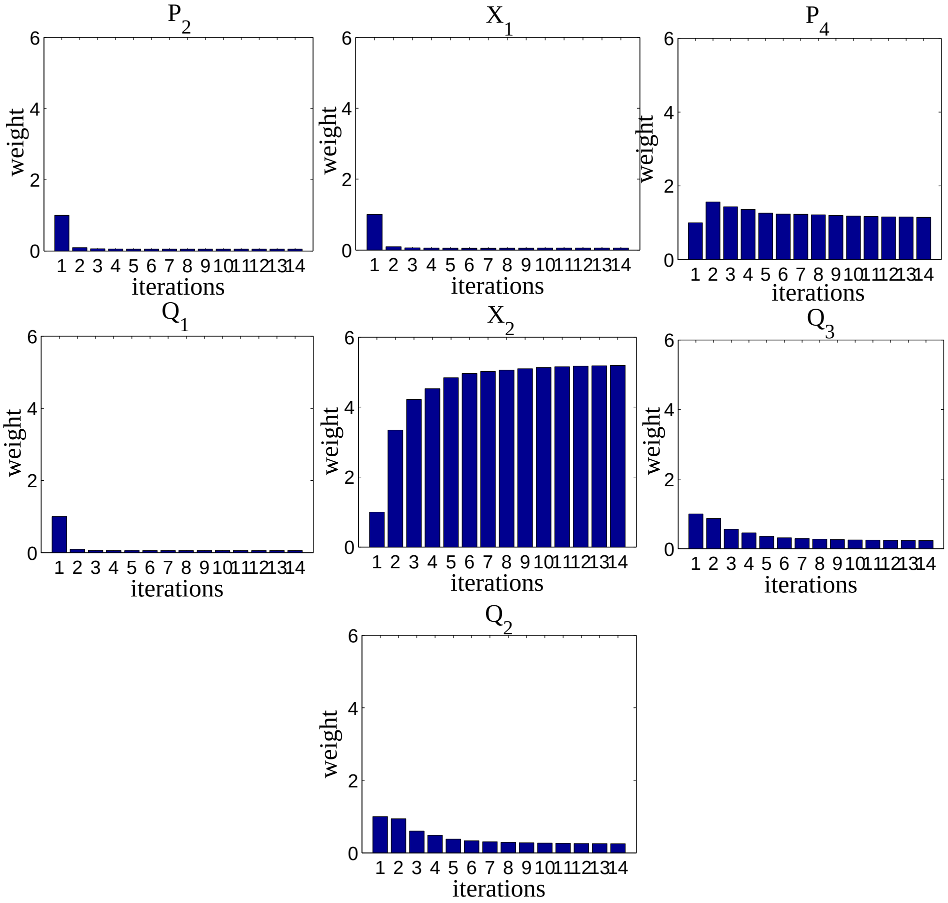


Figure 5.7: DTW variable weights.

In the same way as in the previous chapter, the prediction error sum of variances (PRESV) is used to compare the imputation error. However, in this example, the trimmed score method (TRI) and trimmed score regression method (TSR) are considered since they do not require extra *off-line* considerations and save $\mathbf{S}^{**}$ matrices with different sizes. During the plant monitoring, the future samples $(K - k)$ at sample time k $(k < K)$ are not known, as shown in figure 4.3. Figure 5.8 shows the PRESV values at different sample times for both imputation methods TRI and TSR for the monitoring of the membrane section. Nine different time points along a cycle between two cleaning phases are considered ($k_1 = 599$, $k_2 = 1198$,...$k_9 = 5391$), using the dataset of past NOC cycles. The estimation error decays along the cycle due to the decrease in the unknown data, as figure 5.8 shows.

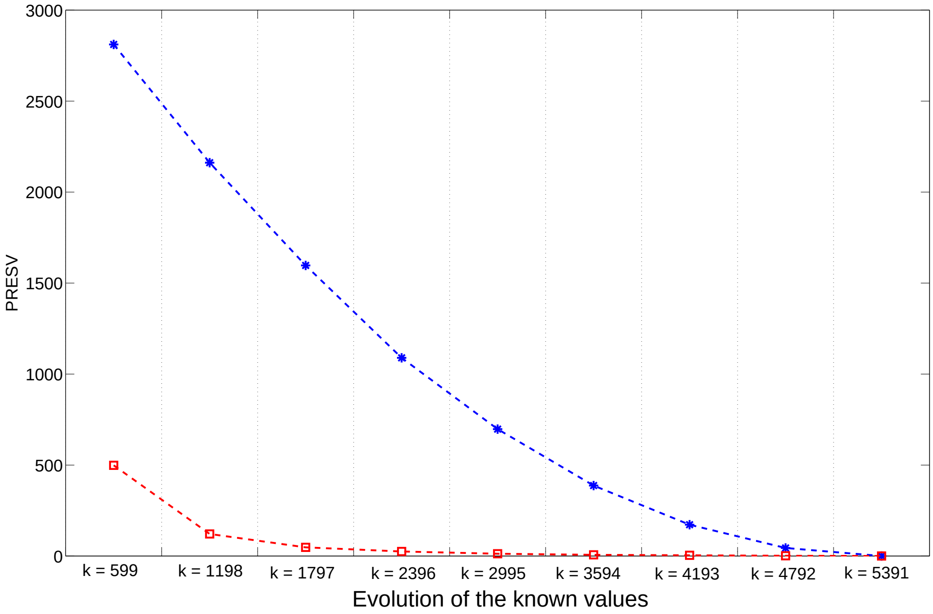


Figure 5.8: PRESV for both imputation methods considered at different sample points k. TSR method results are plotted with red squares and a dashed line and TRI method results are plotted with blue asterisks and a dashed line.

Figure 5.8 shows that the TSR method presents better results than the TRI method, despite the fact that the TRI method is usually more widely used than the TSR method by several authors. The TSR method presents less PRESV without a significant computational cost increment. Similar results were obtained for the sand filter and the carriage filter. So, in this example, the TSR method is the method selected to impute the future unknown data.

The reduction in the false alarms ratio using the explained approach is summarized in table 5.3. This table shows the false alarms ratio obtained using the UPCA approach to the three considered sections and applying the classical PCA to all variables and to the three sections. The UPCA method obtains a decrease in the number of false alarms in the T^2 monitoring statistic. A reduction in the Q statistic is also observed. The decrease is principally obtained in the membrane section.

	Percentage	
Method	T^2	Q
Classical PCA		
All variables	6.7%	16.3%
Sand filter	10.5%	11.6%
Cartridge filter	8.8%	12.9%
Membrane	9.4%	10.5%
UPCA (TSR)		
Sand filter	1.6%	10.2%
Cartridge filter	2.6%	11.3%
Membrane	1.0%	4.0%

Table 5.3: False alarms percentages.

The monitoring using T^2 and Q statistics of the membrane section applying UPCA is shown in figure 5.9. In the T^2 statistic monitoring, false alarms principally appear in the first samples, and in the rest of the monitoring performance, the statistic remains under the control limits. This can be due to the $\mathbf{P}^{*T}_{1:A}\mathbf{P}\mathbf{\Lambda}\mathbf{P}^{*T}\mathbf{P}^{*}_{1:A}$ matrix in Eq. 4.20, which may be ill conditioned at the beginning of the cycle because the known data is scant. In the case of the Q statistic monitoring, the upper limit, with $\alpha = 99\%$, is equal to one and the other control limit and the monitoring statistic are divided by the $\alpha = 99\%$ upper limit to normalize this control plot, because the control limits in this statistic are not constant.

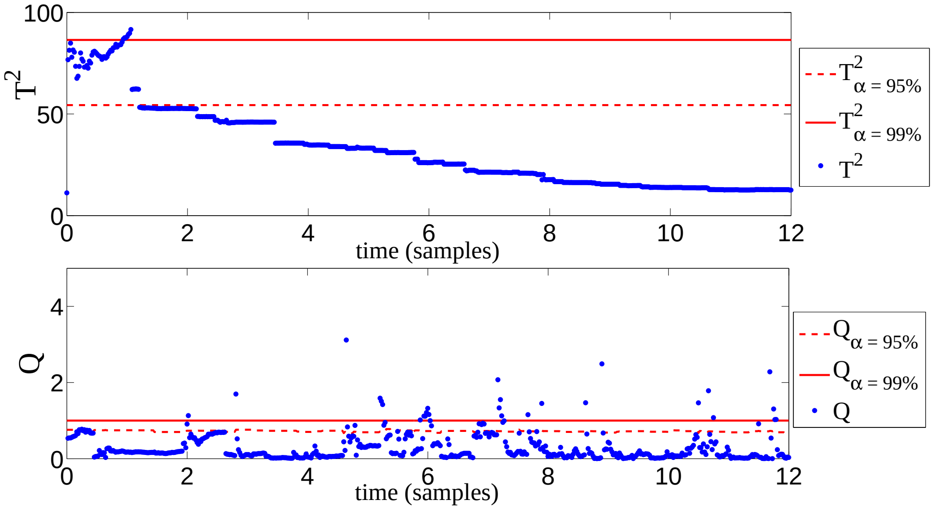
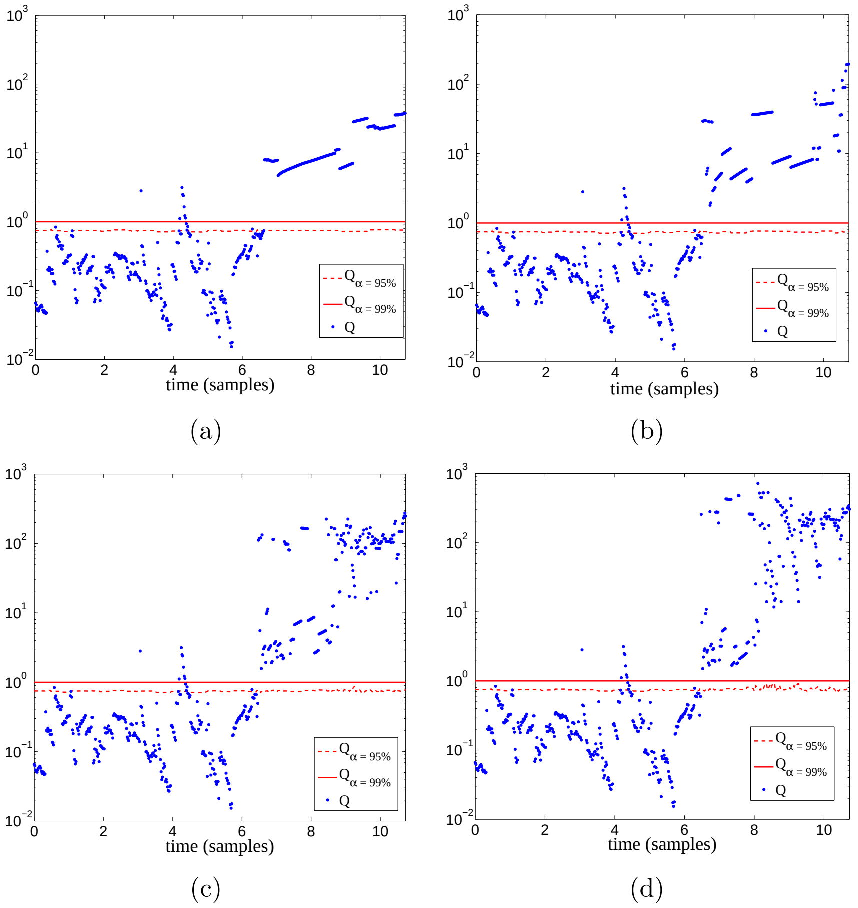


Figure 5.9: Monitoring using UPCA in the membranes section.

Table 5.4 shows the main results achieved in the fault detection task. For each considered fault, four size faults were considered. The third column (statistics) shows what statistic first detected the fault. The Q statistic detected the faults first in all the cases. The fourth column shows the time delay in fault detection. In this work, all faults considered are abrupt faults, which is why there is no delay in the fault detection. Figures 5.10(a), 5.10(b), 5.10(c) and 5.10(d) show the monitoring of a breakage with different fault sizes, and figures 5.11(a), 5.11(b), 5.11(c) and 5.11(d) show the monitoring of a blockage with different fault sizes.

When a fault is detected, contributions can be plotted to identify which variables are involved in the fault. Figure 5.12 shows the contribution plot of the normalized errors of the variables plotted after the detection of a breakage (20%) in the membrane. The figure shows that the principal variables related to this fault are the flows measured after and before the membrane. One of the pressures and one of the



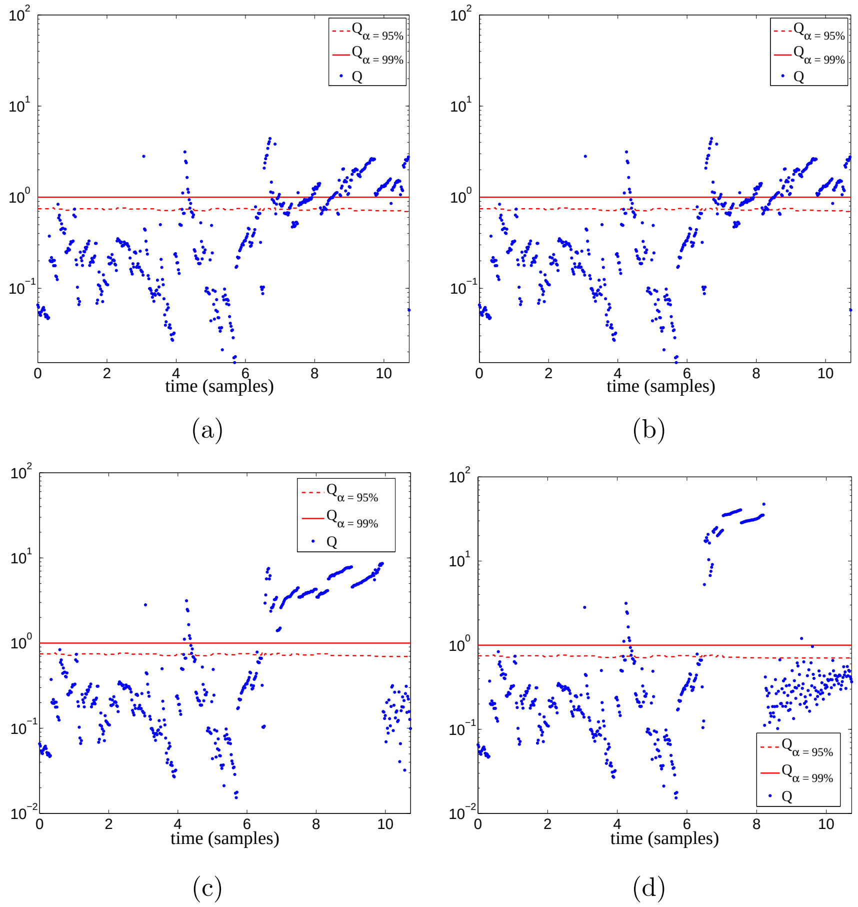
(a)

(b)

(c)

(d)

Figure 5.10: Fault detection of a breakage in the membrane with different fault sizes: 10% (a), 20% (b), 40% (c) and 60% (d).



(a)

(b)

(c)

(d)

Figure 5.11: Fault detection of a blockage in the membrane with different fault sizes: 10% (a), 20% (b), 40% (c) and 60% (d).

Fault	Size	Statistic	Delay
Membrane breakage	10%	Q	Instantaneous
	20%	Q	Instantaneous
	40%	Q	Instantaneous
	60%	Q	Instantaneous
Membrane blockage	10%	Q	Instantaneous
	20%	Q	Instantaneous
	40%	Q	Instantaneous
	60%	Q	Instantaneous
Sensor offset	10%	Q	Instantaneous
	20%	Q	Instantaneous
	40%	Q	Instantaneous
	60%	Q	Instantaneous

Table 5.4: Detection results.

concentrations present anomalous values too.

The faults are detected by the Q statistic in all the cases and it is not necessary to inspect the T^2, so the contribution analysis related with this statistic is not necessary. Despite this, figures 5.13 and 5.14 show an example of the contribution plot of normalized scores and the variable contributions to the three greatest scores plot in the last case, the contributions are grouped by variables in order to show the evolution of the variables along the cycle and identify what variables present an abnormal value.

5.5 Discussion

This chapter proposed the use of the UPCA approach, normally used in batch process monitoring, to monitor continuous processes without a strictly steady behaviour. This work is based on the adaptation of the PCA approach used in batch processes (UPCA). The processes that do not operate a steady state can present a highly non-linear behaviour and they cannot be monitored using classical PCA schemes because the non-linearities can be detected as faults by the monitoring

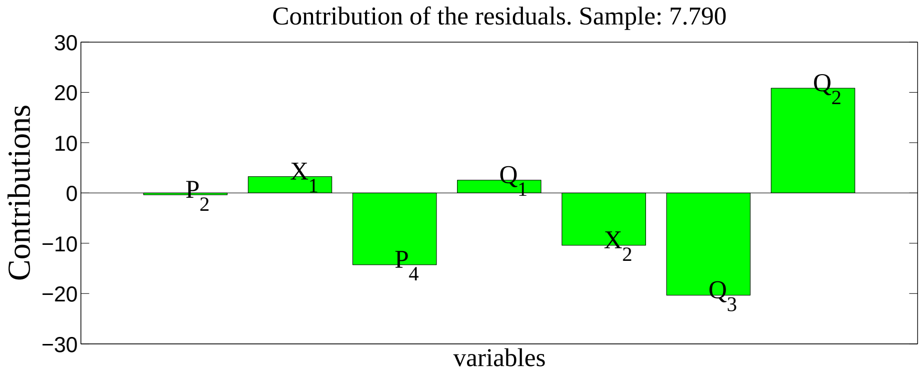


Figure 5.12: Contribution plot of the normalized errors of the variables of a breakage in the membrane.

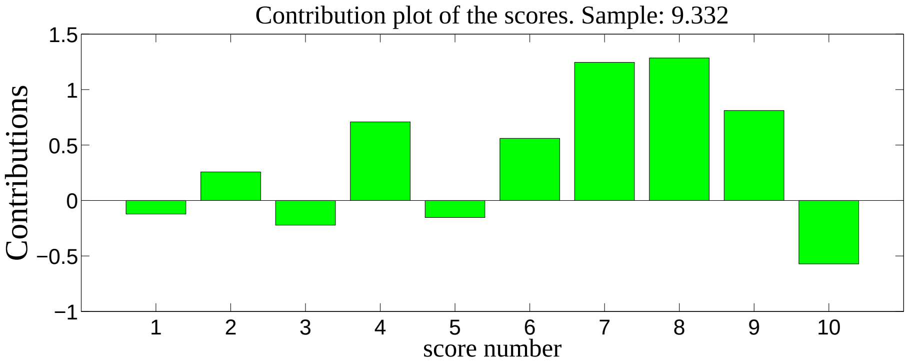


Figure 5.13: Contribution bar plot of normalized scores corresponding to a breakage in the membrane.

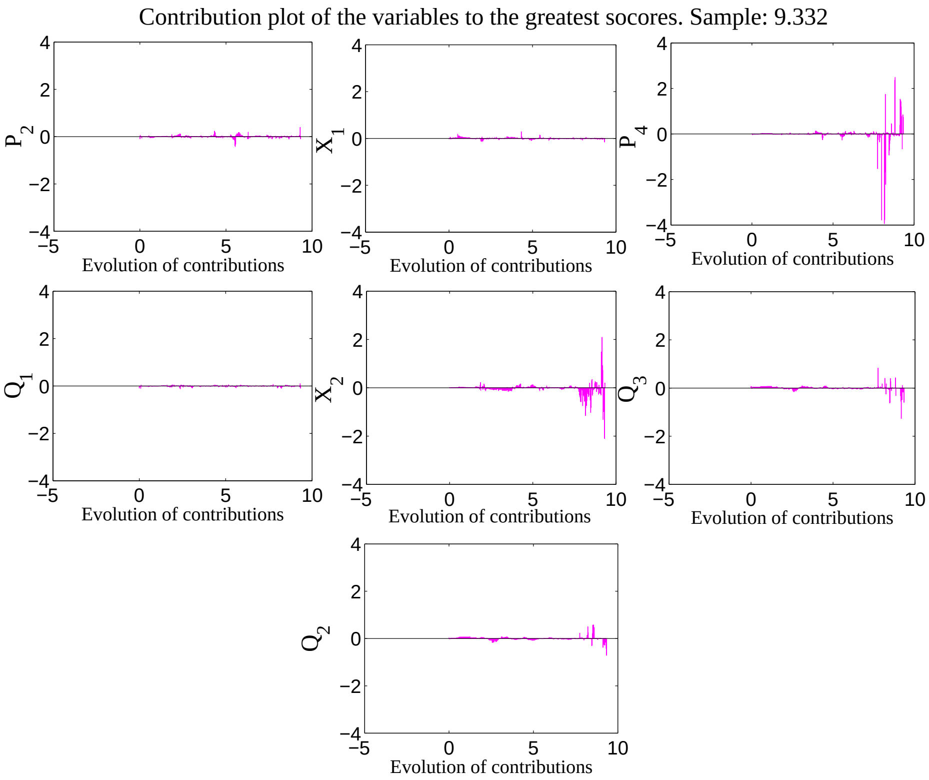


Figure 5.14: Variable contributions bar plot to three greatest scores corresponding to a breakage in the membrane.

statistics.

The used case study consists of a reverse osmosis desalination plant. This type of plants require cleaning cycles in several components, like the filters and the membranes. The running of these cycles means that the process does not operate in a strictly steady state.

The data of this kind of plants are arranged into a three-way matrix using past data from past nominal cycles. The PCA model is calculated using an unfolded two dimensional matrix.

The imputation of unknown data problems that appear during the *on-line* monitoring task are presented. In this case, the TSR method is applied instead of the TRI method, which is normally used by several authors. This choice is based on a study of the PRESV calculated for both methods, also presented in this work.

The main result is a reduction in the false alarms ratio as the UPCA approach is more suitable for this type of processes.

Chapter 6

Combination of structural model decomposition techniques and PCA

6.1 Introduction

Complex systems require efficient monitoring and fault detection and isolation schemes. Nevertheless, accurate and fast real-time monitoring of such systems can be compromised due to the complexity of the system and the size of the measurement space. In fact, monitoring and fault detection of complex systems usually requires the integration of several techniques coming from different research fields such as knowledge-based, case-based, model-based reasoning or machine-learning, among others.

In chapter 4, the monitoring of transient states of continuous processes is performed using the UPCA approach. A PCA model must be performed for each operating mode and a UPCA model for each transient state. In processes with few transitions and steady states, this approach can be performed without many *off-line* tasks.

However, processes with many operating modes and transient states can require many *off-line* tasks. One of the main advantages of the process history-based methods is that *off-line* tasks are less complex than first-principles model-based methods.

When the process has many operating modes, a model-based approach can be a better solution for designing a fault detection and isolation approach. If the model is dynamical and reproduces accurately the behaviour of the real process, it can model both steady and transient states.

Another important drawback of the PCA-based monitoring schemes is that they provide little support for fault isolation. PCA is able to detect faults as well as the set of variables involved in such faults, but this cannot be interpreted as an isolation or diagnosis stage.

On the other hand, diagnosis approaches for *on-line* fault diagnosis based on analytical models (Blanke et al., 2003; Gertler, 1998) require quick and robust detection approaches to detect significant deviations between expected and observed behaviour. The deviations are calculated using residuals, which are related to analytical redundancy derived from the system model. The structural model of these residuals can be computed *off-line.* However, they are evaluated *on-line.* Whenever the value of a residual exceeds a given threshold, the fault detection is performed and the set of constraints used to derive the analytical redundancy expression is considered to be non-consistent with observations. After this process, the fault isolation task is performed, and a reduced set of faulty candidates can be computed.

Residuals can be calculated using different methods, such as parameter estimators, state-observers or parity-equations. In this work, structural analysis techniques are used for residual generation. Specifically, possible conflicts (PCs) (Pulido and Alonso-González, 2004) are used. PCs are equivalent to analytical redundancy relations (ARRs) (Pulido and Alonso-González, 2004). Possible conflicts are able to decompose the system model into minimal structurally overdetermined subsets of equations required for fault diagnosis.

In PCA-based approaches, two control charts only have to be observed in the detection stage. So, the use of this type of approaches can provide simpler detection tests. Moreover, these statistics also provide robustness in fault detection.

So, in this chapter, a first-principles model-based approach to perform fault detection and isolation using analytical and statistical models is performed. A first principles model-based method known as possible conflicts (PCs) is combined with PCA to improve the diagnosis process for complex systems. The residuals computed using PCs are considered as inputs for the PCA computation. The PCA determines significant deviations in the residuals, which will be identified as faults.

6.2 Possible conflicts diagnosis approach

Possible Conflicts (Pulido et al., 2001; Pulido and Alonso-González, 2004) is an *off-line* dependency compilation technique from the artificial intelligence diagnosis (DX) community. This DX method requires the formulation of the system in a first principles model. PCs are minimal subsets of equations with enough redundancy to perform fault diagnosis. The main idea of possible conflicts is that all those subsystems capable of becoming a conflict can be identified *off-line*. A PC represents the structure of an ARR that can be used for fault detection and isolation.

The computation of PCs is performed in three steps: obtain an abstract representation of the system as a hypergraph, search the whole set of minimal over-determined subsets of equations and finally check if they can be solved using local propagation alone.

These three stages are explained as follows:

1. **Generating an abstract representation of the system:** Computation of PCs can be done *off-line* by using an abstract representation of the system: a hypergraph (Pulido et al., 2001). This type of representation only gives information about constraints or equations in the model and their relationship with the unknown and known variables in the model. The hypergraph considers the equations or constraints from a structural point of view, i.e., it considers the equations as a set of variables where each variable can be calculated with respect to the rest of the variables in the set.

2. **Searching the minimal evaluation chains:** In this step, a search algorithm finds minimally overdetermined subsets (sub-

hypergraphs) of equations or constraints, called minimal evaluation chains (MECs), in the hypergraph, i.e., sets of equations with more equations than variables. Each one of these minimal evaluation chain (MECs) sets represents a necessary condition for a conflict to exist. In this search, the equations are also considered from a structural point of view and all the interpretations (each particular causal assignment) are considered. When a variable inside a constraint can be solved assuming the rest of the variables are known, this is called an interpretation, i.e. a feasible causal assignment, and this leads to the third step. For example, for the equation $a = bc$, there are three possible interpretations: $a = bc$, $b = a/c$ and $c = a/c$. Table 6.3 shows an example of the computation of a MEC.

3. **Searching the minimal evaluation models:** This stage searches for all the causally consistent interpretations for each constraint in a MEC, which is called the Minimal Evaluation Model (MEM). Each MEM represents a globally consistent causal assignment within a MEC and can be used to estimate the behaviour of a part of the system. A MEM can be seen as a MEC with a consistent causal assignment. For example, in the equation $a = b+\sqrt{c}$, there are two possible interpretations: $a = b+\sqrt{c}$ and $b = a-\sqrt{c}$, but the value of c cannot be found in this equation. MEMs are the analytical representation of MECs. Figure 6.7 shows all the equations and relationships of a MEM and the possible related conflicts.

Figure 6.1 shows a scheme of the neccesary steps for performing the residuals related with the possible conflicts of a system.

Since conflicts only arise when models are evaluated using the set of available observations, the set of constraints within a MEM is called a possible conflict.

The set of MEMs is used to perform the fault detection task by searching for discrepancies between the estimated and observed variables. These discrepancies can be considered because a variable can be calculated in at least two ways. These discrepancies are also known

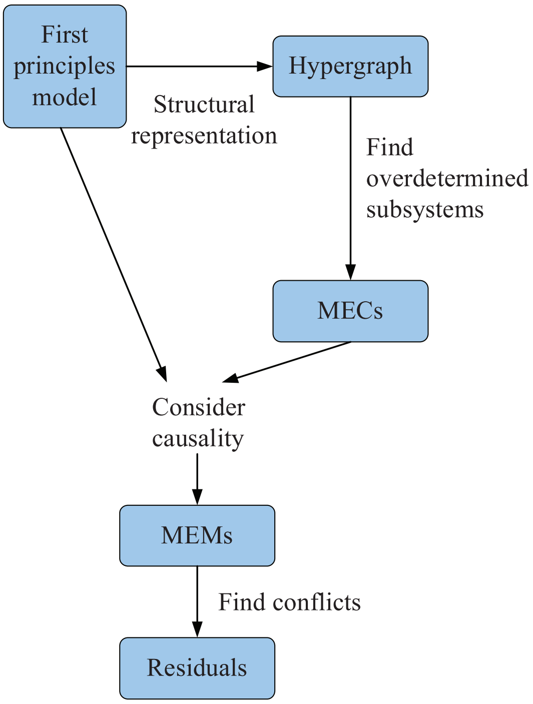


Figure 6.1: Scheme of possible conflicts generation.

as residuals. If a discrepancy is detected, a component of the possible conflict would be responsible for such a discrepancy and it will be confirmed as a real conflict. Then, diagnosis candidates or faulty components can be obtained following Reiter's theory (Reiter, 1987).

Further information about possible conflicts and consistency-based diagnosis using PCs can be found in Pulido et al. (2001) and Pulido and Alonso-González (2004).

6.3 Integration of PCs and PCA for fault detection and diagnosis

PCA has been used as a tool for fault detection in complex industrial plants, but, as mentioned before, it can present problems when changes in the operating mode appear.

The solution proposed in this chapter does not fall into any of the categories considered in section 4.2. In this work, a classical PCA approach is applied over the residuals obtained by a first-principles model-based method instead of the original measured process variables. The residuals generated by possible conflicts are only sensitive to a subset of faults and not sensitive to changes in operating conditions.

PCA does not provide tools to isolate faulty candidates when a fault is detected in the system. There are several authors who have proposed solutions for this problem. Gertler et al. (1999) have used structured partial PCA models with the same isolability properties as the parity relations. Using that equivalence, Gertler et al. (1999) and Huang et al. (2000) decompose the original PCA model into smaller structured PCA models that guarantee the disturbance decoupling for the set of faults considered.

In the integration proposed in this work, instead of designing *off-line* a set of partial structured PCA, a PCA model is designed using the residuals of the PCs, which can be considered as ARRs. When a fault is detected *on-line* using the monitoring statistics, a contribution analysis is performed to find the residuals responsible for the deviation in the PCA monitoring task. The fault is isolated using the theoretical fault signature matrix provided by the PCs and the activated residuals.

Figure 6.2 shows the scheme of the integration of both methods. As cited before, a first principles model is required. This system model is decomposed into minimal structural overdetermined subsystems by computing the set of possible conflicts. The residuals computed by the possible conflicts for training data under normal operation conditions (NOC) are used to establish the PCA model. All these tasks are performed *off-line*.

During the *on-line* execution of this scheme, the residuals R_{PC_1} … R_{PC_i}, one for each PC, are performed as the linear difference between the estimations provided by the PCs, $\mathbf{\hat{y}_{pc_x}}$, and the real plant measurements, $\mathbf{y}$. These residuals are introduced into the PCA block that performs the fault detection task using T^2 and Q statistics.

When the PCA block detects that the system is outside the nominal situation, the contributions computation block identifies the vari-

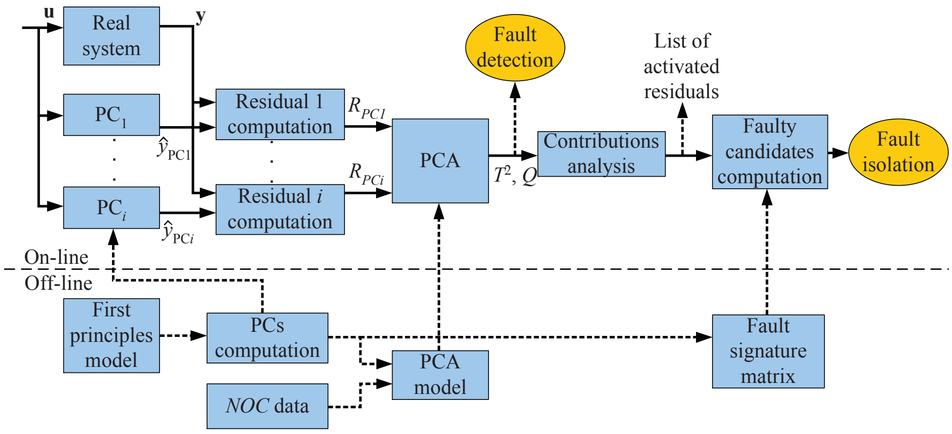


Figure 6.2: Integration proposal scheme.

ables (residuals) responsible for such a faulty situation. Finally, the set of faulty candidates is computed by a minimal hitting set procedure of the residuals responsible for the fault, using the fault signature matrix computed by the possible conflicts.

6.4 Case study: two communicated tanks

The laboratory plant used to test the proposed integration scheme is the controlled two-tank system described in section 3.3, and also used in subsection 4.5.2. The steady states, transient states and experiments considered in this example are the same as those considered in subsection 4.5.2. In this example, the considered measured variables are the tank levels measured by sensors: h_1 and h_2.

The variation of level in the tanks can be modelled as:

$$\mathrm{c}_1: \ A\dot{h}_1(t) = q_1(t) - q_{12}(t) - q_{10}(t) \tag{6.1}$$

$$\mathrm{c}_2: \ A\dot{h}_2(t) = q_2(t) + q_{12}(t) - q_{20}(t) \tag{6.2}$$

where A is the area of the cylindrical tanks.

According the Torricelli's law, flows q_{12}, q_{10} and q_{20} are defined by:

$$\mathrm{c}_3: \; q_{12}(t) = K_{12}\,\mathrm{sign}\,(q_1(t) - q_2(t))\,\sqrt{2g\,|h_1(t) - h_2(t)|} \tag{6.3}$$

$$\mathrm{c}_4: \; q_{10}(t) = K_{10}\sqrt{2gh_1(t)} \tag{6.4}$$

$$\mathrm{c}_5: \; q_{20}(t) = K_{20}\sqrt{2gh_2(t)} \tag{6.5}$$

where the sign() is the sign function.

The following equations show the reading of the levels by the sensors:

$$\mathrm{c}_6: \; h_1(t) = h_1(t)^* \tag{6.6}$$

$$\mathrm{c}_7: \; h_2(t) = h_2(t)^* \tag{6.7}$$

Finally, the level of the tanks can also be calculated by the integration of the derivative of the tank levels:

$$\mathrm{c}_8: \; h_1(t+1) = \int_0^{t+1} \dot{h}_1(t)dt + h(0) \tag{6.8}$$

$$\mathrm{c}_9: \; h_2(t+1) = \int_0^{t+1} \dot{h}_2(t)dt + h(0) \tag{6.9}$$

In this case, the pumps and the controllers are not taken into consideration. Flows q_1 and q_2, provided by pumps P_1 and P_2, are considered as known inputs in this example.

Seven different faults were considered in this example. Table 6.1 shows the fault identifiers, the component affected by each one of them, the related constraint or equation and the description.

Table 6.2 shows the constraints considered in the hypergraph of this example. Variables marked with an asterisk (*) are known variables, such as measured variables or known inputs.

Fault	**Component**	**Constraint**	**Description**
$f_{p_{10}}$	p_{10}	c_4	Blockage of pipe/valve p_{10}
$f_{p_{20}}$	p_{20}	c_5	Blockage of pipe/valve p_{20}
$f_{p_{12}}$	p_{12}	c_3	Blockage of pipe/valve p_{12}
f_{T_1}	T_1	c_1	Leakage in tank T_1
f_{T_2}	T_2	c_2	Leakage in tank T_2
f_{LT_1}	LT_1	c_6	Faulty sensor LT_1
f_{LT_2}	LT_2	c_7	Faulty sensor LT_2

Table 6.1: List of faults with the related components and constraints.

Constraints	**Variables**
c_1	$\{\dot{h}_1,\ q_1^*,\ q_{12},\ q_{10}\}$
c_2	$\{\dot{h}_2,\ q_2^*,\ q_{12},\ q_{20}\}$
c_3	$\{q_{12},\ q_1^*,\ q_2^*,\ h_1,\ h_2\}$
c_4	$\{q_{10},\ h_1\}$
c_5	$\{q_{20},\ h_2\}$
c_6	$\{\dot{h}_1,\ h_1^*\}$
c_7	$\{\dot{h}_2,\ h_2^*\}$
c_8	$\{\dot{h}_1,\ h_1\}$
c_9	$\{\dot{h}_2,\ h_2\}$

Table 6.2: Constraints of the two tank system.

Table 6.3 shows the steps that the algorithm follows to generate a minimal evaluation chain (MEC). In this case, the minimal evaluation chain that generates the possible conflict PC_1. In the first equation considered c_1, three variables are unknown $\dot{h}_1, q_{12}, q_{10}$. In the second step, the variable $\dot{h}_1$ is solved using the constraint c_8, which includes h_1 as unknown variable. This procedures is repeated until the set of unknown variables is empty.

A set of four possible conflicts were found in the simulation model. They are shown in table 6.4. These possible conflicts are minimal with respect to the set of constraints in the models. In the table, the first column shows the PC identifiers, the second column lists the set of components involved in each PC, the third column lists the constraints of each possible conflict and finally, the fourth column indicates the discrepancy node for each PC.

Figures 6.3, 6.4, 6.5 and 6.6 show the hypergraph and physical com-

Constraints	**Unknown variables**
$\mathrm{c}_1 : \{\dot{h}_1, q_1^*, q_{12}, q_{10}\}$	$\dot{h}_1$, q_{12}, q_{10}
$\mathrm{c}_8 : \{\dot{h}_1, h_1\}$	q_{12}, q_{10}, h_1
$\mathrm{c}_3 : \{q_{12}, q_1^*, q_2^*, h_1, h_2\}$	q_{10}, h_1, h_2
$\mathrm{c}_4 : \{q_{10}, h_1\}$	h_1, h_2
$\mathrm{c}_6 : \{\dot{h}_1, h_1^*\}$	h_2
$\mathrm{c}_7 : \{\dot{h}_2, h_2^*\}$	$\emptyset$

Table 6.3: MEC of possible conflict PC_1.

#	**Components**	**Constraints**	**Estimate**
PC_1	T_1, T_2, p_{10}, p_{20}, p_{12}, LT_1	$\mathrm{c}_1, \mathrm{c}_3, \mathrm{c}_4, \mathrm{c}_6, \mathrm{c}_7, \mathrm{c}_8$	h_1
PC_2	T_1, p_{10}, p_{12}, LT_1, LT_2	$\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3, \mathrm{c}_4, \mathrm{c}_6, \mathrm{c}_8, \mathrm{c}_9$	h_1
PC_3	T_1, T_2, p_{10}, p_{20}, p_{12}, LT_2	$\mathrm{c}_2, \mathrm{c}_3, \mathrm{c}_5, \mathrm{c}_6, \mathrm{c}_7, \mathrm{c}_9$	h_2
PC_4	T_2, p_{20}, p_{12}, LT_1, LT_2	$\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3, \mathrm{c}_5, \mathrm{c}_7, \mathrm{c}_8, \mathrm{c}_9$	h_2

Table 6.4: PCs found for the plant: components, constraints and estimated variable for each considered fault.

ponents of each possible conflict. The red arrow with two tips shows the discrepancy between the variables used to compute the residual. The constraint related with each PC can also be seen in the hypergraphs.

Figure 6.7 explains in detail the computation of the residual of the possible conflict PC_1. In this scheme, all the equations related to the possible conflict PC_1 are shown. Each equation has a box above it which contains the variable being estimated. In the right-hand part of the figure, depicted in orange, the value of the level of tank T_1 (h_1) is estimated using equation 6.6 (c_6), i.e., the value read by sensor LT_1. In the left-hand part of the figure, depicted in blue, the value of this level is calculated using several equations. The level is calculated by the integration of the derivative of level $\dot{h}_1(t)$ (equation c_8). The derivative of level $\dot{h}_1(t)$ is calculated by means of equation 6.1 (c_1). This equation requires three values to calculate this level. The first one is the flow $q_1^*(t)$ provided by pump p_1. It is an input and is known. The second one is the flow $q_{10}(t)$ of pipe (p_{10}). This flow is calculated by means of equation 6.4 (c_4) using level $h_1(t)$. The value of level $h_1(t)$ is the level estimated by the blue part of this possible conflict at

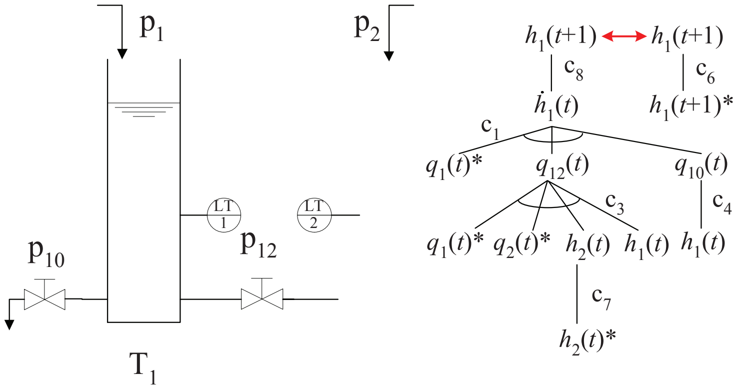


Figure 6.3: Hypergraph and components related with possible conflict PC_1

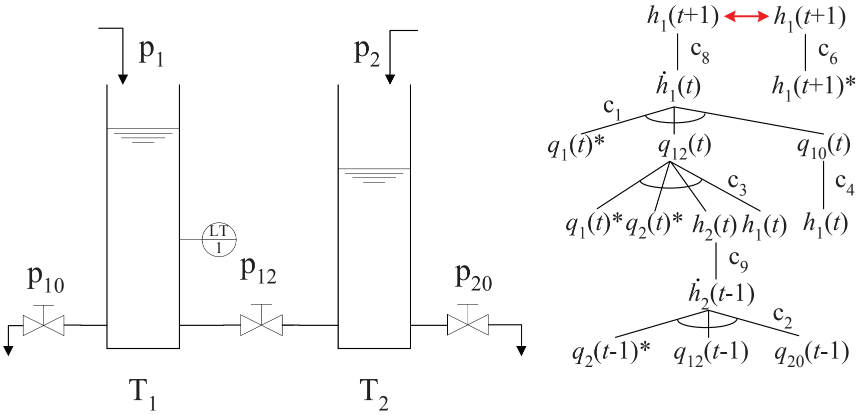


Figure 6.4: Hypergraph and components related with possible conflict PC_2

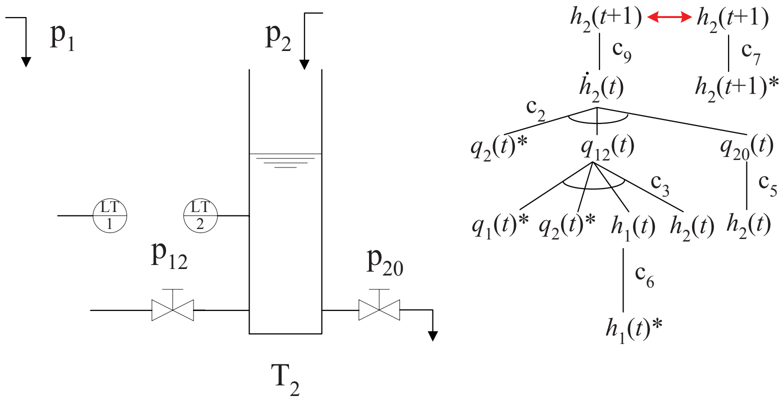


Figure 6.5: Hypergraph and components related with possible conflict PC_3

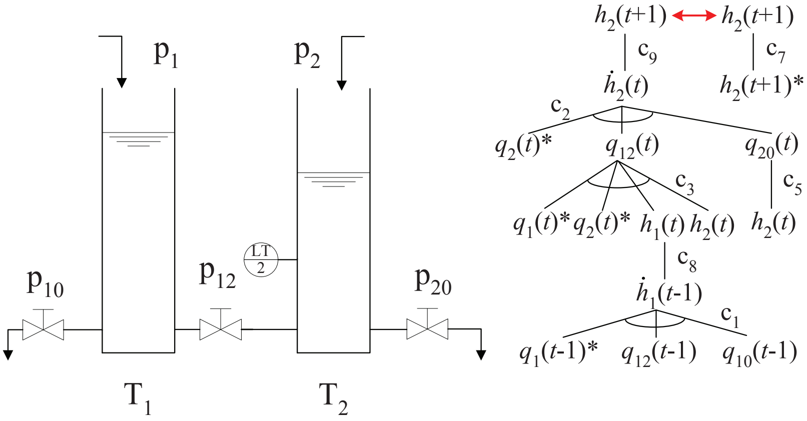


Figure 6.6: Hypergraph and components related with possible conflict PC_4

the previous time instant ($h(0)$ is required). The third one is the flow ($q_{12}(t)$) of the pipe p_{12}, which communicates both tanks. This flow is calculated by means of equation 6.3 (c_3). This equation computes the flow using the known inputs $q_1^*(t)$ and $q_2^*(t)$, the level $h_1(t)$, also known, and the level of tank T_2 ($h_2(t)$). This level is calculated by means of equation 6.7 (c_7) using the measurement of level sensor LT_2. Finally, the difference between both levels, $h_1(t+1)$, is the residual of this possible conflict.

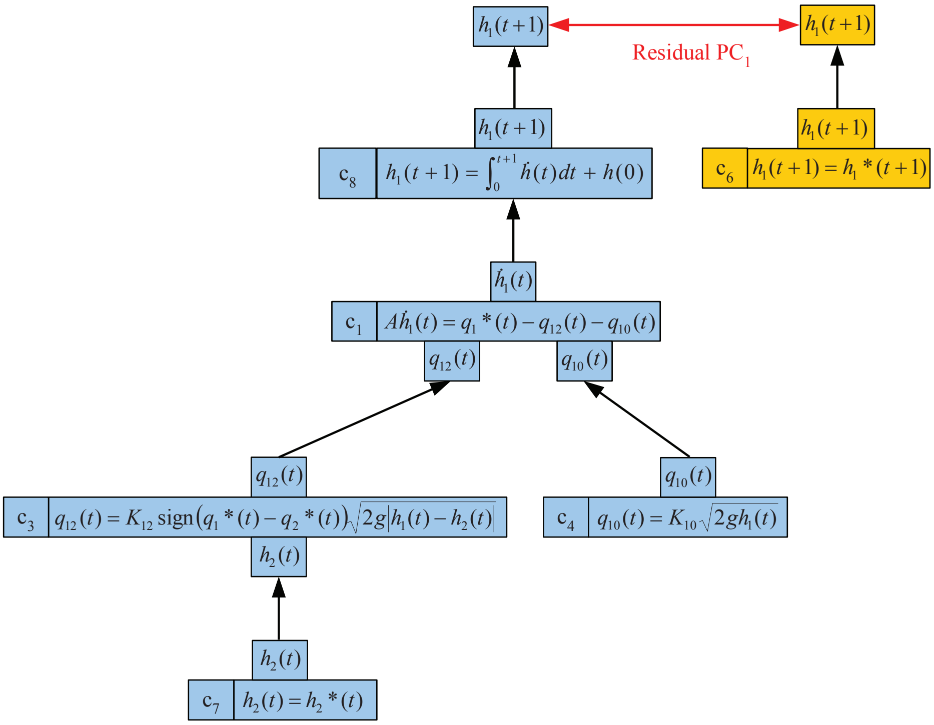


Figure 6.7: Explanation of possible conflict PC_1.

Table 6.5 shows the relation between the PCs and the faulty components. This relation is performed by looking at the components related with each possible conflict. This table is known as the theoretical fault signature matrix. This matrix describes the residuals of each PC that should be triggered when a fault occurs in a component.

The PCA model was performed using the residual of possible con-

	$\mathbf{f_{p10}}$	$\mathbf{f_{p20}}$	$\mathbf{f_{p12}}$	$\mathbf{f_{T_1}}$	$\mathbf{f_{T_2}}$	$\mathbf{f_{LT_1}}$	$\mathbf{f_{LT_2}}$
PC_1	1	1	1	1	1		1
PC_2	1	1	1	1	1	1	
PC_3	1		1	1		1	1
PC_4		1	1		1	1	1

Table 6.5: PCs and their related fault modes. The set of faults considered in this plant are: faulty sensors (f_{LT_1}, f_{LT_2}), blockages of pipes/-valves ($f_{p_{10}}$, $f_{p_{20}}$, $f_{p_{12}}$), and leakages (f_{T_1}, f_{T_2}).

flicts calculated for 10 real experiments in the NOC situation. The experiments were run for 1600 seconds. The PCA model has been fitted with three principal components. The detection thresholds for the PCA model were fitted with a level of significance $\alpha = 95\%$.

Table 6.6 shows the percentage of alarms detected with both approaches: the PCA in the second column and the combination of PCA and possible conflicts (PCA + PCs) in the third column. These data were obtained for 20 experiments in a nominal situation. It is clear that the number of false alarms is reduced when the PCA and PCs are used together looking at the mean of both considered cases. Indeed, in this case, after performing an analysis of variance (ANOVA) for the comparison of both mean values, the *P-value* of the *F-ratio* is lower than 0.05, so there exists a statistically significant difference between the means of the methods at the 95.0% confidence level (results obtained using STATGRAPHICS Centurion XVI ©).

In the fault detection phase, as mentioned in other chapters, the occurrence of a fault in the system is determined by a consecutive number of alarms. Hence, a decrease in the number of false alarms can be seen as a decrease in the number of false positives in the detection task. In this case study, at least one false positive was obtained in all the NOC experiments when the PCA was used alone (due to start-ups and changes in the references). However, no false positives were detected when the proposed integration approach was used. Figure 6.8 shows the monitoring of the T^2 and Q statistics for an example in normal operation conditions (NOC) when the PCA is used alone. Figure 6.9 shows the same example when the integration of PCs and

#	**PCA**	**PC+PCA**
1	7.37	2.06
2	7.80	3.25
3	7.06	1.69
4	6.99	0.68
5	7.93	3.44
6	7.06	3.43
7	7.25	1.12
8	7.81	1.37
9	7.93	1.99
10	7.62	1.87
11	8.62	3.74
12	8.06	0.74
13	7.62	1.18
14	7.99	1.18
15	7.75	1.99
16	7.43	3.31
17	8.81	1.18
18	7.62	1.37
19	7.81	2.18
20	8.50	2.43
Mean	7.75	2.01
Std	0.50	0.96
ANOVA data		
F-ratio		558.71
P-value		0.000001

Table 6.6: Mean value of the false alarms percentage obtained for 20 experiments in nominal situation.

the PCA is applied. By looking figures at 6.8 and 6.9, it is clear that the use of the PCA alone causes a large number of alarms and one false positive (during the start-up) is detected, while the combination of techniques causes a smaller number of alarms and no false positives.

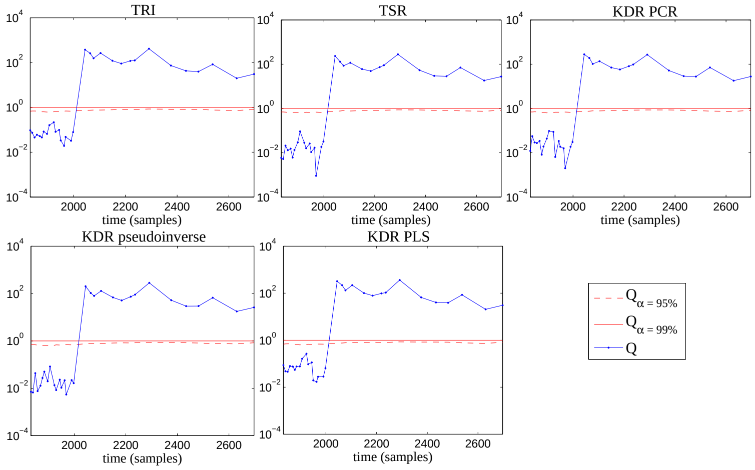


Figure 6.8: T^2 and Q statistics for a nominal experiment when PCA is used alone.

Table 6.7 shows the results obtained in different faulty situations. In this case, only sensor faults (LT$_1$ sensor and LT$_2$ sensor) are considered. 40% and 60% fault sizes are studied. These faults occur at time samples $t = 100$, $t = 500$, $t = 850$ and $t = 1300$ seconds. The faults at time instant $t = 100$ and $t = 850$ are induced during the start-up and transient state respectively. The other two faults are induced during the steady states. Table 6.7 shows the detection times using the T^2 and the Q statistics (T^2 detection time and Q detection time), as well as the time instant after fault detection when the approach is able to uniquely isolate the fault (T^2 isolation time for the T^2 statistic, and Q isolation time for the Q statistic).

Looking at the results, the proposed approach can accurately detect every fault considered. For all the experiments, the T^2 statistic detects the faults during the first time steps after the fault occurrence. It did so before the Q statistic. Regarding fault isolation, the T^2 statistic isolates the faults for all the experiments run except one,

Fault instant	$t = 100$	$t = 500$	$t = 850$	$t = 1300$
Faulty component	$\mathbf{h_1}$ **sensor**			
Fault size	40 % fault size			
T^2 detection time	105	501	900	1301
T^2 isolation time	175	571		1361
Q detection time	164	543		
Q isolation time				
Fault size	60 % fault size			
T^2 detection time	105	501	900	1301
T^2 isolation time	163	541	941	1361
Q detection time	162	521	940	1365
Q isolation time				1381
Faulty component	$\mathbf{h_2}$ **sensor**			
Fault size	40 % fault size			
T^2 detection time	101	501	901	1301
T^2 isolation time	161	521	961	1661
Q detection time		579	969	
Q isolation time				
Fault size	60 % fault size			
T^2 detection time	101	501	901	1301
T^2 isolation time	141	521	941	1341
Q detection time	160	524	953	1462
Q isolation time		524		1462

Table 6.7: Results for different faulty situations when the integration of PCA and PCs is used.

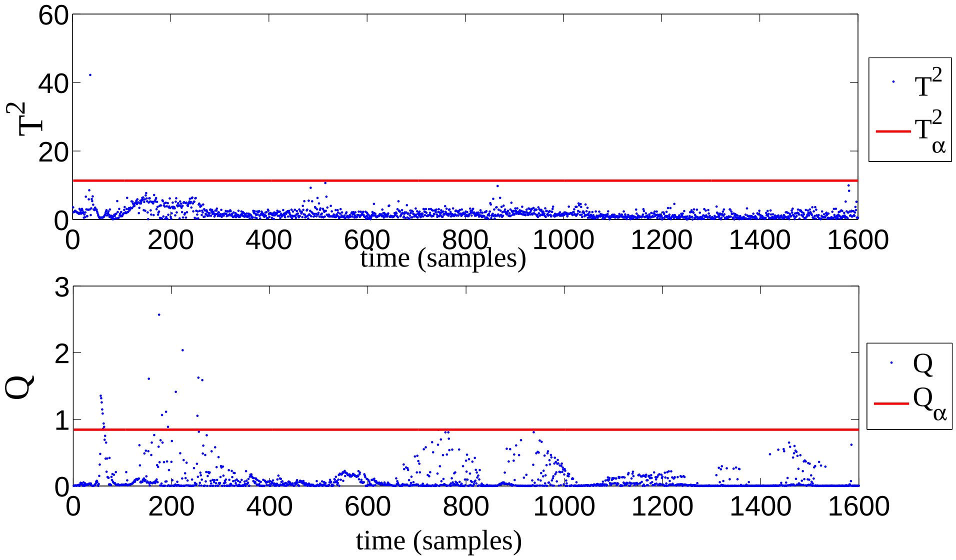


Figure 6.9: T^2 and Q statistics for a nominal experiment when PCA is used together with PCs.

a 40% fault size at $t = 850$ in the LT_1 sensor. In this table, the results only present an isolation time when the faults are uniquely isolated, but for the rest of the cases, the approach could isolate a small subset of faulty candidates, reducing the initial number of faulty candidates.

Figures 6.10, 6.11, and 6.12 show an example of the response of the fault detection and isolation approach when a 40% fault is introduced in sensor h_1 at time sample $t = 500$. Figure 6.10 shows the evolution of the residuals of each possible conflict for such a fault. Using these residuals as the input for the PCA model, the evolution of the T^2 and Q monitoring statistics is shown in figure 6.11. Looking at this figure, the monitoring statistics do not present problems related with the start-up and it was able to quickly and accurately detect the fault. Finally, figure 6.12 shows the plot of the contribution analysis for the T^2 statistic. In this case, residuals for possible conflicts PC_2, PC_3, and PC_4 have a high value and, consequently, can be considered as the cause of the deviation in the T^2 statistic. As the theoretical fault signature matrix shows in table 6.5, this system determines that the faulty component was the sensor h_1, because a fault in this sensor triggers the possible conflicts PC_2, PC_3, PC_4, and not PC_1 (a fault in sensor h_2 would have triggered PC_1 and not PC_2).

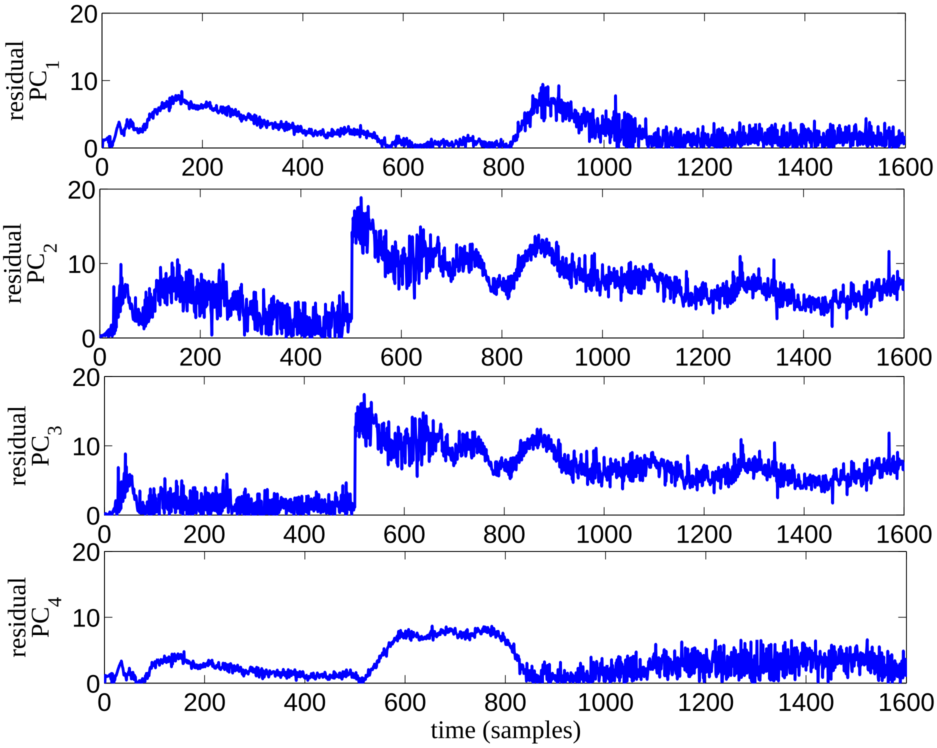


Figure 6.10: PCs residuals for a 40% fault in sensor h_1 at time $t = 500$.

6.5 Discussion

The PCA-based monitoring approaches show problems when changes in the working conditions and the operation modes occur. In this chapter, an approach where the PCA is integrated with a first principles model-based diagnosis approach to improve the overall diagnosis process is presented

This approach improves the classical PCA-based fault detection approach because the first principles model-based detection system

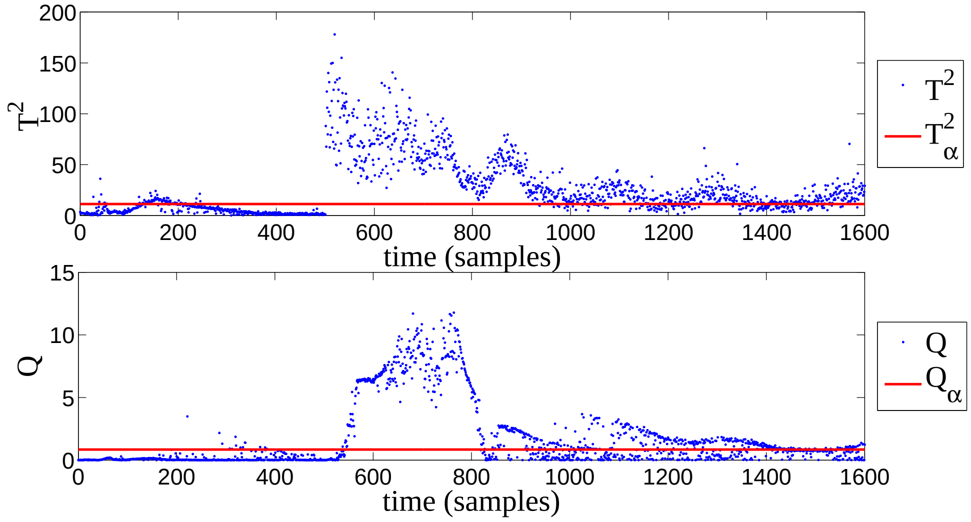


Figure 6.11: T^2 and Q statistics for the faulty experiment shown at Figure 6.10.

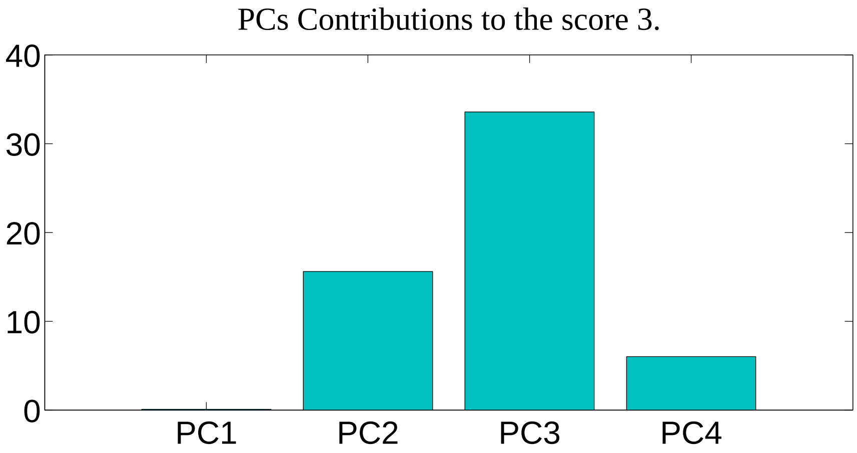


Figure 6.12: Contribution plot for the T^2 statistic for the faulty experiment shown at Figure 6.10.

works as a filter of changes between the different operating modes. In this work, the PCA model is performed using the residual obtained from the process in normal operation conditions. When a fault occurs, one or more of these residuals suffer a deviation that can be detected by the PCA-based detection system.

Also, the proposed integration improves the PCs-based fault diagnosis scheme because the statistical analysis of the residuals provides fault detection capabilities which are robust sufficiently to model uncertainties and noisy measurements.

Moreover, the PCs configurations allow fault isolation tasks to be performed without data from the different faults considered. It is not very common to have enough data from the different faults that can occur in a process.

Another improvement of the proposed approach is the possibility of performing a whole diagnosis task. The PCA approach provides poor fault isolation capabilities. When faults are detected, a contribution analysis is built to identify the variables responsible for the deviation. But this diagnosis procedure should be carefully interpreted as the variables are highly correlated, and variables that are not involved in the fault may have a high contribution value. Using the integration approach proposed in this chapter, this drawback disappears.

The main drawback of this configuration is that a first principles model is required. This drawback cannot be as complex as to perform PCA and UPCA models for each operating mode and transient states in the same cases. Processes without many operating modes can be addressed with PCA and UPCA approaches without a first principles model. However, in processes with many operating modes, this integration approach can be a better solution.

Chapter 7

Soft sensor design based on multivariate statistical techniques

7.1 Introduction

A soft sensor or virtual sensor is based on predictive models. These models are used to estimate process variables which can be determined at low sampling rates or through off-line analysis. These measurements are often related with the process output quality (Kadlec et al., 2009).

Two main different types of soft sensors can be established. On the one hand, the model-driven soft sensors which are usually based on first principle models. On the other hand, the data-driven soft sensor which use large amounts of data, measured and stored in industry, to build prediction models (Kadlec et al., 2009).

Dry substance content sensors are in general expensive and inaccurate, so it is interesting to study and develop soft sensors for this variable. For this purpose, four methods have been proposed in this chapter to design a dry substance (DS %) content sensor of the juice leaving the evaporation station of a real sugar factory. The first one is based on indirect measurements, using physicochemical properties.

The second one uses neural networks where the inputs to the net are selected manually, based on a correlation study of the variables of the evaporation station. The third one uses neural networks whose inputs are the scores calculated by means of principal component analysis. The last method uses an estimation based on partial least squares regression.

The first soft sensor, which is based on physicochemical properties, falls into the model-driven soft sensor class; whereas the other three designed soft sensors fall into the data-driven soft sensor class.

Soft sensors can be used in several configurations. They can be tuned using data from the laboratory and they can be connected to the process to provide an *on-line* estimation of the measured variable. It does not mean that the laboratory measurements are not required, but they can be performed less frequently. In the case of this work, there is an *on-line* analyser available. In this case, the use of soft sensors can be a step towards robustness because both measurements can be compared. This configuration detect faults or disturbances in the measurement to be detected. In fact, in the soft sensors based on the PCA and PLS considered in this work, a fault detection and isolation approach can be performed very easily using the statistical models established.

7.2 Case study: evaporation section of a sugar factory

The dry substance (DS) content is a key measurement in the sugar industry and other dairy industries, such as juice factories and concentrated or soft drinks. Knowing the dry substance content in the juices is fundamental for the control of the evaporation processes, because this measurement indicates whether the desired degree of concentration in the product has been achieved.

It is also important to know the dry substance content in crystallization processes because this measurement allows the supersaturation of the dissolution to be known, thus making it possible to control the process of crystal formation (Mc Ginnis, 1982).

The dry substance content unit, °Brix, is used frequently in sugar industries to represent the dry substance % m/m (DS), giving an idea, together with the purity, of the amount of sugar in the juice. However, Brix is not a scientific term and a dry substance content term should be used.

The dry substance content can be measured in laboratory by means of various methods. The refractometric method is the most commonly used due to its simplicity, low cost and speed of analysis. The procedure to determine the refractometric dry substance content (RDS %) is established in ICUMSA (2009). However, the *on-line* RDS % method, used with relatively high sugar concentrations, presents problems due to the fouling of the prism. For that reason, other methods are being used to determine the *on-line* dry substance content, such as density based methods, microwaves and infra-red waves. In these two last cases, the water content is measured, which gives an indication of the dry substance content. All these sensors can present a high cost. This can limit the widespread use of these sensors, which are acquired in a very restrained way and are located in critical points of the process.

The considered soft sensors of this work are designed to estimate the dry substance content at the outlet of the last evaporator in an evaporation section of a sugar factory. For this purpose, real data of the process, working under normal conditions, without anomalies in either the operation or the measurements, are collected from the plant. Figure 4.9 shows a schematic diagram of the evaporation section of a sugar industry.

Concretely, four sensors have been developed, using different techniques. The first is based on the estimation of the dry substance content as an indirect measurement, which is estimated from the vapour pressure and the juice temperature (Perry and Green, 1997). In the second case, the dry substance content is estimated using an artificial neural network (ANN) with all the measurements of the evaporation section, because neural networks can be used as a non-linear universal function approximator (Narendra and Parthasarathy, 1990; Hunt et al., 1992). The third method is based on the previous one, but, in order to simplify the structure and therefore the training cost of the net, the dry substance content is estimated from the latent variables of the process, calculated using the principal component analysis. The last method consists of the dry substance content estimation using all the evaporation process variables, but using a statistical multivariate regression method based on principal components: partial least squares (PLS).

7.3 Dry substance content estimation based on indirect measurements

The content of solute can be determined using the colligative properties of the dissolution in a thermodynamic system in equilibrium. Two of these properties are the boiling point elevation of the dissolution compared with the pure component and the decrease of the vapour pressure, both phenomena being caused by the presence of a solute. In ideal solutions, the decrease in the vapour pressure can be determined by Raoult's Law (Perry and Green, 1997). In this case study, a concentrated sugar solution is used, so it is not possible to consider an ideal solution. The variation in the vapour pressure is determined experimentally and can be calculated using the following equations (Bubník et al., 1995):

$$k_1 = -0.2 + e^{(-1.5254+0.022962 \cdot DS+2.163 \cdot 10^{-4} \cdot w_{DS}^2)} \tag{7.1}$$

$$k_2 = 0.9985 + 0.01 \cdot e^{(-3.2021+0.066743 \cdot DS+1.161 \cdot 10^{-4} \cdot w_{DS}^2)} \tag{7.2}$$

$$k_3 = 0.0001 \cdot e^{(-1.4276-0.024362 \cdot DS-6.047 \cdot 10^{-4} \cdot w_{DS}^2)} \tag{7.3}$$

$$T_s = \frac{-k_2 + \sqrt{k_2^2 - 4 \cdot k_3 \cdot (k_1 - T)}}{2 \cdot k_3} \tag{7.4}$$

$$P_V = 10^{\frac{2147}{T_s+273.2}+5.7545} \tag{7.5}$$

where P_V is the vapour pressure in bar, T is the solution temperature in K and w_{DS} are DS content percentages. T_s represents the boiling point temperature correction due to the presence of sugar, so $T_s - T$ is the solution boiling point elevation. k_1, k_2 and k_3 are correction terms that depend on the DS content. This equation establishes a relation between the juice temperature and the saturated steam pressure with the sugar concentration. All these variables are usually measured in the plant, so, solving the previous equation system in an implicit way, it is possible to calculate the DS content as a function of the juice temperature and the vapour pressure. It is important to remark that these equations are based on measurements of pure sucrose solutions, and while the juice in the plant is a technical sugar solution, it can result in a certain bias in the estimation.

It has to be taken into account that two other suppositions are made when using this method to determine the DS content. The first one is that it is supposed that the system is in thermodynamic equilibrium. This is not completely true; in real industrial processes, the conditions for thermodynamic equilibrium do not occur, so there will be a bias between the values predicted by the formulas and the process data.

The second hypothesis is that the measured vapour pressure is the pressure of the saturated steam in equilibrium with the juice. In fact, it is probable that the vapour is not saturated, it being possible that, depending on the position in which the sensor is situated, the vapour might be slightly overheated. In the plant used in this work, the sensors are located near the steam condensation chambers, so the assumption of considering saturated steam is correct. However, in plants where the pressure sensors are located near the liquid vapour interface, this consideration has to be taken into account.

One of the drawbacks of this method is that the determination of the dry substance content through the measurement of vapour pressure and juice temperature is very sensitive to changes in these two parameters. If this correlation is represented graphically, (Figure 7.1), a huge gradient of the function in the majority of its domain can be observed, which implies that small variations in the vapour pressure, or in the juice temperature, will generate abrupt changes in the dry substance content.

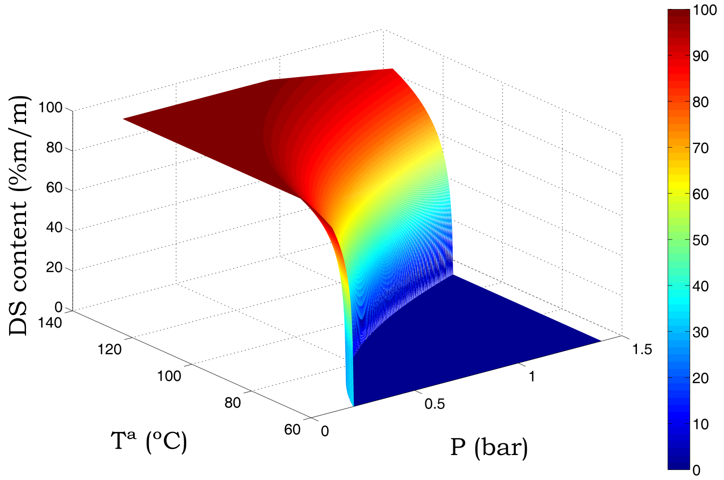


Figure 7.1: Relation vapour pressure – juice temperature – DS %.

In this case study, it has been determined that there exists a bias between the measurement of the pressure provided by the sensor and the real pressure. This offset can be adjusted by setting out an optimization problem:

$$\min_{\text{offset}} \sum_{i=1}^{n} \left(w_{DS_i} - (\hat{w}_{DS_i} + \text{offset})\right)^2 \tag{7.6}$$

where n is the number of samples collected, w_{DS} is the dry substance content measured from the plant, $\hat{w}_{DS}$ is the dry substance content estimated using the equations 7.1 - 7.5 and summing the offset.

The main drawback of this method, derived from the sensitivity described above, is the lack of robustness of the results with respect to deviations in the measurements.

In order to compare the robustness of the implemented methods, a deviation in the pressure of 40% is introduced manually into the system.

7.4 DS content estimation based on neural networks

Due to the main drawback of the previous method related with the sensitivity and the lack of robustness, a new configuration can be performed that takes into account a greater number of process variables for the estimation of the dry substance content, thus obtaining a model that can provide more robust estimates.

The second technique applied to soft sensor design is based on the use of an artificial neural network (ANN) (Narendra and Parthasarathy, 1990; Hunt et al., 1992) to estimate the dry substance content. The neural network used is an Elman neural network, which belongs to the simple recurrent network (SRN), which is an improved *feedforward* network, due to the inclusion of feedback from the output of hidden layers to the layer input of the network, as shown in figure 7.2. Other feedback ANN configurations, such NARX (Non-linear AutoRegresive with eXogenous Response) or NOE (Non-linear output error) could also be used. Besides, neural networks allow non-linearities between the measured variables and the dry substance content to be modelled, which are highly non-linear, as shown in the equations 7.1 - 7.5.

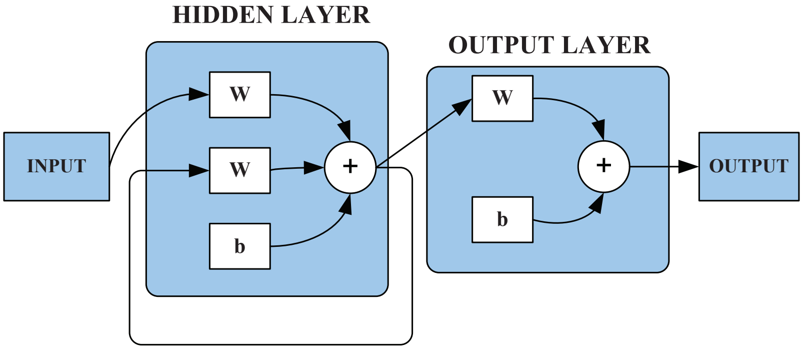


Figure 7.2: Elman Net Diagram.

The large amount of system data makes it difficult to decide on the best structure for the network: the number of inputs, the number of layers, the number of neurons in each layer; and to train the neural network; so a data preprocessing, which consists of a correlation analysis for the omission of those data that provide little information about the system, can be performed (Lynn et al., 2009). If two variables are quite well correlated, one of them brings little information to the training. For each pair of system data (x_i, x_j) the correlation is calculated as:

$$\phi_{x_i,x_j} = \frac{E(x_1 - \mu_{x_1})(x_2 - \mu_{x_2})}{\sigma_{x_1}\sigma_{x_2}} \tag{7.7}$$

where μ_{x_1}, μ_{x_2} and $\sigma_{x_1}, \sigma_{x_2}$ are the mean value and the standard deviation of the measured variables x_1, x_2 respectively. When the correlation between two signals is equal or close to $|1|$, this implies that both signals are correlated. Those variables that are strongly correlated are removed from the data set. For the specific case of the soft sensor, those variables that have a correlation higher than or equal to 0.99 are eliminated.

Once the number of variables has been reduced, the network can be trained to estimate the dry substance content. The variables measured may have different numerical ranges and, during the network training, the inputs with a higher range have a greater influence on the output, so it is advisable to normalize the values of the variables to zero mean and unit variance.

7.5 DS content estimation based on neural networks and principal component analysis

The solution to eliminate the variables that are strongly correlated can be a costly task and it is also possible to eliminate any variable from the neuronal model that, despite having a high correlation with another variable, is essential for the dry substance content estimation.

A solution that allows all the process variables to be used in order to ensure greater robustness in the estimation, and which presents no additional effort in training the neural network due to the large number of inputs, is to use the latent variables that best explain the behaviour of all the measured variables as inputs to the neural network (Choi and Heekyung, 2001). Using the neural network, the non-linearities between the latent variables of the system and the dry substance content measurement can be extracted.

One way to estimate the latent variables of the system is to use principal component analysis (PCA). The PCA model can be established and it is possible to calculate the *scores* of the training set.

The *scores* returned by the PCA model, once standardized, can be used as inputs for an Elman neural network that models the relationship between them and the dry substance content, as shown schematically in figure 7.3.

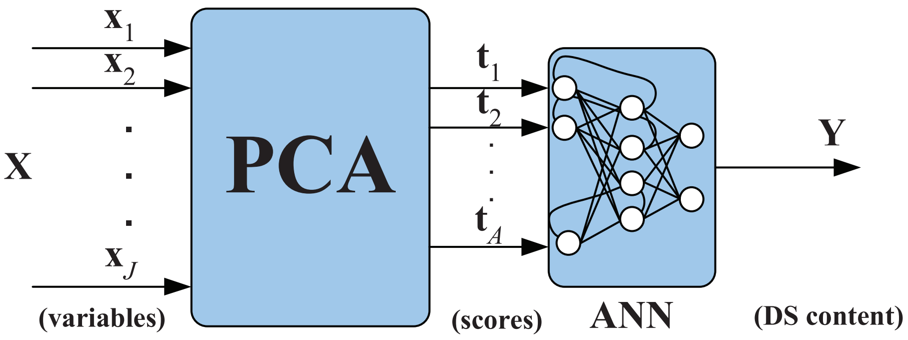


Figure 7.3: Schematic of the soft sensor structure with PCA and a neural network.

In this case, the number of inputs of the neural network will be as many as the latent variables. This means that the neural network can be simpler than the network used in the previous section even though all variables are considered.

The use of PCA presents the possibility of performing a multivariate statistical process control (MSPC) for monitoring the state of the plant, using the two explained control charts. When the statistics detect a special situation, the contribution plots can be used to find the root of the abnormal situation by identifying the variable that contributes to the abnormal situation.

7.6 DS content estimation based on partial least squares

One of the most widely used multivariate statistical techniques in industry, together with the PCA, is the partial least squares (PLS) regression (section 2.7). PLS can establish a projection structure that models the relation between a response matrix $\mathbf{Y} \in \Re^{K \times L}$ (the variable to be estimated by the soft sensor in this case) and the prediction matrix $\mathbf{X} \in \Re^{K \times J}$ where $\mathbf{X}$ matrix is arranged with the evaporation process variables and $\mathbf{Y}$ with the dry substance content measurements.

Using equation 2.27, it is possible to estimate the dry substance content as a function of the measured process variables.

The design of this approach can be performed using the NIPALS algorithm (*Non-linear Iterative Partial Least Squares*) for PLS.

As in the case of PCA, the measured variables may have different numerical ranges and the calculation of the PLS model depends on the variance of the variables and, at the same time, the variance depends on the numerical range, so it is necessary to normalize variables to zero mean value and unit variance.

Once the PLS model is generated, the T^2 and Q control charts (Höskuldsson, 1988) can be used to identify *outliers* and iteratively calculate more robust models. Also, as in the previous section, a multivariate statistical process control (MSPC) can be designed to detect abnormal behaviours in the process using the control charts and to find the fault root through the contribution analysis.

7.7 Soft sensor implementation

The real data used in this work were obtained using a supervisory control and data acquisition (SCADA) tool described in Alves et al. (2004) and Alves (2005). One of the features of this tool allows historical data from the plant to be recorded. The data were recorded with a sampling time of 10 seconds. The dry substance content sensor in the studied process is located at the juice outlet of the last evaporator of the evaporation section. The sensor used in the studied plant is a Micro Motion© by Emerson. This sensor is based on the Coriolis principle.

The implementation of the approaches presented in this work was performed in MATLAB© using different toolboxes. In this section, the main results of applying the different approaches are presented, explained and compared.

Table 7.1 shows all the variables of the evaporation section considered in this work. Principally, these variables are measurements of pressures, temperatures and flows.

7.7.1 DS content estimation based on indirect measurements

The results of applying the first soft sensor, i.e., the sensor based on the resolution of the equations system 7.1 - 7.5, are shown in figure 7.4. The pressure and temperature variables of these equations (T_s and P_V) are the variables numbered 6 and 49 of table 7.1. In the graphic, the dashed blue line represents the real data collected from the plant and the continuous red line is the result of the dry substance content calculation, solving equations 7.1 - 7.5 in an implicit way. It can be observed that, in a first approach, the obtained results do not match the real process measurements due to the cited bias in the estimation.

Figure 7.5 shows the results obtained with the bias correction method based on equation 7.6. The number of samples considered to calculate the optimal offset* was 15,000. When this bias is corrected, the obtained results are those shown in the upper part of figure 7.5.

The solution obtained is considerably better in this case, so this method can be suitable for dry substance content estimation, but taking care to use laboratory measurements periodically to correct any systematic errors in the plant measurements.

If the artificial deviation in the vapour pressure is introduced manually, the response of the estimation can be seen in the lower part of

#	Name	Description	Units
1	w_{DS_1} %	Dry substance content (evaporator 1)	$\% \ m/m$
2	P_1	Vapour pressure (evaporator 1)	bar
3	P_2	Vapour pressure (evaporator 2)	bar
4	P_3	Vapour pressure (evaporator 3)	bar
5	P_4	Vapour pressure (evaporator 4)	$bar\ abs$
6	P_5	Vapour pressure (evaporator 5)	$bar\ abs$
7	P_6	Vapour pressure (evaporator 6)	$mbar\ abs$
8	T_1	Juice temperature (evaporator 1 inlet)	$^{\circ}C$
9	T_2	Juice temperature (heat exchanger R10 inlet)	$^{\circ}C$
10	T_3	Juice temperature (heat exchanger R13 inlet)	$^{\circ}C$
11	T_4	Juice temperature (heat exchanger R14 inlet)	$^{\circ}C$
12	T_5	Juice temperature (heat exchanger R3 inlet)	$^{\circ}C$
13	T_6	Juice temperature (heat exchanger R3B inlet)	$^{\circ}C$
14	T_7	Juice temperature (heat exchanger R8 inlet)	$^{\circ}C$
15	T_8	Juice temperature (heat exchanger R9 inlet)	$^{\circ}C$
16	T_9	Juice temperature (thin juice tank outlet)	$^{\circ}C$
17	T_{10}	Juice temperature (heat exchanger R10 outlet)	$^{\circ}C$
18	T_{11}	Juice temperature (heat exchanger R11 outlet)	$^{\circ}C$
19	T_{12}	Juice temperature (heat exchanger R12 outlet)	$^{\circ}C$
20	T_{13}	Juice temperature (heat exchanger R3 outlet)	$^{\circ}C$
21	T_{14}	Juice temperature (heat exchanger R3A outlet)	$^{\circ}C$
22	T_{15}	Juice temperature (heat exchanger R3B outlet)	$^{\circ}C$
23	T_{16}	Juice temperature (heat exchanger R4 outlet)	$^{\circ}C$
24	T_{17}	Juice temperature (heat exchanger R5 outlet)	$^{\circ}C$
25	T_{18}	Juice temperature (heat exchanger R6 outlet)	$^{\circ}C$
26	T_{19}	Juice temperature (heat exchanger R7 outlet)	$^{\circ}C$
27	T_{20}	Juice temperature (heat exchanger R8 outlet)	$^{\circ}C$
28	T_{21}	Juice temperature (heat exchanger R9 outlet)	$^{\circ}C$
29	T_{22}	Vapour temperature (evaporator 1)	$^{\circ}C$
30	T_{23}	Vapour temperature (evaporator 2)	$^{\circ}C$
31	T_{24}	Vapour temperature (evaporator 3)	$^{\circ}C$
32	T_{25}	Vapour temperature (evaporator 4)	$^{\circ}C$
33	T_{26}	Vapour temperature (evaporator 5)	$^{\circ}C$
34	T_{27}	Vapour temperature (evaporator 6)	$^{\circ}C$
35	T_{28}	Steam temperature (Steam from boilers to evaporation)	$^{\circ}C$
36	W_1	Juice mass flow (to heat exchanger R10)	T/h
37	W_2	Juice mass flow (to heat exchanger R3)	T/h
38	W_3	Juice mass flow (to heat exchanger R3B)	T/h
39	W_4	Juice mass flow (to heat exchanger R4)	T/h
40	W_5	Juice mass flow (to heat exchanger R8)	T/h
41	W_6	Juice mass flow (to heat exchanger R9)	T/h
42	W_7	Juice mass flow (from thin juice tank)	T/h
43	W_8	Juice mass flow (evaporator 6 outlet)	T/h
44	W_9	Steam mass flow (Steam from boilers to evaporation)	T/h
45	T_{29}	Juice temperature (evaporator 1 outlet)	$^{\circ}C$
46	T_{30}	Juice temperature (evaporator 2 outlet)	$^{\circ}C$
47	T_{31}	Juice temperature (evaporator 3 outlet)	$^{\circ}C$
48	T_{32}	Juice temperature (evaporator 4 outlet)	$^{\circ}C$
49	T_{33}	Juice temperature (evaporator 5 outlet)	$^{\circ}C$
50	T_{34}	Juice temperature (evaporator 6 outlet)	$^{\circ}C$
51	T_{35}	Juice temperature (evaporator 1 outlet)	$^{\circ}C$
52	P_7	Steam pressure (Steam from boilers to evaporation)	bar

Table 7.1: Variables description

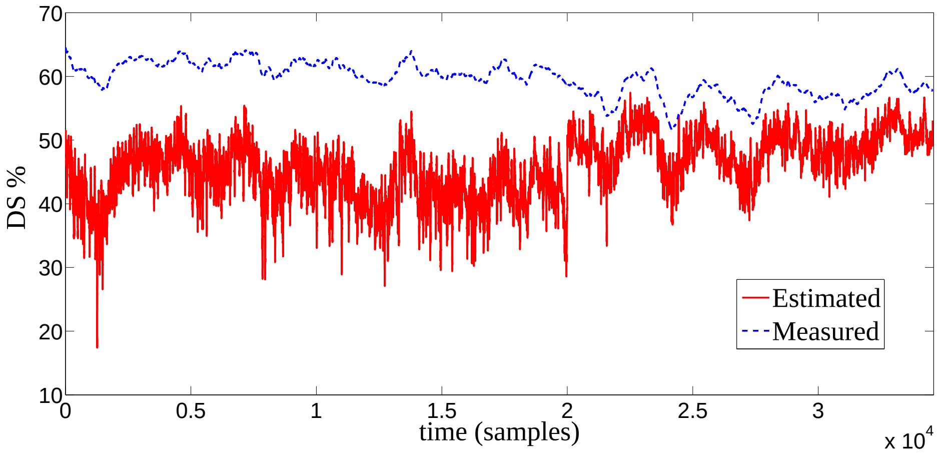


Figure 7.4: Dry substance content estimation from indirect measurements.

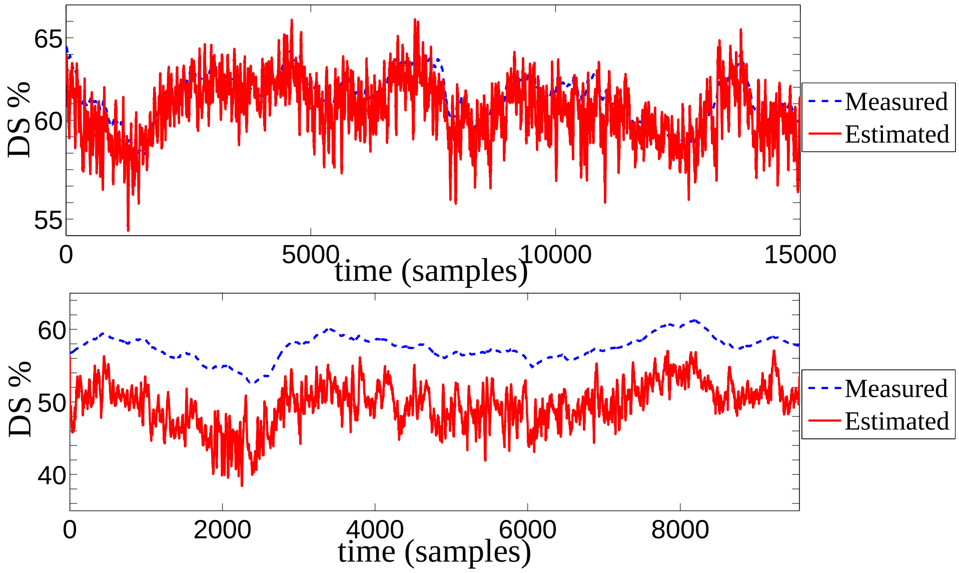


Figure 7.5: Dry substance content estimation from indirect measurements with bias correction.

figure 7.5. The estimated dry substance content is calculated with a great error, i.e., the estimation of the dry substance content obtained with this method presents a high sensitivity to disturbances.

7.7.2 DS content estimation based on neural networks

In the case of using the soft sensor based on neural networks, where the correlation study is applied, the number of variables of the system is reduced from 52 (table 7.1) to 39 measured variables, maintaining the characteristics of the system with the new data set.

This neural network was trained with 15,000 samples extracted from the plant for two days of normal operation with a sampling period of 10 seconds. The data set chosen for training was selected to have a great variability in the dry substance content, exploring different situations or states in the plant; another data set with 10,000 samples was used to test the neural network obtained.

The neural network was generated with the following structure: 39 nodes in the input layer, one hidden layer with 30 neurons and one neuron in the output layer.

Figure 7.6 shows the results for two different data sets. The dashed blue line represents the data collected from the plant and the continuous red line represents the data calculated by the model, estimated with the subset of data reserved for this purpose. The upper graph shows an estimate when none of the variables is modified, which means no disturbances were introduced in the measurements. In the lower graph, a bias on the pressure was introduced with the same characteristics as in the previous case. In both cases, it can be seen that the prediction follows the trend of the real data with an acceptable error for this type of measurement.

More robust responses to disturbances are achieved using this method. This is due to the neural network's property of being an approximator for non-linear functions, and to the fact that many more process variables can be used in the training that may also be related to the dry substance content measurement.

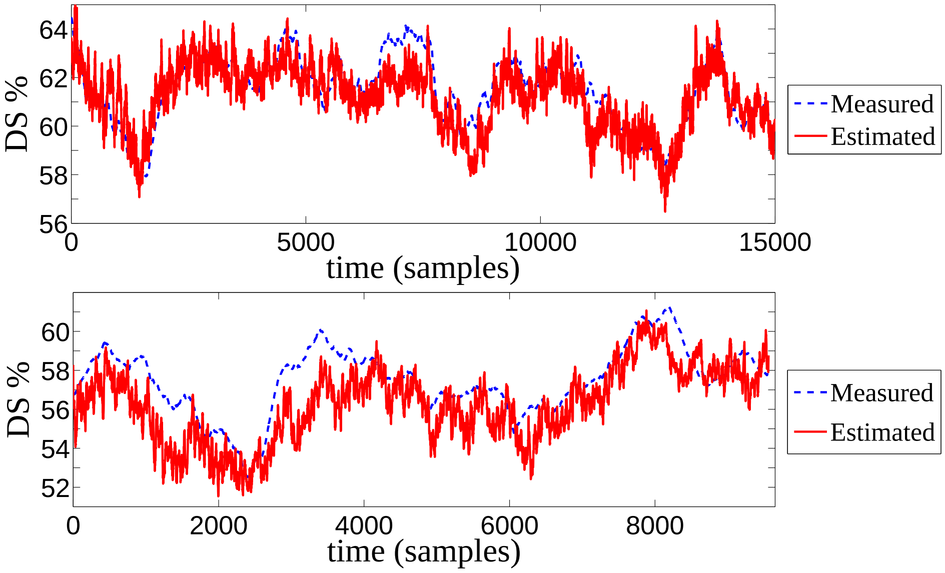


Figure 7.6: Results for the soft sensor based on a neural net, variables selected with a correlation study.

7.7.3 DS content estimation based on neural networks and PCA

If PCA is used, according to the third soft sensor presented, the number of latent variables selected was 21, following a cross-validation scheme.

The structure of the neural network consists of 21 inputs, as many as the number of latent variables, 30 neurons in the hidden layer and one output.

Figure 7.7 shows the response of the soft sensor using new data from the plant and the comparison with the real measurement of the dry substance content. As in the previous cases, the upper graph of the figure shows the estimation of dry substance content without any disturbance in the measured process variables. In the lower graph, the sensor response with a bias in the vapour pressure can be observed. The graphs show that the estimated measurement follows the trend of the real measurement, obtaining a result very similar to that achieved in the previous section with a network of considerably smaller size and adding a previous step not very computationally expensive.

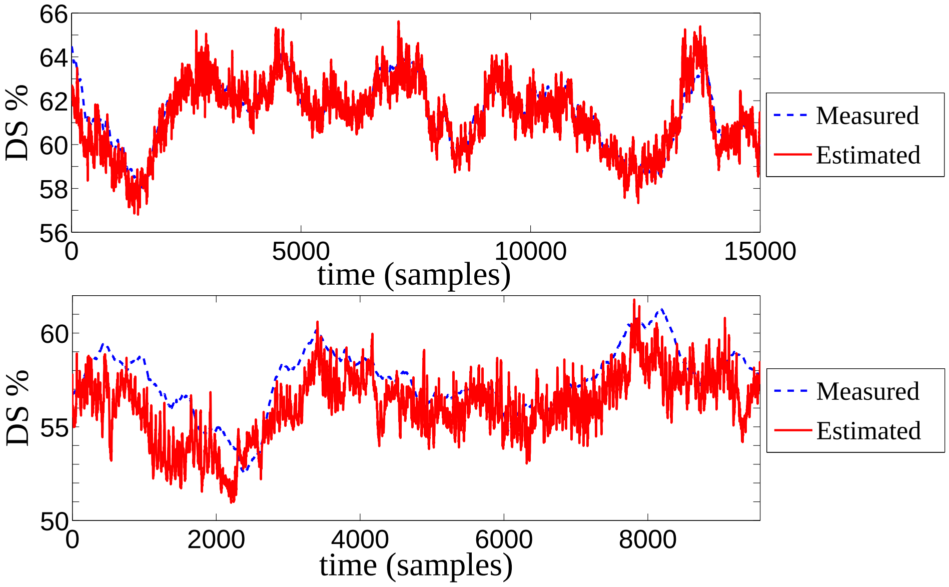


Figure 7.7: Results for the soft sensor based on a neural network and principal component analysis.

Figure 7.8 shows the monitoring of the evaporation section using Hotelling's statistic T^2 and the square prediction error Q statistic based on the PCA model performed. In this case, these control charts are used to detect the bias in the vapour pressure. Both statistics detect the offset introduced in one of the measurements. This offset was introduced at sample time 1000. In this example, both statistics detect the fault at the same time, so it is only necessary to look at the contributions plot of the normalized error of the variables.

Figure 7.9 shows the contributions plot of the normalized error of the variables after detection by the Q statistic. The plot shows that the principal variable related with the fault is the variable affected by the artificial offset. So, the operators could determine that this sensor presents an offset.

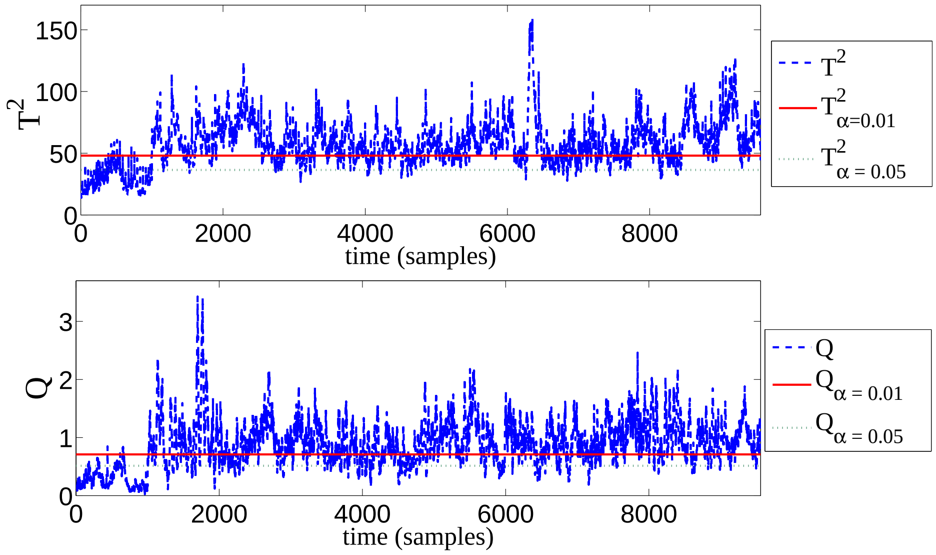


Figure 7.8: Monitoring of the evaporation section using Hotelling's statistic T^2 and the square prediction error Q statistic.

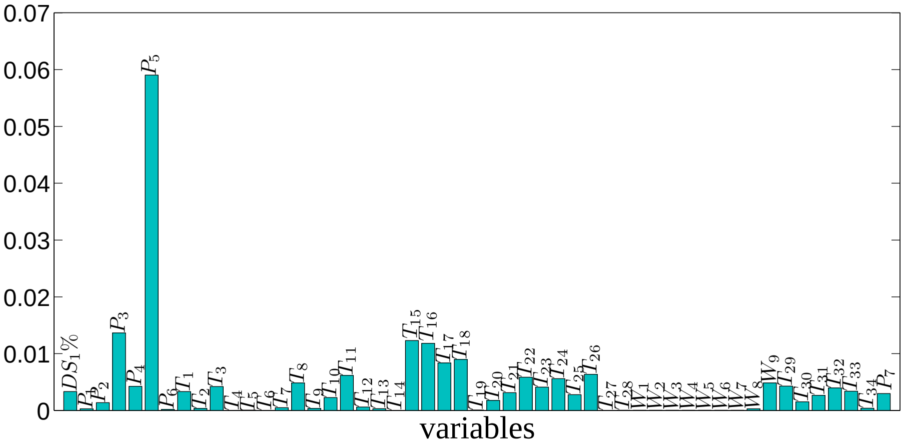


Figure 7.9: Contributions plot of the normalized error of the variables.

7.7.4 DS content estimation based on PLS

As cited before, the PLS method can be used to establish a regression model that relates the measured process variables with the DS content. In this scheme, all the variables of table 7.1 were selected to perform the PLS model. Taking all these variables into account, it is possible to achieve more stable approximations, since the measurement depends on the value of many more variables. If the relation between the first and the second loadings, $w^{*}c[1]/w^{*}c[2]$ (section 2.7), is represented in a graph (Figure 7.10), the variables which are most closely related to the model response or the variable estimated by the sensor can be observed. Within the dashed ellipse appearing in the figure, it can be seen that the dry substance content is highly related to more variables than the two that were used in the first case.

Figure 7.11 shows the results of the PLS estimation for two different data sets. The dashed blue line represents the real data collected from the plant and the continuous red line the data estimated by the model calculated with the subset of data reserved for this purpose. The upper graph shows the estimation when no variable is altered. In the lower graph, a bias was introduced on the vapour pressure with the same characteristics as in the previous cases. In both cases, the prediction follows the trend of real data with an acceptable error for these types of measurements.

7.8 Results

The different graphical results presented in previous sections for each of the solutions show that the method of indirect measurement is enhanced by methods based on neural networks and multivariate statistical techniques. This section gives the quantitative results of this improvement.

Table 7.2 shows the mean square error (MSE) between the real measured dry substance content and the estimated dry substance content without the bias in the pressure sensor, calculated using the four methods: indirect measurements (IM), neural networks (ANN), neural networks and PCA (ANN+ PCA) and PLS. In order to compare

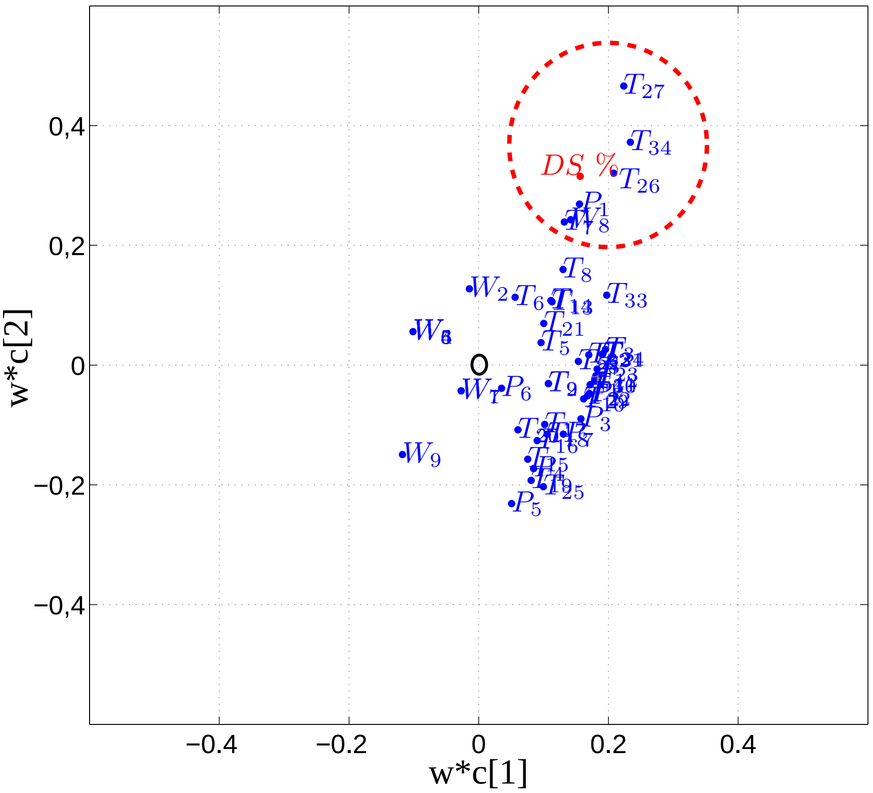


Figure 7.10: Graph of *loadings* $w^*c[1]/w^*c[2]$.

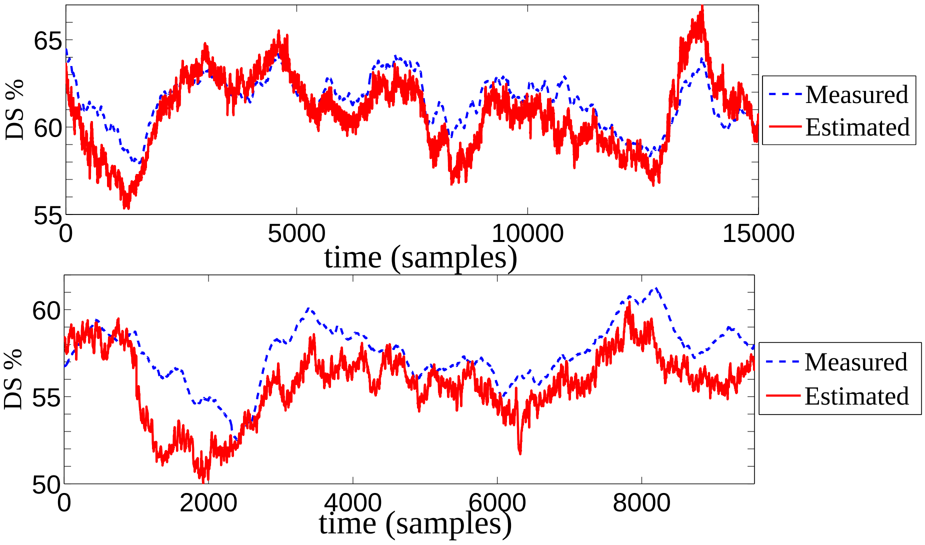


Figure 7.11: Results for the soft sensor based on PLS.

the results, 10 different cases of 1500 samples were considered.

The soft sensor based on the neural network and the PCA is the sensor with the best response results, while the biggest MSE is obtained with the sensor based on indirect measurements. After performing an analysis of variance, the *P-value* of the *F-ratio* is greater than 0.05, so there is no statistically significant difference between the means of the four methods at the 95.0% confidence level (results obtained using STATGRAPHICS Centurion XVI ©).

As shown in Table 7.3, methods based on neural networks and statistical techniques also improve the estimation of the dry substance content with respect to the indirect measurements method when a disturbance in the pressure measurement is introduced. For these approaches, the results are similar, although the third proposed method (ANN and PCA) presents the best result. The ANN without PCA and the PLS methods are remarkably robust, as they have properties of regression and approximation of functions. The indirect measurements method yields a significantly worse result due to the high sensitivity in the estimation of the dry substance content.

	IM	**ANN**	**ANN+PCA**	**PLS**
Case 1	4.15	4.67	3.18	5.69
Case 2	1.28	1.34	0.33	0.68
Case 3	1.20	0.89	0.47	0.59
Case 4	2.60	1.31	0.41	0.57
Case 5	2.38	2.03	0.26	1.74
Case 6	3.01	2.55	0.30	3.61
Case 7	3.13	1.46	0.38	1.71
Case 8	3.30	2.59	0.39	3.25
Case 9	1.59	1.38	0.65	1.50
Case 10	2.55	0.85	0.45	2.61
Mean	2.52	1.91	0.68	2.20
Std	0.94	1.15	0.88	1.63
ANOVA data				
F-ratio				2.79
P-value				0.541

Table 7.2: Soft sensor results without a bias in the pressure sensor.

In this case, after performing an analysis of variance, the *P-value* of the *F-ratio* is lower than 0.05, so there is a statistically significant difference between the means of the four methods at the 95.0% confidence level (results obtained using STATGRAPHICS Centurion XVI ©). After performing a multiple testing, a statistically significant difference between the indirect measurement method and each of the other methods was found, while no statistically significant difference between the ANN, ANN+PCA and PLS method was found. Two homogeneous groups could be distinguished: one formed by the IM method and another formed by the other three methods.

7.9 Discussion

Four approaches for the development of soft sensors to estimate the output dry substance content in an evaporation station of a sugar factory have been described in this chapter.

	IM	**ANN**	**ANN+PCA**	**PLS**
Case 1	57.09	3.68	0.78	0.67
Case 2	86.16	10.32	5.44	14.47
Case 3	85.14	6.22	2.79	4.31
Case 4	66.45	6.19	5.27	7.28
Case 5	62.26	1.57	3.31	2.11
Case 6	70.55	1.34	2.47	1.22
Case 7	64.02	1.62	2.30	3.14
Case 8	54.81	1.25	2.03	2.30
Case 9	50.43	4.67	5.95	5.61
Case 10	47.18	2.88	4.21	4.68
Mean	64.41	3.97	3.45	4.58
Std	13.28	2.94	1.70	4.03
ANOVA data				
F-ratio				178.81
P-value				0.00001

Table 7.3: Soft sensor results with a bias in the pressure sensor.

First, indirect measurements and thermodynamic properties have been used to estimate the dry substance content. This approach provides reasonable results if the sensors are properly calibrated and without systematic errors. The main problem of this method is that, due to the high sensitivity of the dry substance content to indirect measurements, it is necessary to make corrections in the process measurements. The main advantage of this method is that it does not need training or any model adjustment; it only needs occasional laboratory measurements to correct the possible deviations in the measurements.

Neural networks present better results than the PLS regression in this particular system. But the neural network estimation can be improved using the principal component analysis as a previous step in order to extract the latent variables of the measured data and simplify the structure of the ANN.

The data-based methods also produce good results in the estimation of the dry substance content, with the extra work of requiring a great amount of historical data for training and the advantage of being very robust to deviations in the measurements. Also, two of these three methods allow an MSPC or a fault detection system to be performed easily.

Chapter 8

Conclusions

In this dissertation, the monitoring, fault detection and estimation tasks in continuous processes using principal component analysis are studied. This chapter presents the main contributions provided by this dissertation, summarizes the main conclusions and finally discusses future directions for this work.

8.1 Summary of contributions

1. Study of the PCA-based methods for monitoring continuous processes. [García-Álvarez and Fuente (2008), García-Álvarez (2009) and García-Álvarez and Fuente (2011)]. In this first contribution, the PCA fault detection approach is considered. The classical PCA-based fault detection method has several limitations; for example, it is not able to detect consecutive faults and it is not suitable for monitoring non-linear behaviours such as transient states. Several PCA-based methods have been proposed to deal with these limitations. In this work, several of these methods have been studied, implemented and compared. This comparative study has been performed in a simulated two-tank-communicated system. This comparative study can be seen as a guide to select the most suitable method, depending on the type of process to be monitored. Also, this study reveals that

the long transient states and the start-ups in processes, which are non-linear, are not monitored correctly with many of the PCA-based methods presented.

2. Monitoring of transient states using the Unfolded PCA approach. [García-Álvarez et al. (2012b) and García-Álvarez et al. (2012c)]. This second contribution deals with the monitoring of transient states or start-ups. These states are non-linear and the PCA approach is based on a linear transformation. For this reason, the PCA technique is not the most suitable solution for monitoring this type of behaviour. In this case, a technique developed for batch processes, normally non-linear, is applied to these states. This method is known as UPCA. Several consideration must be taken into account to designing a UPCA-based monitoring tool such as the unfolding, synchronization or imputation problems. In this work, they are presented and explained. This part of the dissertation can be seen as a guide for design a UPCA-based method for transient states. The method is also applied to a real laboratory plant and to a simulated evaporation section of a sugar factory. In these examples, the method is applied considering different imputation methods and aims to find if the fault detection delay is related to the selected imputation method.

3. Monitoring of the whole behaviour of continuous processes taking into account the different operating modes and transient states. [García-Álvarez et al. (2009a) and García-Álvarez et al. (2010b)]. In this contribution, the main goal is to design PCA-based monitoring tools of the whole behaviour of a process including all the transient and steady states that can appear in the process operation. In this case, the proposal is to identify the different operating modes and transient states and to build a specific PCA or UPCA model for each of them. Therefore, a rich data base of past data under normal operation conditions of each of the states identified is required. This procedure must be performed *off-line*. Then, during the *on-line* monitoring phase, the fault detection method must identify the current state of the plant and the suitable PCA or UPCA model must be selected

to monitor each case.

4. Monitoring and fault detection of continuous processes without a strict steady behaviour. [García-Álvarez et al. (2011b), García-Álvarez et al. (2011d) and García-Álvarez and Fuente (2013)]. In this case, the UPCA method is applied to a continuous process that does not operate in a strict steady state. In this contribution, a reverse osmosis desalination process plant is studied. This type of plants requires cleaning phases in several of its components. The accumulation of deposits in the different filters and membranes and the required cleaning cycles are the reasons why the plant does not strictly run in a steady state. This is due to the noticeable differences in several of the pressure and concentration measurements when the plant has just been cleaned and when the plant has been cleaned a long time ago. The application of the classical PCA approach is not the most suitable method for these cases, but it is not very complicated to apply the UPCA method if a good data base of past cycles is available, obtaining better results than using the classical approach.

5. Combination of structural model decomposition techniques with the principal component analysis approach for fault detection and isolation. [Bregón et al. (2010)]. In this contribution, the combination of data-driven and model-driven techniques is used to monitor the whole behaviour of the process. A method based on the decomposition of the model is used to generate residuals. Concretely, the structural analysis technique of possible conflicts is used in this case. Using the residuals obtained with this approach, the isolation task can be performed and the root fault can be identified. Moreover, if the model is dynamic and accurate, the transient states can be modelled and the residuals do not detect any abnormal situation. Also, the residuals generated are taken as the input of a PCA-based fault detection method. This consideration improves and simplifies fault detection because the monitoring task is reduced to two control charts and PCA can determine significant deviations in the residuals. The integration method is applied to a real laboratory plant.

6. Design and improvement of soft sensors based on multivariate statistical techniques and neural networks. [García-Álvarez et al. (2012d) and Martí et al. (2011)]. In this case, several soft sensors have been designed for the estimation of the dry substance content in the evaporation section of a sugar factory. The most common method for the estimation of this type of measurements, based on physico-chemical principles, are very sensitive to perturbations. To deal with these drawbacks, a neural network-based sensor can be designed. This configuration can take into account more variables of the process and the estimations can be more robust. Moreover, the neural models implemented can be very complex when many inputs are considered. They can be simplified if a PCA computation is applied to the inputs, since PCA is able to represent the trends of the input data with a few latent variables. Another considered way to design a soft sensor for this real process is the use of the partial least squares (PLS). In this problem, the PLS approach can be applied as a regression method based on principal components. The application of multivariate statistical techniques improves the estimation error, generates more robust responses and is not very difficult to design fault detection and isolation schemes for the sensors related with the estimation.

Other works published during the developing of this dissertation are summarized in the following contributions.

7. Fault detection and diagnosis in wastewater treatment plants. Several configurations were applied in this case. On the one hand, PCA is applied to detect faults and the Fisher discriminant analysis (FDA) is applied to isolating faults [García-Álvarez et al. (2009c) and García-Álvarez et al. (2009b)]. Another application of FDA to a real plant is developed in Fuente et al. (2009). On the other hand, PCA is applied to monitor the residuals between the real process and a neural model. [Fuente et al. (2012)]

8. Monitoring of model predictive controllers using PCA. In this case, the behaviour of predictive controllers is monitored using a

PCA approach and contribution analysis is performed to identify possible malfunctions in the controller. [García-Álvarez et al. (2012a)].

9. Improvement of parameter estimation tasks using minimal parameter improvement with minimal analytically redundant subsystems analysis. [García-Álvarez et al. (2010a) and García-Álvarez et al. (2011a)]. Estimation of biokinetic parameters using several approaches [García-Álvarez et al. (2010c)].

10. Application of model-based fault detection and isolation approaches to several plants and other tasks related with this area. [Bregón et al. (2007), Bregón and García-Álvarez (2007), García-Álvarez and Bregón (2008) and García-Álvarez et al. (2011c)]

8.2 Conclusions

Based on the information, discussions and results presented throughout the work of this dissertation, the main conclusions can be established as follows:

- Classical PCA approaches are not suitable to develop fault detection and monitoring schemes in transient states and start-up stages. UPCA-based methods are most suitable.

- Continuous processes which do not operate in a strict steady state can be monitored using techniques based on UPCA approaches instead of classical PCA approaches.

- The integration of PCA-based and model-based techniques can be a good solution to improve and simplify the fault detection task and to design a complete fault isolation scheme.

- The use of multivariate statistical techniques can be used to improve the estimation task in soft sensors and to produce more robust measurements.

8.3 Future directions

Some research directions that could provide the next steps in monitoring, fault detection and isolation and estimation of continuous processes can be suggested as future paths of this dissertation. Also, some problems, which have not been undertaken in this work, can be studied in future research lines. These problems and paths define the future directions of this work:

- The UPCA approach for fault detection in transient states is studied in this dissertation. Concretely, the relation between the imputation method used and the fault detection delay is studied in detail. A future line of work could be to study the relation between the different unfolded methods proposed and the detection accuracy. Also, another indicator variable method can be applied, for example, in second order systems, an indicator variable based on the cumulative area between the reference and the process variable.

- Contribution analysis is a first step in fault diagnosis, but, as mentioned in this document, it cannot be considered as a complete fault isolation task. The fault isolation task could be a future line of this dissertation. The behaviour of the contributions plots when a fault is detected can be extracted using, for example, an artificial neural network. This ANN model can be used to distinguish between the different faults by looking at the contributions plots.

- Another proposed future line is related to process control. The PLS regression approaches can be used to build a statistical model. This model can be used to perform model predictive controllers (MPC). In this type of controller, it is very easy to develop fault detection techniques. Also, if an adaptive PLS approach is designed, the MPC can adapt to faults and a fault tolerant approach can be developed.

- In the combination of model-based techniques and multivariate statistical techniques proposal, several future lines can be considered. On the one hand, more faults can be diagnosed in this approach and more complex systems can be considered. On the other hand, an estimation scheme such as the PLS regression can be used to estimate variables that cannot be calculated using the model equations as they cannot be solved using mathematical causality.

- In this dissertation, the estimation task in soft sensors based on PLS is applied in the most simple way. In the same way as in the PCA approach, PLS can be performed in a dynamic configuration. This approach is known as dynamic PLS (DPLS). This configuration considers the relationships between the measured variables and the response variable at the current sample and past samples. Also, the proposed soft sensors based on neural networks can be improved using other ANN configurations, such as NARX (Non-linear AutoRegressive with eXogenous Response) or NOE (Non-linear Output Error) instead of Elman neural networks.

Appendix A

NIPALS algorithm for PCA

```
function [X,P,T,E,Lambda] = NIPALS(X)
[n,m] = size(X);
A = rank(X);
E = X;
b = 0;
P = [];
T = [];
lambda = [];
for i=1:1:A
    e = 1;
    random_num = ceil(m*rand);
    t = E(:,random_num);
    while (e>1e-007)
        p = (t'*E)/(t'*t);
        p = p';
        p = p/sqrt(p'*p);
        b = (E*p)/(p'*p);
        r = b-t;
        e = r'*r;
        t = b;
    end
    P = [P p];
    T = [T t];
    lambda = [lambda; t'*t];
    E = E - t*p';
end
lambda = lambda./(n-1);
for i=1:1:length(lambda);
    Lambda(i,i) = lambda(i);
end
```

Bibliography

Aguado, D., Ferrer, A., Ferrer, J., and Seco, A. (2007). Multivariate SPC of a sequencing batch reactor for wastewater treatment. *Chemometrics and Intelligent Laboratory Systems*, 85(1):82–93.

Alves, R. (2005). *Entornos distribuidos para simuladores de procesos.* PhD thesis, Universidad de Valladolid.

Alves, R., Normey-Rico, J. E., Merino, A., Acebes, L., and Prada, C. (2008). Distributed continuous process simulation: An industrial case study. *Computers & Chemical Engineering*, 32(6):1195–1205.

Alves, R., Normey-Rico, J. E., Merino, M., and Prada, C. (2004). Um SCADA com acesso a dados via OPC aplicado a uma planta piloto. *C&I Controle & Instrumentação*, ano 10 n^{o} 94:55–59.

Arteaga, F. and Ferrer, A. (2002). Dealing with missing data in MSPC: several methods, different interpretations, some examples. *Journal of Chemometrics*, 16:408–418.

Arteaga, F. and Ferrer, A. (2005). Framework for regression-based missing data imputation methods in on-line MSPC. *Journal of Chemometrics*, 19:439–447.

Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. (2003). *Diagnosis and Fault Tolerant Control.* Springer.

Bregón, A. (2010). *Integration of FDI and DX techniques within consistency-based diagnosis with possible conflicts.* PhD thesis, Universidad de Valladolid.

Bregón, A. and García-Álvarez, D. (2007). A reconfigurable system for multiple off-line simulations supporting fault diagnosis tasks. In *Proceedings of the International Forum Information Systems. Problems, Perspectives, Innovation Approaches.*, Saint Petersburg, Russia.

Bregón, A., García-Álvarez, D., Prieto, O., Pulido, B., and Alonso, C. (2007). Sistema reconfigurable para la simulación en bloque de experimentos en un entorno de supervisión y diagnosis. In *Proceedings of the Spanish Conference of Computing (CEDI)*, Zaragoza, Spain.

Bregón, A., García-Álvarez, D., Pulido, B., and Fuente, M. J. (2010). Combination of analytical and statistical models for dynamic systems fault diagnosis. In *Proceedings of the Conference of the Prognostics and Health Management Society (PHM)*, Portland, USA.

Bro, R., Kjeldahl, K., Smilde, A. K., and Kiers, H. A. (2008). Cross-validation of component models: a critical look at current methods. *Analytical and bioanalytical chemistry*, 390(5):1241–51.

Bubník, Z., Kadlec, P., Urban, D., and Bruhns, M. (1995). *Sugar technologists manual: Chemical and physical data for sugar manufacturers and users.* Bartens, Berlin.

Camacho, J. (2007). *New methods based on the projection to latent structures for monitoring, prediction and optimization of batch processes.* PhD thesis, Universidad Politécnica de Valencia. Spain.

Camacho, J., Picó, J., and Ferrer, A. (2009). The best approaches in the on-line monitoring of batch processes based on PCA: does the modelling structure matter?. *Analytica chimica acta*, 642(1-2):59–68.

Camacho, J., Picó, J., and Ferrer, A. (2010). Cross-validation in principal component analysis: searching for the 'best' approach. In *Chemometrics in Analytical Chemistry*, Antwerp (Belgium).

Chiang, L. H., Russell, E. L., and Braatz, R. D. (2001). *Fault Detection and Diagnosis in Industrial Systems.* Springer, Nueva York.

Choi, D. J. and Heekyung, P. (2001). A hybrid artificial neural network as a software sensor for optimal control of a wastewater treatment process. *Water Research*, 35(16):3959–3967.

Choi, S. W., Lee, C., Lee, J. M., Park, J. H., and Lee, I. B. (2005). Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 75(1):55–67.

Cundari, T. R., Sârbu, C., and Pop, H. F. (2002). Robust fuzzy principal component analysis (FPCA). A comparative study concerning interaction of carbon-hydrogen bonds with molybdenum-oxo bonds. *Journal of chemical information and computer sciences*, 42(6):1363–9.

Detroja, K. P., Gudi, R. D., and Patwardhan, S. C. (2007). Plant-wide detection and diagnosis using correspondence analysis. *Control Engineering Practice*, 15(12):1468–1483.

Dong, D. and McAvoy, T. J. (1996). Nonlinear principal component analysis based on principal curves and neural networks. *Computers & Chemical Engineering*, 20(1):65–78.

Duchesne, C., Kourti, T., and Macgregor, J. F. (2002). Multivariate SPC for start-ups and grade transitions. *AIChE Journal*, 48(12):2890–2901.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Clasiffication.* Wiley, New York.

Eastment, H. T. and Krzanowski, W. J. (1982). Cross-validatory choice of the number of components from a principal component analysis. *Technometrics*, 24(1):73–77.

Ferrer, A. (2007). 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.

Ferrer, A., Aguado, D., Vidal-Puig, S., Prats, J. M., and Zarzo, M. (2008). PLS: A versatile tool for industrial process improvement and optimization. *Applied Stochastic Models in Business and Industry*, 24(6):551–567.

Fourie, S. H. and de Vaal, P. (2000). Advanced process monitoring using an on-line non-linear multiscale principal component analysis methodology. *Computers & Chemical Engineering*, 24(2):755–760.

Fuente, M. J., García, G., and Sainz, G. I. (2008). Fault diagnosis in a plant using Fisher discriminant analysis. In *Proceedings of the Mediterranean Conference on Control and Automation Congress Centre*, Ajaccio, France.

Fuente, M. J., García-Álvarez, D., Sainz, G. I., and Vega, P. (2012). Fault detection in a wastewater treatment plant based on neural networks and PCA. In *Proceedings of the Mediterranean Conference on Control and Automation (MED)*, Barcelona, Spain.

Fuente, M. J., García-Álvarez, D., Sainz, G. I., and Villegas, T. (2009). Detection and identification method based on multivariate statistical techniques. In *Proceedings of the International Conference on Emerging Technologies and Factory Automation (ETFA)*, Mallorca, Spain.

García-Álvarez, D. (2009). Fault detection using principal component analysis (PCA) in a Wastewater Treatment Plant (WWTP). In *Proceedings of the International Forum Information Systems. Problems, Perspectives, Innovation Approaches.*, Saint Petersburg, Russia.

García-Álvarez, D. and Bregón, A. (2008). Design of a model-based diagnosis system for a three-tank plant using possible conflicts. In *Proceedings of the International Forum Information Systems. Problems, Perspectives, Innovation Approaches.*, Saint Petersburg, Russia.

García-Álvarez, D., Bregón, A., Fuente, M. J., and Pulido, B. (2010a). Comparativa de técnicas para la estimación de parámetros en modelos útiles para tareas de detección y diagnóstico. In *Proceedings of the Spanish Conference of Automation*, Jaén, Spain.

García-Álvarez, D., Bregón, A., Fuente, M. J., and Pulido, B. (2011a). Improving parameter estimation using minimal analytically redundant subsystems. In *Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, Orlando, USA.

García-Álvarez, D., Francisco, M., Fuente, M. J., and Vega, P. (2012a). Monitorización de controladores predictivos. In *Proceedings of the Spanish Conference of Automation*, Vigo, Spain.

García-Álvarez, D. and Fuente, M. J. (2008). Análisis comparativo de técnicas de detección de fallos utilizando análisis de componentes principales (PCA). In *Proceedings of the Spanish Conference of Automation*, Huelva, Spain.

García-Álvarez, D. and Fuente, M. J. (2011). Estudio comparativo de técnicas de detección de fallos basadas en el análisis de componentes principales (PCA). *Revista Iberoamericana de Automática e Informática Industrial (RIAII)*, 8(3):182–195.

García-Álvarez, D. and Fuente, M. J. (2013). A PCA-based monitoring and fault detection approach for reverse osmosis desalination plant. *Desalination and Water Treatment*, In presss.

García-Álvarez, D., Fuente, M. J., and Palacín, L. G. (2011b). Monitoring and fault detection in a reverse osmosis plant using principal component analysis. In *Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, Orlando, USA.

García-Álvarez, D., Fuente, M. J., and Sainz, G. I. (2010b). Monitoring and fault detection in processes with multiple operating modes, transitory phases and start-ups using principal component analysis. In *Proceedings of the International Conference on Emerging Technologies and Factory Automation (ETFA)*, Bilbao, Spain.

García-Álvarez, D., Fuente, M. J., and Sainz, G. I. (2011c). Design of residuals in a model-based fault detection and isolation system using statistical process control techniques. In *Proceedings of the International Conference on Emerging Technologies and Factory Automation (ETFA)*, Toulouse, France.

García-Álvarez, D., Fuente, M. J., and Sainz, G. I. (2012b). Fault detection and isolation in transient states using principal component analysis. *Journal of Process Control*, 22(3):551–563.

García-Álvarez, D., Fuente, M. J., and Vega, P. (2009a). Fault detection in processes with multiple operation modes using Switch-PCA and analysis of grade transitions. In *Proceedings of the European Control Conference (ECC)*, Budapest, Hungary.

García-Álvarez, D., Fuente, M. J., Vega, P., and Sainz, G. I. (2009b). Detección y diagnóstico de fallos en una EDAR mediante técnicas estadísticas multivariantes. In *Proceedings of the Spanish Conference of Automation*, Valladolid, Spain.

García-Álvarez, D., Fuente, M. J., Vega, P., and Sainz, G. I. (2009c). Fault detection and diagnosis using multivariate statistical techniques in a wastewater treatment plant. In *Proceedings of Symposium on Advanced Control of Chemical Processes (ADCHEM)*, Istanbul, Turkey.

García-Álvarez, D., Merino, A., and Fuente, M. J. (2012c). Monitorización de estados transitorios mediante el análisis de componentes principales. In *Proceedings of the Spanish Conference of Automation*, Vigo, Spain.

García-Álvarez, D., Merino, A., Martí, R., and Fuente, M. J. (2012d). Soft sensor design for dry substance content estimation in the sugar industry. *Zuckerindustrie*, 137(62):645–653.

García-Álvarez, D., Palacín, L. G., González-Benito, G., and Coca, M. (2010c). Estimación de parámetros biocinéticos utilizando diferentes herramientas de cálculo en el máster en investigación en ingeniería de procesos y sistemas. In *Proceedings of the Spanish Conference of Automation*, Jaén, Spain.

García-Álvarez, D., Palacín, L. G., Tadeo, F., Fuente, M. J., Salazar, J., and Prada, C. (2011d). Control of a desalination plant, including U-PCA-based monitoring. In *Proceedings of the International Conference on Environmental Science and Technology (CEST)*, Rhodes Islands, Greece.

Geladi, P. and Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185:1–17.

Gertler, J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, Inc.

Gertler, J., Li, W., Huang, Y., and McAvoy, T. (1999). Isolation enhanced principal component analysis. *AIChE Journal*, 45(2):323–334.

González-Martínez, J. M., Ferrer, A., and Westerhuis, J. A. (2011). Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping. *Chemometrics and Intelligent Laboratory Systems*, 105(2):195–206.

He, Q. P., Qin, S. J., and Wang, J. (2005). A new fault diagnosis method using fault directions in Fisher discriminant analysis. *AIChE Journal*, 51(2):555–571.

Helland, I. S. (2001). Some theoretical aspects of partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 58(2):97–107.

Heo, G., Gader, P., and Frigui, H. (2009). RKF-PCA: robust kernel fuzzy PCA. *Neural networks*, 22(5-6):642–650.

Höskuldsson, A. (1988). PLS regression methods. *Journal of Chemometrics*, 2(3):211–228.

Huang, Y., Gertler, J., and McAvoy, T. J. (2000). Sensor and actuator fault isolation by structured partial pca with nonlinear extensions. *Journal of Process Control*, 10(5):459–469.

Hunt, K. J., Sbarbaro, D., Żbikowski, R., and Gawthrop, P. J. (1992). Neural networks for control systems – A survey. *Automatica*, 28(6):1083–1112.

Hunter, J. S. (1986). The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210.

Hwang, D. H. and Han, C. (1999). Real-time monitoring for a process with multiple operating modes. *Control Engineering Practice*, 7(7):891–902.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis.* John Wisley and Sons, Inc.

ICUMSA, editor (2009). *ICUMSA methods book.* England.

Jackson, J. E. (2003). *A user's guide to principal components.* Wiley.

Jackson, J. E. and Mudholkar, G. S. (1979). Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349.

Kadlec, P., Gabrys, B., and Strandt, S. (2009). Data-driven Soft Sensors in the process industry. *Computers & Chemical Engineering*, 33(4):795–814.

Kano, M., Hasebe, S., Hashimoto, I., and Ohno, H. (2004). Evolution of multivariable statistical process control: aplication or independent component analysis and external analysis. *Computers & Chemical Engineering*, 28(6-7):1157–1166.

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

Kiers, H. (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3):105–122.

Kourti, T. (2003a). Abnormal situation detection, three-way data and projection methods; robust data archiving and modeling for industrial applications. *Annual Reviews in Control*, 27(2):131–139.

Kourti, T. (2003b). Multivariable dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *Journal of Chemometrics*, 17(1):93–109.

Kourti, T. (2005). Application of latent variable methods to process control and multivariable statistical process control in industry. *International journal of adaptive control and signal processing*, 19(4):213–246.

Kourti, T. and MacGregor, J. F. (1996). Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, 28(4):409–428.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural network. *AIChE Journal*, 37(2):233–243.

Kramer, M. A. (1992). Autoassociative neural networks. *Computers & Chemical Engineering*, 16(4):313–328.

Ku, W., Storer, R. H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 30(1):179–196.

Lane, S., Martin, E. B., Morris, A. J., and Gower, P. (2003). Application of exponentially weighted principal component analysis for the monitoring of a polymer film manufacturing process. *Transactions of the Institute of Measurement and Control*, 25(1):17–35.

Lee, G., Tosukhowong, T., and Lee, J. H. (2006). Fault detection and diagnosis of pulp mill process. *Computer Aided Chemical Engineering*, 21:1461–1466.

Lee, J. M., Yoob, C. K., and Lee, I. B. (2003). Statistical process monitoring with independent component analysis. *Journal of process control*, 14(5):467–485.

Lee, J. M., Yoob, C. K., and Lee, I. B. (2004). Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chemical Engineering Science*, 59(14):2995–3006.

Lennox, B., Montague, G. A., Hiden, H. G., Kornfeld, G., and Goulding, P. R. (2001). Process monitoring of an industrial fed-batch fermentation. *Biotechnology and bioengineering*, 74(2):125–135.

Li, W., Yue, H. H., Valle-Cervantes, S., and Qin, S. J. (2000). Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10:471–486.

Luukka, P. (2011). A new nonlinear fuzzy robust PCA algorithm and similarity classifier in classification of medical data sets. *International Journal of Fuzzy Systems*, 13(3):153–162.

Lynn, S., Ringwood, J., Ragnoli, E., McLoone, S., and MacGearailty, N. (2009). Virtual metrology for plasma etch using tool variables. In *Advanced Semiconductor Manufacturing Conference*.

MacGregor, J. F. and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403–414.

Martí, R., García-Álvarez, D., Merino, A., and Fuente, M. J. (2011). Diseño de un sensor soft para la estimación del Brix en la industria azucarera. In *Proceedings of the Spanish Conference of Automation*, Sevilla, Spain.

Mc Ginnis, R. (1982). *Beet sugar technology*. Beet Sugar Development Foundation, Colorado, USA.

Merino, A. (2008). *Librería de modelos del cuarto de remolacha de una industria azucarera para un simulador de entrenamiento de operarios*. PhD thesis, Universidad de Valladolid.

Merino, A., Alves, R., and Acebes, L. F. (2005). A training simulator for the evaporation section of a beet sugar production process. *The 2005 European Simulation and Modelling*, (1).

Misra, M., Yue, H. H., Qin, S. J., and Ling, C. (2002). Multivariable process monitoring and fault diagnosis by multi-scale PCA. *Computers & Chemical Engineering*, 26(9):1281–1293.

Narendra, K. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE transactions on neural networks*, 1(1):4–27.

Nomikos, P. (1996). Detection and diagnosis of abnormal batch operations based on multi-way principal component analysis. *ISA Transactions*, 35(3):259–266.

Nomikos, P. and MacGregor, J. (1995). Multivariable SPC Charts for Monitoring Batch Processes. *Technometrics*, 37(1):41–59.

Odiowei, P. P. and Cao, Y. (2010). State-space independent component analysis for nonlinear dynamic process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 103(1):59–65.

Page, E. (1954). Continuous inspection scheme. *Biometrika*, 41(1/2):100–115.

Palacín, L. G., Tadeo, F., Salazar, J., and de Prada, C. (2011). Operation of desalination plants using renewable energies and hybrid control. *Desalination and Water Treatment*, 25(1-3):119–126.

Peña, D. (2002). *Análisis multivariante de datos*. McGraw-Hill.

Perry, R. and Green, D. (1997). *Perry's Chemical Engineers' Handbook*. McGraw-Hill.

Prats-Montalbán, J. M., Ferrer, A., Malo, J. L., and Gorbeña, J. (2006). A comparison of different discriminant analysis techniques in a steel industry welding process. *Chemometrics and Intelligent Laboratory Systems*, 80(1):109–119.

Puigjaner, L., Ollero, P., Prada, C., and Jiménez, L. (2006). *Estrategias de modelado, simulación y optimización de procesos químicos.* Editorial Sintesis.

Pulido, B., Alonso, C., and Acebes, L. F. (2001). Lessons learned from diagnosing dynamic systems using possible conflicts and quantitative models. In *Engineering of Intelligent Systems. XIV Conf. IEA/AIE-2001*, Budapest, Hungary.

Pulido, B. and Alonso-González, C. (2004). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, 34(5):2192–2206.

Ramaker, H. J., van Sprang, E. N. M., Westerhuis, J. A., Gurden, S. P., Smilde, A. K., and van der Meulen, F. H. (2006). Performance assessment and improvement of control charts for statistical batch process monitoring. *Statistica Neerlandica*, 60(3):339–360.

Reiter, R. (1987). A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32(1):57–95.

Russell, E. L., Chiang, L. H., and Braatz, R. D. (2000). Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 51(1):81–93.

Sârbu, C. and Pop, H. F. (2005). Principal component analysis versus fuzzy principal component analysis. A case study: the quality of danube water (1985-1996). *Talanta*, 65(5):1215–20.

Schölkopf, B., Smola, A., and Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319.

Shewhart, W. A. (1938). Application of statistical methods to manufacturing problems. *Journal of the Franklin Institute*, 226(2):163–186.

Shlens, J. (2005). *A Tutorial on Principal Component Analysis.* Center for Neural Scienc, New York University.

Stanimirova, I., Walczak, B., Massart, D. L., and Simeonov, V. (2004). A comparison between two robust PCA algorithms. *Chemometrics and Intelligent Laboratory Systems*, 71(1):83–95.

Tan, S. and Mavrovouniotis, M. L. (1995). Reducing data dimensionality through optimizing neural network inputs. *AIChE Journal*, 41(6):1471–1480.

Tien, D. X., Lim, K. W., and Jun, L. (2004). Compartive study of pca approaches in process monitoring and fault detection. *The 30th annual conference of the IEEE industrial electronics society.*

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K. (2003a). A review of process fault detection and diagnosis. Part III: Process history based methods. *Computers & Chemical Engineering*, 27(3):327–346.

Venkatasubramanian, V., Rengaswamy, R., and Surya, N. (2003b). A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27(3):313–326.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N. (2003c). A review of process fault detection and diagnosis. Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3):291–311.

Vilar, J. F. (2005). *Control Estadístico de los Procesos.* Fundación Confemetal.

Villegas, T., Fuente, M. J., and Sainz-Palmero, G. I. (2010). Fault diagnosis in a wastewater treatment plant using dynamic Independent Component Analysis. *18th Mediterranean Conference on Control and Automation, MED'10*, pages 874–879.

Villez, K., Steppe, K., and De Pauw, D. J. (2009). Use of Unfold PCA for on-line plant stress monitoring and sensor failure detection. *Biosystems Engineering*, 103(1):23–34.

Weighell, M., Martin, E. B., and Morris, A. J. (2001). The statistical monitoring of a complex manufacturing process. *Journal of Applied Statistics*, 28(3-4):409–425.

Westerhuis, J. A., Kourti, T., and MacGregor, J. F. (1999). Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics*, 13(3-4):397–413.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Wold, S., Kettaneh, N., Fridén, H., and Holmberg, A. (1998). Modelling and diagnostics of batch processes and analogous kinetic experiments. *Chemometrics and intelligent laboratory systems*, 44(1-2):331–340.

Wold, S., Sjöström, M., and Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130.

Woodward, R. and Goldsmith, P. (1964). *Cumulative sum techniques, Mathematical and Statistical Techniques for Industry.* Oliver and Boyd, Edinburgh.

Zarzo, M. (2004). *Aplicación de Técnicas Estadísticas Multivariantes al Control de la Calidad de Procesos por Lotes.* PhD thesis, Universidad Politécnica de Valencia.

Zarzo, M. and Ferrer, A. (2004). Batch process diagnosis: PLS with variable selection versus block-wise PCR. *Chemometrics and Intelligent Laboratory Systems*, 73(1):15–27.

Zhang, Y., Dudzic, M., and Vaculik, V. (2003). Integrated monitoring solution to start-up and run-time operations for continuous casting. *Annual Reviews in Control*, 27(2):141–149.

Zhang, Y. and Dudzic, M. S. (2006). Online monitoring of steel casting processes using multivariate statistical technologies: From continuous to transitional operations. *Journal of Process Control*, 16(8):819–829.

Zhao, C., Wang, F., Lu, N., and Jia, M. (2007). Stage-based soft-transition multiple PCA modeling and on-line monitoring strategy for batch processes. *Journal of Process Control*, 17(9):728–741.

Zumoffen, D. and Basualdo, M. (2007). From large chemical plant data to fault diagnosis integrated to decentralized fault tolerant control: pulp mill process application. *Industrial & Engineering Chemistry Research*, 47(4):1201–1220.