



Universidad de Valladolid
Departamento de Ingeniería
de Sistemas y Automática

Métodos Distribuidos De Detección E Identificación
De Fallos Basados En Análisis De Componentes
Principales (PCA)

Trabajo de Fin de Máster

Master en Investigación en Ingeniería de Procesos y Sistemas
Industriales.

Alumno: Álvar Sánchez Fernández

Tutor: M^a Jesús de la Fuente Aparicio

Julio de 2015

Índice general

1. Introducción y objetivos	1
1.1. Introducción	1
1.1.1. Métodos de detección y diagnóstico de fallos	2
1.2. Objetivos	6
1.3. Organización de la memoria	6
2. Método PCA	9
2.1. PCA	9
2.2. Cálculo de contribuciones al fallo de las variables	13
2.2.1. Estadístico T^2	13
2.2.2. Estadístico Q	14
2.3. Métodos PCA utilizados	15
2.3.1. PCA centralizado	15
2.3.2. PCA distribuido	15
2.3.3. Método PCA distribuido con factorización QR	16
2.3.4. Método de componentes principales colectivos (CPCA)	18
2.3.5. Método Merging PCA distributed	19
2.3.6. PCA distribuido usando factorización QR y agrupamiento de datos en clústers	21
3. Depuradora de aguas residuales	25
3.1. PCA con datos realistas	36
3.2. Simulación de fallos en la instalación	36
3.2.1. Fallo en el sensor de oxígeno:	37
3.2.2. Variación del valor de un parámetro del fluido	39
3.2.3. Fallos en la distribución del caudal por la instalación	41
4. Red de distribución de agua	47
4.1. Simulación de fallos en la instalación	51
4.1.1. Fallo en la bomba	51
4.1.2. Aparición de un contaminante	52
4.1.3. Fuga en una tubería	52

ÍNDICE GENERAL

5. Resultados y conclusiones. Depuradora.	55
5.1. Ejemplo: Fallo en el sensor de oxígeno de los reactores	56
5.2. Resultados y conclusiones	65
6. Resultados y conclusiones. Red de agua	69
6.1. Ejemplo: Fallo en la bomba	69
6.2. Resultados y conclusiones	79
7. Conclusiones finales	83
7.1. Conclusiones	83
7.2. Líneas futuras de trabajo	84
7.3. Artículos publicados basados en este trabajo	85

Capítulo 1

Introducción y objetivos

Esta memoria corresponde al *Trabajo Fin de Máster* requerido para la obtención del título *Master en Investigación en Ingeniería de Procesos y Sistemas Industriales* de la Universidad de Valladolid. El trabajo que se expone aquí se ha desarrollado en el Departamento de Ingeniería de Sistemas y Automática bajo la dirección de la Profesora María Jesús de la Fuente Aparicio.

1.1. Introducción

En la actualidad todos los procesos industriales presentan sistemas de control que dirigen el funcionamiento de los mismos, pero eso no da garantías de que no puedan ocurrir imprevistos que puedan hacer que el sistema deje de trabajar adecuadamente y provoque problemas como paradas de las instalaciones, averías de costosa reparación, accidentes, deterioro del producto, pérdidas de calidad, incumplimiento de plazos, etc. Esto hace que sea necesario realizar un seguimiento de diferentes parámetros del proceso que permita una rápida y eficaz detección de anomalías para que, en el caso de que estas aparezcan, actuar adecuadamente antes de que el problema tenga consecuencias demasiado graves.

Para llevar a cabo esa tarea es necesario recopilar diferentes variables, y analizarlas una por una. Pero en el caso de una instalación con cierto grado de complejidad, lo normal es encontrarse con multitud de variables a analizar entre las que suele haber un grado alto de correlación, tanto espacial como temporal, lo que complica enormemente el análisis y diagnóstico. Además, como este análisis se realiza en línea, es decir, mientras está funcionando el sistema, se recogen datos de las variables cada poco tiempo y puede llegar un momento en el que sea inasequible el tratamiento de tal cantidad de información.

Para facilitar el tratamiento de esa información se utilizan métodos de detección y diagnóstico de fallos. Este documento se va a centrar uno de

estos métodos, concretamente en una técnica estadística multivariante denominada Análisis de Componentes Principales PCA (Principal Component Analysis). Este método analiza el conjunto de datos recogidos por los sensores, y reduce su dimensionalidad, buscando unas pocas variables latentes que son las que realmente explican la dinámica del proceso (esto es debido a la gran correlación que existe entre las variables). Gracias a este espacio que generan las variables latentes se pueden construir diferentes gráficos estadísticos de control que permiten monitorizar el estado del proceso, permitiendo detectar un fallo cuando alguno de ellos viole el umbral fijado.

El trabajo desarrollado aquí consistirá en aplicar esta técnica de forma distribuida, es decir, dividiendo el conjunto de variables medidas en grupos, y analizando cada grupo de forma individual.

1.1.1. Métodos de detección y diagnóstico de fallos

A pesar del desarrollo al que han llegado los sistemas de control actuales, hay cambios en los procesos que los controladores no pueden manejar correctamente: los fallos. En el caso de sistemas de seguridad crítica, las consecuencias de estos fallos pueden ser extremadamente serias en términos de vidas humanas, impacto medioambiental o pérdidas económicas. Por lo tanto hay una gran demanda en sistemas de supervisión y diagnóstico de fallos en línea que permitan garantizar un alto nivel de eficiencia, mantenibilidad, disponibilidad y fiabilidad de la planta. Es necesario que los sistemas de control dispongan de mecanismos de monitorización que detecten fallos incipientes o en el inicio de la degradación del proceso para evitar daños o pérdidas que ocurrirían en el caso del desarrollo completo de los mismos.

La monitorización se lleva a cabo mediante los siguiente pasos:

1. Detección de la presencia de un fallo a partir de información extraída del proceso (módulo de detección).
2. Identificación del fallo: búsqueda de las variables más relevantes para diagnosticarlo.
3. Diagnóstico del fallo: determinación del tipo de fallo, localización del mismo, tamaño del fallo y causa que lo produjo (módulo de diagnóstico).
4. Recuperación del proceso: eliminación del efecto del fallo.

Por tanto, las principales funciones a realizar serán las de predicción, detección, identificación, y corrección del fallo durante la operación del sistema. La implementación automática de todas las funcionalidades descritas dependerá de aspectos económicos y el nivel de automatización general de la planta.

La definición de fallo (fault en la literatura anglosajona) dice que es un cambio inesperado del sistema que lleva a una degradación del comportamiento global del mismo, aunque no tiene porque representar un defecto de

ninguno de sus componentes, sino que se refiere a un mal funcionamiento del sistema que puede ser tolerable en su actual estado de desarrollo y debe ser diagnosticado tan pronto como sea posible para prevenir consecuencias más serias.

El término avería (failure) se refiere a un comportamiento catastrófico del sistema, es decir, se refiere a la interrupción permanente de la capacidad para mantener una función específica del sistema.

Métodos generales de detección y diagnóstico de fallos

Se presenta ahora una clasificación de métodos de detección y diagnóstico de fallos [1]. Estos métodos permiten convertir los datos tomados en línea de un proceso en unas pocas medidas útiles que permitan a los operadores determinar el estado de operación del proceso y si es necesario diagnosticar los fallos. Esto se puede abordar desde tres perspectivas distintas: métodos basados en señal, métodos analíticos y métodos basados en conocimiento.

Este tipo de métodos no son excluyentes sino complementarios. La experiencia demuestra que no hay un único método capaz de hacer una diagnosis completa.

1. Métodos basados en señal

Dentro de estos métodos existen diversas técnicas que pueden ser utilizadas:

- Alarmas: Las variables medidas son comparadas con unos valores límites constantes y preestablecidos que, de ser sobrepasados, indicarían un fallo.
- Técnicas estadísticas multivariantes: Las variables medidas en un proceso están correlacionadas con otras variables del proceso (correlación espacial) y con otras medidas de la misma variable tomadas en tiempo distinto (correlación temporal). En este apartado existen técnicas como: el Análisis de Componentes Principales (PCA), que es una técnica que reduce la dimensionalidad en términos de la varianza de los datos, el análisis del discriminante de Fisher (FDA), que es otra técnica de reducción de la dimensionalidad de un espacio de variables, o Mínimos cuadrados parciales (PLS).
- Redundancia física: Se establecen múltiples componentes físicos (sensores, actuadores y componentes del sistema) distribuidos espacialmente en toda la planta para garantizar una cierta protección en presencia de algún fallo localizado. La comprobación de las señales de salida de elementos supuestamente idénticos y un análisis de consistencia de los datos permiten detectar la presen-

cia de malfuncionamientos, fallos y la localización y aislamiento de los mismos.

- **Análisis en frecuencia:** Determinadas variables son susceptibles de ser estudiadas en el dominio de la frecuencia, aportando gran cantidad de información sobre el estado de operación de la máquina. Cada fallo puede estar representado además por un patrón de frecuencias y amplitudes característico, lo que permite su diagnóstico.

2. Métodos analíticos

El principal de ellos consiste en la redundancia analítica. Son métodos que emplean un modelo de la planta. Estas técnicas de FDI (detección e identificación de fallos) basadas en modelos constan de dos pasos: la generación de los residuos y la toma de decisión basada en esos residuos.

En el primer paso, las entradas y las salidas del sistema se procesan mediante un algoritmo determinado para generar los residuos que nominalmente son cero, y se desvían del cero de forma característica cuando ocurre un fallo particular. Las técnicas para generar los residuos son diferentes de unos métodos a otros; fundamentalmente se diferencian en la forma de calcular el modelo del proceso, que se puede hacer usando: filtros de Kalman, ecuaciones de paridad, observadores de estados, cálculo de los posibles conflictos y estimación de parámetros. En el segundo paso (toma de decisión) los residuos se examinan para determinar si hay o no fallos. Un proceso de decisión puede consistir en un simple test de umbral en los valores instantáneos, en la media móvil de los residuos, o pueden utilizarse métodos de la teoría estadística de decisión.

3. Métodos basados en conocimiento

Hay distintas técnicas que se pueden utilizar:

- **Sistemas expertos:** El sistema experto representa el conocimiento de los expertos en forma de reglas que relacionan los síntomas con las causas que los han provocado, e incluso pueden ofrecer soluciones o recomendaciones. Las dos formas más usuales de diagnóstico son el diccionario de fallos y el árbol de decisión. El diccionario de fallos contiene una lista de causas y efectos, mientras que el árbol de decisión consiste en un conjunto estructurado de reglas lógicas del tipo SI síntoma Y síntoma ENTONCES conclusión. Cada conclusión puede servir a su vez de síntoma de la siguiente regla, y así sucesivamente hasta que se alcance una conclusión final. Estas técnicas requieren la existencia de al menos un experto en el sistema estudiado.

- **Redes neuronales:** La aplicación de redes neuronales en el diagnóstico admite varias posibilidades. El modo más directo consiste en la clasificación de los diferentes estados del sistema a partir de las variables medidas. Dichos estados se agrupan en normales y defectuosos, permitiendo clasificar estos últimos para los diferentes tipos de fallos. El aprendizaje de los diferentes modos de operación del sistema, así como los modelos de fallos, se realizan mediante la presentación de muestras de ejemplos. Esto realmente entra dentro de las técnicas basadas en reconocimiento de patrones. El inconveniente de este método es que si no existen datos de fallo o son muy escasos, el modelo de red neuronal no puede aprender dicho modo de funcionamiento.

Otra forma de utilizar redes neuronales en la detección y diagnóstico de fallos es calculando un modelo nominal de la planta, ya que una red neuronal artificial es capaz de encontrar relaciones funcionales que representan el proceso, filtrando el ruido. La comparación entre la salida de la red neuronal y la salida del proceso permite la detección de posibles fallos.

- **Sistemas basados en lógica difusa:** estas técnicas se pueden utilizar tanto para la evaluación de los residuos, incorporando conocimiento del sistema en forma de reglas difusas, como en la generación de residuos, ya que de nuevo un sistema lógico difuso es un aproximador universal de funciones, en particular del modelo de un sistema. La comparación entre la salida del sistema lógico difuso y la salida del proceso permite la detección de posibles fallos.
- **Búsqueda causal:** se basa en el seguimiento de los fallos hasta su causa. La causalidad se representa mediante grafos directos dirigidos donde los nodos representan las variables de estado y las ramas las relaciones entre ellas, es decir, las restricciones que se tienen que cumplir. Una búsqueda heurística por el grafo nos da la causa final del fallo.

Requisitos de diseño

Un método de detección y diagnosis debe poseer las siguientes características:

- **Tiempo de detección:** tiempo que tarda en detectar el fallo desde que se produjo; normalmente se minimiza para un índice de falsas alarmas fijo.
- Un alto grado de detecciones correctas.
- Un índice bajo de falsas alarmas.

- Buena capacidad de aislamiento: capacidad para distinguir entre los distintos fallos.
- Sensibilidad: la magnitud del fallo necesaria para poder aislar dicho fallo.
- Robustez: habilidad de aislar fallos en presencia de errores de modelado y/o perturbaciones desconocidas.

1.2. Objetivos

Una vez visto lo que es la detección y diagnóstico de fallos de sistemas industriales y los diferentes métodos que existen, en este trabajo nos centraremos en los métodos PCA distribuidos, es decir métodos basados en señales. El objetivo global del trabajo es la detección y diagnósticos de fallos en plantas completas usando técnicas basadas en PCAs distribuidos. Este objetivo global se puede dividir en los siguientes sub-objetivos:

1. Estudio de la detección de fallos mediante el método de Análisis de Componentes Principales.
2. Propuesta de un método de detección e identificación de fallos basado en una estrategia distribuida (creando grupos de variables que serán analizados).
3. Implementación y análisis del método propuesto para analizar su rendimiento sobre datos de dos plantas complejas (con gran número de variables).
4. Búsqueda en referencias bibliográficas de otros métodos PCA distribuidos.
5. Implementación y análisis de las técnicas más destacadas que se encuentren en la revisión bibliográfica, aplicados, también, a dos plantas de gran tamaño con las que determinar el rendimiento global de cada método.
6. Analizar los resultados de todos los métodos probados y valorar la eficacia de cada una de ellos.

1.3. Organización de la memoria

Esta memoria está organizada de la siguiente forma:

- **Capítulo 2: Método PCA** En este capítulo se explica la base teórica del método PCA y de los otros métodos de detección e identificación de fallos utilizados en este trabajo.

- **Capítulo 3: Depuradora de aguas residuales** Se describe una de las dos plantas utilizadas: una depuradora de aguas residuales, su funcionamiento, características, etc. así como la forma en la que se ha trabajado con ella para provocar la aparición de fallos.
- **Capítulo 4: Red de distribución de agua** Este capítulo es similar al anterior, y en él se muestra la otra planta que se ha utilizado en este trabajo: una red de distribución de agua. Como antes, también se detalla la forma de simular fallos para el análisis posterior.
- **Capítulo 5: Resultados y conclusiones. Depuradora.** En este capítulo se exponen los resultados de aplicar los métodos de detección y diagnóstico de fallos en la planta de la depuradora, con un ejemplo del análisis de un fallo y los resultados finales obtenidos por cada método.
- **Capítulo 6: Resultados y conclusiones. Red de agua** Similar al anterior. También incluye un ejemplo de aplicación de los métodos usados y los resultados finales.
- **Capítulo 7: Conclusiones finales** Se presentan en este capítulo las conclusiones finales a las que se ha llegado con el trabajo realizado y las líneas futuras de trabajo por las que seguir la investigación.
- **Bibliografía** Aquí se muestran las referencias bibliográficas utilizadas para la realización de este trabajo.

Capítulo 2

Método PCA

2.1. PCA

El método de análisis de componentes principales (PCA) es una técnica que permite reducir la dimensionalidad de un conjunto de datos, y, por otro lado, permite encontrar las causas de la variabilidad de los mismos. Es una técnica muy usada en análisis de datos y en la construcción de modelos predictivos [2].

Consiste en una transformación lineal que proyecta los datos sobre un nuevo sistema de coordenadas en el cual el vector que representa la dirección en la que hay mayor variabilidad de los datos es capturada en el primer eje del nuevo sistema (llamado Primer Componente Principal), el vector que representa la dirección con la segunda varianza más grande es el segundo eje, y así sucesivamente.

Para crear un modelo PCA se procede de la siguiente manera: al extraer datos de una planta se obtiene una matriz de datos X , de n filas y m columnas (matriz de n observaciones y m variables observadas). Para evitar que las variables de mayor magnitud influyan más sobre el resultado, se debe normalizar esta matriz, en algunos casos restando a cada elemento la media de su columna, y después dividiendo por la desviación típica de su columna, de esta manera todos los datos tendrán media 0 y desviación típica 1; y en otros casos, convirtiendo los datos de modo que estén contenidos el rango $[0,1]$. En ambos casos se obtiene una matriz normalizada X^{norm} .

Nota: se debe tener especial precaución cuando haya que normalizar datos a media 0 y desviación típica 1, ya que, lo que se hace es:

$$X^{norm} = \frac{x - \mu}{\sigma}$$

si la desviación típica es muy baja o cero, el valor normalizado será anormalmente alto o no se podrá calcular. Si aparece una columna cuyos elementos presenten una desviación típica por debajo de cierto nivel, lo mejor es eli-

minar esa columna para evitar problemas al tratar los datos con el método PCA.

Una vez hecho esto, se calcula la matriz de covarianza:

$$S = \frac{1}{(n-1)}(X^{norm})^T X^{norm}$$

Y se descompone en valores singulares (descomposición SVD) para obtener:

$$S = V\Lambda V^T$$

Siendo Λ ($m \times m$) una matriz diagonal que contiene los valores singulares de X^{norm} (que son, también, los valores propios de la matriz de covarianza), ordenados de mayor a menor, y que representan, a su vez, los valores de varianza del conjunto de datos. Si se elige un número a de elementos de la diagonal de Λ que representen un porcentaje lo suficientemente alto (usualmente un 60% o 70%) de la varianza total de los datos (que es la suma de todos los elementos de esa diagonal), obtenemos una matriz Λ_a de dimensiones $a \times a$. Al tomar el mismo número a de entre las primeras columnas de V , resulta así una matriz P (m filas \times a columnas), llamada *Matriz de cargas*. Si multiplicamos la matriz X^{norm} por la matriz P , obtenemos otra matriz T llamada *Componentes principales de X (scores)*:

$$T = X^{norm} * P$$

Lo que se está haciendo es proyectar los datos originales en un subespacio de menor dimensión, reduciendo la cantidad de datos a manejar y perdiendo poca información, ya que se están usando las variables que explican la mayor parte de la varianza del conjunto, desechando el resto. Por ejemplo, si se parte de datos bidimensionales, como los que aparecen en la figura 2.1 (a), se observa que los datos tienen mayor variabilidad en la dirección marcada por el vector PC1 de la figura 2.1 (b) (que es una combinación lineal de las variables de los datos), luego ese vector marca el primer componente principal de los datos. También se ve variabilidad en la dirección marcada por el vector PC2 de la figura de la derecha, luego ese puede ser otro componente principal. Si los datos tuvieran más dimensiones se podrían obtener más componentes principales buscando las direcciones en las que los datos tienen más variabilidad.

Finalmente, se escogen los componentes principales que representen la mayor parte de la variabilidad de los datos. En el ejemplo, se podrían representar los datos usan el componente principal PC1, ya que representa la dirección en la que más varían los datos, mientras en PC2 la variabilidad es baja (es decir, en esa dirección hay poca información). Y de esa forma, se podrían representar la mayor parte de las informaciones contenidas en los datos con una variable (el componente principal PC1) en vez de con dos como en

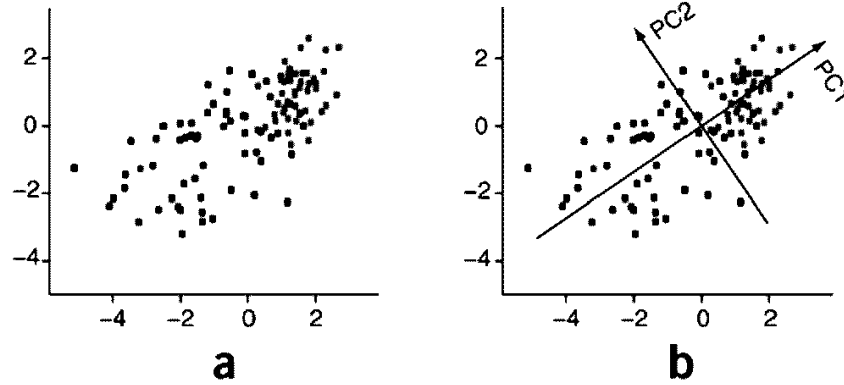


Figura 2.1: Componentes principales

los datos originales, consiguiendo así una reducción de la dimensionalidad de los datos sin perder demasiada información.

Una vez obtenidos los componentes principales, se puede reconstruir la matriz de datos original de la siguiente forma:

$$\hat{X}^{norm} \cong T * P^T$$

Que es una aproximación de la matriz X original. Si se usa esa aproximación, el error cometido es:

$$E = X^{norm} - \hat{X}^{norm}$$

Análisis de fallos: Estadísticos T^2 y Q

Para detectar fallos se dispone de las herramientas estadísticas T^2 y Q , para todos los métodos que se aplicarán posteriormente, excepto para uno concreto, en el que se usarán clústers (se agrupan datos según su proximidad, y según cómo vayan cambiando esas agrupaciones se podrá afirmar si hay fallo o no; esto se verá más en detalle en el apartado correspondiente).

- T^2

Este estadístico se calcula según la expresión siguiente:

$$T^2 = z^T P \Lambda_a^{-1} P^T z$$

Siendo z una nueva observación ($1 \times m$) de la que se quiere calcular el valor de T^2 y Λ_a es la matriz ($a \times a$) diagonal formada con las primeras a filas y a columnas de Λ . Este estadístico da una medida de la variabilidad

del proceso capturada por los componentes principales. Si el valor de este parámetro permanece por debajo de un umbral, se puede decir que el sistema está trabajando en condiciones normales. Este umbral, para un nivel de significación α , es:

$$T_{\alpha}^2 = \frac{n^2 - 1}{n(n - a)} F_{\alpha}(a, n - a)$$

donde $F_{\alpha}(a, n - a)$ es el valor crítico superior de la distribución F de Fischer-Snedecor, con n y $n - a$ grados de libertad. Los valores típicos de α que se suelen tomar son 0,01 y 0,05.

- Q

Para estudiar los $m - a$ valores singulares no contemplados en el modelo PCA, se utiliza el estadístico Q :

$$Q = r^T r$$

donde

$$r = (I - PP^T)z^T$$

siendo r el vector de residuos, calculado como la proyección de la observación z en el espacio de los residuos, I es la matriz identidad de dimensiones: m x m . El valor que no debe superar el parámetro Q para que el proceso esté bajo control es:

$$Q_{\alpha} = g * \chi^2(\alpha, h) \tag{2.1}$$

con:

$$g = \sigma_Q^2 / (2 * \bar{Q}); \tag{2.2}$$

$$h = (2 * \bar{Q}^2) / \sigma_Q^2 \tag{2.3}$$

donde $\chi^2(\alpha, h)$ es la función de distribución χ^2 acumulada con h grados de libertad y probabilidad α , \bar{Q} es la media de los valores de Q y σ_Q^2 su varianza.

Para estos dos estadísticos se establece un criterio para decir cuándo se debe activar la alarma: La alarma saltará para T^2 o Q cuando uno de ellos sobrepase su umbral un número determinado de observaciones consecutivas. En ese caso sabremos que el sistema está funcionando de forma anómala. Al establecer ese requisito se evitan falsas alarmas (se debe escoger un número adecuado de observaciones para activar la alarma de modo que este número sea lo suficientemente alto para evitar falsas alarmas pero no debe ser tan elevado que provoque que el sistema tarde mucho en avisar del fallo).

- Corrección de los umbrales de T^2 y Q

Los valores de los umbrales calculados antes son teóricos, y pueden servir como una primera aproximación, pero como se dispone de datos del sistema en condiciones normales de funcionamiento, es posible obtener unos umbrales más realistas. Para ello, se toman los datos del sistema funcionando sin fallo y se calculan los valores de T^2 y Q para cada instante. Después se buscan un valor para cada uno de ellos, para el cuál sólo un $\alpha\%$ de las veces, el valor de cada estadístico lo supere.

Para este proceso se aplica validación cruzada, que consiste en dividir los datos de forma aleatoria en varios bloques, tomar todos ellos menos uno (que será el bloque de test) y calcular ese valor del umbral (un umbral para T^2 y otro para Q) y probarlo en el bloque que no se haya usado. Repetir esto hasta que todos los bloques hayan sido alguna vez bloques de test. Y finalmente calcular el valor medio de los sucesivos umbrales que se fueron obteniendo (por supuesto, será la media de los umbrales calculados para T^2 y la media de los calculados para Q).

Finalmente, cuando se analicen los datos en situaciones de fallo, serán estos nuevos umbrales calculados ahora los que se utilicen para determinar si el sistema está en condiciones anómalas o no. En el caso de este trabajo, se han tomado los siguientes valores en todos los casos: 5% para α (es decir, en condiciones normales, sólo un 5% de las observaciones tendrán sus estadísticos T^2 y Q por encima de los respectivos umbrales) y las observaciones se dividirán en 5 bloques (4 para hacer cálculos y otro para hacer tests).

Finalmente, se elegirá entre el valor teórico y el obtenido por validación cruzada, el de más alto valor, que será el que se use para los tests.

2.2. Cálculo de contribuciones al fallo de las variables

Una vez que se produzcan estas alarmas, nos interesa saber qué variable ha contribuido más para provocar el fallo. Esto se puede hacer de la siguiente forma [1, 3, 4]:

2.2.1. Estadístico T^2

Cuando un valor supere el umbral de T^2 , se puede analizar qué variable es la que ha influido más para provocar ese alto valor del estadístico. Para ello, primero se deben buscar los componentes principales normalizados (elevados al cuadrado) de mayor valor en el momento del fallo. Esto es:

$$\left(\frac{t_i}{s_i}\right)^2$$

siendo t_i el valor del componente principal o score i en el momento de detección del fallo, y s_i es la raíz cuadrada del elemento i de la matriz Λ_d .

Una vez calculados todos los scores normalizados se buscan los más altos (es buena idea representarlos gráficamente con barras para elegir los de mayor valor más fácilmente). En este caso, se han tomado los que superan el valor de 2 (en el caso de que ninguno lo supere, se toma el más alto de todos).

Ahora se calcula la contribución de cada variable j al componente principal de alto valor a (en la observación i):

$$cont_{j,a} = \frac{t_a}{\lambda_a} p_{j,a} X_{i,j}^{norm}$$

Siendo t_a la proyección del vector de datos de la observación i normalizado sobre el vector de carga a -ésimo, λ_a es el a -ésimo valor singular obtenido en la descomposición SVD de la matriz de covarianza, $p_{j,a}$ es el elemento de la fila j y columna a de la matriz de cargas P , y $X_{i,j}^{norm}$ es el elemento de la fila i y columna j de la matriz de observaciones normalizada. Si alguno de estos valores es negativo, se le hace que valga cero, ya que eso quiere decir que la contribución tiene distinto signo que el score normalizado y, al sumar contribuciones, va a restar, de modo que tiende a bajar el valor del estadístico T^2 , es decir, no contribuye al fallo.

Se suman ahora estas contribuciones por variables, es decir, se calcula la contribución de cada una de las m variables a los scores de más alto valor, obteniéndose así lo que ha contribuido cada variable al fallo. La contribución de la variable j es:

$$Cont_j = \sum_{a_altos} cont_{j,a}$$

Analizando los valores obtenidos se puede saber qué variable es la que más ha contribuido a provocar el fallo, facilitando la localización del problema en la instalación.

2.2.2. Estadístico Q

Si se supera el umbral del estadístico Q, se puede calcular cuál es la variable que más ha influido en el fallo. Esto se hace calculando el error normalizado para cada variable j de una observación anómala x :

$$cont_{x_j} = \frac{x_j - \mu_j}{s_j}$$

Siendo μ_j el componente j del vector de medias por columnas de la matriz de observaciones (la de funcionamiento normal del sistema, usada para crear el PCA), y s_j es el elemento j del vector de desviaciones típicas, también por columnas, de la misma matriz. Es decir, estamos calculando los elementos de la matriz normalizada de observaciones. Con estos valores, identificaremos la variable que más ha influido para que el estadístico Q supere su umbral.

2.3. Métodos PCA utilizados

Para analizar los fallos de la instalación se van a usar distintas técnicas que se van a explicar a continuación. En todos los casos se aplica el análisis de componentes principales, pero con distintas particularidades que ya se verá si aportan ventajas o no.

2.3.1. PCA centralizado

En este caso se aplicará el método PCA explicado en la sección anterior usando todos los datos obtenidos de la planta, únicamente se eliminan las variables con desviación típica menor que 0,0001 para evitar que, al normalizar los datos, alguna de estas variables con baja desviación quede magnificada y falsee los resultados.

2.3.2. PCA distribuido

El método PCA se puede aplicar a todos los datos juntos, pero también se puede aplicar a varios conjuntos de datos de forma individual y después analizar los diferentes resultados obtenidos. En este caso se va a dividir las instalaciones en bloques y se va a crear y aplicar un método PCA a cada uno de esos bloques por separado, y finalmente, se compararán los resultados de los diferentes bloques. Se quiere ver si esta técnica es capaz de detectar y localizar una anomalía de funcionamiento, ya que, en caso de conseguirlo, permitiría encontrar y resolver los fallos más rápidamente.

La forma de trabajar de este método consiste en que cada bloque analice sus datos, y envíe a un nodo central información sobre si ha detectado un fallo o no. Y en el caso de que lo haya hecho, deberá comunicar el instante de detección del fallo, la variable o variables que se consideren responsables del fallo, y la contribución de cada una de esas variables al fallo. Posteriormente, en el nodo central, se analizan los datos recibidos y se determina el bloque y la variable correspondiente responsables del fallo en el sistema.

Para ello se ha considerado el siguiente procedimiento: en un instante se reciben informaciones de los bloques, si ninguno detecta fallo, se espera hasta recibir información en el siguiente instante, pero si algún bloque detecta un fallo, se comprueba la información enviada por el resto en ese instante, y se busca el bloque o bloques que han detectado también un fallo, y entre estos últimos se analiza la variable que tenga una contribución más alta. Es decir, la variable o variables responsables del fallo serán aquellas variables con mayor contribución que pertenezcan a los bloques que hayan detectado primero el fallo. (Es lógico pensar, que unos instantes después de aparecer el fallo, muchas variables de bloques en los que no haya fallo se vean afectadas por el mismo, pero eso no quiere decir que tengan responsabilidad en la anomalía).

2.3.3. Método PCA distribuido con factorización QR

Se va a aplicar un modelo PCA utilizado en sistemas distribuidos en los que los bloques envían datos para que sea un nodo central quien los procese. Se busca reducir la cantidad de información transmitida desde cada bloque que supondría el método PCA centralizado en el que se deben transmitir todos los datos obtenidos a un nodo central que los analiza en su conjunto. El objetivo es evitar pérdidas de tiempo, saturación de las comunicaciones, etc ... pero sin perder demasiada información en el proceso.

Método: [5]

El proceso a seguir es el siguiente: en cada nodo se miden el mismo número de variables (m), de modo que sea posible construir la matriz global de observaciones X (de dimensiones: $n \times m$):

$$X = \begin{pmatrix} X_0 \\ X_1 \\ \cdot \\ \cdot \\ X_{s-1} \end{pmatrix}$$

Cada submatriz X_i corresponde a las observaciones realizadas en cada uno de los s bloques en los que está dividido el sistema (los bloques van desde el bloque 0 al bloque $s - 1$). En estas submatrices, cada fila representa una observación en un instante y cada columna representa una variable medida a lo largo de diferentes instantes. Sus dimensiones son n_i filas y m columnas. De tal forma, que X será una matriz $n \times m$, con:

$$n = \sum_{i=0}^{s-1} n_i$$

Como se ha dicho, el número de bloques será s , para simplificar el proceso supondremos que $s = 2^l$. (si no fuese así, habrá que crear unas matrices vacías X_i de unos bloques ficticios para completar el hueco que hay entre s y s_{++} , con $s_{++} = 2^l$; es decir, se necesita tener un número de bloques que sea una potencia de 2, si no fuera así habrá que rellenar ese hueco hasta que la s de todo el conjunto de bloques, reales y ficticios, sea 2^l).

Algoritmo

Paso 1: se deben normalizar los datos de cada nodo de modo que estén comprendidos los datos entre 0 y 1, para ellos se usan el máximo y el mínimo local:

$$X_i^{norm01} = \frac{X_i - \min_i}{\max_i - \min_i}$$

Después se calcula en todos los bloques la matriz normalizada con la media local (para normalizar los datos a media 0), para ello, se obtiene el

vector de medias, compuesto por las medias de cada columna, y se resta a cada elemento de X_i la media de su columna. (Este paso es diferente al citado en [5], y es necesario para poder hacer los cálculos de T^2 y Q). Se obtiene finalmente la matriz normalizada \bar{X}_i . Entonces se calcula la descomposición QR de \bar{X}_i :

$$\bar{X}_i = Q_i^{(0)} R_i^{(0)}$$

Donde $R_i^{(0)}$ es una matriz triangular superior de dimensión $n \times n$. Una vez hecho esto se envían n_i y \bar{x}_i^T (vector de medias por columnas) al bloque 0, y, los bloques i , con $i \geq \frac{s}{2}$, envían $R_i^{(0)}$ al bloque $(i - \frac{s}{2})$. No se envía $Q_i^{(0)}$.

Paso 2: en el procesador i , con $0 \leq i < \frac{s}{2}$, se calcula la descomposición de la matriz siguiente:

$$\begin{pmatrix} R_i^{(0)} \\ R_{i+s/2}^{(0)} \end{pmatrix} = Q_i^{(1)} R_i^{(1)}$$

$R_i^{(1)}$ son matrices $m \times m$. Ahora, el bloque i enviará la matriz $R_i^{(1)}$ al bloque $i - \frac{s}{4}$.

Paso 3: se repiten estas descomposiciones QR y envíos de matrices hasta que el bloque 0 reciba la matriz $R_1^{(l-1)}$, esto será después de $l = \log_2 s$ etapas.

Ahora el Bloque 0 finalmente calcula la descomposición de la matriz siguiente:

$$\begin{pmatrix} R_0^{(l-1)} \\ R_1^{(l-1)} \end{pmatrix} = Q_0^{(l)} R_0^{(l)}$$

Con $R_0^{(l)}$ matriz $m \times m$. A continuación se construye la siguiente matriz y se calcula su descomposición QR:

$$\begin{pmatrix} \sqrt{n_0}(\bar{x}_0 - \bar{x}) \\ \sqrt{n_1}(\bar{x}_1 - \bar{x}) \\ \vdots \\ R_0^{(l)} \end{pmatrix} = QR$$

con:

$$\bar{x} \equiv \frac{1}{n} \sum_{i=0}^{s-1} n_i \bar{x}_i$$

Da como resultado una matriz R ($m \times m$), que se puede utilizar para calcular los componentes principales, haciendo la descomposición SVD de su matriz de covarianza S :

$$S = \frac{1}{n-1} R^T R$$

Que al descomponerla da como resultado:

$$S = V\Lambda V^T$$

A partir de esta descomposición es posible aplicar el método PCA, visto anteriormente en este capítulo, resultando las matrices de cargas P , la matriz Λ_a con los valores singulares principales, número de componentes principales, etc.

Si se envía esa información a cada bloque, se pueden calcular los estadísticos T^2 y Q locales, con sus umbrales.

Al recibir nuevas observaciones, se normalizarán al rango $[0,1]$, usando los máximos y mínimos globales, y se les restará la media global para poder usarlos para calcular los estadísticos.

2.3.4. Método de componentes principales colectivos (CP-CA)

Método:[6]

En cada uno de los s bloques se recogerán datos en un estado de funcionamiento normal, sin fallos (matriz X_i para el nodo i). Estos datos pueden contener diferente número de variables m_i (columnas de X_i), pero en todos los bloques se debe coger el mismo número de observaciones n (filas de X_i), para que todo cuadre cuando se unan los datos de todo el sistema. Con estos datos se calcula un PCA (matriz de covarianza, autovalores más representativos, componentes principales, matriz de cargas, etc...). Ahora se proyectan los datos sobre la matriz de cargas para obtener los componentes principales de cada bloque i :

$$T_i = X_i P_i^T$$

(*Nota:* en el artículo en el que se basa este método se seleccionan las filas más representativas, reduciendo así los datos a transmitir, pero en este caso se han dejado todas las filas obtenidas).

Ahora cada bloque transmite la matriz T_i y la matriz P_i a un procesador central. (Evitando así enviar las X_i de mayor dimensión que T_i). En este procesador, se construye la matriz T que recoge todos los componentes principales de los bloques (la matriz T tendrá n filas y m columnas, con $m = \sum_{i=1}^s m_i$):

$$T = [T_1, T_2, \dots, T_s]$$

y la matriz diagonal con las matrices de cargas P_i :

$$P = \begin{bmatrix} P_1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & P_2 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & & & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & P_s \end{bmatrix}$$

De tal forma que se verifica que:

$$T = XP$$

siendo X la matriz que reúne a todas las X_i de los diferentes bloques (que en este método no se calcula); es posible aproximar la matriz global de observaciones de la siguiente forma:

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

Sobre esta matriz \hat{X} se aplicará otro método PCA, y se calcularán componentes principales, etc... De cara a simplificar las operaciones, se puede evitar el cálculo de la matriz \hat{X} , ya que si calculamos los autovectores " v_i " correspondientes a los "ag" autovalores más altos de la matriz de covarianza de "T", al multiplicar esos autovectores por la matriz "P" se obtienen los autovectores " w_i " correspondientes de la matriz de covarianza de "X":

$$w_i = Pv_i$$

Los autovalores más altos de la matriz de covarianza de "T" coinciden con los de la matriz de covarianza de X. También se obtendrán los umbrales globales para T^2 y Q. Se calculan los estadísticos a nivel global con la matriz \hat{X} , y ahora, para detectar fallos, se comparan estos estadísticos con los umbrales globales calculados antes.

2.3.5. Método Merging PCA distributed

En este método se calculará un modelo PCA en cada bloque, que será enviado a un procesador central, éste unirá los datos y creará un modelo PCA global. Después retornará los datos de este modelo a cada bloque para que lo usen en la detección de fallos.

Método: [7]

Ahora hay un sistema con s bloques, en cada uno de ellos se construye una matriz X_i con observaciones ordenadas (n_i observaciones o filas, y m variables o columnas). A estas matrices, normalizadas sus columnas al rango $[0,1]$ usando máximos y mínimos locales (igual al método seguido con los datos en la técnica QR), se les resta la media local, y se calcula un modelo PCA,

individual para cada bloque. De estos PCA se toman los valores singulares más altos, según los parámetros que se hayan establecido (α_i : porcentaje de varianza explicada por los mayores valores singulares del bloque i ; o también k_i : número de valores singulares o de componentes principales elegidos en el bloque i). Con lo cual, en cada bloque se dispone de los siguientes datos que serán enviados al nodo central:

- n_i : número de observaciones del bloque i
- α_i o k_i , parámetros para la selección de los componentes principales
- \bar{x}_i : media de las columnas de X_i
- $\tilde{\Lambda}_i$: matriz diagonal con los autovalores más altos
- \tilde{P}_i : matriz con los autovectores correspondientes a los autovalores más altos

Nota: al hacer la descomposición SVD de la matriz de covarianza nS_i , se obtiene:

$$nS_i = P_i \Lambda_i P_i^T$$

con:

$$nS_i = (X_i^{norm})^T X_i^{norm}$$

pero, en este caso, se hará lo siguiente:

$$X_i^{norm} = V_i \lambda_i P_i^T$$

con:

$$\Lambda_i = \lambda_i^T \lambda_i = \lambda_i^2$$

para X_i^{norm} la matriz de observaciones del bloque i normalizada. Se toman los componentes principales y se construyen las matrices reducidas: $\tilde{\lambda}_i$ y \tilde{P}_i . Y son estas matrices las que se envían.

Se mandan los datos, y el procesador central calcula la siguiente aproximación a la matriz global de covarianza:

$$n\tilde{S} = \sum_{i=1}^s \tilde{P}_i \tilde{\lambda}_i^2 \tilde{P}_i^T + \sum_{i=1}^s n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

Ahora se hace la descomposición siguiente:

$$\tilde{S} = \frac{1}{n} U D U^T$$

Nota: en el artículo del que se ha extraído este método se calcula $n\tilde{S}$, pero para poder calcular los estadísticos correctamente, calcularemos la matriz \tilde{S} .

Sobre esta matriz \tilde{S} se crea un modelo PCA (descomposición SVD, componentes principales, etc. . .). (Se deberá haber especificado también un α_g o un k_g para elegir un número de componentes principales en el modelo global, al igual que se hizo para cada bloque). Y se obtendrán unas matrices: $\tilde{\Lambda}$ y \tilde{P} . A cada bloque se envía ahora la matriz \tilde{P} y el vector de medias global \bar{x} .

También hay que calcular aquí los umbrales de T^2 y Q y hay que enviarlos a cada bloque.

Ahora los nuevos datos que se vayan registrando en los bloques se proyectarán en el espacio de dimensión reducida global de la siguiente manera, si z_i es un vector de observaciones del bloque i , normalizado al rango $[0,1]$:

$$\tilde{z}_i = z_i^{norm} \tilde{P}$$

siendo z_i^{norm} el vector de observaciones normalizado a media 0 con la media local. Se hacen ahora los cálculos de los estadísticos T^2 y Q para cada observación z_i y se comprueba si se superan sus respectivos umbrales:

$$T^2 = z_i^{norm} \tilde{P} \tilde{\Lambda}^{-1} \tilde{P}^T (z_i^{norm})^T$$

$$Q = r^T r$$

con:

$$r = (I - \tilde{P} \tilde{P}^T) (z_i^{norm})^T$$

2.3.6. PCA distribuido usando factorización QR y agrupamiento de datos en clústers

En este caso se aplicará un PCA sobre un sistema distribuido en el que los bloques envían datos para que sea un nodo central quién los procese. Estos datos se envían agrupados (en clusters) y según cómo vayan variando con el tiempo esas agrupaciones de datos se determinará si hay fallo o no. También se busca con este método reducir la cantidad de información transmitida desde cada bloque al nodo central.

Método: [8]

Antes de nada, en cada uno de los $(s - 1)$ bloques del sistema se toman datos en condiciones normales de funcionamiento (matriz X_i para el bloque i , de dimensiones $n_i \times m$, todos los bloques medirán el mismo número de variables). Estos datos se normalizan, de modo que estén contenidos en el rango $[0,1]$. Para ello cada bloque calcula el máximo y mínimo de cada variable, y los usa para normalizar los datos.

Una vez hecho esto, se aplica el método QR descrito en la sección 2.3.3 con los $(s - 1)$ bloques del sistema. Se obtiene una matriz R con la que se construye la matriz de covarianza:

$$S = \frac{1}{n} R^T R$$

siendo n el número total de observaciones:

$$n = \sum_{i=0}^{s-1} n_i$$

Se aplica el método PCA a esta matriz de covarianza, y se calculan sus componentes principales, de estos, se selecciona el componente principal asociado al autovalor mayor: φ .

Ahora se debe establecer el perfil normal de funcionamiento. Para ello, en los bloques, se agrupan los datos de funcionamiento normal en clústers, de modo que en cada bloque haya un conjunto de agrupaciones, dentro de las cuales estén contenidas cada una de las observaciones realizadas. Cada clúster será una región del espacio definida por un centroide y un radio, y dentro de ese espacio se encontrarán diferentes puntos u observaciones.

El proceso es el siguiente: se establece un radio ω para los clúster (cuanto más grande sea el radio, mayores serán los clúster y más puntos contendrán, pero aportarán menos información; cuanto más pequeño, mayor número de clúster y más información), se toma un bloque, en ese bloque se coge la primera observación, y se hace un clúster con ella, de forma, que su centroide será ese punto y su radio ω . Se toma la siguiente observación, si está dentro de este clúster, se la incluye en él, y se actualiza el centroide, que será el punto medio entre estas dos observaciones. Si no está contenida, se crea un nuevo clúster para ella, con el centroide coincidente con esa observación. Se toma otra observación, y se repite el proceso, si está dentro de alguno de los clúster anteriores se la incluye en el que corresponda y se actualiza el centroide (que será el punto medio entre todas las observaciones contenidas en el clúster), si no, se crea un nuevo clúster.

Repetimos este proceso hasta asignar todas las observaciones a un clúster. Una vez hecho esto, se comprueba cuál es el punto de cada clúster más alejado de su centroide, y se establece que el nuevo radio del clúster será la distancia entre ese punto y el centroide.

Finalmente se obtiene, para cada bloque, un conjunto de clúster que agrupan todas las observaciones. Habrá clúster con un único punto, otros con dos, etc... Ahora, todos los bloques envían sus clúster al procesador central (es decir, envían los centroides y los radios de sus clúster, pero no envían las matrices X_i).

Para establecer el perfil de funcionamiento normal, el procesador central calcula la máxima distancia entre los clúster y el primer componente principal calculado antes:

$$d_{max} = \max\{d(C_i^j, \varphi)\}$$

siendo la distancia de un clúster C_i^j (caracterizado por c_i^j y r_i^j , centroide y radio, respectivamente, del clúster número j del bloque i) a ese componente:

$$d(C_i^j, \varphi) = d_p(c_i^j, \varphi) + r_i^j$$

con:

$$d_p(c_i^j, \varphi) = \text{sqr}t(\left(\|c_i^j - \bar{x}\|\right)^2 - (\varphi^T(c_i^j - \bar{x}))^2)$$

De entre todas las distancias calculadas, se toma la mayor, que se usará cuando entren datos nuevos (también agrupados en clúster), de modo que cuando un nuevo clúster de datos esté a una distancia mayor del primer componente principal que esa distancia máxima, significará que el sistema se está desviando del funcionamiento normal, es decir:

$$\left. \begin{array}{l} d(C_i^j, \varphi) > d_{max} \implies \text{Datos anómalos} \\ d(C_i^j, \varphi) \leq d_{max} \implies \text{Datos normales} \end{array} \right\}$$

Si en un determinado momento, hay demasiados clústers anómalos, querrá decir que hay algún problema en el sistema.

Identificación de las variables responsables del fallo

En éste método, cuando se detecta un fallo, sólo se conoce en qué bloques se ha producido un fallo, pero no se puede, a priori, decir qué variable es la responsable del fallo.

Para realizar esto, el proceso que se ha seguido ha sido el siguiente, se toman los bloques en los que se detectan fallos, de estos, se seleccionan aquellos en los que el fallo haya aparecido antes, y finalmente, se realiza un cálculo de contribuciones de las variables de ese bloque al estadístico Q en el instante del fallo, para identificar a la responsable del fallo.

Capítulo 3

Depuradora de aguas residuales

Una planta de depuración de aguas residuales es una planta que consta de diferentes elementos que permiten transformar un agua contaminada con los residuos y contaminantes que se vierten en los desagües de una población, en un agua apta para ser vertida en un río, o en cualquier otra masa de agua. Para ello se debe conseguir que la cantidad de contaminantes presentes en ese agua residual sea lo suficientemente baja para que el impacto en la zona de vertido sea lo menor posible.

Estas aguas están formadas por un 99 % de agua y un 1 % de sólidos en suspensión y en solución. Estos sólidos pueden clasificarse en orgánicos e inorgánicos:

- Los sólidos inorgánicos están formados principalmente por nitrógeno, fósforo, cloruros, sulfatos, carbonatos, bicarbonatos y algunas sustancias tóxicas como arsénico, cianuro, cadmio, cromo, cobre, mercurio, plomo y zinc.
- Los sólidos orgánicos se pueden clasificar en nitrogenados y no nitrogenados. Los nitrogenados, es decir, los que contienen nitrógeno en su molécula, son proteínas, ureas, aminas y aminoácidos. Los no nitrogenados son principalmente celulosa, grasas y jabones.

El proceso de tratamiento de estas aguas se puede dividir en varias etapas: pretratamiento, primaria, secundaria, terciaria, desinfección del agua tratada y tratamiento de fangos. (Algunos autores unen las etapas preliminar y primaria y las denominan como etapa primaria).

- *Etapla preliminar* La etapa preliminar debe cumplir dos funciones:
 - Medir y regular el caudal de agua que llega a la planta.
 - Extraer los sólidos flotantes grandes y la arena (a veces, también la grasa).

- *Etapa primaria* Tiene como objetivo eliminar los sólidos en suspensión por medio de un proceso de sedimentación por gravedad (o asistida por coagulantes y floculantes). Las estructuras encargadas de esta función son los estanques de sedimentación primarios o decantadores primarios, cuya profundidad fluctúa entre 2 y 5 m. En esta etapa se elimina por precipitación alrededor del 60 al 70 % de los sólidos en suspensión. En la mayoría de las plantas existen varios sedimentadores primarios y su forma puede ser circular, cuadrada o rectangular.
- *Etapa secundaria* Aquí se busca eliminar la materia orgánica en disolución y en estado coloidal mediante un proceso de naturaleza biológica seguido de otro proceso de sedimentación. El proceso biológico es un proceso natural que tiene lugar en un reactor y que es llevado a cabo por los microorganismos presentes en el agua residual. Estos microorganismos, principalmente bacterias, se alimentan de la materia orgánica presente en el agua, mediante procesos anóxicos y aerobios (en estos últimos, existe un sistema de aireación para aportar oxígeno a la reacción química); produciéndose así anhídrido carbónico y agua, y resultando de este proceso una biomasa bacteriana que es precipitada en el decantador secundario. De esta forma, el agua queda limpia, retirándose de esta unos fangos para los que hay que buscar un tratamiento adecuado. Los microorganismos presentes en el reactor pueden estar en suspensión en el agua (procesos de crecimiento suspendido o fangos activados), adheridos a un medio de suspensión (procesos de crecimiento adherido: lechos bacterianos, biodiscos) o distribuidos en un sistema mixto (procesos de crecimiento mixto).

En el decantador secundario, hay un flujo lento de agua, de forma que la biomasa, es decir, los flóculos bacterianos producidos en el reactor, sedimentan. El sedimento que se produce y que, como se dijo, está formado fundamentalmente por bacterias, se denomina fango activo.

Las estructuras usadas para el tratamiento secundario incluyen filtros de arena intermitentes, filtros percoladores, contactores biológicos rotatorios, lechos fluidizados, estanques de fangos activos, lagunas de estabilización u oxidación y sistemas de digestión de fangos.

- *Etapa terciaria* En esta etapa se suprimen algunos contaminantes específicos presentes en el agua residual tales como los fosfatos que provienen del uso de detergentes domésticos e industriales y cuya descarga en cursos de agua favorece la eutrofización, es decir, un desarrollo incontrolado y acelerado de la vegetación acuática que agota el oxígeno, y mata la fauna existente en la zona. No todas las plantas tienen esta etapa ya que dependerá de la composición del agua residual y el destino que se le dará.

- *Desinfección del agua tratada* Las aguas tratadas normalmente contienen microorganismos patógenos que sobreviven a las etapas anteriores de tratamiento. Por tal razón es necesario proceder a la desinfección del agua. Este proceso es especialmente importante si estas aguas van a ser descargadas a aguas de uso recreacional (piscinas, etc.), aguas destinadas a acuicultura (cría de pescados y mariscos en cautividad) o aguas que pudieran usarse como fuente para consumo humano.

Los métodos de desinfección son principalmente la cloración y la ozonización, pero también se ha usado la bromación y la radiación ultravioleta. El más usado es la cloración por ser barata, fácilmente disponible y muy efectiva. Sin embargo, como el cloro es tóxico para la vida acuática, el agua tratada con este elemento debe ser sometida a dechloración antes de verterla a cursos de agua natural.

Desde el punto de vista de la salud pública se encuentra aceptable un agua que contenga menos de 1.000 bacterias coliformes totales por 100 ml y con una demanda biológica de oxígeno (DBO) inferior a 50 mg/L.

- *Tratamiento de fangos* Los sedimentos que se generan en las etapas primaria y secundaria se denominan fangos. Estos fangos contienen gran cantidad de agua, microorganismos patógenos y contaminantes orgánicos e inorgánicos. Se han desarrollado varios métodos para el tratamiento de los fangos: *digestión anaerobia*, *digestión aerobia*, *compostaje*, *acondicionamiento químico* y *tratamiento físico*. El propósito del tratamiento de los fangos es destruir los microbios patógenos y reducir el porcentaje de humedad, haciéndolos aptos para ser llevados a vertederos, ser usados como fertilizante, etc.
 - El proceso de *digestión anaerobia* se realiza en un estanque cerrado llamado digestor y no requiere la presencia de oxígeno pues se realiza por medio de bacterias que se desarrollan en su ausencia. Para el óptimo crecimiento de estos microorganismos se requiere una temperatura de 35° C. Las bacterias anaerobias degradan la materia orgánica presente en el agua, en una primera fase, a ácido propiónico, ácido acético y otros compuestos intermedios, para posteriormente dar como producto final metano (60–70 %), anhídrido carbónico (30 %) y trazas de amoníaco, nitrógeno, anhídrido sulfuroso e hidrógeno. El metano y el anhídrido carbónico son inodoros; en cambio, el ácido propiónico tiene olor a queso rancio y el ácido acético tiene olor a vinagre.
 - La *digestión aerobia* se realiza en un estanque abierto y requiere la presencia de oxígeno y, por tanto, la inyección de aire u oxígeno. En este caso la digestión de la materia orgánica es efectuada por bacterias aerobias, que realizan su actividad a temperatura ambiente. El producto final de esta digestión es anhídrido carbónico

y agua. No se produce metano. Este proceso bien efectuado no produce olores.

- El *compostaje* es el proceso de digestión aeróbica del fango mezclado con madera u otros desechos orgánicos, con el objetivo de obtener un producto utilizable como fertilizante
- El *acondicionamiento químico* busca concentrar los lodos, reduciendo así su contenido en agua. Se puede aplicar tanto a los fangos crudos como digeridos e incluye la aplicación de coagulantes tales como el sulfato de aluminio, el cloruro férrico y los polímeros, los que tienen como función ayudar a la sedimentación de las materias en suspensión y solución en el fango.
- El *tratamiento físico* incluye el tratamiento por calor y la congelación de los fangos.

Una vez concluidos los procesos anteriores, los fangos aún contienen mucha agua (alrededor de un 90 %) por lo que se requiere deshidratarlos para su eliminación final. Para ello se han diseñado dos métodos principales: secado por aire y secado mecánico:

- Los procesos de *secado por aire* de los fangos se dividen en: lechos de arena, lechos asistidos de arena, lagunas de fangos, lechos adoquinados y eras de secado.
- Para el *secado mecánico* existen filtros banda, filtros prensa, filtros de vacío y centrifugas.

Estos fangos deshidratados deben eliminarse de una forma ambientalmente segura. Para ello, según el caso, pueden llevarse a vertederos, ser depositados en terrenos agrícolas y no agrícolas o ser incinerados. La aplicación en terrenos agrícolas requiere que el fango no presente sustancias tóxicas para las plantas, animales y seres humanos, en caso contrario, si contienen niveles por encima de valores aceptables, deben ser llevados a vertederos o ser incinerados.

Hasta aquí se han visto los procesos que pueden tener lugar en una depuradora, y ahora se va describir el modelo de depuradora utilizado en este trabajo. No es necesario que se apliquen todas las etapas vistas anteriormente, se deben usar sólo las necesarias de acuerdo con la calidad y condiciones del agua concreta que se vaya a tratar. No es lo mismo tratar aguas residuales domésticas que industriales, no es lo mismo tratar aguas procedentes de una zona con clima seco que otra que tenga un clima lluvioso, que contendrá gran cantidad de agua procedente de la red de alcantarillado (esta red recoge contaminantes presentes en los suelos de las calles (hojas muertas, elementos procedentes de los vehículos (aceites, combustibles, restos de goma de neumático, etc.), pequeños embalajes (plásticos, papeles, etc.), polvo, etc.).

También, a la hora de diseñar la instalación, se deberá tener en cuenta en qué condiciones debe salir el agua de la instalación una vez que ya ha sido tratada (niveles máximos de contaminantes para ser un agua apta para el vertido en la naturaleza).

En este estudio se va a utilizar un modelo de depuradora implementado por Ulf Jeppsson, existen varias versiones de la instalación según incluyan más o menos elementos (digestores, decantadores, sistemas de tratamiento de fangos, etc.) y según el entorno en el que se hayan programado (Matlab, FORTRAN, etc.). Su funcionamiento se explica en la web de la International Water Association (IWA) [9] (para conseguir los modelos de simulación se debe contactar con la anterior asociación, ya que no están disponibles en su web) . Para las simulaciones que se van a realizar se usará el modelo BSM2, que es el más completo.

Descripción del modelo BSM2:

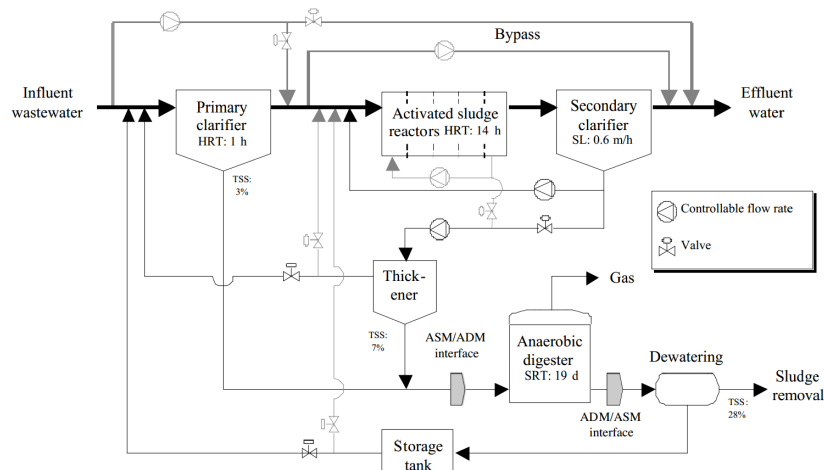


Figura 3.1: Depuradora BSM2

Este modelo se puede dividir en varios bloques compuestos por varios elementos (ver Figura: 3.1). Podemos distinguir un primer bloque en el que se encuentra el decantador primario (Primary clarifier) que separa parte de la materia en suspensión del caudal influente. Otro bloque consiste en una serie de 5 reactores químicos (Activated sludge reactors) que, a través de procesos biológicos (llevados a cabo por bacterias y otros microorganismos, de forma anóxica en los dos primeros reactores y aerobia en los tres últimos), descomponen la materia orgánica reduciendo la demanda química de oxígeno del agua residual, amonificando el nitrógeno orgánico, etc. A la salida de estos reactores se sitúa un segundo decantador (Secondary clarifier) que reduce la cantidad de sólidos en suspensión en el líquido, obteniendo un agua

que ya está lista para abandonar la depuradora. Por otro lado, los sólidos retirados del agua residual se envían a un bloque en el que serán tratados, excepto una pequeña parte que se envían a la entrada de los reactores en serie explicados antes. Finalmente, se dispone de una serie de elementos cuya finalidad es el tratamiento de los sólidos retirados por los decantadores, es el bloque de tratamiento de fangos. Aquí se someten a diversos tratamientos, en los que se reduce la cantidad de materia orgánica presente generándose gas combustible (Anaerobic Digester), se reduce la cantidad de agua que llevan (Thickener y Dewatering), se almacenan para ser luego reintroducidos en el sistema (Storage tank) y una parte se retira para ser eliminados (en vertederos, etc.).

Este modelo cuenta con un sistema de control que regula su funcionamiento. Este sistema mide el nivel de oxígeno disuelto en el líquido del reactor 4 del bloque de reactores “ Activated sludge reactors” y en función de ese dato ajusta la cantidad de oxígeno que inyecta en los reactores 3, 4 y 5 del bloque “ Activated sludge reactors”. Otro control que tiene el sistema es que permite variar los caudales Q_r (caudal de retorno del decantador secundario a los reactores), Q_{int} (caudal de retorno del reactor 5 al reactor 1) y Q_w (caudal de entrada al “espesador” de lodos (Thickener)). Además existen algunas tuberías que permiten desviar el flujo de agua, si las condiciones lo requieren, para que no atraviese determinados elementos de la red (tuberías “by-pass”).

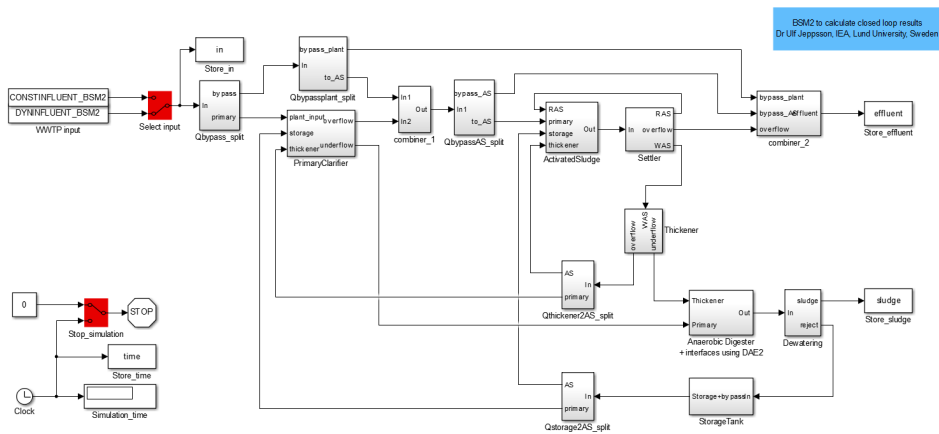


Figura 3.2: Modelo BSM2 en Matlab

El modelo de simulación usado es el que está implementado en Matlab (más concretamente, en el módulo Simulink) (ver Figura: 3.2). Está compuesto de varios bloques que representan, algunos de ellos, los diferentes elementos de la instalación, y de otros bloques auxiliares que sirven para facilitar la conexión entre los bloques anteriores (realizando algún tipo de

tratamiento a los datos, uniendo caudales, etc.).

Las líneas que unen los bloques representan la transmisión de datos entre bloques (un bloque opera con sus variables de entrada, su estado interno, etc. y el resultado lo envía al bloque siguiente). Esos datos pueden ser unidimensionales o multidimensionales. En este caso, aunque habrá algunas variables de una dimensión (por ejemplo, la variable tiempo (time)), la mayoría de las variables (y por tanto, la mayoría de las líneas entre bloques) van a ser multidimensionales, y van a contener datos sobre el estado del fluido en ese determinado punto. Así el estado del fluido se caracterizará con variables de dimensión 21 (o múltiplos de 21), de forma que cada uno de los 21 datos contenidos en estas variables responderán al esquema de la figura 3.3.

1. SI(inert soluble material, g COD.m⁻³);
2. SS(readily biodegradable substrate, g COD.m⁻³);
3. XI(inert particulate material, g COD.m⁻³);
4. XS(slowly biodegradable substrate, g COD.m⁻³);
5. XB,H (heterotrophic biomass, g COD.m⁻³);
6. XB,A (autotrophic biomass, g COD.m⁻³);
7. XP(inert particulate material from biomass decay, g COD.m⁻³);
8. SO (dissolved oxygen, g (-COD).m⁻³);
9. SNO(nitrate and nitrite, g N.m⁻³);
10. SNH (ammonia and ammonium, g N.m⁻³);
11. SND(soluble organic nitrogen associated with SS, g N.m⁻³);
12. XND(particulate organic nitrogen associated with XS, g N.m⁻³);
13. SALK(alkalinity);
14. TSS (total suspended solids, g SS.m⁻³);
15. flow rate, m³.d⁻¹;
16. temperature, °C;
17. dummy variable (soluble) no 1;
18. dummy variable (soluble) no 2;
19. dummy variable (soluble) no 3;
20. dummy variable (particulate) no 1;
21. dummy variable (particulate) no 2

Figura 3.3: Esquema de las variables en el modelo BSM2

Es decir, que si se accede al contenido de una variable que tiene dimensión 21, el primer elemento será el material soluble inerte (S_I), el segundo será el substrato fácilmente biodegradable (S_S), etc. Si nos encontramos con una variable con dimensión múltiplo de 21, quiere decir que está almacenando el estado del fluido en varios puntos.

Por ejemplo, si se analiza la variable que almacena los datos de un decantador, habrá un bloque de 21 valores que correspondan al flujo que sale

por la parte superior del mismo (fluido al que se han eliminado sólidos en suspensión), otros 21 serán el flujo inferior (partículas que se han decantado), otros 21 serán el estado del fluido en el primer nivel del decantador, habrá otros 21 para el segundo nivel, y así sucesivamente.

Las últimas cinco variables de cada conjunto de 21 no se usan en la ejecución normal del modelo, y su valor es siempre cero. Son variables vacías que están disponibles para el usuario que quiera hacer modificaciones y ampliar la información de salida del programa. En nuestro caso no se usarán.

Volviendo al esquema de Simulink, para simplificar la representación del modelo se ha implementado éste mediante bloques del tipo “Subsystem”. Estos bloques se usan para representar subconjuntos de elementos de la instalación (por ejemplo, uno de los bloques del modelo, denominado “Activated Sludge”, representa el bloque de reactores de fangos activados, si accedemos al contenido de este bloque, encontraremos cada uno de los 5 reactores). Dentro de esos bloques “Subsystem” se hallarán otros bloques “Subsystem” u otros elementos que realicen operaciones con los datos (sumas, productos, multiplexores, retardos, llamada a un programa S-file, etc.). Haciendo doble clic sobre un bloque tipo “Subsystem”, el programa nos mostrará lo que contiene, y si se hace doble clic otra vez sobre algún elemento “Subsystem” contenido en él, podremos acceder a otro subnivel, y así sucesivamente. Si el bloque es algún tipo de operador (suma, producto, etc.) nos permitirá cambiar los parámetros de ese elemento.

Por ejemplo, si se hace doble clic sobre el bloque settler del esquema general (ver Figura: 3.4), se accede al contenido de ese bloque (ver Figura: 3.5).

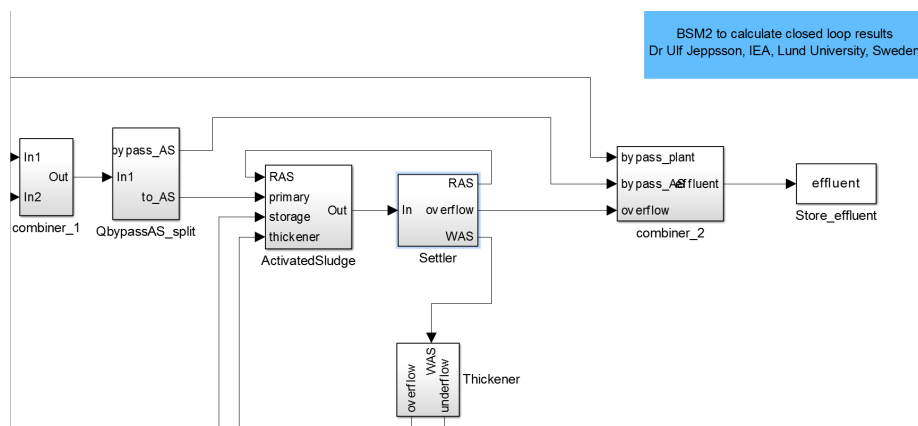


Figura 3.4: BSM2: settler

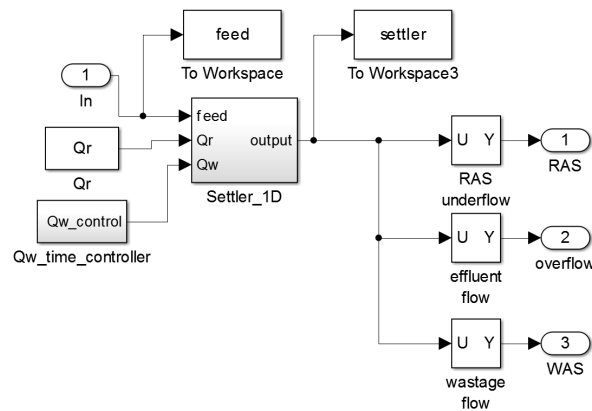


Figura 3.5: Contenido del bloque Settler

En el interior del bloque se pueden ver otros bloques: en la parte izquierda aparecen las tres entradas al bloque Settler (In, Qr, Qw_control), la entrada In tiene una derivación a un elemento del tipo To Workspace, que almacenará en la variable *feed* los valores que vaya tomando In a lo largo de la simulación. Estas entradas van a parar a un bloque llamado Settler_1D con tres entradas y una salida (si hacemos doble clic sobre él aparecerá otro esquema con más bloques que trabajan con las tres entradas anteriores para darnos la salida de este bloque). Esta salida se almacena, a lo largo de la simulación en la variable settler (de una forma similar a cómo ocurría con In). Y, a su vez esta salida de Settler_1D llega a tres bloques que separan los datos en tres partes que son las tres salidas del bloque Settler (RAS, overflow y WAS).

Esta forma de representar el modelo en forma de bloques que contienen partes de la instalación permite implementar un modelo de simulación bastante complejo, pero estructurado en niveles o capas de modo que se facilite su manejo y comprensión.

De cara a extraer datos para analizarlos luego, lo que se va a hacer es extraer ciertas variables que se irán almacenando a lo largo de la simulación (como se comentó anteriormente, esto se hará mediante los bloques del tipo To Workspace). Estas variables nos informarán del estado del agua residual a la salida de distintos elementos de la red. Las variables a utilizar son las que se indican en la figura 3.6.

A lo largo de la simulación, que en este caso tendrá una duración de 609 días, se irán almacenando, cada 15 minutos (4 medidas cada hora), los valores que van tomando diferentes magnitudes del fluido medidas en varios puntos de la instalación. De tal forma, que las variables enumeradas en la figura 3.6 tendrán forma de matriz de 21 columnas y 58464 filas ($4 \frac{\text{medidas}}{\text{hora}} * 24 \frac{\text{horas}}{\text{día}} * 609 \text{ días} = 58464 \text{ filas}$), representando cada fila a cada

Variable en Matlab	Contenido de la variable
in	Flujo de entrada a la planta
primaryout (elementos 1-21)	Flujo de salida primer decantador
primaryout (elementos 22-42)	Flujo inferior primer decantador
primaryout (elementos 43-63)	Valores internos primer decantador
reac1	Flujo de salida del reactor 1 (Activated Sludge reactors)
reac2	Flujo de salida del reactor 2 (Activated Sludge reactors)
reac3	Flujo de salida del reactor 3 (Activated Sludge reactors)
reac4	Flujo de salida del reactor 4 (Activated Sludge reactors)
reac5	Flujo de salida del reactor 5 (Activated Sludge reactors)
settler (elementos 1-21)	Flujo inferior decantador secundario
settler (elementos 22-42)	Flujo salida decantador secundario
thickenerout (elementos 22-42)	Flujo de salida concentrador de fangos (thickener)
thickenerout (elementos 1-21)	Flujo inferior concentrador de fangos (thickener)
digesterinpreinterface (elementos 1-21)	Flujo de entrada al digestor anaeróbico (Anaerobic digester)
digesteroutpostinterface (elementos 1-21)	Flujo de salida del digestor anaeróbico (Anaerobic digester)
dewateringout (elementos 1-21)	Flujo de salida del deshidratador (Dewatering)
storageout (elementos 1-21)	Flujo de salida del tanque de almacenamiento de fangos
storagebypass (elementos 1-21)	Flujo de desviación del tanque de almacenamiento de fangos (Storage Tank)
storagetotout (elementos 1-21)	Flujo de salida y de desviación (bypass) del tanque de almacenamiento de fangos (Storage Tank)
effluent (elementos 1-21)	Flujo de salida de la depuradora (WWTP effluent)
sludge (elementos 1-21)	Flujo de fango eliminado (WWTP sludge disposal)

Figura 3.6: Variables extraídas de la simulación en Matlab

una de las 58464 medidas realizadas a lo largo de toda la simulación.

Como se acaba de explicar, las instalaciones de depuración de aguas residuales presentan un grado importante de complejidad, ya que están compuestas de muchos elementos (decantadores, digestores, reactores, etc.) y a su vez, estas partes de las depuradoras funcionan siguiendo una serie de reacciones químicas complejas, sometidas a una gran variedad de factores internos y externos (características del agua residual, diseño de la instalación, temperatura, etc). Debido a esto se hace difícil su manejo, y es muy importante contar con datos fiables y representativos de su estado de funcionamiento.

Esto, a su vez, plantea otros dos problemas: por un lado, como ya se dijo, la cantidad de datos que se pueden extraer de una instalación de este tipo (por ejemplo, oxígeno disuelto, biomasa, alcalinidad, nitritos, amoníaco, etc.) puede ser tan grande que hagan que sea inasequible su estudio en tiempo real,

y por otro lado, no suele ser posible acceder instantáneamente a todos los datos de funcionamiento del sistema; es decir, valores como la temperatura o el pH del líquido se pueden medir simplemente introduciendo un sensor en un punto en el que queramos obtener esos datos, pero no hay sensores que midan, por ejemplo, la cantidad de sustrato fácilmente biodegradable (S_s). Para conocer estos datos, habría que tomar una muestra y analizarla en un laboratorio mediante un proceso complejo, lento y caro, que no permite un control en tiempo real de la planta depuradora. Así que, para hacer viable el análisis en tiempo real y hacer más realista el proceso de toma de datos, se aplicará el método PCA (que permite analizar muchos datos de forma simple y rápida) y se combinarán los datos recogidos de la simulación de modo que se obtengan medidas similares a las que es posible recoger, en el momento, en una instalación real.

El proceso de recogida de datos consistirá en simular el modelo tal como está, para tener datos del funcionamiento en condiciones normales, y posteriormente ir introduciendo diferentes fallos (descenso del nivel de oxígeno, variaciones de caudal, etc.) en el modelo y ejecutar la simulación para conseguir datos del funcionamiento del sistema en condiciones anómalas. Después se comprobará si los métodos PCA propuestos son capaces de detectar esos fallos.

En este trabajo se va a trabajar de dos formas, la primera consistirá en analizar el sistema en bloque, y la segunda, se dividirá el sistema en varias partes y se analizará cada una por separado. Se puede realizar esta división de varias formas: por topología (agrupando los elementos más próximos, según las conexiones que haya entre ellos, etc.), en función de la relación que exista entre los elementos a lo largo del proceso, analizando la correlación entre las variables del sistema de forma que se puedan agrupar las que tengan más relación entre ellas (usando métodos como SparsePCA [10]), etc. En este caso se hará mediante topología, y se dividirá la instalación por puntos de medida, es decir, los datos disponibles se refieren a medidas de magnitudes realizadas en diferentes puntos del sistema (por ejemplo, en el caudal de entrada a la planta, en el flujo de entrada al decantador, etc.), así que se creará un bloque en cada uno de esos puntos de medida, y en cada bloque estarán contenidas las variables que se han medido en ese punto, resultando un sistema dividido en 20 bloques.

Además, otra razón para hacerlo así, es que algunos de los métodos empleados, requieren que cada uno de los bloques contenga un determinado número de variables, ya que si alguno de ellos tiene diferente tamaño al resto, es imposible ejecutar los cálculos necesarios de método (aparecerían multiplicaciones de matrices imposibles al no cuadrar las dimensiones, etc.). De esta forma se asegura que todos los bloques tengan el mismo número de variables.

3.1. PCA con datos realistas

A la hora de trabajar en modo 'on-line' en una depuradora real no es posible disponer instantáneamente de las 21 variables vistas anteriormente (tabla 3.3), que son realmente 16 si no se usan las últimas 5, como es el caso que se va a ver, ya que los instrumentos de medida disponibles sólo son capaces de medir determinadas magnitudes. (Para conocer las variables anteriores se requeriría tomar muestras en diferentes puntos de la instalación y analizarlas en un laboratorio, lo que hace imposible tener un control constante y actualizado de la planta). Como, realmente sólo se pueden controlar en tiempo real algunas de esas variables, o combinaciones de las mismas, para las pruebas que se van a realizar se tomarán las siguientes magnitudes:

- DQO: demanda química de oxígeno en $\text{g COD}\cdot\text{m}^{-3}$. Es la suma de los valores de las 4 primeras variables: S_I (materia orgánica inerte soluble), S_S (substrato de biodegradación rápida), X_I (materia orgánica inerte insoluble) y X_S (substrato de biodegradación lenta).
- O_2 : oxígeno disuelto en el líquido en $\text{g (-COD)}\cdot\text{m}^{-3}$, se corresponde con el elemento 8 de la lista de variables
- S_{alk} : mide la alcalinidad del fluido, es la variable número 13
- N : mide el nitrógeno presente en el agua residual en $\text{g N}\cdot\text{m}^{-3}$, es la suma de las variables 9, 10 y 11; S_{NO} (contenido en nitratos y nitritos), S_{NH} (contenido de amoníaco y amonio) y S_{ND} (nitrógeno orgánico soluble asociado a S_S)
- SS : sólidos suspendidos en $\text{g N}\cdot\text{m}^{-3}$, corresponde a la variable número 14
- Caudal: caudal del fluido en $\text{m}^3\cdot\text{d}^{-1}$, variable número 15
- Temperatura: mide la temperatura del agua en $^{\circ}\text{C}$, es la variable 16 de la lista

De modo que se aplica el PCA pero sólo sobre estas 7 variables, y lo que se busca es conseguir aplicar un método PCA funcionando en tiempo real en una instalación depuradora.

3.2. Simulación de fallos en la instalación

Para analizar si la aplicación de un método PCA es efectiva para detectar fallos, necesitamos realizar simulaciones en las que el sistema trabaje en esas condiciones de fallo. Para ello se modificará ligeramente el esquema de la planta introduciendo elementos que cambien las condiciones de trabajo.

Uno de los fallos que se implementarán será el del fallo del sensor de oxígeno del reactor 4. Según el valor medido por este sensor se inyecta más o menos aire en los reactores 3,4 y 5. Por lo tanto si se falsea esta medida, el sistema de control aportará una cantidad inadecuada de oxígeno, provocando así un malfuncionamiento del sistema.

Otra fallo consiste en modificar algún valor de alguna variable en un punto determinado y en un instante determinado. Por ejemplo, se puede hacer que la alcalinidad del fluido en el instante “i”, a la salida del elemento “x” de la instalación, pase a a ser un 20 % mayor de su valor real, consiguiendo que el fluido que llega al siguiente elemento “x+1” no sea el que debería ser. De esta misma forma se podría simular una fuga de líquido de la instalación, haciendo que en el punto de la fuga, el caudal baje un determinado porcentaje (por ejemplo, a partir del instante “i” el caudal va a ser un 80 % del caudal normal).

Un tercer fallo puede ser simular una distribución inadecuada de caudales en un punto en el que se separe el caudal en dos (por un atasco, por un ajuste incorrecto de válvulas, etc.). Es decir si el flujo que viene de una tubería se separa en la tubería 1 y en la tubería 2, se puede hacer que el caudal del tubo 1 se incremente en la misma cantidad en la que desciende el flujo del tubo 2.

Veamos ahora en detalle cómo se han implementado estos fallos en el modelo de Simulink.

3.2.1. Fallo en el sensor de oxígeno:

Para simular el fallo en el sensor del oxígeno, se va a través de los bloques del modelo Simulink (figura 3.7) hasta encontrar los reactores biológicos (figura 3.8).

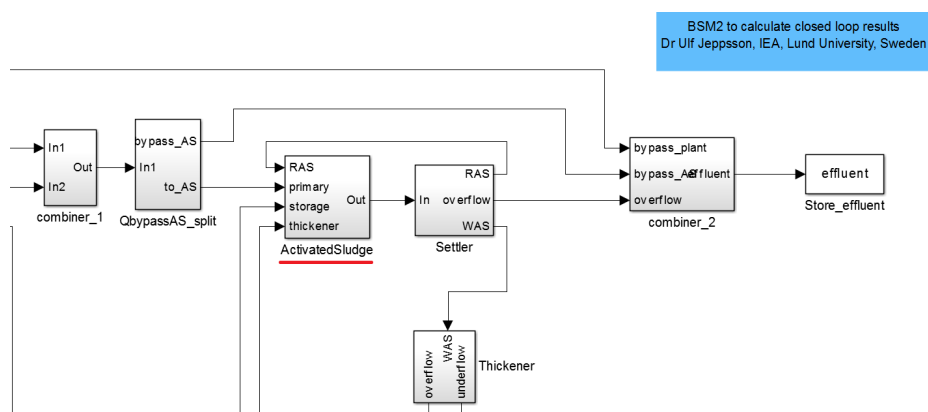


Figura 3.7: Bloque de reactores Activated Sludge Reactors

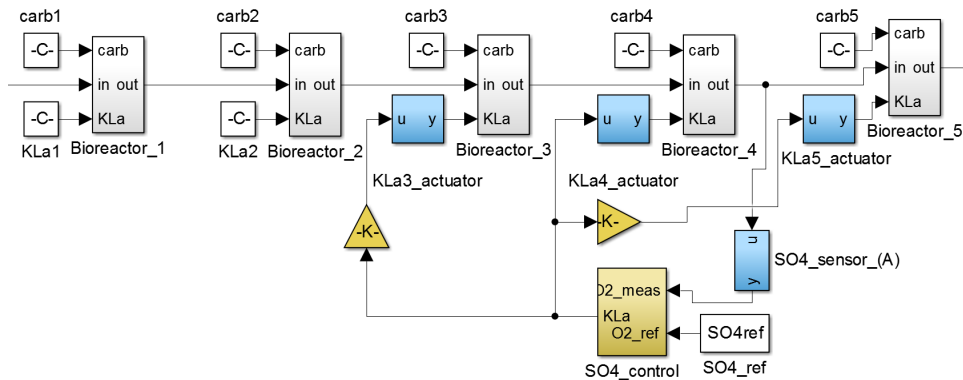


Figura 3.8: Bloques de los reactores en el elemento Activated Sludge

Si hacemos doble clic en el elemento “SO4_sensor” aparece lo que está en su interior (figura 3.9). Por un lado se ha marcado la parte que inyecta ruido en la medida del sensor para simular el comportamiento de un sensor real, en el que la medida se ve alterada por diversas fuentes de interferencia y ruido. Por otra parte, la medida entra al sensor a través de “u”, pasa por una serie de tratamientos en los que no nos vamos a detener, y, finalmente, la señal tratada sale por el elemento “y”.

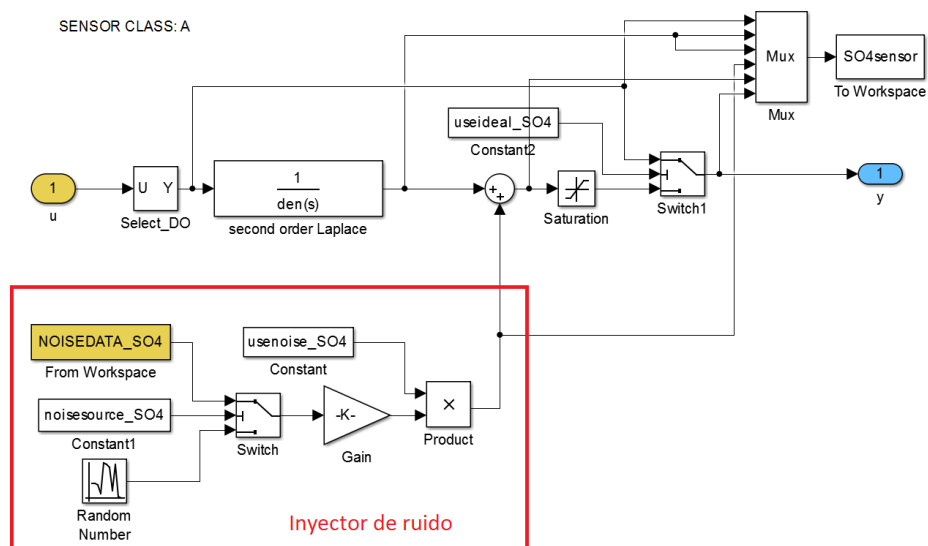


Figura 3.9: Sensor de oxígeno del reactor 4

Para falsear la medida del sensor, se han introducido en el esquema una

serie de bloques que buscan alterar la medida de oxígeno (figura 3.10). A la señal que debería ser la salida se le añade (usando un sumador) otra señal. Esta segunda señal se obtiene al multiplicar la señal original de salida, atenuada o incrementada por una ganancia, que corresponde la tamaño del fallo que queremos implementar, por una señal de tipo escalón que, en la figura, pasa de valer 0 a valer 1 en el instante en que se quiera provocar el fallo.

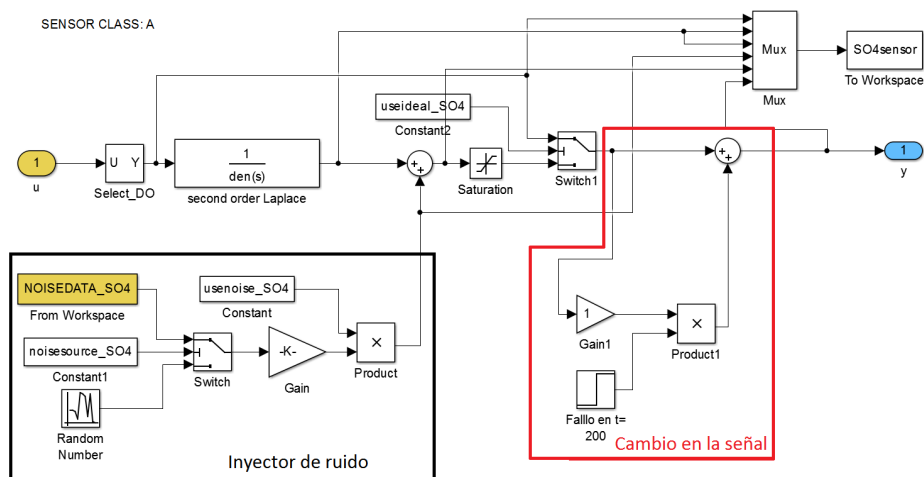


Figura 3.10: Sensor de oxígeno del reactor 4 con sistema de fallo

De esta forma, la salida del sensor permanece inalterada hasta el instante en el que la función escalón pasa a tomar un valor distinto de cero, momento a partir del cuál la salida cambia, y aparece la situación de fallo. Así podemos comprobar si el método PCA es capaz de detectar el fallo, ya que va a haber dos etapas claramente diferenciadas en la simulación, una primera sin fallo y una segunda con fallo. Una vez implementado, se puede lanzar la simulación para obtener los datos de las variables que se usarán luego para el PCA.

Para provocar este fallo se aumentará la lectura del sensor de oxígeno en un 50% y, en otra simulación, en un 100% (el control entenderá que hay exceso de oxígeno en el agua).

3.2.2. Variación del valor de un parámetro del fluido

Ahora se va a manipular algún dato de las características del fluido buscando simular algún mal funcionamiento del sistema (por ejemplo, se puede alterar el valor de la alcalinidad del fluido a la salida de algún reactor, para simular una anomalía en ese reactor, como puede ser algún vertido accidental de algún producto químico, etc.).

No es una forma demasiado realista de provocar un fallo, ya que se está variando artificialmente una característica del fluido a la salida de un elemento, simulando una anomalía en él, cuando realmente ese elemento del sistema sí está funcionando correctamente, pero sirve para comprobar si el PCA utilizado es capaz de detectar ese cambio en el parámetro modificado.

La forma de implementar esto es la siguiente: se interrumpe la línea que transmite la información del fluido entre dos bloques, se modifica la magnitud escogida para simular el fallo (alcalinidad, sólidos suspendidos, temperatura, etc.), y se envían los nuevos datos, ya modificados, al siguiente bloque. Esto no se puede hacer directamente ya que, como se dijo antes, la información del fluido se envía en bloques de 21 datos, de modo que primero se debe separar la magnitud que se va a modificar, alterarla, y luego volverla a juntar con el resto de magnitudes del fluido. En Simulink se puede hacer de la siguiente manera: se envía el bloque de 21 datos a unos bloques del tipo “Selector”, estos bloques toman, del grupo de 21 variables, sólo las que se les hayan indicado, de modo que algunos de estos “selectores” separarán los valores que no se modifican y otro u otros “selectores” tomarán la variable o variables a modificar para enviarlas a una serie de elementos que cambiarán sus valores (tal como se vio en la sección anterior). Una vez hecho esto, se reúnen las variables modificadas y las variables sin modificar usando un multiplexor que las agrupa otra vez en un conjunto de 21 variables (figura 3.11).

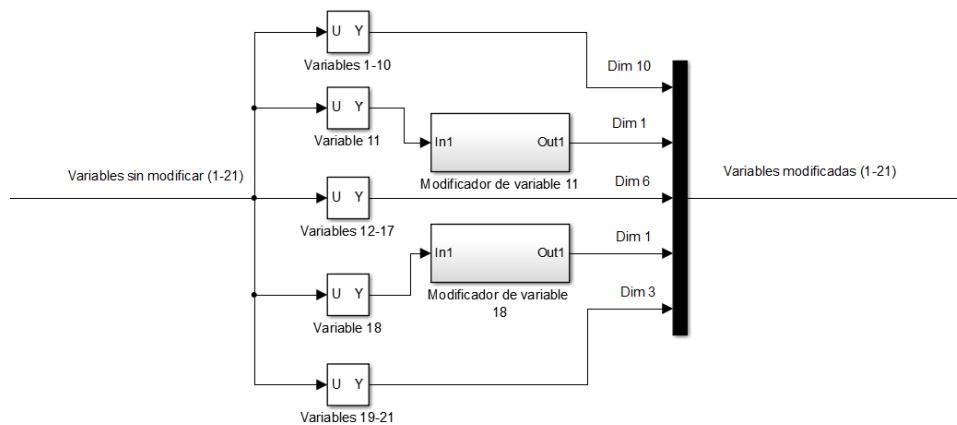


Figura 3.11: Forma de cambiar el valor de una variable del fluido

En el caso de la figura 3.11 se separan las variables que no se van a modificar en bloques (1-10, 12-17, 19-21), y por otro lado se separan las variables a modificar (11 y 18) para enviarlas, cómo se comentó antes, a unos bloques que van a alterar su valor. Una vez hecho esto, se reúnen los conjuntos de variables no modificadas con las variables modificadas usando

un multiplexor, cuya salida, con las 21 variables juntas, se envía al siguiente elemento del modelo. Es importante que en este multiplexor se unan las variables en el mismo orden en el que estaban antes de separarlas, para no provocar errores en las operaciones del sistema (figura 3.12).

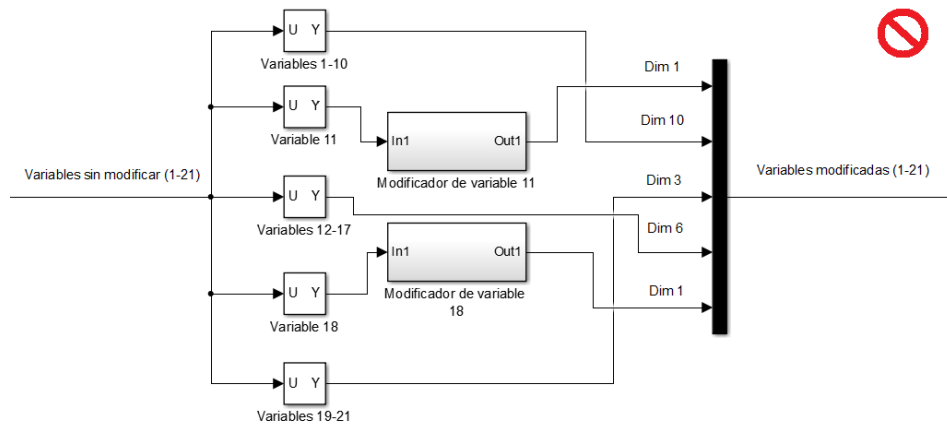


Figura 3.12: Se deben evitar alteraciones en el orden de las variables

Este tipo de fallos se implementará alterando el valor de la alcalinidad en el agua residual que entra a la planta (figura 3.13), simulando el vertido de algún químico que cambie el pH aguas arriba de la depuradora. Esto se consigue, como se explicó anteriormente, separando la variable S_{alk} del conjunto de 21 variables que caracterizan el agua residual que llega a la depuradora, modificándola, y luego volverla a unir al resto de variables. Se harán pruebas con diferentes valores de esta variable: se reducirá un 50 % y se aumentará un 20 % y un 40 %.

3.2.3. Fallos en la distribución del caudal por la instalación

Otro posible fallo que se puede implementar consiste en cambiar la distribución de caudal en una bifurcación de la instalación. Para hacer esto en el modelo de Simulink se disponen una serie de bloques de forma similar a como se ha hecho con los fallos anteriores, pero teniendo en cuenta que si el caudal que circula por uno de los tubos de la bifurcación disminuye, el caudal que va al otro tubo debe aumentar en la misma cantidad. Una de las pruebas que se ha hecho ha sido variar la distribución entre los caudales Q_r y Q_w a la salida del decantador secundario “Settler” (caudal de recirculación y de envío a tratamiento de fangos, respectivamente). En la figura 3.14 se puede ver el interior del bloque “Settler” antes de modificarlo, siendo la salida “RAS” el caudal Q_r y la salida “WAS” el caudal Q_w :

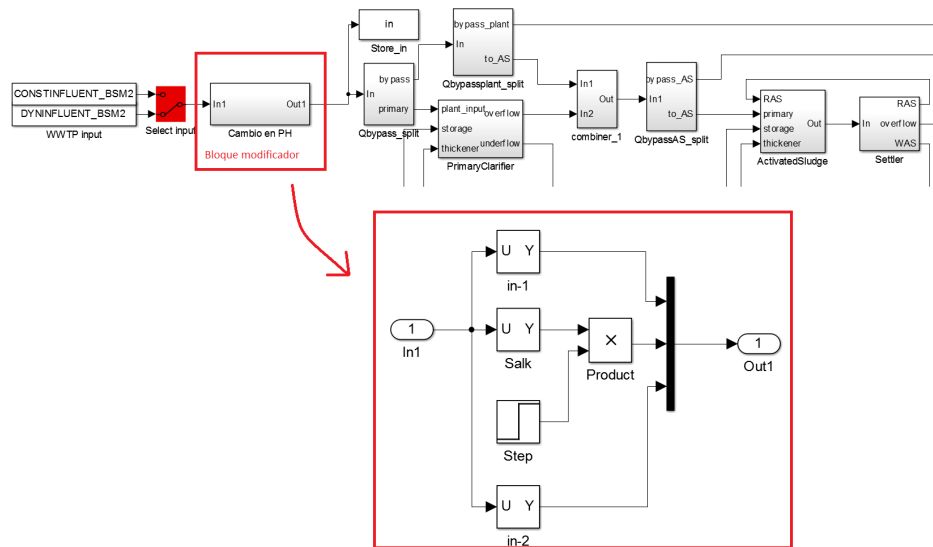


Figura 3.13: Modificación del valor de alcalinidad en el fluido que ingresa a la instalación

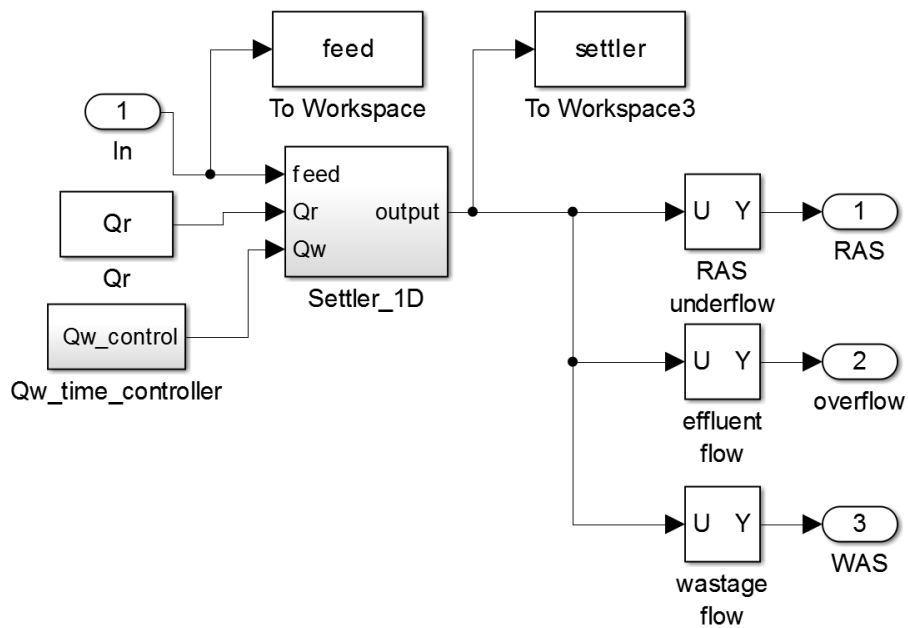


Figura 3.14: Contenido del bloque “Settler” sin modificar

Para introducir la modificación del caudal se colocan unos bloque mo-

dificadores antes de las salidas “RAS” y “WAS” (figura 3.15). En el bloque de modificación de Q_r se realiza una variación de la variable del caudal, y esta variación se envía a la salida “Out2” que llega a una de las entradas del bloque modificador de Q_w llamada “In2”. Esta entrada se suma al valor de la variable correspondiente al caudal en el grupo de 21 variables de “WAS” (para evitar ruido, errores, etc. se ha multiplicado esta entrada “In2” por una función Step que pasa de valer 0 a valer 1 en el mismo instante en el que el caudal Q_r cambia de valor en el otro bloque modificador) (ver figuras 3.16 y 3.17):

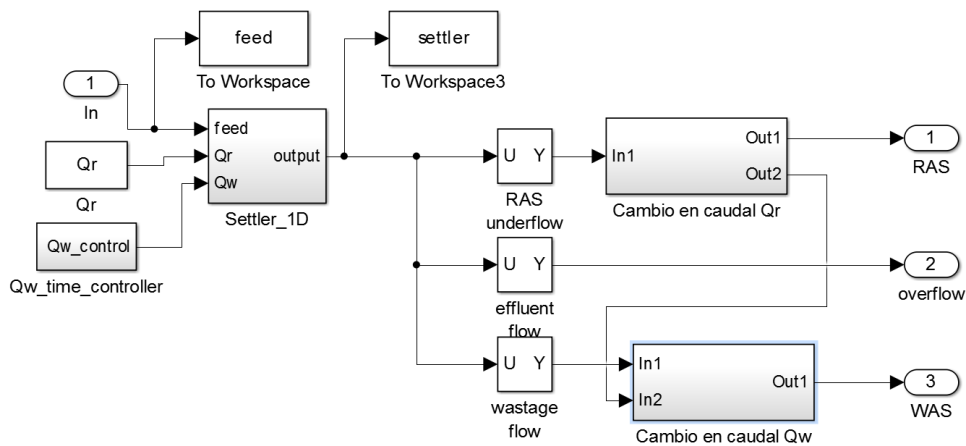


Figura 3.15: Modificación de los caudales Q_r y Q_s a la salida del decantador secundario

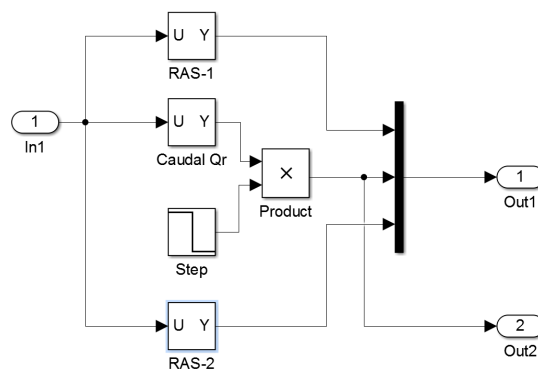


Figura 3.16: Bloque modificador del caudal Q_r

Así se consigue variar el reparto de caudales entre las dos tuberías si-

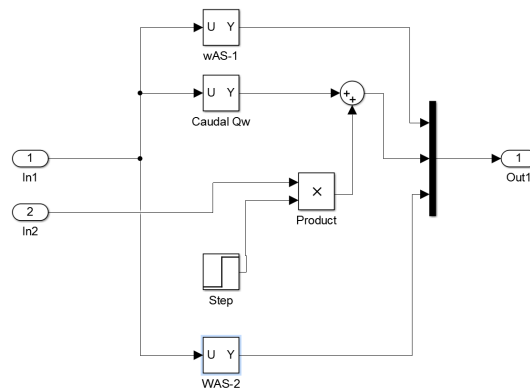


Figura 3.17: Bloque modificador del caudal Q_r , con la entrada “In2” procedente del bloque modificador de Q_r

mulando un atasco en alguna de ellas, la manipulación de alguna válvula, etc.

Otro fallo relacionado con cambios en los caudales consiste en simular una fuga en algún punto de la instalación. Para ello se recurre al método explicado antes en el apartado de “Variación del valor de un parámetro del fluido”. Para aplicarlo a este fallo simplemente hay que bajar el valor del parámetro del caudal. Por ejemplo, en la figura 3.18 se ve un bloque que se ha insertado a la salida del caudal inferior del decantador secundario. Ese bloque separa la variable del caudal, y a partir de un determinado instante modifica su valor para simular la fuga mencionada antes como se hizo en los fallos anteriores.

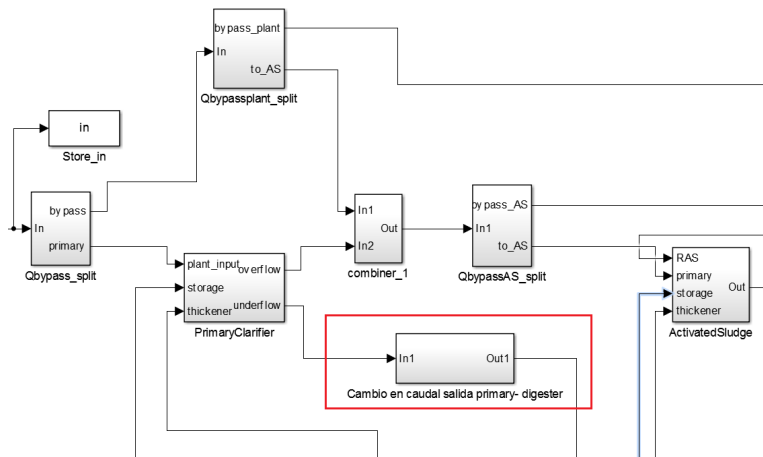


Figura 3.18: Bloque modificador del caudal de salida inferior del decantador primario

Los fallos referidos a caudales consistirán en un atasco y una fuga de líquido. Primero se va a simular un atasco en las conducciones a la salida de decantador secundario que provoquen un cambio en los valores de los caudales Q_r (caudal de recirculación de fango hacia los reactores) y Q_w (caudal hacia el área de tratamiento de fangos), de tal forma que se probará a disminuir Q_r un 50 % (cantidad que pasará a la conducción de Q_w) y, después, se aumentará Q_r un 50 % a costa de disminuir Q_w . La otra prueba consistirá en una fuga en la conducción que lleva el flujo inferior del decantador primario al digestor. Se simulará con una fuga del 30 % (70 % restante de caudal) y del 50 % (50 % restante de caudal).

Capítulo 4

Red de distribución de agua

Los métodos que se van a poner a prueba se aplicarán también sobre una red de distribución de agua potable que está implantada en un software de simulación de este tipo de instalaciones (EPANET) [11], y que nos va a aportar datos de diferentes variables del sistema en diferentes situaciones y en diferentes instantes. Esos serán los datos que se usarán en los cálculos (Manual del programa: [12]).

La red utilizada es la mostrada en la Figura: 4.1. Se ha tomado una red de tamaño medio, con varios puntos de entrada de agua, muchos puntos de consumo, y una red de tuberías que distribuirán el agua a los consumidores. Se realizarán simulaciones de la red en diferentes situaciones y se extraerán datos para realizar los análisis.

En algunos de estos análisis se tomará tal cuál está, trabajando con ella como un bloque, y en otros análisis se dividirá la red en varios bloques (Figura: 4.2) de los que se extraerá información por separado y, será a esta información a la que se aplicarán las técnicas de detección de fallos.

La red a estudio consta de diferentes elementos:

- nudos (representados por puntos), que son elementos de unión de varias tuberías y en los cuales se puede extraer agua hacia un punto de consumo, o introducir agua en la red. En nuestra red, todos los nudos son puntos de extracción de agua, excepto los nodos: 1, 17, 34, 36, y 38, por los que se introduce agua. El nodo 1 es el punto dónde se introduce una mayor cantidad de agua a la red, los otros nodos mencionados aportan cantidades más pequeñas.
- tuberías (representadas por líneas rectas), conducen el agua entre los nodos, tienen diferentes diámetros y longitudes en función de las necesidades de transporte del agua.
- reservas de agua (en el esquema es el elemento 1), permiten aportar agua al circuito.

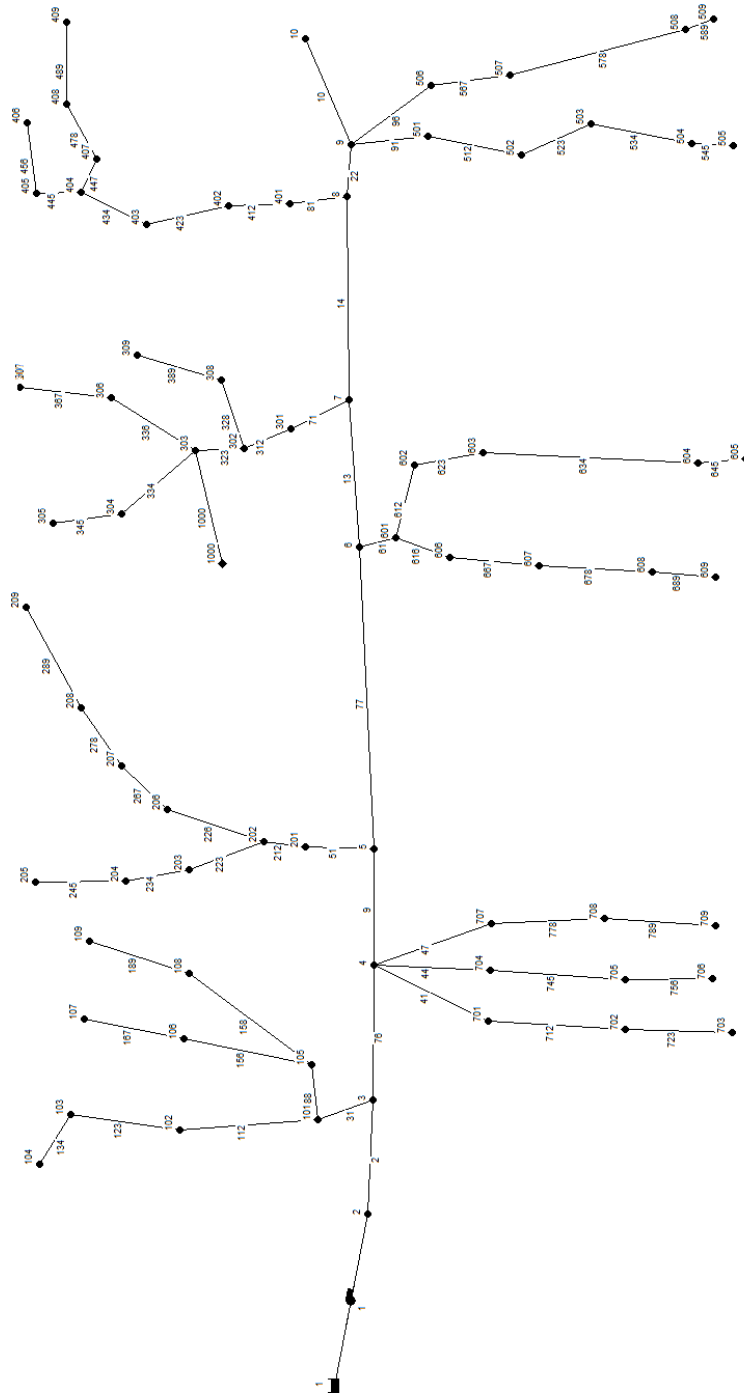


Figura 4.1: Red de agua

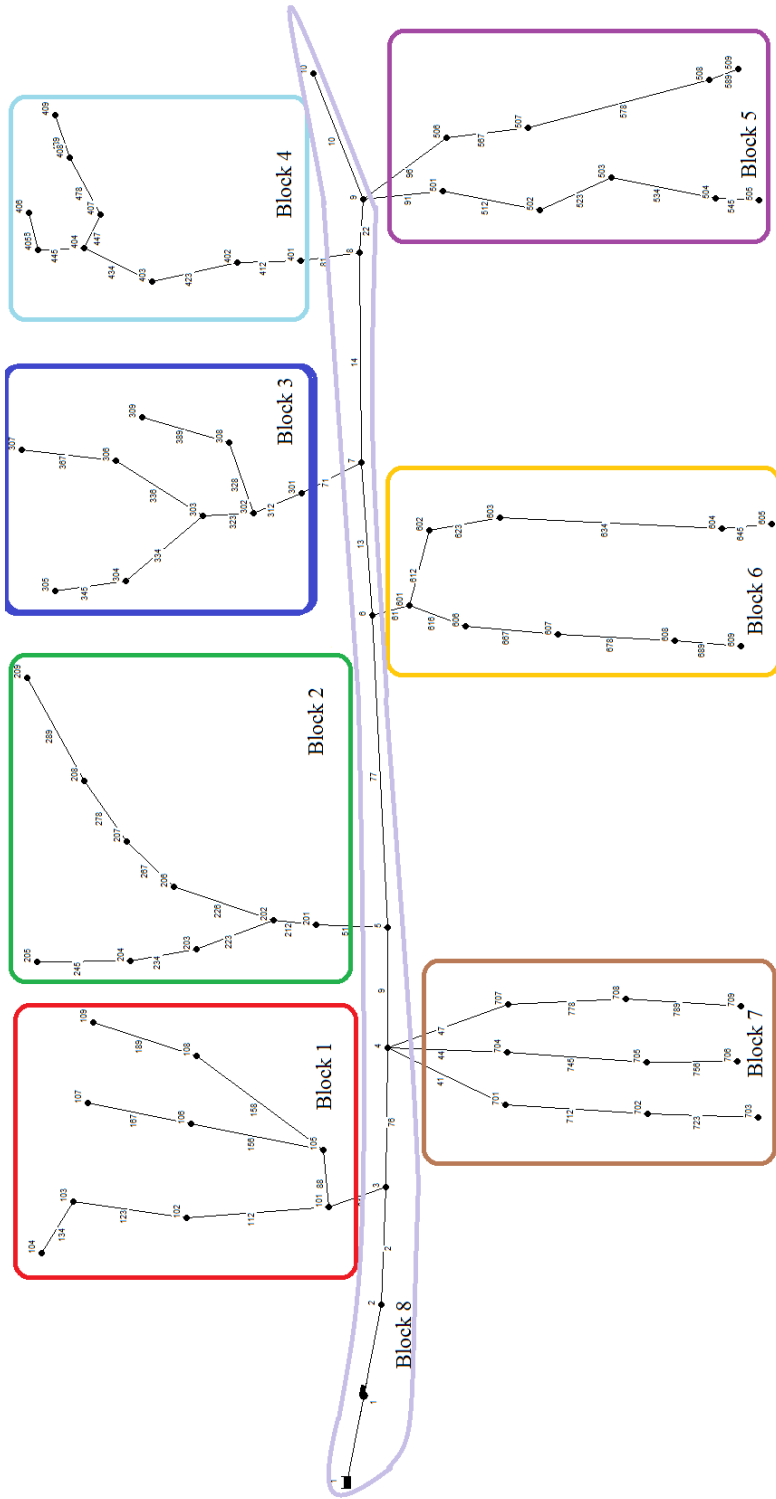


Figura 4.2: Red de agua por bloques

- bombas, son elementos que aportan energía al líquido y permite que éste circule correctamente por todos los tramos de la red (en este programa se representa la bomba sobre una tubería, en la red usada es la tubería 1)

A la hora de analizar el sistema, será necesario tomar datos en los nodos y en las tuberías de la red. Concretamente, se trabajará con los siguientes parámetros:

- Presión en el nodo
- Demanda de agua en el nodo (cantidad de agua que sale de la red por ese nodo, hacia un punto de consumo)
- Altura piezométrica en el nodo
- Concentración de aditivo en el nodo (en nuestro caso será flúor)
- Caudal en la tubería
- Velocidad del agua en la tubería
- Pérdida de altura piezométrica en la tubería (pérdidas por unidad longitud de tubería)
- Concentración media de aditivo en la tubería

Todos estos datos se reunirán en tablas, que antes de usarlos, habrá que revisar para eliminar variables que no aporten nada porque sean constantes o que puedan causar mal funcionamiento de los algoritmos, por ejemplo, divisiones por cero, etc. (columnas con varianza muy baja o cero).

Nota: el software EPANET no permite extraer los datos de simulación de todo el sistema, únicamente ofrece datos de un elemento (tubería, nodo, etc.) a lo largo de toda la simulación, o datos de todos los elementos del sistema en un determinado instante de la simulación. Así que para poder tener recopilado en una tabla toda la información requerida, se ha recurrido a las aplicaciones "AWARE – P software suite" creadas por el grupo BASEFORM [13], que permiten, a partir del modelo de la red creado con EPANET, simularlo y obtener los datos de todo el sistema, en todos los instantes de la simulación, dentro de un archivo de hoja de cálculo.

En estas pruebas, se realizarán dos tipos de simulaciones, en la primera se dejará que la red funcione en condiciones normales, y una vez que el sistema alcance un estado de funcionamiento normal, se extraerán unos valores que serán usados para definir las condiciones normales de funcionamiento de la red. Esta simulación durará 200 horas, y se tomarán datos desde la hora 100, a partir de la cuál el sistema ya habrá llegado a un estado normal de funcionamiento. Los datos se extraerán cada hora.

Las otras simulaciones consistirán en generar fallos en la red, una vez que esta se haya estabilizado, y ver qué ocurre con las variables del sistema. A los valores de estas variables, se les aplicaran técnicas de identificación y diagnóstico de fallos (PCA), para ver si es posible detectarlos.

4.1. Simulación de fallos en la instalación

Se probarán tres tipos de fallos:

- Fallo en la bomba
- Aparición de un contaminante
- Fuga en una tubería

Para los análisis, se tomarán datos entre las observaciones 100 y 200, de modo que se dispondrá de 100 datos, con el fallo apareciendo en el dato 50 (que sería el instante 150 de la simulación completa de 200 horas). En la simulación se ha introducido un patrón de variación en la demanda para simular cambios en el consumo de agua (figura 4.3), de modo que no se extrae de la red una cantidad constante de agua, sino que va variando a lo largo del tiempo; con esto se busca una mayor similitud entre la red simulada y una red real (en una red de agua potable, el consumo no es constante, si no que varía según la hora del día en la que nos encontremos).

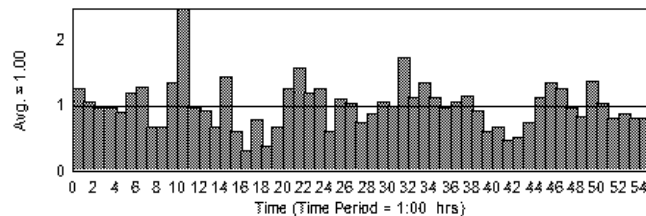


Figura 4.3: Patrón de demanda de agua

4.1.1. Fallo en la bomba

Para este tipo de fallo lo que se va a hacer es variar la velocidad de giro de la bomba (simulando alguna avería en el motor de la bomba, suciedad en la misma, etc.), lo que provocará un cambio en la energía suministrada al fluido. Esto se ha hecho variando el parámetro de velocidad relativa de la bomba, mediante un control simple del programa EPANET, en el instante 150 de la simulación (de duración total de 200 horas).

4.1.2. Aparición de un contaminante

Ahora se va a simular la aparición de una sustancia contaminante en la red a partir del instante 150. La forma de hacer esto (figura 4.4) es colocar un depósito (elemento 1000) con agua con una determinada concentración de contaminante, unido mediante una tubería (link 1000) al nodo 6 de la red. Esta tubería estará inicialmente cerrada, pero a partir del instante 150 (en una simulación de 200 horas) se abrirá introduciendo contaminante en la red.

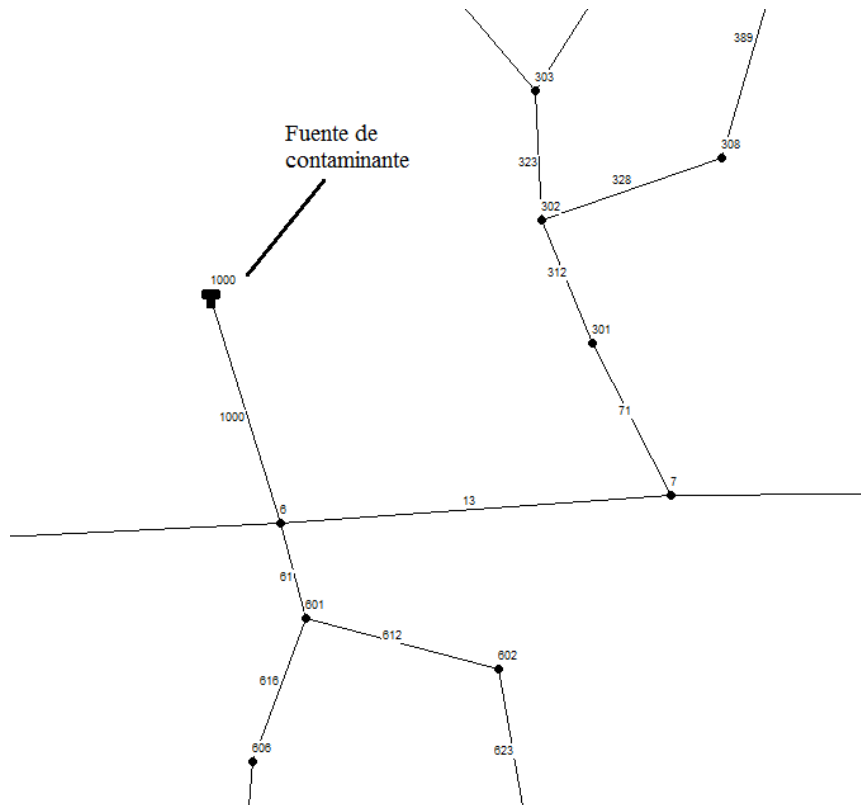


Figura 4.4: Introducción de contaminante a la red

4.1.3. Fuga en una tubería

El fallo va a consistir en simular una fuga en un nodo del bloque 3 (concretamente el nodo 303). Para simular la fuga se ha dispuesto en ese nodo, una tubería que finaliza en un nodo (figura 4.5). Esta tubería, que normalmente estará cerrada, en el momento en el que se decida que aparezca la fuga se abrirá, produciendo una pérdida de agua por el nodo final de la misma. Este nodo se ha implementado para que funcione como un aspersor

(es decir, la cantidad de agua que salga por él dependerá de la presión de la red: a más presión, más salida de agua). Esta simulación también durará 200 horas y se extraerán datos desde la hora 100. El fallo aparecerá en la hora 150 (150 horas desde el inicio de la simulación).

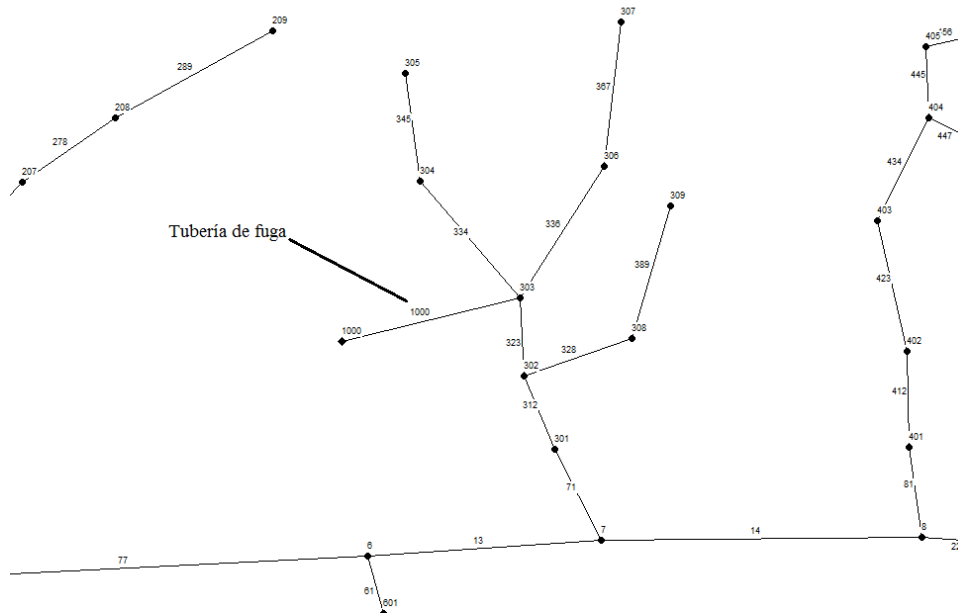


Figura 4.5: Tubería de fuga de agua

Capítulo 5

Resultados y conclusiones. Depuradora.

En este apartado se pretende dar una visión general del funcionamiento de diversos métodos distribuidos de detección de fallos aplicados a una planta depuradora de aguas residuales.

Para todos los métodos se ha reducido el número de variables presentes en la simulación a un conjunto de magnitudes similar a las que se podrían medir en una planta real instantáneamente. En el caso del método PCA centralizado, se analizará el sistema en bloque (que tiene en cuenta conjuntamente las 7 variables medidas en cada uno de los 20 puntos de medición, dando un total de 140 variables). Al aplicar los métodos PCA distribuido y CPCA se usará un sistema dividido en los 20 bloques explicados en la sección relativa a la descripción de la depuradora (sección: 3) (en el caso del CPCA también se aplicará a nivel global), conteniendo cada bloque 7 variables; aunque estos métodos permiten mucha flexibilidad a la hora de elegir los bloques, y se podrían agrupar los elementos de muchas otras formas. Con los métodos Merged PCA Distributed, PCA distribuido con factorización QR y PCA distribuido con factorización QR y agrupamiento en clústers, se divide el sistema, también, en 20 bloques conteniendo cada uno un sensor que mide las 7 variables realistas (DQO, SS, O_2 , etc...). Pero en estos casos, estas tres últimas técnicas requieren que en cada bloque analizado haya el mismo número de nodos, y aunque se podrían agrupar los sensores en varios bloques, es complejo encontrar agrupaciones con la misma cantidad de sensores entre los que haya cierta relación (no tiene mucho sentido agrupar nodos de cualquier forma, se debe hacer siguiendo algún criterio: influencia entre ellos, topología, etc. . .), de modo que se ha decidido crear un bloque para cada sensor (con 7 variables medidas en cada uno de ellos).

Todos los métodos se aplican con un porcentaje del $\alpha = 70\%$ de la varianza total representada por sus componentes principales, este valor fue elegido por proporcionar buen rendimiento y un número reducido de com-

ponentes principales. Después de probar algunos valores se tomó el valor de 83 observaciones consecutivas por encima del límite de los estadísticos para que salte la alarma. Para el caso concreto del método de clustering, se utilizó un radio para los clústers de $w_c = 0,1$ y se ha establecido que para que salte la alarma en un bloque debe haber, al menos, un 10 % de los clústers por encima del umbral.

Se van a implementar 16 fallos diferentes (4 tipos de fallo con diferentes magnitudes) y se va a comprobar cómo se comportan los distintos métodos a la hora de detectarlos. Se tomarán los datos de un fallo y se aplicarán los diferentes métodos para comparar los resultados que ofrece cada uno. El instante de aparición del fallo variará entre $t = 150$ días y $t = 350$ días dependiendo de la simulación. Aunque la simulación dura 609 días y genera 58464 observaciones, se tomarán datos a partir de la observación 10001, para eliminar posibles perturbaciones que pudieran aparecer al inicio.

Durante las pruebas se va a medir el porcentaje de observaciones anómalas en ausencia de fallo mediante el índice OTI (Overall Type I risk),

$$OTI = \frac{n_f}{I_{NOC}} * 100$$

donde n_f es el número de observaciones por encima del límite cuando no hay fallo, y I_{NOC} es el número total de observaciones también en situación sin fallo. Además, para evitar falsas alarmas en condiciones normales, se impone que sólo si alguna de los estadísticos, sobrepasa los límites durante un cierto número de observaciones consecutivas se activa una alarma. Ese número se debe ajustar para no tener falsas alarmas durante la parte de las simulaciones en la que aún no hay fallo. Esto se mide usando el índice TI (Type I risk),

$$TI = \frac{n_{fa}}{n_s}$$

donde n_{fa} cuenta el número de simulaciones con falsas alarmas en el período de la simulación en el que aún no hay fallos, y n_s es el número de simulaciones realizadas.

5.1. Ejemplo: Fallo en el sensor de oxígeno de los reactores

Se va mostrar ahora un ejemplo práctico en el que se aplican los diferentes métodos estudiados a los datos de la instalación cuando se ha producido un fallo en ella. Este fallo consiste en falsear la medida del oxígeno disuelto en el tanque 4, de forma que el control calcule un valor de oxigenación para los tanques 3, 4 y 5 inadecuado. (Ver sección 3.2.1).

Se van a provocar los siguientes fallos: se aumentará la lectura de este sensor un 50 %, un 100 % y un 200 % (el control entenderá que hay exceso

de oxígeno en el agua), y se reducirá esa lectura en un 20 % y un 80 % (se verá como un defecto de oxígeno).

Como ejemplo, se muestra cómo funciona cada método al ser aplicado sobre los datos de la situación de fallo consistente en un incremento del 50 % del valor medido por el sensor de oxígeno, con gráficas de los valores de los estadísticos, y tablas con los resultados obtenidos.

Incremento del 50 % del valor del sensor

El fallo aparecerá en $t = 300$ días, que corresponderá con la observación 18799 (ya descontadas las 10001 observaciones eliminadas) y está localizado en la variable S_o del reactor 4. (Nota las gráficas se han representado tomando un dato de cada 100, para clarificar la representación, ya que si no no se podría apreciar correctamente la evolución de los estadísticos).

Método PCA centralizado

Variables: 133 (116 variables al eliminar las que tienen varianza cero). 4 componentes principales (con al menos un 70 % de la varianza).

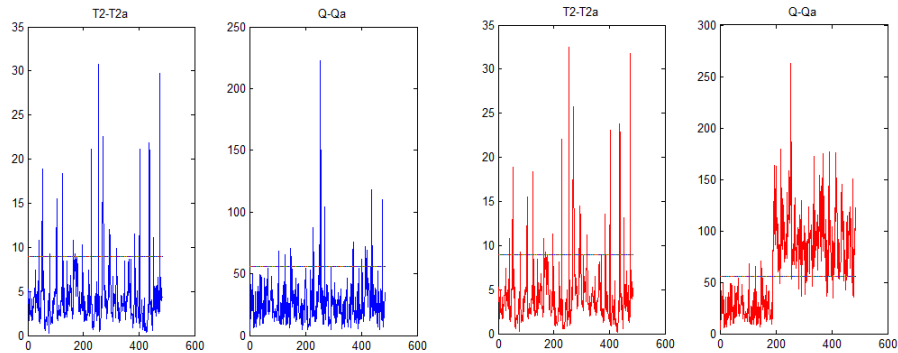


Figura 5.1: PCA centralizado con datos realistas. Fallo 50 %. Simulación sin fallo - Simulación con fallo

Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Contribución T^2	Contribución Q
-	19271	-	'react4 O2'	-	7.4230

Cuadro 5.1: Resultados. Método PCA centralizado

El estadístico T^2 no detecta el fallo. Q sí lo hace en el instante 19271, la variable más influyente es O_2 (oxígeno disuelto del fluido) a la salida del reactor 4 con una contribución de 7.4230.

Método PCA distribuido

No se representarán las gráficas de todos los bloques, sólo se mostrarán aquellas en las que se observe algún fallo.

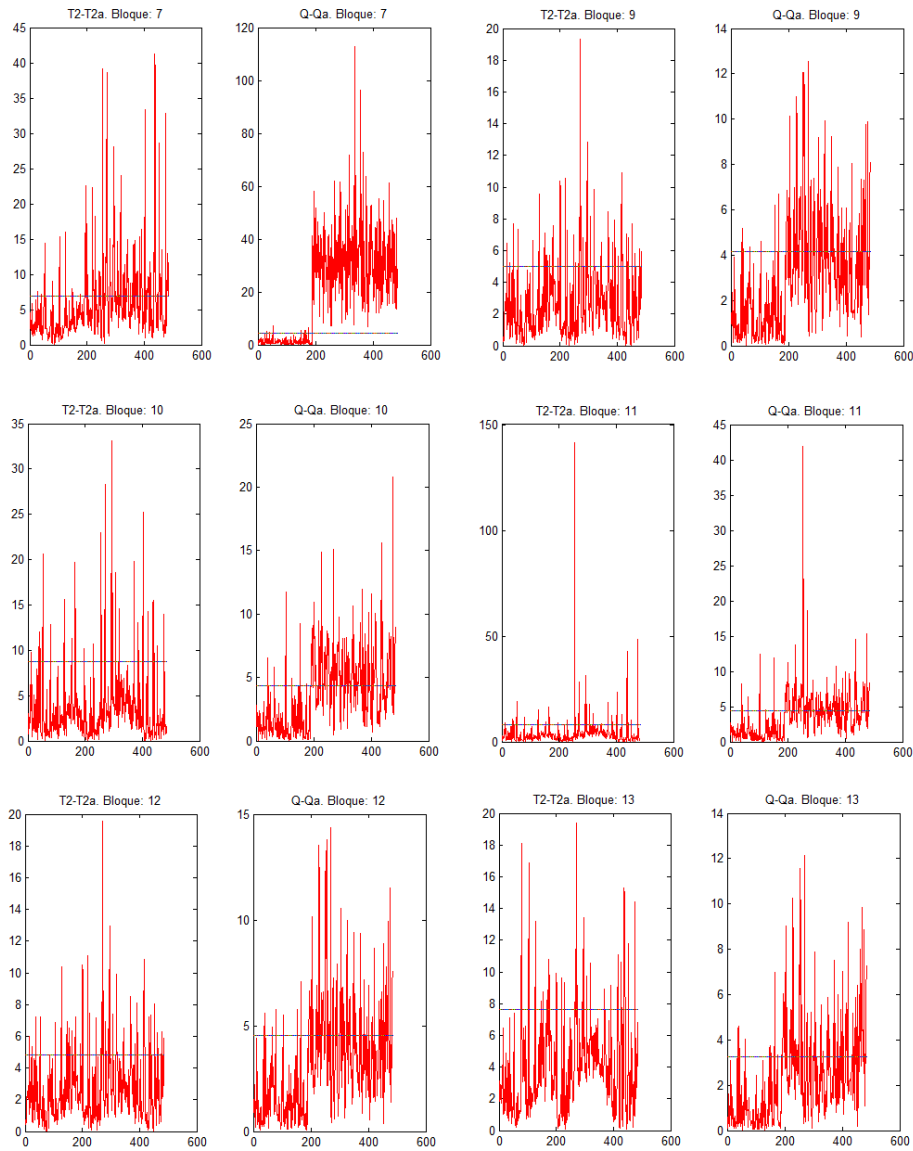


Figura 5.2: PCA distribuido. Fallo 50 %. Simulación con fallo

Se observa que el estadístico T^2 , aunque llega a superar su umbral, no lo hace claramente en ningún bloque, ya que oscila mucho. En cambio, el

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	47466	-	'in Flow'	-	2,6495
2	-	-	-	-	-	-
3	-	-	-	-	-	-
4	26842	-	'react1 Flow'	-	3,2230	-
5	-	-	-	-	-	-
6	-	26897	-	'react3 Flow'	-	3,6823
7	25358	18815	'react4 O2'	'react4 O2'	4,6409	7,0132
8	25362	-	'react5 Flow'	-	6,96735248	-
9	26919	-	'settler-underflow O2'	-	1,6037	-
1-	25388	-	'settler-effluent Flow'	-	3,3351	-
11	-	-	-	-	-	-
12	26927	-	'thickener-overflow Salk'	-	4,0369	-
13	43652	-	'thickener-underflow Flow'	-	8,6893	-
14	-	47466	-	'digester-influent Flow'	-	2,6986
15	2702-	-	'digester-effluent SS'	-	1,5743	-
16	-	-	-	-	-	-
17	-	-	-	-	-	-
18	-	-	-	-	-	-
19	-	-	-	-	-	-
20	27003	-	'WWTP sludge Salk'	-	1,8870	-

Cuadro 5.2: Resultados. Método PCA distribuido

estadístico Q, supera claramente su límite en el bloque 7, mientras que en el resto de bloques, al igual que T^2 , oscila arriba y abajo de esa cota. Si ahora miramos los resultados de cada bloque que se muestran en la tabla 5.1, podemos observar que no todos los bloques detectan fallo sino solamente algunos de ellos. Esta información es enviada al nodo central que la analiza de la siguiente forma: primero busca en qué bloques se detecta antes el fallo, es decir, aquellos cuyo tiempo de detección sea más bajo y, en concreto, se ve que es para la estadística Q en el bloque 7 (sólo 16 observaciones después de producirse el fallo), y coincide con el valor de la máxima contribución que en este caso es 7,013 y la variable responsable del fallo es el O_2 del reactor 4.

Método PCA distribuido con factorización QR

En este método también se analizan los datos de la planta dividida en bloques, por lo tanto se obtendrán resultados y gráficas para cada uno de ellos. Pero, por simplicidad, sólo se va a representar la gráfica del bloque 7, ya que es el bloque en el que se detecta antes el fallo. En algunos de los otros bloques también aparece un fallo, pero lo hace muy tarde y no son relevantes para el proceso de detección e identificación, como se muestra en la tabla 5.1. Tampoco se van a representar las gráficas del sistema trabajando en condiciones normales, ya que no aportan ninguna información de importancia.

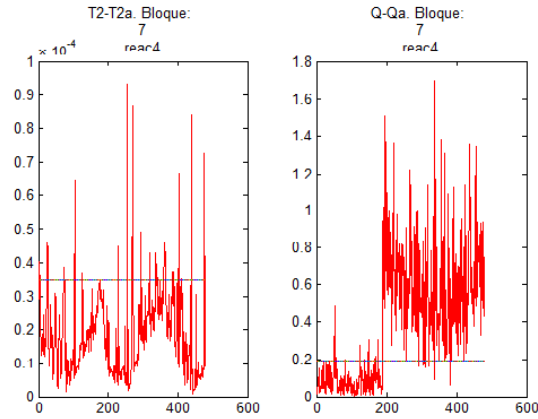


Figura 5.3: Factorización QR. Fallo 50 % en el sensor de oxígeno.

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	-	-	-	-	-
2	-	27468	-	'primaryout-overflow Temp'	-	0,4967
3	-	27468	-	'primaryout-underflow Temp'	-	0,4967
4	-	25135	-	'react1 Flow'	-	0,4581
5	-	25181	-	'react2 Flow'	-	0,5262
6	-	-	-	-	-	-
7	-	18910	-	'react4 O2'	-	0,8117
8	-	-	-	-	-	-
9	-	-	-	-	-	-
10	-	27463	-	'settler-effluent Temp'	-	0,52762
11	-	-	-	-	-	-
12	-	-	-	-	-	-
13	-	27467	-	'thickener-underflow Temp'	-	0,5298
14	-	27466	-	'digester-influent Temp'	-	0,4980
15	-	-	-	-	-	-
16	-	-	-	-	-	-
17	-	-	-	-	-	-
18	-	-	-	-	-	-
19	-	-	-	-	-	-
20	-	-	-	-	-	-

Cuadro 5.3: Resultados. Método PCA con factorización QR

El estadístico Q detecta un fallo en la observación 18910 (el fallo aparece en 18799) en el nodo “*react₄*”, variable “*O₂*” con contribución 0.8117. En otros bloques también se detecta fallo, pero, como se dijo, lo hace muy tarde. Por otro lado, se ve que el estadístico T^2 no detecta ningún fallo en ningún bloque.

Método CPCA

El método trabaja con 3 componentes principales a nivel global (con un

70 % de la varianza).

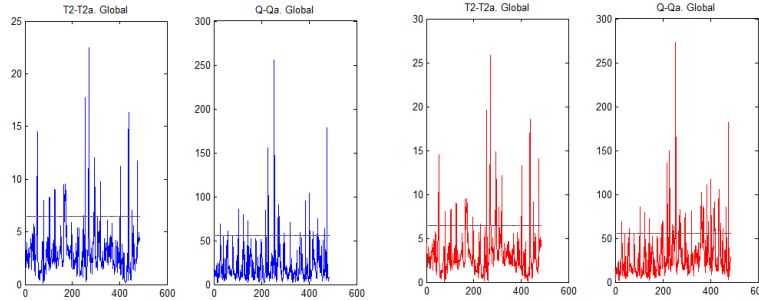


Figura 5.4: CPCA. Fallo 50 %. Simulación sin fallo - Simulación con fallo

Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Contribución T^2	Contribución Q
26836	25378	'react4 Salk'	'react4 Salk'	0,3678	4,6354

Cuadro 5.4: Resultados. Método CPCA

En la tabla 5.1 se pueden ver los resultados obtenidos, donde se ve que T^2 detecta un fallo en la observación 26836 (el fallo se produce en la 18799) y señala a la alcalinidad del reactor 4 (su contribución es de 0.3678). Q detecta un fallo en el instante 25378 y la variable más influyente es S_{alk} (alcalinidad) en “react4”, con contribución de 4.6354. Parece que el método no puede detectar (es muy lento) y diagnosticar correctamente el fallo.

Método Merging PCA distributed

Los resultados se muestran en las siguientes gráficas donde sólo se representan las estadísticas T2 y Q para los bloques que detectan el fallo.

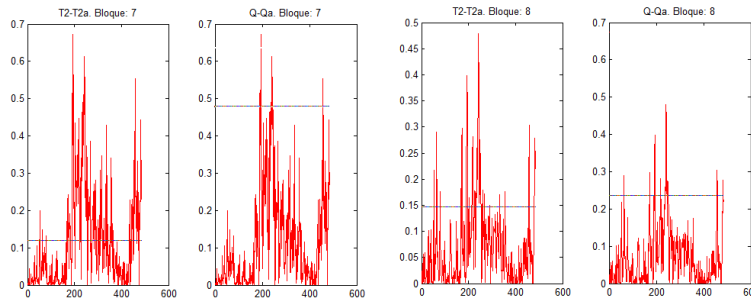


Figura 5.5: Merging PCA. Bloques 7 y 8.

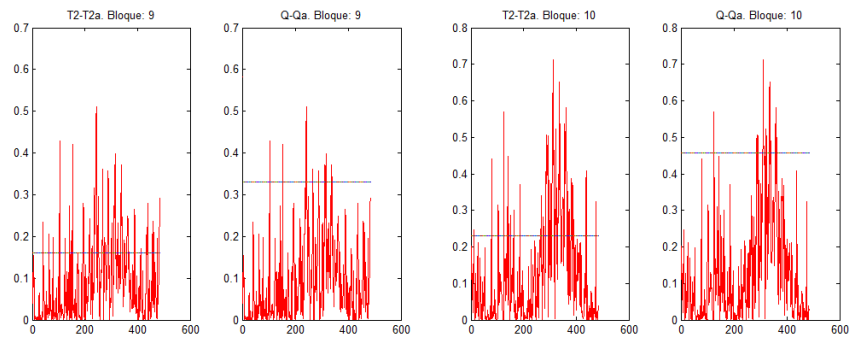


Figura 5.6: Merging PCA. Bloques 9 y 10.

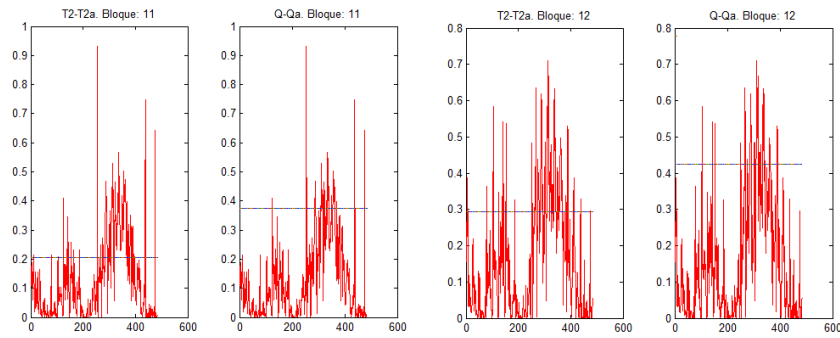


Figura 5.7: Merging PCA. Bloques 11 y 12.

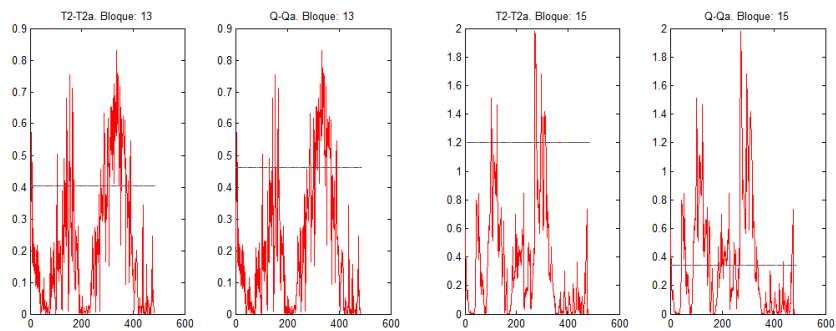


Figura 5.8: Merging PCA. Bloques 13 y 15.

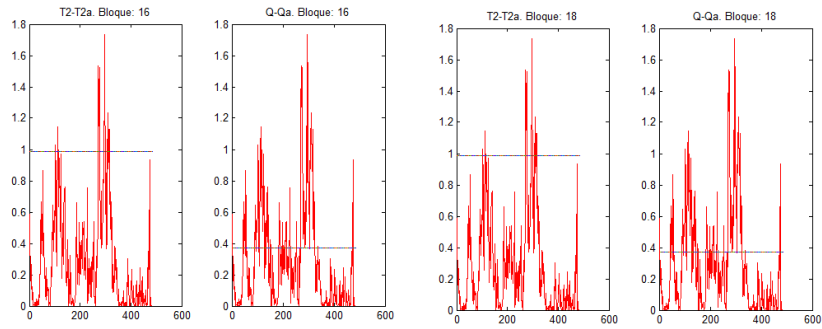


Figura 5.9: Merging PCA. Bloques 16 y 18.

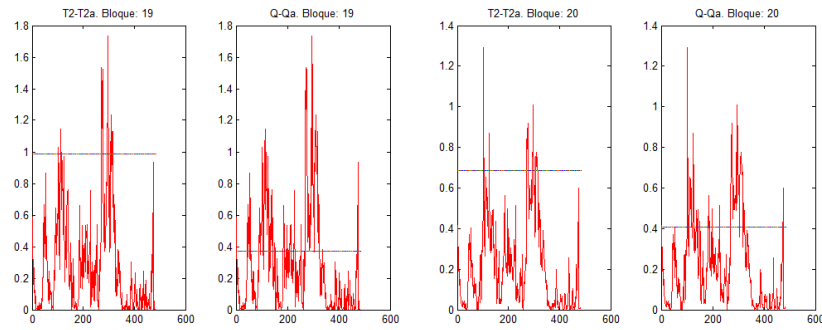


Figura 5.10: Merging PCA. Bloques 19 y 20.

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	-	-	-	-	-
2	-	-	-	-	-	-
3	-	47466	-	'primaryout-underflow Flow'	-	0,40453
4	23550	25363	'react1 DQO'	'react1 Flow'	1,17E-06	0,5682
5	-	-	-	-	-	-
6	-	-	-	-	-	-
7	18802	25362	'react4 Temp'	'react4 O2'	4,46E-06	0,6190
8	23546	25364	'react5 N'	'react5 Flow'	2,83E-06	0,6411
9	23552	25380	'settler-underflow N'	'settler-underflow O2'	1,35E-05	0,5569
10	29449	25360	'settler-effluent Salk'	'settler-effluent O2'	3,76E-05	0,4685
11	28875	25360	'WWTP effluent Temp'	'WWTP effluent O2'	9,54E-06	0,4684
12	27437	25382	'thickener-overflow Temp'	'thickener-overflow O2'	1,44E-06	0,5893
13	28686	25380	'thickener-underflow Temp'	'thickener-underflow O2'	1,05E-05	0,5569
14	-	25377	-	'digester-influent O2'	-	0,3663
15	26948	26851	'digester-effluent N'	'digester-effluent Flow'	4,31E-05	0,3932
16	27040	26856	'dewatering-effluent N'	'dewatering-effluent Flow'	7,05E-05	0,45560
17	-	-	-	-	-	-
18	27040	26856	'storage-bypass N'	'storage-bypass Flow'	7,05E-05	0,4556
19	27040	26856	'storage output+bypass N'	'storage output+bypass Flow'	7,05E-05	0,4556
20	27136	36268	'WWTP sludge N'	'WWTP sludge Temp'	6,92E-05	0,3662

Cuadro 5.5: Resultados. Método Merging PCA

Se ve en la tabla 5.1 que el estadístico Q detecta el fallo y pero no lo identifica correctamente, al tener los bloques 10 y 11 la máxima rapidez en detectar el fallo, pero éste aparece en el bloque 7, ya que se está variando la medida del oxígeno disuelto en el tanque 4 (bloque 7). Por otro lado, Q tarda mucho tiempo en hacer la detección. T^2 detecta el fallo más rápido pero no lo identifica correctamente. En el resto de bloques se detectan algunos fallos pero mucho más tarde de producirse y además con gran cantidad de falsas alarmas.

Método PCA distribuido usando factorización QR y agrupamiento de datos en clústers

En este caso se obtiene el número de clusters por encima del umbral para cada bloque para el instante de la detección y la evolución de ese valor para el bloque responsable del fallo (sólo se representan unas observaciones por delante y por detrás del fallo para mayor claridad de las gráficas) como se ve en las gráficas 5.11 y 5.12 :

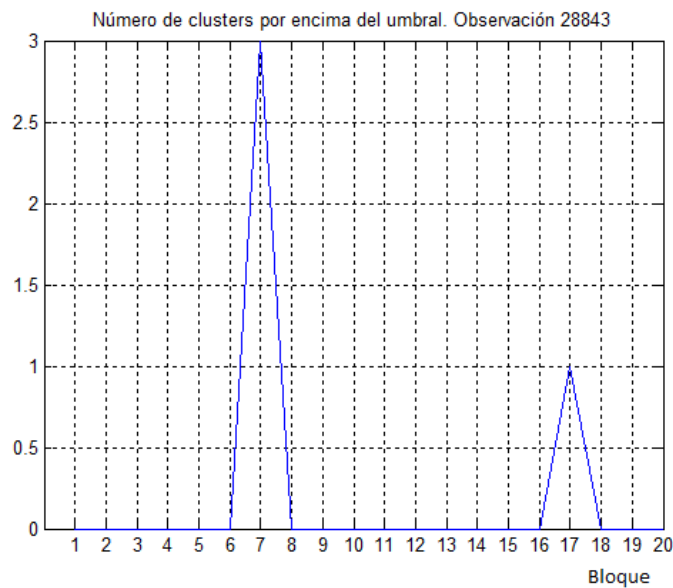


Figura 5.11: Fallo 50% en el sensor de oxígeno. Número de clusters por encima del umbral en el instante del fallo para cada bloque.

Aquí sólo aparecen clusters por encima del umbral (más del 10% de los clusters por encima de la distancia máxima) en el nodo reac4, en la observación 28843 (el fallo aparece en la 28799, en este caso no se han eliminado las 10001 primeras observaciones), la identificación apunta a la variable “Temp”

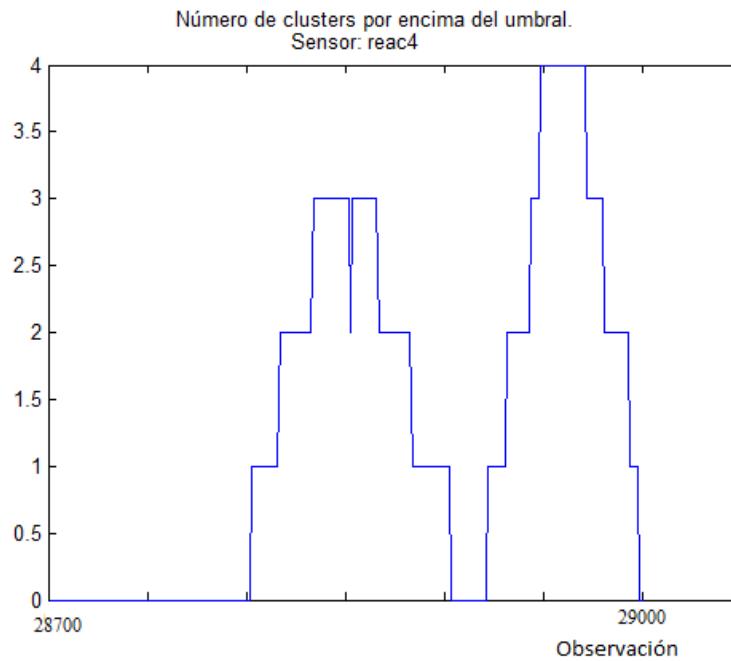


Figura 5.12: Fallo 50 % en el sensor de oxígeno. Evolución del número de clusters anómalos del reactor 4.

Tiempo	Sensor	Variable	Falsas alarmas
28843	'reac4'	'Temp'	0

Cuadro 5.6: Resultados. Método QR con clusters

que no es la que realmente provoca el fallo, así que se puede decir que detecta el fallo pero no lo identifica bien, como se ve en la tabla 5.1 (probablemente debido a que la normalización aplicada a la matriz de datos X no es adecuada para realizar la identificación).

5.2. Resultados y conclusiones

Se exponen aquí los resultados de aplicar los métodos de detección de fallos expuestos antes en la planta de la depuradora, con las condiciones ya especificadas, y se han ejecutado 16 simulaciones, 4 tipos diferentes de fallos con diferentes magnitudes. Los datos obtenidos son, para cada método: fallos detectados por cada una de las estadísticas consideradas T^2 y Q , fallos identificados, índice de falsas alarmas (FI), índice de observaciones anómalas (OTI) y retraso en la detección.

Los resultados obtenidos se agrupan en las tablas: 5.7 y 5.8.

Cuadro 5.7: Fallos detectados e identificados, y OTI (tasa de falsas alarmas)

Método	Fallos detectados		Fallos identificados		OTI	
	T^2	Q	T^2	Q	T^2	Q
PCA Centralizado	16	16	9	15	1.16	1.26
PCA Distribuido	16	16	11	15	0.97	0.86
PCA Colectivo	16	15	0	0	3.67	0.84
Merging PCA	16	16	10	13	3.82	5.02
DPCA (QR)	13	16	6	12	4.18	9.56
DPCA & Clustering	16		11		0	

Cuadro 5.8: Tiempo de detección y TI

Método	Retraso en la detección (no. observaciones)		TI (% de pruebas)	
	T^2	Q	T^2	Q
PCA Centralizado	1429.8	404.6	0 % (0/16)	6.25 % (1/16)
PCA Distribuido	558.6	5.13	6.25 % (1/16)	6.25 % (1/16)
PCA Colectivo	2063.87	1151.93	0 % (0/16)	0 % (0/16)
Merging PCA	9.6	444.8	62.5 % (10/16)	100 % (16/16)
DPCA (QR)	108.92	166.8	25 % (4/16)	100 % (16/16)
DPCA & Clustering	28.73		0 % (0/16)	

En general, se puede decir que, para todos los métodos, el estadístico T^2 funciona peor que el Q. Suele detectar menos fallos, e identifica también menos. Además de ser, en general, más lento en la detección. Se ha observado que el fallo del sensor de oxígeno es el más difícil de detectar, de hecho, cuando algún método lo hace, tarda bastante más que cuando suceden otros fallos (esto ha penalizado la rapidez de algún método, ya que es capaz de detectar el fallo, pero lo hace muy tarde, mientras que otros métodos, que son peores ya que no lo detectan, no aumentan su retraso en la detección, viéndose beneficiados en este sentido).

Si se analiza método a método, se puede ver que el método centralizado con parámetros reales, es capaz de detectar con T^2 y Q todos los fallos, e identifica el 37.5 % y el 93.75 %, también respectivamente, de ellos. T^2 no presenta simulaciones con falsas alarmas y Q tiene una; los retrasos en la detección son de 1429.8 y 404.6 observaciones, respectivamente, datos no muy buenos. Es un método que funciona, aunque es mejorado por los otros que se han probado.

El método distribuido propuesto mejora los datos anteriores, ya que T^2 y Q detectan también todos los fallos, y además, identifican el 68.75 % y el 93.75 %. Sólo aparece una simulación con falsas alarmas en Q, y los retrasos medios son de 558.6 y 5.13 observaciones, respectivamente. Es decir, mejora bastante al método anterior, con la desventaja de tener que contar con un sistema de procesamiento de datos en cada nodo.

El método CPCA no presenta resultados buenos, ya que detecta casi todos los fallos pero no consigue identificarlos. No tiene simulaciones con falsas alarmas, pero los retrasos medios son muy altos, de 2063.87 y 1151.93 observaciones, respectivamente. Es el peor método de los evaluados aquí.

El método Merging PCA puede detectar con T^2 y Q el 77.78 % y el 100 %, respectivamente, de fallos, y puede identificar el 62.5 % y el 81.25 %. Presenta muchas falsas alarmas y los retrasos medios son de 9.6 y 444.8 observaciones, respectivamente.

La técnica de Factorización QR permite detectar con Q todos los fallos, e identificar el 37.5 % y el 75 %, de los fallos detectados. Presenta muchas simulaciones con falsas alarmas, y los retrasos medios son de 108.92 y 166.8 observaciones, respectivamente. Es algo peor que el método anterior, a lo que hay que añadir un mayor tiempo de cálculo y mayor cantidad de datos enviados.

El algoritmo de factorización QR y clústers puede detectar el 100 % de fallos, e identifica el 68.75 %. No presenta falsas alarmas. Y el retraso medio es de 28.73 observaciones, respectivamente. Es un método que funciona bien, y ofrece un tiempo de detección muy bueno. Pero presenta el problema de la complejidad de cálculo, la cantidad de información que debe tratar y la necesidad de un sistema de cálculo en cada uno de los nodos o bloques.

A la vista de los resultados, se puede decir que el mejor método es el PCA distribuido propuesto, ya que, con el estadístico Q, es capaz de detectar todos los fallos, identificar casi todos, y es el más rápido, y todo ello con bajo índice de falsas alarmas y de observaciones anómalas. Además, con el otro estadístico, T^2 se comporta mejor que el resto de métodos en casi todos los aspectos (aunque es muy lento).

Capítulo 6

Resultados y conclusiones. Red de agua

Ahora se van a simular diferentes fallos para comprobar cómo se comportan los métodos puestos a prueba. Al igual que en el caso de la depuradora de aguas residuales, el número de componentes principales se fijará de tal forma que los valores propios de la matriz de covarianza asociados a esos componentes expliquen el 70 % de la varianza total. El fallo aparecerá en la hora 150 en todos los fallos. Las simulaciones duran 200 horas, lo que dará lugar a 200 observaciones de cada variable en cada sensor, para crear y probar los modelos PCA se tomarán datos a partir de la observación 100, a partir de la cual se considera que el sistema se ha estabilizado. Para que se detecte un fallo, será necesario que alguno de los dos estadísticos supere su umbral al menos 5 observaciones seguidas (este valor se ha tomado después de hacer pruebas y ver que es el valor más bajo que funciona bien en todos los métodos).

Para el método PCA global el análisis del sistema se hará en bloque. Los métodos distribuidos trabajarán con el sistema dividido en 8 bloques (figura 4.2), conteniendo cada uno de ellos 9 nodos o puntos de consumo y 9 links o tuberías de conexión (de esta forma, todos los bloques cuentan con el mismo número de variables). A la hora de agrupar elementos se han seguido criterios topológicos, agrupando componentes por proximidad.

Se va a mostrar ahora los resultados de un experimento para que se vea el resultado obtenido.

6.1. Ejemplo: Fallo en la bomba

El fallo aparecerá en el tiempo 150h (las 100 primeras observaciones no se tendrán en cuenta ya que se consideran como un tiempo de estabilización del sistema) y está localizado en la bomba, que se encuentra en la tubería 1, así que el fallo debería localizarse en esa tubería o en el nodo 2 o en la

tubería número 2, que son los que están a continuación de la bomba. El fallo consistirá en la reducción de la velocidad de la bomba hasta un valor del 90% de la velocidad normal de trabajo.

Método PCA centralizado

VARIABLES: 648. 3 componentes principales (con al menos un 70% de la varianza). Los resultados están en la gráfica 6.1 y en la tabla 6.1.

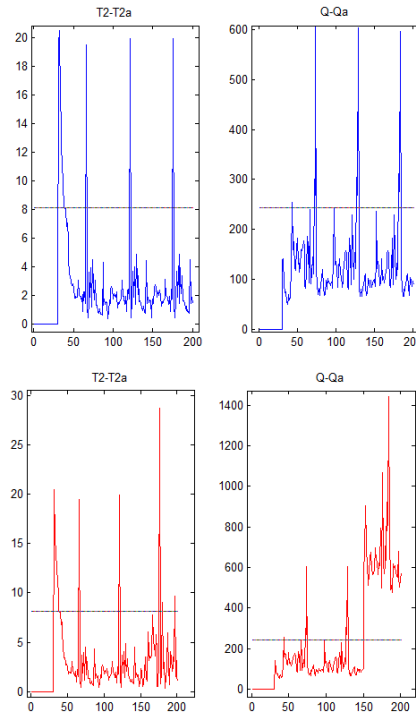


Figura 6.1: PCA centralizado. Fallo bomba 90%. Simulación sin fallo-con fallo

Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Contribución T^2	Contribución Q
-	150	-	'Node quality 307'	-	1,899346775

Cuadro 6.1: Resultados. Método PCA centralizado

El estadístico T^2 no detecta el fallo. Q sí lo hace en el instante 150 y la variable más influyente es la cantidad de contaminante (quality) en el nodo 307, que no es el responsable del fallo.

Método PCA distribuido

CAPÍTULO 6. RESULTADOS Y CONCLUSIONES. RED DE AGUA 71

Los resultados se pueden ver en las gráficas 6.2, 6.3, 6.4 y 6.5 y en la tabla 6.2

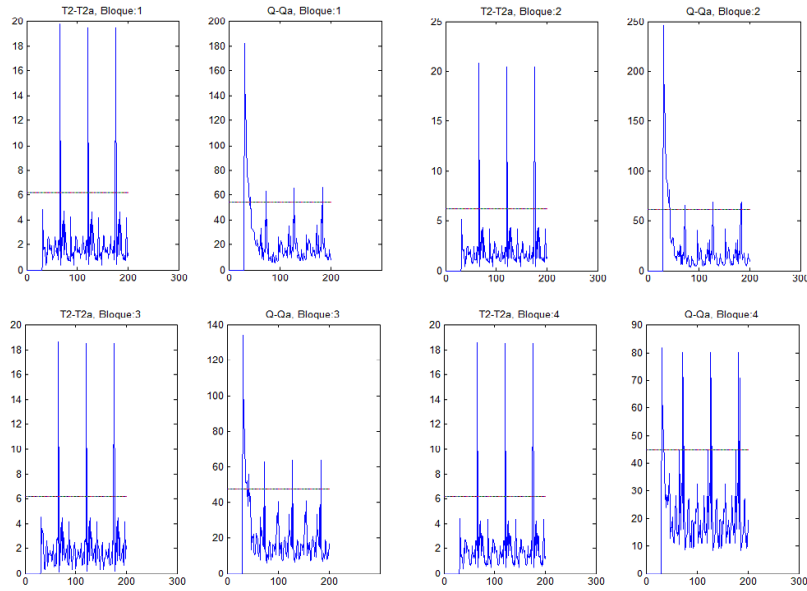


Figura 6.2: PCA distribuido. Fallo bomba 90 %. Simulación sin fallo

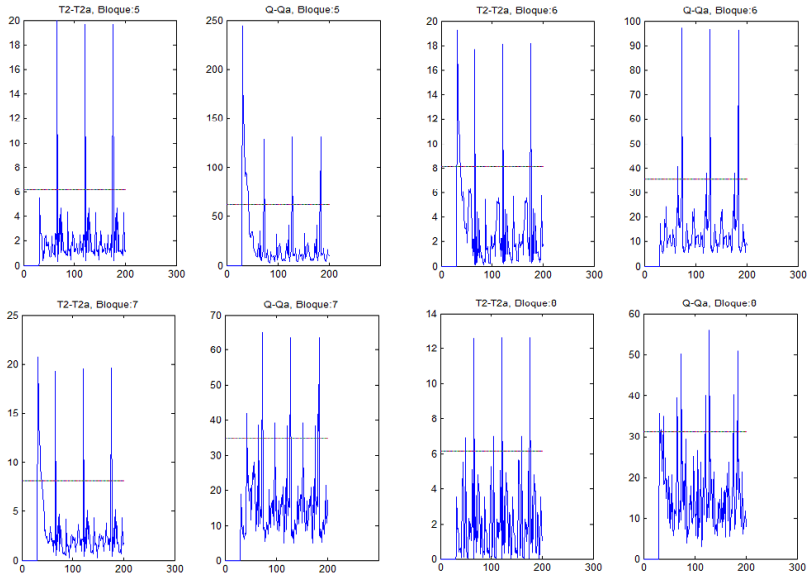


Figura 6.3: PCA distribuido. Fallo bomba 90 %. Simulación sin fallo

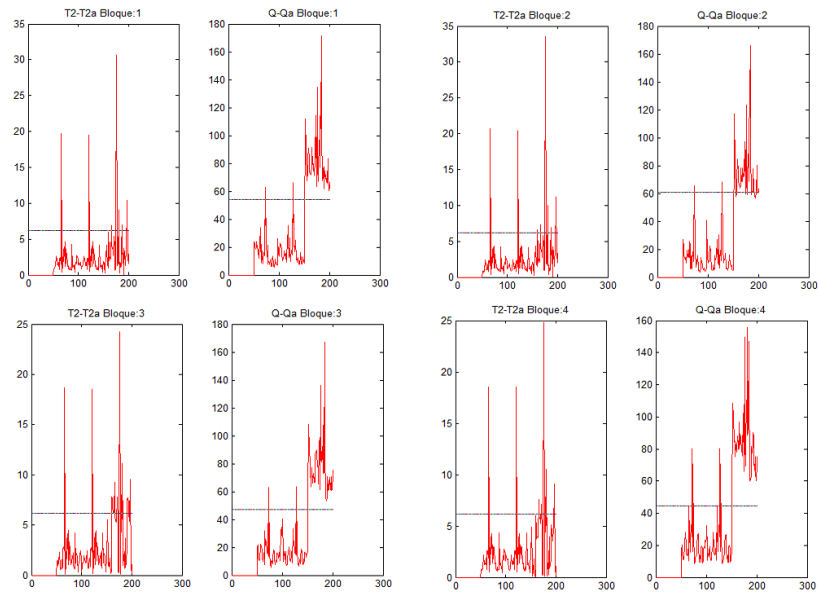


Figura 6.4: PCA distribuido. Fallo bomba 90%. Simulación con fallo

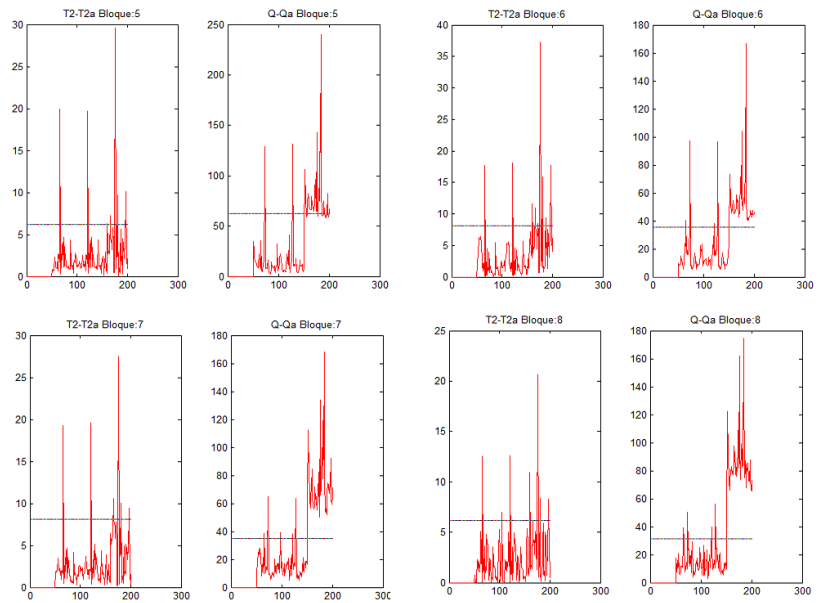


Figura 6.5: PCA distribuido. Fallo bomba 90%. Simulación con fallo

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	155	-	'Node head 101'	-	2,2499
2	-	162	-	'Link friction factor 212'	-	1,7248
3	-	155	-	'Node head 301'	-	2,2253
4	-	155	-	'Node pressure 401'	-	2,24271
5	-	162	-	'Node head 506'	-	1,6500
6	-	155	-	'Node head 601'	-	2,2428
7	-	155	-	'Node head 704'	-	2,2475
8	-	155	-	'Node pressure 2'	-	2,2510

Cuadro 6.2: Resultados. Método PCA distribuido

Se observa que el estadístico T^2 , aunque, en algunos bloques, llega a superar su umbral al final de la simulación, no lo hace claramente en ningún bloque, ya que oscila mucho. En cambio, el estadístico Q, supera claramente su límite en el bloque 8 (que es el bloque que contiene la bomba), en el resto de bloques también se supera, pero no con tanta claridad. Además, en los bloques 1,3,4,5,6,7 y 8 es donde Q antes detecta el fallo (solo 5 observación después de producirse el fallo), pero el bloque 8 es el que tiene una mayor contribución al fallo: 2,2510, el resto ofrecen valores un poco más bajos. La identificación apunta a la presión en el nodo 2 (que es el que está justo después de la bomba).

Método PCA distribuido con factorización QR

Al aplicar este método, también se obtendrán los resultados de la detección e identificación para cada uno de los bloques. En este caso, se representan los datos de simulación pero habiendo eliminado las primeras 100 observaciones (que son las que se consideraban como tiempo de estabilización). Y los resultados se presentan en las gráficas 6.6, 6.7, 6.8 y 6.9 y los tiempos de detección, variables responsables del fallo y contribuciones para los estadístico T^2 y Q se muestran en la tabla 6.3

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	-	-	-	-	-
2	-	-	-	-	-	-
3	-	-	-	-	-	-
4	-	164	-	'Node head 401'	-	0,5484
5	-	-	-	-	-	-
6	167	-	'Link velocity 634'	-	0,0071	-
7	-	-	-	-	-	-
8	-	-	-	-	-	-

Cuadro 6.3: Resultados. Método PCA con factorización QR

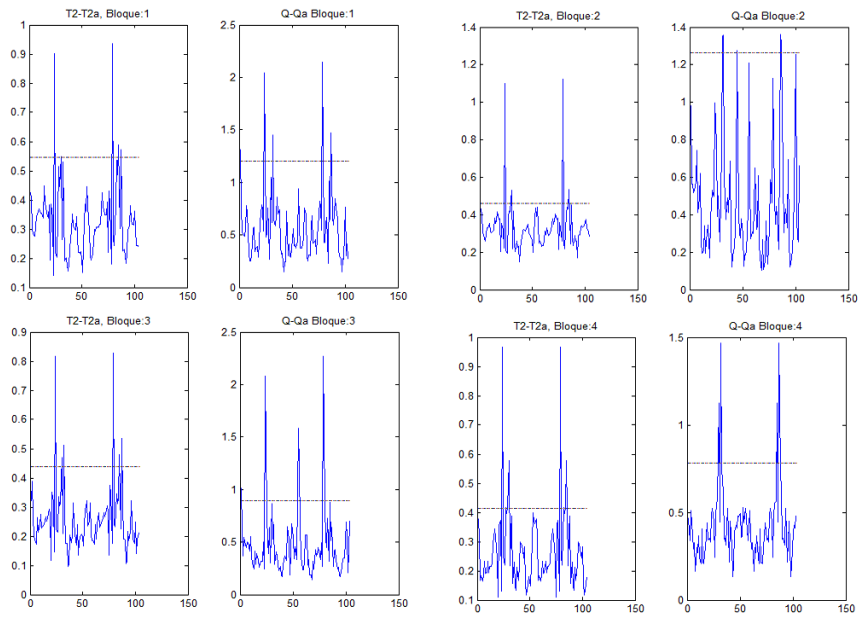


Figura 6.6: Factorización QR. Fallo bomba 90%. Simulación sin fallo.

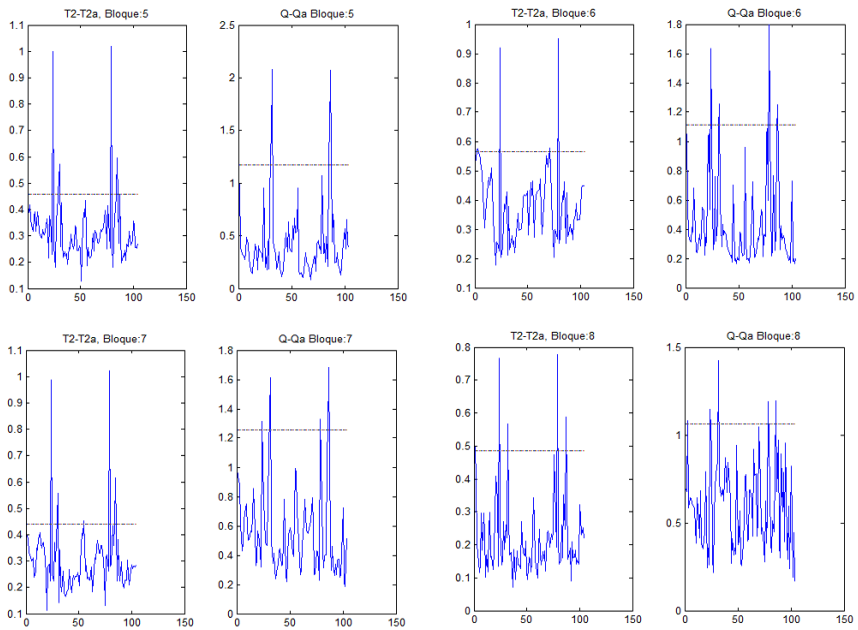


Figura 6.7: Factorización QR. Fallo bomba 90%. Simulación sin fallo.

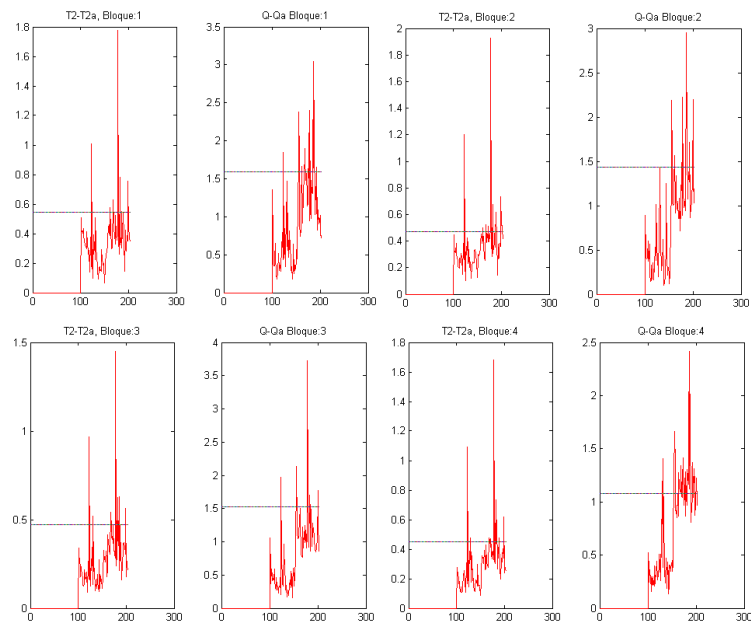


Figura 6.8: Factorización QR. Fallo bomba 90%. Simulación con fallo.

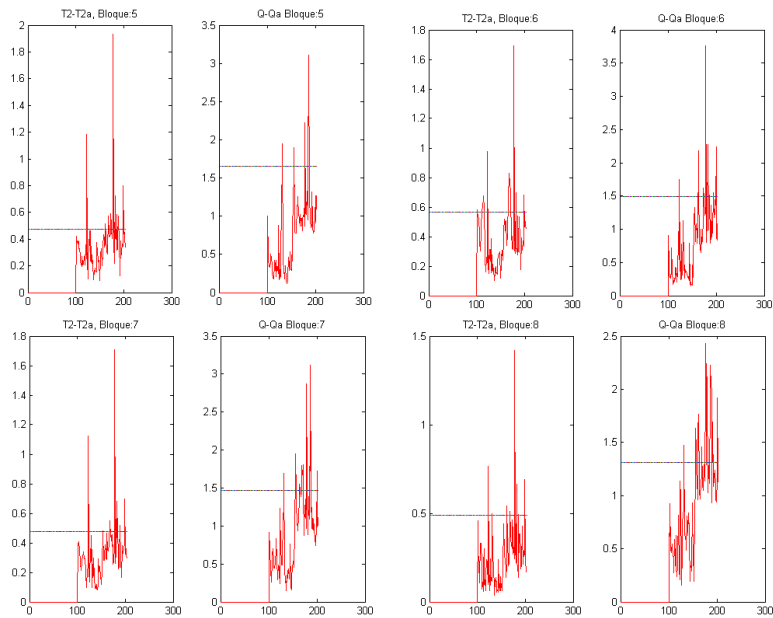


Figura 6.9: Factorización QR. Fallo bomba 90%. Simulación con fallo.

En este caso parece que el método no trabaja tan bien. Ya que detecta el fallo en bloques en los que no está éste (aunque son bloques que sufren las consecuencias del fallo, al estar todos interconectados). Al igual que antes, es el estadístico Q el que más cambio presenta al producirse los fallos (a partir del instante 50 de las gráficas, que es el 150 del total de las 200 observaciones de la simulación). Se detecta un fallo en el bloque 4, que detecta el fallo 14 observaciones después de aparecer éste. Apunta a la variable que mide la altura piezométrica del nodo 401, que obviamente no es la responsable. El estadístico T^2 detecta el fallo 17 observaciones después de producirse el fallo y tampoco identifica correctamente la variable responsable.

Método CPCA

En este caso se obtiene una gráfica global (6.10) para la instalación y la tabla 6.4

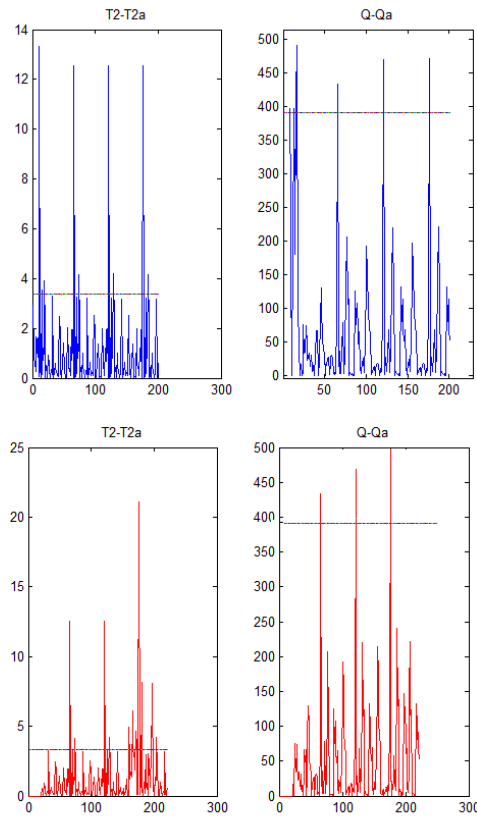


Figura 6.10: CPCA. Fallo bomba 90 %. Simulación sin fallo - Simulación con fallo

Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Contribución T^2	Contribución Q
-	-	-	-	-	-

Cuadro 6.4: Resultados. Método CPCA

Ninguno de los estadísticos es capaz de detectar el fallo. Parece que el método no es capaz de detectar y diagnosticar correctamente el fallo.

Método Merging PCA distributed

Para este método se obtienen otra vez resultados de la detección e identificación en cada bloque. La evolución de los estadísticos T^2 y Q se puede ver representados en las gráficas 6.11y 6.12 y 6.9 (sólo se representan las gráficas del sistema para la simulación con fallo.) y en la tabla 6.5

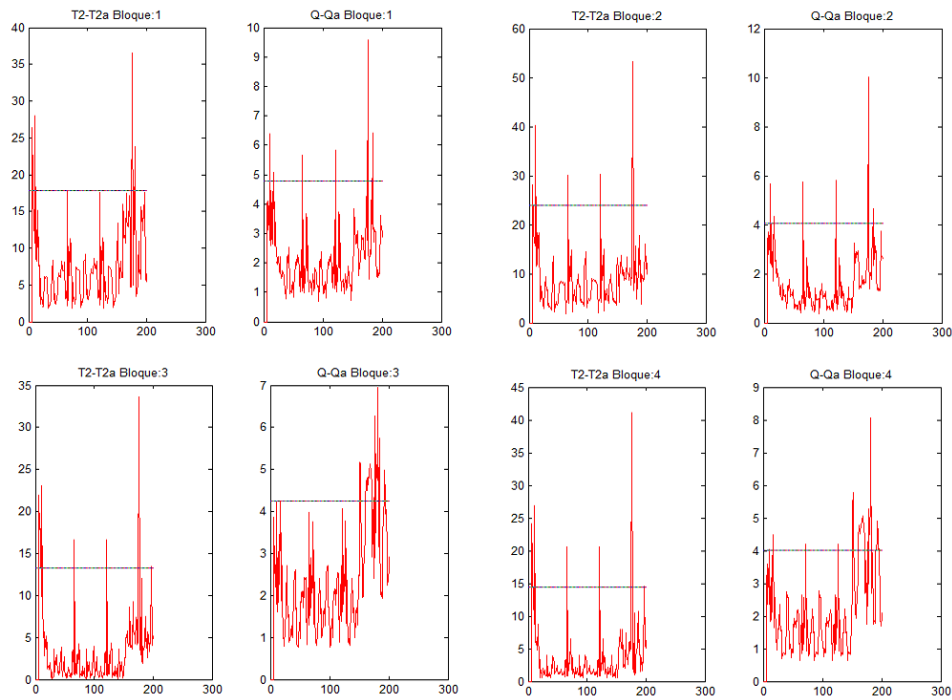


Figura 6.11: Merged PCA. Fallo bomba 90%. Simulación con fallo

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	165	-	'Node actual demand 106'	-	0,5051	-
2	-	-	-	-	-	-
3	-	161	-	'Node actual demand 306'	-	0,5483
4	-	160	-	'Node pressure 402'	-	0,6715
5	-	-	-	-	-	-
6	153	-	'Link flows 623'	-	0,8741	-
7	-	161	-	'Link unit headloss 789'	-	0,6118
8	-	-	-	-	-	-

Cuadro 6.5: Resultados. Método Merging PCA

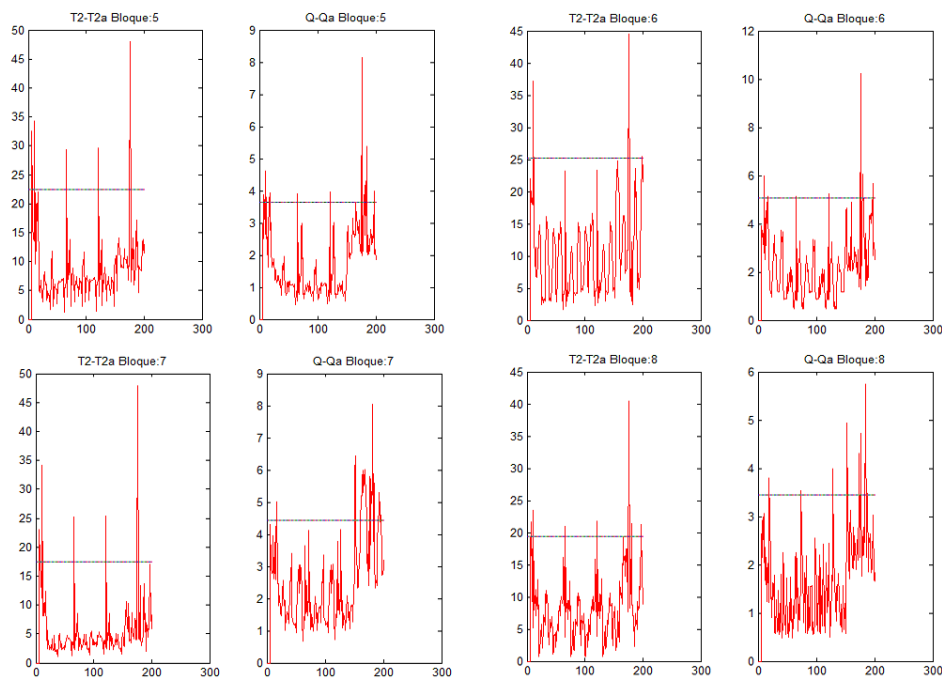


Figura 6.12: Merged PCA. Fallo bomba 90%. Simulación con fallo

Se observa como los dos estadísticos parecen sufrir alguna variación después del instante 150, principalmente Q. La primera alarma salta en el instante 153 para T^2 , pero no identifica correctamente la variable responsable del fallo. Q presenta una alarma en el instante 161 en el bloque 4, apuntando a la variable “Node 402 pressure”, que no es la responsable del fallo, así que éste método no es capaz de identificar el fallo, y además lo detecta más tarde que otros métodos.

Método PCA distribuido usando factorización QR y agrupamiento de datos en clústers

Dos instantes después de producirse el fallo, algunos bloques ya tienen clústers por encima del umbral, como se ve en la figura 6.13.

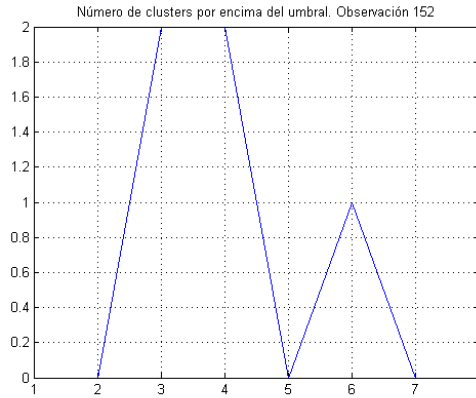


Figura 6.13: QR y clustering. Fallo bomba 90%. Clústers por encima del umbral en instante 152

Se puede analizar ahora la evolución del número de clústers por encima del umbral que aparecen para cada bloque. En la gráfica 6.14 se ve cómo, unos instantes después de producirse el fallo, empiezan a aparecer, en todos los bloques, clústers por encima del umbral.

Tiempo	Variable	Falsas alarmas
152	'Node 306 actual demand'	0

Cuadro 6.6: Resultados. Método QR con clusters

Después de hacer la identificación de variables responsables del fallo en el instante 152 (que es cuando se detecta el problema), se obtiene que la variable del bloque 3, 'Node 306 actual demand', es la que el método identifica como la más influyente. No es esta la responsable real del fallo, así que, en este caso, el método no funciona adecuadamente (probablemente debido a que la normalización aplicada a la matriz de datos X no es adecuada para realizar la identificación).

6.2. Resultados y conclusiones

Se han probado diferentes métodos para detectar fallos en una red de distribución de agua. Además, se ha analizado si era beneficioso analizar todo el sistema en un bloque o era mejor hacerlo dividiéndolo en subsistemas. Los

fallos implementados han sido de 3 tipos, con 2 magnitudes diferentes cada uno, dando lugar a 6 simulaciones diferentes para cada método.

Como resumen de las pruebas realizadas se muestran las tablas 6.7 y 6.8.

Cuadro 6.7: Fallos detectados e identificados, y tasa de falsas alarmas

Método	Fallos detectados		Fallos identificados		OTI	
	T^2	Q	T^2	Q	T^2	Q
PCA Centralizado	0	6	0	2	1.33	5.11
PCA Local	5	6	3	6	1.62	4
PCA Colectivo	1	1	0	0	2.67	0.44
Merging PCA	2	6	2	4	5	2.1
DPCA (QR)	4	4	3	3	5	1.875
DPCA & Clustering	5		4		0	

Cuadro 6.8: Tiempo de detección y TI

Método	Retraso en detección (n° observaciones)		TI (% de pruebas)	
	T^2	Q	T^2	Q
	PCA Centralizado	-	1.5	0% (0/16)
PCA Local	5.67	1.33	0% (0/16)	0% (0/16)
PCA Colectivo	34	10	0% (0/16)	0% (0/16)
Merging PCA	10.5	5.67	0% (0/16)	0% (0/16)
DPCA (QR)	6.75	4.5	0% (0/16)	0% (0/16)
DPCA & Clustering	5.4		0% (0/16)	

Resumiendo lo visto anteriormente, podemos decir que los métodos PCA centralizado (sección: 2) y PCA distribuido (sección: 2.3.2), funcionan bastante bien, ya que son capaces de detectar todos los fallos con Q, y lo hacen muy rápido. Además, con ese mismo estadístico Q, el PCA distribuido es capaz de identificar las variables responsables de los fallos en todos los casos.

El método de factorización QR (sección: 2.3.3) detecta 4 de los 6 fallos, y de estos identifica bien 3 de ellos, lo que no es un buen resultado. Además, es una técnica bastante lenta. El método CPCA (sección: 2.3.4) solamente detecta 1 fallo, pero es incapaz de identificar la variable responsable. Hay que recordar que estos métodos se han diseñado para transmitir información usando la menor cantidad de información posible, pero no son métodos específicos de detección de fallos, y es posible que al aplicarlos se pierda información en el proceso, a pesar de que, como se ha dicho, la técnica QR trabaja razonablemente bien, al menos en la detección.

En el caso del método Merging PCA distribuido (sección: 2.3.5), el estadístico Q funciona muy bien, ya que detecta todos los fallos, aunque no es de los más rápidos, e identifica 4 de los 6 fallos detectados. El estadístico T^2 detecta e identifica sólo 2 fallos.

Y el método con factorización QR y clustering (sección: 2.3.6): es el método más complejo de los tres, pero da un resultado positivo, una vez que se ha elegido un radio de clúster adecuado (en este caso, radio 0,1). Es capaz de detectar 5 de los 6 fallos, e identifica 4 de ellos. No es un método muy rápido.

Se observa que el método PCA distribuido es el más rápido de los métodos distribuidos y es mucho más sencillo de implantar, además de ser el único capaz de identificar las variables responsables del fallo. Como se dijo antes, los otros métodos están más pensados para reducir la cantidad de información transmitida por los bloques que para detección de fallos, de modo que el resultado obtenido, de pérdida de eficacia en la detección e identificación (sobre todo ésta última), es lógico.

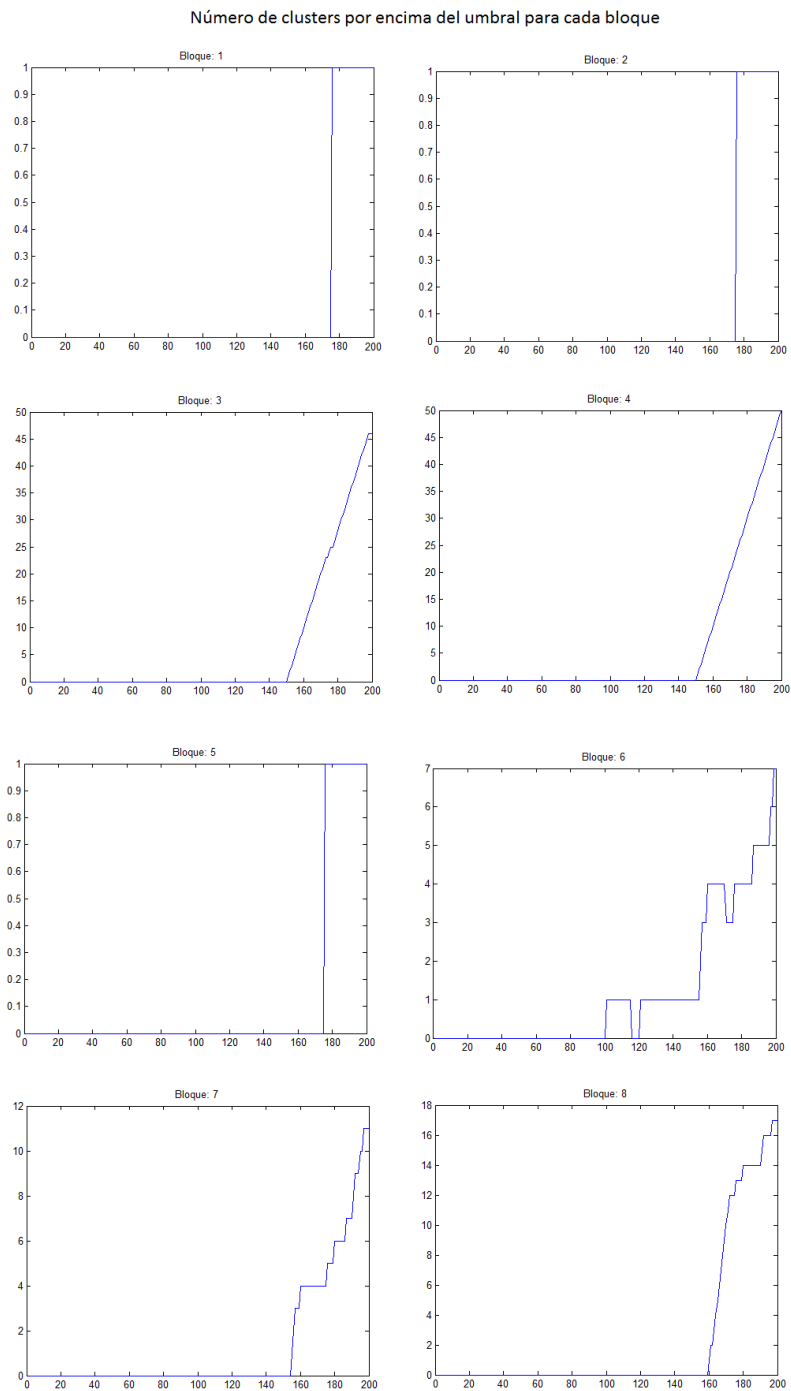


Figura 6.14: QR y clustering. Fallo bomba 90%. Evolución de cada bloque. Clústers por encima del umbral

Capítulo 7

Conclusiones finales

7.1. Conclusiones

En este trabajo se ha presentado un método de detección e identificación de fallos, basado en el análisis de componentes principales (PCA). La forma de aplicarlo ha sido a través de un planteamiento distribuido, es decir, dividiendo la planta analizada en bloques y aplicando el método a cada uno de esos bloques.

El proceso seguido consiste en dividir el sistema en bloques que agrupen cada uno una serie de variables. En cada bloque se calcula un modelo PCA usando datos de la planta trabajando en condiciones normales. Después, cada bloque procesará los valores de las nuevas observaciones, y obtendrá un resultado que dirá si en ese momento hay fallo o no, y, en el caso de que sí lo haya, se enviarán el instante de detección y el análisis de contribuciones de las variables (para determinar qué variable es la responsable del fallo) a un procesador central. Este, a su vez, si le llegan datos de fallo en algunos bloques, debe analizar esta información para determinar dónde está el fallo. Para ello, tomará sólo los datos de los bloques en los que se haya detectado antes el fallo y procesará la información de contribuciones enviada por esos bloques. Finalmente, asignará la responsabilidad del fallo a la variable con mayor valor de contribución.

Se ha comparado este método con un PCA aplicado a nivel global, es decir, un modelo que agrupe todas las variables del sistema. Además se han implementado otros métodos PCA distribuidos, para ver si la técnica propuesta aquí era capaz de dar mejores resultados. Son métodos que no están pensados para la detección de fallos, sino más bien para poder enviar información desde los bloques al nodo central, reduciendo, en lo posible, la cantidad de datos enviados, a cambio de perder una parte de la información en los cálculos a nivel local.

Todos estos métodos han sido aplicados a dos plantas: una depuradora de aguas residuales, y una red de distribución de agua. En los dos casos se trata

de modelos de simulación, de los que se extraen datos de funcionamiento, que luego analizarán cada uno de los métodos puestos a prueba.

Procesando los resultados obtenidos, se puede decir que el método de PCA distribuido presentado aquí muestra un buen rendimiento, ya que es un método capaz de detectar todos los fallos, e identificar la gran mayoría de ellos. Además, es el más rápido de los métodos ensayados, y, entre los distribuidos, es el más sencillo de implementar, aparte de tener que enviar menores cantidades de datos desde los bloques al nodo central.

Otra ventaja que presenta, es que en cada bloque los datos recogidos se analizan directamente, sin ninguna transformación, mientras que en los otros métodos distribuidos, requieren procesar los datos, obtener unos cálculos previos que se envían al nodo central, donde son otra vez procesados. Todo estos pasos intermedios, que implican tanta manipulación de la información acaban por hacer que se deteriore la información transmitida.

7.2. Líneas futuras de trabajo

Se proponen varios caminos para continuar con la propuesta de este trabajo. Por un lado, se puede cambiar la forma en la que se realiza la división de la planta. Aquí se ha hecho simplemente creando un bloque para cada uno de los sensores existentes o agrupando elementos basándose en su proximidad, pero se podría comprobar qué tal trabaja el método al agrupar variables según la correlación que haya entre ellas, también se podría agrupar variables pertenecientes a bloques funcionales del proceso desarrollado en la planta, etc.

Otra línea de trabajo que se puede explorar consiste en usar modelos de plantas más realistas, en los que no haya tantos sensores disponibles (sobre todo en la red de agua, ya que en instalaciones reales de este tipo, no es común encontrarse un sensor en cada tubería y punto de consumo de la red). Esto haría necesario la estimación de algunos valores, complicando la tarea de detección e identificación de fallos.

En este trabajo se ha utilizado el método de los diagramas de contribuciones para identificar a la variable responsable del fallo, pero es un método que no siempre funciona adecuadamente. Hay otras técnicas que se pueden utilizar para realizar esta función, como pueden ser el método de reconstrucción ([14]), diagramas de contribución modificados ([15]), etc.

Otra posibilidad de cara a la identificación consiste en que cada bloque de la planta aplique varios métodos de identificación y envíe los resultados de cada uno de ellos al nodo central, que los procesará para obtener la variable o variables responsables. La forma de procesar esta información recibida requerirá aplicar técnicas de fusión de información como puedan ser métodos basados en la entropía ([10]), método OWA (ordered weighted averaging) ([16]), etc.

7.3. Artículos publicados basados en este trabajo

- A. Sánchez-Fernández, M.J. Fuente and G.I. Sainz-Palmero. *Fault detection with Distributed PCA methods in Water Distribution Networks*. 23rd Mediterranean Conference on Control and Automation, 2015
- A. Sánchez-Fernández, M.J. Fuente and G.I. Sainz-Palmero. *Fault Detection In Wastewater Treatment Plants Using Distributed PCA methods*. 20th IEEE International Conference on Emerging Technologies and Factory Automation, 2015
- A. Sánchez-Fernández, M.J. Fuente y G.I. Sainz-Palmero. *Detección De fallos En Una Planta De Tratamiento De Aguas Residuales Mediante PCA Distribuido*. XXXVI Jornadas de Automática, 2015

Bibliografía

- [1] L. Puigjaner, P. Ollero, C. de Prada, and L. Jiménez., *Estrategias de modelado, simulación y optimización de procesos químicos*. Editorial Síntesis, 2006.
- [2] J. E. Jackson, *A user's guide to principal components*. Wiley, 1991.
- [3] T. Kourti and J. F. MacGregor, "Multivariate spc methods for process and product monitoring," *Journal of Quality Technology*, 1996.
- [4] D. G. Álvarez, *Monitoring, fault detection and estimation in processes using multivariate statistical techniques*. PhD thesis, Departamento de Ingeniería de Sistemas y automática, Universidad de Valladolid, 2013.
- [5] Z. Bai, R. Chan, and F. Luk, "Principal Component Analysis for distributed data sets with updating," in *6th International Workshop on advanced parallel processing technologies*, 2005.
- [6] H. Kargupta, W. Huang, K. Sivakumar, and E. Johnson, "Distributed clustering using collective principal component analysis," *Knowledge and Information Systems*, 1999.
- [7] Y. Qu, G. Ostrouchov, N. Samatova, and A. Geist, "Principal component analysis for dimension reduction in massive distributed data sets," *Proceedings of IEEE International Conference on Data Mining*, 2002.
- [8] M. Livani and M. Abadi, "Distributed PCA-based anomaly detection in wireless sensor networks," in *2010 IEEE International Conference for Internet Technology and Secure Transactions*, 2010.
- [9] International Water Association, "IWA Task Group on Benchmarking of Control Strategies for WWTPs.." <http://www.benchmarkwwtp.org/>.
- [10] M. Grbovic, W. Li, P. Xu, A. Usadi, L. Somg, and S. Vucetic, "Decentralized fault detection and diagnosis via sparse PCA based decomposition and maximum entropy decision fusion," *Journal of Process Control*, vol. 22, pp. 738–750, 2012.
- [11] L. A. Rossman, "Epanet."

- [12] Lewis A. Rossman (Traducción: Grupo M. de Modelación de Fluidos, U. Politécnica de Valencia)., *EPANET 2, Manual de usuario*. Universidad Politécnica de Valencia.
- [13] “Baseform.org, platform of applications for networked infrastructures. <http://www.baseform.org/np4/home/>.”
- [14] C. F. Alcalá and S. J. Qin, “Reconstruction-based contribution for process monitoring,” *Automatica*, 2009.
- [15] J. Liu and D.-S. Chen, “Fault isolation using modified contribution plots,” *Computers and Chemical Engineering*, 2014.
- [16] R.R. Yager, “On ordered weighted averaging aggregation operators in multi-criteria decision making,” *IEEE Transactions on Systems, Man and Cybernetics*, 1988.