



Universidad de Valladolid



PROGRAMA DE DOCTORADO EN INGENIERIA INDUSTRIAL

TESIS DOCTORAL:

**MÉTODOS DE DETECCIÓN Y DIAGNÓSTICO DE
FALLOS MEDIANTE APROXIMACIONES
DISTRIBUIDAS: MODELOS, MÉTODOS Y
COMPUTACIÓN**

Presentada por Álgvar Sánchez Fernández para optar al
grado de
Doctor/a por la Universidad de Valladolid

Dirigida por:
Dra. Dña. Maria Jesús de la Fuente Aparicio
Dr. D. Gregorio Ismael Sainz Palmero

Agradecimientos

Quiero expresar mi agradecimiento a todas las personas e instituciones que han hecho posible la realización de esta Tesis Doctoral:

A mis directores de tesis, María Jesús de la Fuente Aparicio y Gregorio Ismael Sainz Palmero, por toda la ayuda que me han prestado, por su dedicación, por sus consejos, y por su esfuerzo para conseguir que esta tesis llegase a término.

Al personal del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, en especial a José Manuel Benítez Sánchez y a Francisco Javier Baldán Lozano, por su ayuda y su disposición.

Al personal del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Valladolid, por el apoyo prestado, y, en especial, a Jesús, Lorenzo, Lidia, Paula, Jaime y Fabio, por los buenos momentos pasados.

Al Ministerio de Economía y Competitividad de España por su apoyo a través de los proyectos: DPI2012-39381-C02-02 y DPI2015-67341-C2-2-R.

Finalmente, a Celia y a Rebeca por su ánimo, apoyo y sacrificio, y por el tiempo que no hemos podido compartir durante el desarrollo de esta tesis.

MUCHAS GRACIAS

Contenidos

I	INTRODUCCIÓN	1
1.	INTRODUCCIÓN Y OBJETIVOS	5
1.1.	Introducción	5
1.1.1.	Detección y diagnóstico de fallos	6
1.1.2.	Métodos de FDI basado en datos	7
1.1.3.	Métodos distribuidos de FDI basados en datos	8
1.2.	Motivación	10
1.3.	Objetivos	11
1.4.	Organización de la tesis	13
2.	ESTADO DEL ARTE	15
2.1.	Introducción	15
2.2.	Control estadístico de procesos	15
2.2.1.	Variabilidad en los procesos de producción	16
2.2.2.	Control estadístico univariable de procesos	16
2.3.	Métodos de detección y diagnóstico de fallos	18
2.3.1.	Métodos basados en datos	19
2.3.2.	Métodos analíticos	20
2.3.3.	Métodos basados en conocimiento	21
2.3.4.	Características de un sistema de detección y aislamiento de fallos	21
2.3.4.1.	Evaluación de los métodos de detección de fallos	22
2.4.	Métodos de detección e identificación de fallos basados en datos	23
2.5.	Análisis de Componentes Principales (PCA)	24
2.5.1.	Elección del número de Componentes Principales	26
2.5.2.	Detección de fallos: Estadísticos T^2 y Q	27
2.5.3.	Identificación de fallos	29
2.6.	Análisis de Componentes Principales Dinámico (DPCA)	31
2.7.	Análisis de Variables Canónicas (CVA)	31
2.7.1.	Modelo de espacio de estados	32

2.7.2.	Detección de fallos: Estadísticos de control	33
2.7.3.	Identificación de fallos con CVA	34
2.8.	Métodos distribuidos de FDI basados en datos	35
2.8.1.	Métodos de FDI con distribución de los modelos	36
2.8.2.	Métodos de FDI con distribución de los métodos	39
2.8.3.	Métodos de FDI con distribución de la computación	40
2.9.	Toma de decisiones multi-criterio	42
2.9.1.	Operadores OWA	43
2.10.	Resumen	44
II	FDI MEDIANTE APROXIMACIONES DISTRIBUIDAS	47
3.	MÉTODOS PCA DISTRIBUIDOS. ESTUDIO PREVIO	51
3.1.	Introducción	51
3.2.	Métodos distribuidos de cálculo de PCA	51
3.2.1.	PCA distribuido. Cálculo de PCA locales	52
3.2.2.	Método PCA distribuido con factorización QR	52
3.2.3.	Método CPCA	54
3.2.4.	Método PCA distribuido fusionado	55
3.2.5.	PCA distribuido usando factorización QR y agrupamiento de datos en clústers	56
3.3.	Comparativa entre método centralizado y métodos distribuidos	57
3.3.1.	Caso de estudio I: Red de distribución de agua potable	58
3.3.1.1.	Metodología experimental	58
3.3.1.2.	Análisis de resultados	59
3.3.2.	Caso de estudio II: Depuradora de aguas residuales	70
3.3.2.1.	Metodología experimental	70
3.3.2.2.	Análisis de resultados	71
3.4.	Resumen	81
4.	MÉTODOS FDI DISTRIBUIDOS MULTIBLOQUE BASADOS EN DPCA Y CVA. DISTRIBUCION DE MODELOS	83
4.1.	Introducción	83
4.2.	Fundamentos teóricos	84
4.2.1.	Métodos de regresión lineal	84
4.2.1.1.	Regresión LASSO	84
4.2.1.2.	Sparse PLS	85
4.2.1.3.	Sparse PCA	86
4.2.2.	Métodos de regresión no-lineal	86
4.2.2.1.	Redes Neuronales	86
4.2.2.2.	Bosques Aleatorios	87
4.2.2.3.	Regresión SVM	87

4.2.3.	Métodos basados en información lineal y no-lineal . . .	88
4.2.3.1.	Correlación	88
4.2.3.2.	DCCA	88
4.2.3.3.	Información Mutua	89
4.2.4.	Métodos basados en agrupamiento (<i>Clustering</i>)	89
4.2.5.	Métodos de toma de decisiones	90
4.3.	Método FDI propuesto: Método distribuido con PCA dinámico multi bloque: MB-DDPCA	91
4.3.1.	Métodos de división de la planta basados en datos . . .	92
4.3.2.	Métodos locales de detección de fallos	94
4.3.3.	Fusión de los resultados locales	94
4.4.	Caso de estudio: Tennessee Eastman Process	94
4.4.1.	MB-DDPCA: DPCA distribuido multi-bloque	95
4.4.1.1.	Metodología experimental	95
4.4.1.2.	Análisis de resultados	96
4.4.2.	MB-DDPCA, sensibilidad al grado de distribución . . .	104
4.4.2.1.	Metodología experimental	104
4.4.2.2.	Análisis de resultados	106
4.4.3.	MB-DCVA: CVA distribuido multi-bloque	112
4.4.3.1.	Metodología experimental	112
4.4.3.2.	Análisis de resultados	113
4.4.4.	MB-DCVA, sensibilidad al grado de distribución . . .	121
4.4.4.1.	Metodología experimental	121
4.4.4.2.	Análisis de resultados	122
4.5.	Resumen	127
5.	IDENTIFICACIÓN DE FALLOS MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN. DISTRIBUCIÓN DE MÉTODOS	129
5.1.	Introducción	129
5.2.	Fundamentos teóricos	131
5.2.1.	Índices basados en reconstrucción	131
5.2.2.	Diagramas de contribución modificados	132
5.3.	Método propuesto: Identificación de fallos distribuida basada en operadores lingüísticos OWA-RIM: ORAFI	133
5.3.1.	Descripción del Método ORAFI	134
5.4.	Caso de estudio I: Tennessee Eastman Process	137
5.4.1.	Metodología experimental	137
5.4.2.	Análisis de resultados	139
5.5.	Caso de estudio II: Depuradora de Aguas Residuales	151
5.5.1.	Metodología experimental	151
5.5.2.	Análisis de resultados	154
5.6.	Resumen	164

6. FDI BASADO EN EL ANÁLISIS DE SERIES TEMPORALES Y CONTROL ESTADÍSTICO MULTIVARIANTE. DISTRIBUCIÓN DEL CÁLCULO	165
6.1. Introducción	165
6.2. Fundamentos teóricos	166
6.2.1. Selección dinámica de características	166
6.2.2. Modelado de series temporales	168
6.2.3. Detección de fallos basada en las gráficas de control de SPC	172
6.3. Método FDI propuesto: Detección de fallos basada en series temporales y control estadístico de procesos: TSR-FDI	172
6.4. Caso de estudio I: Tennessee Eastman Process	177
6.4.1. Metodología experimental	177
6.4.2. Análisis de resultados	180
6.5. Caso de estudio II: Depuradora de Aguas Residuales	186
6.5.1. Metodología experimental	186
6.5.2. Análisis de resultados	188
6.6. Resumen	190

III CONCLUSIONES, APÉNDICES Y BIBLIOGRAFÍA
193

7. CONCLUSIONES FINALES	195
7.1. Contribuciones y publicaciones de la tesis	195
7.2. Conclusiones	198
7.3. Líneas futuras de trabajo	199

Apéndice A. MODELOS DE SIMULACIÓN	201
A.1. Red de distribución de agua potable	201
A.1.1. Fallos	203
A.2. Planta depuradora de aguas residuales	206
A.2.1. Fallos	207
A.3. Tennessee Eastman Process	208

BIBLIOGRAFÍA	213
---------------------	------------

Índice de figuras

2.1. Gráfica de control <i>Shewhart</i>	18
2.2. Componentes Principales	26
2.3. Componentes principales	28
3.1. Red de agua por bloques	60
3.2. PCA centralizado. Fallo bomba 10 %. Simulación con fallo . .	62
3.3. PCA distribuido. Fallo bomba 10 %. Bloques 1-4	62
3.4. PCA distribuido. Fallo bomba 10 %. Bloques 5-8	63
3.5. PCA con factorización QR. Fallo bomba 10%. Bloques 1-4. .	64
3.6. PCA con factorización QR. Fallo bomba 10%. Bloques 5-8. .	65
3.7. CPCA. Fallo bomba 10%.	66
3.8. <i>Merging</i> PCA. Fallo bomba 10%. Bloques 1-4	66
3.9. <i>Merging</i> PCA. Fallo bomba 10%. Bloques 5-8	67
3.10. QR con clustering. Fallo bomba 10%. Bloques 1-2	68
3.11. QR con clustering. Fallo bomba 10%. Bloques 3-4	68
3.12. QR con clustering. Fallo bomba 10%. Bloques 5-6	69
3.13. QR con clustering. Fallo bomba 10%. Bloques 7-8	69
3.14. QR con clustering. Fallo bomba 10%. Clústers anómalos en cada bloque.	70
3.15. PCA centralizado. Fallo 50%.	73
3.16. PCA distribuido. Fallo 50%.	74
3.17. Factorización QR. Fallo 50%.	76
3.18. CPCA. Fallo 50%.	77
3.19. <i>Merging</i> PCA. Bloques 7, 8, 9, 10, 11 y 12.	78
3.20. <i>Merging</i> PCA. Bloques 13, 15, 16, 18, 19 y 20.	79
3.21. Fallo 50 % en el sensor de oxígeno. Número de clústers por encima del umbral en el instante 28843.	80
3.22. Fallo 50 % en el sensor de oxígeno. Evolución del número de clústers anómalos del reactor 4.	80
5.1. Fallo 7. Evolución de las variables candidatas.	150
6.1. Esquema de la matriz aumentada.	167
6.2. Metodología <i>Box-Jenkins</i>	171

ÍNDICE DE FIGURAS

6.3. Diagrama de flujo del método propuesto	173
A.1. Red de distribución de agua potable	202
A.2. Patrón de demanda de agua	204
A.3. Introducción de contaminante a la red	205
A.4. Tubería de fuga de agua	205
A.5. Planta BSM2	206
A.6. Esquema de la planta Tennessee Eastman Process (TEP) . .	209

Índice de tablas

3.1. Fallos detectados e identificados, y tasa de falsas alarmas . . .	59
3.2. Retraso en la detección (en instantes) y TI	59
3.3. Resultados. Método PCA centralizado	61
3.4. Resultados. Método PCA distribuido	63
3.5. Resultados.PCA con factorización QR	64
3.6. Resultados. Método <i>Merging</i> PCA	67
3.7. Resultados. Método QR con clústers	68
3.8. Fallos detectados e identificados, y OTI (tasa de falsas alarmas)	71
3.9. Tiempo de detección y TI	71
3.10. Resultados. Método PCA centralizado	73
3.11. Resultados. Método PCA distribuido	75
3.12. Resultados. Método PCA con factorización QR	76
3.13. Resultados. Método CPCA	76
3.14. Resultados. Método <i>Merging</i> PCA	77
3.15. Resultados. Método QR con clustering	81
4.1. Parámetros de los métodos distribuidos basados en DPCA. . .	96
4.2. Retardo en la detección (en instantes). T^2	97
4.3. Retardo en la detección (en instantes). Q	98
4.4. Tasa de no detección de fallos (MDR), en porcentaje. T^2 . . .	100
4.5. Tasa de no detección de fallos (MDR), en porcentaje. Q . . .	101
4.6. Bloques, FAR y falsas alarmas.	103
4.7. Parámetros de los métodos distribuidos basados en DPCA. . .	104
4.8. Retardo en la detección (en instantes). T^2	107
4.9. Retardo en la detección (en instantes). Q	108
4.10. Tasa de no detección de fallos (MDR), en porcentaje. T^2 . . .	109
4.11. Tasa de no detección de fallos (MDR), en porcentaje. Q . . .	110
4.12. FAR (en porcentaje), fallos detectados, falsas alarmas y número de bloques.	111
4.13. Parámetros de los métodos distribuidos basados en CVA. . .	113
4.14. Retardo en la detección (en instantes). T_s^2	114
4.15. Retardo en la detección (en instantes). T_r^2	115
4.16. Retardo en la detección (en instantes). Q	116

ÍNDICE DE TABLAS

4.17. Tasa de no detección de fallos (MDR), en porcentaje. T_s^2 . . . 117

4.18. Tasa de no detección de fallos (MDR), en porcentaje. T_r^2 . . . 118

4.19. Tasa de no detección de fallos (MDR), en porcentaje. Q . . . 119

4.20. FAR (en porcentaje), fallos detectados, falsas alarmas y bloques. 120

4.21. Parámetros de los métodos distribuidos basados en CVA. . . 121

4.22. Retardo en la detección (en instantes). T_s^2 122

4.23. Retardo en la detección (en instantes). T_r^2 123

4.24. Retardo en la detección (en instantes). Q 124

4.25. Tasa de no detección de fallos (MDR), en porcentaje. T_s^2 . . . 125

4.26. Tasa de no detección de fallos (MDR), en porcentaje. T_r^2 . . . 125

4.27. Tasa de no detección de fallos (MDR), en porcentaje. Q . . . 126

4.28. FAR (en porcentaje), fallos detectados, falsas alarmas y bloques. 126

5.1. Ajuste de la longitud de la ventana *Temprana* 139

5.2. Ajuste del tamaño de la ventana *Estacionaria*. Varianza=0.001139

5.3. Ajuste del tamaño de la ventana *Estacionaria*. Varianza=0.002140

5.4. Ajuste del tamaño de la ventana *Estacionaria*. Varianza=0.003140

5.5. Comparativa para la identificación *Temprana*. Alta aversión al riesgo (*orness*=0.2) 142

5.6. Comparativa para la identificación *Temprana*. Moderada aversión al riesgo (*orness*=0.5) 143

5.7. Comparativa para la identificación *Temprana*. Baja aversión al riesgo 144

5.8. Retardo en la diagnosis *Estacionaria*. 145

5.9. Diagnosis Estacionaria. Alta aversión al riesgo (*orness*=0.2) . 147

5.10. Diagnosis Estacionaria. Moderada aversión al riesgo (*orness*=0.5) 148

5.11. Diagnosis Estacionaria. Baja aversión al riesgo (*orness*=0.9) . 149

5.12. Ajuste de la longitud de la ventana *Temprana* 152

5.13. Ajuste de la longitud de ventana *Estacionaria*, Var.=0.0005 . 154

5.14. Ajuste de la longitud de la ventana *Estacionaria*, Var.=0.001 154

5.15. Ajuste de la longitud de ventana *Estacionaria*, Var.=0.002 . . 154

5.16. Comparativa para la identificación *Temprana*. Alta aversión al riesgo (*orness* = 0.2) 156

5.17. Comparativa para la identificación *Temprana*. Moderada aversión al riesgo (*orness* = 0.5) 157

5.18. Comparativa para la identificación *Temprana*. Baja aversión al riesgo (*orness* = 0.9) 158

5.19. Tasas de éxito de cada diagnosis 159

5.20. Tasas de éxito en la diagnosis *Temprana* 159

5.21. Retardo en la diagnosis *Estacionaria* (en instantes). BSM2 . . 160

5.22. Diagnosis *Estacionaria*. Alta aversión al riesgo (*orness* = 0.2) 161

5.23. Diagnosis *Estacionaria*. Moderada aversión al riesgo (*orness* = 0.5) 161

5.24. Diagnósis <i>Estacionaria</i> . Baja aversión al riesgo (<i>orness</i> = 0.9)	162
6.1. Modelo de serie temporal seleccionado para cada variable . . .	179
6.2. Fallos detectados por cada método. TEP	181
6.3. Tasa de falsas alarmas (FAR), en porcentaje. TEP	181
6.4. Tasa de no detección de fallos (MDR), en porcentaje. TEP . .	183
6.5. Retardo en la detección (en instantes). TEP	185
6.6. Modelos seleccionados para algunas variables. BSM2	187
6.7. Tasa de falsas alarmas (FAR), en porcentaje. BSM2	188
6.8. Tasa de no detección de fallos (MDR), en porcentaje. BSM2	189
6.9. Retardo en la detección (en instantes). BSM2	190
A.1. Variables en la planta BSM2	207
A.2. Fallos disponibles en la planta BSM2	208
A.3. Variables en la planta TEP	210
A.4. Fallos disponibles en la planta TEP	211

Parte I:

INTRODUCCIÓN

Resumen

En esta primera parte de la tesis se expone la motivación para realizar esta investigación en el área de la detección y diagnóstico de fallos. Se comienza exponiendo los conceptos básicos de la monitorización de procesos y de la detección y diagnóstico de fallos, así como los retos que se plantean en dicho campos. Partiendo de dichos retos, se detallan la motivación y los objetivos planteados para el desarrollo de esta tesis: la detección y el diagnóstico de fallos distribuido en plantas de gran tamaño.

Por otro lado, se hace una revisión de los conceptos y métodos más destacados utilizados para la monitorización de plantas industriales, especialmente, las técnicas basadas en datos, que son la base para el trabajo desarrollado en esta tesis. Dentro de estos métodos se presentan alternativas para hacer tanto la detección como la identificación de los diferentes fallos. Realizando dichas tareas de forma centralizada así como de forma descentralizada. Además, se incluyen conceptos de métodos de toma de decisiones.

Palabras claves

Monitorización de procesos industriales. Detección y diagnóstico de fallos. Métodos basados en datos. Métodos descentralizados. Toma de decisiones

Capítulo 1

INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción

Uno de los objetivos que debe perseguir cualquier industria es conseguir una producción óptima en términos de cantidad, calidad, plazos, seguridad, etc. La aplicación de la ingeniería de control en las últimas décadas ha permitido acercarse cada vez más a ese óptimo, reduciendo costes, desperdicio de material, accidentes y paradas no programadas, y aumentando la versatilidad, rentabilidad y fiabilidad de las instalaciones [Ge and Song, 2013, Ge and Chen, 2016, Jiang and Braatz, 2017, Tian et al., 2019].

A pesar de los avances conseguidos, sigue siendo necesario realizar un chequeo o monitorización de la instalación para comprobar que no existen desviaciones respecto al comportamiento esperado. Dichas desviaciones pueden llevar a la instalación a trabajar en condiciones no adecuadas, generando pérdidas de calidad, pérdidas de producción, averías, o, incluso, accidentes. Todo ello con graves repercusiones para la empresa y para el personal de la misma.

Por tanto, es imprescindible contar con un sistema de monitorización que analice el comportamiento de la planta y que sea capaz de detectar las citadas desviaciones de una forma eficaz. Además, para evitar daños a las instalaciones, debe ser un método que trabaje en línea con la planta, para poder detectar los fallos rápidamente y facilitar la aplicación de medidas correctivas cuanto antes.

Otros de los requerimientos a los que se enfrenta la industria es conseguir un nivel de calidad adecuado a los requerimientos de los clientes. Las empresas, para ser eficientes, deben comprobar si el resultado de sus procesos de producción se ajusta a dichos requerimientos, ya que de no hacerlo, puede implicar la pérdida de clientes (si la calidad es más baja de lo exigido) o desaprovechamiento de recursos materiales, económicos, etc. (en el caso de obtener una calidad mayor de la demandada). En cualquier caso, implicaría pérdidas económicas que hay que evitar. La monitorización de procesos, y

en concreto, la detección y el diagnóstico de fallos pueden ser utilizados para evaluar la producción de la empresa y detectar la presencia de factores que modifiquen la calidad del producto, permitiendo tomar medidas correctivas para retornar a los estándares de calidad demandados.

1.1.1. Detección y diagnóstico de fallos

Un esquema de detección y diagnóstico de fallos (FDD), o, a veces, llamado también esquema FDI (detección e identificación de fallos) permite incrementar la fiabilidad de la planta, al mismo tiempo que mejora su eficiencia, mantenibilidad y disponibilidad. Sin embargo para diseñar un esquema de FDI se necesita conocer el significado de algunos conceptos básicos.

En este dominio se puede hacer la distinción entre fallo, avería y mal funcionamiento [Frank et al., 2000, Isermann, 2006]:

Un *fallo* se puede definir como *una desviación no permitida de al menos una característica del sistema respecto al estado usual, aceptable y estándar del mismo*. Siendo una desviación no permitida la distancia entre el valor anómalo y el límite del área que engloba los valores permitidos. Debe quedar claro que el fallo es un estado del sistema que causa una reducción de la capacidad de una instalación para realizar las tareas encomendadas. Como características principales se pueden citar las siguientes: un fallo puede aparecer tanto si el sistema está en marcha como si no; puede llevar a un fallo catastrófico o no; puede ser difícil de detectar si es de pequeña intensidad; y puede aparecer tanto de forma abrupta como de forma progresiva.

Además, existen gran variedad de fallos: de diseño, de mantenimiento, del operador, etc.

Una *avería* es *una interrupción permanente de la capacidad del sistema para realizar una determinada función según las condiciones de funcionamiento especificadas*. La presencia de una *avería* implica la incapacidad de una instalación para realizar alguna de las funciones que se le han encomendado. Por otro lado, suele aparecer a consecuencia de uno o más fallos; puede ser múltiple o no; puede ser aleatorio (o impredecible), determinista (predecible bajo ciertas condiciones) o sistemático (aparece cuando se dan ciertas condiciones conocidas); y suele aparecer al comienzo de la operación, o cuando se exige al sistema mas de lo usual.

Un *mal funcionamiento* es *una irregularidad intermitente en el cumplimiento de las funciones exigidas al sistema*. Implica una interrupción temporal de las funciones del sistema. Suele ser consecuencia de uno o más fallos y sucede a menudo cuando arranca la instalación o cuando se la somete a un sobreesfuerzo.

Al ser las *averías* y los *mal funcionamientos* consecuencia de los *fallos*, haciendo una monitorización de estos últimos es posible controlar la aparición de los primeros. Por tanto, en esta tesis el objetivo ha sido trabajar sobre los *fallos*.

El proceso de detección y aislamiento (FDI) de fallos se puede dividir en cuatro etapas [Puigjaner et al., 2006]:

1. **Detección de la existencia del fallo.**
2. **Identificación del fallo:** Se busca qué variables medidas en la planta son responsables del fallo.
3. **Diagnóstico del fallo:** Se busca el tipo y la causa del fallo.
4. **Recuperación del sistema:** Retorno al funcionamiento normal, si es posible.

Aunque existen diversas clasificaciones de métodos de detección y diagnóstico de fallos [Chiang et al., 2001, Venkatasubramanian et al., 2003a, Venkatasubramanian et al., 2003b, Venkatasubramanian et al., 2003c, Isermann, 2006], una clasificación muy usada es la siguiente [Miljković, 2011]:

- **Métodos basados en datos:** se basan en analizar datos medidos en la planta sin necesidad de tener un gran conocimiento de la misma.
- **Métodos basados en modelos del proceso:** requieren un modelo preciso de la planta y analizan los residuos entre los datos medidos y los estimados con el modelo.
- **Métodos basados en conocimiento:** se basan en crear reglas basadas en la experiencia al trabajar con la planta.

1.1.2. Métodos de FDI basado en datos

Como se ha expuesto, existen tres grandes grupos de métodos FDI, pero los métodos basados en datos presentan varias ventajas que han hecho que sean los elegidos por gran cantidad de autores [Qin, 2012, Yin et al., 2014, Ge, 2017]. Por un lado, en las plantas industriales cada vez se dispone de una mayor cantidad de datos debido a la creciente presencia de sensores, controladores, etc. en las mismas, y al aumento del grado de interconexión de todos estos elementos. Además, el incremento en la capacidad de almacenamiento de datos, permite aumentar la frecuencia de muestreo de los sensores y, por tanto, permite recoger un mayor volumen de información en el mismo tiempo.

Por otro lado, los métodos basados en datos basan su funcionamiento en analizar los datos recogidos, pero sin la necesidad de modelar la instalación ni de poseer un amplio conocimiento de la misma, como sí ocurre con los métodos basados en modelos y los basados en conocimiento, respectivamente. En instalaciones que presenten un alto grado de complejidad puede ser inasumible tanto hacer modelos como disponer de un sólido conocimiento previo de dichas plantas [Zhu et al., 2017]. Además, en el caso del modelado, al disponer de una mayor cantidad de datos, se hace más complejo realizar

dicho modelado, así como usar e interpretar estos modelos [Ge, 2017]. Al contrario que estos dos tipos de métodos, los métodos basados en datos son capaces, por sí mismos, de extraer esa información de los datos recogidos [Yin et al., 2014].

Por tanto, los métodos FDI basados en datos se pueden utilizar sencillamente contando con los datos que se pueden recoger en la planta, y esto va a facilitar mucho la implementación y el posterior uso de los mismos, pudiendo hacerse, incluso, de forma automatizada. La monitorización de la planta usando estos métodos se puede abordar de dos formas: como un problema de *clasificación*, en el que se debe disponer de datos de la planta trabajando en diferentes condiciones de fallo y crear un modelo de cada uno de esos fallos, de forma que se pueda determinar en qué condición está trabajando: normal o fallo, y en qué tipo de fallo se encuentra el sistema, en su caso; o construyendo un *modelo del sistema* trabajando en condiciones normales, para detectar cuándo los datos recogidos del sistema no encajan con este modelo. Obviamente, la segunda propuesta requiere un menor esfuerzo de modelado y es mucho más sencillo disponer de datos de funcionamiento normal que de funcionamiento bajo condiciones de fallo [Escobet et al., 2019].

Existe un amplia variedad de métodos FDI basados en datos, por ejemplo, los métodos llamados computacionales, que son los que se obtienen utilizando técnicas de inteligencia computacional como redes neuronales, árboles de decisión, métodos de clustering o agrupamiento, máquinas de vectores soporte (SVM), etc. Estos métodos se pueden usar haciendo clasificación o desarrollando modelos de regresión.

Otra clase de métodos de FDI basados en datos son los métodos estadísticos, en los que se supone un comportamiento estadístico de los datos. Estos métodos, a su vez, se pueden dividir en métodos paramétricos, en los que se supone conocida la distribución de probabilidad de los datos, como son los métodos basados en SPC: control estadístico de procesos, entre los que se incluyen los métodos PCA (análisis de componentes principales), PLS (mínimos cuadrados parciales), ICA (análisis de componentes independientes), etc. y los métodos no paramétricos, en los que la distribución de probabilidad de los datos no es conocida ni está parametrizada, como son los métodos basados en histogramas [Escobet et al., 2019].

1.1.3. Métodos distribuidos de FDI basados en datos

Cuando se analizan los datos recogidos en una planta es común tratar todos los datos en bloque. Pero debido a la mejora en los sensores y en los sistemas de control distribuido de las plantas, que recogen una enorme cantidad de datos de operación, así como a la creciente complejidad de las plantas industriales, que han ocurrido en los últimos años, estos métodos de FDI tradicionales han dejado de ser válidos para trabajar con la planta completa. Además hay que tener en cuenta las nuevas características que

tienen los datos industriales y que deben ser abordadas por los métodos de FDI [Ge and Chen, 2016, Ge, 2017]:

- La **cantidad de datos** recogidos en una planta compleja es enorme y eso plantea retos en su captura, gestión y almacenamiento. Además, es difícil interpretar y extraer la información que contienen esos datos.
- En este tipo de plantas se van a recoger **datos de diferente tipo**: medidas de sensores, imágenes, grabaciones de audio, etc.
- **Diferentes frecuencias de muestreo** de cada variable, ya que se pueden encontrar sensores que envían su información cada poco tiempo (segundos, minutos), pero hay medidas que requieren tiempos más largos (horas), como puedan ser análisis realizados en laboratorio.
- Puede no haber **datos suficientes** de acuerdo al tamaño de la instalación para su caracterización.

Todo esto hace que para monitorizar plantas de gran escala hayan surgido un nuevo tipo de métodos de FDI basados en datos, los métodos distribuidos. Sin embargo, existen una serie de retos que se deben abordar y resolver dentro del campo de los métodos FDI distribuidos [Ge, 2017]:

1. **Monitorización de calidad e indicadores KPI:** Hasta ahora no se ha tenido en cuenta la información relativa a indicadores de calidad en los métodos de monitorización, a pesar de su crucial importancia en los procesos industriales. Aunque, en métodos distribuidos esto puede ser muy complejo debido precisamente a la descentralización del proceso.
2. **Localización de las causas de los fallos:** Al incrementarse el tamaño de la planta, cada vez se hace más complejo encontrar las causas de los fallos, a pesar de que la parte de la detección de fallos sí se haya visto mejorada. Este problema puede ser abordado incorporando los conocimientos previos disponibles de la planta a los métodos FDI basados en datos.
3. **Fusión de la información disponible para realizar la toma de decisiones:** Es necesario disponer de técnicas que permitan la fusión de información de diferente tipo y proveniente de diferentes fuentes que nos vamos a encontrar. Como se ha dicho, no toda la información es del mismo tipo (medidas de sensores, imágenes, etc.) y, además, al descentralizar la planta, van a existir varias unidades de procesamiento que van a entregar una amplia variedad de resultados. Todo esto plantea problemas de cara a fusionar la información disponible para conseguir ejecutar una toma de decisiones relativa al diagnóstico de la planta.
4. **Sistemas multiagente para un modelado colaborativo:** Al dividir la planta se va a contar con una serie de entes individuales (agentes)

que, aunque sean independientes, deben trabajar de forma coordinada. Por tanto, se deben diseñar estructuras de colaboración que den lugar a una monitorización más eficaz.

5. **Big data:** Como se ha dicho anteriormente, la cantidad de datos disponibles en las plantas industriales ha crecido enormemente, y esto plantea varios retos como son el almacenamiento y procesado de estos datos. Una forma de abordarlos puede ser mediante el trabajo “en la nube” con estos datos, también se pueden adaptar técnicas propias del *Big data* para llevar a cabo la monitorización, como el procesado masivo en paralelo.

Parte de estos retos serán abordados en el desarrollo de la presente tesis. Además, esta nueva forma de trabajar, que aplica métodos de FDI distribuidos, permite solventar algunos problemas que aparecen al trabajar de forma centralizada [Jiang and Huang, 2016, Jiang et al., 2016, Ge and Chen, 2016]:

- **Capacidad de tolerancia a fallos:** Un sistema centralizado, una vez que ha detectado un fallo en una parte de la planta, pierde parte de su capacidad para detectar otro fallo en otro punto de la instalación. Es decir, su capacidad para detectar fallos simultáneos es reducida.
- **Aumenta la fiabilidad:** El sistema centralizado al procesar todos los datos en un único bloque, si en algún momento un sensor no envía un dato, o lo envía con errores, o se pierde alguna vía de comunicación, es posible que el proceso de monitorización se detenga.
- **Eficiencia económica:** Si los sensores están muy separados entre sí, puede ser más económico contar con una unidad de procesamiento en cada uno de los emplazamientos de los sensores.
- **Aumenta la flexibilidad y escalabilidad:** Los métodos distribuidos son más flexibles que los centralizados. Si se quiere añadir o borrar alguna parte del proceso, la reconfiguración y la actualización del modelo distribuido es mucho más fácil, incluyendo la puesta en marcha del sistema, el modelado basado en datos y la toma de decisión. Es decir, la escalabilidad de los métodos distribuidos es mucho mejor que la de los métodos centralizados.
- **Disminuye la complejidad computacional del sistema.**

1.2. Motivación

La principal motivación de esta tesis es por tanto, el desarrollo de nuevos métodos de detección y diagnóstico de fallos, basados en datos y, además, distribuidos y aplicados a plantas complejas y de gran escala.

En primer lugar, los métodos con los que se trabajará en esta tesis son métodos basados en datos, que pongan en valor la gran cantidad de datos

disponibles hoy en día en las plantas industriales. Esto se puede hacer ya que son métodos muy flexibles y no necesitan obtener modelos de primeros principios que suelen ser muy complejos y costoso. Tampoco requieren un conocimiento experto de la planta, que no siempre está disponible. Además los métodos serán distribuidos, ya que tienen muchas ventajas sobre los métodos centralizados, como se ha comentado anteriormente, especialmente cuando se aplican a plantas complejas y de gran escala.

Dentro de las metodologías distribuidas se pueden encontrar de diferentes tipos, en esta tesis se proponen tres estrategias diferentes para llevar a cabo la monitorización distribuida:

1. **Distribución de los modelos.** Dividir la planta en bloques de variables y aplicar un método FDI en cada bloque. Luego, en un bloque central, se usará un método de toma de decisión múltiple que fusione la salida de cada bloque.
2. **Distribución de métodos FDI.** En cada unidad de procesamiento se tratan todos los datos recogidos pero cada bloque se implementa un método FDI diferente. Nuevamente, en este caso se necesita un método de toma de decisión múltiple que considere la salida de cada método y avise de la situación global de la planta, normal o con fallo.
3. **Distribución de los análisis.** Cuando se tienen muchos datos y/o es necesario realizar cálculos complejos y que consumen mucho tiempo se pretende procesar esta información de forma distribuida utilizando técnicas de procesamiento en paralelo.

Por tanto, lo que se ha buscado con las tres aproximaciones mencionadas es la implementación de varias unidades de procesamiento independientes que monitorizarán la planta, cada una de ellas trabajando de forma autónoma. Finalmente se dispondrá una unidad de procesamiento central que proporcionará un diagnóstico único para toda la planta, a partir de la fusión de los resultados de cada una de las unidades de procesamiento anteriores.

1.3. Objetivos

Los principales objetivos de este trabajo doctoral son:

1. **Testear diferentes métodos FDI distribuidos basados en PCA,** para comprobar si la detección de fallos distribuida ofrece ventajas frente a la centralizada.

Para conseguir este objetivo, se ha hecho una búsqueda bibliográfica de métodos de FDI distribuidos basados en PCA, y diferentes metodologías de distribuir el cálculo del PCA. Estos métodos se han implementado aplicándose sobre dos plantas complejas para probar la eficacia de los diferentes métodos distribuidos.

2. **Diseño y desarrollo de nuevos métodos distribuidos de FDI basados en datos**, donde se hace una distribución de los modelos.

En este caso se dividirá la planta en bloques de variables y en cada uno de ellos se implementará un método FDI. Esto implica que hay que desarrollar métodos adecuados para realizar la construcción de los bloques y contar con técnicas que sirvan para fusionar los resultados obtenidos en cada uno de los bloques.

Para dividir la planta se quiere poder hacerlo analizando los datos disponibles, es decir, sin necesidad de disponer de conocimientos previos de dicha planta. Esto se hace con la intención de facilitar el proceso de descentralización.

En cuanto a los métodos FDI de cada bloque, en esta tesis se ha optado por métodos estadísticos de FDI basados en datos, ya que, al igual que con los métodos aplicados para la descentralización, se pueden aplicar sin disponer de un conocimiento previo de la planta.

3. **Diseño y desarrollo de nuevos métodos de FDI basados en métodos distribuidos.**

En este caso lo que se pretende distribuir son los métodos FDI. Es decir, se dispondrá de varios métodos, cada uno de los cuales analizará los datos provenientes de la planta por separado. Posteriormente, estos resultados se fusionarán para obtener una diagnosis única y global de la planta.

4. **Diseño y desarrollo de nuevos métodos FDI basados en computación distribuida.**

En este caso, en cada nodo de procesamiento se realizarán diferentes modelos dinámicos de cada variable del proceso, mediante técnicas de series temporales, y donde se elegirá el mejor modelo de todos los calculados para cada variable. Las desviaciones que se encuentren entre los modelos obtenidos y los valores reales de las variables serán enviados a una unidad central que las usará para realizar la detección de fallos en la planta usando varias técnicas de detección de fallos basadas en control estadístico de procesos (SPC).

5. **Comprobar las prestaciones de las metodologías propuestas mediante casos de estudio usando *benchmarks* de plantas complejas.**

Por tanto, este último objetivo, consiste en la validación y verificación de la eficacia de los métodos propuestos, en tres *benchmarks* ampliamente utilizados en la literatura: una depuradora de aguas residuales, concretamente el modelo de Simulación BSM2 desarrollado por los Grupos de Trabajo COST Action 682 y 624, y el IWA Task Group [Alex et al., 2008].

El segundo *benchmark* es un planta química, la *Tennessee Eastman Process* (TEP) [Downs and Vogel, 1993]. Finalmente, la tercera planta utilizada es un modelo de red de distribución de agua potable implementada mediante el software de simulación EPANET [Rossman, 2000b].

1.4. Organización de la tesis

Este trabajo está organizado de la siguiente forma: en el Capítulo 1 se presenta una breve introducción a la monitorización de procesos en plantas industriales así como algunos conceptos básicos relacionados con los métodos para llevar a cabo dicha monitorización. También se detallan la motivación y los objetivos del presente trabajo.

El Capítulo 2 contiene el estado del arte correspondiente al control estadístico de procesos y la detección e identificación de fallos. Concretamente se presenta un introducción a algunas técnicas del monitorización de procesos basadas en gráficas de control. A continuación se profundiza en diferentes métodos que trabajan con datos.

Seguidamente, se pasa a detallar distintas formas de llevar a cabo la detección y diagnosis de fallos de forma distribuida. Debido a que la metodología distribuida genera un resultado en cada uno de los bloques utilizados, también se presenta una forma de abordar la fusión de dichos resultados mediante toma de decisiones.

El Capítulo 3 presenta una primera aproximación a las técnicas FDI distribuidas. Aquí se prueba un método para aplicar PCA de forma distribuida y se compara con otras propuestas distribuidas y no distribuidas halladas en la literatura. El objetivo es comprobar si es ventajoso trabajar de forma distribuida.

En el Capítulo 4 se muestran los trabajos realizados para llevar a cabo la división de una planta en la que poder hacer una diagnosis descentralizada y que genere una decisión única respecto a esta diagnosis. Seguidamente, se muestran y se analizan los resultados obtenidos con esta metodología en el benchmark *Tennessee Eastman Process* (TEP).

El Capítulo 5 está enfocado en la identificación de fallos. En este Capítulo se propone disponer de varios centros de procesamiento en los que cada uno analiza todos los datos de la planta pero usando métodos de identificación distintos. Los métodos propuestos se han aplicado en dos casos de estudio distintos: una depuradora de aguas residuales y la planta *Tennessee Eastman Process* (TEP), mostrándose y comentándose los resultados obtenidos.

En el Capítulo 6 se hace una propuesta de método de detección de fallos distribuida donde la distribución propuesta se halla en la computación: cada uno de los nodos realizará una serie de modelos de las variables del proceso. Esta propuesta está basada en el análisis de series temporales. Dicho método

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS

se ha testeado sobre la planta *Tennessee Eastman Process* (TEP) y sobre la depuradora de aguas residuales.

Finalmente, en el Capítulo 7 se presentan las conclusiones y las contribuciones principales de este trabajo, dándose, además, las futuras líneas de trabajo con las que se puede ampliar este estudio.

Capítulo 2

ESTADO DEL ARTE

2.1. Introducción

Realizar una correcta monitorización de toda instalación es vital para asegurar unas condiciones óptimas de la producción de una empresa. El enorme avance que han experimentado las técnicas y los equipos de control industrial ha permitido mejorar la producción de las empresas en términos de calidad, plazos, costes, etc. A pesar de ello, siguen surgiendo problemas en las plantas que pueden hacer que estas se alejen del comportamiento esperado y/o planificado. De hecho es frecuente que la abundancia de sistemas de control complejos, incluyendo todos los dispositivos asociados: sensores, actuadores, etc. haga que la tarea de monitorizar el proceso sea enormemente compleja, y los operadores de la planta requieren de ayudas para realizar dicha tarea.

El tener en todo momento la planta bajo control y tener conocimiento del estado real de la planta puede evitar la aparición y propagación de fallos que implicarían graves pérdidas para la empresa como: reducción de la calidad, de la seguridad, aparición de accidentes, reducción de la producción, etc., por tanto, es imprescindible que sean detectados y corregidos en el menor tiempo posible.

La monitorización en plantas complejas y de gran tamaño, y, en concreto la detección y el diagnóstico de fallos, han sido estudiados por muchos autores desde varias perspectivas. Y, concretamente, los métodos de control estadístico multivariante se han utilizado de forma efectiva para las tareas de detección de fallos [Venkatasubramanian et al., 2003c, Qin, 2003, Qin, 2012, Yin et al., 2014, Ge, 2017, Bakdi and Kouadri, 2017].

2.2. Control estadístico de procesos

El control de calidad y la detección de fallos son conceptos muy relacionados. Ambos tratan de detectar malfuncionamientos o fallos que pueden

desembocar en graves pérdidas económicas para una empresa. Dentro de estos métodos se encuentra el Control Estadístico de Procesos (SPC) que aplica técnicas estadísticas para comprobar si el comportamiento observado en un proceso encaja con lo planificado [Heizer and Render, 2013].

Es un método que se ejecuta mediante el uso de gráficos de control, que se basan en técnicas estadísticas y permiten disponer de criterios objetivos para distinguir pequeñas variaciones del proceso, de poca relevancia, frente a eventos graves que requieren de acciones por parte de los operadores de la planta.

2.2.1. Variabilidad en los procesos de producción

Ningún proceso es capaz de generar dos productos o servicios iguales, aunque el proceso esté trabajando en condiciones óptimas. Esto es debido a que dichos procesos incluyen muchas fuentes de variación. Uno de los objetivos de los operadores de las plantas es eliminar, en lo posible, las fuentes de variación o reducir sus efectos. Existen dos causas principales de variaciones: causas comunes y causas especiales. [Paz and Gomez, 2014]

Las primeras, causas comunes, son fuentes de variación puramente aleatorias, no identificables e imposibles de evitar. Dentro de esta categoría se pueden encontrar causas como variaciones en las condiciones ambientales de un proceso, variaciones en la materia prima utilizada (siempre dentro de unas tolerancias), etc.

Dentro del segundo tipo de causas, causas especiales, se encuentran factores de variación que pueden ser identificados y eliminados o reducidos en gran parte. Ejemplos de este grupo son formación inadecuada del personal, desgaste de las instalaciones, etc.

2.2.2. Control estadístico univariable de procesos

En instalaciones de pequeñas dimensiones, con pocas variables medidas, es posible analizarlas de forma individual. Esto se ha venido realizando mediante las llamadas gráficas de control. Estas representan gráficamente los valores medios y la variación de cada una de las variables medidas con el objetivo de detectar anomalías, y determinar si la instalación trabaja en condiciones normales de operación o no. Es decir, es una forma rápida de ver la evolución del proceso y identificar su estado, aunque tiene la desventaja de que no permite analizar un gran número de variables simultáneamente, ya que cada una de ellas tiene su gráfica de control.

Existen gran variedad de gráficas de control, por ejemplo: gráficas Shewhart [Shewhart, 1938], gráficas CUSUM [Page, 1961], gráficas de Media Móvil Exponencialmente Ponderada (EWMA, por sus siglas en inglés (*Exponentially Weighted Moving Average*)) [Hunter, 1986]. Estas gráficas han sido usadas en aplicaciones como el control de calidad en la industria, econo-

mía, informática, ecología, gestión sanitaria, etc. [Frisén, 2009, Harrou et al., 2013, Park et al., 2014, Kadri et al., 2016]. Se presenta a continuación una pequeña descripción de estas gráficas:

- **Gráficas de control *Shewhart*** [Shewhart, 1938]: representa la evolución temporal de la media de una variable estableciendo unos límites superior e inferior. Si se supera alguno uno de estos umbrales, se detecta un fallo. De cara a reducir el nivel de falsas alarmas, se suele requerir un cierto número de instantes consecutivos con valores anormales para activar la alarma de fallo.
- **Gráficas de sumas acumuladas (CUSUM)** [Woodward and Goldsmith, 1964, Page, 1954]: en esta gráfica se representa la suma acumulada de desviaciones en cada observación respecto de la media. Es capaz de detectar pequeñas variaciones más rápido que los gráficos de Shewhart. Esta técnica requiere que se establezca un valor k de observaciones pasadas que son las que se usan para calcular la suma acumulada en el momento actual.
- **Gráfica de media móvil exponencialmente ponderada (EWMA)** [Hunter, 1986]: Este método aplica un filtro a los datos. Calcula una función de decisión para cada observación $z_i(t)$ según los datos actuales y los valores promedio anteriores:

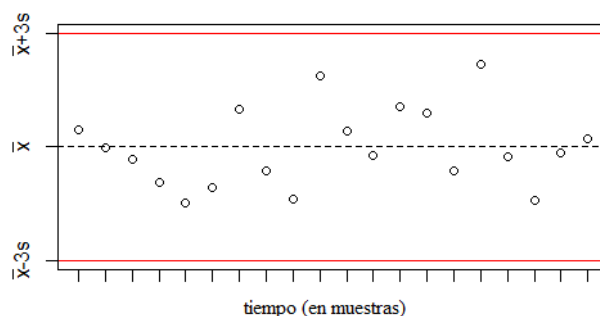
$$z_i(t) = \lambda x_i(t) + (1 - \lambda)z_i(t - 1) \quad (2.1)$$

donde x_i es el i -ésimo valor de la variable analizada en tiempo t , y λ es el grado de ponderación que determina el nivel de memoria de la función decisión EWMA. Con $0 \leq \lambda \leq 1$, valores bajos de λ dan mayor influencia a los valores del pasado, mientras que valores altos dan más importancia al valor actual. Los límites de control se calculan así:

$$UCL, LCL = \mu_0 \pm L\sigma_0 \sqrt{\frac{\lambda}{2 - \lambda}} \quad (2.2)$$

donde L es el ancho de los límites de control (y que determina los límites de confianza, y se ajusta buscando un cierto nivel de falsas alarmas) y μ_0 y σ_0 son, respectivamente, la media y la desviación típica de los datos en condiciones sin fallo.

En la figura 2.1 se muestra un ejemplo de gráfica de control del tipo Shewhart. Se muestran los valores medidos de una variable x , que oscilan alrededor de su valor medio \bar{x} . En este caso, los umbrales se han establecido entre los valores $\bar{x} - 3s$ y $\bar{x} + 3s$ siendo s la desviación típica de la variable x .

Figura 2.1: Gráfica de control *Shewhart*

Un grave problema que tienen estas gráficas es que no tienen en cuenta las correlaciones entre variables, presentes en cualquier fenómeno físico. Esto puede hacer que cuando se presente un fallo que implique un cambio en las relaciones entre variables (porque la planta haya cambiado su modo de funcionamiento), ninguna variable sobrepase los límites de las gráficas y no se detecte el fallo [Nomikos and MacGregor, 1995].

La forma de abordar este problema es aplicar métodos que no sólo analicen el comportamiento de las variables de forma individual sino que tengan en cuenta, también, las relaciones entre variables. Existen métodos como el Análisis de Componentes Principales (PCA, por sus siglas en inglés, (*Principal Component Analysis*)), o el Análisis de Variaciones Canónicas (CVA, *Canonical Variate Analysis*) que son capaces de ello, y además, pueden analizar la correlación temporal de las variables (la influencia del estado de las variables en un instante de tiempo sobre el estado de las mismas instantes después).

2.3. Métodos de detección y diagnóstico de fallos

Por tanto, se hace necesario disponer de métodos que permitan monitorizar la planta teniendo en cuenta todas las variables medidas en ella de forma simultánea. Estos métodos son los métodos de detección y diagnóstico de fallos. Estas técnicas son capaces de procesar la información recogida en la planta, no sólo los valores individuales de cada una de las variables sino, también, las relaciones entre ellas, y devolver un número reducido de indicadores que permitirán al usuario conocer el estado de la instalación: si hay o no un fallo y, si lo hay, en qué punto se ha producido el mismo. El hecho de extraer la información de la planta y concentrarla en unos pocos indicadores facilita enormemente la tarea de la monitorización, haciendo, incluso que se pueda automatizar este proceso. Como se comentó en el capítulo an-

terior, las etapas que llevan a cabo estos métodos son cuatro: detección de la presencia de un fallo, identificación de las variables responsables de dicho fallo, diagnóstico del fallo (determinación del tipo, localización, tamaño y causa del fallo), y, finalmente, recuperación del proceso eliminando el efecto del mismo [Qin, 2003, Liu and Chen, 2014].

Como se vio en el capítulo anterior, existe una clasificación de estos métodos ampliamente aceptada, que los divide en métodos basados en datos o en señal, métodos analíticos y métodos basados en conocimiento [Miljković, 2011].

2.3.1. Métodos basados en datos

Estos métodos procesan datos medidos en la planta para extraer información relevante de las variables y de las relaciones entre ellas. Son métodos que no requieren de un profundo conocimiento previo de la instalación ni de entrenar modelos de la misma, y por tanto se pueden implementar de una forma relativamente simple.

En esta categoría se pueden encontrar diversas técnicas como son:

- **Alarmas:** Las variables medidas son comparadas con unos valores límites constantes y preestablecidos que, de ser sobrepasados, indicarían un fallo [Isermann, 2006].
- **Técnicas estadísticas multivariantes:** En un proceso se toman medidas de una serie de variables que comparten un cierto grado de correlación entre ellas (correlación espacial) y, a su vez, el valor de cada variable está relacionado con los valores de dicha variable en los instantes previos (correlación temporal). Dentro de este tipo de métodos se encuentran técnicas como: el Análisis de Componentes Principales (PCA), que es una técnica que permite reducir la dimensionalidad en términos de la varianza de los datos [Kourti and MacGregor, 1996, Isermann, 2006]; también se puede encontrar en este campo el análisis del discriminante de Fisher (FDA), que es otro método que también aplica una reducción de la dimensionalidad de un espacio de variables [Song et al., 2010, Wang et al., 2012]; otro método dentro de este área es el método de Mínimos cuadrados parciales (PLS) [Kourti and MacGregor, 1995].
- **Redundancia física:** Consiste en emplazar múltiples componentes físicos idénticos (sensores, actuadores y componentes del sistema) por toda la planta de modo que se puedan realizar comprobaciones de las posibles discrepancias entre las salidas de elementos supuestamente idénticos. Un análisis posterior de consistencia de los datos recogidos permiten encontrar fallos y el emplazamiento de los mismos [Dorr et al., 1997, Venkatasubramanian et al., 2003a].

- **Análisis en frecuencia:** En determinados casos se pueden realizar análisis en el dominio de la frecuencia de algunas de las variables medidas en la planta. Esto permite obtener una valiosa información acerca del estado de operación de la planta. Adicionalmente, cada fallo puede ir asociado a un cierto patrón de frecuencias, de modo que sea posible diagnosticar el tipo de fallo una vez que se ha detectado [Edwards et al., 1998, Senobari et al., 2018].
- **Técnicas basadas en inteligencia computacional:** Como pueden ser redes neuronales, máquinas de vectores soporte (SVM), árboles de decisión, etc. Estas técnicas se usan en este campo como clasificadores, es decir, para clasificar 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 se realizan mediante la presentación de datos de muestra de cada una de la situaciones que se quieren clasificar. El principal inconveniente es que si no se dispone de datos de fallo estos métodos no pueden aprender dicho modo de funcionamiento [Isermann, 2006, Abiodun et al., 2018, Wu and Zhao, 2018].

2.3.2. Métodos analíticos

Este tipo de métodos emplean un modelo de la planta y constan de dos pasos: primero se deben obtener los residuos procedentes de la discrepancia entre las predicciones del modelo y los valores medidos en la planta, y, posteriormente, se debe realizar una toma de decisión basada en esos residuos.

El método más utilizado dentro de los métodos analíticos es el de redundancia analítica. En una primera etapa, las entradas y las salidas del sistema se procesan mediante un algoritmo determinado para generar los residuos que inicialmente son cero, pero que se desvían del cero de forma característica cuando se está en presencia de un fallo. La forma en la que se generan los residuos depende del método aplicado; fundamentalmente, la principal diferencia estriba en la forma de calcular el modelo del proceso, que se lleva a cabo mediante: redes neuronales, filtros de Kalman, ecuaciones de paridad, observadores de estados, cálculo de los posibles conflictos y estimación de parámetros. Cuando se realiza el segundo paso, que es la toma de decisión, se analizan los valores de los residuos para decidir si hay fallo o no. Una forma de tomar esa decisión consiste en comprobar si los residuos superan un cierto umbral. También se pueden filtrar los residuos aplicando filtros como el de media móvil, o pueden utilizarse métodos de la teoría estadística de decisión [Venkatasubramanian et al., 2003a, Castillo and Edgar, 2008].

Por supuesto, este tipo de métodos requieren de un buen modelado de la planta para su correcto funcionamiento, cosa que no siempre es sencilla, sobre todo en plantas complejas.

2.3.3. Métodos basados en conocimiento

El fundamento de estos métodos es utilizar el conocimiento previo que se tenga de la planta y que se utilizará para diagnosticar el estado de la planta. Hay distintas técnicas que se pueden utilizar:

- **Sistemas expertos:** En un sistema experto se recoge el conocimiento de una serie de 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 diagnosis 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 [Angeli, 2010].
- **Sistemas basados en lógica difusa:** La diagnosis de un sistema no tiene porque ser binaria: hay fallo o no hay fallo. Se puede dar un resultado que exprese la aparición de un fallo con un cierto grado de intensidad. En base al conocimiento disponible y a los datos medidos en la planta se puede crear un sistema de reglas para realizar la diagnosis de la planta [Miljković, 2011]. Un sistema de lógica difusa se puede utilizar tanto para la evaluación de los residuos, incorporando el conocimiento disponible del sistema en forma de reglas difusas, como para llevar a cabo la generación de residuos, aprovechándose de que se puede utilizar un sistema lógico difuso como aproximador universal 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 [Manali et al., 2009, Azar, 2010].

Como se ha dicho, para aplicar estos métodos es necesario contar con un adecuado conocimiento de la planta, cosa que no siempre es posible.

2.3.4. Características de un sistema de detección y aislamiento de fallos

Un sistema de detección y aislamiento de fallos, para ser eficaz, debe cumplir los siguientes requisitos [Puigjaner et al., 2006]:

- **Rapidez en la detección del fallos:** debe ser lo mayor posible.
- **Capacidad de aislamiento:** para distinguir entre distintos fallos.
- **Robustez:** para detectar fallos en condiciones ruidosas, cambiantes, etc.

- **Adaptabilidad:** a nuevas condiciones de operación, cambios ambientales, etc.
- **Ratio de detecciones correctas:** si los fallos detectados corresponden con fallos reales.
- **Ratio de falsas alarmas:** número de fallos detectados que no corresponden con un fallo real en la planta.
- **Sensibilidad:** el mínimo tamaño de fallo que se puede detectar.

2.3.4.1. Evaluación de los métodos de detección de fallos

Cuando se prueba un nuevo método de detección de fallos, se requiere medir su eficacia para poderlo comparar con métodos ya existentes y ver si aporta alguna mejora. Hay cuatro índices ampliamente utilizados para realizar esta evaluación: la tasa de fallos no detectados (en inglés, *Missed Detection Rate* (MDR)), la tasa de detección falsas alarmas (en inglés, *False Alarm Rate* (FAR)), el retardo en la detección del fallo y el número de fallos detectados ([Russell et al., 2000, Detroja et al., 2007, Bakdi and Kouadri, 2017, Tong et al., 2017]).

Una buena técnica de detección de fallos debe ser robusta frente a cambios en los datos de entrenamiento, sensible a todos los posibles fallos que aparezcan en la planta, y rápida en la detección. La robustez del método se determina comparando la tasa de falsas alarmas cuando se trabaja en condiciones de operación normales con el nivel de significación en el que se basa el umbral. La sensibilidad se mide usando la tasa de no detección, que cuantifica, en porcentaje, cuántos datos con fallo son calificados como datos normales, y usando el número de fallos que el método es capaz de detectar. La rapidez de la detección se mide con el retardo en la detección.

Por tanto, la tasa MDR se calcula de la siguiente forma:

$$MDR = 100 \frac{N_{F,N}}{N_F} \% \quad (2.3)$$

donde $N_{F,N}$ es el número de datos con fallo que son detectados como normales y N_F es el número de datos con fallo.

La tasa FAR mide, también en porcentaje, el número de datos sin fallo que el método ha clasificado como datos con fallo en comparación con el total de datos sin fallo analizados, es decir:

$$FAR = 100 \frac{N_{N,F}}{N_N} \% \quad (2.4)$$

donde $N_{N,F}$ es el número de datos sin fallo (o normales) identificados como datos con fallo y N_N es el número total de datos sin fallo analizados.

El retardo en la detección indica cuántos instantes, desde que el fallo aparece, necesita el método para avisar de dicho fallo. Y el número de fallos

detectados cuantifica el número de fallos diferentes que es capaz de detectar el método, del total de fallos probados.

Por tanto, cuando se hace una comparación de métodos, será mejor aquel que tenga menores tasas de MDR y de FAR, menores retardos en la detección y que sea capaz de detectar una mayor número de fallos distintos.

2.4. Métodos de detección e identificación de fallos basados en datos

En este tipo de métodos se trabaja con los datos tomados en la planta cuando esta trabaja en condiciones normales de operación. Si se dispone de m sensores en la planta y se toman medidas en n instantes, se dispone de una matriz de datos de dimensiones $n \times m$. En general, estos métodos requieren de una gran cantidad de datos para ser entrenados [Miljković, 2011, Ge et al., 2013, Tidiri et al., 2016].

Los métodos que trabajan con datos del proceso se pueden clasificar en métodos de extracción de información cualitativa y cuantitativa [Venkatasubramanian et al., 2003c]. Dentro de los primeros se pueden encontrar los sistemas expertos [Poli and Boudet, 2018] y los métodos de modelado de tendencias [Gao et al., 2010]. Dentro de los métodos de extracción de información cuantitativa existen métodos basados en inteligencia computacional, como las redes neuronales [Heo and Lee, 2018, Wu and Zhao, 2018] y métodos estadísticos como PCA [Jackson, 1991, Liu and Chen, 2014, Bakdi and Kouadri, 2017], PLS (*Partial Least Squares*) [Wold, 1982, Zhang and Hu, 2011], etc. El trabajo desarrollado en esta tesis se ha centrado en los métodos estadísticos de detección de fallos, presentándose, a continuación, un pequeño resumen de los mismos.

El método PCA básicamente permite reducir la dimensionalidad de los datos disponibles, analizando las interrelaciones entre variables, mientras que retiene la máxima cantidad de información presente en dichos datos [Chiang et al., 2001]. Su uso como método de detección de fallos se basa en decidir si las nuevas observaciones se pueden considerar dentro del modelo PCA obtenido o no. Este método ha sido utilizado por gran cantidad de autores, algunos de los cuales lo han modificado para obtener un mejor rendimiento en la monitorización. Por ejemplo, para analizar la posible correlación temporal en la planta existe el PCA dinámico (DPCA), que trabaja con datos que incluyen información con retardo [Ku et al., 1995]; también está el PCA adaptativo (APCA), que permite actualizar el modelo cuando aparece un fallo, buscando poder detectar nuevos fallos [Zumoffen and Basualdo, 2008]; para hacer el método menos sensible a los *outliers* se ha desarrollado el PCA robusto [Stanimirova et al., 2004, Candès et al., 2011]. Se puede encontrar más información sobre estos métodos basados en PCA en [Alvarez, 2013].

Otra técnica que reduce la dimensionalidad de los datos, al igual que el PCA, es el método PLS, que modela las relaciones entre dos bloques de variables del proceso (X : entradas, e Y : salidas), a la vez que reduce su dimensionalidad de forma simultánea. Extrae unas variables latentes que no sólo explican la variabilidad en los datos de entrada X , sino también, qué variabilidad de X es más útil para predecir las salidas Y [Zhang and Hu, 2011].

En los últimos años ha ido ganando popularidad otra técnica de reducción de dimensionalidad, el método CVA, para detección de fallos [Russell et al., 2000, Jiang et al., 2015, Li et al., 2018]. Realiza un análisis entre dos conjuntos de variables en un contexto temporal, es decir, entre datos pasados y datos futuros del proceso, de modo que transforma las variables a su forma canónica. Esta forma canónica define combinaciones lineales de las variables predictoras (datos pasados) de acuerdo a su capacidad de predecir el otro conjunto de variables (datos futuros). Esta característica hace que este método sea dinámico, es decir, que modela no sólo la correlación espacial, entre variables, sino temporal, entre los estados sucesivos de la planta.

Además de los métodos expuestos anteriormente, se pueden encontrar muchos otros métodos de monitorización multivariantes, por ejemplo, el Análisis de Componentes Independientes (ICA). Este método busca encontrar una representación lineal de datos de tipo no gaussiano de forma que los componentes obtenidos en el proceso sean independientes o, al menos, tan independientes como sea posible. El resultado es que se consigue capturar la estructura esencial de los datos y se puede aplicar para extracción de características, separación de señales, etc. [Wang et al., 2016, Zhang and Zhao, 2019]. El PCA, a diferencia del ICA, busca componentes ortogonales o no correlados. Por otro lado, la monitorización es similar, ya que se establecen unos límites para los componentes independientes, de tal modo, que si al analizar una nueva observación, sus componentes los sobrepasan se detectará un fallo. La ventaja que presenta este método es que los límites se pueden ajustar mejor y de forma más precisa que en el PCA [Kano et al., 2003].

2.5. Análisis de Componentes Principales (PCA)

El método de análisis de componentes principales o 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 [Jackson, 1991, Ge, 2017, Tong et al., 2017].

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 \mathbf{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 \mathbf{X}^{norm} .

Una vez hecho esto, se calcula la matriz de correlaciones y se descompone en valores singulares:

$$\mathbf{S} = \frac{1}{(n-1)} (\mathbf{X}^{norm})^T \mathbf{X}^{norm} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T \quad (2.5)$$

siendo $\mathbf{\Lambda}$ ($m \times m$) una matriz diagonal que contiene los valores singulares de \mathbf{X}^{norm} (que son, también, los valores propios de la matriz de correlación), 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 $\mathbf{\Lambda}$ 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 $\mathbf{\Lambda}_a$ de dimensiones $a \times a$. Al tomar el mismo número a de entre las primeras columnas de \mathbf{V} , resulta así una matriz \mathbf{P} (m filas y a columnas), llamada *Matriz de cargas*. Si multiplicamos la matriz \mathbf{X}^{norm} por la matriz \mathbf{P} , obtenemos otra matriz \mathbf{T} llamada de *Componentes principales de \mathbf{X} (scores)*:

$$\mathbf{T} = \mathbf{X}^{norm} \mathbf{P} \quad (2.6)$$

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 2.2 (a), se observa que los datos tienen mayor variabilidad en la dirección marcada por el vector PC1 de la figura 2.2 (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.

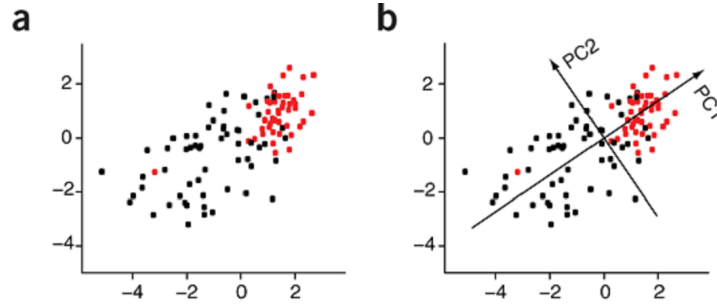


Figura 2.2: Componentes Principales

Finalmente, se escogen los componentes principales que representen la mayor parte de la variabilidad de los datos. En el ejemplo anterior, se podrían representar los datos usando 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 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:

$$\hat{\mathbf{X}}^{norm} \cong \mathbf{TP}^T \quad (2.7)$$

Si se usa $\hat{\mathbf{X}}^{norm}$ como aproximación de \mathbf{X}^{norm} , el error cometido es:

$$\mathbf{E} = \mathbf{X}^{norm} - \hat{\mathbf{X}}^{norm} \quad (2.8)$$

2.5.1. Elección del número de Componentes Principales

El problema de elegir cuantos componentes principales se toman para el modelo no es trivial. El objetivo que se persigue es conseguir una reducción de la dimensionalidad, para simplificar el modelo, al mismo tiempo que se retenga la mayor cantidad de información posible. En la literatura se pueden encontrar diferentes formas de realizar esta selección del número de componentes: porcentaje de varianza, Criterio de información de Akaike, error de reconstrucción, etc. Se suele optar frecuentemente por la selección en base al porcentaje de varianza, cuyos valores típicos suelen oscilar entre el 60 % y 90 % [Ferré, 1995, Jolliffe, 2002]. Aunque todos estos criterios requieren de la intervención del usuario y, por tanto, son subjetivos.

Algunos autores han propuesto métodos basados en validación cruzada que permiten hacer la elección del número de componentes principales de

una forma objetiva [Bro et al., 2008, Russo et al., 2019]. En estos casos lo que se busca es el número de componentes principales que minimiza el error de predicción.

2.5.2. Detección de fallos: Estadísticos T^2 y Q

Para detectar fallos se dispone de las herramientas estadísticas T^2 o estadístico de *Hotelling* y Q o Error Cuadrático de Predicción (en inglés: *Squared Prediction Error* o SPE) [Kourti and MacGregor, 1996].

El estadístico T^2 se calcula según la expresión siguiente:

$$T^2 = \mathbf{z}^T \mathbf{P} \mathbf{\Lambda}_a^{-1} \mathbf{P}^T \mathbf{z} \quad (2.9)$$

siendo \mathbf{z} una nueva observación ($1 \times m$), normalizada a media 0 y desviación típica 1, de la que se quiere calcular el valor de T^2 y $\mathbf{\Lambda}_a$ es la matriz ($a \times a$) diagonal formada con las primeras a filas y a columnas de $\mathbf{\Lambda}$. 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:

$$T_\alpha^2 = \chi_\alpha^2(a) \quad (2.10)$$

donde $\chi_\alpha^2(a)$ es el valor de la distribución acumulada χ^2 a grados de libertad [Jiang and Huang, 2016]. Los valores típicos de α que se suelen tomar son 0,01 y 0,05.

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

$$Q = \mathbf{r}^T \mathbf{r} \quad (2.11)$$

donde \mathbf{r} es la proyección de la observación \mathbf{z} en el subespacio de los residuos, y se obtiene de la forma:

$$\mathbf{r} = (\mathbf{I} - \mathbf{P} \mathbf{P}^T) \mathbf{z}^T \quad (2.12)$$

siendo \mathbf{I} la matriz identidad de dimensiones: $m \times m$. El límite que Q no debe superar para que el proceso esté bajo control es:

$$Q_\alpha = g^{SPE} \chi_\alpha^2(h^{SPE}) \quad (2.13)$$

con:

$$g^{SPE} = \frac{\theta_2}{\theta_1}, \quad h^{SPE} = \frac{\theta_1^2}{\theta_2} \quad (2.14)$$

donde $\chi_\alpha^2(h^{SPE})$ es el valor de la función de distribución χ^2 acumulada con h^{SPE} grados de libertad y probabilidad α . Y $\theta_1 = \sum_{i=a+1}^n \lambda_i$, $\theta_2 =$

$\sum_{i=a+1}^n \lambda_i^2$, siendo λ_i el i -ésimo autovalor de la matriz de covarianza S [Liu et al., 2014].

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 el método PCA detectará un fallo en el sistema. Con este 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 retarde demasiado la detección del fallo).

En resumen, se puede decir que T^2 mide la variabilidad del proceso capturada por los componentes principales y Q mide la distancia de las nuevas observaciones respecto al modelo PCA obtenido, es decir, detecta si el modelo de correlación ya no es válido. Un ejemplo de un caso de tres dimensiones reducidas a dos componentes principales se puede ver en la figura 2.3. El punto e_1 está muy alejado de los puntos de comportamiento normal, por lo que superará el umbral de T^2 , pero el modelo PCA con los componentes PC1 y PC2 sigue siendo válido puesto que ese punto está contenido en el subespacio definido por esos dos componentes principales. En el caso del punto e_2 no sólo hay una distancia relevante a los puntos de comportamiento normal (superará, por tanto, el umbral de T^2) sino que además está fuera de la región definida por PC1 y PC2, por lo que se superará también el umbral de Q .

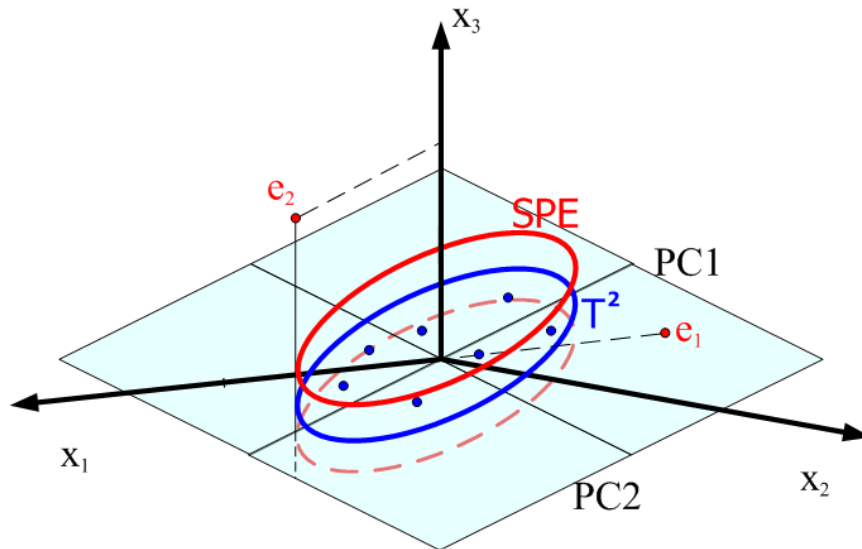


Figura 2.3: Componentes principales

Existe un tercer estadístico que es combinación de los dos anteriores:

φ [Yue and Qin, 2001, Qin, 2012].

$$\varphi = \frac{Q}{Q_a} + \frac{T^2}{T_\alpha^2} \quad (2.15)$$

El proceso se considerará bajo control si $\varphi < \varphi_\alpha$. El umbral φ_α se calcula así:

$$\varphi_\alpha = g^\varphi \chi_\alpha^2(h^\varphi) \quad (2.16)$$

con:

$$\begin{aligned} g^\varphi &= \frac{\frac{a}{(T_\alpha^2)^2} + \frac{\theta_2}{Q_\alpha^2}}{\frac{a}{T_\alpha^2} + \frac{\theta_1}{Q_\alpha}} \\ h^\varphi &= \frac{\left(\frac{a}{T_\alpha^2} + \frac{\theta_1}{Q_\alpha}\right)^2}{\frac{a}{(T_\alpha^2)^2} + \frac{\theta_2}{Q_\alpha^2}} \end{aligned} \quad (2.17)$$

La alarma saltará para T^2 , Q o φ cuando uno de ellos sobrepase su umbral un número determinado de observaciones consecutivas.

2.5.3. Identificación de fallos

Después de una alarma de fallo, es necesario identificar la variable o variables responsables del fallo. El análisis de contribuciones [Kourti and MacGregor, 1996, Liu et al., 2014] es una de las técnicas más utilizadas para diagnosticar fallos. Se basa en la influencia de cada variable del sistema en el fallo. Las variables con mayor contribución al estadístico fuera de control (que ha detectado el fallo) son las responsables. Adicionalmente, se han desarrollado otros métodos de identificación de fallos como son los basados en el índice de reconstrucción y que se detallarán mas adelante en esta sección.

Contribuciones a T^2 : En la observación \mathbf{x} , para cada una de los componentes principales normalizados de más alto valor se calculan las contribuciones de cada variable x_j en el componente principal normalizado a : $\left(\frac{t_a}{\sigma_a}\right)^2$ [Kourti and MacGregor, 1996], que es:

$$cont_{a,j} = \frac{t_a}{(\sigma_a)^2} (x_j - \mu_j) \quad (2.18)$$

si es negativa, $cont_{a,j}$ se deja a cero. Se calcula la contribución total de cada variable j :

$$CONT_j = \sum_{a=1}^n cont_{a,j} \quad (2.19)$$

Las variables que tengan los mayores valores de contribución serán las que se consideren responsables del fallo.

Contribuciones a Q : Una vez que se ha detectado un fallo con el estadístico Q en la observación \mathbf{x} , se puede calcular la contribución a Q (o error normalizado) de cada una de las variables, x_j [Kourti and MacGregor, 1996], de la forma:

$$cont_{x_j} = (x_j - \hat{x}_j)^2 \quad (2.20)$$

siendo \hat{x}_j el elemento j -ésimo de la aproximación obtenida por el método: $\hat{\mathbf{x}} = (\mathbf{xP})\mathbf{P}^T$.

Al igual que en el caso de la contribución a T^2 , las variables que presenten los valores más elevados de contribución a Q serán identificadas como responsables del fallo.

Contribución a φ : Este método de diagnosis es equivalente a las Contribuciones a T^2 y Q , pero aplicado al estadístico combinado φ [Yue and Qin, 2001]. De acuerdo con lo expuesto en [Alcala and Qin, 2009], para una nueva observación \mathbf{x} la contribución a φ de la variable i es:

$$c_i^\varphi = \left(\boldsymbol{\xi}_i^T \boldsymbol{\Phi}^{\frac{1}{2}} \mathbf{x} \right)^2 \quad (2.21)$$

donde $\boldsymbol{\xi}_i$ es la i -ésima columna de la matriz identidad, y la matriz $\boldsymbol{\Phi}$ es:

$$\boldsymbol{\Phi} = \frac{\tilde{\mathbf{C}}}{T_\alpha^2} + \frac{\mathbf{D}}{Q_\alpha} \quad (2.22)$$

siendo $\tilde{\mathbf{C}} = \tilde{\mathbf{P}}\tilde{\mathbf{P}}^T$, $\mathbf{D} = \mathbf{P}\boldsymbol{\Lambda}^{-1}\mathbf{P}^T$, y $\tilde{\mathbf{P}}$ es la matriz formada por las últimas $(m - a)$ columnas de \mathbf{V} (matriz de vectores singulares del PCA).

Otros métodos de identificación: Aparte de los métodos basados en PCA, existen más trabajos acerca de métodos de identificación que aplican otras técnicas: [Jiang et al., 2015] utilizó el método CVA para la identificación (tal como se indica en la Sección 2.7.3). En este caso, se dispone de dos valores de contribuciones al fallo para cada variable: contribución al espacio de estados y contribución al espacio de residuos. Es un método similar al análisis de contribuciones aplicado al PCA. Por otro lado, [Kariwala et al., 2010] implementó el método de Ramificación y Poda (en inglés, *Branch and Bound* (B&B)) sobre un PCA probabilístico para buscar las variables responsables del fallo a través de la minimización de los estadísticos de control. Este método presenta el problema de que no es capaz de analizar sistemas con un número elevado de variables. Otro método de identificación es el desarrollado por [Mahadevan and Shah, 2009], en el que se aplica el método SVM para realizar una selección de variables que resulte en la máxima separación entre los datos con fallo y los datos sin fallo.

2.6. Análisis de Componentes Principales Dinámico (DPCA)

El método PCA explicado anteriormente construye un modelo que analiza las correlaciones entre variables (correlaciones espaciales) pero no analiza la correlación temporal, es decir, no es dinámico. Para generar un modelo PCA que tenga en cuenta dichas correlaciones temporales se puede implementar el PCA dinámico [Ku et al., 1995, Zhou et al., 2016, Tong et al., 2017, Vanhatalo et al., 2017], que incorpora datos actuales y datos del pasado para crear su modelo. Esto se hace mediante el uso de la matriz aumentada o ampliada, para ello, a partir de la matriz de datos en condiciones normales de operación de la planta, \mathbf{X} , con n observaciones/filas y m variables/columnas, se construye la matriz aumentada, \mathbf{X}_a , que incorpora a \mathbf{X} datos con retardo:

$$\mathbf{X}_a = \begin{bmatrix} \mathbf{X}_{l+1}^T & \mathbf{X}_l^T & \cdots & \mathbf{X}_1^T \\ \mathbf{X}_{l+2}^T & \mathbf{X}_{l+1}^T & \cdots & \mathbf{X}_2^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_n^T & \mathbf{X}_{n-1}^T & \cdots & \mathbf{X}_{n-l}^T \end{bmatrix} \quad (2.23)$$

siendo \mathbf{X}_t el vector de medidas tomadas en el instante t , y l el número de retardos que se quieren incluir. El valor óptimo para l se puede determinar mediante el Criterio de Información de Akaike (AIC) según lo indicado en [Larimore, 1997b, Russell et al., 2000]. A partir de aquí, el método es idéntico al PCA explicado anteriormente, simplemente se sustituye la matriz de datos de partida por la matriz aumentada.

2.7. Análisis de Variables Canónicas (CVA)

Partiendo de una división de los datos en dos subconjuntos, el método de Análisis de Variables Canónicas (CVA) permite realizar una reducción de la dimensionalidad así como maximizar la correlación entre dichos subconjuntos [Russell et al., 2000]. Los datos disponibles de un proceso se dividen en datos de entrada: $\mathbf{x} \in \mathbb{R}^m$, y de salida: $\mathbf{y} \in \mathbb{R}^n$. Cada uno con sus correspondientes matrices de covarianza: Σ_{xx} y Σ_{yy} , además de la matriz de covarianza cruzada: Σ_{xy} . Por otro lado, las matrices $\mathbf{J} \in \mathbb{R}^{m \times m}$ y $\mathbf{L} \in \mathbb{R}^{n \times n}$ cumplen lo siguiente:

$$\begin{aligned} \mathbf{J}\Sigma_{xx}\mathbf{J}^T &= \mathbf{I}_{\bar{m}} \\ \mathbf{L}\Sigma_{yy}\mathbf{L}^T &= \mathbf{I}_{\bar{n}} \end{aligned} \quad (2.24)$$

además de:

$$\mathbf{J}\Sigma_{xy}\mathbf{L}^T = \mathbf{D} = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_r, 0, \dots, 0) \quad (2.25)$$

donde m , n y r son los rangos de Σ_{xx} , Σ_{yy} y Σ_{xy} , respectivamente. Además, γ_i ($i = 1, 2, \dots, r$) representan las correlaciones canónicas, satisfaciendo: $\gamma_1 \geq \dots \geq \gamma_r$. \mathbf{I}_j es una matriz diagonal por bloques, cuyo primer bloque es una matriz identidad de j filas y columnas, y su segundo bloque es una matriz de ceros. Los vectores canónicos son $\mathbf{c} = \mathbf{J}\mathbf{x}$ y $\mathbf{d} = \mathbf{L}\mathbf{y}$, que contienen un conjunto de variables independientes, que cumplen lo siguiente: $\Sigma_{cc} = \mathbf{J}\Sigma_{xx}\mathbf{J}^T = \mathbf{I}_{\bar{m}}$ y $\Sigma_{dd} = \mathbf{L}\Sigma_{yy}\mathbf{L}^T = \mathbf{I}_{\bar{n}}$.

Aplicando una descomposición en valores singulares se obtiene:

$$(\Sigma_{xx})^{-1/2}\Sigma_{xy}(\Sigma_{yy})^{-1/2} = \mathbf{U}\Sigma\mathbf{V}^T \quad (2.26)$$

siendo $\mathbf{D} = \Sigma$. Las matrices de proyección son: $\mathbf{J} = \mathbf{U}^T (\Sigma_{xx})^{-1/2}$ y $\mathbf{L} = \mathbf{V}^T (\Sigma_{yy})^{-1/2}$. Finalmente, este proceso conduce a obtener unas variables canónicas con varianza unidad e incorreladas. También, los vectores \mathbf{c} y \mathbf{d} están correlados por pares. La matriz \mathbf{D} contiene los valores de correlación entre dichos pares.

2.7.1. Modelo de espacio de estados

Larimore implementó el método CVA sobre modelos de espacio de estados [Larimore, 1997a]. Si $\mathbf{y}(t) \in \mathcal{R}^{m_y}$ son las variables de salida de una planta y $\mathbf{u}(t) \in \mathcal{R}^{m_u}$ son las entradas, en el instante t , el modelo se representa así [Russell et al., 2000]:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}_x\mathbf{x}(t) + \mathbf{B}_x\mathbf{u}(t) + \mathbf{e}(t) \\ \mathbf{y}(t) &= \mathbf{A}_y\mathbf{x}(t) + \mathbf{B}_y\mathbf{u}(t) + \mathbf{C}_y\mathbf{e}(t) + \mathbf{g}(t) \end{aligned} \quad (2.27)$$

donde $\mathbf{x}(t) \in \mathcal{R}^d$ es el vector de estados, \mathbf{A}_x , \mathbf{B}_x , \mathbf{A}_y , \mathbf{B}_y y \mathbf{C}_y son las matrices de coeficientes y $\mathbf{e}(t)$ y $\mathbf{g}(t)$ son procesos independientes de ruido blanco.

El vector $\mathbf{p}(t)$ contiene la información de instantes pasados en tiempo t :

$$\begin{aligned} \mathbf{p}(t) &= [\mathbf{y}^T(t-1), \mathbf{y}^T(t-2), \dots, \mathbf{u}^T(t-1), \\ &\quad \mathbf{u}^T(t-2), \dots]^T \end{aligned} \quad (2.28)$$

y el vector $\mathbf{f}(t)$ contiene la información presente y futura sobre la salida de la planta:

$$\mathbf{f}(t) = [\mathbf{y}^T(t), \mathbf{y}^T(t+1), \dots]^T \quad (2.29)$$

Tomando Σ_{pf} como Σ_{xy} , Σ_{pp} como Σ_{xx} y Σ_{ff} como Σ_{yy} , es posible obtener las matrices \mathbf{J} , \mathbf{L} y \mathbf{D} usando la descomposición en valores singulares. El valor asignado a d debe ser mayor o igual que el orden de la mínima

construcción del modelo de espacio de estados de la planta. El vector de estados se obtiene así:

$$\mathbf{x}_s(t) = \mathbf{J}_s \mathbf{p}(t) = \mathbf{U}_s^T (\hat{\boldsymbol{\Sigma}}_{pp})^{-1/2} \mathbf{p}(t) \quad (2.30)$$

donde \mathbf{U}_s está formado por las d primeras columnas de \mathbf{U} . Es habitual seleccionar un número limitado de retardos en \mathbf{p} :

$$\begin{aligned} \mathbf{p}(t) = & [\mathbf{y}^T(t-1), \mathbf{y}^T(t-2), \dots, \mathbf{y}^T(t-l), \\ & \mathbf{u}^T(t-1), \mathbf{u}^T(t-2), \dots, \mathbf{u}^T(t-l)]^T \end{aligned} \quad (2.31)$$

al igual que un número limitado de retardos en \mathbf{f} :

$$\mathbf{f}(t) = [\mathbf{y}^T(t), \mathbf{y}^T(t+1), \dots, \mathbf{y}^T(t+h)]^T \quad (2.32)$$

Los valores de l y h no son triviales, una posibilidad consiste en ajustar modelo autoregresivos con múltiples valores de retardos, eligiendo aquellos valores de l y h que minimicen el criterio de información de *Akaike* (AIC) [Larimore, 1996].

2.7.2. Detección de fallos: Estadísticos de control

Algunos autores han explorado la posibilidad de usar este método para detectar fallos [Negiz and Cinar, 1997, Borsje, 1999, Russell et al., 2000, Jiang et al., 2015, Li et al., 2018]. Se han aplicado dos tipos de estadísticos: T_s^2 , para monitorizar las variaciones en el subespacio canónico, y T_r^2 , para analizar las variaciones en el subespacio de residuos:

$$\begin{aligned} T_s^2 &= \mathbf{x}_s^T(t) \mathbf{x}_s(t) \\ T_r^2 &= \mathbf{x}_r^T(t) \mathbf{x}_r(t) \end{aligned} \quad (2.33)$$

donde $\mathbf{x}_r^T(t) = \mathbf{J}_r \mathbf{p}(t) = \mathbf{U}_r^T (\hat{\boldsymbol{\Sigma}}_{pp})^{-1/2} \mathbf{p}(t)$ y \mathbf{U}_r son las $l(m_u + m_y) - d$ columnas restantes de \mathbf{U} después de extraer \mathbf{U}_s . El estado del proceso se determina comprobando si los estadísticos superan o no sus respectivos umbrales [Russell et al., 2000]. En el caso de que T_s^2 supere su límite, se podrá decir que los estados del sistema están en condiciones de fallo. De forma similar, si T_r^2 supera su umbral, el método indica que el patrón de ruido ha cambiado o que han aparecido nuevos estados en la planta.

Otra opción consiste en usar el vector de residuos del modelo de espacio de estados:

$$r(t) = (\mathbf{I} - \mathbf{J}_s^T \mathbf{J}) \mathbf{p}(t) \quad (2.34)$$

que permite calcular el estadístico Q :

$$Q(t) = r^T r \quad (2.35)$$

Como antes, si Q supera su umbral [Jackson and Mudholkar, 1979], el método detecta un fallo, indicando que el ruido medido ha sufrido cambios o que han aparecido nuevos estados en el proceso.

2.7.3. Identificación de fallos con CVA

Cuando se producen cambios en las variables que las alejan de los valores considerados normales, los estadísticos de control aumentan y llegan a superar sus respectivos umbrales. Es posible identificar qué variables han contribuido a hacer que dichos estadísticos crezcan tanto usando las matrices \mathbf{J}_s y \mathbf{J}_r [Jiang et al., 2015]. En el caso de la contribución de cada variable dentro del espacio de estados, si se tiene un vector de datos $\mathbf{p}(t)$, en el instante t , la contribución se define como:

$$\begin{aligned} c^s(t) &= \mathbf{x}_s^T(t) \mathbf{x}_s(t) = \mathbf{x}_s^T(t) (\mathbf{J}_s \mathbf{p}(t)) \\ &= \mathbf{x}_s^T(t) \sum_{k=1}^{(m_y+m_u)l} (p_k(t) \mathbf{J}_{s,k}^T)^T \\ &= \sum_{k=1}^{(m_y+m_u)l} \mathbf{x}_s^T(t) (p_k(t) \mathbf{J}_{s,k}^T)^T \\ &= \sum_{k=1}^{(m_y+m_u)l} c_{p_k}^s \end{aligned} \quad (2.36)$$

siendo $p_k(t)$ el elemento k -ésimo del vector de datos $\mathbf{p}(t)$, $\mathbf{J}_{s,k}^T$ es la fila k -ésima de \mathbf{J}_s^T y la contribución de $p_k(t)$ en tiempo t en el vector $\mathbf{p}(t)$ es:

$$c_{p_k}^s(t) = \mathbf{x}_s^T(t) (p_k(t) \mathbf{J}_{s,k}^T)^T \quad (2.37)$$

En el método CVA, a diferencia del PCA, existe una correlación temporal en el vector $\mathbf{p}(t)$, es decir, las observaciones actuales están correladas con las pasadas. Partiendo de la ecuación 2.31, $p_k(t)$ se puede escribir como:

$$p_k(t) = \begin{cases} y_i(t-j), y k = (i-1)m_y + j & \text{cuando } k \leq m_y l \\ u_i(t-j), y k = m_y l + (i-1)m_u + j & \text{cuando } k > m_y l \end{cases} \quad (2.38)$$

Finalmente, la contribución de una variable y_m al espacio de estados es el producto de todas sus l observaciones con retardo $y_m(t-j)$, con $j =$

$1, 2, \dots, l$:

$$c_{y_m}^s(t) = \sum_{j=1}^l \mathbf{x}_s^T(t)(y_m(t-j)\mathbf{J}_{s,m_j}^T)^T \quad (2.39)$$

En el caso de la contribución de una variable y_m al espacio de residuos la fórmula es similar a la de la ecuación 2.39:

$$c_{y_m}^r(t) = \sum_{j=1}^l \mathbf{x}_r^T(t)(y_m(t-j)\mathbf{J}_{r,m_j}^T)^T \quad (2.40)$$

Las contribuciones de una variable de entrada u_m en el espacio de estados y el de residuos se obtienen de forma similar.

En el caso de ser detectado un fallo, aquellas variables con mayores valores de contribución serán consideradas como responsables de dicho fallo. Si una variable tiene un alto valor de contribución en el espacio de estados, c^s , se asocia con grandes variaciones en los estados modelados por el CVA. Si el valor de contribución es alto en el espacio de residuos, c^r , dicha variables será responsable de la aparición de nuevos estados no modelados en el CVA.

2.8. Métodos distribuidos de FDI basados en datos

Hasta ahora se ha hablado de los métodos de detección e identificación basados en datos en los que toda la información recogida en la planta se enviaba a una única unidad de procesamiento. Ahora se quieren implementar varias unidades de procesamiento, cada una de las cuales se ocupará de una tarea en el proceso de monitorización, distribuyendo así la detección e identificación de fallos en la planta.

En [Wu et al., 2018] se propone una clasificación para los métodos PCA distribuidos, aunque es extrapolable a otros métodos de monitorización distintos al PCA. Concretamente, esta clasificación propone dividir los métodos en función de cómo se dividen los datos en la red de bloques de la planta, y, también, en función de la estructura de comunicación entre los diferentes bloques. En el primer caso los métodos se pueden dividir entre aquellos en los que cada bloque tiene acceso a un conjunto diferente de datos, y aquellos en los que los bloques acceden a los mismos datos, pero en diferentes condiciones. Esta última propuesta se suele corresponder con bloques de sensores que toman datos de un mismo fenómeno, pero desde diferentes ubicaciones. Si se define una matriz global de datos, en el primer caso los datos globales se agrupan por columnas (DCO), es decir, cada columna o grupo de columnas serán los datos analizados por cada bloque y en el segundo caso, se organizan los datos por filas (DRO).

Además, según la estructura de comunicación de la red, nos podemos encontrar con redes jerárquicas del tipo maestro-esclavo y también redes no jerárquicas. En el primer caso, el sistema está dividido en bloques esclavo que realizan una serie de tareas y un bloque central o maestro que fusiona los resultados de los esclavos. Las estructuras de este tipo coinciden con el tipo DCO visto antes. En el segundo caso todos los nodos o bloques realizan el mismo tipo de tareas y comparten resultados pero no los datos originales. Estas estructuras suelen ser del tipo DRO.

2.8.1. Métodos de FDI con distribución de los modelos

La monitorización descentralizada con distribución de los modelos ha sido aplicada por varios autores [Grbovic et al., 2012, Ge and Song, 2013, Jiang and Yan, 2014, Jiang and Huang, 2016, Tong et al., 2017, Tian et al., 2019] con buenos resultados. El paso más crítico de esta técnica es la división de la planta en bloques de variables en los que se implementarán los distintos modelos y, para ello, se necesita establecer un criterio que indique qué variables deben ser agrupadas y cuáles no. Esto es posible hacerlo aplicando criterios topológicos, esto es, creando bloques que incluyan los datos de sensores que estén próximos entre sí o que pertenezcan a un mismo dispositivo de la planta o a dispositivos entre los que haya algún tipo de conexión o que compartan elementos comunes (tuberías, etc.) [Zhang et al., 2010, Li et al., 2010]. Otra forma de hacer la división consiste en analizar los datos disponibles para extraer información que permita saber qué variables tienen algún tipo de dependencia entre ellas [Grbovic et al., 2012, Ge and Song, 2013, Tong and Shi, 2016, Tong et al., 2017]. Esta última opción permite agrupar aquellas que tengan más relación sin necesidad de disponer de ningún conocimiento previo de la planta.

También, se pueden clasificar los tipos de descentralización en función de cómo se distribuyan las variables en los grupos [Grbovic et al., 2012]:

- **Descentralización completa:** a cada variable se le asigna un grupo, en el que se incluye dicha variable, y, quizá, datos retardados de esa variable. Aquí se pierde la información relativa a la correlación entre variables. En el caso de disponer de una sola variable, sin retardos, se pueden aplicar técnicas estadísticas univariantes como las expuestas en la Sección 2.2.2.
- **Descentralización multi-bloque:** En este caso existen varios bloques que incluyen, cada uno de ellos, distintas variables. Existe la posibilidad de que haya variables presentes en varios bloques simultáneamente.
 - **Un bloque por variable:** Cada variable tiene asignado un bloque que incluye, aparte de dicha variable, todas las que tengan

algún tipo de influencia en ella. También se pueden añadir variables con retardo para incluir información acerca de la correlación temporal.

- **Menos bloques que variables:** Una vez encontradas las correlaciones entre variables, o habiendo realizado un análisis topológico de la planta, se construyen bloques formados por las variables (y, en su caso, variables con retardo) que tengan algún tipo de correlación, que estén próximas, etc. Si algunas variables no están incluidas en ningún bloque, pueden ser descartadas o incluidas en un bloque extra.

La utilización de esta metodología distribuida plantea los siguientes problemas [Ge and Chen, 2016]:

1. ¿Cómo llevar a cabo la división de la planta? Se debe buscar una partición que permita una monitorización eficaz.
2. ¿Cómo combinar los diferentes tipos de datos provenientes de diferentes fuentes?
3. ¿Cómo integrar los datos con diferentes frecuencias de muestreo?
4. Y, finalmente, ¿cómo fusionar los resultados provenientes de los diferentes bloques?

Desde el punto de vista de las plantas industriales, la forma de recoger los datos es la mostrada en el caso DCO, ya que cada nodo o bloque recogerá datos medidos por diferentes sensores. Además, los métodos estudiados son de estructura jerarquizada y se va a disponer de nodos de procesado y de un nodo central que procese los resultados de dichos nodos.

En [Livani and Abadi, 2010] se propone un método basado en PCA distribuido usando factorización QR y agrupamiento de datos en clústers. Aquí los bloques analizan los datos para agruparlos en clústers y son estos clústers lo que se envía al procesador central. La detección de fallos que aplica este método se basa en el análisis de la evolución de los clústers a lo largo del tiempo.

Existen otros métodos que tienen en común el hecho de aplicar los siguientes tres pasos: dividir la planta en bloques, definir un método FDI local en cada bloque y fusionar toda la información de los bloques para dar un resultado global.

El método basado en *Sparse PCA* (SPCA) [Zou et al., 2006, d'Aspremont et al., 2007] es una variante del PCA que se puede aplicar para crear grupos de variables. Mediante el *Sparse PCA* se consiguen obtener unos componentes principales, que son combinación lineal de las variables originales, pero, a diferencia del PCA normal, los valores que multiplican a algunas variables en dicha combinación lineal son cero. De tal forma que se puede usar cada uno de los componentes principales para decidir qué variables formarán

parte de cada uno de los bloques: en un bloque sólo se incluirán aquellas variables con valores no nulos dentro del componente principal asignado a dicho bloque. Este método se utilizó en [Grbovic et al., 2012] para descentralizar una planta. Posteriormente, se desarrolló un método de clasificación de fallos en el que cada bloque fue entrenado para clasificar fallos, pero con la particularidad de que cada uno de los bloques sólo era capaz de detectar un número limitado de fallos. Finalmente, se usó un método basado en entropía para fusionar los resultados procedentes de cada uno de los bloques.

Otro método es el DDPCA que aplica una descentralización usando datos pasados [Tong and Shi, 2016], y crea un bloque para cada variable basándose en la información contenida en la matriz de correlaciones. En este método se obtiene un bloque por variable, incluyéndose únicamente en cada grupo aquellas variables altamente correlacionadas con la variable a la que pertenece el bloque. En cada uno de los bloques se implementa un método local de detección de fallos, cuyos resultados son fusionados con un método bayesiano para llegar a un resultado único global.

El método WDDPCA parte del DDPCA anterior, pero introduce un *pesado* o ponderado de las variables [Tong et al., 2017] a la hora de crear los bloques, de tal forma que sigue habiendo un bloque por variable, pero en cada uno de ellos están incluidas ahora todas las variables. A la hora de trabajar con los métodos locales de detección de fallos, cada variable verá multiplicados sus valores instantáneos por un peso que dependerá del grado de correlación de dicha variable con la variable a la que pertenezca el bloque.

Los métodos que explicaremos a continuación son métodos distribuidos, que siempre usan el método PCA en cada bloque para detectar fallos, y el método de Bayes para fusionar los resultados. La principal diferencia entre unos métodos y otros es la forma de dividir la planta en bloques.

En [Ge and Song, 2013] se realiza la descentralización de la planta creando cada uno de los bloques según las diferentes direcciones de los componentes principales del modelo PCA. Las variables incluidas en cada bloque se deciden analizando los coeficientes correspondientes en la matriz de cargas del modelo PCA. Además, se construye un bloques adicional en el subespacio de residuos. En cada uno de los bloques se trabaja con la parte del modelo PCA que corresponde a las variables incluidas en el mismo, obteniéndose valores locales para los estadísticos T^2 y Q , de modo que un fallo se detecta cuando en alguno o algunos de los bloques estos estadísticos superan sus respectivos umbrales (el usuario debe decidir en cuantos bloques debe haber una anomalía para considerar la existencia de fallo).

Otra posibilidad para distribuir la planta a partir del análisis de las relaciones entre variables se muestra en [Jiang and Yan, 2014], donde se plantea el uso de la *información mutua*. Esta permite cuantificar el nivel de influencia que una variable tiene en otra, por tanto, los grupos de variables se crean uniendo aquellas variables con alto valor de información mutua. Posteriormente, en cada bloque se crea un modelo PCA local que servirá

para detectar los fallos en el correspondiente bloque. Los resultados de los estadísticos de cada bloque se usan para definir una posición en el espacio que, posteriormente, se comprueba si está alejada de la región considerada como de comportamiento normal, y, en ese caso, se detecta un fallo.

También en [Tian et al., 2019] se analizan las relaciones entre variables para configurar los grupos. En este caso se utiliza la función de cópula que tiene en cuenta tanto los niveles de correlación entre las variables como los patrones de correlación, de tal forma que se agrupan las variables cuya función de cópula supere un cierto umbral. Luego se implementa un PCA en cada bloque y se obtiene el resultado de la detección global fusionando los resultados locales usando inferencia bayesiana.

En [Jiang and Huang, 2016, Jiang et al., 2016] se plantea el uso de un algoritmo genético para realizar la división de la planta. Para ello se define una función objetivo que mide el rendimiento del método de detección de fallos para cada una de las configuraciones descentralizadas. De tal forma que el algoritmo genético seleccionará aquella configuración de bloques que proporcione un valor óptimo de rendimiento. Posteriormente, en cada bloque se implementa un modelo PCA y los resultados locales se fusionan mediante inferencia bayesiana.

2.8.2. Métodos de FDI con distribución de los métodos

En el área de la monitorización de procesos hay gran diversidad de métodos para realizar la detección, la identificación y la diagnosis de fallos, pero no se puede afirmar que ninguno sea superior a otros. Es imposible encontrar un método que sea más eficaz que los demás en cualquier tipo de situación, y mucho menos cuando se trabaja en plantas de gran complejidad [Ng and Srinivasan, 2010, Ghosh et al., 2011b, Zhang and Ge, 2015]. Usualmente, se suele trabajar con un sólo método FDI, pero en los últimos años se han presentado propuestas que aplican varios métodos trabajando de forma paralela que procesan toda la información recogida en la planta y fusionarlos, de forma que al combinar un conjunto de métodos se puedan aprovechar las fortalezas de cada uno de ellos y compensar las debilidades [Dasarathy and Sheela, 1979, Hansen and Salamon, 1990, Ghosh et al., 2011a]. Cuanto mayor diversidad de métodos se utilicen mejor será el resultado obtenido, esto se debe a que cada uno de los métodos aplicados trabajará con un modelo o visión de la planta diferente a la del resto de métodos, y, por tanto, cada método presentará errores diferentes a los de los demás. Por tanto, al ensamblar todos los métodos, estos errores individuales se compensen y el error total se reduzca, dando lugar a un resultado final mejor que si sólo se usase un único método FDI [Zhang and Ge, 2015].

Al aplicar la distribución a los métodos FDI es importante contar con una técnica de toma de decisiones que permita fusionar los resultados de cada uno de los métodos de cara a obtener un único resultado. Este aspecto

es de vital importancia ya que no siempre los métodos utilizados van a dar el mismo resultado de diagnóstico [Ghosh et al., 2011a].

En el método desarrollado por [Ghosh et al., 2011a], que se basa en clasificación de fallos, se toman los resultados de seis métodos FDI distintos: análisis de componentes principales (PCA), análisis de componentes independientes (ICA), *kernel* PCA, k vecinos más cercanos (KNN), análisis del discriminante de *Fisher*, y redes neuronales artificiales; en la siguiente etapa se fusionan los resultados de cada método usando un tipo de fusión denominando *Dempster-Shafer* que combina las evidencias proporcionadas por diversas fuentes para obtener una decisión final con un cierto grado de creencia. Finalmente, se aplica un sistema de fusión selectiva que elimina del proceso aquellos clasificadores que presentan un alto grado de similitud con otros.

Otro método de clasificación de fallos con métodos distribuidos es el propuesto por [Zhang and Ge, 2015] se usan varios métodos de detección e identificación de fallos, pero aplicados a distintos niveles de la planta, de modo que se puede aplicar un método para una sección grande de la planta, a la vez que se emplea otro método para monitorizar únicamente un dispositivo contenido en esa sección. Finalmente, los resultados que entregan los métodos son fusionados mediante el método *Dempster-Shafer*, ya que este método permite fusionar los clasificadores obtenidos, al permitir trabajar con clases que no son excluyentes.

En el caso del método propuesto por [Ghosh et al., 2011b] la fusión de los resultados de diversos métodos FDI, como son filtros de Kalman, PCA, Redes neuronales y Mapas auto-organizativos, se realiza usando varias técnicas de fusión: voto ponderado, fusión bayesiana y fusión *Dempster-Shafer*. Finalmente se comparan los resultados obtenidos con estas tres técnicas.

Pero, estos métodos basados en clasificación presentan un problema importante y es que se hace necesaria mucha información previa de los diferentes fallos que se pueden presentar en el sistema y no se identifican correctamente nuevos fallos.

2.8.3. Métodos de FDI con distribución de la computación

En las últimas décadas se han desarrollado diversas formas de ampliar la capacidad de cálculo de los ordenadores. Dos de las más populares son la computación en paralelo y la computación distribuida. La primera de ellas, computación en paralelo, consta de varios procesadores que trabajan de forma independiente pero comparten una única memoria, esto ocurre con los procesadores con varios núcleos; mientras que la segunda alternativa, computación distribuida, cada procesador dispone de su propia memoria y se comunica con los demás mediante mensajes, en esta caso se trata de ordenadores completos unidos a través de una red de comunicación [Garraway, 1999].

Dentro de la segunda categoría existen diferentes arquitecturas como *Hadoop*, *MapReduce*, *Spark* y otros [Ghemawat et al., 2003, Dean and Ghemawat, 2008, Shvachko et al., 2010, Zaharia et al., 2010].

En grandes plantas industriales se puede plantear esta forma de trabajar para evitar la transmisión de grandes cantidades de datos, ya que cada uno de los procesadores de la red se ocupa de hacer los cálculos necesarios con los datos que tiene disponibles y sólo transfiere a la red los resultados. Esto, que ha sido ampliamente utilizado dentro del campo del *Big data* [Fassò and Cameletti, 2009, Zaharia et al., 2010, Chen and Zhang, 2014], es posible aplicarlo tanto a la computación distribuida y en paralelo para calcular un modelo PCA (que puede ser usado posteriormente para detectar fallos) usando diferentes librerías de las arquitecturas antes mencionadas [Elgamal and Hefeeda, 2015], o computación en la nube [Wu et al., 2016], como directamente a la diagnosis de fallos [Nakamura et al., 2017, Zhu et al., 2017]. Así, [Nakamura et al., 2017] emplea la computación distribuida para entrenar muchos modelos de clasificación de fallos, para, finalmente, elegir el más eficaz que es el que usará luego para monitorizar la planta.

El trabajo de [Wu et al., 2016] aplica el método de análisis de componentes principales de forma distribuida mediante técnicas de computación en la nube. Aquí es la arquitectura *Hadoop* la que sirve para realizar el almacenamiento de datos de los bloques, y la arquitectura *Spark* para realizar el cálculo distribuido y paralelo de los modelos PCA (que facilita y agiliza la descomposición de la matriz de covarianza del PCA).

En el caso de [Zhu et al., 2017] se divide la instalación en bloques, agrupando las variables del proceso por su pertenencia a cada uno de los dispositivos de la planta. Después se utiliza la arquitectura *MapReduce* para llevar a cabo la computación en cada uno de los bloques de los modelos PCA. Finalmente, se establece una monitorización jerarquizada en la que se analiza la planta completa, cada uno de los bloques y, finalmente, las variables individuales.

Otra alternativa para hacer la distribución del cálculo sin utilizar estas arquitecturas, es simplemente tener varios nodos de procesamiento y realizar parte de los cálculos en cada uno de esos nodos, como se muestra en [Wu et al., 2018]. Haremos aquí una revisión de algunos de estos métodos del tipo DCO y con estructura jerarquizada, se pueden ampliar la información sobre estos métodos en [Wu et al., 2018].

Estos métodos se utilizan sobre todo en el área de aprendizaje automático y tratan de crear un modelo PCA global en el nodo maestro, con la información procesada y enviada por los nodos esclavos. Luego, este modelo PCA global es el que se usa para resolver el problema final, bien desde el nodo central o bien enviando el PCA global a cada nodo y trabajando de forma distribuida con dicho PCA global en cada nodo.

En [Bai et al., 2005] se propone un método basado en la factorización QR. Para ello, los bloques en los que se ha dividido la planta procesan los datos

que se les han asignado mediante una factorización QR y la mitad de ellos envían sus resultados a la otra mitad. Esta otra mitad que recibe datos uno lo que ha recibido a sus resultados y repite la factorización. Este proceso se repite hasta que sólo queda un bloque que reciba datos. Finalmente en este bloque se hacen una serie de operaciones que dan como resultado un modelo PCA con los resultados transmitidos por los bloques. Los parámetros del PCA se envían luego a cada uno de los nodos para ser usado en los cálculos que realizarán estos posteriormente cuando trabajen en línea tomando datos.

Otro método para realizar la descentralización es el método CPCA, propuesto en [Kargupta et al., 2001]. En esta propuesta cada nodo obtiene un modelo PCA de los datos que le corresponden y envía las matrices resultantes de dicho PCA al nodo central. Éste, a su vez, construye un modelo PCA global a partir de los modelos locales, que será el que se use posteriormente en el sistema para procesar nuevos datos.

Otra posibilidad es la propuesta de [Qu et al., 2002]. El proceso seguido en este método, (en inglés, *Merging PCA*), parte de una división de la planta en bloques, en cada uno de los cuales se genera un modelo PCA que enviará sus datos al procesador central, que procesará todos los resultados recibidos mediante un PCA global y, finalmente, devolverá a los bloques los datos de este modelo para que lo usen en el procesado local de los datos.

2.9. Toma de decisiones multi-criterio

Cuando se aplican varios métodos en un sistema de monitorización surge el problema de fusionar los resultados de cada uno de los métodos, es decir, se necesita hacer una toma de decisiones basada en los distintos métodos para alcanzar una diagnosis final.

La toma de decisiones basada en múltiples criterios, (*Multicriteria Decision Making* (MCDM)), es una técnica muy aplicada en diferentes áreas como pueden ser la inversión financiera [Kou et al., 2014, Pineda et al., 2018], mantenimiento de instalaciones [Dhanisetty et al., 2018], gestión energética [Villacreses et al., 2017], detección e identificación de fallos [Zhang and Ge, 2015], etc., que busca dar una solución única cuando se dispone de diferentes opciones evaluadas usando múltiples criterios [Behzadian et al., 2012, Mardani et al., 2015].

En muchas ocasiones se plantea el problema de tener que elegir entre varias soluciones u opciones para sortear un problema. Además, no suele existir un criterio único para valorar la idoneidad de cada solución. Aquí es donde entra el MCDM, cuyo objetivo es dar soporte al responsable de la toma de decisiones en la elección de la solución. Las técnicas MCDM se basan en los diferentes criterios utilizados para dar una valoración única a cada solución, y facilitar así la elección, ya que, simplemente, se opta por aquella opción con mejor resultado final.

Las etapas de un proceso de toma de decisiones MCDM son [Majumder, 2015]:

1. Identificar el objetivo perseguido con la toma de decisiones.
2. Selección de los criterios para valorar cada solución/opción.
3. Selección de las soluciones/opciones a considerar.
4. Definición de la ponderación a aplicar a cada criterio.
5. Definir el criterio de agregación.
6. Toma de decisiones basada en los resultados de la agregación.

El método MCDM debe definir cómo se hace la ponderación y agregación de los resultados entregados por cada criterio de cada una de las soluciones/opciones consideradas.

2.9.1. Operadores OWA

Un método muy utilizado para realizar tomas de decisiones son los operadores de media ponderada ordenada (*Ordered Weighted Averaging (OWA)*) introducidos por [Yager, 1988]. Son operadores lingüísticos basados en Lógica Difusa que buscan obtener una decisión final y global basada en las puntuaciones que ha obtenido cada opción al aplicar los criterios utilizados, de tal forma que, si una opción obtiene las puntuaciones: $\langle \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n \rangle$, siendo α_1 la puntuación otorgada por el criterio 1, α_2 la del criterio 2, etc. y se ordenan de mayor a menor, se obtiene el vector: $\langle \beta_1, \beta_2, \beta_3, \dots, \beta_n \rangle$, donde β_1 es la mayor puntuación y β_n es la menor, y n es el número de criterios aplicados.

El operador OWA se define con un vector de pesos $\mathbf{w} = \langle \omega_1, \omega_2, \omega_3, \dots, \omega_n \rangle$ que multiplica al vector de puntuaciones ordenadas $\langle \beta_1, \beta_2, \beta_3, \dots, \beta_n \rangle$ de la siguiente forma:

$$F(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n) = \sum_{j=1}^n \omega_j \beta_j \quad (2.41)$$

El valor de cada uno de los elementos del vector de pesos \mathbf{w} determina cómo se va a realizar la agregación. Este vector puede ser obtenido mediante cuantificadores lingüísticos difusos [Yager, 1996, Liu and Han, 2008, Yager, 2020], como son los del tipo regular monótono creciente (RIM), que permiten usar criterios de agregación basados en términos lingüísticos, haciendo más sencillo su manejo.

Un cuantificador Q del tipo RIM debe cumplir que: $Q(0) = 0$, $Q(1) = 1$ y $Q(x) \geq Q(y)$ si $x \geq y$ [Yager, 1996]. La forma de obtener cada elemento j

del vector de pesos \mathbf{w} es mediante la expresión:

$$\omega_j = Q\left(\frac{j}{n}\right) - Q\left(\frac{j-1}{n}\right) \quad (2.42)$$

El cuantificador definido con la expresión: $Q(r) = r^\alpha$ es del tipo RIM, y requiere seleccionar un valor para el parámetro α . Sus pesos se calculan aplicando la expresión 2.42:

$$\omega_j = \left(\frac{1}{n}\right)^\alpha - \left(\frac{j-1}{n}\right)^\alpha \quad (2.43)$$

Un operador OWA puede ser caracterizado mediante su valor de *orness* y/o *andness* [Yager, 1993, Yager, 1996, Fuller and Majlender, 2001], de tal forma que:

$$\begin{aligned} orness(\mathbf{w}) &= \frac{1}{n-1} \sum_{j=1}^n (n-j)\omega_j \\ andness(\mathbf{w}) &= 1 - orness(\mathbf{w}) \end{aligned} \quad (2.44)$$

Los valores de *orness* y *andness* siempre van a estar comprendidos en el rango $[0, 1]$. Un valor elevado de *orness* implica que el operador realizará la toma de decisiones tomando en cuenta sólo las puntuaciones más altas obtenidas y desechando el resto (baja aversión al riesgo). Por contra, un valor bajo de *orness* (es decir, un valor alto de *andness*) implica que el operador OWA va a tener en cuenta la mayoría de las puntuaciones obtenidas (incluso, todas) y por tanto va a ser más exigente, ya que para obtener un resultado final alto, cada opción analizada necesitará ser puntuada con altos valores por todos o casi todos los criterios empleados (alta aversión al riesgo). Esto permite al usuario, en función del riesgo que quiera o que pueda asumir, elegir un operador u otro.

Para el cuantificador RIM $Q(r) = r^\alpha$, citado antes, su valor de *orness* se obtiene con la expresión:

$$orness_{r^\alpha} = \frac{1}{1 + \alpha} \quad (2.45)$$

de modo que un valor alto de α lleva a un valor próximo a cero de *orness*; y un valor de α próximo a cero, a un valor cercano a 1 de *orness*.

2.10. Resumen

En este capítulo se presenta la monitorización de procesos industriales y se justifica su importancia. Dentro de este campo se introduce el concepto

del control estadístico de procesos y se presentan las principales técnicas del control univariable.

Después se explica el proceso de detección e identificación de fallos, con una clasificación de los métodos existentes para llevarlo a cabo. Además se presentan las características que deben tener estos métodos así como la forma de evaluar su efectividad.

Dentro de los métodos existentes para realizar la detección e identificación de fallos basada en datos y con métodos de control estadístico de procesos se exponen dos de los más utilizados como son el análisis de componentes principales y el análisis de variables canónicas. Además se presentan algunos métodos usados por otros autores.

A continuación se presenta un estado del arte de los métodos de detección e identificación de fallos de tipo distribuido, justificando su utilización y mostrando sus ventajas, y se muestra una clasificación de los diferentes métodos existentes. Algunos de estos métodos se explican brevemente.

También se presentan unos conceptos básicos acerca de la toma de decisiones y de un método concreto como son los operadores OWA para realizar dicha toma de decisiones.

Parte II:

**FDI MEDIANTE APROXIMACIONES
DISTRIBUIDAS**

Resumen

En esta segunda parte de la tesis se van a exponer las propuestas de métodos FDI distribuidos, las pruebas que se han llevado a cabo con ellas y los resultados obtenidos.

Primero, se presentan una serie de pruebas que se han llevado a cabo con un método descentralizado sencillo y se comparan sus resultados con propuestas de métodos descentralizados propuestos por otros autores. Esto ha servido para comprobar si este tipo de aproximación distribuida o descentralizada es eficaz en la detección y diagnóstico de fallos, y, por tanto, si era interesante seguir investigando en esta línea de trabajo.

A continuación se presenta una propuesta de esta tesis de método FDI basado en la distribución de modelos, de modo que divide la planta en varias partes y con cada una de ellas se crea un modelo para la detección de fallos. Posteriormente los resultados de cada uno de los modelos son fusionados para obtener un resultado global.

A continuación se presenta la propuesta de esta tesis para hacer una diagnosis distribuida de métodos FDI, de modo que se dispone de varias técnicas que se aplican a la diagnosis de la planta completa. Luego, los resultados de dichas técnicas se fusionan para conseguir un único resultado.

Finalmente, se presenta otra propuesta, esta vez es una aproximación distribuida en la que la distribución se hace en los cálculos realizados para la diagnosis. De tal forma que se dispondrán varios nodos independientes que realizarán operaciones con la información recibida de cada una de las variables medidas en la planta. Como en los casos anteriores, se fusionarán los resultados para conseguir un resultado global.

Palabras claves

Diagnosis de fallos. Métodos distribuidos. Distribución de los modelos. Distribución de los métodos. Distribución de los cálculos.

Capítulo 3

MÉTODOS PCA DISTRIBUIDOS. ESTUDIO PREVIO

3.1. Introducción

Cuando se trabaja en plantas de gran tamaño, con un alto grado de complejidad, disponer únicamente de un procesador central puede no ser factible debido a que este procesador no sea capaz de procesar todos los datos recogidos en la planta. La división de la instalación en bloques independientes, contando cada uno con su propio procesador, puede facilitar la monitorización del proceso [Grbovic et al., 2012, Yin et al., 2014, Ge and Chen, 2016, Jiang and Huang, 2016, Ge, 2017]. De esta forma, cada bloque realiza el procesado y monitorización de la parte de la planta que le corresponda, y envía al procesador central únicamente los resultados de sus cálculos. El procesador central recopila los resultados de las monitorizaciones locales y los fusiona para dar un resultado global. Esta forma de trabajar es la base de los métodos distribuidos o descentralizados.

3.2. Métodos distribuidos de cálculo de PCA

Como un primer acercamiento a los métodos distribuidos se va a comprobar si realmente es efectivo aplicar una descentralización. Por un lado, se implementará un PCA centralizado, que servirá como punto de partida y como base para realizar la comparativa. También se implementará un método PCA descentralizado sencillo siguiendo la aproximación I, es decir, proponer un método PCA local en cada nodo en que se divida la planta, y, por otro lado, se implementarán diversas opciones obtenidas de la literatura para realizar PCA descentralizados [Wu et al., 2018]. En estos métodos no se crea un PCA local, sino que se trata de calcular un PCA global, pero descentralizando su cálculo, es decir, realizando ciertas operaciones en cada nodo, enviando estos cálculos al nodo central donde se utilizarán para gene-

rar un PCA global. Después este PCA global se pasa a cada nodo y es el que se usa para detectar fallos en cada nodo. Por lo tanto, se trata de estudiar si desde el punto de vista de la detección de fallos, es más útil detectar fallos con PCA locales o con PCA globales calculados de forma descentralizada. A continuación, describiremos con más detalle los métodos utilizados en esta comparativa.

3.2.1. PCA distribuido. Cálculo de PCA locales

En este caso se dividirán las variables de la planta en diferentes grupos. La forma de hacer esa división será aplicando criterios topológicos, es decir, las variables se agruparán en función de su proximidad. En cada uno de estos bloques se creará un modelo PCA con los datos correspondientes a las variables de dicho bloque.

Ahora, en cada bloque se analizarán los datos medidos en la planta en tiempo real con el correspondiente modelo PCA y, una vez que un bloque detecte un fallo, hará la identificación usando la contribución a Q . Después enviará el instante de detección del fallo y los valores de contribución de cada variable al nodo central.

En cuanto al procesador central, si en un instante ningún bloque ha enviado información acerca de la detección de un fallo, se espera a la siguiente toma de datos, pero si se recibe información referente a la detección de un fallo, se comprobará en cuál de los bloques ha detectado antes el fallo, quedándose con la información de dicho bloque y descartando la de los demás. El instante de detección del bloque seleccionado se convertirá en el instante de detección global, y la variable de este bloque con el mayor valor de contribución a Q será la responsable del fallo, también a nivel global.

Por tanto, a nivel global, el fallo se detectará en aquel bloque que informe antes de un fallo y las variables responsables del fallo serán aquellas que tengan mayor contribución al estadístico Q dentro del bloque en el que antes se detecte el fallo. Es de esperar que en otros bloques se acabe detectando el fallo si éste se extiende por la planta, pero se hará unos instantes después. En caso de que se detecte el fallo en el mismo instante en varios bloques, se tomarán los resultados de contribuciones a Q de estos bloques y se seleccionará la variable que tenga el valor más alto.

3.2.2. Método PCA distribuido con factorización QR

El primer paso de este método [Bai et al., 2005] consiste en dividir la planta en s bloques con el mismo número de variables cada uno: m , de forma que en cada bloque i se disponga de una matriz de observaciones \mathbf{X}_i , de dimensiones $n_i \times m$, siendo n_i el número de observaciones del bloque i .

Se dispone de s bloques, y se elegirá dicho valor de modo que sea una potencia de 2, es decir: $s = 2^l$. En caso de no ser así, se crearán matrices

vacías, \mathbf{X}_i para unos bloques ficticios que completarán los huecos entre el valor real de s y s_{++} , siendo $s_{++} = 2^l$.

Ahora se seguirán los siguientes pasos:

Paso 1: Se normalizan los datos de cada bloque al rango $[0, 1]$ y, luego, a media 0. En todos los bloques, se normalizan los datos a media 0, restando a cada elemento de \mathbf{X}_i la media de su columna. (Este paso difiere del método de [Bai et al., 2005], pero es necesario para calcular T^2 y Q). La matriz normalizada $\bar{\mathbf{X}}_i$ es descompuesta según la factorización QR:

$$\bar{\mathbf{X}}_i = \mathbf{Q}_i^{(0)} \mathbf{R}_i^{(0)} \quad (3.1)$$

siendo $\mathbf{R}_i^{(0)}$ una matriz triangular superior de dimensiones $n \times n$. Después, se envían los datos: n_i y $\bar{\mathbf{x}}_i^T$ (vector de medias por columnas) al bloque $s = 0$, y, los bloques i , con $i \geq \frac{s}{2}$, envían $\mathbf{R}_i^{(0)}$ al bloque $(i - \frac{s}{2})$. No se envía $\mathbf{Q}_i^{(0)}$.

Paso 2: El bloque i , con $0 \leq i < \frac{s}{2}$, realiza la descomposición:

$$\begin{pmatrix} \mathbf{R}_i^{(0)} \\ \mathbf{R}_{i+s/2}^{(0)} \end{pmatrix} = \mathbf{Q}_i^{(1)} \mathbf{R}_i^{(1)} \quad (3.2)$$

siendo $\mathbf{R}_i^{(1)}$ de dimensión $m \times m$. Después, el bloque i envía la matriz $\mathbf{R}_i^{(1)}$ al bloque $i - \frac{s}{4}$.

Paso 3: Se ejecuta este bucle de descomposiciones QR y envíos de matrices hasta que el bloque 0 reciba la matriz $\mathbf{R}_1^{(l-1)}$, después de $l = \log_2 s$ etapas.

Finalmente, el bloque 0 descompondrá la matriz siguiente:

$$\begin{pmatrix} \mathbf{R}_0^{(l-1)} \\ \mathbf{R}_1^{(l-1)} \end{pmatrix} = \mathbf{Q}_0^{(l)} \mathbf{R}_0^{(l)} \quad (3.3)$$

siendo $\mathbf{R}_0^{(l)}$ una matriz de dimensiones $m \times m$. A continuación se construye la siguiente matriz y se calcula su descomposición QR:

$$\begin{pmatrix} \sqrt{n_0}(\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}) \\ \sqrt{n_1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}) \\ \vdots \\ \mathbf{R}_0^{(l)} \end{pmatrix} = \mathbf{QR} \quad (3.4)$$

siendo:

$$\bar{\mathbf{x}} \equiv \frac{1}{n} \sum_{i=0}^{s-1} n_i \bar{\mathbf{x}}_i \quad (3.5)$$

Y se obtiene una matriz \mathbf{R} ($m \times m$), sobre la que se genera un PCA, haciendo la descomposición en valores singulares de su matriz de covarianza \mathbf{S} . Después, se envía la información del PCA a cada bloque.

Para aplicarlo a detección de fallos, se calculan los estadísticos T^2 y Q locales, con sus umbrales. Las nuevas medidas serán normalizadas y procesadas en cada bloque, siguiendo con el proceso de diagnóstico de fallos de un PCA normal.

3.2.3. Método CPCA

Según se especifica en [Kargupta et al., 2001], se divide el sistema en s bloques, en cada uno de los cuales se toman datos del sistema funcionando en condiciones normales, obteniendo en cada bloque i , una matriz de datos: \mathbf{X}_i . Estas matrices tendrán el mismo número de observaciones, n , pero cada una de ellas contará con un número de variables: m_i , específico de cada bloque. En cada uno de los bloques, se hace un modelo PCA con las matrices de datos correspondientes, y se calculan los componentes principales locales, \mathbf{T}_i usando la matriz de cargas \mathbf{P}_i :

$$\mathbf{T}_i = \mathbf{X}_i \mathbf{P}_i^T \quad (3.6)$$

Después, cada bloque i envía su correspondiente matriz T_i y su matriz de cargas \mathbf{P}_i al procesador central. Esto evita enviar las matrices de datos completas \mathbf{X}_i , que son de mayor dimensión que T_i . El procesador central construye una matriz global de componentes principales, \mathbf{T} :

$$\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_s] \quad (3.7)$$

Además de la matriz diagonal por bloques que incluye las matrices de cargas \mathbf{P}_i :

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \mathbf{P}_2 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & \mathbf{P}_s \end{bmatrix} \quad (3.8)$$

y se debe cumplir que:

$$\mathbf{T} = \mathbf{X} \mathbf{P} \quad (3.9)$$

donde \mathbf{X} es la matriz que agrupa las matrices de datos \mathbf{x}_i de cada uno de los bloques. La matriz global de observaciones se aproxima de la siguiente forma:

$$\hat{\mathbf{X}} = \mathbf{T} \mathbf{P}^T \quad (3.10)$$

Sobre esta matriz $\hat{\mathbf{X}}$ se crea un modelo PCA, y se calcularán sus componentes principales, matriz de cargas, etc. Este modelo PCA global se utiliza para calcular los umbrales globales para los estadísticos T^2 y Q del mismo modo que con un PCA usual. Finalmente, cada nueva observación se procesa en los bloques y en el nodo central, generando el vector de datos estimado: $\hat{\mathbf{x}}$, y se obtienen los valores de los estadísticos para realizar la detección. Para la identificación se procede con la contribución a Q como en el caso anterior.

3.2.4. Método PCA distribuido fusionado

En este método [Qu et al., 2002] el sistema se divide en s bloques, que cuentan cada uno con el mismo número de variables: m . En cada bloque i se crea una matriz de datos \mathbf{X}_i , de dimensiones $n_i \times m$, que es normalizada al rango $[0, 1]$ y a media 0. En cada bloque se crea un modelo PCA, tomando un cierto número de componentes principales en cada uno de ellos, definido previamente por el usuario.

A continuación, cada bloque i enviará al nodo central los siguientes datos de sus respectivos modelos PCA: número de observaciones del bloque, n_i ; número de componentes principales seleccionados, k_i ; media de las columnas de \mathbf{X}_i , $\bar{\mathbf{x}}_i$; matriz diagonal con los cuadrados de los autovalores más altos en la diagonal ($\boldsymbol{\lambda}_i$), $\tilde{\boldsymbol{\Lambda}}_i$; y matriz de cargas, $\tilde{\mathbf{P}}_i$.

En cada modelo PCA local, cuando se efectúa la descomposición en valores singulares de la matriz de covarianza $n_i \mathbf{S}_i$ del bloque, y se obtiene:

$$n_i \mathbf{S}_i = (\mathbf{X}_i^{norm})^T \mathbf{X}_i^{norm} = \mathbf{P}_i \boldsymbol{\Lambda}_i \mathbf{P}_i^T \quad (3.11)$$

Pero, este paso se va a hacer de la siguiente forma:

$$\mathbf{X}_i^{norm} = \mathbf{V}_i \boldsymbol{\Lambda}_i \mathbf{P}_i^T \quad (3.12)$$

Siendo: $\boldsymbol{\Lambda}_i = \boldsymbol{\lambda}_i^T \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_i^2$ y siendo \mathbf{X}_i^{norm} la matriz de datos normalizada del bloque i . Se calculan y envían al procesador central las matrices reducidas: $\tilde{\boldsymbol{\Lambda}}_i$ y $\tilde{\mathbf{P}}_i$, obtenidas a partir de los componentes principales.

En dicho procesador central se genera la aproximación a la matriz global de covarianza de la siguiente forma:

$$\tilde{\mathbf{S}} = \sum_{i=1}^s \tilde{\mathbf{P}}_i \tilde{\boldsymbol{\lambda}}_i^2 \tilde{\mathbf{P}}_i^T + \sum_{i=1}^s n_i (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T \quad (3.13)$$

Y se hace la siguiente descomposición en valores singulares:

$$\tilde{\mathbf{S}} = \frac{1}{n} \mathbf{U} \mathbf{D} \mathbf{U}^T \quad (3.14)$$

La matriz $\tilde{\mathbf{S}}$ servirá para crear un modelo PCA global, en el que se obtendrán las matrices: $\tilde{\boldsymbol{\Lambda}}$ y $\tilde{\mathbf{P}}$. El procesador central enviará a cada bloque:

la matriz $\tilde{\mathbf{P}}$, el vector de valores medios de cada columna a nivel global: $\bar{\mathbf{x}}$ y los valores de los umbrales de T^2 y Q , que habrá que calcular también.

Para la detección de fallos, cada nueva observación para un bloque i : \mathbf{z}_i , se proyectará en el espacio de dimensión reducida global de la siguiente forma:

$$\tilde{\mathbf{z}}_i = \mathbf{z}_i^{norm} \tilde{\mathbf{P}} \quad (3.15)$$

donde \mathbf{z}_i^{norm} representa el vector de observaciones \mathbf{z}_i normalizado a media 0 con la media local. Finalmente se calculan los valores de T^2 y Q para dicha observación y se comprueba si se superan los correspondientes umbrales.

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

Este método [Livani and Abadi, 2010] incluye ya el procedimiento para aplicarlo a la detección de fallos, y se entrena de la siguiente forma: se parte de los $(s - 1)$ bloques en los que se ha dividido el sistema y en los que, inicialmente, se toman datos en condiciones normales de funcionamiento y se incluyen en las matrices \mathbf{X}_i de cada uno de los bloques ($0 \leq i \leq (s - 1)$, todas ellas con n_i filas y m columnas). Por tanto, todos los bloques deben tener el mismo número de variables.

Después, todas las matrices de datos se normalizan por columnas al rango $[0, 1]$, y cada uno de los $(s - 1)$ bloques aplica las sucesivas descomposiciones y factorizaciones QR descritas en la Sección 3.2.2. Finalmente se llega a una matriz \mathbf{R} a partir de la cual se obtiene la matriz de covarianza:

$$\mathbf{S} = \frac{1}{n} \mathbf{R}^T \mathbf{R}, \quad n = \sum_{i=0}^{s-1} n_i \quad (3.16)$$

Se crea un modelo PCA usando esta matriz de covarianza, y, una vez se tienen los componentes principales, se selecciona aquel que lleva asociado el autovalor más alto: φ .

Se establece ahora el perfil normal de funcionamiento agrupando, en cada bloque, los datos sin fallo en clústers. Cada uno de estos clústers ocupará una región del espacio, y estará definido por el centroide y el radio de dicho clúster.

Finalmente, cada bloque enviará al nodo central la información de sus clústers: centroides y radios. El nodo central establece el perfil de funcionamiento normal obteniendo la distancia entre los clústers y el primer componente principal calculado antes, y tomando la máxima:

$$\mathbf{d}_{max} = \max\{\mathbf{d}(C_i^j, \varphi)\} \quad (3.17)$$

donde $d(C_i^j, \varphi)$ es la distancia del clúster j perteneciente al bloque i a ese componente φ . Si c_i^j y r_i^j , representan el centroide y el radio, respectivamente, del citado clúster, la distancia a φ es:

$$\mathbf{d}(C_i^j, \varphi) = \mathbf{d}_p(c_i^j, \varphi) + r_i^j \quad (3.18)$$

siendo :

$$\mathbf{d}_p(c_i^j, \varphi) = \text{sqr}t(\left(\|c_i^j - \bar{\mathbf{x}}\|\right)^2 - (\varphi^T(c_i^j - \bar{\mathbf{x}}))^2) \quad (3.19)$$

y $\bar{\mathbf{x}}$ es el vector de medias global.

A continuación se puede empezar el proceso de detección de fallos, cuyo umbral será la distancia mayor de entre las calculadas antes entre los clústers y el primer componente principal. Cuando entren nuevos datos, se agruparán en clústers, como se ha hecho con los datos sin fallo, se enviarán al procesador central que comprobará si alguno de los nuevos clústers está situado a una distancia del primer componente principal mayor que la que se estableció como umbral. En caso afirmativo se entenderá que el sistema ya no trabaja en condiciones normales y existe un fallo. O dicho de otra forma:

$$\left\{ \begin{array}{l} \mathbf{d}(C_i^j, \varphi) > \mathbf{d}_{max} \implies \text{Datos anómalos} \\ \mathbf{d}(C_i^j, \varphi) \leq \mathbf{d}_{max} \implies \text{Datos normales} \end{array} \right\} \quad (3.20)$$

De cara a evitar falsas alarmas, se puede establecer un cierto número de clústers anómalos, por encima del cuál se avise de un fallo en el sistema.

Para realizar la identificación de fallos, se propone tomar aquellos bloques en los que se han encontrado clústers a una distancia mayor de la establecida en el perfil de funcionamiento normal y, de estos, se eligen los que hayan detectado antes el fallo. En estos bloques se procede a calcular la contribución de cada variable al estadístico Q en el instante del fallo, identificando como responsables a aquellas variables con valores más altos de contribución.

3.3. Comparativa entre método centralizado y métodos distribuidos

Se presentan ahora los resultados de aplicar los métodos expuestos anteriormente en dos plantas: una red de distribución de agua potable, y una depuradora de aguas residuales. La forma de evaluar el desempeño de las propuestas es comparar la cantidad de fallos detectados, la cantidad de fallos identificados correctamente, el retraso en la detección del fallo desde que este aparece y los índices OTI y TI que se explican a continuación.

El porcentaje de observaciones anómalas en ausencia de fallo se mide con el índice OTI (*Overall Type I risk*):

$$OTI = \frac{n_f}{I_{NOC}} * 100 \quad (3.21)$$

siendo n_f el número de observaciones detectadas como anómalas cuando no existe fallo, y I_{NOC} el número total de observaciones en ausencia de un fallo. De cara a evitar la aparición de falsas alarmas, se propone el requisito de que solamente se detecte un fallo cuando hayan sucedido un cierto número de observaciones anómalas. Esa cantidad de observaciones se debe ajustar de modo que, en ausencia de fallo, no se detecte ninguna anomalía en el sistema. La presencia de falsas alarmas aún a pesar de establecer ese número de observaciones anómalas consecutivas se mide mediante el índice TI (en inglés, *Type I risk*):

$$TI = \frac{n_{fa}}{n_s} \quad (3.22)$$

donde n_{fa} es el número de simulaciones en las que hay falsas alarmas en los instantes previos a la aparición del fallo y n_s es el número de simulaciones.

3.3.1. Caso de estudio I: Red de distribución de agua potable

Los métodos propuestos antes se van a poner a prueba en una red de distribución de agua potable, descrita en el Anexo A.1.

3.3.1.1. Metodología experimental

Se trabajará con datos de simulaciones cuya duración será de 200 instantes, apareciendo el fallo en el instante 150 y se descartarán las 100 primeras observaciones ya que se consideran parte de la estabilización inicial de la planta.

A la hora de aplicar los métodos, después de probar con diferentes valores y seleccionar los que dan mejor resultado, se han establecido las siguientes condiciones: todos los PCA seleccionarán un número de componentes principales tal que estos retenga un 70% de la varianza de los datos y un fallo será detectado cuando hayan sucedido 5 observaciones anómalas consecutivas. Para el método de factorización QR con clustering, el radio de los clústers se ha establecido en 0,1 después de probar con varios valores y ser éste el que dio mejor resultado.

En cuanto a la descentralización de la planta, el PCA global se ha aplicado creando un modelo PCA de toda la planta, mientras que los métodos distribuidos se han aplicado sobre la red dividida en 8 bloques (ver Figura 3.1). Cada bloque esta formado por 9 nodos, desde los que se extrae agua a la red para su consumo, y 9 tuberías. Por tanto todos los bloques tienen el mismo número de variables, esto se hace así para poder aplicar los métodos

de detección e identificación que requieren bloques del mismo tamaño. El criterio para asignar cada elemento a cada bloque ha sido topológico, de tal forma que se han agrupado los nodos y tuberías pertenecientes a la misma rama de la red.

Para probar la capacidad de detección e identificación de los métodos se dispone de datos sin fallo y de datos con 3 tipos de fallo. Se han hecho simulaciones para cada tipo de fallo aplicándolos con 2 intensidades de fallo diferentes, dando lugar a dos simulaciones por cada tipo de fallo. Por tanto, se dispone de 6 simulaciones diferentes con fallo.

3.3.1.2. Análisis de resultados

Como resumen de las pruebas realizadas se muestran las Tablas 3.1 y 3.2. Los métodos probados han sido el PCA centralizado, PCA distribuido, Método de componentes principales colectivos (*PCA Colectivo*), PCA distribuido fusionado (*Merging PCA*), PCA distribuido con factorización QR (*PCA (QR)*), y PCA distribuido usando factorización QR y agrupamiento de datos en clústers (*PCA & Clustering*).

Tabla 3.1: 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 Distribuido	5	6	3	6	1.62	4
PCA Colectivo	1	1	0	0	2.67	0.44
<i>Merging PCA</i>	2	6	2	4	5	2.1
PCA (QR)	4	4	3	3	5	1.875
PCA & <i>Clustering</i>	5		4		0	

Tabla 3.2: Retraso en la detección (en instantes) y TI

Método	Retraso medio		TI (% de pruebas)	
	T^2	Q	T^2	Q
PCA Centralizado	-	1.5	0% (0/6)	0% (0/6)
PCA Distribuido	5.67	1.33	0% (0/6)	0% (0/6)
PCA Colectivo	34	10	0% (0/6)	0% (0/6)
<i>Merging PCA</i>	10.5	5.67	0% (0/6)	0% (0/6)
PCA (QR)	6.75	4.5	0% (0/6)	0% (0/6)
PCA & <i>Clustering</i>	5.4		0% (0/6)	

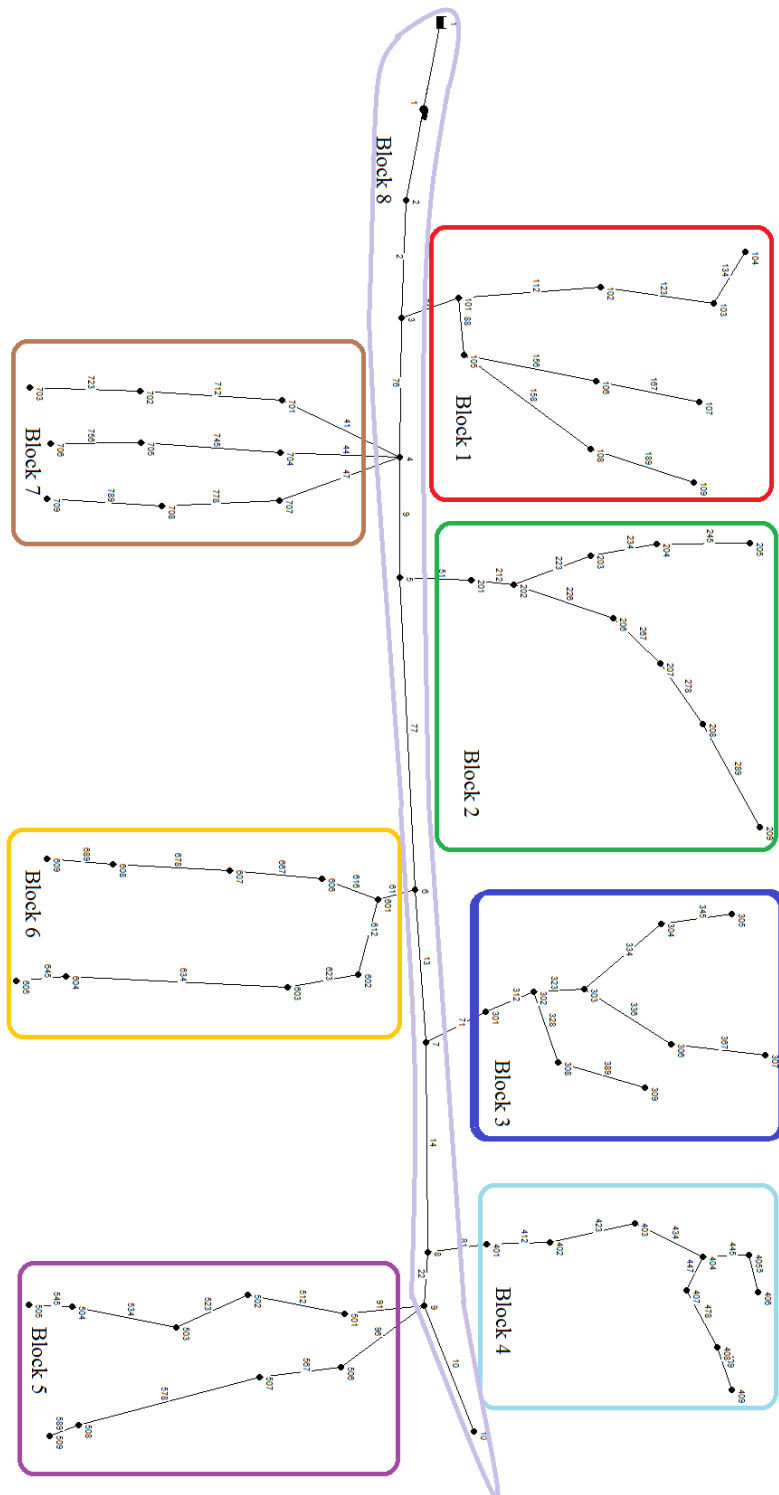


Figura 3.1: Red de agua por bloques

A la vista de los resultados, se observa que los métodos PCA centralizado y PCA distribuido son capaces de detectar todos los fallos con el estadístico Q y además con un retardo bajo. Además, el PCA distribuido ha identificado correctamente todos los fallos con este mismo estadístico. Con un resultado inferior se encuentra el método de factorización QR, ya que detecta 4 de los 6 fallos, y solo identifica bien a 3 de ellos. Además no es tan rápido para detectar los fallos.

El método CPCA solamente es capaz de detectar 1 fallo, y ni siquiera es capaz de hacer la identificación correctamente. Otro método probado es el *Merging* PCA que ha detectado todos los fallos con el estadístico Q e identifica bien 4 de los 6 fallos. Por contra, sus tiempos de detección son los segundos peores. Con T^2 sólo detecta e identifica 2 fallos.

Finalmente, el método con factorización QR y clustering ha podido detectar 5 de los 6 fallos, e identificar correctamente a 4, aunque no consigue unos tiempos de detección muy buenos.

Como resumen se puede afirmar que el método PCA distribuido ha sido el más rápido de todos en detectar los fallos, siendo más sencilla su implantación y procesado que el de los otros métodos. Además es el único que ha sido capaz de identificar correctamente las variables responsables de los fallos en todos los casos.

Se van a mostrar ahora los resultados de un fallo concreto para que se vea el resultado obtenido.

Este fallo consiste en la reducción en un 10% de la velocidad de la bomba que aparece en el instante 150 (no se han analizado las primeras 100 observaciones). Al estar localizado en la bomba, situada en la tubería 1, el fallo debería detectarse en esa tubería o en la tubería número 2, o en el nodo 2, que están justo a continuación de dicha bomba.

Método PCA centralizado: Se dispone de 648 variables y se ha generado un modelo PCA con 3 componentes principales que retienen un 70% de la información contenida en los datos originales. En la gráfica 3.2 y en la Tabla 3.3 se muestran los resultados obtenidos.

Tabla 3.3: Resultados. Método PCA centralizado

Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Contribución T^2	Contribución Q
-	150	-	Contaminante nodo 307	-	1,899

El método no es capaz de detectar el fallo con el estadístico T^2 , mientras que con Q sí que lo consigue. La detección se produce en el instante 150 y la identificación da como resultado que la variable responsable es la cantidad de contaminante en el nodo 307. Esta identificación es incorrecta ya que el fallo es de la bomba no de la concentración de contaminante.

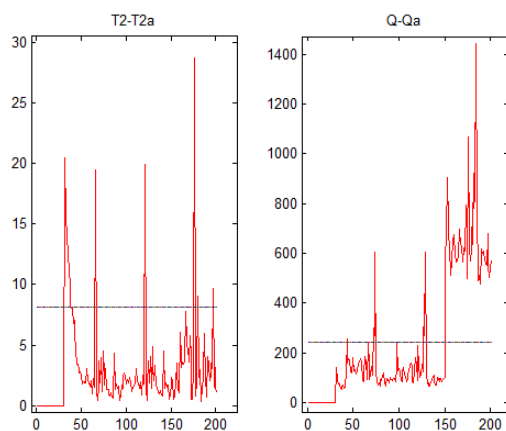


Figura 3.2: PCA centralizado. Fallo bomba 10 %. Simulación con fallo

Método PCA distribuido: En este caso se dispone de 8 bloques en los que se realiza la detección e identificación del fallo. Los resultados obtenidos se muestran en las gráficas 3.3 y 3.4 y en la Tabla 3.4.

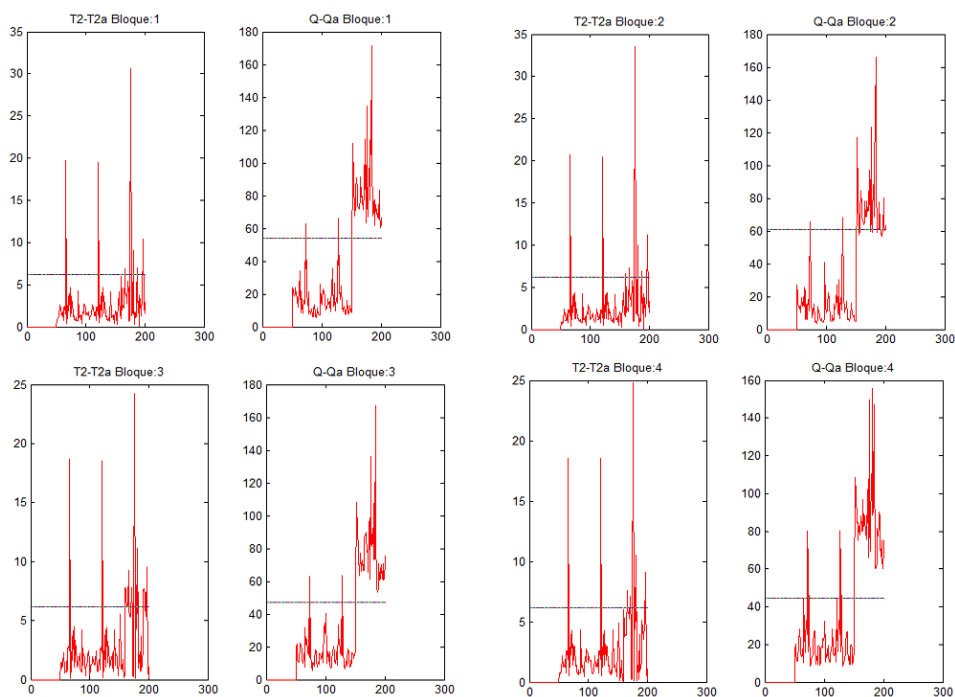


Figura 3.3: PCA distribuido. Fallo bomba 10 %. Bloques 1-4

En ningún caso el estadístico T^2 llega a superar claramente el umbral

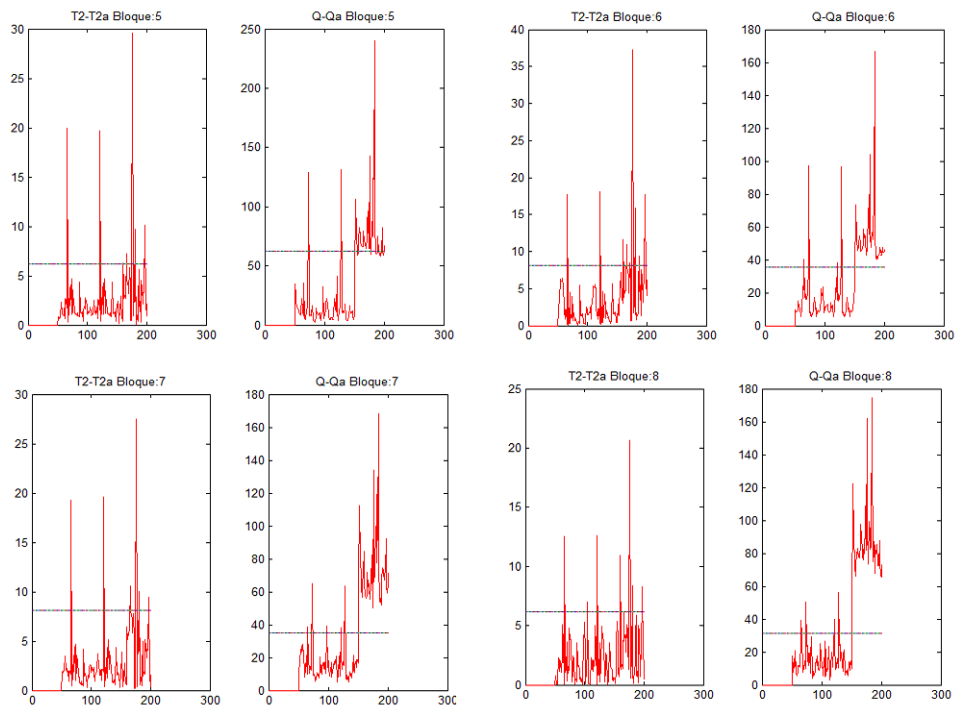


Figura 3.4: PCA distribuido. Fallo bomba 10%. Bloques 5-8

Tabla 3.4: Resultados. Método PCA distribuido

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	155	-	Altura nodo 101	-	2,2499
2	-	162	-	Fricción tubería 212	-	1,7248
3	-	155	-	Altura nodo 301	-	2,2253
4	-	155	-	Presión nodo 401	-	2,24271
5	-	162	-	Altura nodo 506	-	1,6500
6	-	155	-	Altura nodo 601	-	2,2428
7	-	155	-	Altura nodo 704	-	2,2475
8	-	155	-	Presión nodo 2	-	2,2510

aunque, en algunos bloques, si se ve que al final de la simulación dicho índice cambia ligeramente su tendencia. Por el contrario, Q sí supera su umbral en varios bloques. Concretamente, en la gráfica del bloque 8, que es el que contiene la bomba averiada, es dónde más claramente se observa el cambio de Q . En los bloques 1, 3, 4, 6, 7 y 8 es donde se detecta primero el fallo, 5 instantes después de aparecer éste, pero la variable seleccionada como responsable en el bloque 8 es la que tiene el valor de contribución a Q mas alto. Por tanto esa será la variable responsable de todo el sistema. Esta variable es la presión del agua en el nodo 2 que es el que está a continuación

CAPÍTULO 3. MÉTODOS PCA DISTRIBUIDOS

de la bomba. Por tanto, se considera correcta esta identificación.

Método PCA distribuido con factorización QR: En este caso, al ser también un método distribuido, se obtienen resultados de los 8 bloques. En las gráficas 3.5 y 3.6 se muestran los resultados de las 100 observaciones analizadas (se han eliminado las 100 primeras), apareciendo el fallo en la 50. en la Tabla 3.5 se han recogido los tiempos de detección, variables encontradas responsables y los valores de contribución en cada bloque.

Tabla 3.5: Resultados.PCA con factorización QR

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	-	-	-	-	-
2	-	-	-	-	-	-
3	-	-	-	-	-	-
4	-	164	-	Altura nodo 401	-	0,5484
5	-	-	-	-	-	-
6	167	-	Velocidad tubería 634	-	0,0071	-
7	-	-	-	-	-	-
8	-	-	-	-	-	-

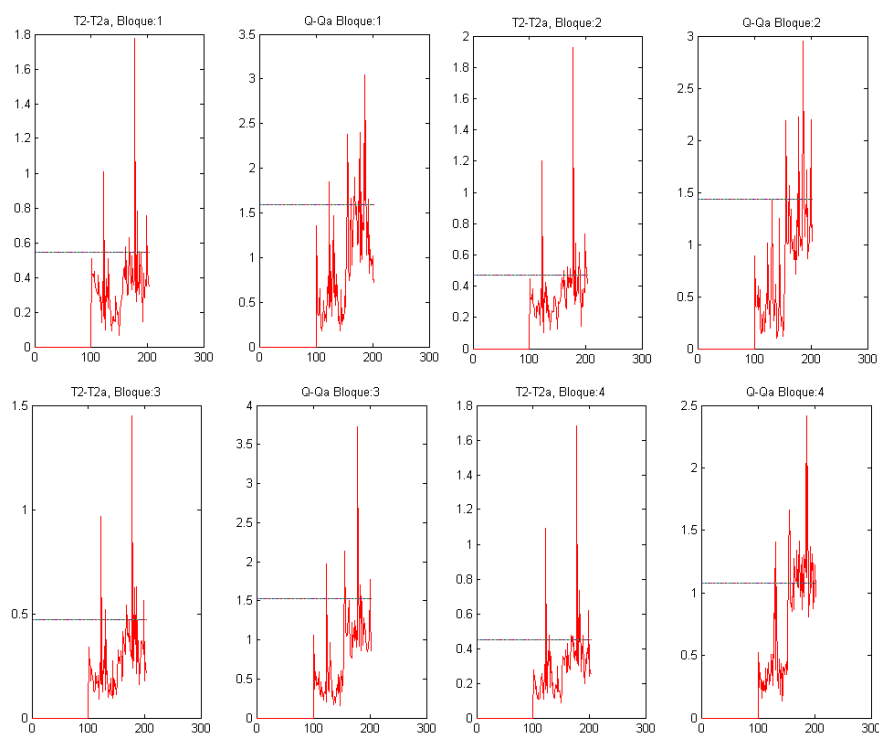


Figura 3.5: PCA con factorización QR. Fallo bomba 10%. Bloques 1-4.

Se observa que el método no trabaja de forma satisfactoria ya que no

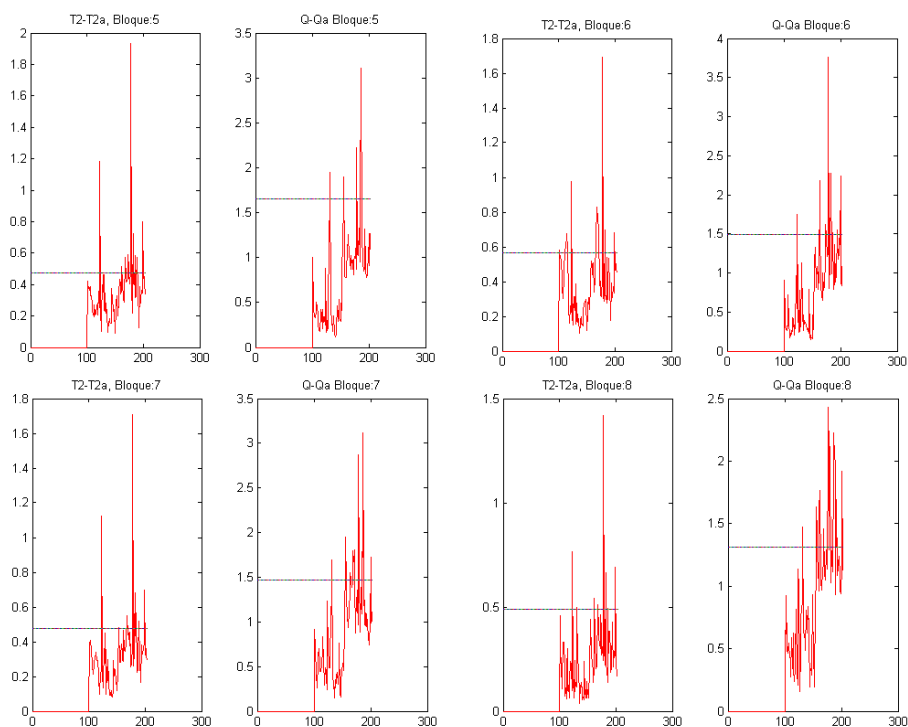


Figura 3.6: PCA con factorización QR. Fallo bomba 10%. Bloques 5-8.

detecta el fallo en el bloque en el que aparece éste (bloque 8). De igual forma que en el método anterior, es el estadístico Q el que se ve más afectado una vez que se ha producido el fallo. El bloque que antes detecta el fallo es el número 4, usando el índice Q , y lo hace 14 instantes después de aparecer dicho fallo, identificando a la variable que corresponde a la altura piezométrica del nodo 401 como responsable, que no tiene nada que ver con el origen del fallo. En cuanto al índice T^2 , el fallo se detecta, en el bloque 6, 17 instantes después de aparecer y no se identifica correctamente a la variable responsable. Por tanto, este método no da una buena identificación.

Método CPCA: Con este método, aunque la planta esté dividida en bloques, el resultado obtenido es una gráfica para todo el sistema (Figura 3.7).

En este caso no se obtiene ningún resultado ya que el método no es capaz de detectar el fallo ni es capaz, por tanto, de hacer una identificación.

Método PCA distribuido fusionado: En este método se vuelven a tener los resultados locales de cada uno de los 8 bloques. Estos resultados

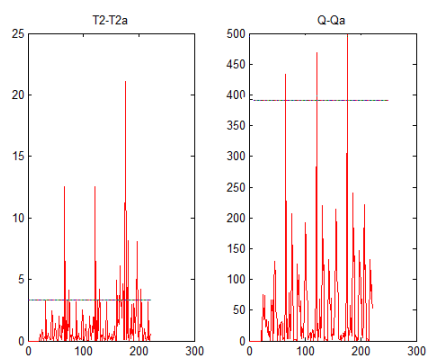


Figura 3.7: CPCA. Fallo bomba 10 %.

se muestran en las Figuras 3.8 y 3.9, y en la Tabla 3.6.

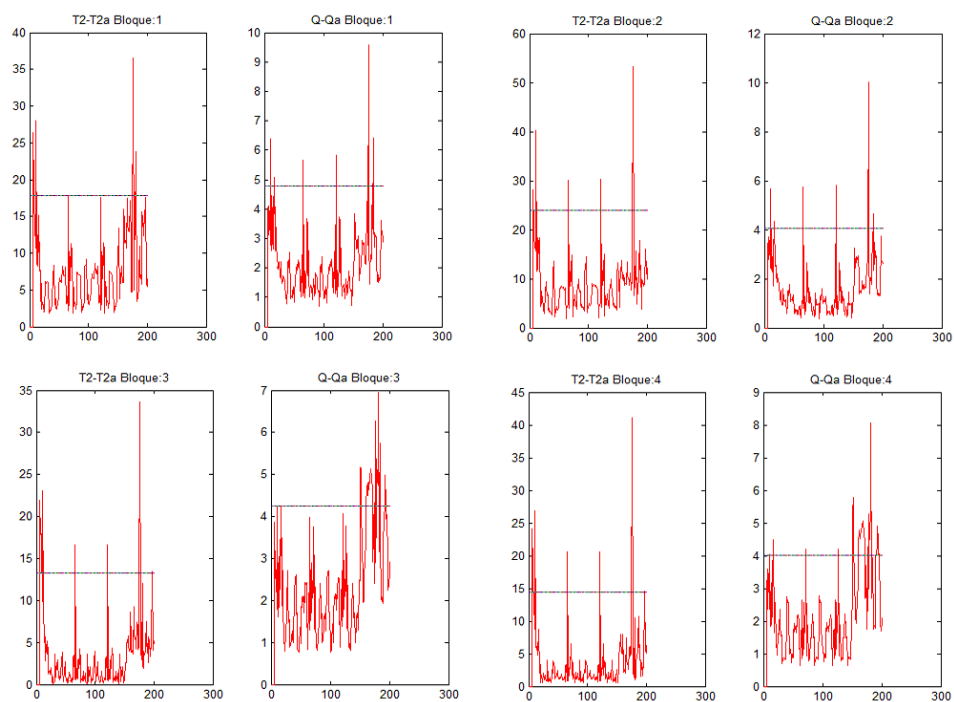


Figura 3.8: *Merging* PCA. Fallo bomba 10 %. Bloques 1-4

Analizando las gráficas, se observa que los dos estadísticos sufren cambios a partir del instante de aparición del fallo. En el caso de T^2 , el fallo es detectado en el bloque 6, justo 3 instantes después de aparecer, pero no identifica correctamente la variable responsable. El estadístico Q , en el blo-

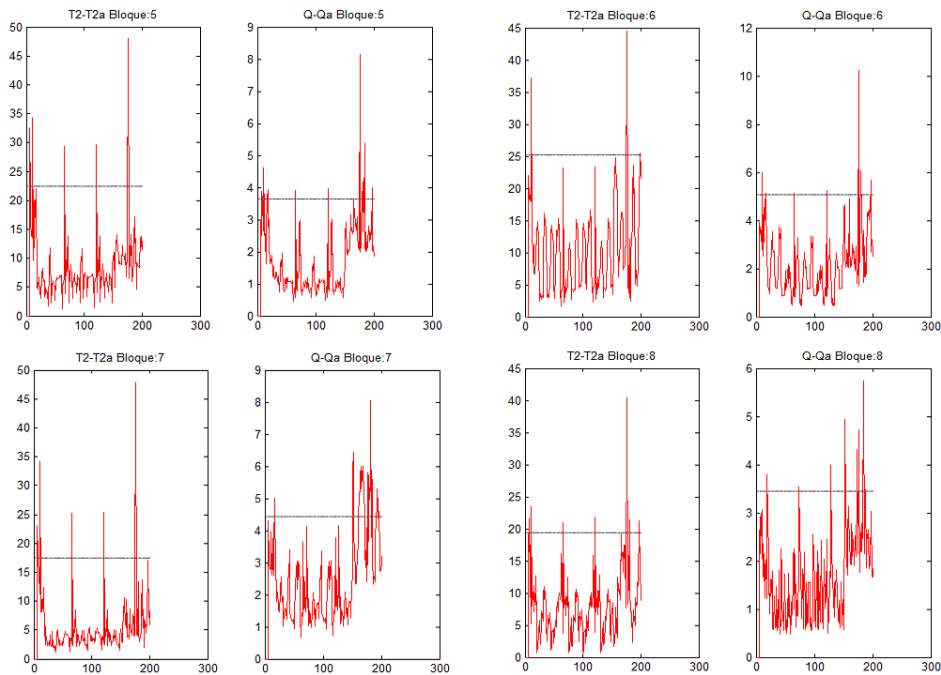


Figura 3.9: *Merging* PCA. Fallo bomba 10%. Bloques 5-8

Tabla 3.6: Resultados. Método *Merging* PCA

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	165	-	Demanda nodo 106	-	0,5051	-
2	-	-	-	-	-	-
3	-	161	-	Demanda nodo 306	-	0,5483
4	-	160	-	Presión nodo 402	-	0,6715
5	-	-	-	-	-	-
6	153	-	Flujo tubería 623	-	0,8741	-
7	-	161	-	Pérdidas altura tubería 789	-	0,6118
8	-	-	-	-	-	-

que 4, detecta el fallo en el instante 160 y apunta a la variable de presión en el nodo 402 como responsable, pero esta variable no es el origen del fallo y por tanto el método no identifica correctamente.

Método PCA distribuido usando factorización QR y clustering:

En este caso, el análisis va a ser un poco diferente al del resto de métodos, ya que ahora se va a ir comprobando, en cada instante, si algún bloque tiene clústers por encima del umbral.

Se analiza ahora cuántos clústers superan el umbral en cada bloque a lo largo de la simulación, esto se ve en las Figuras 3.10, 3.11, 3.12 y 3.13,

donde, unos instantes después de aparecer el fallo ya hay bloques con clústers anómalos.

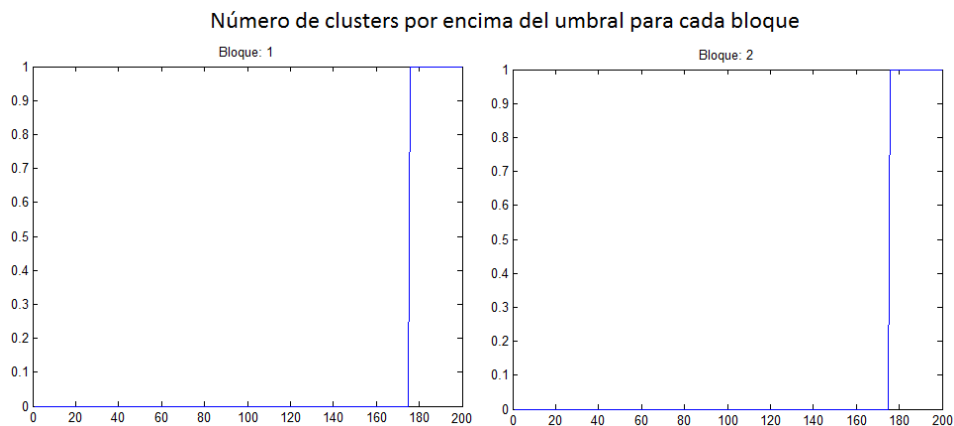


Figura 3.10: QR con clustering. Fallo bomba 10 %. Bloques 1-2

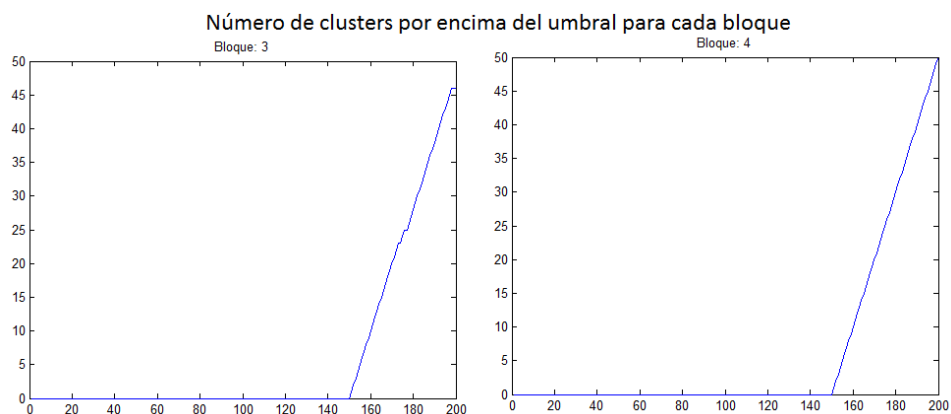


Figura 3.11: QR con clustering. Fallo bomba 10 %. Bloques 3-4

Concretamente, en el instante 152 el método de monitorización ya ha localizado bloques que contienen clústers que superan el umbral (ver Figura 3.14), concretamente, se detecta el fallo en los bloques 3, 4 y 6.

Tabla 3.7: Resultados. Método QR con clústers

Tiempo	Variable	Falsas alarmas
152	Demanda nodo 306	0

En la Tabla 3.7 se muestran los resultados de la detección y de la identificación. Para la identificación, como se comentó anteriormente, se analiza

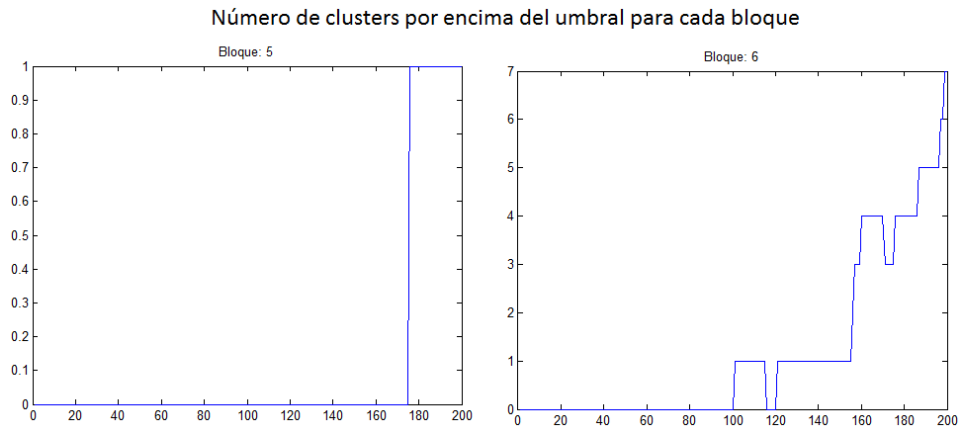


Figura 3.12: QR con clustering. Fallo bomba 10 %. Bloques 5-6

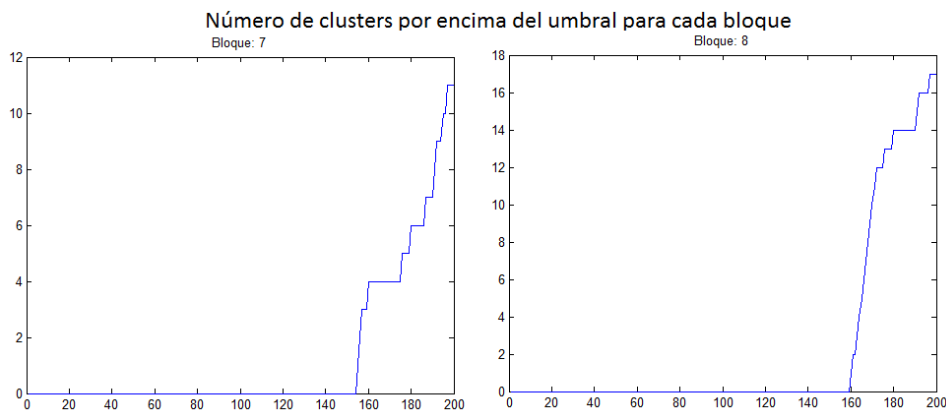


Figura 3.13: QR con clustering. Fallo bomba 10 %. Bloques 7-8

la contribución a Q en los bloques con mayor número de clústers por encima del umbral en el instante de la detección del fallo. Los resultados de la identificación indican que el fallo está localizado en el bloque 3 y, dentro de este, en la variable que mide la demanda de agua en el nodo 306, que no tiene nada que ver con el origen del fallo, que estaba situado en la bomba de la instalación, por tanto, no se ha conseguido una identificación correcta.

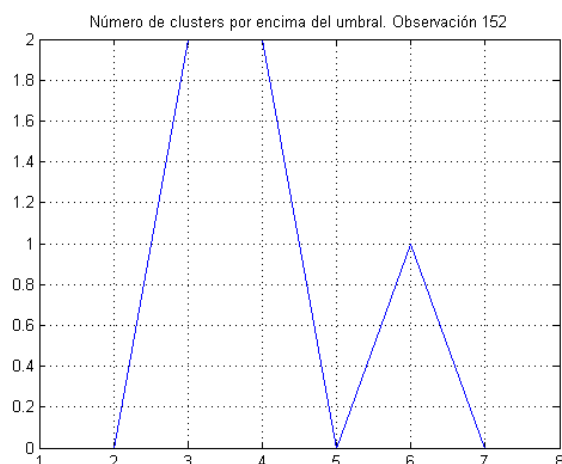


Figura 3.14: QR con clustering. Fallo bomba 10%. Clústers anómalos en cada bloque.

3.3.2. Caso de estudio II: Depuradora de aguas residuales

Ahora se van a aplicar los métodos expuestos antes sobre una planta depuradora de aguas residuales. Esta planta corresponde al modelo BSM2, explicada en el Anexo A.2.

3.3.2.1. Metodología experimental

En esta planta se dispone de datos de 7 variables medidas en 20 puntos diferentes por lo que hay disponibles 140 datos. En cuanto a la división de la planta: en el PCA centralizado se han procesado todas las variables en el mismo bloque mientras que en los métodos distribuidos (PCA distribuido, CPCA, etc.) se ha dividido la planta en 20 bloques que se corresponden con los 20 puntos de medición de los que se dispone.

Todos los PCA se han entrenado para conseguir un 70% de varianza incluida en los componentes principales seleccionados, siendo este valor un compromiso entre reducir la dimensionalidad del modelo y no perder mucha información al usar los componentes principales elegidos. Después de varias pruebas el número de observaciones anómalas consecutivas para detectar un fallo se ajustó al valor de 83 para todos los métodos. En el caso de los clústers del método con factorización QR y *Clustering*, el umbral para el radio de los clústers se estableció en $w_c = 0,1$ y para que se avise de un fallo debe haber, al menos, un 10% del total de clústers sobrepasando el umbral.

Se cuenta con datos de 16 simulaciones diferentes de fallos, con 58464 observaciones cada una, de las que se descartan las 10001 primeras al con-

siderarse que forman parte del período inicial de estabilización de la planta.

Al igual que en el caso de la red de distribución de agua se van a mostrar resultados de fallos detectados, fallos identificados correctamente, retraso en la detección, índice OTI (*Overall Type I risk*) e índice TI (*Type I risk*).

3.3.2.2. Análisis de resultados

En las Tablas 3.8 y 3.9 aparecen reflejados los resultados obtenidos por cada cada método.

Tabla 3.8: 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
<i>Merging</i> PCA	16	16	10	13	3.82	5.02
PCA (QR)	13	16	6	12	4.18	9.56
PCA & <i>Clustering</i>	16		11		0	

Tabla 3.9: 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)
<i>Merging</i> 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 & <i>Clustering</i>	28.73		0 % (0/16)	

En líneas generales, se observa que el estadístico Q da mejores resultados que el T^2 , ya que detecta e identifica una mayor cantidad de fallos, y es, casi siempre, más rápido.

El método centralizado es capaz de detectar todos los fallos, tanto con T^2 como con Q , además de identificar correctamente el 37,5 % y el 93,75 %, respectivamente, de los 16 fallos. Además, T^2 no presenta falsas alarmas mientras que con Q aparece en una de las 16 simulaciones con fallo. Los resultados de retraso en la detección arrojan unos valores medios de 1429.8 para T^2 y 404.6 observaciones para Q , que no están cerca de los mejores valores de la comparativa. Se puede decir que es un método que funciona bastante bien, pero no es el mejor, como se verá a continuación.

El PCA distribuido da mejores resultados que el centralizado: es capaz de

detectar todos los fallos con los dos estadísticos e identifica correctamente el 68,75 % con T^2 y el 93,75 % con Q . Sólo aparece una simulación con falsas alarmas en Q y consigue unos tiempos de detección muy buenos al presentar unos retrasos de 558.6 y 5.13 observaciones, respectivamente con cada estadístico.

Los resultados del CPCA muestran que detecta casi todos los fallos pero no consigue identificar ninguno de ellos. No presenta falsas alarmas pero sus tiempos de detección no son buenos ya que es el método que tarda mas en detectar los fallos. Da el peor resultado de la comparativa.

En cuanto al método *Merging* PCA detecta el 77,78 % e identifica el 62,5 % de los fallos con T^2 . Detecta el 100 % e identifica el 81,25 % de los fallos con Q . Tiene un número alto de falsas alarmas y los retardos en la detección son de 9.6 y 444.8 observaciones para cada estadístico.

El método de factorización QR es capaz de detectar todos los fallos con Q , mientras que identifica el 37,5 % de los fallos con T^2 , y el 75 %, con Q . Los retardos medios son de 108.92 y 166.8 observaciones para cada método. Aunque es un método con buenos resultados, es peor que el anterior y además presenta una complejidad de cálculo alta y una mayor cantidad de datos transmitidos.

Respecto al método basado en factorización QR y clustering, consigue detectar el 100 % de los fallos e identificar el 68,75 %. Mientras que el retardo medio en la detección es de 28.73 observaciones y no tiene simulaciones con falsas alarmas. Se puede afirmar que es un método que funciona bien y da buenos resultados, pero tiene como desventajas que requiere efectuar muchos cálculos, trabaja con gran cantidad de información y necesita un dispositivo de cálculo en cada bloque.

Por tanto, según lo visto en los resultados, el mejor resultado se consigue con el PCA distribuido propuesto al ser capaz, con Q , de detectar todos los fallos e identificar todos salvo uno. Además, con dicho estadístico, es el más rápido, presentando un bajo nivel de falsas alarmas. Adicionalmente, con T^2 trabaja mejor que los demás métodos en casi todos los casos, aunque da unos retardos de detección algo altos.

Para ver cómo trabajan los métodos, se van a mostrar un ejemplo de aplicación para uno de los 16 fallos de los que se dispone en la planta BSM2. Concretamente, este fallo consiste en simular un error en la lectura del sensor de oxígeno disuelto en el reactor 4, que es el dato que utiliza el sistema de control para calcular la cantidad de oxígeno que hay que introducir en los reactores 3, 4 y 5. El fallo implicará un aumento del 50 % del valor medido por el sensor, lo que provocará que el control detecte un exceso ficticio de oxígeno en los reactores y aplique acciones de control destinadas a reducir dicho nivel, provocando que haya menos oxígeno del necesario en los reactores.

El fallo aparecerá en el instante 18799, es decir, a los 3 días de comenzar la simulación. En las gráficas se ha representado una observación de cada 100

para facilitar su visualización, por tanto, en dichas gráficas, el fallo aparecerá en el instante 188.

Método PCA centralizado: En este caso, la evolución de los índices de detección se muestran en la gráfica 3.15, y los resultados obtenidos aparecen en la Tabla 3.10.

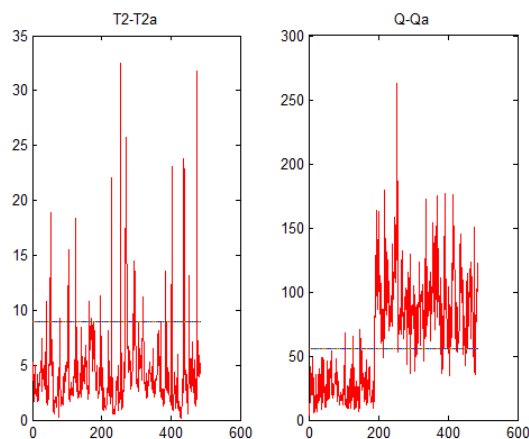


Figura 3.15: PCA centralizado. Fallo 50 %.

Tabla 3.10: Resultados. Método PCA centralizado

Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Contribución T^2	Contribución Q
-	19271	-	O_2 reactor 4	-	7.4230

Como se puede observar, T^2 no detecta ningún fallo y Q lo detecta claramente en el instante 19271, identificando como variable responsable del fallo al nivel de oxígeno en el reactor 4, que es justamente donde está el fallo. Por tanto, Q da un buen resultado de diagnóstico.

Método PCA distribuido: En este caso, al haber 20 bloques, para facilitar la visualización sólo se han representado las gráficas de aquellos bloques en los que se ha detectado el fallo. La evaluación de los estadísticos se muestra en la Figura 3.16, mientras que los resultados de cada uno de los bloques están contenidos en la Tabla 3.11

El índice de detección T^2 sí parece que presenta variaciones en su comportamiento a partir del instante en que aparece el fallo, pero no supera claramente el umbral en ninguno de los bloques. Por el contrario, el índice Q supera claramente su umbral en el bloque 7, y en el resto de bloques se aprecian cambios después del fallo, pero no con tanta claridad.

Analizando los resultados de cada uno de los bloques (Tabla 3.11) se

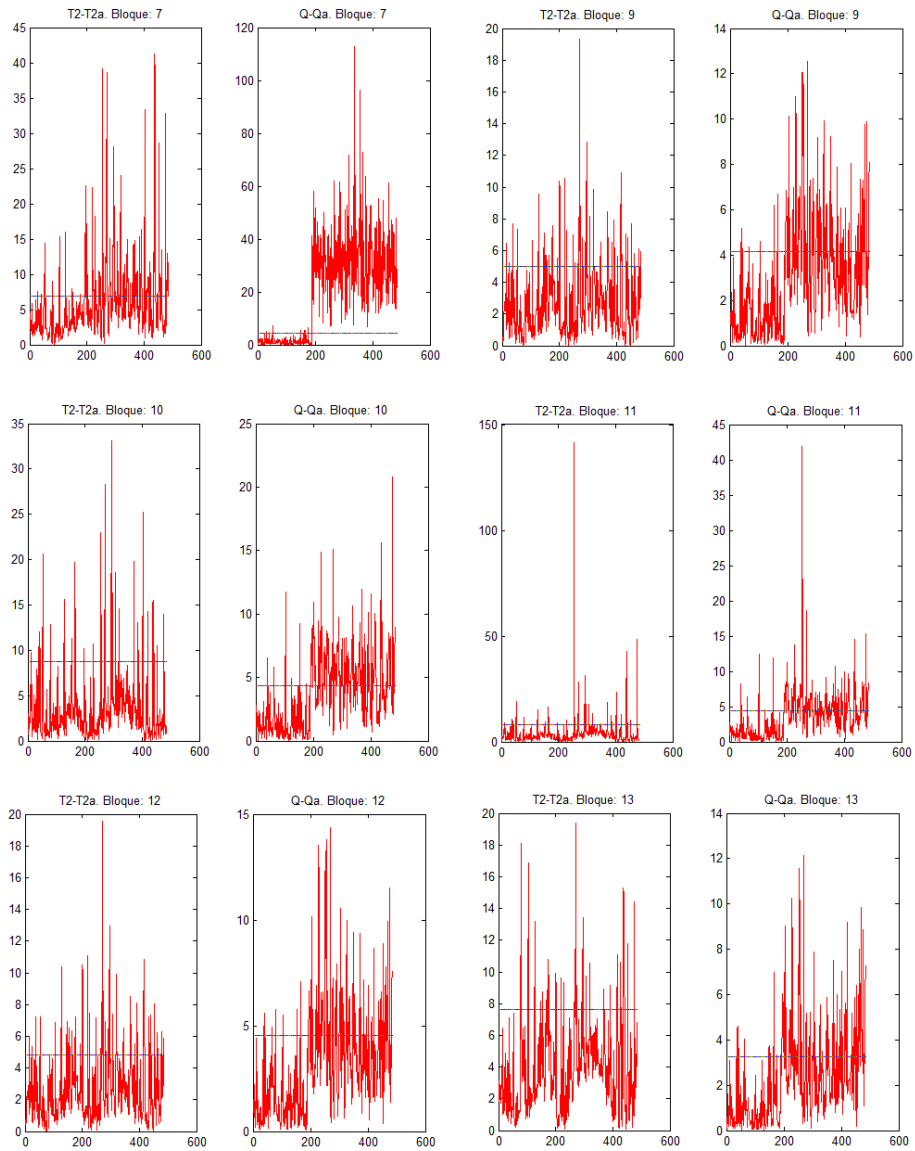


Figura 3.16: PCA distribuido. Fallo 50 %.

Tabla 3.11: Resultados. Método PCA distribuido

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	47466	-	Flujo entrada	-	2,6495
2	-	-	-	-	-	-
3	-	-	-	-	-	-
4	26842	-	Flujo reactor 1	-	3,2230	-
5	-	-	-	-	-	-
6	-	26897	-	Flujo reactor 3	-	3,6823
7	25358	18815	O_2 reactor 4	reactor 4 O_2	4,6409	7,0132
8	25362	-	Flujo reactor 5	-	6,96735248	-
9	26919	-	O_2 asentador flujo-inf.	-	1,6037	-
1-	25388	-	Flujo asentador salida	-	3,3351	-
11	-	-	-	-	-	-
12	26927	-	S_{alk} espesador flujo-sup.	-	4,0369	-
13	43652	-	Flujo espesador flujo-inf.	-	8,6893	-
14	-	47466	-	Flujo digestor entrada	-	2,6986
15	27020	-	SS digestor salida	-	1,5743	-
16	-	-	-	-	-	-
17	-	-	-	-	-	-
18	-	-	-	-	-	-
19	-	-	-	-	-	-
20	27003	-	S_{alk} fangos	-	1,8870	-

observa que el fallo sólo se detecta en algunos bloques, siendo el bloque 7, con el estadístico Q el que lo hace más rápido (sólo 16 instantes después de aparecer el fallo) y por tanto será el bloque cuyos datos se usen para la identificación. Esta apunta a la variable que indica el nivel de oxígeno en el reactor 4, que es justamente donde está el fallo, luego la identificación proporcionada por este método es correcta. También el estadístico T^2 detecta el fallo con el bloque 7 en primer lugar y proporciona una identificación correcta.

Método PCA distribuido con factorización QR: Igual que en el método anterior no se van mostrar gráficas de todos los bloques, en concreto, en la Figura 3.17 se puede ver el comportamiento del bloque 7, que es dónde antes se detecta el fallo. En la Tabla 3.12 están los resultados de todos los bloques.

Lo primero que llama la atención es que ningún bloque es capaz de detectar el fallo con el estadístico T^2 . Por otro lado sólo algunos bloques detectan el fallo con Q , siendo el bloque 7 el más rápido, como se ha dicho antes. Al ser el más rápido, este bloque será el que se use para hacer la identificación, y la variable con mayor contribución en este bloque es la cantidad de oxígeno en el reactor 4, que es precisamente donde se ha provocado el fallo. Por tanto la identificación es correcta.

Método CPCA: En la Figura 3.18 se muestran la evolución de los estadísticos T^2 y Q para el sistema global y en la Tabla 3.13 está el resultado obtenido.

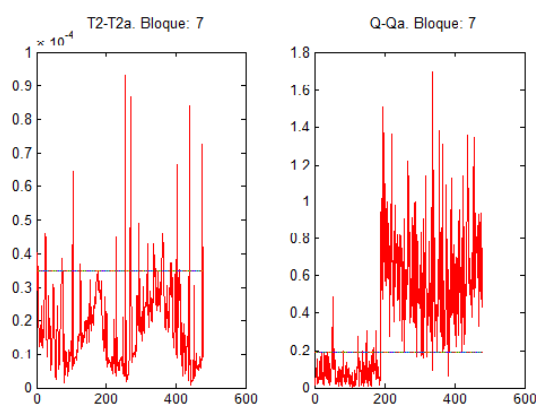


Figura 3.17: Factorización QR. Fallo 50 %.

Tabla 3.12: Resultados. Método PCA con factorización QR

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	-	-	-	-	-
2	-	27468	-	Temp. decantador flujo-sup.	-	0,4967
3	-	27468	-	Temp. decantador flujo-inf.	-	0,4967
4	-	25135	-	Flujo reactor 1	-	0,4581
5	-	25181	-	Flujo reactor 2	-	0,5262
6	-	-	-	-	-	-
7	-	18910	-	O ₂ reactor 4	-	0,8117
8	-	-	-	-	-	-
9	-	-	-	-	-	-
10	-	27463	-	Temp. asentador salida	-	0,52762
11	-	-	-	-	-	-
12	-	-	-	-	-	-
13	-	27467	-	Temp. espesador flujo-inf.	-	0,5298
14	-	27466	-	Temp. digestor entrada	-	0,4980
15	-	-	-	-	-	-
16	-	-	-	-	-	-
17	-	-	-	-	-	-
18	-	-	-	-	-	-
19	-	-	-	-	-	-
20	-	-	-	-	-	-

Tabla 3.13: Resultados. Método CPCA

Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Contribución T^2	Contribución Q
26836	25378	S_{alk} reactor 4	S_{alk} reactor 4	0,3678	4,6354

Los resultados muestran que T^2 detecta un fallo en el instante 26836 e identifica a la alcalinidad del reactor 4 como variable responsable. Esta no es la que origina el fallo, por tanto, la identificación no es correcta. De forma

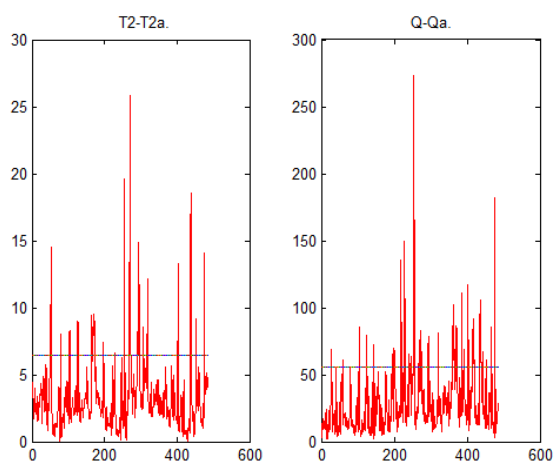


Figura 3.18: CPCA. Fallo 50 %.

similar, Q detecta el fallo también bastante tarde, en el instante 25378, e identifica a la misma variable que T^2 . Por tanto, se observa que el método, en este fallo, es lento y no proporciona una buena identificación.

Método PCA distribuido fusionado: Al igual que en los métodos con 20 bloques independientes, aquí sólo se mostrarán las gráficas de los bloques que detecten el fallo (Figuras 3.19 y 3.20). Los resultados de cada bloque aparecen en la Tabla 3.14.

Tabla 3.14: Resultados. Método *Merging* PCA

Bloque	Tiempo T^2	Tiempo Q	Variable T^2	Variable Q	Cont T^2	Cont Q
1	-	-	-	-	-	-
2	-	-	-	-	-	-
3	-	47466	-	Flujo decantador flujo-inf.	-	0,40453
4	23550	25363	DQO reactor 1	Flujo reactor 1	1,17E-06	0,5682
5	-	-	-	-	-	-
6	-	-	-	-	-	-
7	18802	25362	Temp. reactor 4	O_2 reactor 4	4,46E-06	0,6190
8	23546	25364	N reactor 5	Flujo reactor 5	2,83E-06	0,6411
9	23552	25380	N asentador flujo-inf.	O_2 asentador flujo-inf.	1,35E-05	0,5569
10	29449	25360	S_{alk} asentador salida	O_2 asentador salida	3,76E-05	0,4685
11	28875	25360	Temp. WWTP salida	O_2 WWTP salida	9,54E-06	0,4684
12	27437	25382	Temp. espesador flujo-sup.	O_2 espesador flujo-sup.	1,44E-06	0,5893
13	28686	25380	Temp. espesador flujo-inf.	O_2 espesador flujo-inf.	1,05E-05	0,5569
14	-	25377	-	O_2 digestor entrada	-	0,3663
15	26948	26851	N digestor salida	Flujo digestor salida	4,31E-05	0,3932
16	27040	26856	N deshidratador-salida	Flujo deshidratador salida	7,05E-05	0,45560
17	-	-	-	-	-	-
18	27040	26856	N almacenado- <i>bypass</i>	Flujo almacenado- <i>bypass</i>	7,05E-05	0,4556
19	27040	26856	N almacenado-salida+ <i>bypass</i>	Flujo almacenado-salida+ <i>bypass</i>	7,05E-05	0,4556
20	27136	36268	N fangos	Temp. fangos	6,92E-05	0,3662

Los resultados muestran que el bloque 7 con el estadístico T^2 es el más rápido al hacer la detección, pero no consigue identificar correctamente el

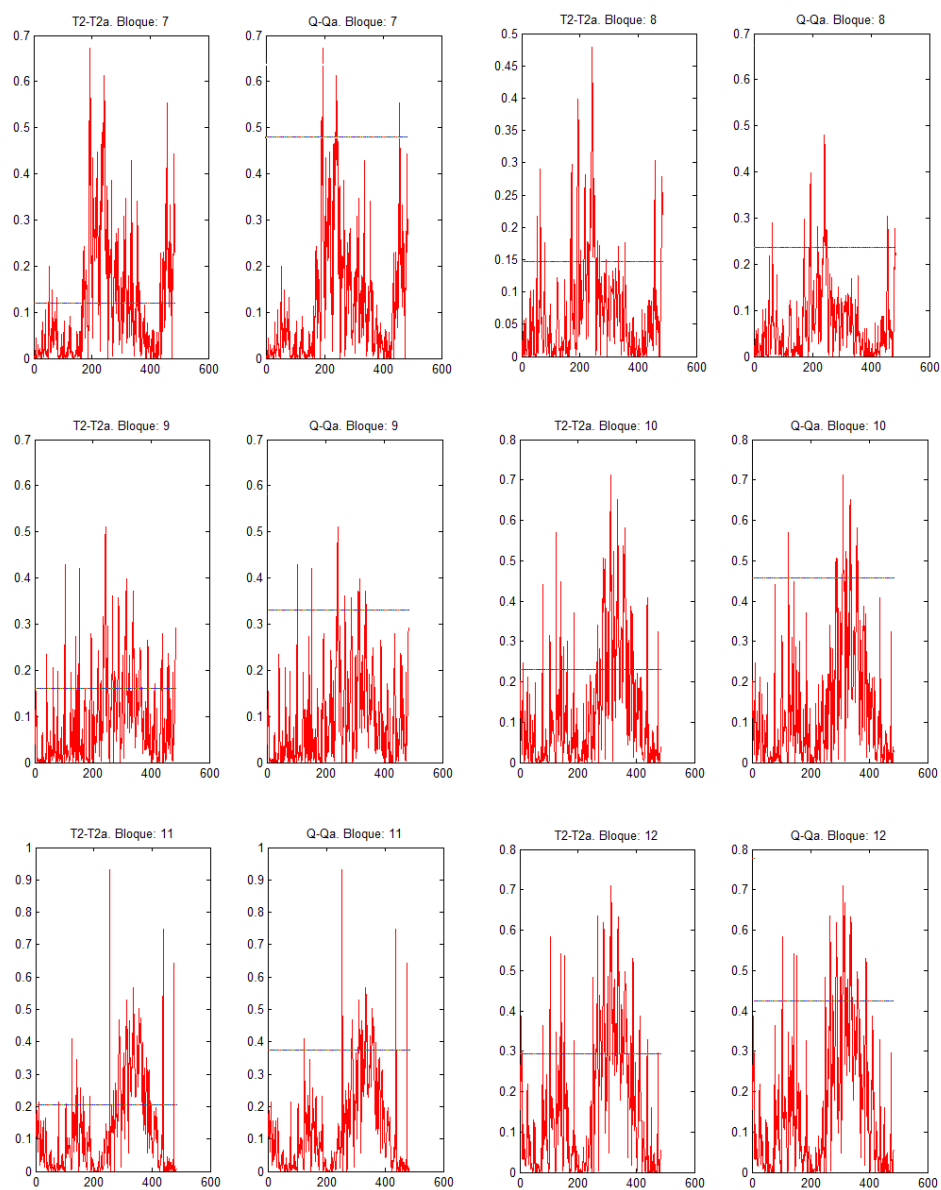


Figura 3.19: *Merging* PCA. Bloques 7, 8, 9, 10, 11 y 12.

fallo al apuntar a la temperatura del reactor 4, que no es la que origina el fallo. Por otro lado, varios bloques detectan el fallo usando Q , siendo los más rápidos el 10 y el 11. Al hacer la identificación, no se obtiene el origen del fallo ya que el origen de éste se encuentra en el bloque 7 y no en el 10 ni en el 11. Por tanto, este método no da buen resultado y además, presenta muchas falsas alarmas.

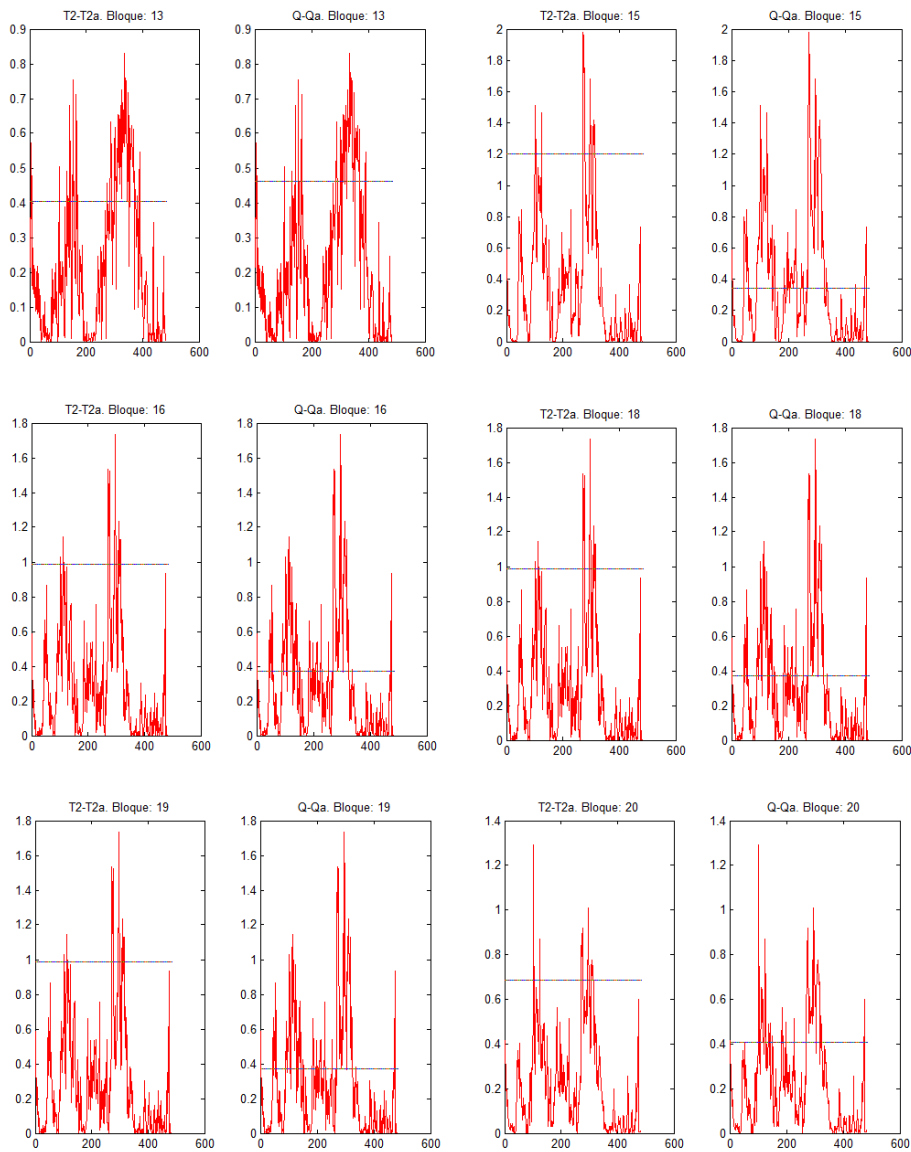


Figura 3.20: *Merging* PCA. Bloques 13, 15, 16, 18, 19 y 20.

Método PCA distribuido basado en factorización QR y clustering:

En este método se analiza el número de clústers que superan el umbral para cada uno de los bloques en cada instante. La primera detección del fallo se produce en el instante 28843. En ese instante, se puede ver qué bloques tienen clústers anómalos en la Figura 3.21, esos bloques son, concretamente, el bloque 7 y el 17. En el 7 es donde hay más de un 10 % de clústers anómalos, y por tanto, se considera que el fallo se ha detectado en ese bloque.

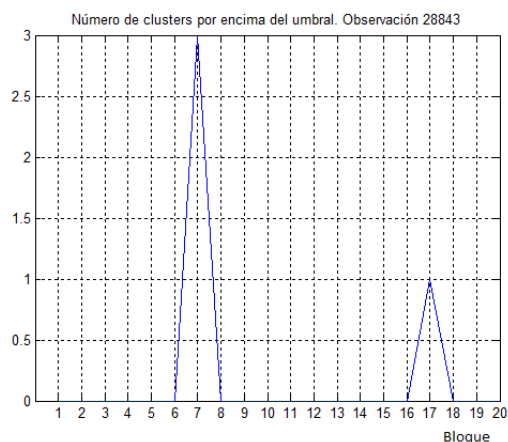


Figura 3.21: Fallo 50% en el sensor de oxígeno. Número de clústers por encima del umbral en el instante 28843.

En la Figura 3.22 se observan cuántos clústers anómalos tiene el bloque 7 (que es el bloque que corresponde a los sensores del reactor 4). Se observa como a partir de la aparición del fallo el número de clústers que superan el umbral crece hasta un máximo de 3, bajando luego a cero y luego subiendo otra vez hasta 4.

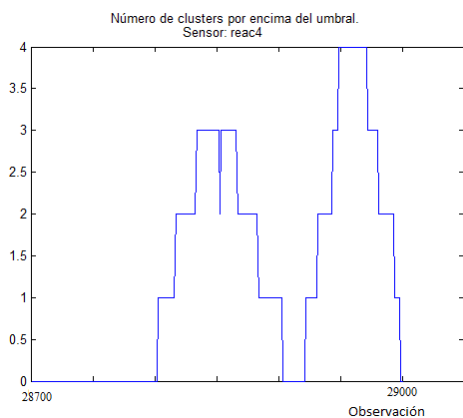


Figura 3.22: Fallo 50% en el sensor de oxígeno. Evolución del número de clústers anómalos del reactor 4.

En la Tabla 3.15 se muestran los resultados del análisis de contribuciones a Q para el bloque 7 (reactor 4). Dicho análisis identifica la temperatura del este reactor como la variable que origina el fallo. Como se ha dicho, el fallo es del sensor de oxígeno de este reactor, y por tanto el resultado de la identificación no es correcto.

Tabla 3.15: Resultados. Método QR con clustering

Tiempo	Bloque	Variable	Falsas alarmas
28843	reactor 4	Temp.	0

3.4. Resumen

Se ha presentado en este capítulo una primera aproximación a los métodos distribuido basados en PCA. Se ha propuesto un método en el que la distribución se hace en base a criterios topológicos (las variables se agrupan en función de la proximidad entre ellas). Los resultados globales se obtienen al recoger los datos locales y tomar una decisión en base a ellos. Los experimentos llevados a cabo demuestran que es un método eficaz y que funciona mejor que un PCA centralizado: identifica mejor los fallos y es más rápido.

Además, se han probado otros métodos distribuidos basados en PCA, no diseñados específicamente para diagnóstico de fallos, sino para reducir la cantidad de información transmitida entre los nodos de un sistema. Los resultados muestran que algunos de ellos funcionan bien en términos de fallos detectados e identificados correctamente, tiempos de detección, etc., aunque no llegan a dar el resultado del PCA distribuido propuesto.

El trabajo de este capítulo sirve para confirmar que la estrategia distribuida puede ser más eficaz que la centralizada, pero requiere de un mayor desarrollo del método distribuido que se expondrá en los siguientes capítulos. Concretamente, el método PCA es un método que aunque puede ser efectivo en ocasiones, es de tipo lineal y no dinámico, y esto puede hacer difícil el diagnóstico de plantas complejas. Por tanto se deben buscar alternativas con otros métodos basados en datos, en modelos, etc. que permitan analizar sistemas no lineales y dinámicos.

También se debe mejorar el proceso de fusión de los resultados de los diferentes bloques, buscando un método que tenga una base teórica más sólida que el método presentado en este capítulo. Además, la forma de dividir la planta basada en criterios topológicos es demasiado subjetiva y requiere de un sólido conocimiento de la planta que no siempre se tiene.

Otro punto sobre el que trabajar es el uso de técnicas de identificación más avanzadas que el método de análisis de contribuciones, que aunque da buenos resultados en ciertas ocasiones, no siempre es efectivo.

Capítulo 4

MÉTODOS FDI DISTRIBUIDOS MULTIBLOQUE BASADOS EN DPCA Y CVA. DISTRIBUCION DE MODELOS

4.1. Introducción

El principal objetivo de esta tesis es desarrollar métodos de monitorización de plantas complejas. La forma más convencional de realizar esta tarea en este tipo de plantas consiste en procesar todos los datos en un único procesador central, pero debido a las características de este tipo de plantas, esta forma de trabajar puede no ser la más adecuada. En efecto, en una planta compleja es esperable contar con gran número de dispositivos que envíen mucha cantidad de información: sensores, controladores, etc. y esto puede hacer que una única unidad de procesamiento sea incapaz de manejar todos esos datos. Además, para transmitir toda esa información se requerirá una red de comunicaciones compleja y costosa, y no siempre va a ser factible su implementación. Por tanto, es preferible optar por otras alternativas, una de ellas es la descentralización de la planta, en la que se divide la misma en bloques de variables que son procesados de forma independiente, y el procesador central únicamente tendrá que fusionar los resultados de cada uno de los bloques. Esto simplifica mucho los requisitos de comunicaciones y de capacidad de procesamiento.

En este capítulo se va a trabajar con un tipo de distribución en la que se descentralizan los modelos. Es decir, se divide la planta en bloques y en cada uno de ellos se genera un modelo diferente para procesar los datos.

4.2. Fundamentos teóricos

Como se pretende dividir la planta en bloques de variables se presentan a continuación una serie de técnicas que pueden ser utilizadas para encontrar relaciones entre variables. Estas relaciones serán la base que se utilizará posteriormente para agrupar las variables en la descentralización. Y se podrá llevar a cabo sin necesidad de tener ningún conocimiento previo.

Entre los métodos que se presentarán a continuación se pueden encontrar métodos basados en información de la señal como correlación, información mutua, etc, así como métodos basados en regresión lineal y en regresión no lineal que permiten conocer de alguna forma la relación entre las variables del proceso. Aquí se explicará brevemente su fundamento teórico y después se comentará como se usan estas técnicas para dividir las variables de la planta en bloques.

4.2.1. Métodos de regresión lineal

4.2.1.1. Regresión LASSO

El método de regresión LASSO (en inglés, *Least Absolute Shrinkage and Selection Operator*) presentado en [Tibshirani, 1996] es una versión mejorada del modelo de regresión lineal. Se basa en aplicar una penalización a los coeficientes de regresión, haciendo nulos algunos de ellos. Con esto se consigue un modelo con menos variables, y, por tanto, más sencillo e interpretable. Además, como sólo aquellas variables relevantes en la regresión tendrán coeficientes no nulos, este método se puede usar como una técnica de selección de características de tipo lineal.

Un modelo de regresión lineal con m predictores:

$$\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_m(t)) \quad (4.1)$$

y una variable respuesta $y(t)$, se puede representar de la forma:

$$y(t) = \beta_0 + \beta_1 x_1(t) + \beta_2 x_2(t) + \dots + \beta_m x_m(t) + \varepsilon(t) \quad (4.2)$$

donde β_i ($i = 1, \dots, m$) son los coeficientes de regresión y $\varepsilon(t)$ es el error del modelo en el instante t . Usando notación matricial:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \quad (4.3)$$

donde \mathbf{y} es el vector de salida con dimensiones $n \times 1$, \mathbf{X} (de dimensión $n \times m$) contiene las medidas de los predictores en los instantes 1 a n , $\boldsymbol{\beta}$ es el vector de coeficientes de regresión ($\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_m)$) y \mathbf{e} es el vector de errores en cada instante (de dimensión $n \times 1$).

El método LASSO busca la solución del problema:

$$\min \left(\frac{\|y - \mathbf{X}\boldsymbol{\beta}\|_2^2}{n} \right) \quad \text{subject to } \sum_{j=1}^m \|\beta_j\|_1 \leq l \quad (4.4)$$

donde $l \geq 0$ es el parámetro que se debe ajustar para controlar el número de coeficientes no nulos. Este problema es equivalente a:

$$\hat{\boldsymbol{\beta}}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\frac{\|y - \mathbf{X}\boldsymbol{\beta}\|_2^2}{n} + \lambda \|\boldsymbol{\beta}\|_1 \right) \quad (4.5)$$

donde $\lambda \geq 0$. El parámetro λ controla el proceso: cuanto mayor sea su valor, más coeficientes se harán cero.

4.2.1.2. Sparse PLS

El método de Mínimos Cuadrados Parciales (PLS), presentado en [Wold, 1966], se basa en aplicar conceptos de componentes principales a una regresión [Colombani et al., 2012]. Inicialmente, se tiene una matriz de datos \mathbf{Z} ($n \times m$) de un sistema con m variables. Esta matriz se divide en dos subconjuntos: una matriz \mathbf{X} que recoge los datos de las variables consideradas como predictoras, y otra matriz \mathbf{Y} que recopila los datos de las variables respuesta. Este método permite realizar reducciones de dimensionalidad, regresiones, etc. mediante la obtención de un conjunto de nuevas variables latentes que son combinación lineal de las originales. Luego, ajusta un modelo lineal usando mínimos cuadrados sobre estas nuevas variables.

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \quad (4.6)$$

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^T + \mathbf{F} \quad (4.7)$$

donde \mathbf{X} es la matriz de predictores ($n \times m$), \mathbf{Y} es la matriz de respuestas ($n \times p$), \mathbf{T} son los k primeros términos de las variables latentes, \mathbf{P} y \mathbf{Q} , son, respectivamente, los vectores de cargas de las matrices de datos \mathbf{X} e \mathbf{Y} , y \mathbf{E} y \mathbf{F} son los residuos del modelo. Cada variable latente se obtiene a través de la deflacción de \mathbf{X} e \mathbf{Y} usando el conocido algoritmo de Mínimos cuadrados parciales no lineales iterativos (NIPALS) hasta que toda la información contenida en los datos ha sido explicada [Wold et al., 2001]. Estos vectores de variable latentes se calculan así:

$$\mathbf{T} = \mathbf{X}\mathbf{R} \quad (4.8)$$

La matriz $\mathbf{R} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}$ es muy útil para mostrar que variables de \mathbf{X} tienen más influencia en el modelo de la matriz de respuestas \mathbf{Y} . El modelo de regresión aplicando PLS es:

$$\mathbf{Y} = \mathbf{X}\mathbf{B}_{\text{PLS}} + \mathbf{F}, \quad \mathbf{B}_{\text{PLS}} = \mathbf{R}\mathbf{Q}^T \quad (4.9)$$

Cuando se trabaja con métodos de regresión con gran cantidad de variables, la interpretabilidad del modelo suele ser reducida ya que hay muchos predictores relevantes y no es fácil discriminar cuáles son los que tienen más influencia en la salida. Para simplificar el modelo y facilitar su manejo se puede recurrir a métodos que consiguen hacer nulos algunos de los elementos del vector de coeficientes, al estilo de lo que propone el método LASSO. El método *Sparse* PLS (SPLS) [Chun and Keles, 2010], basado en PLS, calcula un vector de regresión \mathbf{B}_{PLS} en el que varios de sus elementos son nulos, mientras que aquellas variables más relevantes en la regresión tendrán asociados coeficientes no nulos. Por tanto, se puede usar como método de selección de características si se toman esas variables con mayor relevancia en el modelo.

4.2.1.3. Sparse PCA

El método *Sparse* PCA (SPCA), [d'Aspremont et al., 2007, Zou et al., 2006], permite distribuir las variables de la planta en bloques en función de las posiciones no nulas de la matriz de cargas del modelo PCA obtenido.

Partiendo de una matriz de datos $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_m^T]$ y calculando su matriz de covarianzas \mathbf{S} , el método busca obtener la descomposición de \mathbf{S} en componentes *sparse*: $[\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$, que deben tener, cada uno de ellos, un número r de sus elementos distintos de cero (cardinalidad). El problema a resolver es maximizar la varianza de cada vector $\mathbf{v}_i \in \mathbb{R}^m$, para $i = 1, \dots, m$, con la restricción de la cardinalidad. Es decir:

$$\max \mathbf{v}^T \mathbf{S} \mathbf{v} \text{ subject to } \|\mathbf{v}\|_2 = 1, \text{ Card}(\mathbf{v}) \leq r \quad (4.10)$$

Una vez obtenido el primer componente *sparse* se actualiza la matriz \mathbf{S} :

$$\mathbf{S} = \mathbf{S} - (\mathbf{v}_1^T \mathbf{S} \mathbf{v}_1) \mathbf{v}_1 \mathbf{v}_1^T \quad (4.11)$$

repetiéndose el proceso con la matriz de covarianza actualizada para obtener el siguiente componente principal. Esto se haría hasta tener los m componentes principales.

4.2.2. Métodos de regresión no-lineal

4.2.2.1. Redes Neuronales

Una de la técnicas de modelado no lineal más utilizadas son las redes neuronales [Kuhn and Johnson, 2013], de las que existen diferentes modelos de redes: Perceptrón Multicapa (MLP), Redes recurrentes, *Autoencoders*, etc. En el modelo de Perceptrón Multicapa, que es ampliamente utilizado, el modelo que se genera procesa los datos de varias variables de entrada para entregar una o más salidas.

Una red neuronal está compuesta por un conjunto de unidades, o neuronas, conectadas entre sí que intercambian información. Los datos que se introducen en la red sufren diversas operaciones, obteniéndose unos valores a la salida de la red producto de dichas operaciones.

4.2.2.2. Bosques Aleatorios

El método de Árboles Aleatorios (*Random Forest* (RF)) es un método de clasificación basado en los árboles de decisión muy utilizado gracias a su sencillez y robustez frente a cambios en los parámetros [Breiman, 2001]. Aunque se suele usar para realizar clasificaciones [Zhou and Qiu, 2018] también es posible aplicarlos para regresión [Jog et al., 2017, Tsagkraloulis and Montana, 2018].

El método implementa varios árboles de decisión independientes entre sí, creados a partir de conjuntos de datos aleatorios con igual distribución. Si se parte de una matriz de datos \mathbf{X} ($n \times m$) con n observaciones y m variables, el proceso seguido consiste en ir creando las ramas de un árbol utilizando una muestra aleatoria de p predictores, con $p < m$, escogidos aleatoriamente entre las m variables. Al crear la siguiente rama se tomará otro subconjunto de p predictores seleccionado también de forma aleatoria. El valor p recomendado, en el caso de hacer una regresión, es de $p = m/3$, y en el caso de hacer clasificación, se recomienda $p \approx \sqrt{m}$. Esta metodología evita que, cuando se generen varios árboles distintos y existan varios predictores que tienen mucha influencia, los árboles estén correlacionados entre ellos [Breiman, 2001, Ho, 2002].

Cuando se quiera obtener un modelo de regresión, se crean varios árboles siguiendo éste método, de modo que, para obtener una predicción de una nueva observación, se calculará un promedio de las predicciones de cada árbol para dicha observación.

4.2.2.3. Regresión SVM

El método de Máquinas de Vector Soporte (en inglés, *Support Vector Machines* (SVM)) se usa para hacer clasificación de datos [Nor et al., 2017, Liu et al., 2017], selección de variables [Paul et al., 2016] y regresión (SVR) [Nieto et al., 2013, Zhang and Gao, 2018], y fue desarrollado en [Vapnik, 1995]. Esta técnica parte de un conjunto de datos de entrenamiento (x_i, y_i) , con $i = 1, \dots, N \subset \mathcal{X} \times \mathbb{R}$, donde \mathcal{X} denota el espacio de los patrones de entrada. El objetivo de la regresión SVM (o SVR) es encontrar una función $f(x)$ que tenga, a lo sumo, una desviación ε respecto de las variables de salida y_i para los datos de entrenamiento y, al mismo tiempo, tenga una geometría lo mas suavizada posible. Esto se conoce como ε -regresión SV:

$$f(x) = \langle w, x \rangle + b \tag{4.12}$$

con $w \in \mathcal{X}$, $b \in \mathbb{R}$. Si se trabaja con un problema no lineal, se puede usar SVM como función de regresión no lineal incorporando la técnica de *kernel trick* [Boser et al., 1992] para transformar el espacio de entrada a otro de dimensión superior, en el que se puede construir una función lineal [Elattar et al., 2010, Saludes-Rodil and Fuente, 2010]. El objetivo perseguido es encontrar una transformación del espacio de datos de entrada a otro espacio de dimensión mayor en el que se pueda realizar una regresión lineal [Smola and Schölkopf, 2004]. Se establece un valor umbral ε de tal forma que el método encuentre un modelo de regresión cuyos residuos no superen dicho umbral.

4.2.3. Métodos basados en información lineal y no-lineal

4.2.3.1. Correlación

La correlación entre dos variables \mathbf{x} e \mathbf{y} se define como una medida de la relación lineal entre dichas variables [James et al., 2013]. De tal forma que, si se tienen datos de n instantes de $\mathbf{x} = [x_1, \dots, x_n]$ e $\mathbf{y} = [y_1, \dots, y_n]$, la correlación es:

$$\mathcal{R}_{x,y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.13)$$

siendo \bar{x} y \bar{y} los valores medios de \mathbf{x} e \mathbf{y} . El valor obtenido varía entre -1 y 1 , de tal manera que si se obtiene 0 no hay ninguna influencia de una variable en otra; si sale 1 la influencia es total; y si sale -1 una variable influye completamente en la otra pero de forma inversa (cuando una crece la otra decrece y viceversa). Cuando se dispone de varias variables $[x^1, x^2, \dots, x^m]$, se puede obtener una matriz de correlación que contiene todos los valores de de correlación cruzada entre todas las variables. Tiene la siguiente forma:

$$\mathcal{R} = \begin{bmatrix} 1 & \mathcal{R}_{x^1, x^2} & \dots & \mathcal{R}_{x^1, x^m} \\ \mathcal{R}_{x^2, x^1} & 1 & \dots & \mathcal{R}_{x^2, x^m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{R}_{x^m, x^1} & \mathcal{R}_{x^m, x^2} & \dots & 1 \end{bmatrix} \quad (4.14)$$

4.2.3.2. DCCA

El análisis de correlación cruzada sin tendencia (en inglés, *Detrended Cross-Correlation Analysis* (DCCA)) se basa en analizar la covarianza entre variables eliminando las posibles tendencias que pudieran existir. Fue propuesto por [Podobnik and Stanley, 2008, Zebende, 2011] como una medida de la correlación entre variables aleatorias no estacionarias.

El valor de DCCA entre dos variables, x e y , es la relación entre la covarianza sin tendencia, $F_{x,y}^2$, y las varianzas sin tendencia, F_x y F_y , y se calcula de la forma:

$$\rho_{DCCA} = \frac{F_{x,y}^2}{F_x F_y} \quad (4.15)$$

El resultado obtenido es similar al de la correlación convencional, siendo ρ_{DCCA} un valor entre -1 y 1 , y si se obtiene un valor de $\rho_{DCCA} = 0$ quiere decir que no existe correlación cruzada sin tendencia entre las dos variables.

4.2.3.3. Información Mutua

La Información Mutua (*Mutual Information* (MI)), es, según la teoría de la información, una medida que cuantifica la dependencia mutua entre dos variables [Cover and Thomas, 2006]. Se puede definir como la cantidad de información que una variable aleatoria puede aportar acerca de otra variable aleatoria. También se puede considerar como la reducción de la incertidumbre sobre una variable cuando se conoce otra variable, de tal forma que si el valor de MI es alto, dicha reducción de incertidumbre es alta, y si es bajo, la reducción será baja. En el caso de obtener un valor nulo, no existe ninguna dependencia entre variables.

El valor de MI para dos variables aleatorias x e y se obtiene de la forma:

$$MI(x, y) = \sum_y \sum_x P_{(x,y)} \log\left(\frac{P_{(x,y)}}{P_{(x)}P_{(y)}}\right) \quad (4.16)$$

donde $P_{(x,y)}$ es la probabilidad conjunta de x e y , y $P_{(x)}$ y $P_{(y)}$ son las probabilidades marginales para dichas variables, respectivamente. Un valor de información mutua entre dos variables es relevante si supera un cierto umbral $(1 - \alpha)$, que debe ser definido por el usuario previamente. Si se supera ese umbral se entiende que entre esas variables hay una dependencia relevante que debe ser considerada.

4.2.4. Métodos basados en agrupamiento (*Clustering*)

El agrupamiento es una técnica de minería de datos que busca crear grupos de elementos de una forma automática basándose en la similitud o disimilitud entre dichos elementos. El objetivo es que los elementos contenidos en un grupo o *cluster* posean una gran similitud entre ellos a la vez que la similitud con elementos de otros *clusters* sea mínima.

- **DBSCAN:** Dentro de la gran variedad de algoritmos desarrollados para realizar *clustering* [Ahmad and Dey, 2007], como son: *K-Means*, Modelos de Mezcla Gaussianos, etc. existe un algoritmo ampliamente utilizado que es el Agrupamiento Espacial basado en Densidad de

Aplicaciones con Ruido (DBSCAN). En este método cada grupo estará caracterizado por un elemento llamado núcleo. Y cada elemento se asignará al grupo más próximo a él, siempre que la distancia de dicho elemento al núcleo de ese sea menor que un cierto valor ϵ , fijado inicialmente. Si algún elemento no entra en ningún grupo se considera como valor atípico [Kriegel et al., 2011].

- **Agrupamiento EM:** Otra técnica para realizar agrupamiento es el algoritmo de Maximización de las Expectativas (EM) para modelos de Mezcla Gaussianos [Zhang et al., 2003]. El algoritmo EM calcula las probabilidades de pertenencia de cada uno de los elementos a un grupo en función de una distribución de probabilidad Gaussiana. Cada uno de los grupos tendrá asignado una serie de parámetros que definirán la distribución de probabilidad que le corresponda (en el caso gaussiano: media y varianza, además del peso del grupo dentro de la mezcla), de modo que cuando se analice una observación nueva se pueda conocer la probabilidad de pertenencia a cada uno de los grupos y se le pueda asignar a aquel grupo en el que sea más alta. El objetivo del algoritmo es que cada punto quede asignado a un grupo de modo tal que se alcance la máxima probabilidad global general.

4.2.5. Métodos de toma de decisiones

Si se va a trabajar con una planta dividida en bloques, cada uno de ellos va a generar un resultado de diagnóstico, y, por tanto, se plantea el problema de la fusión de los resultados locales para obtener una diagnóstico global de la planta. Uno de los métodos más usados en la bibliografía consultada es el método de inferencia bayesiana [Tong et al., 2017].

El Criterio de Inferencia Bayesiano (*Bayesian Inference Criterion* (BIC)), en un método FDI descentralizado, toma en cada instante los valores de los estadísticos de control (T^2 y Q , en el caso del PCA, por ejemplo) que envía cada bloque y los fusiona. Esto se lleva a cabo usando el índice BIC [Tong and Shi, 2016, Jiang et al., 2016]. Para obtener este índice se calcula, para cada bloque i y para un estadístico ST (ST puede ser T^2 , Q , T_s^2 , etc.), lo siguiente:

$$P_{(F|x_i)} = P_{(x_i|F)}P_{(F)}/P_{(x_i)} \quad (4.17)$$

con

$$P_{(x_i)} = P_{(x_i|N)}P_{(N)} + P_{(x_i|F)}P_{(F)} \quad (4.18)$$

donde N es el estado normal de la planta, y F el estado con fallo. $P_{(N)}$ y $P_{(F)}$ son las probabilidades a priori de que el sistema esté trabajando sin

fallo y con fallo, respectivamente. Se toma un nivel de confianza de α para la primera y de $1 - \alpha$ para la segunda. Por otro lado:

$$P_{(x_i|N)} = e^{-ST_i/ST_{i,lim}}, P_{(x_i|F)} = e^{-ST_{i,lim}/ST_i} \quad (4.19)$$

siendo $ST_{i,lim}$ el umbral del estadístico ST en el bloque i . Finalmente, se fusionan los resultados de cada bloque según la expresión:

$$BIC_{ST} = \sum_{i=1}^m \frac{P_{(x_i|F)} P_{(F|x_i)}}{\sum_{i=1}^m P_{(x_i|F)}} \quad (4.20)$$

El criterio para detectar un fallo es el valor de BIC_{ST} para alguno de los estadísticos utilizados supere el valor umbral de $(1 - \alpha)$.

Se usará un determinado valor de α en los datos de entreno y luego se reajustará en los datos de test (usando siempre datos sin fallo) de modo que no aparezcan falsas alarmas (es decir, que no haya un determinado número de valores de BIC_{ST} anómalos consecutivos).

4.3. Método FDI propuesto: Método distribuido con PCA dinámico multi bloque: MB-DDPCA

El método propuesto para hacer la diagnosis descentralizada se compone de tres etapas que se detallan a continuación:

1. **División de la planta:** Se analizan las variables medidas en la planta en busca de posibles relaciones entre las mismas. El objetivo es crear bloques de variables que reúnan a aquellas que presenten un mayor nivel de correlación entre ellas, de modo que se consiga un método distribuido de monitorización con el mayor nivel de eficacia.
2. **Método local de detección de fallos:** Se crea un modelo de detección de fallos en cada uno de los bloques. Este modelo analiza únicamente la información contenida en las variables asignadas a cada bloque.
3. **Fusión de los resultados locales:** A partir de la detección de fallos en cada bloque es necesario dar una única diagnosis de toda la planta. Esto requiere de alguna técnica de fusión de información que, a partir de los datos enviados por los bloques, genere un resultado de detección único y global.

Por otro lado, los procesos que tienen lugar en una instalación industrial raramente son de tipo estático. Lo normal es que el estado actual del sistema dependa de la evolución que haya sufrido el mismo en el pasado. Por tanto,

hay que tener en cuenta que las variables medidas tienen carácter dinámico y sufren un cierto grado de autocorrelación.

Por tanto, cuando se realice el paso 2 de la metodología se requerirá el uso de métodos dinámicos, concretamente, en este trabajo se implementarán dos métodos distintos: análisis de componentes principales dinámico (DPCA) y análisis de variables canónicas (CVA).

Ahora veremos las propuestas que hacemos en cada uno de estos pasos.

4.3.1. Métodos de división de la planta basados en datos

Existen en la literatura diferentes métodos para dividir la planta en varios bloques basados solo en datos, como por ejemplo en [Ge and Song, 2013], que usa cada dirección de los componentes principales para la construcción de cada bloque, seleccionando las variables más relevantes. Otra idea es usar SPCA (*Sparse PCA*), donde nuevamente cada dirección de los componentes principales del modelo PCA se usa para seleccionar las variables, pero en este caso se obliga a que las variables no seleccionadas tengan valor nulo en el componente principal que corresponda al grupo en cuestión [Grbovic et al., 2012]. Otras ideas consisten en utilizar la correlación existente entre las variables [Tong and Shi, 2016, Tong et al., 2017], o la Información Mutua, que permite considerar las relaciones no-lineales entre las variables [Jiang and Yan, 2014], o, también, plantear la división de la planta como un problema de optimización [Jiang et al., 2016].

En esta tesis se propone usar otras técnicas de división de la planta basadas únicamente en el análisis de datos, como son:

- Técnicas basadas en regresión, en donde para cada una de las variables se hace un método de regresión y aquellas variables con coeficiente más alto en ese modelo se agrupará con la variable modelada para crear cada uno de los bloques.
- Técnicas basadas en información, que aportan información de la influencia que tienen entre sí las variables, y esto permite tomar decisiones acerca de qué variables deben ser agrupadas.
- Técnicas basadas en agrupamiento, en las que se considere diferente información de las variables, como pueda ser su media, desviación estándar, correlación, información mutua, etc. Con esto se consigue agrupar las variables que sean lo más similares posibles, y se lleva a cabo mediante conocidos métodos de agrupamiento como son DBSCAN (Agrupamiento Espacial basado en Densidad de Aplicaciones con Ruido) y EM (maximización de las expectativas).

Veamos ahora brevemente como se utilizan algunas de estas técnicas:

- **LASSO:** Se crea un modelo LASSO para cada variable, y, en el bloque que se vaya a crear para dicha variable, se la incluirá a ella misma junto a aquellas variables cuyos coeficientes de regresión sean no nulos en su correspondiente modelo LASSO. Se pueden hacer los modelos usando como predictores las variables sin retardo o incluir las variables con un cierto número de retardos haciendo que el método tenga carácter dinámico, en cuyo caso se denominará DLASSO.
- **SPLS:** Se creará un bloque por cada variable. Para cada bloque se implementará un modelo SPLS en el que la variable a la que corresponda dicho bloque se le considerará como variable respuesta (\mathbf{Y}), y el resto de variables serán las variables predictoras (\mathbf{X}). Una vez creado el modelo, se analizará qué variables poseen coeficientes no nulos en el vector de regresión \mathbf{B}_{PLS} y esas serán las que se incluyan en el bloque junto a la variable respuesta. Este proceso se repetirá con todas las variables (o bloques).
- **Red neuronal:** se entrena una red neuronal perceptrón multicapa para predecir cada variable, tomando como entradas las demás variables (predictoras o entradas). Las entradas que tengan un mayor influencia en la salida serán las que se elijan para formar parte de un grupo. Este procedimiento se propuso en [Olden and Jackson, 2002] para analizar la forma en que cada entrada de una red neuronal afecta a las salidas de la misma y comprender así el funcionamiento de la red. Si se utiliza la red neuronal para crear un modelo de una variable (salida) a partir de otras variables (entradas), el análisis de los pesos de la red nos permite conocer qué variables o entradas son más relevantes y, por tanto, es posible usar este método para hacer selección de características [de Oña and Garrido, 2014, Xue, 2019]. Esto se realiza analizando los pesos de las conexiones de la red: para cada entrada, se multiplican los pesos que afectan a dicha entrada a través de cada trayectoria y luego se suman los resultados de cada trayectoria que corresponda a esa entrada, de la siguiente forma:

$$R_i = \sum_{H=1}^k W_{ik} \dot{W}_k \quad (4.21)$$

donde R_i es la importancia relativa de la variable i en la salida de la red. H es el número de neuronas en la capa oculta, W_{ik} es el peso asignado a la conexión entre la neurona de entrada i y la neurona de la capa oculta k . Y W_k es el peso que corresponde a la conexión entre la neurona oculta k y la salida de la red.

- **Árboles aleatorios:** Se puede medir la importancia de cada variable en el modelo RF creado para predecir cada una de las variables. Para ello, durante la fase de entrenamiento se va midiendo el error *out-of-bag* [James et al., 2013] de cada observación y se hace un promedio

para todos los árboles. Para medir la importancia de una determinada variable, los valores de dicha variable se permutan para los datos de entrenamiento y se observa como cambia el error *out-of-bag* para estos datos de entrenamiento modificados. De este modo, la importancia de dicha variable es el promedio de la variación de dicho error entre el conjunto de datos original y el modificado. Este proceso se detalla en [Breiman, 2001]. De esta forma se seleccionan las variables que se incluirán en cada uno de los bloques.

- **SVR:** Se modela cada variable en función de las demás. Y a cada variable se le asigna un grupo construido a partir del modelo correspondiente a esa variable. En cada modelo se realiza un análisis de sensibilidad y aquellas variables que presenten valores más altos serán incluidas en el correspondiente grupo [Cortez and Embrechts, 2013].
- **DCCA:** Se calcula el valor de DCCA para cada par de variables y se agrupan aquellas que presenten un nivel elevado de DCCA, en valor absoluto.

4.3.2. Métodos locales de detección de fallos

En los métodos FDI en aproximaciones descentralizadas [Jiang and Yan, 2014, Tong and Shi, 2016, Jiang et al., 2016], el método usado para detectar fallos siempre ha sido el método PCA. Sin embargo, como se ha dicho en la introducción de este capítulo, dada la naturaleza dinámica de los datos recogidos de las plantas industriales, se requiere que el método usado para detectar fallos en cada uno de los bloques sea de tipo dinámico. Esto permitirá tener en consideración la influencia de los estados pasados de la planta en el estado actual de la misma, es decir, se analizará el carácter dinámico de la planta. Para ello en esta tesis se propone usar tanto el PCA dinámico como el CVA en los bloques.

4.3.3. Fusión de los resultados locales

La fusión de los resultados locales se hará para cada estadístico local ST usando el índice BIC. Se tomará un determinado valor de α en los datos de entreno y luego se reajustará en los datos de test (usando siempre datos sin fallo) de modo que no aparezcan falsas alarmas (es decir, que no haya un determinado número de valores de BIC_{ST} anómalos consecutivos).

4.4. Caso de estudio: Tennessee Eastman Process

Los métodos propuestos se han probado en la planta *Tennessee Eastman Process* (TEP) [Downs and Vogel, 1993], descrita en el Apéndice A.3. Se va

a usar la planta para poner a prueba los diferentes métodos de descentralización explicados antes y comprobar si consiguen dar un buen resultado cuando se utilizan para detectar los fallos de la planta. Se mostrarán datos de tiempos de retardos en la detección, tasas de no detección (MDR), fallos detectados, número de bloques y tasas de falsas alarmas (FAR). Hay que hacer notar que, como se ha dicho anteriormente, se necesita un cierto número de observaciones anómalas consecutivas para detectar el fallo, y esas observaciones no han sido incluidas en los resultados de las tablas mostradas a continuación. Por tanto, los valores mostrados aquí deben incrementarse en el número de observaciones anómalas establecidas para cada método (que se indicará en su momento), para tener el instante real en el que el sistema avisa de un fallo.

4.4.1. MB-DDPCA: DPCA distribuido multi-bloque

A continuación se presentan los resultados de las pruebas que se hicieron al hacer una detección de fallos descentralizada usando modelos PCA Dinámicos (DPCA) locales (en cada bloque).

4.4.1.1. Metodología experimental

En esta sección se muestran los parámetros utilizados para hacer la detección descentralizada aplicando varios de los métodos de descentralización que se han detallado antes este capítulo: LASSO, *Sparse* PLS, etc. El proceso seguido ha sido: normalizado de los datos, agrupado de las variables (se han hecho pruebas usando las variable sin retardo y, también, con retardo), creación de los modelos PCA en los bloques y fusión de los resultados locales mediante BIC. Cabe destacar que en esta comparación también se ha incluido la descentralización basada en *Sparse* PCA propuesta en [Grbovic et al., 2012] y la basada en Información mutua (MI), propuesta por [Jiang and Yan, 2014].

Todos los parámetros de los métodos se ajustaron de modo que se consiguiera el mayor número de fallos detectados, la mayor rapidez en la detección y la ausencia de falsas alarmas o, al menos, la aparición de las mismas en el menor número de tests realizados. De modo que, después de probar con distintas combinaciones de parámetros, las que dieron mejor resultado fueron las mostradas en la Tabla 4.1.

Para cada método se muestra la normalización inicial que se aplicó a los datos, en caso de que se aplicase alguna; el número de retardos aplicados a los datos para hacer la descentralización; los bloques obtenidos (y, en su caso, cuántas variables por bloque se impusieron); el número de retardos aplicados a los datos utilizados para hacer los DPCA locales; la varianza en dichos modelos locales; el número de observaciones anómalas consecutivas para detectar un fallo; y el valor del parámetro α del índice BIC.

Tabla 4.1: Parámetros de los métodos distribuidos basados en DPCA.

Método	Norm. datos	Datos para división en bloques	Bloques obtenidos	Datos para PCA locales	Varianza PCA locales	Observ. para fallo	BIC α
LASSO	Estándar	0 retardos	1 por variable	0 a 2 retardos	80 %	3	0.8
DLASSO	Estándar	0 a 4 retardos	1 por variable		80 %	5	0.99
SPLS	Estándar	0 retardos	1 por variable	0 a 3 retardos	60 %	5	0.9
SDPLS	Estándar	0 a 4 retardos	1 por variable		90 %	5	0.8
SPCA	Estándar	0 retardos	1 por variable (8 vars./ bloque)	0 a 2 retardos	90 %	6	0.9
NNET	Rango [0, 1]	0 retardos	1 por variable	0 a 2 retardos	60 %	3	0.9
SVR		0 retardos	1 por variable (20 vars./bloque)	0 a 2 retardos	75 %	5	0.9
RF		0 retardos	1 por variable (20 vars./bloque)	0 a 1 retardo	75 %	3	0.9
MI		0 retardos	1 por variable Min. 4 vars/bloque	0 a 3 retardos	75 %	4	0.9
DCCA	Estándar	0 retardos	1 por variable Min. 4 vars/bloque	0 a 4 retardos	90 %	5	0.8

Los métodos DLASSO y SDPLS realizan la descentralización usando variables con y sin retardo, de modo que en cada bloque ya van a quedar agrupadas variables retardadas y no retardadas, por tanto, los modelos FDI locales serán PCA, y no DPCA, al estar ya incluida la correlación temporal entre variables en la composición de los bloques. En el caso de los métodos MI y DCCA se eliminaron los bloques con menos de 4 y de 5 variables, respectivamente, y aquellas variables que perdieron su bloque se agruparon en un bloque extra.

4.4.1.2. Análisis de resultados

Para comprobar la efectividad de los métodos propuestos se ha comparado con los resultados obtenidos con el modelo DPCA descentralizado (DDPCA), basado en la correlación lineal, propuesto en [Tong and Shi, 2016] y con el DPCA centralizado mostrado en [Russell et al., 2000].

En las Tablas 4.2 y 4.3 se muestra el retardo, en instantes, de cada método en detectar cada uno de los 21 fallos. La primera tabla muestra los resultados para el estadístico T^2 y la segunda para el estadístico Q .

Tabla 4.2: Retardo en la detección (en instantes). T^2

Fallo	LASSO	DLASSO	Sparse PLS	Sparse DPLS	Sparse PCA	NNET	SVR	RF	MI	DCCA	DDPCA δ 1.5	Central DPCA
1	2	2	0	2	2	1	2	1	0	0	3	6
2	10	10	9	11	9	9	10	10	9	9	13	16
3	86	<i>nd</i>	81	84	<i>nd</i>	40	<i>nd</i>	85	<i>nd</i>	39	<i>nd</i>	<i>nd</i>
4	74	0	58	0	0	0	5	72	74	0	0	151
5	0	0	0	0	0	0	0	0	0	0	0	2
6	0	0	0	0	4	0	0	0	0	1	5	11
7	0	0	0	0	0	0	0	0	0	0	0	1
8	19	18	18	19	20	7	19	17	18	11	18	23
9	3	<i>nd</i>	0	1	<i>nd</i>	0	2	<i>nd</i>	2	<i>nd</i>	<i>nd</i>	<i>nd</i>
10	47	50	33	51	23	22	37	38	52	22	54	101
11	9	4	9	4	4	9	9	8	8	2	5	195
12	0	0	0	0	0	0	0	0	0	0	2	3
13	36	43	34	39	42	35	35	36	41	37	42	45
14	0	0	0	0	0	0	0	0	0	0	0	6
15	676	675	572	677	634	573	574	573	675	628	<i>nd</i>	<i>nd</i>
16	195	307	30	31	33	0	33	33	33	193	36	199
17	21	20	20	21	19	20	22	21	19	20	22	28
18	84	82	91	84	83	83	85	84	83	82	85	93
19	75	<i>nd</i>	<i>nd</i>	<i>nd</i>	180	8	<i>nd</i>	280	420	<i>nd</i>	17	<i>nd</i>
20	80	83	78	79	79	78	80	80	77	78	80	87
21	283	483	450	283	474	415	485	281	505	462	256	522

nd=no detectado

Tabla 4.3: Retardo en la detección (en instantes). Q

Fallo	LASSO	DLASSO	Sparse PLS	Sparse DPLS	Sparse PCA	NNET	SVR	RF	MI	DCCA	DDPCA δ 1.5	Central DPCA
1	0	0	0	0	1	0	0	0	0	0	1	5
2	10	9	7	8	13	10	9	7	7	8	18	13
3	566	<i>nd</i>	<i>nd</i>	42	<i>nd</i>	86	534	566	316	<i>nd</i>	<i>nd</i>	<i>nd</i>
4	0	0	0	0	0	0	0	0	0	0	0	2
5	0	0	0	0	0	0	0	0	0	0	0	2
6	0	0	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	0	0	0	1
8	16	13	12	16	15	16	15	16	15	13	16	21
9	728	<i>nd</i>	358	<i>nd</i>	<i>nd</i>	3	728	<i>nd</i>	359	<i>nd</i>	<i>nd</i>	<i>nd</i>
10	45	43	31	23	31	32	34	31	32	29	30	50
11	3	2	3	3	3	3	3	3	2	1	5	7
12	0	0	0	0	0	0	0	0	0	0	1	8
13	36	34	36	36	41	36	36	36	35	36	36	40
14	0	0	0	0	0	0	0	0	0	0	0	1
15	742	738	233	<i>nd</i>	<i>nd</i>	571	233	727	461	<i>nd</i>	<i>nd</i>	<i>nd</i>
16	13	31	13	12	13	16	192	11	15	11	11	196
17	18	17	17	17	17	18	18	16	16	15	19	24
18	15	78	77	14	76	78	12	78	76	72	17	84
19	43	57	7	9	111	79	8	11	7	24	12	82
20	78	77	78	78	79	78	78	78	73	73	74	84
21	283	281	254	263	482	265	253	264	253	517	240	286

nd=no detectado

Como se puede ver en las tablas, algunos fallos son muy fáciles de detectar y prácticamente todos los métodos los detectan instantáneamente, como los fallos 1, 5, 6, 7, 12 y 14. Mientras que otros fallos (el 3, el 9 y el 15) son muy difíciles y no todos pueden detectarlos, pero cabe destacar que LASSO, SPLS y las redes neuronales detectan los tres fallos con ambos estadísticos.

Por otro lado, en el caso del retardo en la detección usando el estadístico T^2 , el método basado en redes neuronales es el más efectivo, y es más rápido que el DDPCA en 15 fallos, e igual que él en 4 de ellos. Un poco por detrás está *Sparse* PLS que es más rápido que el DDPCA en 14 fallos y obtiene el mismo retardo en 4 fallos. Es mejor en 13 fallos el método basado en DCCA, mientras que los métodos *Sparse* PLS, RF y MI lo son en 12 fallos. Ligeramente por detrás quedan *Sparse* PCA y LASSO DPCA que son más rápidos que el DDPCA en 11 fallos. Por último se encuentran los métodos DLASSO y SVR que sólo son más rápidos que el DDPCA en 9 fallos, aunque dan el mismo tiempo de detección en 7 de los 21 fallos. En todos los fallos se comprueba que los métodos distribuidos obtienen mejores tiempos de detección que el DPCA centralizado (salvo DLASSO en el fallo 16).

Si se utiliza el estadístico Q , el método más rápido es el de información mutua (MI) ya que es más rápido que el DDPCA en 12 fallos, y da el mismo tiempo de detección en 5. Un poco peor es el basado en SVR que da mejores tiempos en 11 fallos e igual retardo en 6 fallos. Más atrás quedan el resto de métodos, ya que LASSO, *Sparse* PLS y *Sparse* DPLS son mejores que DDPCA en 9 fallos, e iguales en 7 fallos, los dos primeros, y en 9 el tercero. Un poco más lentos son DLASSO, Redes neuronales, RF y DCCA, e iguales en 7, 7, 9 y 10 fallos, respectivamente. A mucha distancia queda el método basado en *Sparse* PCA que sólo es más rápido que el DDPCA en 5 fallos, e igual en 9. Por otro lado, las propuestas distribuidas son más rápidas detectando los fallos que el DPCA centralizado, excepto algún método concreto en algunos fallos (*Sparse* PCA en los fallos 13 y 21 y DCCA en el 21).

Las Tablas 4.4 y 4.5 contienen los valores de tasa de no detección (MDR), en porcentaje, de cada uno de los métodos en cada uno de los 21 fallos. Los resultados aplicando el estadístico T^2 muestran que el método basado en redes neuronales (NNET) y el basado en DCCA dan el mejor resultado de detección, ya que obtienen el mejor valor de MDR en 14 fallos frente a los resultados obtenidos por el DDPCA, y son iguales a este en 2 fallos. Casi el mismo resultado se obtiene con el *Sparse* PCA, ya que es mejor que el DDPCA en 13 fallos y da el mismo resultado en 2 fallos. Siguiendo con la comparativa con el DDPCA, el método basado en *Sparse* DPLS es mejor en 11 fallos, el basado en información mutua (MI) en 10, LASSO, *Sparse* PLS y RF en 9 fallos, SVR es mejor en 8, y a cierta distancia queda DLASSO que sólo es mejor que el DDPCA en 6 fallos. Al igual que en el caso del retardo en la detección con T^2 , las tasas MDR con dicho estadístico para los métodos distribuidos son más bajas que las del DPCA centralizado, excepto para algún método en los fallos 4, 9, 10, 13, 15 y 19.

Tabla 4.4: Tasa de no detección de fallos (MDR), en porcentaje. T^2

Fallo	LASSO	DLASSO	Sparse PLS	Sparse DPLS	Sparse PCA	NNET	SVR	RF	MI	DCCA	DDPCA δ 1.5	Central DPCA
1	0.25	0.25	0	0.25	0.25	0.13	0.25	0.13	0	0	0.38	0.6
2	1.25	1.26	1.13	1.38	1.13	1.13	1.25	1.25	1.13	1.13	1.63	1.9
3	99.5	99.87	96.61	98.75	97.74	94.11	99.25	99.12	99.62	98.49	98.13	99.1
4	92.23	25.88	63.74	6.02	48.87	27.44	54.51	96.87	88.96	0	2.75	93.9
5	74.56	75.5	73.15	73.68	70.05	70.3	73.68	75.44	74.78	74.62	73.13	75.8
6	0	0	0	0	0.5	0	0	0	0	0.13	0.63	1.3
7	0	0	0	0	0	0	0	0	0	0	0	15.9
8	2.38	2.26	2.26	2.38	2.51	0.88	2.26	2.01	2.26	1.38	2.25	2.8
9	99.5	100	97.99	98.5	97.99	93.61	98.5	99.37	99	98.87	97.63	99.5
10	52.26	60.3	44.54	49.87	46.37	38.22	48.75	52.38	51.07	54.77	55.13	58
11	65.16	33.79	55.33	22.93	45.86	43.98	52.88	65.16	59.85	11.56	21.5	80.1
12	0.63	0.38	0.5	0.63	0.63	0.38	0.63	0.63	0.5	0.38	0.88	1
13	4.51	4.77	4.27	4.89	4.64	4.39	4.39	4.51	4.89	4.65	5.25	4.9
14	0	0	0	0	0	0	0	0	0	0	0	6.1
15	97.24	98.74	88.58	95.24	90.73	83.08	94.74	95.74	97.37	95.85	97.38	96.4
16	75.31	83.67	60.73	64.54	66.92	48.75	68.42	74.19	73.9	76.13	76.63	78.3
17	13.41	10.3	12.3	4.76	6.14	8.77	12.66	10.9	9.66	4.27	6.5	24
18	10.4	10.18	11.04	10.4	10.15	9.77	10.53	10.53	10.41	10.3	10.5	11.1
19	97.12	99.5	99.75	97.24	84.34	97.74	96.49	99.25	92.6	98.62	63.25	99.3
20	52.13	57.79	47.3	45.24	44.74	42.11	52.13	47.74	46.42	40.08	46.75	64.4
21	60.15	60.68	55.83	56.52	57.39	55.51	60.03	57.52	62.36	60.05	46.75	64.4

Tabla 4.5: Tasa de no detección de fallos (MDR), en porcentaje. Q

Fallo	LASSO	DLASSO	Sparse PLS	Sparse DPLS	Sparse PCA	NNET	SVR	RF	MI	DCCA	DDPCA δ 1.5	Central DPCA
1	0	0	0	0	0.13	0	0	0	0	0	0.13	0.5
2	1.25	1.13	0.75	0.88	1.63	1.25	1	0.88	0.88	1.01	2.63	1.5
3	98.37	98.74	96.49	94.86	97.99	98.5	92.73	97.37	97.74	98.62	96.13	99
4	0	0	0	0	0	0	0	0	0	0	0	0
5	21.3	0	73.53	59.9	78.95	75.56	63.78	74.31	71.39	75	0	74.8
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	2.01	1.63	1.51	2.01	2.63	2.01	1.75	2.38	1.88	2.01	2.38	2.5
9	98.12	97.86	97.11	95.11	98.75	98.37	92.98	97.62	96.99	99.25	94.88	99.4
10	61.53	52.64	36.01	27.57	59.65	38.72	47.24	40.73	47.93	45.85	26.13	66.5
11	11.53	9.17	10.16	5.26	8.65	8.9	8.27	8.02	3.76	4.4	4.38	19.3
12	0.88	0	0.38	0.25	1.38	0.5	0.5	2.01	0.5	0.63	0.13	2.4
13	4.51	4.27	4.52	4.51	5.14	4.76	4.51	4.51	4.39	6.91	4.38	19.3
14	0	0	0	0	0	0	0	0	0	0	0	2.4
15	98.37	97.86	90.46	92.48	97.74	96.24	91.35	95.74	95.11	99.12	92.75	97.6
16	67.54	59.92	50.06	41.6	58.9	54.39	68.05	41.98	54.83	44.6	27.88	70.8
17	2.76	2.51	2.38	2.13	2.76	2.88	2.76	2.13	2.01	1.88	2.38	5.3
18	9.4	9.55	9.16	8.77	9.52	9.65	8.65	9.77	9.16	9.05	9.13	10
19	63.28	67.46	31.74	20.30	49.12	75.31	28.32	43.23	25.22	24.12	32.88	73.5
20	35.96	25.75	33	28.32	44.11	38.97	31.08	34.71	31.49	35.43	20.25	49
21	53.88	53.77	51.82	49.87	57.27	47.49	47.87	55.26	47.43	73.74	40.75	55.8

Cuando se usa Q es el método *Sparse* DPLS y SVR los que dan el mejor resultado, ya que proporciona tasas más bajas de MDR en 8 de los 21 fallos frente al DDPCA, siendo iguales a éste en 4 fallos. El resto de métodos quedan a bastante distancia, ya que el basado en MI y el basado en DDCA son mejores en 6 fallos e iguales en 4; LASSO dinámico y el *Sparse* PLS son mejores en 5 fallos e iguales en otros 5; LASSO, las redes neuronales y RF son mejores que el DDPCA en 3 fallos, e iguales en 4, 4 y 5 fallos, respectivamente. En la comparativa de los métodos descentralizados frente al centralizado, aquellos salen ganando ya que dan tasas MDR más bajas excepto en los fallos 2, 5, 8, 15 y 21, en los que algún método da tasas ligeramente por encima del DPCA centralizado.

Finalmente, la Tabla 4.6 muestra el número de fallos detectados, las tasas de falsas alarmas (FAR), la presencia o no de falsas alarmas en alguna simulación y el número de bloques que generó cada método.

Se observa que todos los métodos propuestos presentan tasas de falsas alarmas FAR inferiores al DDPCA de [Tong and Shi, 2016]. La mayoría de métodos propuestos han sido capaces de detectar mayor número de fallos, destacando LASSO con T^2 y Q , las redes neuronales con T^2 y Q , SVR con Q y MI con Q que han sido capaces de detectar todos los fallos. También dan buen resultado *Sparse* PLS con T^2 y Q , *Sparse* DPLS con T^2 , RF con T^2 y Q y MI con T^2 , ya que son capaces de detectar 20 de los 21 fallos.

En cuanto a la presencia de falsos positivos, ocurrieron en los métodos LASSO, *Sparse* PLS y redes neuronales, y siempre con el estadístico T^2 . Usando Q ningún método detectó fallos al estar la planta en condiciones normales de operación. Cabe resaltar que el método *Sparse* PCA consigue trabajar con un número menor de bloques que los demás métodos, simplificando el método de detección.

Por tanto, los resultados muestran que los métodos descentralizados propuestos funcionan bien y dan resultados comparables o incluso mejores que otros métodos descentralizados, y, por supuesto, son más rápidos y sensibles en la detección que el método centralizado. Concretamente se observa que se obtienen mejores resultados sobre todo al trabajar con T^2 , dando a entender que la descentralización y los modelos de detección aplicados han sido capaces de modelar mejor la planta. De los métodos probados, los más eficaces con T^2 han sido: el basado en redes neuronales, *Sparse* PLS, DCCA y *Sparse* PCA, y, con Q : *Sparse* DPLS, SVR y MI; ya que son los más rápidos en la detección y son más sensibles frente a observaciones anómalas.

Tabla 4.6: Bloques, FAR y falsas alarmas.

	LASSO	DLASSO	Sparse PLS	Sparse DPLS	Sparse PCA	NNET	SVR	RF	MI	DCCA	DDPCA δ 1.5	Central DPCA
FAR T^2	0	0	0	0	0	0.2004	0	0	0	0	0.63	0.6
FAR Q	0	0	0.2004	0	0	0	0	0	0.2004	0	1.77	28.1
Fallos d. T^2	21	18	20	20	19	21	19	20	20	19	18	16
Fallos d. Q	21	19	20	19	18	21	21	20	21	18	18	18
Falsas a. T^2	Sí	No	Sí	No	No	Sí	No	No	No	No		
Falsas a. Q	No	No	No	No	No	No	No	No	No	No		
Bloques	52	52	52	52	38	52	52	52	42	52	52	1

4.4.2. MB-DDPCA, sensibilidad al grado de distribución

Aquí se van a mostrar los resultados de aplicar varios métodos de descentralización que generen modelos con diferente número de bloques seguidos por la implantación de modelos DPCA en cada uno de los bloques obtenidos. Se pretende comprobar el efecto que tiene usar un número mayor o menor de bloques, y si sigue siendo efectivo usar un descentralización con pocos bloques.

4.4.2.1. Metodología experimental

Igual que en el caso anterior se han probado varios métodos, cuyos parámetros han sido ajustados hasta dar con el mejor método posible. Los parámetros aplicados para realizar la detección de fallos con cada uno de los métodos se muestran en la Tabla 4.7. Como antes, los métodos propuestos se compararon también con la descentralización basada en correlación propuesta en [Tong and Shi, 2016] pero modificando dicha propuesta disminuyendo el número de bloques utilizados, y nuevamente con el método basado en información mutua [Jiang and Yan, 2014], modificando el método para considerar bloques solapados.

Tabla 4.7: Parámetros de los métodos distribuidos basados en DPCA.

Método	Normalizado datos	Datos para división en bloques	Bloques obtenidos	Datos para PCA locales	Varianza PCA locales	Observ. para fallo	BIC α
NNET	Rango [0, 1]	0 retardos	1 por variable	0 a 2 retardos	60 %	3	0.9
MI solapado		0 retardos	Menos bloques que variables	0 a 2 retardos	60 %	4	0.8
MI no solapado		0 retardos	Menos bloques que variables	0 a 2 retardos	90 %	4	0.9
Correlación solapado	Estándar	0 retardos	Se eliminan bloques < 4	0 a 4 retardos	60 %	5	0.9
Correlación no solapado	Estándar	0 retardos	Menos bloques que variables	0 a 3 retardos	90 %	4	0.9
DCCA	Estándar	0 retardos	Menos bloques que variables	0 a 2 retardos	90 %	3	0.9
DBSCAN		Kurtosis, <i>skewness</i> , media, varianza, etc.	Menos bloques que variables	0 a 4 retardos	60 %	5	0.99
EM Clust.		Kurtosis, <i>skewness</i> , media, varianza, etc.	Menos bloques que variables	0 a 2 retardos	75 %	5	0.9

Se comentan ahora algunos detalles específicos de cada método:

Redes Neuronales (NNET): Los parámetros de este método son los mismos que los usados en el apartado anterior.

Información mutua (MI): Una vez obtenida la matriz de MI, se crea un bloque por variable, analizando cada fila de la matriz, y comprobando qué elementos superaban el umbral teórico. Este método se aplicó de dos formas: con y sin solapamiento entre los bloques.

- *MI* con solapamiento (12 bloques): se parte de la lista completa de variables y se realiza el siguiente proceso con cada grupo:

1. Agrupar la primera variable de la lista con aquellas con las que tenga un valor MI superior al umbral.
2. Umbral: mediana de la fila correspondiente multiplicada por 1,5.
3. Se eliminan de la lista las variables incluidas en este grupo.

Se repite este proceso hasta que no queden variables en la lista. Por tanto, al crearse cada bloque usando la filas completas de la matriz MI, es posible que haya variables que estén en varios grupos simultáneamente.

- *MI* sin solapamiento (5 bloques): Cada variable quedará agrupada con aquellas con las que tenga un mayor índice MI. Por tanto, no habrá variables que estén en más de un bloque.

Correlacion: En este caso se parte de la matriz de correlación entre variables y se crean los bloques analizando los valores de correlación entre ellas. Se han probado dos planteamientos: con bloques solapados y con bloques sin solapar.

- Con solapamiento: El procedimiento es igual al considerado para la MI con solapamiento, pero usando la matriz de correlación y un umbral de 0,2. Los bloques con menos de 4 variables se eliminan, y las variables de los bloques eliminados se incluyen en un bloque extra.
- Sin solapamiento: El método es el mismo que el explicado para el método de información mutua sin solapamiento, pero usando la matriz de correlación y estableciendo un umbral de 0,1 que deben superar las variables para entrar a un bloque.

DCCA: El método aplicado es el mismo que el de Correlación con solapamiento, pero usando la matriz de DCCA en vez de la de correlación. Para crear el grupo de un variable se tomó la fila de la matriz DCCA que correspondía a esa variable y el umbral para decidir si una variable entraba en un bloque se ajustó tomando la mediana de los valores de esa fila multiplicada por 1,5.

Clustering (DBSCAN): En este caso se creó una matriz de datos con los valores de kurtosis, *skewness*, los valores medios y de varianza de la correlación y de la información mutua de cada una de las variables. El *Clustering* aplicado fue del tipo DBSCAN con un valor de 0,8 para el parámetro ϵ . Cada uno de los *clusters* creados se utilizó para generar un bloque.

EM Cluster: Agrupamiento *EM Cluster*: Se hizo un agrupamiento de variables a partir de los datos para cada variable de: kurtosis, *skewness*, media y varianza de correlación, media y varianza de información mutua. Después de varias pruebas, el número de clústers más eficaz para la detección de fallos fue de 3. Estos clústers sirvieron para crear los grupos.

4.4.2.2. Análisis de resultados

En este apartado se busca comprobar el efecto del número de bloques de la descentralización, probando métodos que trabajen con distintos tamaños de bloques. Para comparar el desempeño de estas propuestas se considerarán los resultados del DPCA centralizado propuesto por [Russell et al., 2000] para comprobar si estos métodos con descentralizaciones de menos bloques son preferibles a la aplicación del método centralizado. También se incluye un método de descentralización total (un bloque por variable) como el basado en redes neuronales para comprobar si se pierde rendimiento al reducir el número de bloques.

Las Tablas 4.8 y 4.9 contienen los retardos en la detección para cada método en cada uno de los fallos.

Los resultados del retardo en la detección usando T^2 muestran que el método basado en redes neuronales es el mejor, ya que es más rápido que el DPCA centralizado en los 21 fallos considerados. Aunque hay que destacar que la descentralización basada en DCCA también es capaz de mejorar los tiempos de detección del DPCA centralizado en los 21 fallos. El DCCA es un poco más lento que el método de redes neuronales, pero, hay que tener en cuenta que dispone de un menor número de bloques. El resto de métodos descentralizados siguen siendo mejores que el centralizado: el MI de 12 bloques es más rápido en 20 fallos e igual en el fallo restante; y el MI de 5 bloques y el de Correlación de 7 bloques son mejores en 19 fallos e iguales en los otros 2 fallos; y el EM *Cluster* es más rápido en 16 fallos e igual 2. DBSCAN mejora al método centralizado en 18 fallos y lo iguala en 1. Finalmente, el método de Correlación con 3 bloques es capaz de detectar más rápido que el método centralizado en 16 fallos y da el mismo tiempo en 3 de ellos.

Al observar los resultados al usar el estadístico Q las conclusiones que se pueden sacar son similares a las de T^2 , ya que todos los métodos funcionan mejor que el método centralizado. El método basado en redes neuronales es mejor que el centralizado en 21 fallos, y un poco más atrás queda el MI de 12 bloques que es más rápido en 18 fallos e igual en los otros 2. DBSCAN es mejor en 19 fallos e igual al centralizado en los 2 restantes. El método de Correlación con 7 bloques es más rápido en 19 fallos e igual en 1; DCCA destaca en 15 fallos y es igual al centralizado en 3 fallos; Correlación con 3 bloques y EM *Cluster* son mejores en 15 fallos e iguales en 3 fallos; finalmente, el MI con 5 bloques es más rápido que el centralizado en 14 fallos e igual en 3 fallos. Por tanto, aunque los resultados empeoran un poco respecto al estadístico T^2 siguen funcionando mejor que el método DPCA centralizado.

Tabla 4.8: Retardo en la detección (en instantes). T^2

Fallo	NNET	MI (12 bl.)	MI (5 bl.)	Corr. (7 bl.)	Corr. (3 bl.)	DCCA	DBSCAN	EM <i>Cluster</i>	DPCA central.
1	1	3	0	2	3	1	3	6	6
2	9	10	14	9	14	10	8	13	16
3	40	399	<i>nd</i>	<i>nd</i>	<i>nd</i>	40	80	84	<i>nd</i>
4	0	148	0	10	<i>nd</i>	0	144	<i>nd</i>	151
5	0	0	0	0	0	0	0	6	2
6	0	0	4	2	11	4	3	4	11
7	0	0	0	0	0	0	0	0	1
8	7	20	17	20	22	19	20	22	23
9	0	3	673	0	229	729	2	3	<i>nd</i>
10	22	52	91	24	70	49	56	66	101
11	9	9	4	2	47	4	5	45	195
12	0	0	0	0	2	0	0	5	3
13	35	44	41	39	42	40	44	44	45
14	0	0	0	0	0	0	0	0	6
15	573	674	676	671	769	676	671	771	<i>nd</i>
16	0	192	194	187	196	195	304	33	199
17	20	22	19	23	27	22	35	19	28
18	83	85	85	91	91	82	84	91	93
19	8	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	75	<i>nd</i>	<i>nd</i>	<i>nd</i>
20	78	79	83	79	83	76	80	80	87
21	415	462	255	278	529	453	514	508	522

nd=no detectado

Tabla 4.9: Retardo en la detección (en instantes). Q

Fallo	NNET	MI (12 bl.)	MI (5 bl.)	Corr. (7 bl.)	Corr. (3 bl.)	DCCA	DBSCAN	EM <i>Cluster</i>	DPCA central.
1	0	0	2	0	1	2	0	0	5
2	10	9	20	12	20	10	12	12	13
3	86	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>
4	0	0	0	0	0	0	0	0	2
5	0	0	0	0	0	0	0	0	2
6	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	1
8	16	18	19	15	17	18	16	17	21
9	3	<i>nd</i>	<i>nd</i>	682	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>
10	32	37	33	31	20	47	44	45	50
11	3	3	4	1	3	4	2	4	7
12	0	0	0	0	0	0	0	5	8
13	36	36	44	37	44	44	33	39	40
14	0	0	0	0	0	0	0	0	1
15	571	700	<i>nd</i>	666	<i>nd</i>	<i>nd</i>	572	716	<i>nd</i>
16	16	192	13	13	7	16	33	32	196
17	18	19	26	18	19	18	19	27	24
18	78	78	75	13	81	77	77	82	84
19	79	182	44	16	9	112	79	183	82
20	78	80	74	77	73	74	78	84	84
21	265	254	495	480	380	520	281	445	286

nd=no detectado

Las Tablas 4.10 y 4.11 muestran los resultados de la tasa de no detección (MDR) para cada fallo.

Tabla 4.10: Tasa de no detección de fallos (MDR), en porcentaje. T^2

Fallo	NNET	MI (12 bl.)	MI (5 bl.)	Corr. (7 bl.)	Corr. (3 bl.)	DCCA	DBSCAN	EM <i>Cluster</i>	DPCA central.
1	0.13	0.38	0	0.25	0.13	0.13	0.38	0.75	0.6
2	1.13	1.25	1.75	1.13	1.76	1.25	1.01	1.63	1.9
3	94.11	98.62	99.62	99.37	99.87	98.50	98.24	97.37	99.1
4	27.44	95.36	73.06	74.12	100.00	12.28	85.43	99.75	93.9
5	70.30	74.94	76.94	75.00	79.05	75.31	75.38	76.69	75.8
6	0	0	0.50	0.25	1.38	0.50	0.38	0.50	1.3
7	0	0	0	8.29	0	0	0	0	15.9
8	0.88	2.51	2.01	2.51	3.26	2.13	2.51	2.76	2.8
9	93.61	99.00	98.37	98.49	99.50	99.50	99.12	99.12	99.5
10	38.22	55.26	64.91	52.14	67.13	57.52	58.29	59.27	58
11	43.98	72.06	48.12	45.98	94.48	38.97	56.66	86.47	80.1
12	0.38	0.63	0.63	0.63	2.51	0.63	0.50	1.13	1
13	4.39	5.51	5.14	4.90	5.27	5.01	5.40	5.51	4.9
14	0	0.25	0	0	0	0	0	0	6.1
15	83.08	96.49	97.87	95.10	97.99	97.99	96.61	96.87	96.4
16	48.75	73.81	83.58	72.11	86.07	79.57	78.02	74.81	78.3
17	8.77	14.91	5.26	23.24	34.00	14.16	38.32	11.65	24
18	9.77	10.65	10.65	11.06	11.42	10.28	10.55	10.90	11.1
19	97.74	99.62	98.50	99.12	100.00	88.72	99.75	99.87	99.6
20	42.11	57.14	50.00	62.69	77.79	39.47	59.80	66.04	64.4
21	55.51	60.65	55.76	56.91	67.50	58.02	67.71	66.04	64.4

Tabla 4.11: Tasa de no detección de fallos (MDR), en porcentaje. \mathcal{Q}

Fallo	NNET	MI (12 bl.)	MI (5 bl.)	Corr. (7 bl.)	Corr. (3 bl.)	DCCA	DBSCAN	EM <i>Cluster</i>	DPCA central.
1	0	0	0.25	0	0.13	0.25	0	0	0.5
2	1.25	1.13	2.13	1.51	2.38	1.25	1.51	1.50	1.5
3	98.50	99.00	98.37	97.36	100.00	99.50	98.99	99.12	99
4	0	0	0	0	0	0	0	0	0
5	75.56	76.19	73.56	72.24	78.54	82.71	66.96	76.44	74.8
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	6.52	0	0	0
8	2.01	2.26	2.51	1.88	4.52	4.51	2.51	2.76	2.5
9	98.37	98.50	96.87	96.36	99.87	99.87	99.50	99.50	99.4
10	38.72	57.89	54.14	54.27	24.97	72.18	59.30	60.28	66.5
11	8.90	9.40	21.43	2.76	20.20	11.15	13.44	17.04	19.3
12	0.50	0.63	4.26	0.38	2.26	3.38	0.63	2.76	2.4
13	4.76	4.76	5.26	5.78	8.28	6.52	4.15	4.76	4.9
14	0	0	0.25	0	0	0.13	0	0	0
15	96.24	98.37	97.99	96.61	100.00	99.37	95.73	97.74	97.6
16	54.39	84.46	55.26	55.40	19.95	72.18	60.05	61.53	70.8
17	2.88	2.76	14.29	2.89	4.77	3.13	3.89	21.68	5.3
18	9.65	9.65	9.15	8.92	9.91	9.52	9.42	10.03	10
19	75.31	73.43	76.32	58.79	99.50	83.33	72.99	92.23	73.5
20	38.97	43.11	39.35	40.08	42.66	42.36	46.23	47.49	49
21	47.49	55.64	60.65	58.04	55.21	71.18	52.39	57.14	55.8

Los resultados de la tasa MDR muestran unos resultados parecidos a los de los tiempos de detección. Usando T^2 como estadístico de detección, el

método basado en redes neuronales es el mejor, ya que ofrece las mejores tasas MDR en los 21 fallos. El método de Correlación con 7 bloques también consigue mejorar los resultados del centralizado en 19 fallos e igualarlo en 1. Ligeramente por detrás quedan los métodos MI de 12 bloques y el DCCA, que dan mejor resultado que el centralizado en 17 fallos y el MI de 5 bloques que es mejor en 15 fallos. DBSCAN es mejor en 15 y EM *Cluster* en 10 fallos. En cambio el método de Correlación con 3 bloques sólo es mejor en 4 fallos e igual en 1.

Al usar el estadístico Q el método basado en redes neuronales sigue siendo el mejor: da mejores valores de MDR en 15 fallos y iguala al centralizado en 4. El método de Correlación de 7 bloques y DBSCAN son mejores en 14 fallos e iguales en 4 fallos. En un nivel similar esta MI con 12 bloques que es mejor en 14 fallos e igual en otros 5; MI con 5 bloques consigue mejores tasas MDR en 8 fallos y los mismos valores en 3; Correlación de 3 bloques se impone en 8 fallos y empata en 4; DCCA es superior en 6 e igual en 2; y, finalmente, EM *Cluster* es mejor en 6, y da los mismo valores en 5 fallos.

En la Tabla 4.12 se muestran el número de bloques de cada método, y las tasas FAR, falsas alarmas y número de fallos detectados de cada uno.

Tabla 4.12: FAR (en porcentaje), fallos detectados, falsas alarmas y número de bloques.

	NNET	MI (12 bl.)	MI (5 bl.)	Corr. (7 bl.)	Corr. (3 bl.)	DCCA	DBSCAN	EM <i>Cluster</i>	DPCA central.
FAR T^2	0.2004	0	0.40	0	0	0.2004	0	0	1.6
FAR Q	0	0	0	0	0	0	0	0	28.1
Fallos d. T^2	21	20	19	19	18	21	20	19	16
Fallos d. Q	21	19	18	20	18	18	19	19	18
Falsas a. T^2	Sí	No	No	No	No	No	No	No	No
Falsas a. Q	No	No	No	No	No	No	No	No	No
Bloques	52	12	5	7	3	9	4	3	1

Todos los métodos propuestos presentan menores valores de falsas alarmas que el método centralizado y son capaces de detectar más fallos que él en casi todos los casos. Destacan, con T^2 , el basado en redes neuronales y el DCCA, que detectan los 21 fallos, además de MI con 12 bloques y DBSCAN, que consiguen detectar 20 fallos. Usando Q , las redes neuronales pueden detectar los 21 fallos, mientras que el método de Correlación de 7 bloques sólo deja un fallo sin detectar. Además, sólo aparecieron falsas alarmas en algún fallo con el método de redes neuronales usando T^2 . Y en cuanto al número de bloques, las redes neuronales, al ser una descentralización completa, dispone de un bloque por variable, pero los demás métodos dan descentralizaciones bastante reducidas, desde los 12 bloques con el método basado en MI hasta los 3 de Correlación y de EM *Cluster*, pasando por los 9 del DCCA, los 7 de la otra variante del de Correlación, los 5 del MI y los 4 del DBSCAN.

En resumen, se comprueba que los métodos descentralizados propuestos

son más rápidos detectando fallos que el método centralizado, independientemente de usar mayores o menores grados de descentralización. Destacando el DBSCAN, que con sólo 4 bloques, es capaz de obtener unos tiempos de detección muy buenos. En cuanto a las tasas MDR los resultados empeoran un poco con T^2 pero siguen siendo muy superiores a los del DPCA centralizado. Con Q bajan un poco los resultados, pero salvo DCCA y EM *Cluster*, el resto de métodos siguen dando mejores resultados que el DPCA de [Russell et al., 2000]. Además, todos los métodos propuestos son capaces detectar igual o mayor número de fallos que el método centralizado.

Por tanto, existen alternativas descentralizadas con diferente número de bloques que son más eficaces que un método centralizado, y esto permite aplicar la descentralización a plantas, en las que, por su complejidad, no sea viable crear un grupo por cada variable.

4.4.3. MB-DCVA: CVA distribuido multi-bloque

Ahora se muestran los resultados de implementar modelos descentralizados que cuenten con modelos de detección CVA en los bloques obtenidos después de la descentralización. Para realizar la división en grupos de variables no se hará juntando aquellas más correlacionadas, como se hizo con DPCA, sino que se van agrupar aquellas que tengan menor correlación. Esto se hace así ya que el método CVA lo que hace es crear modelos en espacio de estados y, para modelar un sistema hay que trabajar con aquellas variables que aporten mayor cantidad de información diferente al modelo.

4.4.3.1. Metodología experimental

De igual forma que en la pruebas anteriores se han elegido varios métodos de descentralización y se han ajustado sus respectivos parámetros para conseguir la máxima eficacia en términos de número de fallos detectados, ausencia de falsas alarmas y menor retardo en la detección.

Los datos utilizados para cada método se muestran en la Tabla 4.13. En dicha Tabla se indica la normalización inicial que se aplicó a los datos, en caso de que se aplicase alguna; los bloques obtenidos; el orden usado para los CVA locales; la varianza en dichos modelos locales; el número de observaciones anómalas consecutivas para detectar un fallo; y el valor del parámetro α del índice BIC.

En todos los modelos de regresión, LASSO, Sparse PLS, Sparse PCA, RF, SVR, se modeló cada variable en base a las demás variables para crear cada grupo, pero ahora se eligen las variables que tienen el coeficiente de regresión igual a cero. Así se consigue agrupar variables con poca correlación.

Tabla 4.13: Parámetros de los métodos distribuidos basados en CVA.

Método	Norm. datos	Bloques	Orden CVA	Varianza en CVA locales	Observaciones para fallo	BIC α
LASSO	Estándar	1 por variable	4	90%	5	0.9
<i>Sparse PLS</i>	Estándar	1 por variable	4	90%	5	0.9
<i>Sparse PCA</i>	Estándar	Cardinalidad=40	3	90%	3	0.8
NNET	Rango [0, 1]	1 por variable	5	90%	5	0.9
RF		1 por variable	3	75%	3	0.9
SVR		1 por variable	5	90%	4	0.8
Correlación	Estándar	1 por variable	3	90%	3	0.9
DCCA	Estándar	1 por variable	5	60%	5	0.9
MI		Se eliminan bloques < 4	3	90%	5	0.9

Para las redes neuronales se modeló una red neuronal para cada variable, probándose con diferentes configuraciones de la capa oculta de la red. Para ello se hizo variar el número de neuronas en esta capa entre **1** y el número de entradas de la red, seleccionándose aquella configuración que diese el menor valor de error rMSE. En este modelo se estudió la relevancia de cada variable tal como se comentó anteriormente en este capítulo y se estableció un umbral de relevancia, seleccionando las variables por debajo de él. Este umbral se ajustó tomando la media de los valores mínimo y máximo de los valores de relevancia en la red de las variables. De esta forma se seleccionaron las variables menos relevantes en cada caso. Esto se repitió para cada variable de cara a crear los distintos grupos.

Para los métodos basados en información: correlación, DCCA e información mutua, la forma de aplicarlos para hacer la descentralización ha sido la misma que para el caso del DPCA distribuido, es decir, crear un bloque para cada variable y analizar los valores de correlación, DCCA o información mutua, respectivamente, entre las variables para decidir si deben ser incluidas o no en un bloque. Aquí la diferencia es que una variable se incorpora a un bloque si su valor de correlación, DCCA o información mutua con la variable a la que pertenece dicho bloque es inferior a un umbral. Este umbral, después de varias pruebas, se ha ajustado de la siguiente forma: para la correlación se toman los valores de correlación de la variable propietaria del bloque con las demás, se calcula la mediana y se multiplica por 0,3. Para el DCCA se toman los valores de DCCA de la variable cuyo grupo se va a construir con las demás variables, se calcula la mediana y se multiplica por 0,75. Para la información mutua el umbral es 0,05. Adicionalmente, en el caso de la información mutua, si alguna variable tiene un bloque de tamaño inferior a 4 variables se elimina su bloque, y estas variables pasan a un grupo extra. Si esto ocurre, habrá más variables que bloques.

4.4.3.2. Análisis de resultados

Los resultados de las pruebas realizadas, que se muestran a continuación, incluyen datos de tasa MDR, retardo en la detección, tasa FAR, número de

bloques, y fallos detectados. Para ver la efectividad de las propuestas se las ha comparado con un modelo de detección centralizado basado en CVA y cuyos resultados aparecen en [Russell et al., 2000]. En las tablas 4.14, 4.15 y 4.16 se muestran los instantes de retraso con los que cada método ha detectado cada fallo, usando los tres estadísticos del CVA.

Tabla 4.14: Retardo en la detección (en instantes). T_s^2

Fallo	LASSO	Sparse PLS	Sparse PCA	NNET	RF	SVR	Correlación	DCCA	MI	Central CVA
1	3	3	3	2	7	3	2	4	3	2
2	13	13	15	9	20	17	15	15	13	13
3	43	nd	45	nd	621	nd	51	41	51	nd
4	1	1	1	1	1	1	1	1	1	462
5	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	18	20	19	17	22	18	17	19	20	20
9	5	nd	6	nd	23	678	731	NA	nd	nd
10	22	23	21	22	45	55	21	21	21	25
11	7	7	7	7	8	10	7	7	7	292
12	2	2	3	2	2	3	3	3	2	2
13	36	37	39	32	49	47	37	39	39	42
14	1	2	1	1	2	1	1	1	1	2
15	577	635	241	3	239	241	241	575	679	677
16	9	9	10	9	15	101	9	9	10	14
17	19	20	23	19	24	35	20	21	23	27
18	77	78	76	76	85	85	79	79	81	83
19	3	12	11	3	13	33	2	11	11	nd
20	71	71	66	67	80	85	65	76	75	82
21	255	300	285	207	519	336	286	255	272	273

nd = no detectado

Usando T_s^2 el método NNET detecta más rápido que el CVA centralizado en 14 fallos y obtiene el mismo resultado en los 7 restantes. LASSO es más

rápido en 15 fallos y da el mismo tiempo en 5 fallos. Junto a ellos también están el basado en Correlación que es más rápido que el CVA centralizado en 14 fallos e igual en 4, y el *Sparse* PCA que también es más rápido en 14 fallos y es igual en 3. Siguiendo con la comparativa con el CVA de [Russell et al., 2000], MI es más rápido en 12 fallos e igual en 7; *Sparse* PLS mejora en 10 e iguala en 9 fallos; DCCA gana en 14 fallos, y es igual en 3. Por debajo ya quedan RF, que gana al centralizado en 8 y lo iguala en 5 fallos; y SVR que es más rápido en 7 fallos y empata en otros 4.

Tabla 4.15: Retardo en la detección (en instantes). T_r^2

Fallo	LASSO	<i>Sparse</i> PLS	<i>Sparse</i> PCA	NNET	RF	SVR	Correlación	DCCA	MI	Central CVA
1	2	2	2	2	2	4	2	2	3	3
2	11	11	16	12	13	17	15	16	15	15
3	nd	nd	nd	nd	nd	nd	nd	54	nd	nd
4	1	1	1	1	1	1	1	1	1	1
5	1	1	2	1	1	2	2	1	1	1
6	1	1	1	1	1	2	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	17	18	20	18	19	20	16	19	19	20
9	nd	nd	nd	nd	732	nd	nd	nd	nd	nd
10	22	22	25	22	24	506	21	23	24	23
11	7	7	6	7	6	8	6	7	7	11
12	2	2	3	2	2	3	2	2	2	2
13	35	35	39	34	39	48	36	38	41	39
14	1	1	1	2	1	2	1	1	1	1
15	643	643	nd	3	662	nd	nd	746	nd	nd
16	9	9	9	10	11	629	8	8	10	9
17	20	20	23	19	20	34	20	20	23	20
18	76	77	80	76	79	91	76	80	79	79
19	2	3	2	3	2	20	2	11	11	11
20	66	67	65	66	70	69	65	67	72	66
21	257	226	286	287	286	539	286	286	479	511

nd = no detectado

Si se utiliza T_r^2 , y siguiendo la comparativa de los métodos propuestos frente al CVA centralizado de [Russell et al., 2000], el método con mejores

resultados de retardo en la detección es el basado en redes neuronales que consigue mejores tiempos en 11 fallos y empata con el CVA en 8. A un nivel similar está LASSO que gana en 10 fallos y empata en otros 11, y Correlación y *Sparse* PLS que son superiores en 10 fallos, igualando al CVA centralizado en 10 fallos. A cierta distancia queda el método de bosques aleatorios (RF) que gana en 8 y empata en 10 fallos. Por último quedan DCCA que es mejor en 8 fallos e iguala tiempos con el centralizado en 9; *Sparse* PCA, que gana en 5 fallos y empata en 10; y, finalmente, MI que iguala al CVA centralizado en 13 fallos, pero sólo lo supera en 3. Por tanto, se observa que los métodos han perdido efectividad con T_r^2 con respecto a T_s^2 .

Tabla 4.16: Retardo en la detección (en instantes). \mathcal{Q}

Fallo	LASSO	<i>Sparse</i> PLS	<i>Sparse</i> PCA	NNET	RF	SVR	Correlacion	DCCA	MI	Central CVA
1	2	3	8	4	5	7	3	4	3	2
2	14	14	25	19	17	17	19	19	19	25
3	nd	nd	nd	nd	nd	nd	nd	nd	nd	nd
4	1	1	1	249	1	1	1	11	1	0
5	1	1	4	1	1	1	3	1	1	0
6	1	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0
8	18	20	26	21	23	21	22	23	20	21
9	nd	43	nd	nd	nd	nd	nd	nd	nd	nd
10	24	22	29	25	33	58	22	26	22	44
11	8	6	9	9	9	9	6	7	11	27
12	2	2	10	2	2	3	4	3	2	0
13	41	35	47	35	46	49	39	50	43	43
14	2	2	3	10	2	2	2	3	3	1
15	nd	nd	118	nd	679	677	nd	nd	46	nd
16	10	9	14	10	14	624	9	9	9	11
17	21	20	28	21	24	44	22	23	25	23
18	80	78	85	79	86	85	79	83	80	84
19	12	12	11	19	41	19	2	11	12	nd
20	71	68	64	74	77	88	64	76	77	72
21	276	283	140	489	565	472	480	492	299	302

nd = no detectado

Usando Q , los mejores métodos son LASSO, que obtiene, al compararlo con el CVA centralizado, mejores tiempos en 11 fallos y empata en 4; y *Sparse* PLS, que es mejor en 12 fallos y empata en 2. Después viene el método de Correlación y el de MI, con mejores tiempos en 9 fallos e iguales en 3; las redes neuronales, que son más rápidas que el centralizado en 8 fallos y dan el mismo tiempo en 4 fallos. A más distancia quedan *Sparse* PCA con 6 fallos mejores que el CVA centralizado e igual en 3; DCCA, que es más rápido en 6 fallos e igual en 1; RF, que es mejor en 5 fallos y empata en 2; y SVR, que es más rápido que el CVA centralizado en 4 fallos e igual en 3.

En las Tablas 4.17, 4.18 y 4.19 se muestran los resultados de MDR.

Tabla 4.17: Tasa de no detección de fallos (MDR), en porcentaje. T_s^2

Fallo	LASSO	<i>Sparse</i> PLS	<i>Sparse</i> PCA	NNET	RF	SVR	Correlación	DCCA	MI	Central CVA
1	0.38	0.38	0.38	0.25	0.88	0.38	0.25	0.5	0.38	0.10
2	1.5	1.38	1.75	0.88	2.63	2.13	1.63	1.75	1.63	1.1
3	89.25	97.25	95	96.5	98.63	98.75	96.25	92.75	94.38	98.1
4	0.13	0.13	0.13	0.13	56.63	97.5	0.13	0.38	0.13	68.8
5	0.13	0.13	67.13	0.13	0.13	74.75	4.38	0.13	0.13	0
6	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0
7	0.13	0.13	0.13	0.13	0.13	62.38	0.13	0.13	0.13	38.6
8	2.13	2.5	2.25	1.63	3.38	2.25	2.13	2.25	2.38	2.1
9	92.88	97	95.88	98.38	98.13	98.25	98.13	94.5	95.38	98.6
10	7.25	10	15.25	6.75	48.13	66.88	9.75	9.63	13	16.6
11	12.38	18.88	12.5	10.38	50.38	43.88	15.38	22.25	24.75	51.5
12	0.25	0.25	0.63	0.25	0.25	0.38	0.38	0.38	0.25	0
13	4.38	4.5	4.88	4.00	6.13	7.88	4.63	4.5	4.88	4.7
14	0.13	0.25	0.13	0.13	0.63	0.13	0.13	0.13	0.13	0
15	75.75	88.25	91.13	78.88	97	94.25	94.63	84.88	89.63	92.8
16	2.75	4.63	8.25	3.63	41.63	85.5	7.25	3.75	10.5	16.6
17	2.75	3.25	3.75	2.38	11.13	29.63	3	3	5.75	10.4
18	8.38	9.38	9.38	9.25	10.38	10.25	9.63	9.38	9.38	9.4
19	2.25	11.25	9.88	0.50	81	52	2.5	42.13	40.5	84.9
20	9.5	12.13	20.25	8.50	29.75	35.25	20	17.13	17.5	24.8
21	42.5	48.13	56.63	30.25	71.88	50.88	57	48.13	51.25	44

Con el estadístico T_s^2 , el mejor método, en cuanto a resultados de MDR, es el basado en redes neuronales, ya que es el que ofrece los valores más bajos en 16 fallos. Por detrás quedan: LASSO, que es el mejor en 14 fallos; los métodos *Sparse* PLS y DCCA, que ganan en 13 fallos; *Sparse* PCA y MI, que obtienen mejores MDR en 12 fallos; y Correlación que consigue mejores resultados en 11 fallos. Por detrás están RF que sólo mejora los resultados en 5 fallos; y SVR en 3 fallos.

Tabla 4.18: Tasa de no detección de fallos (MDR), en porcentaje. T_r^2

Fallo	LASSO	<i>Sparse</i> PLS	<i>Sparse</i> PCA	NNET	RF	SVR	Correlación	DCCA	MI	Central CVA
1	0.25	0.25	0.38	0.25	0.25	0.5	0.25	0.25	0.38	0
2	1.13	1.25	2	1.38	1.63	2.13	1.88	2	1.88	1.00
3	97.5	97.63	98.5	96.88	98.25	99.88	95.63	97.75	98.13	98.6
4	0.13	0.13	0.13	0.13	0.13	99.38	0.13	0.125	0.13	0
5	0.13	0.13	71.25	0.13	0.13	79.75	0.25	0.125	0.13	0
6	0.13	0.13	0.13	0.13	0.13	0.25	0.13	0.125	0.13	0
7	0.13	0.13	0.13	0.13	0.13	69.75	0.13	0.125	0.13	0
8	1.75	1.88	2.25	1.75	2.13	2.5	2	2.375	2.38	1.60
9	98.88	98.63	98.38	98.5	98.75	100	96.75	96.75	99.13	99.3
10	7.00	8.25	18.13	7.5	15.63	89.88	10.13	9.625	15.5	9.9
11	10.75	12.25	10.88	10.63	9.5	42.13	10.25	7.625	19.25	19.5
12	0.25	0.25	0.88	0.25	0.25	1.38	0.25	0.25	0.25	0
13	4.38	4.38	4.88	4.25	4.88	10	4.5	4.625	5	4.00
14	0.13	0.13	0.13	0.25	0.13	0.25	0.13	0.125	0.13	0
15	82.00	86	98.13	82.13	93.88	100	95.5	94.875	95.25	90.3
16	3.63	5.13	10.5	4.5	14.38	94.88	5.75	3.25	14	8.4
17	2.5	2.5	3.75	2.38	2.5	32.63	2.63	2.5	5.13	2.4
18	9.38	9.5	9.75	9.38	9.63	11.13	9.25	9.375	9.75	9.20
19	0.63	1.63	1.63	0.38	3.5	66.63	0.63	1.125	8.63	1.9
20	8.63	9.13	22.88	8.50	13.25	26.13	18	10.5	16	8.7
21	35.13	36.63	59.75	40.75	54.25	71.63	55.75	55.25	58.63	34.20

Al trabajar con T_r^2 el mejor resultado lo consigue, otra vez, el método basado en redes neuronales, ya que sus tasas MDR son mejores que las del CVA de [Russell et al., 2000] en 9 fallos. LASSO es mejor en 8 fallos; *Sparse* PLS en 7; Correlación en 5; DCCA en 6 fallos; *Sparse* PCA es mejor en 4; RF y MI en 3; y, por último, SVR no consigue mejores tasas MDR que el CVA centralizado en ningún fallo.

Tabla 4.19: Tasa de no detección de fallos (MDR), en porcentaje. Q

Fallo	LASSO	<i>Sparse</i> PLS	<i>Sparse</i> PCA	NNET	RF	SVR	Correlación	DCCA	MI	Central CVA
1	0.25	0.38	1	0.5	0.63	0.88	0.38	0.5	0.38	0.30
2	1.63	1.50	3.13	2.5	2	2.13	2.13	2.375	2	2.6
3	93.25	90.50	99.13	97.88	98.75	99.63	98.75	98.5	99	98.5
4	0.13	0.13	46.25	84.38	74.25	99.25	7.75	24	17.75	97.5
5	0.13	0.13	76.38	0.13	0.13	77.88	0.38	0.13	0.13	0
6	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0
7	0.13	0.13	0.13	10.38	0.13	66	0.13	0.13	0.13	48.6
8	2.25	2.00	5.38	3.38	2.88	2.63	2.63	2.5	2.38	48.6
9	93.13	88.13	99.38	98.88	99.13	99	99	98.5	99.38	99.3
10	10	9.63	38.5	16.88	36.75	76.63	8.38	17.25	10.88	59.9
11	17.5	15.13	50.13	45.13	50.63	62	27.63	41	35.38	66.9
12	0.25	0.25	5.88	0.25	0.25	0.75	0.63	0.38	0.25	2.1
13	4.88	4.25	8.88	4.13	5.75	8.63	4.75	6.25	5.13	5.5
14	0.25	0.25	0.63	2.75	0.25	0.25	0.25	0.38	0.38	12.2
15	89.13	85.38	96.63	98.5	97.38	97.5	98.25	98.38	83.50	97.9
16	6.5	4.75	27.25	18.38	42.38	91.38	8	12	10.25	42.9
17	3.63	4	19.25	3.88	10.5	46.13	5.13	6.63	10.13	13.8
18	9.25	8.38	10.5	9.5	10.75	10.38	9.75	9.5	9.75	10.2
19	10.13	4.88	35	62.5	57.88	37.13	3.00	41.88	52.13	92.3
20	16.25	16.38	24.38	34.75	33.75	48.25	21.5	31.38	40.13	35.4
21	51.88	50.63	71.5	71.38	70.25	58	67	61.25	62.13	54.7

Por último, los resultados MDR con Q muestran un gran resultado de LASSO, que consigue ser el mejor en 19 fallos. Justo detrás quedan *Sparse PLS* que es el mejor en 18 fallos, y, un poco más atrás, pero también con buenos resultados se sitúan las redes neuronales, que mejoran al CVA centralizado en 16 fallos; Correlación es mejor en 15 fallos; y RF, MI y DCCA consiguen mejorar las tasas MDR en 14 fallos. A cierta distancia ya quedan *Sparse PCA* con el mejor resultado en 10 fallos, y SVR que sólo consigue superar al CVA centralizado en 8 fallos.

Finalmente, la Tabla 4.20 muestra las tasas FAR, número de fallos detectados, presencia o no de falsas alarmas y el número de bloques de cada método.

Tabla 4.20: FAR (en porcentaje), fallos detectados, falsas alarmas y bloques.

	LASSO	<i>Sparse PLS</i>	<i>Sparse PCA</i>	NNET	RF	SVR	Correlación	DCCA	MI	Central CVA
FAR T_s^2	0	0	0.2004	0	0.2004	0.4008	2.7112	0.2004	4.6924	8.3
FAR T_r^2	0	0	0	0	0	0	3.0240	0	1.5641	12.6
FAR Q	0.4	0.2004	0	0	0.2004	0	1.1470	0.8016	0.8342	8.7
Fallos d. T_s^2	21	19	21	19	21	20	21	19	20	18
Fallos d. T_r^2	19	19	18	19	20	18	20	20	18	18
Fallos d. Q	18	19	19	18	19	19	18	18	19	17
Falsas a. T_s^2	No	No	No	No	No	No	No	No	No	No
Falsas a. T_r^2	No	No	No	No	No	No	No	No	No	No
Falsas a. Q	No	No	No	No	No	No	No	No	No	No
Bloques	52	52	26	52	52	52	52	52	52	1

Las tasas FAR son bastante más bajas que las del método CVA centralizado de [Russell et al., 2000], reduciendo así la posibilidad de que aparezcan falsas alarmas. En cuanto al número de fallos detectados, usando T_s^2 y Q , todos los métodos son capaces de detectar más fallos que el CVA centralizado; y con el estadístico T_r^2 , *Sparse PLS*, LASSO, las redes neuronales, RF y DCCA, superan al CVA centralizado, mientras que el resto de métodos lo igualan en número de fallos detectados.

Ninguno de los métodos propuestos detectó falsas alarmas usando el número de observaciones anómalas consecutivas fijado para cada uno de ellos. Finalmente, todos los métodos aplicaron una descentralización completa, de un bloque por variable (52), excepto *Sparse PCA*, que usó 26 bloques.

En resumen, se ha observado que en cada uno de los estadísticos los métodos dan diferentes resultados: con T_s^2 LASSO y las redes neuronales son los que mejor funcionan, dando un resultado mucho mejor que un CVA de tipo centralizado, siendo, el resto de métodos, en general mejores que el centralizado; con T_r^2 todos los métodos bajan su rendimiento, aunque siguen dando, la mayoría de métodos, tiempos de detección mejores que los del método centralizado, pero pierden efectividad en lo que a tasa MDR se refiere; con el estadístico Q los métodos recuperan eficacia en tasas MDR frente al CVA centralizado, aunque en los resultados de tiempo de detección sólo son mejores LASSO y *Sparse PLS* frente al método de un bloque.

4.4.4. MB-DCVA, sensibilidad al grado de distribución

Ahora se quiere comprobar el efecto de usar descentralizaciones con menor número de bloques. Para ello se emplearán metodologías que generen un número reducido de bloques y se comprobará si sus resultados son lo suficientemente buenos como para que merezca la pena su uso. Estas propuestas pueden ser interesantes, si son eficaces, para aplicarlas en plantas de gran tamaño donde una descentralización con muchos bloques no sea posible.

4.4.4.1. Metodología experimental

En estas pruebas se ha procedido de forma similar las anteriores de este mismo capítulo. Por tanto, se han utilizado varios métodos para descentralizar y se han ajustado los parámetros correspondientes para conseguir la máxima eficacia. También, al igual que en los anteriores métodos basados en CVA, se han agrupado las variables con menor influencia entre ellas, ya que se obtenían mejores resultados.

Los parámetros aplicados a cada método se muestran en la Tabla 4.21. Allí se indica la normalización que se aplicó a los datos, en caso de aplicarse; los bloques obtenidos; el orden de los CVA locales; la varianza en dichos modelos locales; el número de observaciones anómalas consecutivas para detectar un fallo; y el valor del parámetro α del índice BIC.

Tabla 4.21: Parámetros de los métodos distribuidos basados en CVA.

Método	Normalizado datos	Bloques	Orden CVA	Varianza en CVA locales	Observaciones para fallo	BIC α
NNET	Rango [0,1]	1 por variable	5	90 %	5	0.9
Correlación	Estándar	Menos bloques que variables	3	75 %	3	0.9
DCCA no solapado	Estándar	Se eliminan bloques < 5	5	60 %	3	0.9
DCCA solapado	Estándar	Menos bloques que variables	5	75 %	5	0.8
MI		Menos bloques que variables	3	90 %	3	0.9
DBSCAN		Menos bloques que variables	3	60 %	5	0.9

A continuación se comentan algunas particularidades de cada uno de los métodos empleados:

La metodología para las redes neuronales es la misma de la sección anterior, con los mismos parámetros. Se introduce aquí para comparar sus resultados con otras propuestas con menor número de bloques.

Para los métodos de correlación, DCCA e información mutua, de forma similar a la sección anterior, una variable se incluye en un bloque si su valor de correlación, DCCA o información mutua, respectivamente, con la variable a la que pertenece el bloque es inferior a un umbral. Este umbral se

ajustó, para la correlación, al valor de la mediana de las correlaciones analizadas para ese bloque multiplicado por 0,3. Para DCCA sin solapamiento el umbral fijado fue el valor de la mediana de los DCCA considerados en ese bloque multiplicado por 0,25. Si algún grupo queda de un tamaño menor a 5 variables, se elimina y la variable a la que pertenecía ese bloque se introduce en un bloque extra. Para el DCCA solapado el umbral se estableció de la siguiente forma: se toman los valores de DCCA de la variable cuyo grupo se va a construir con las demás variables, se calcula la mediana y se multiplica por un valor, que en este caso se fijó en 0,75 después de hacer unas pruebas. Para el MI también se usó el caso solapado, y el umbral considerado es la mediana multiplicada por el factor 1,3.

En el caso del método basado en agrupamiento (DBSCAN) se partió de los siguientes datos de cada variable: kurtosis, *skewness*, media y varianza de correlación, y media y varianza de información mutua. Dicho método se aplicó con un valor de ϵ de 0,2.

4.4.4.2. Análisis de resultados

Los resultados de tiempo de retardo en la detección usando T_s^2 se muestran en la Tabla 4.22.

Tabla 4.22: Retardo en la detección (en instantes). T_s^2

Fallo	NNET	Correlación	DCCA sin solap.	DCCA con solap.	MI	DBSCAN	Central CVA
1	2	3	3	7	2	3	2
2	9	13	15	25	11	15	13
3	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	45	<i>nd</i>	<i>nd</i>
4	1	1	1	1	1	1	462
5	1	1	2	2	1	1	1
6	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1
8	17	21	19	21	18	20	20
9	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	5	<i>nd</i>	<i>nd</i>
10	22	22	23	21	22	22	25
11	7	7	8	7	7	7	292
12	2	3	3	3	2	2	2
13	32	39	42	41	37	44	42
14	1	1	2	2	1	2	2
15	3	681	706	657	631	639	677
16	9	9	14	9	14	10	14
17	19	21	21	25	20	21	27
18	76	80	78	81	80	79	83
19	3	12	15	11	11	47	<i>nd</i>
20	67	74	69	88	67	71	82
21	207	271	227	251	263	490	273

nd = no detectado

El mejor resultado se obtiene con el método MI que mejora en 15 fallos al centralizado y lo iguala en 6; muy cerca está el DCCA solapado que

supera al CVA centralizado en 12 fallos y da el mismo resultado en 3 fallos; además, el método de redes neuronales, que es el que tiene más bloques, da un buen resultado también, ya que es más rápido que el método centralizado en 14 fallos, dando el mismo resultado en los 7 fallos restantes. Los demás métodos también mejoran al método centralizado: Correlación consigue el mejor tiempo en 11 fallos y empata en 6; DBSCAN es mejor en 9 y da el mismo resultado en 8 fallos; y DCCA sin solapar es más rápido en 9 fallos e iguala al centralizado en 7 fallos.

Tabla 4.23: Retardo en la detección (en instantes). T_r^2

Fallo	NNET	Correlación	DCCA sin solap.	DCCA con solap.	MI	DBSCAN	Central CVA
1	2	3	3	4	2	2	3
2	12	15	13	18	12	13	15
3	<i>nd</i>	484	538	485	<i>nd</i>	<i>nd</i>	<i>nd</i>
4	1	1	1	1	1	1	1
5	1	1	2	1	1	1	1
6	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1
8	18	19	20	21	16	19	20
9	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	731	<i>nd</i>
10	22	24	36	24	24	20	23
11	7	6	7	7	6	6	11
12	2	2	3	2	2	2	2
13	34	39	38	40	36	39	39
14	2	1	2	2	1	1	1
15	3	741	132	786	391	659	<i>nd</i>
16	10	9	10	8	11	9	9
17	19	20	21	23	20	20	20
18	76	79	76	18	76	76	79
19	3	2	2	11	2	2	11
20	66	65	66	94	65	65	66
21	287	484	211	466	174	456	511

nd = no detectado

Si se utiliza el estadístico T_r^2 (Tabla 4.23), el mejor resultado lo da el DBSCAN, ya que es mejor que el CVA centralizado en 11 fallos y empata en los 10 restantes; así como MI que es mejor en 10 fallos e igual en 9. También dan buen resultado el método de redes neuronales mejorando al centralizado en 11 fallos y empatando en 8; DCCA sin solapamiento es más rápido en 8 fallos e igual en 7; y Correlación tiene mejores tiempos en 7 fallos y da el mismo resultado en 13. Finalmente el método basado en DCCA con solapamiento solo es mejor en 6 fallos y empata en 7 con el centralizado.

Finalmente, se analizan los resultados de retardo en la detección con Q (Tabla 4.24). Las redes neuronales son mejores que el método centralizado en 8 fallos e iguales en 4; MI es mejor en 9 y da el mismo tiempo en 2 fallos. Por detrás y con peores resultado que el método centralizado están: DCCA con solapamiento que es mejor en 8 fallos, DBSCAN (7 fallos más rápido y

1 igual), Correlación (4 fallos mejor y 3 igual) y DCCA sin solapamiento (3 fallos con mejor tiempo y otros 3 con igual retardo).

Tabla 4.24: Retardo en la detección (en instantes). Q

Fallo	NNET	Correlación	DCCA sin solap.	DCCA con solap.	MI	DBSCAN	Central CVA
1	4	10	10	10	4	6	2
2	19	27	46	27	15	13	25
3	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>
4	249	5	<i>nd</i>	<i>nd</i>	1	1	0
5	1	7	4	1	1	1	0
6	1	1	1	1	1	1	0
7	1	1	3	1	1	1	0
8	21	26	30	25	22	22	21
9	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	<i>nd</i>	239	<i>nd</i>
10	25	25	36	31	27	25	44
11	9	7	99	11	9	11	27
12	2	27	4	3	2	2	0
13	35	47	58	54	45	49	43
14	10	3	2	6	2	2	1
15	<i>nd</i>	<i>nd</i>	<i>nd</i>	775	677	524	<i>nd</i>
16	10	10	20	9	12	14	11
17	21	24	30	26	22	28	23
18	79	89	88	86	82	82	84
19	19	13	15	14	13	17	<i>nd</i>
20	74	76	78	101	65	73	72
21	489	639	648	213	288	485	302

nd = no detectado

En cuanto a los resultados de tasas MDR usando el estadístico T_s^2 (Tabla 4.25), el mejor método es el basado en redes neuronales que es el mejor en 16 fallos. Después, DBSCAN consigue mejores tasas MDR que el centralizado en 10 fallos; MI y DCCA con solapamiento son mejores en 9 fallos; Correlación en 7 y DCCA sin solapamiento en 5 fallos.

Los resultados de tasas MDR con T_r^2 (Tabla 4.26) muestran que las redes neuronales y DBSCAN consiguen los mejores resultados en 9 fallos; Correlación en 7; MI en 5; y los dos DCCA son mejores en 4 fallos.

Ahora, con el estadístico Q (Tabla 4.27), los resultados de los métodos mejoran: MI es el mejor en 17 fallos; DBSCAN en 16 fallos y empata en 1 con el método centralizado y las redes neuronales dan la mejor tasa MDR en 16 fallos; mientras que DCCA con solapamiento es mejor en 14 fallos. Por contra, Correlación sólo es mejor en 9 fallos y DCCA sin solapamiento en 6.

CAPÍTULO 4. MÉTODOS FDI DISTRIBUIDOS MULTIBLOQUE

Tabla 4.25: Tasa de no detección de fallos (MDR), en porcentaje. T_s^2

Fallo	NNET	Correlación	DCCA sin solap.	DCCA con solap.	MI	DBSCAN	Central CVA
1	0.25	0.38	0.38	0.88	0.25	0.38	0.10
2	0.88	1.63	1.88	2.75	1.38	1.88	1.1
3	96.5	98.75	99.5	95.62	98.63	97.13	98.1
4	0.13	0.13	4.38	84	0.13	5.38	68.8
5	0.13	74.13	0.25	0.25	0.13	0.13	0
6	0.13	0.13	0.13	0.12	0.13	0.13	0
7	0.13	0.38	0.13	0.12	0.13	0.13	38.6
8	1.63	2.63	2.38	2.62	2.25	2.5	2.1
9	98.38	99.13	99.75	96.38	98.5	99.13	98.6
10	6.75	11	20.13	10.62	15.63	11.38	16.6
11	10.38	24.63	35.38	36.88	20.13	35.5	51.5
12	0.25	0.75	0.38	0.38	0.25	0.25	0
13	4.0	4.88	5.25	5.0	4.63	5.13	4.7
14	0.13	0.13	0.25	0.25	0.13	0.25	0
15	78.88	97.5	98.25	88.62	93.38	90.25	92.8
16	3.63	8.38	19.88	3.12	26.25	9.75	16.6
17	2.38	3.75	6.75	6.5	2.63	9.75	10.4
18	9.25	9.88	9.75	9.5	10	9.75	9.4
19	0.5	37.38	55.25	54	13.25	55.5	84.9
20	8.5	31.13	25.25	32.5	14.13	22.38	24.8
21	30.25	56	53.63	49.75	50.25	57.5	44

Tabla 4.26: Tasa de no detección de fallos (MDR), en porcentaje. T_r^2

Fallo	NNET	Correlación	DCCA sin solap.	DCCA con solap.	MI	DBSCAN	Central CVA
1	0.25	0.38	0.38	0.5	0.25	0.25	0
2	1.38	1.75	1.63	2.25	1.5	1.63	1.00
3	96.88	95.38	98	98.5	96.13	93.00	98.6
4	0.13	0.13	0.13	0.12	0.13	0.13	0
5	0.13	70.63	0.25	0.12	0.13	0.13	0
6	0.13	0.13	0.13	0.12	0.13	0.13	0
7	0.13	0.13	0.13	0.12	0.13	0.13	0
8	1.75	2.13	2.5	2.62	2	2	1.60
9	98.5	96.13	98.75	98.88	98.38	96.00	99.3
10	7.5	9.88	57.75	14.38	17.88	7.63	9.9
11	10.63	13.63	11.5	14.62	8.13	11.5	19.5
12	0.25	0.38	0.38	0.25	0.25	0.25	0
13	4.25	4.75	4.75	5	4.5	4.75	4.00
14	0.25	0.13	0.25	0.25	0.13	0.13	0
15	82.13	95.88	98.13	97.62	88.5	79.25	90.3
16	4.5	6.88	8.5	5.37	17.5	5.25	8.4
17	2.38	2.38	2.88	3.63	2.5	3.5	2.4
18	9.38	9.25	9.38	9.25	9.38	9.00	9.2
19	0.38	1.88	0.75	19.25	1.25	1.13	1.9
20	8.5	22.25	14.25	46.75	11.13	8.25	8.7
21	40.75	59	49.25	56.62	53.63	54.5	34.20

Tabla 4.27: Tasa de no detección de fallos (MDR), en porcentaje. Q

Fallo	NNET	Correlación	DCCA sin solap.	DCCA con solap.	MI	<i>DBSCAN</i>	Central CVA
1	0.5	1.13	1.13	1.25	0.5	0.75	0.30
2	2.5	3.25	5.5	2.75	2.00	1.5	2.6
3	97.88	99.13	99	94.38	95.88	93.38	98.5
4	84.38	20.88	98.88	93.25	3.63	23.5	97.5
5	0.13	83.75	0.5	0.12	0.13	0.13	0
6	0.13	0.13	0.13	0.12	0.13	0.13	0
7	10.38	74	19.63	0.12	0.13	0.13	48.6
8	3.38	11.88	20.75	6.13	2.88	2.25	48.6
9	98.88	98.5	99	95.75	98.13	90.88	99.3
10	16.88	21.63	49.88	26.5	33.13	24.13	59.9
11	45.13	40	77.38	63	36.63	25.38	66.9
12	0.25	15.38	6.88	0.38	0.38	0.38	2.1
13	4.13	8.5	16.5	6.37	5.38	5.5	5.5
14	2.75	1	4.25	8.5	0.25	0.25	12.2
15	98.5	99.63	98.25	94.25	94.5	87.25	97.9
16	18.38	20.25	52.88	10.25	34.63	20.88	42.9
17	3.88	11	25.88	13.88	6.75	12.5	13.8
18	9.5	11.13	10.88	9.13	9.88	9.50	10.2
19	62.5	80.13	85.75	61.38	20.38	62.88	92.3
20	34.75	43.25	39.25	68.38	17.25	31	35.4
21	71.38	93.38	85.88	50.62	61.88	58.88	54.7

Por último, en la Tabla 4.28 se muestran los resultados de tasas FAR, número de fallos detectados, falsas alarmas y bloques de cada método.

Tabla 4.28: FAR (en porcentaje), fallos detectados, falsas alarmas y bloques.

Fallo	NNET	Correlación	DCCA sin solap.	DCCA con solap.	MI	<i>DBSCAN</i>	Central CVA
FAR T_s^2	0	0.94	0	1	1.36	1.67	8.3
FAR T_r^2	0	2.29	0	0	2.61	4.07	12.6
FAR Q	0	0.52	0	0	1.77	3.55	8.7
Fallos d. T_s^2	19	19	19	19	21	19	18
Fallos d. T_r^2	19	20	20	20	19	20	18
Fallos d. Q	18	18	17	18	19	20	17
Falsas a. T_s^2	No	No	No	No	No	No	
Falsas a. T_r^2	No	No	No	No	No	No	
Falsas a. Q	No	No	No	No	No	No	
Bloques	52	5	4	18	10	3	1

Las tasas FAR de los métodos propuestos son más bajas que las del CVA centralizado (Tabla 4.28). Y son capaces de detectar más fallos en todos los estadísticos, salvo DCCA sin solapamiento con Q que detecta el mismo número. Destacan en este aspecto MI con T_s^2 , que detectan los 21 fallos; Correlación, DCCA con y sin solapamiento y *DBSCAN* con T_r^2 que son capaces de detectar 20 fallos; y *DBSCAN* con Q , que también detecta 20 fallos. Por otro lado, ningún método presentó falsas alarmas. En cuanto al número de bloques, el método de redes neuronales es del tipo de descen-

tralización total y tiene, por tanto, un bloque por variable, es decir, 52. En cuanto a los métodos con menos bloques: MI es un método con 10 bloques, Correlación reduce a 5 bloques, DCCA sin solapamiento tiene 4 bloques y con solapamiento trabaja con 18; y DBSCAN, que es el de menor tamaño, cuenta con 3 bloques.

En resumen, los tiempos de detección son mejores con los métodos propuestos que con el CVA centralizado cuando se usa T_s^2 y T_r^2 . Y con Q , sólo con los métodos de redes neuronales y MI. En cuanto a las tasas MDR, con T_s^2 es mejor el método de redes neuronales frente al CVA centralizado, y DBSCAN y MI dan un resultado casi igual, teniendo en cuenta, que en muchos fallos en los que estos métodos dan tasas más altas de MDR la diferencia es mínima (por ejemplo, en los fallos 5, 6 y 7, DBSCAN da una tasa de 0.13 y el CVA centralizado da 0). Con T_r^2 , DBSCAN queda próximo al CVA centralizado, pero, como en el caso anterior, hay fallos en los que los resultados de ambos métodos son muy similares. En Q los resultados mejoran, y es MI el que da el mejor resultado de MDR, seguido de las redes neuronales y DBSCAN, métodos estos que son mejores que el CVA centralizado. Correlación queda cerca de éste, y DCCA no funciona bien.

Por tanto, a la vista de los resultados, MI y DBSCAN son métodos con pocos bloques que funcionan con un buen rendimiento y en la mayoría de ocasiones son preferibles al método centralizado.

4.5. Resumen

En este capítulo se han presentado una serie de métodos de detección descentralizados y que aplican técnicas dinámicas (DPCA y CVA). La forma de hacer la descentralización se ha basado tanto en analizar las correlaciones entre variables (Correlación, DCCA o Información Mutua), como en modelar cada variable en función de las demás (LASSO, redes neuronales, etc.); y, una vez hechos estos análisis, usar esta información para crear los grupos. Se han hecho pruebas tanto con DPCA como con CVA como método de detección de fallos en los bloques generados en la descentralización, y se han fusionado los resultados locales usando el Criterio de Inferencia Bayesiano.

En el caso del DPCA se ha comprobado que los diferentes métodos descentralizados propuestos dan mejor resultado que un DPCA distribuido equivalente, ya que son más rápidos y más sensibles en el proceso de detección (mejores tasas MDR). Además, también se ha comprobado que usar descentralizaciones con un número reducido de bloques sigue siendo preferible a usar un método centralizado que cuente con un sólo bloque, ya que se consiguen mejores resultados, tanto en tiempos de detección, como en tasas MDR y en número de fallos detectados.

En el caso del método de detección CVA, se ha propuesto un proceso alternativo de descentralización en el que, en vez de agrupar las variables

que tienen más relación entre sí en el mismo bloque, se han agrupado las que están menos relacionadas. Esto ha hecho que, con casi todos los métodos propuestos, se obtengan buenos resultados, llegando a ser mejores que un CVA centralizado en retardo en la detección; y dan mejores tasas de MDR con T_s^2 y Q . Por lo tanto parece que el método CVA descentralizado funciona mejor al incluir variables poco correlacionadas que aporten información nueva a cada modelo local.

Además, los modelos descentralizados basados en CVA con menor número de bloques funcionan mejor que el CVA centralizado en cuanto a tiempo de detección y están próximos a él en cuanto a tasas MDR. Siendo capaces, también, de detectar más fallos.

Capítulo 5

IDENTIFICACIÓN DE FALLOS MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN. DISTRIBUCIÓN DE MÉTODOS

5.1. Introducción

En el proceso de monitorización de una planta la etapa que se ejecuta después de detectarse un fallo es el de identificación de la variable o variables responsables del mismo. El objetivo es conocer el origen del fallo y poder hacer una diagnosis correcta del tipo de fallo para tomar las medidas correctivas más adecuadas. En el caso de aplicar métodos de monitorización basados en datos las técnicas existentes de detección utilizan unos índices y unos umbrales de tal forma que si alguno de esos índices supera su correspondiente umbral, se detecta un fallo. A continuación, los métodos de identificación asociados tratan de cuantificar la influencia de cada variable en el hecho de que un estadístico haya superado el umbral, de tal forma que aquellas variables con mayor influencia serán consideradas como responsables del fallo.

Por lo visto en el Capítulo 2, del estado del arte, se constata que existe una gran variedad de métodos de identificación, y ahora se plantea el problema de seleccionar uno de ellos, eligiendo aquel que sea el más eficaz. De acuerdo con las pruebas realizadas por los autores de dichos métodos sobre una misma planta industrial, se observa que los resultados de la identificación no siempre coinciden entre los métodos [Kourti and MacGregor, 1996, Detroja et al., 2007, Mahadevan and Shah, 2009, Kariwala et al., 2010, Liu, 2012, Jiang and Braatz, 2017]. Por tanto, esto lleva a pensar que no hay un método que esté por encima de los demás en términos de eficacia, y por tanto, no es posible quedarse con uno de ellos como método universal.

De aquí surge la idea de descentralizar la identificación mediante la im-

plantación de varios centros de procesamiento, contando, cada uno de los cuales, con un método de identificación diferente. El objetivo es utilizar todos los métodos de forma simultánea aprovechando las sinergias mutuas entre métodos y su complementariedad. Es decir, al contar con varios métodos, cuando se da una situación de fallo en la que algunos métodos concretos no sean capaces de dar una buena diagnosis, se dispone de los restantes métodos para realizar una identificación correcta.

Por tanto, la aproximación considerada ahora es distinta a la del capítulo anterior, ahora se trata de distribuir los métodos de FDI entre los nodos que se han distribuido por la planta, es decir, se trata de implementar un método diferente de identificación de fallos en cada unidad de cálculo y después fusionar los resultados de cada método en un nodo central, usando la teoría de toma de decisiones multicriterio.

Además, también se busca solucionar otros dos problemas que aparecen al realizar la identificación de fallos, por un lado, evitar el uso de métodos de identificación como clasificadores para diagnosticar fallos, ya que normalmente en las plantas industriales, no se dispone de suficientes datos de todos los posibles fallos que puedan ocurrir en el sistema; y, por otro lado, evitar tener que elegir un único método de identificación ya que se va a llevar a cabo una fusión de todos ellos, con el objetivo de obtener las ventajas de cada uno de los métodos.

Hay otro problema adicional relacionado con la identificación que se ha encontrado al estudiar la bibliografía, y consiste en saber el momento exacto en el que la identificación del fallo debe llevarse a cabo. Inicialmente, se puede pensar que hay que hacerla nada más detectarse el fallo, para tener el resultado de la identificación cuanto antes. Ese es el planteamiento de muchos autores [Kourti and MacGregor, 1996, Dunia and Qin, 1998, Yue and Qin, 2001, Detroja et al., 2007, Alcalá and Qin, 2009, Mahadevan and Shah, 2009, Kariwala et al., 2010]; mientras que otros autores realizan la identificación analizando gran cantidad de instantes posteriores la detección del fallo [Liu, 2012, Liu and Chen, 2014, Jiang et al., 2015].

La identificación en el instante del fallo presenta el problema de que puede verse afectada por la influencia de ruidos, perturbaciones, etc. además de tener en cuenta sólo las variables afectadas por el fallo en los primeros instantes.

Por otro lado, la segunda opción para la identificación tiene en cuenta muchas variables que se ven influenciadas por la expansión de los efectos del fallo por la planta. Además, si la planta cuenta con sistemas de control, estos tratarán de corregir el fallo en la planta modificando ciertas variables controladas, que pasarán a tomar valores anómalos. Todo esto aparecerá reflejado en esta identificación a largo plazo dando como resultado una diagnosis incierta.

Además, en esta segunda opción, el proceso de la identificación se realiza con datos tomados de la planta funcionando bajo condiciones de fallo (inicio

y evolución del fallo, y, posteriormente, entrada en estado estacionario) y, por tanto, el resultado no va a aportar una información clara del origen del fallo pero sí del estado en el queda la planta al expandirse este.

5.2. Fundamentos teóricos

En el Capítulo 2, del estado del arte, se muestran diversos métodos de identificación de fallos basados en PCA, aunque no son los únicos existentes. Diversos autores han presentado otras técnicas para identificar a las variables responsables de un fallo, como son los índices de reconstrucción o los diagramas de contribución modificados que se presentan a continuación.

5.2.1. Índices basados en reconstrucción

Una vez que se ha detectado un fallo, porque alguno de los índices estadísticos (T^2 , Q o φ) ha superado su correspondiente umbral, se denomina Contribución Basada en Reconstrucción (en inglés, *Reconstruction Based Contribution* (RBC)) para una cierta variable, a la cantidad de reconstrucción que minimiza el índice de detección aplicada en la dirección de dicha variable [Alcala and Qin, 2009]. De esta forma se puede obtener un valor de RBC para cada variable, identificando como responsables del fallo a aquellas con un valor más alto.

Esta propuesta surge para paliar algunas desventajas que presentan los métodos de identificación basados en contribuciones a T^2 , Q y φ , como que, en algunos casos, las variables afectadas por el fallo no obtienen los valores más altos de contribución [Yoon and MacGregor, 2001].

El índice RBC para una variable i -ésima se obtiene de la siguiente manera:

$$RBC_i^{index} = \mathbf{x}^T M \boldsymbol{\xi}_i \left(\boldsymbol{\xi}_i^T M \boldsymbol{\xi}_i \right)^{-1} \boldsymbol{\xi}_i^T M \mathbf{x} \quad (5.1)$$

donde x es la observación actual y la matriz M se calcula de la forma siguiente:

$$\begin{aligned} \mathbf{M} = \mathbf{D} &= \mathbf{P} \boldsymbol{\Lambda}^{-1} \mathbf{P}^T \text{ para } T^2 \\ \mathbf{M} = \tilde{\mathbf{C}} &= \tilde{\mathbf{P}} \tilde{\mathbf{P}}^T \text{ para } Q \\ \mathbf{M} = \boldsymbol{\Phi} &= \frac{\tilde{\mathbf{C}}}{T_\alpha^2} + \frac{\mathbf{D}}{Q_\alpha} \text{ para } \varphi \end{aligned}$$

Este método se aplica de forma similar a los métodos basados en contribuciones a los estadísticos de detección: una vez detectado el fallo con un estadístico, se obtiene el valor de RBC para este estadístico en las direcciones de cada una de las variables, pudiendo tener así una tabla de resultados

en la que identificar a aquellas variables con valores de RBC más altos como responsables del fallo.

5.2.2. Diagramas de contribución modificados

Este método se basa en encontrar las variables que, después de ser reconstruidas, consiguen las mayores reducciones del estadístico combinado φ hasta que este caiga por debajo de su umbral [Liu and Chen, 2014]. Busca evitar el efecto de expansión del fallo que hace que las variables no responsables se vean contaminadas por las realmente responsables, dificultando el proceso de identificación.

El método trabaja siempre con el índice de detección combinado φ . Una vez que se ha detectado un fallo con este estadístico se procede a la identificación, en la que se aplica una reconstrucción en las direcciones de cada una de las variables y se comprueba qué reconstrucción consigue una mayor reducción del índice anómalo φ .

Si se reconstruye la variable i , la contribución de dicha variable a la reducción del índice (o estadístico) combinado (en inglés, *Reduction of Combined Index* (RCI)) es:

$$c_i^{RCI} = \left[(\mathbf{x}_{nf} - \mathbf{x}_{nf}^*)^T (\boldsymbol{\xi}^T \Phi \boldsymbol{\xi})^{0,5} \boldsymbol{\xi}_i \right]^2 \quad (5.2)$$

donde $\boldsymbol{\xi}_i$ es un vector columna con un 1 en la posición i y 0 en el resto, \mathbf{x}_{nf} es el conjunto de variables anómalas y \mathbf{x}_{nf}^* es la reconstrucción de esas variables anómalas:

$$\mathbf{x}_{nf}^* = - \left(\boldsymbol{\xi}^T \Phi \boldsymbol{\xi} \right)^{-1} \boldsymbol{\xi}^T \Phi (1 - \mathbf{\Gamma}) \mathbf{x} \quad (5.3)$$

donde $\mathbf{\Gamma}$ es una matriz diagonal con unos en las posiciones de la diagonal que corresponden con las variables anómalas y ceros en en las posiciones de las variables no anómalas.

La reducción del estadístico combinado después de la reconstrucción se puede calcular así:

$$\varphi - \varphi_{nf}^* = (\mathbf{x}_{nf} - \mathbf{x}_{nf}^*)^T (\boldsymbol{\xi}^T \Phi \boldsymbol{\xi}) (\mathbf{x}_{nf} - \mathbf{x}_{nf}^*) \quad (5.4)$$

Aquella variable que consiga una mayor reducción de φ será incorporada al conjunto de variables anómalas, después se aplica la reconstrucción a lo largo de esa variable a los datos y se repite el proceso, buscando otra variable que maximice RCI que será incluida, también, en el conjunto de variables anómalas. Cada vez que se repita el ciclo, se comprueba el valor obtenido para el estadístico combinado después de la reconstrucción, de modo que cuando éste cae por debajo de su umbral, se detiene el proceso, y las variables contenidas en el conjunto de variables anómalas son las consideradas

como responsables del fallo, analizando ahora, cuáles de estas han conseguido un mayor valor de RCI y que, por tanto, serán las que tienen mayor responsabilidad en el fallo.

5.3. Método propuesto: Identificación de fallos distribuida basada en operadores lingüísticos OWA-RIM: ORAFI

Por todo lo dicho anteriormente, surge la necesidad de buscar un método fiable para llevar a cabo la identificación de fallos. Esto se pretende realizar con la descentralización de la identificación mediante la implantación de varios centros de procesamiento, contando, cada uno de los cuales, con un método de identificación diferente. El objetivo es evitar tener que decantarse por uno u otro método, teniendo en cuenta que ninguno de ellos nos va a dar un resultado satisfactorio siempre. Además, al contar con varios métodos, cuando se de una situación de fallo en la que algunos métodos concretos no sean capaces de dar una buena diagnosis, se dispone de los restantes métodos para realizar una identificación correcta.

Por tanto, tal como se explicó en el Capítulo 1, de introducción, el segundo objetivo que se planteó a la hora de abordar esta tesis consistió en proponer un método de identificación de fallos distribuido. La aproximación considerada es distinta a la del capítulo anterior, ahora se trata de distribuir los métodos de FDI entre los nodos que se han distribuido por la planta, es decir, se trata de implementar un método diferente de identificación de fallos, de los más utilizados en la literatura, como diagramas de contribución a T^2 y a Q , índices de reconstrucción, etc., en cada unidad de cálculo y después fusionar los resultados de cada método en un nodo central, usando la teoría de toma de decisiones multicriterio, para, a partir de los resultados obtenidos por cada método, obtener una decisión única de identificación del fallo.

Por otro lado, tomando en consideración el momento donde aplicar la identificación del fallo, en este trabajo se proponen dos posibles soluciones a este problema. Por un lado se propone realizar identificaciones tanto en el instante de detección del fallo como varios instantes más adelante, cuando la planta ha evolucionado a un nuevo estado estacionario después de expandirse el fallo por la planta, realizando la identificación en ventanas de tiempo, no en instantes únicos.

En el caso de la identificación inicial se busca hacer el proceso más robusto frente a ruidos, perturbaciones, etc y en la diagnosis tardía, analizar sólo instantes que pertenezcan a la planta ya en estado estacionario después del fallo. De esta forma el objetivo es obtener información del origen del fallo (identificación temprana) y del estado en el que queda la planta una vez se ha evolucionado bajo el efecto de dicho fallo (identificación estacionaria).

Además, así se podrá comprobar si los resultados de las diagnosis hechas por varios autores usando todos los datos disponibles sirven para conocer el origen del fallo.

5.3.1. Descripción del Método ORAFI

Este método propuesto, denominado ORAFI (en inglés, OWA-RIM aggregation based Fault Diagnosis) [Sánchez-Fernández et al., 2018e] consiste en un método de identificación de fallos que realiza una agregación de las diagnosis de 7 métodos de identificación ampliamente usados en los últimos años, como son: Contribuciones de las variables a T^2 [Kourti and MacGregor, 1996] y φ [Alcala and Qin, 2009], Error normalizado de las variables [Kourti and MacGregor, 1996], Índices de reconstrucción (para T^2 , Q y φ) [Alcala and Qin, 2009], y Diagramas de contribución modificados [Liu and Chen, 2014], a través, de un operador lingüístico (OWA-RIM) usado como un método de toma de decisión multicriterio, que además se puede sintonizar por parte del usuario dependiendo del riesgo que quiera asumir al tomar la decisión.

Por otro lado, esta identificación de las variables responsables del fallo, se realiza con la información recogida en dos ventanas de tiempo: la primera se aplica inmediatamente después de la detección del fallo y se denomina diagnosis *Temprana* y la segunda se aplica después de que el sistema ha evolucionado a un nuevo estado estacionario después del fallo: diagnosis *Estacionaria*. Esta segunda fase se va a usar para ver cómo ha evolucionado la planta en condiciones de fallo y comprobar si realizar una identificación tantos instantes después del fallo es válida para encontrar las variables responsables del fallo.

Por lo tanto en este segundo método de FDI distribuido propuesto, la distribución se hace por métodos, es decir, un método de identificación de fallo en cada nodo. En el Algoritmo 1 se puede ver una visión general del método.

La sintonización del método de agregación se hará en función de la aversión al riesgo del usuario, que podrá elegir diferentes operadores OWA y, en cada caso, la agregación de métodos será más o menos exigente. Esto se mide con el carácter *orness* (o con el *andness*) del operador. De modo que el usuario del sistema de monitorización va a poder elegir cómo hacer la toma de decisiones en función del riesgo que pueda asumir: un operador con bajo valor de *orness* implica menor riesgo, y viceversa.

La detección del fallo se realiza usando un PCA estándar usando los estadísticos T^2 , Q y φ , de modo que si alguno de dichos estadísticos supera su umbral un cierto número de veces, se activará la alarma de fallo y se pasará a realizar el proceso de identificación. Más tarde, durante una corta ventana de tiempo después de que el fallo ha sido detectado, se aplican todos

Algoritmo 1 *Diagnosis de fallos basada en operadores OWA-RIM*

```

1: Entrenamiento:
2: for Datos sin fallo do
3:   Ajustar un nivel de riesgo para MCDM.
4:   Métodos identificación: {  $Cont^{T^2}$ ,  $Cont^Q$ ,  $Cont^\varphi$ ,  $RBC^{T^2}$ ,  $RBC^Q$ ,  $RBC^\varphi$  y  $RCI$  }
5:   Ajustar  $\Delta t_{Temprana}$  para  $min$ (retardo) con  $max$ (separabilidad)
6:   Ajustar  $\Delta t_{Estacionaria}$ ,  $Th_{Estacionaria}$  y  $Var_{Estacionaria}$  para  $min$ (retardo) con
    $max(Detecciones_{Estabilización})$ 
7: end for
8:
9: Datos a analizar
10: for t=1 to n do ▷ Para cada instante
11:   Diagnosis Temprana:
12:   if FALLO = TRUE then
13:     while t < ( $t_{Detección} + \Delta t_{Temprana}$ ) do ▷ Ventana Temprana
14:        $D_{FI_j} \leftarrow DiagnosisOf\{FI_j\}$  ▷ Diagnosis de cada método identificación
15:        $F_t^{Variable_k}(D_{FI_1}, \dots, D_{FI_n}) = \sum_{j=1}^n \omega_j \beta_j$  ▷ Puntuaciones ( $\omega_j$ : pesos OWA)
16:         ( $\beta_j$  = Valores ordenados de  $Variable_k$  en  $\langle D_{FI_1}, D_{FI_2}, \dots, D_{FI_n} \rangle$ )
17:     end while
18:
19:     if t = ( $t_{Detección} + \Delta t_{Temprana} - 1$ ) then ▷ Fin de ventana Temprana
20:        $F_t^{Variable_k} \leftarrow media(F_t^{Variable_k})$ ,  $t \in \{Ventana Temprana\}$ 
21:        $Diagnosis(Temprana) \leftarrow Ranking(F^{Variable_1}, \dots, F^{Variable_m})$ 
22:     end if
23:   end if
24:
25:   Diagnosis Estacionaria:
26:   Comprobar Estado Estacionario
27:    $D_{FI_j} = DiagnosisBasadaEn\{FI_j\}$  ▷ Diagnosis de cada método identificación
28:    $F_t^{Variable_k}(D_{FI_1}, \dots, D_{FI_n}) = \sum_{j=1}^n \omega_j \beta_j$  ▷ Puntuaciones en t ( $\omega_j$ : pesos OWA)
29:     ( $\beta_j$  = Valores ordenados de  $Variable_k$  en  $\langle D_{FI_1}, D_{FI_2}, \dots, D_{FI_n} \rangle$ )
30:
31:   if  $F_t^{Variable_k}$  no cambian en  $\Delta t_{Estacionaria}$  then
32:     Estado Estacionario = TRUE
33:      $F_t^{Variable_k} = media(F_s^{Variable_k})$  con  $t_s \in$  último  $\Delta t_{Estacionaria}$ 
34:      $Diagnosis(Estacionaria) = Ranking(F^{Variable_1}, \dots, F^{Variable_m})$ 
35:   end if
36: end for
37:
38: Criterio para Estado Estacionario:
39: Primeras posiciones  $Ranking F_t^{Variable_k}$  no cambian en  $\Delta t_{Estacionaria}$ 
40: Puntuaciones de las primeras posiciones varían menos que  $Var_{Estacionaria}$  en
    $\Delta t_{Estacionaria}$ 

```

los métodos de identificación de fallos considerados en esta propuesta, dando cada uno sus propios resultados.

Estos resultados se agregan utilizando un enfoque lingüístico basado en una función OWA-RIM, lo que resulta en una clasificación global puntuada, es decir, en un ranking de las variables candidatas responsables de los fallos. Aquellas variables que estén en las posiciones más altas del ranking serán consideradas como las responsables del fallo. Esta es la identificación de fallos temprana.

Después de esta identificación inicial, se va verificando el sistema hasta que se alcanza un nuevo estado estacionario después del fallo. En este momento, se lleva a cabo otra identificación del fallo basada en las mismas técnicas, el operador OWA-RIM y los datos recogidos durante este periodo de tiempo, dando lugar a otra identificación que se ha denominado identificación estacionaria.

Este método requiere ajustar varios parámetros, como son: el riesgo asumido durante la diagnosis, la longitud de las ventanas de tiempo de la toma de datos para la identificación y la forma de detectar que el sistema ha llegado a un punto estacionario en su evolución después del fallo. Los criterios que se han seguido para ajustarlos son los siguientes:

- **Riesgo asumido en la diagnosis:** se ha utilizado un operador OWA-RIM para hacer la agregación de los resultados de los métodos de identificación. Los pesos del operador OWA se han obtenido a partir del cuantificador RIM: $Q(r) = r^\alpha$. El valor de α escogido por el usuario determinará el nivel de riesgo en la toma de decisiones: un valor alto de α implica un valor bajo de *orness* (ver ecuación 2.43), es decir, una alta aversión al riesgo (toma de decisiones poco arriesgada).
- **Longitud de las ventanas de tiempo:** es necesario ajustar el tamaño de las ventanas en las que se toman datos para la identificación: para la ventana *Temprana*: $\Delta t_{Temprana}$, justo después de detectarse el fallo, y para la ventana *Estacionaria*: $\Delta t_{Estacionaria}$, que se aplica al encontrar el sistema un nuevo valor estacionario en su evolución después del fallo.
 - $\Delta t_{Temprana}$: este valor se ajusta buscando la máxima separabilidad entre las puntuaciones de las variables en el ranking obtenido al hacer la agregación. Se trata de retrasar el diagnóstico lo menos posible, y por tanto tomar el menor tamaño de ventana que permita una mejor discriminación entre variables candidatas.
 - $\Delta t_{Estacionaria}$: el tamaño de esta ventana se ajusta buscando el valor con el que las variables diagnosticadas como candidatas no cambien, partiendo de datos de diferentes fallos. El tamaño de esta ventana se busca entre varios valores, tomando aquel que ofrezca una diagnosis lo más pronto posible, para la mayoría de los fallos, a la vez que dicho valor elegido permita detectar que el sistema llega a un nuevo estado estacionario de operación (en cada fallo, el valor de $t_{Estacionaria}$ es diferente).
- **Instante en que se llega a un nuevo estado estacionario:** este valor se obtiene aplicando el método diagnosis OWA-RIM usando el valor de $\Delta t_{Estacionaria}$ seleccionado en cada instante después del fallo. Esto da lugar a un ranking de variables candidatas en cada instante. Y

la planta se considera en estado estacionario si se dan las condiciones siguientes:

- Las variables de las primeras posiciones que superen un cierto umbral $Th_{Estacionaria}$ no cambien durante un número de instantes igual a $\Delta t_{Estacionaria}$.
- Las puntuaciones de dichas variables candidatas no cambie durante $\Delta t_{Estacionaria}$. La forma de comprobar que no cambian es analizar la varianza ($Var_{Estacionaria}$) de estas variables durante ese espacio de tiempo y comprobar que no supere un cierto umbral (elegido por el usuario).

5.4. Caso de estudio I: Tennessee Eastman Process

La planta TEP, detallada en la Sección A.3, es la que se ha utilizado para probar el método de toma de decisiones basada en operadores OWA. En este caso se han aplicado el método ORAFI, tal como se explica en la Sección 5.3.1.

5.4.1. Metodología experimental

La metodología experimental: parámetros y características, se pueden consultar en el Algoritmo 2.

Se crea un PCA usando los datos sin fallo que retiene un 70 % de la varianza en los componentes principales seleccionados (dicho valor se selecciona después de hacer test de validación cruzada, buscando el menor número de componentes que retenga la mayor cantidad de información). Los umbrales para T^2 y Q se ajustaron para tener un 0.5 % de observaciones por encima de ellos en condiciones sin fallo, buscando evitar falsas alarmas, pero sin perder sensibilidad ante fallos de pequeño tamaño.

Los pesos de la diagnosis basada en OWA-RIM se calcularon en función de la aversión al riesgo que quiera aplicar el operador: alta ($orness=0.2$), moderada ($orness=0.5$) y baja ($orness=0.9$). La longitud de la primera ventana de tiempo ($Temprana$) se ajustó a 3 muestras, consiguiendo una diagnosis lo más rápida posible con el mejor valor posible de *Separabilidad* entre las puntuaciones de las variables (ver Tabla 5.1).

Una ventana de 10 observaciones fue seleccionada para la diagnosis *Estacionaria* para conseguir la diagnosis más rápida pero evitando falsas detecciones de estados estacionarios de la planta. También las variables seleccionadas deben tener una varianza en sus puntuaciones por debajo de 0.002 (el tiempo de detección del estado estacionario se incrementa para valores más bajos, y con valores más altos crece el número de falsas detecciones

Algoritmo 2 *Metodología experimental*

```
1: Preproceso datos con fallo de entrenamiento: normalización Z-score
2: Modelo PCA: 70% de la varianza incluida, umbrales ajustados a per-
  centil 99,5.
3:
4: Ajuste de  $\Delta t_{Temprana}$  (Al detectar fallo):
5: for cada fallo do
6:   for RiskAversion=HIGH to LOW do
7:     for  $\Delta t_{Temprana} = 2$  to  $7$  do
8:       Calcular:  $Separabilidad = Varianza(Puntuaciones(\Delta t_{Temprana}))$ 
       $\triangleright$  Separación entre puntuaciones
9:     end for
10:    return ( $\Delta t_{Temprana}$ ) con  $min(\Delta t_{Temprana})$  y  $max(Separabilidad)$ 
11:  end for
12:
13: Ajuste  $\Delta t_{Estacionaria}$  (Planta en estado estacionario con fallo):
14: for cada fallo do
15:   for Aversión-al-Riesgo=ALTA to BAJA do
16:     for  $\Delta t_{Estacionaria} = 5$  to  $15$  do
17:       for  $Var_{Estacionaria} = 0.001$  to  $0.003$  do
18:         for  $Th_{Estacionaria} = 0.3$  to  $0.7$  do
19:           Calcular:  $Retardo_{Estacionaria}$  y  $Detecciones_{Estacionaria}$ 
20:         end for
21:       end for
22:     end for
23:     return ( $\Delta t_{Estacionaria}$ ) y ( $Var_{Estacionaria}$ ) con
       $min(Retardo_{Estacionaria})$  y  $max(Detecciones_{estacionaria})$ .
24:   end for
25:
26: Detección e identificación de fallos:
27: Preproceso datos de test con fallo: normalización Z-score
28: if Fallo = TRUE then
29:   Realizar identificación fallo Temprana en  $\Delta t_{Temprana}$ 
30:   for cada instante do
31:     Comprobar estado estacionario
32:     if Estado Estacionario = TRUE then
33:       Realizar identificación fallo Estacionaria en  $\Delta t_{Estacionaria}$ 
34:     end if
35:   end for
36: end if
```

de estados estacionarios). Las Tablas 5.2, 5.3 y 5.4 muestran algunos de los experimentos llevados a cabo para seleccionar dichos valores. La planta se considera en estado estacionario si las variables del ranking con una puntuación superior a 0.5 no cambian durante las observaciones que dura la ventana *Estacionaria* (ver el Algoritmo 2 para el resto de valores probados).

Tabla 5.1: Ajuste de la longitud de la ventana *Temprana*

Aversión al riesgo	Alta (<i>orness</i>=0.2)			
Tamaño ventana (muestras)	2	3	5	7
Separabilidad	0.006	0.0062	0.0049	0.0039
Aversión al riesgo	Moderada (<i>orness</i>=0.5)			
Tamaño ventana (muestras)	2	3	5	7
Separabilidad	0.009	0.012	0.01	0.008
Aversión al riesgo	Baja (<i>orness</i>=0.9)			
Tamaño ventana (muestras)	2	3	5	7
Separabilidad	0.024	0.027	0.022	0.019

Tabla 5.2: Ajuste del tamaño de la ventana *Estacionaria*. Varianza=0.001

Aversión al riesgo	Alta (<i>orness</i>=0.2)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	351.21	357.45	458.51	465.89	461.52
Estados Estacionarios detectados	12	15	16	17	17
Aversión al riesgo	Moderada (<i>orness</i>=0.5)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	326.84	354.21	399.54	416.27	429.95
Estados Estacionarios detectados	15	16	17	17	18
Aversión al riesgo	Baja (<i>orness</i>=0.9)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	388.14	409.53	419.74	457.15	468.29
Estados Estacionarios detectados	15	17	17	17	18

5.4.2. Análisis de resultados

Los resultados de la diagnosis sólo mostrarán las seis primeras variables de los ranking junto con sus respectivas puntuaciones. Con esto es suficiente para ver los resultados del método.

El desempeño del método ha sido comparado con otras propuestas encontradas en la literatura: la diagnosis *Temprana* se ha comparado con: un PCA estándar que aplica la contribución a Q como método de identificación [Kourti and MacGregor, 1996], el método de Ramificación y Poda (B&B) [Kariwala et al., 2010], Análisis de Correspondencia (CA) [Detroja et al., 2007] y el método de Máquina de Vector Soporte de Clase Única

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

Tabla 5.3: Ajuste del tamaño de la ventana *Estacionaria*. Varianza=0.002

Aversión al riesgo	Alta (<i>orness</i> =0.2)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	320.46	347.56	453.05	458.88	459.55
Estados Estacionarios detectados	13	16	18	18	18
Aversión al riesgo	Moderada (<i>orness</i> =0.5)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	316.75	348.19	398.63	405.33	424.49
Estados Estacionarios detectados	16	16	18	18	18
Aversión al riesgo	Baja (<i>orness</i> =0.9)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	382.18	403.88	414.47	449.11	459.5
Estados Estacionarios detectados	16	17	18	18	18

Tabla 5.4: Ajuste del tamaño de la ventana *Estacionaria*. Varianza=0.003

Aversión al riesgo	Alta (<i>orness</i> =0.2)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	310.55	331.55	442.15	449.84	451.57
Estados Estacionarios detectados	10	14	15	17	17
Aversión al riesgo	Moderada (<i>orness</i> =0.5)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	306.53	344.84	401.36	402.81	421.41
Estados Estacionarios detectados	13	14	16	16	16
Aversión al riesgo	Baja (<i>orness</i> =0.9)				
Tamaño ventana (muestras)	5	7	10	12	15
Retardo medio (muestras)	373.22	399.51	409.74	431.01	439.57
Estados Estacionarios detectados	14	16	17	17	17

(SVM) [Mahadevan and Shah, 2009]. Todos estos métodos entregan un resultado de diagnóstico en el momento de detectarse el fallo.

La diagnosis *Estacionaria* se ha comparado con métodos muy conocidos que utilizan todo los datos disponibles del conjunto de datos para dar la diagnosis. Estos métodos son: Análisis de Variables Canónicas (CVA) [Jiang et al., 2015] y Diagramas de Contribución Modificados (RCI) [Liu, 2012].

Identificación de fallos Temprana Los resultados para esta diagnosis se muestran en las Tablas 5.5, 5.6 y 5.7. Cada una de ellas representa los resultados obtenidos para cada grado de aversión al riesgo elegido: alta (*orness*=0.2), moderada (*orness*=0.5) y baja (*orness*=0.9).

La diagnosis de esta planta no es conocida y no está muy claro qué variables son las responsables de cada uno de los fallos, de modo que no se dispone de un resultado correcto de diagnosis con el que comparar los resul-

tados del método a prueba. Por esta razón se han tomado varios métodos de identificación aplicados a esta planta y se han comparado los resultados obtenidos con cada uno de ellos.

En dichas tablas se observa que las variables seleccionadas no siempre coinciden entre los métodos. A pesar de ello, en muchos fallos, nuestra propuesta elige como candidatas algunas de las variables elegidas por otras propuestas. Además, el método ORAFI entrega una lista de variables candidatas con puntuaciones, lo que permite la selección de aquellas realmente relevantes. Posteriormente, se deberá llevar a cabo la diagnosis, en la que, partiendo de los resultados proporcionados por la identificación y del conocimiento previo de la planta, se encuentre el tipo de fallo ocurrido y se decidan las medidas a tomar para corregirlo.

Existe cierta diferencia entre aplicar un grado de aversión al riesgo u otro. Con el valor alto ($orness=0.2$), las puntuaciones obtenidas por las variables son más bajas, pero la separabilidad entre puntuaciones del ranking es mejor. Por ejemplo, en el fallo 5 las variables candidatas al fallo son las variables: 11 y 35, ya que obtienen puntuaciones bastante más altas que las siguientes variables. En efecto, las puntuaciones son 0.5098 y 0.4717, respectivamente para 11 y 35, mientras que la tercera variable obtiene 0.1348, con lo que se aprecia claramente un salto que permite extraer las dos primeras variables.

En cambio, si observamos los resultados para el mismo fallo 5 aplicando baja aversión al riesgo ($orness=0.9$), las puntuaciones de las 3 primeras variables son, respectivamente: 0.8427, 0.7423 y 0.6653, por tanto, no se observa un salto claro en las puntuaciones que permita encontrar un grupo de variables candidatas.

En cualquier caso, todos los valores de aversión al riesgo dan como resultado un grupo de variables candidatas similares, lo que cambia, sobre todo, es la distancia de las puntuaciones de dichas variables con respecto a las demás, de tal forma que adoptando una estrategia de alta aversión al riesgo se facilita una identificación más clara.

Además, como se mencionó antes, nuestra propuesta aporta un ranking ordenado y cuantificado de variables candidatas y esto ayudará en la diagnosis del fallo ocurrido en la planta. Por otra parte, este ranking incluye muchas de las variables que los otros métodos de la comparativa consideran como responsables del fallo, por lo que parece que nuestra propuesta no va mal encaminada, siempre teniendo en cuenta que no existe un resultado correcto de identificación de los fallos de la planta TEP.

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

Tabla 5.5: Comparativa para la identificación *Temprana*. Alta aversión al riesgo (*orness*=0.2)

Fallo	Ranking	Puntuación	PCA	B & B	CA	SVM
1	16 20 7 25 13 23	0.5483 0.3303 0.2297 0.2212 0.1836 0.1127	16 20 46	16	16 13 7 20	16 18
2	30 24 10 47 16 7	0.4621 0.2364 0.1477 0.1090 0.0602 0.0318	30 24 16 40	21 24 30	40 4 16 13 20 7	47 10 16
3	37 32 8 45 14 18	0.1333 0.0931 0.0867 0.0760 0.0744 0.0709	nd		nd	nd
4	51 9 24 30 28 35	0.5870 0.0158 0.01520 0.0109 0.0102 0.0085	51 21 9	9 51	51	51
5	11 35 8 14 24 30	0.5098 0.4717 0.1348 0.0909 0.0845 0.0816	11 9 35 22 18	11 22	4 21 22 11	52 17
6	44 1 20 21 13 7	0.4531 0.1857 0.0367 0.01807 0.0123 0.0119	44 1	1 44	44 1	1
7	16 7 13 4 45 21	0.5968 0.4071 0.3661 0.2771 0.2078 0.1355	4 16 36 38 31 25	4 6 9 16 22 45 51	16 7 13 21 4	45
8	20 16 46 31 25 23	0.5221 0.3970 0.3239 0.2416 0.2331 0.1594	3 46 29 16 20	37	46 13 7 16 20	16 23 20 30
9	nd	nd	nd		nd	nd
10	18 8 23 20 3 31	0.2912 0.1154 0.0918 0.0872 0.07819 0.0704	19 34 35		19 20 18	18 38 50
11	51 9 39 42 21 14	0.3858 0.0280 0.0086 0.0065 0.0061 0.0060	51 9 21	51	51	51
12	11 22 35 9 36 34	0.3660 0.2213 0.0685 0.0344 0.0265 0.0204	11 37 22 4	22	21 4 11 22	11 22 4
13	16 7 32 33 13 6	0.3655 0.2485 0.2407 0.2181 0.2024 0.0934	37 32 26 42	37	51 16 20 13 7	34 16
14	51 9 21 2 42 28	0.2783 0.2737 0.1914 0.0247 0.0138 0.0061	21 51 9 2 42	9 51	21 51	9 51
15	nd	nd	nd		nd	nd
16	19 18 32 29 1 8	0.5600 0.3840 0.0787 0.0413 0.0310 0.0268	50 3 27		50 19	50 19
17	21 2 42 27 24 35	0.5978 0.0125 0.0091 0.0042 0.0038 0.0034	21 2 42		51 21	29 9
18	22 11 35 33 51 24	0.8027 0.0712 0.0108 0.0096 0.0083 0.0082	22 11 40		4 11 40 22	51 17
19	5 32 39 43 3 27	0.3553 0.0555 0.0353 0.0331 0.0313 0.0274	5 27 37 46 38		40 51 20 4	5 20 46
20	46 13 20 34 5 40	0.6478 0.1466 0.0932 0.0446 0.0265 0.0235	46 39 13 23		8 42 11 40	13 52 17 7
21	19 21 50 40 23 3	0.1670 0.1221 0.0979 0.0755 0.0715 0.0641	39 26 5		13 7 42 20	19 37 16

nd = no detectado

Tabla 5.6: Comparativa para la identificación *Temprana*. Moderada aversión al riesgo ($\alpha=0.5$)

Fallo	Ranking	Puntuación	PCA	B & B	CA	SVM
1	16 20 25 7 23 46	0.669 0.5514 0.3217 0.2976 0.2807 0.2722	16 20 46	16	16 13 7 20	16 18
2	30 24 10 47 31 25	0.7137 0.4399 0.3570 0.3453 0.2020 0.1871	30 24 16 40	21 24 30	40 4 16 13 20 7	47 10 16
3	8 37 45 32 24 14	0.3073 0.2953 0.2849 0.2744 0.2301 0.2151	nd		nd	nd
4	51 9 34 24 6 14	0.7876 0.1627 0.0735 0.0690 0.0647 0.0610	51 21 9	9 51	51	51
5	35 11 4 14 22 24	0.6524 0.6151 0.3231 0.2916 0.2796 0.2460	11 9 35 22 18	11 22	4 21 22 11	52 17
6	44 1 20 46 13 7	0.7700 0.5422 0.2089 0.1537 0.1119 0.1052	44 1	1 44	44 1	1
7	16 4 7 13 45 31	0.8118 0.5931 0.5528 0.5126 0.3457 0.2975	4 16 36 38 31 25	4 6 9 16 22 45 51	16 7 13 21 4	45
8	20 16 46 31 25 29	0.7388 0.5849 0.5831 0.3528 0.3222 0.2801	3 46 29 16 20	37	46 13 7 16 20	16 23 20 30
9	nd	nd	nd		nd	nd
10	18 20 4 23 31 3	0.5900 0.2833 0.2602 0.2385 0.1984 0.1983	19 34 35		19 20 18	18 38 50
11	51 9 6 21 27 39	0.4678 0.1938 0.0575 0.0482 0.0468 0.0416	51 9 21	51	51	51
12	11 22 35 41 36 6	0.5313 0.4559 0.1965 0.1279 0.1122 0.0982	11 37 22 4	22	21 4 11 22	11 22 4
13	16 32 33 7 13 21	0.5990 0.4806 0.4682 0.4142 0.3614 0.2817	37 32 26 42	37	51 16 20 13 7	34 16
14	51 9 21 2 42 6	0.4287 0.4250 0.3707 0.163 0.1137 0.0608	21 51 9 2 42	9 51	21 51	9 51
15	nd	nd	nd		nd	nd
16	19 18 29 32 31 50	0.7436 0.5232 0.2450 0.1937 0.1509 0.1487	50 3 27		50 19	50 19
17	21 2 42 51 9 41	0.7028 0.1031 0.0863 0.0428 0.0387 0.0348	21 2 42		51 21	29 9
18	22 11 13 7 2 40	0.8922 0.3307 0.0818 0.0707 0.0554 0.0528	22 11 40		4 11 40 22	51 17
19	5 43 24 46 32 34	0.5538 0.1826 0.1793 0.1517 0.1517 0.1488	5 27 37 46 38		40 51 20 4	5 20 46
20	46 13 20 7 5 23	0.7359 0.2849 0.2179 0.1628 0.1570 0.1295	46 39 13 23		8 42 11 40	13 52 17 7
21	19 50 21 40 18 29	0.5140 0.3886 0.3439 0.2643 0.2599 0.2506	39 26 5		13 7 42 20	19 37 16

nd = no detectado

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

Tabla 5.7: Comparativa para la identificación *Temprana*. Baja aversión al riesgo

Fallo	Ranking	Puntuación	PCA	B & B	CA	SVM
1	16 20 46 23 18 25	0.8361 0.7934 0.5562 0.5550 0.5312 0.5088	16 20 46	16	16 13 7 20	16 18
2	30 24 31 47 25 10	0.9153 0.7108 0.598 0.5673 0.5643 0.5578	30 24 16 40	21 24 30	40 4 16 13 20 7	47 10 16
3	8 37 45 24 32 29	0.5919 0.5365 0.5276 0.4857 0.4833 0.4781	nd		nd	nd
4	51 9 34 20 14 6	0.9210 0.4219 0.2461 0.2151 0.2024 0.1994	51 21 9	9 51	51	51
5	35 11 4 22 26 14	0.8427 0.7423 0.6653 0.6149 0.5854 0.5771	11 9 35 22 18	11 22	4 21 22 11	52 17
6	44 1 20 46 13 7	0.9346 0.8263 0.5802 0.4979 0.3622 0.3403	44 1	1 44	44 1	1
7	16 4 7 13 31 8	0.9437 0.7974 0.7398 0.7216 0.7190 0.6742	4 16 36 38 31 25	4 6 9 16 22 45 51	16 7 13 21 4	45
8	20 46 16 18 29 31	0.8985 0.7995 0.7876 0.5995 0.5527 0.5492	3 46 29 16 20	37	46 13 7 16 20	16 23 20 30
9	nd	nd	nd		nd	nd
10	18 20 4 29 23 3	0.8054 0.5397 0.5149 0.4375 0.4226 0.3912	19 34 35		19 20 18	18 38 50
11	51 9 27 21 6 29	0.6107 0.5558 0.1685 0.1621 0.1589 0.1526	51 9 21	51	51	51
12	11 22 35 41 6 24	0.8321 0.7427 0.4923 0.4377 0.3396 0.3112	11 37 22 4	22	21 4 11 22	11 22 4
13	16 32 33 21 7 40	0.7887 0.7642 0.7417 0.6040 0.6028 0.5612	37 32 26 42	37	51 16 20 13 7	34 16
14	51 9 21 2 42 22	0.5604 0.5545 0.5355 0.4569 0.3639 0.2035	21 51 9 2 42	9 51	21 51	9 51
15	nd	nd	nd		nd	nd
16	19 18 29 50 31 32	0.9176 0.6899 0.6144 0.5489 0.4319 0.4130	50 3 27		50 19	50 19
17	21 2 42 51 9 22	0.8322 0.2951 0.2511 0.1296 0.1139 0.1080	21 2 42		51 21	29 9
18	22 11 13 7 16 40	0.9628 0.7411 0.3182 0.2880 0.2020 0.1879	22 11 40		4 11 40 22	51 17
19	5 24 43 29 46 34	0.7335 0.5094 0.4483 0.4451 0.4240 0.4198	5 27 37 46 38		40 51 20 4	5 20 46
20	46 13 7 5 23 20	0.8672 0.5104 0.4676 0.3955 0.3828 0.3807	46 39 13 23		8 42 11 40	13 52 17 7
21	19 50 21 18 3 29	0.9080 0.7351 0.5726 0.5140 0.5078 0.4908	39 26 5		13 7 42 20	19 37 16

nd = no detectado

Identificación de fallos *Estacionaria* El retardo respecto a la detección del fallo con el que cada método da la identificación *Estacionaria*, se muestra en la Tabla 5.8, en la que se compara nuestra propuesta con los métodos CVA y RCI.

Tabla 5.8: Retardo en la diagnosis *Estacionaria*.

Fallo	<i>Estacionaria</i> diagnosis			CVA	RCI
	Alta aversión al riesgo (<i>orness</i> =0.2)	Moderada aversión al riesgo (<i>orness</i> =0.5)	Baja aversión al riesgo (<i>orness</i> =0.9)		
1	417	464	494	800	800
2	43	74	122	800	800
3	800	800	800	nd	nd
4	69	34	33	800	800
5	168	168	168	800	800
6	41	42	189	800	800
7	59	52	158	800	800
8	800	89	773	800	800
9	nd	nd	nd	nd	nd
10	800	800	800	800	800
11	800	800	800	800	800
12	800	800	800	800	800
13	113	101	114	800	800
14	184	83	74	800	800
15	nd	nd	nd	nd	nd
16	800	800	265	800	800
17	93	56	55	800	800
18	342	276	116	800	800
19	800	800	800	800	800
20	800	800	800	800	800
21	679	535	514	800	800
Media	453	399	414	800	800
Desv. Típ.	339	342	324	0	0
Media (sólo estacionarios)	201	165	237	800	800
Desv. Típ. (sólo estacionarios)	201	171	221	0	0

nd = no detectado

Los métodos CVA y RCI usan todos los datos de los conjuntos de datos, por tanto necesitan 800 observaciones para generar la diagnosis. Cuando el método ORAFI no es capaz de detectar la entrada de la planta en un nuevo estado estacionario, el retado en entregar el resultado es el mismo que con los otros métodos: 800 observaciones. Pero en los fallos en los que sí se detecta que se ha llegado a un nuevo estado estacionario, nuestro método es claramente más rápido. Por ejemplo, en el fallo 4, se obtienen retardos de 69, 34 y 33 instantes, según se aplique alta, moderada o baja aversión al riesgo,

respectivamente, frente a los 800 instantes que se demora la identificación con los otros métodos de la comparativa. Por tanto la mejora es muy significativa.

Cuando se adopta la estrategia de moderada aversión al riesgo ($orness=0.5$) se consigue el menor valor medio de retardo, aunque no hay diferencias significativas entre los tres grados de aversión al riesgo.

Cabe destacar que la metodología propuesta reduce el tiempo de la diagnosis para el sistema en estado estacionario en hasta un 50 % en comparación con los otros métodos. En algunos casos se reduce en 15 veces. La varianza de los retardos es alta porque hay fallos en los que no es posible detectar la llegada a un estado estacionario de la planta, y esto hace subir el valor de este estadístico. Al final de la Tabla 5.8 se muestran los valores medios y la varianza tomando en cuenta sólo los fallos en los que el método ha detectado la entrada del sistema en estado estacionario, resultado en una reducción del retardo en una cuarta parte del retardo obtenido con el CVA y el RCI. Por tanto, en la mayoría de fallos el método propuesto es mucho más rápido que las otras propuestas.

Las Tablas 5.9, 5.10 y 5.11 muestran los resultados de la diagnosis para cada valor de aversión al riesgo ($orness$) elegido. Al reducir la aversión al riesgo las puntuaciones del ranking crecen, como se esperaba, pero se pierde, en parte, la separabilidad entre las variables candidatas y las demás. Los resultados de la identificación son similares a los ofrecidos por el CVA y el RCI, pero se obtienen más pronto. Además, el usuario tiene la opción de ajustar el grado de riesgo que quiere asumir con su toma de decisiones. En el caso de alta aversión al riesgo ($orness=0.2$) las primeras variables del ranking obtienen unas puntuaciones claramente más altas que el resto, es decir, se consigue una mejor separabilidad, y se facilita la toma de decisiones, aunque tomando las otras dos opciones de aversión al riesgo sigue siendo posible obtener un grupo de candidatas en casi todos los fallos.

También, como en el caso de la identificación *Temprana*, nuestra propuesta da información acerca de las variables mas relevantes en la identificación, mientras que el CVA entrega, en muchos casos (fallos 2, 4, 6, 8, etc.), una lista demasiado larga de variables candidatas, dificultando el análisis del estado de la planta. Si se compara con RCI, ORAFI da una lista de todas las variables con su puntuación, es decir, con el nivel de influencia de cada una de ellas en el fallo y esto facilitará mucho la diagnosis del fallo.

Por tanto, a la vista de los resultados, queda claro que las diagnosis en dos etapas es imprescindible, ya que los resultados de los métodos que analizan los instantes inmediatamente posteriores al fallo son muy diferentes de los que analizan todos los datos disponibles. La planta evoluciona después del fallo, y, en muchos casos, la diagnosis varía entre los primeros instantes después del fallo y cuando la planta llega, tiempo más tarde, a un estado estacionario. En el primer caso, la identificación sirve para encontrar el origen del fallo, y en el segundo, para analizar el nuevo punto de funcionamiento de la planta después de suceder el fallo. Además, disponemos de una lista de

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

variables ordenadas con sus correspondientes puntuaciones, lo que facilitará mucho la diagnosis.

Tabla 5.9: Diagnosis Estacionaria. Alta aversión al riesgo ($orness=0.2$)

Fallo	Ranking	Puntuaciones	CVA-SSFV	CVA-RSFV	RCI
1	50 1 44 18 19 20	0.4918 0.2993 0.2947 0.1484 0.09517 0.0599	18 19 50	1 4 34 44 45	1 4 18 44 45 50
2	47 30 10 24 31 16	0.2475 0.2440 0.2428 0.1048 0.0251 0.0214	18 19 28 34 47 50	3 9 13 25 29 34 43 46 47	34 28 10 47
3	36 40 30 4 24 41	0.1142 0.0613 0.0546 0.0524 0.05134 0.0487	nd	nd	nd
4	51 14 9 5 8 42	0.5160 0.0150 0.0128 0.0122 0.0107 0.0099	nd	51	51
5	50 19 18 38 7 13	0.5423 0.4254 0.2556 0.1482 0.1132 0.1070	50	52	38 19
6	44 1 16 29 20 7	0.5857 0.2291 0.1702 0.1086 0.1003 0.0690	2 3 7 11 13 16 21 22 23 25 26 30 31 33 35 36 38 43 44 47	1 2 3 7 8 9 11 13 16 17 21 25 28 29 31 34 35 36 37 38 39 42 43 45 46 51 52	51
7	45 4 8 21 20 36	0.7031 0.1379 0.1256 0.0236 0.0210 0.0177	nd	45	45
8	16 7 13 10 47 34	0.7097 0.4684 0.4069 0.1811 0.1537 0.1481	7 10 11 13 16 18 19 20 23 30 47 50	1 4 7 10 13 20 24 25 28 29 31 34 35 43 44 46 47	16 10 47 7 30 13
9	nd	nd	nd	nd	nd
10	32 6 19 27 14 5	0.0695 0.0590 0.0554 0.0482 0.0441 0.0397	7 13 19 50	18	18
11	51 9 32 6 5 40	0.3401 0.0411 0.0095 0.0076 0.0075 0.0074	nd	9 51	51 9
12	38 11 50 22 21 8	0.2803 0.1574 0.0863 0.0659 0.0562 0.0539	7 9 11 13 16 18 19 20 22 23 27 29 30 31 33 36 38 47 50	1 3 4 8 20 21 25 35 43 52	11 38 4 22
13	19 50 18 16 7 13	0.7503 0.4427 0.3292 0.1066 0.0774 0.0705	7 11 13 16 18 19 20 21 22 23 26 27 30 31 33 36 38 41 44 47 50	3 4 7 8 11 13 16 18 19 20 25 27 28 29 30 32 34 35 36 37 43 50 51	19 38 50 16 7 13
14	51 9 21 2 42 6	0.2952 0.2412 0.1819 0.0104 0.0095 0.0068	50	9 21 51	51 9 21
15	nd	nd	nd	nd	nd
16	19 18 38 35 40 32	0.2900 0.1471 0.1326 0.0678 0.0583 0.0555	7 13 18 19 20 50	16	19 80 18
17	21 9 2 42 51 6	0.5324 0.0208 0.0085 0.0059 0.0029 0.0014	7 9 11 13 16 18 19 21 30 50	9 21 51	21
18	43 51 46 42 45 4	0.8269 0.5617 0.2013 0.1088 0.0711 0.0547	4 7 9 11 13 14 16 20 22 23 24 26 27 28 29 30 33 35 36 39 41 46 47 50 51	6 7 12 13 16 17 19 21 23 25 27 28 30 32 33 35 36 37 38 41 42 43 45 46 48 51 52	43 51 46
19	5 46 35 41 39 34	0.0770 0.0603 0.0525 0.0492 0.0484 0.0268	nd	5	5
20	46 20 13 11 7 29	0.5190 0.1469 0.0649 0.0253 0.0142 0.0126	7 11 13 16 18 20 13 50	20 46	46
21	8 19 50 7 16 13	0.4648 0.2124 0.1414 0.1237 0.1231 0.1156	nd	nd	8

nd = no detectado

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

Tabla 5.10: Diagnósis Estacionaria. Moderada aversión al riesgo ($or-ness=0.5$)

Fallo	Ranking	Puntuaciones	CVA-SSFV	CVA-RSFV	RCI
1	44 1 4 50 19 18	0.6463 0.6412 0.6061 0.4114 0.3218 0.2525	18 19 50	1 4 34 44 45	1 4 18 44 45 50
2	30 10 47 24 25 37	0.5662 0.5495 0.5473 0.1810 0.1135 0.1120	18 19 28 34 47 50	3 9 13 25 28 29 34 43 46 47	34 28 10 47
3	36 40 30 4 24 41	0.1142 0.0613 0.0546 0.0524 0.0513 0.0487	nd	nd	nd
4	51 9 28 34 6 43	0.6505 0.1527 0.0811 0.0634 0.0628 0.0432	nd	51	51
5	38 1 44 50 19 52	0.6612 0.2033 0.2010 0.1073 0.1046 0.1041	50	52	38 19
6	44 1 16 29 20 7	0.7842 0.5435 0.310 0.2545 0.1719 0.1568	2 3 7 11 13 16 21 22 23 25 26 30 31 33 35 36 38 43 44 47	1 2 3 7 8 9 11 13 16 17 21 25 28 29 31 34 35 36 37 38 39 42 43 45 46 51 52	51
7	45 4 8 46 20 30	0.8464 0.3166 0.2050 0.1609 0.1539 0.1156	nd	45	45
8	16 7 13 10 47 34	0.8154 0.5558 0.4963 0.3885 0.3440 0.3379	7 10 11 13 16 18 19 20 23 30 47 50	1 4 7 10 13 20 24 25 28 29 31 34 35 43 44 46 47	16 10 47 7 30 13
9	nd	nd	nd	nd	nd
10	38 19 50 36 24 26	0.4743 0.2748 0.2723 0.1899 0.1761 0.1675	7 13 19 50	18	18
11	51 9 32 6 5 40	0.3401 0.0411 0.0095 0.0076 0.0075 0.0074	nd	9 51	51 9
12	11 16 35 7 13 36	0.5120 0.3317 0.2744 0.2595 0.2349 0.1910	7 9 11 13 16 18 19 20 22 23 27 29 30 31 33 36 38 47 50	1 3 4 8 20 21 25 35 43 52	11 38 4 22
13	16 7 19 13 38 50	0.6855 0.4980 0.4720 0.4536 0.4410 0.3799	7 11 13 16 18 19 20 21 22 23 26 27 30 31 33 36 38 41 44 47 50	3 4 7 8 11 13 16 18 19 20 25 27 28 29 30 32 34 35 36 37 43 50 51	19 38 50 16 7 13
14	51 21 9 2 6 42	0.3991 0.3748 0.3484 0.08745 0.0821 0.0794	50	9 21 51	51 9 21
15	nd	nd	nd	nd	nd
16	50 34 38 29 20 28	0.5430 0.1863 0.160 0.1542 0.1424 0.1373	7 13 18 19 20 50	16	19 80 18
17	21 2 42 9 51 6	0.5630 0.0861 0.0735 0.0612 0.0370 0.0259	7 9 11 13 16 18 19 21 30 50	9 21 51	21
18	22 11 13 7 16 35	0.5848 0.2475 0.1233 0.1166 0.1067 0.0521	4 7 9 11 13 14 16 20 22 23 24 26 27 28 29 30 33 35 36 39 41 46 47 50 51	6 7 12 13 16 17 19 21 23 25 27 28 30 32 33 35 36 37 38 41 42 43 45 46 48 51 52	43 51 46
19	5 46 35 41 39 34	0.0770 0.0603 0.0525 0.0492 0.0484 0.0268	nd	5	5
20	46 11 13 20 7 22	0.4837 0.1424 0.1319 0.1213 0.1197 0.1145	7 11 13 16 18 20 13 50	20 46	46
21	8 19 50 16 7 44	0.5760 0.2630 0.2252 0.1845 0.150 0.1451	nd	nd	8

nd = no detectado

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

Tabla 5.11: Diagnosis Estacionaria. Baja aversión al riesgo (*orness*=0.9)

Fallo	Ranking	Puntuaciones	CVA-SSFV	CVA-RSFV	RCI
1	1 44 4 45 24 21	0.9134 0.9103 0.7810 0.4620 0.3950 0.3745	18 19 50	1 4 34 44 45	1 4 18 44 45 50
2	10 47 30 37 34 25	0.9013 0.8562 0.7942 0.5145 0.4254 0.4139	18 19 28 34 47 50	3 9 13 25 28 29 34 43 46 47	34 28 10 47
3	36 40 30 4 24 41	0.1142 0.0613 0.0546 0.0524 0.0513 0.0487	nd	nd	nd
4	51 9 28 6 34 21	0.8690 0.4667 0.2311 0.1998 0.1654 0.1568	nd	51	51
5	52 17 40 33 38 23	0.5027 0.4127 0.3544 0.3238 0.3115 0.3029	50	52	38 19
6	51 9 16 1 7 44	0.9955 0.7129 0.7108 0.5244 0.5158 0.4922	2 3 7 11 13 16 21 22 23 25 26 30 31 33 35 36 38 43 44 47	1 2 3 7 8 9 11 13 16 17 21 25 28 29 31 34 35 36 37 38 39 42 43 45 46 51 52	51
7	45 20 29 4 50 46	0.8831 0.7119 0.7102 0.6520 0.5045 0.4424	nd	45	45
8	10 47 30 37 34 25	0.8914 0.8741 0.7907 0.6181 0.4568 0.3878	7 10 11 13 16 18 19 20 23 30 47 50	1 4 7 10 13 20 24 25 28 29 31 34 35 43 44 46 47	16 10 47 7 30 13
9	nd	nd	nd	nd	nd
10	38 19 50 36 24 26	0.4743 0.2748 0.2723 0.1899 0.1761 0.1675	7 13 19 50	18	18
11	51 9 27 6 21 34	0.6658 0.5394 0.2029 0.1888 0.1883 0.1789	nd	9 51	51 9
12	22 11 35 40 16 21	0.5881 0.5442 0.3419 0.270 0.2347 0.2059	7 9 11 13 16 18 19 20 22 23 27 29 30 31 33 36 38 47 50	1 3 4 8 20 21 25 35 43 52	11 38 4 22
13	19 16 50 7 18 13	0.9651 0.8872 0.7976 0.6509 0.6312 0.5949	7 11 13 16 18 19 20 21 22 23 26 27 30 31 33 36 38 41 44 47 50	3 4 7 8 11 13 16 18 19 20 25 27 28 29 30 32 34 35 36 37 43 50 51	19 38 50 16 7 13
14	21 51 9 2 42 6	0.6249 0.4823 0.4454 0.2880 0.2671 0.2402	50	9 21 51	51 9 21
15	nd	nd	nd	nd	nd
16	50 19 27 29 5 3	0.8526 0.4840 0.3764 0.3377 0.3281 0.3278	7 13 18 19 20 50	16	19 80 18
17	21 2 42 9 51 6	0.6541 0.2516 0.2097 0.1422 0.1156 0.0904	7 9 11 13 16 18 19 21 30 50	9 21 51	21
18	22 11 13 7 16 38	0.7693 0.5793 0.4750 0.4519 0.3908 0.1635	4 7 9 11 13 14 16 20 22 23 24 26 27 28 29 30 33 35 36 39 41 46 47 50 51	6 7 12 13 16 17 19 21 23 25 27 28 30 32 33 35 36 37 38 41 42 43 45 46 48 51 52	43 51 46
19	5 46 34 41 6 30	0.5828 0.4188 0.3130 0.3032 0.2987 0.2787	nd	5	5
20	46 7 13 22 42 11	0.6063 0.2564 0.2514 0.2490 0.2290 0.2249	7 11 13 16 18 20 13 50	20 46	46
21	8 16 19 25 50 33	0.7612 0.3622 0.3516 0.3276 0.3140 0.3003	nd	nd	8

nd = no detectado

Fallo 7 Como ejemplo, se muestra ahora lo que ocurre en el Fallo 7 (pérdida de presión en la entrada del componente C). En la diagnosis *Temprana*, las variables candidatas más importantes son: 16, 7, 13, 4, 21 y 45 (para alta y moderada aversión al riesgo, $orness=0.2$ y 0.5). Para la diagnosis *Estacionaria* es la variable 45. En la figura 5.1 se puede ver como, después del instante 161, las variables 4, 7, 13, 16, 21 y 45 sufren un gran cambio.

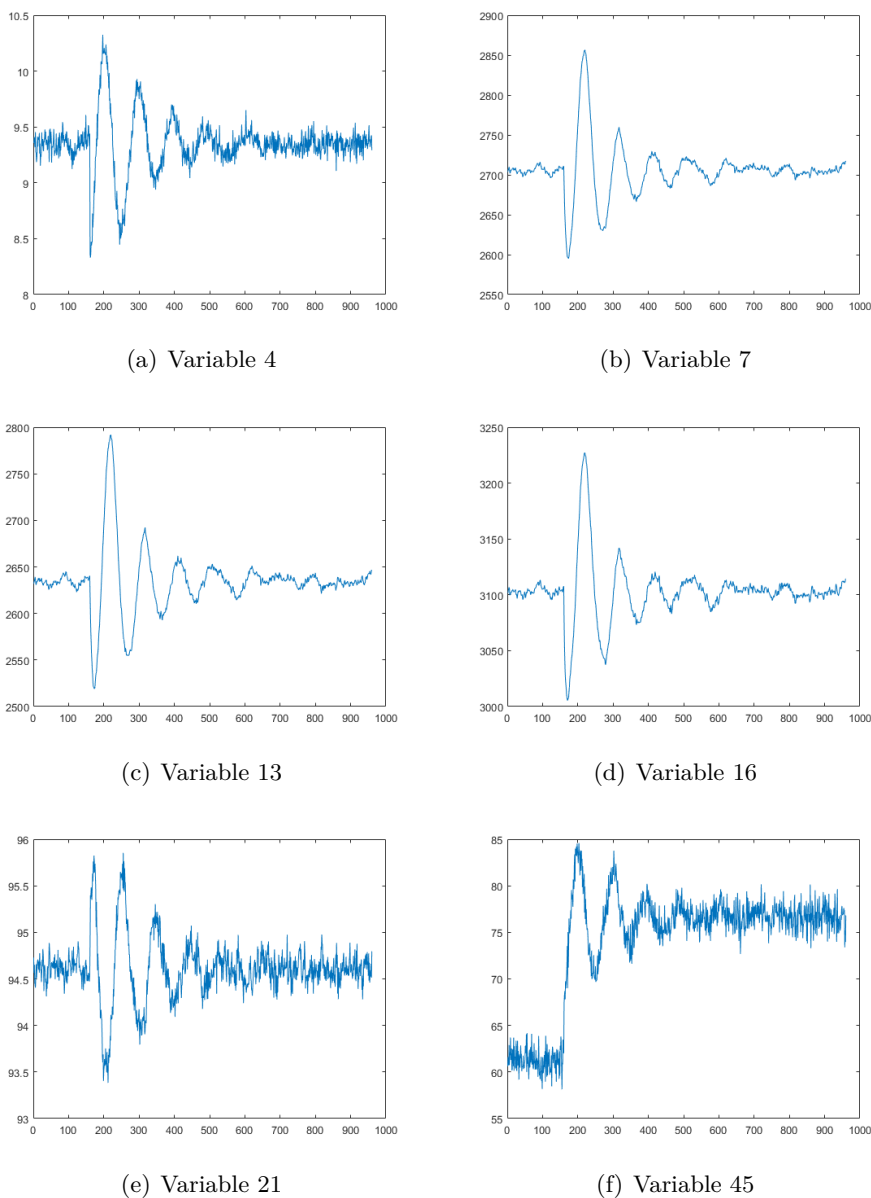


Figura 5.1: Fallo 7. Evolución de las variables candidatas.

Este fallo es un salto en la presión en la entrada 4 y el método indica que afecta especialmente a las variables 4, 16, 7, y 13, que miden: el caudal de la entrada 4 (variable 4), la presión en el *stripper* (variable 16), en el reactor (variable 7) y en el separador de producto (variable 13). Al no ser dichas presiones controladas por ningún sistema, varían proporcionalmente con el flujo de entrada que ha cambiado, al estar los citados elementos aguas abajo de la entrada 4. La válvula que corresponde con la variable 45 se abre para compensar la reducida disponibilidad del reactivo que se introduce por la entrada 4.

Algunos instantes más tarde, estas variable comienzan a retornar a sus valores normales, debido al efecto del sistema de control, excepto la variable 45, que se mantiene en valores anómalos. Esto se refleja en la diagnosis *Estacionaria* y es el comportamiento esperado de la planta, que cambia el ajuste de la válvula a la entrada para compensar la caída de presión en dicha entrada. Esto demuestra que el método propuesto es capaz de encontrar las variables más implicadas en el fallo, asignándoles una alta puntuación en el ranking, y descartar las demás variables, dándoles una puntuación baja. Esto hace más sencillo el proceso de diagnosis de fallos por parte de los operadores de la planta, y requiere de menor tiempo para hacer la citada diagnosis.

Una vez más, las dos ventanas de tiempo se muestran necesarias para diagnosticar los fallos, ya que algunas variables se ven afectadas en los primeros instantes de los fallos, y otras, al propagarse el fallo, cambian un cierto tiempo después. Si sólo se utilizase una de las dos ventanas (*Temporana* o *Estacionaria*) se perdería información valiosa sobre el estado de la planta, y, por tanto, esto sumado a la agregación de diferentes métodos de identificación da como resultado una diagnosis más robusta.

5.5. Caso de estudio II: Depuradora de Aguas Residuales

Se han probado los métodos propuestos en el modelo de simulación de la planta de tratamiento de aguas residuales descrita en el Apéndice A.2. Este modelo dispone de 16 conjuntos de datos con fallo, de los que se han utilizado para estos experimentos los datos de los fallos: 1 al 11, 15 y 16 (ver Tabla A.2). Además se dispone de datos sin fallo, para entrenar los métodos. Cada conjunto de datos contiene 58464 observaciones de 141 variables.

5.5.1. Metodología experimental

En este caso el método aplicado obtiene los pesos del operador OWA usando el cuantificador $Q(r) = r^\alpha$, tal como se indica en la ecuación 2.43, variando los valores de α para obtener operadores con diferentes valores de

orness, en función de la aversión al riesgo que quiera aplicar el operador.

Los parámetros usados para este caso de estudio se detallan en el Algoritmo 3. En primer lugar, se creó un modelo PCA utilizando datos sin fallo con un 70% de varianza capturada por los componentes principales seleccionados. Este valor se seleccionó después de hacer pruebas de validación cruzada en busca del mejor equilibrio entre el número de componentes principales y la varianza almacenada en dichos componentes principales. Valores más bajos no retenían una cantidad suficiente de información, mientras que valores más altos no aumentaban significativamente la varianza retenida, pero tomaban demasiados componentes principales. Los umbrales para los estadísticos T^2 y Q se ajustaron para que solo hubiera un 0,5% de observaciones por encima de ellos al analizar datos sin fallo, evitando falsas alarmas antes de que ocurriera el fallo.

Respecto al proceso de agregación basado en operadores OWA-RIM: los pesos se calcularon para una aversión al riesgo alta ($orness=0.2$), moderada ($orness=0.5$) y baja ($orness=0.9$). De esta forma se pueden hacer pruebas con tres políticas distintas de identificación. La ventana de identificación *Temprana* se ajustó a 2 instantes después de probar con diferentes valores (ver Algoritmo 3). El objetivo fue encontrar el mínimo tamaño de ventana que maximizase la separación de las puntuaciones del ranking OWA-RIM. Esto facilita distinguir las variables candidatas de las demás. En la Tabla 5.12, se muestran los resultados de las pruebas para longitudes de ventana de 2, 3 y 5 muestras.

Tabla 5.12: Ajuste de la longitud de la ventana *Temprana*

Aversión al riesgo	Alta ($orness=0.2$)			Moderada ($orness=0.5$)			Baja ($orness=0.9$)		
Tamaño ventana (muestras)	2	3	5	2	3	5	2	3	5
Separabilidad	0.0012	0.0011	0.00079	0.009	0.008	0.006	0.055	0.049	0.038

Los parámetros de la identificación *Estacionaria* se ajustan para detectar lo antes posible si se ha alcanzado un nuevo estado estacionario, pero asegurando que el sistema está realmente en situación estacionaria:

- $\Delta t_{Estacionaria}$ se ajustó a longitud 10. Valores más altos retrasaban demasiado la detección del estado estacionario y valores más pequeños detectaban falsos estados estacionarios de la planta.
- Las variables con puntuaciones mayores de $Th_{Estacionaria} = 0.5$ en el ranking final tienen que ser las mismas durante 10 observaciones consecutivas. Se probó con otros valores (ver Algoritmo 3), pero no se apreció diferencia entre ellos.
- La varianza entre las variables del punto anterior ($Var_{Estacionaria}$) debe permanecer por debajo de 0.001. Valores más bajos retrasaban de-

Algoritmo 3 *Metodología experimental*

```

1: Preproceso datos con fallo de entrenamiento: normalización Z-score
2: Modelo PCA: 70% de la varianza incluida, umbrales ajustados a per-
   centil 99,5.
3:
4: Ajuste de  $\Delta t_{Temprana}$  (Al detectar fallo):
5: for cada fallo do
6:   for RiskAversion=HIGH to LOW do
7:     for  $\Delta t_{Temprana} = 2$  to 10 do
8:       Calcular:  $Separabilidad=Varianza(Puntuaciones(\Delta t_{Temprana}))$ 
       $\triangleright$  Separación entre puntuaciones
9:     end for
10:    return ( $\Delta t_{Temprana}$ ) con  $min(\Delta t_{Temprana})$  y  $max(Separabilidad)$ 
11:  end for
12:
13: Ajuste  $\Delta t_{Estacionaria}$  (Planta en estado estacionario con fallo):
14: for cada fallo do
15:   for Aversión-al-Riesgo=ALTA to BAJA do
16:     for  $\Delta t_{Estacionaria} = 3$  to 20 do
17:       for  $Var_{Estacionaria} = 0.00001$  to 0.005 do
18:         for  $Th_{Estacionaria} = 0.3$  to 0.7 do
19:           Calcular:  $Retardo_{Estacionaria}$  y  $Detecciones_{Estacionaria}$ 
20:         end for
21:       end for
22:     end for
23:     return ( $\Delta t_{Estacionaria}$ ) y ( $Var_{Estacionaria}$ ) con
       $min(Retardo_{Estacionaria})$  y  $max(Detecciones_{estacionaria})$ .
24:   end for
25:
26: Detección e identificación de fallos:
27: Preproceso datos de test con fallo: normalización Z-score
28: if Fallo = TRUE then
29:   Realizar identificación fallo Temprana en  $\Delta t_{Temprana}$ 
30:   for cada instante do
31:     Comprobar estado estacionario
32:     if Estado Estacionario = TRUE then
33:       Realizar identificación fallo Estacionaria en  $\Delta t_{Estacionaria}$ 
34:     end if
35:   end for
36: end if

```

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

masiado la detección del estado estacionario, y valores más altos daban como resultado la detección del estado estacionario cuando éste aún no había ocurrido.

Las Tablas 5.13, 5.14 y 5.15 muestran algunas de las pruebas realizadas para ajustar estos parámetros: se prueban diferentes tamaños de ventana, y se muestra el retardo (en instantes) de la detección del estado estacionario y el número de conjuntos de datos con fallo en los que se consigue detectar correctamente dicho estado estacionario. Por ejemplo, en la Tabla 5.13, con un valor de $orness=0.2$ y una longitud de ventana de 5 muestras, el tiempo medio de detección del estado estacionario es de 9554 muestras, y el método detecta dicho modo estacionario en 10 de los 13 conjuntos de datos con fallo.

Tabla 5.13: Ajuste de la longitud de ventana *Estacionaria*, $Var.=0.0005$

Aversión al riesgo	Alta ($orness=0.2$)			Moderada ($orness=0.5$)			Baja ($orness=0.9$)		
Tamaño de ventana (muestras)	5	10	15	5	10	15	5	10	15
Retardo medio (muestras)	9554	10501	12542	3125	3406	4998	3457	12549	15988
Estados estacionarios detectados	10	12	11	12	12	10	12	13	13

Tabla 5.14: Ajuste de la longitud de la ventana *Estacionaria*, $Var.=0.001$

Aversión al riesgo	Alta ($orness=0.2$)			Moderada ($orness=0.5$)			Baja ($orness=0.9$)		
Tamaño de ventana (muestras)	5	10	15	5	10	15	5	10	15
Retardo medio (muestras)	1580	3501	6189	510	2406	4965	2385	11744	15885
Estados estacionarios detectados	10	13	13	12	13	13	12	13	13

Tabla 5.15: Ajuste de la longitud de ventana *Estacionaria*, $Var.=0.002$

Aversión al riesgo	Alta ($orness=0.2$)			Moderada ($orness=0.5$)			Baja ($orness=0.9$)		
Longitud de ventana (muestras)	5	10	15	5	10	15	5	10	15
Retardo medio (muestras)	1352	3412	3987	487	2105	4875	2133	10144	13578
Estados estacionarios detectados	10	12	12	11	12	12	11	12	12

5.5.2. Análisis de resultados

A continuación se muestran los resultados de la identificación de fallos. De cara a facilitar la comprensión de los resultados sólo se muestran las

6 primeras variables identificadas como candidatas y su puntuación en el ranking final.

Identificación de fallos *Temprana* Se presentan ahora los resultados de la identificación realizada en la ventana *Temprana*, que se ejecuta nada más detectarse el fallo. Para evaluar su desempeño se van a comparar los resultados de la propuesta con los obtenidos al aplicar un método PCA estándar con un 70% de varianza en los componentes principales seleccionados. Este PCA usa T^2 y Q para detectar fallos, y la contribución a Q (o Error normalizado de las variables) como método de identificación. En las Tablas 5.16, 5.17 y 5.18 se muestran los resultados de la identificación de los dos métodos, usando diferentes valores de *orness*. Como se puede ver nuestra propuesta da su diagnóstico un instante después que el PCA estándar, esto se debe a que dicha propuesta realiza la identificación en una ventana de tiempo de 2 instantes, mientras que el PCA sólo analiza el instante posterior a la detección del fallo. Como se ha dicho, el uso de la ventana de tiempo está justificado por el objetivo de aumentar la robustez frente a ruidos, perturbaciones, etc. como se puede ver en las Tablas 5.16, 5.17 y 5.18.

En esta identificación de fallos, se ve clara la diferencia entre los distintos niveles de aversión al riesgo: valores altos (es decir, criterios más estrictos) implican una alta capacidad de discriminación entre las variables candidatas, ya que sólo las candidatas con altas puntuaciones en la mayoría de los 7 métodos de identificación llegarán a posiciones altas en el ranking global con altas puntuaciones. Esto lleva a que las variables en las primeras posiciones del ranking obtengan una puntuación en el ranking mucho más alta que el resto, haciendo más fácil la identificación del fallo. Por tanto, hay menos riesgo de tener en cuenta variables irrelevantes en la diagnosis.

Por el contrario, si se utilizan valores de aversión al riesgo bajos, las puntuaciones finales obtenidas en el ranking son muy similares entre todas las variables, y se dificulta la selección de las variables candidatas. Esto ocurre porque a cualquier variable identificada como responsable por unos pocos métodos de identificación se le asigna una puntuación relativamente alta en el ranking final y acaba siendo tenida en cuenta en la diagnosis. En este caso, el nivel de riesgo es mayor ya que se pueden incluir variables poco relevantes en la identificación (el criterio aplicado es, ahora, más permisivo), pero también puede ocurrir que una variable realmente responsable del fallo que muy pocos métodos identifiquen sea tenida en cuenta en la diagnosis final y, por tanto, se puedan paliar los fallos de identificación de algunos de los 7 métodos utilizados.

Se pasa ahora a analizar los resultados concretos en cada fallo. Los primeros 5 fallos corresponden a un fallo en el sensor de oxígeno, y las variables en las que se provocó dicho fallo fueron la 42 y la 125, por tanto, si se ha

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

Tabla 5.16: Comparativa para la identificación *Temprana*. Alta aversión al riesgo ($orness = 0.2$)

Fallo	OWA-RIM			PCA	
	Retraso (instantes)	Ranking	Puntuación	Retraso (instantes)	Ranking
1	2	42 125 49 83 76 56	0.1230 0.06064 0.03825 0.02424 0.02424 0.02424	1	125 42 62
2	2	42 125 83 76 56 82	0.1354 0.0873 0.03634 0.03634 0.03634 0.0271	1	125 21 42
3	2	125 42 56 51 76 82	0.1490 0.08385 0.0103 0.00957 0.008966 0.008598	1	125 49 21
4	2	125 49 42 76 83 56	0.3705 0.1142 0.07138 0.05388 0.05388 0.05388	1	21 125 42 28 49 35
5	2	42 49 125 28 41 124	0.2395 0.1308 0.1174 0.03817 0.0136 0.01241	1	21 35 28 62 69 56
6	2	8 15 90 121 103 109	0.5316 0.5316 0.01553 0.0009643 0.000964 0.000964	1	8 15
7	2	15 8 90 21 83 56	0.4999 0.4915 0.007869 0.00421 0.004049 0.004049	1	8 15
8	2	8 15 90 102 120 96	0.5116 0.5116 0.0122 0.00170 0.00170 0.00170	1	8 15
9	2	22 55 75 21 7 14	0.3290 0.1676 0.1649 0.1552 0.1249 0.1249	1	22
10	2	22 29 21 55 75 7	0.3356 0.1017 0.05937 0.05614 0.05394 0.04435	1	22
11	2	22 29 36 21 75 43	0.8172 0.3294 0.1038 0.0206 0.01924 0.01746	1	22
15	2	86 80 8 15 14 7	0.7666 0.4341 0.02955 0.0295 0.02824 0.02817	1	86 80
16	2	86 80 8 15 14 7	0.6439 0.4271 0.03668 0.03668 0.02902 0.02902	1	86 80

realizado correctamente la identificación, estas variables deberían aparecer en las primeras posiciones del ranking. Esto es así en el caso de alta aversión al riesgo ($orness=0.2$, ver Tabla 5.16), en 3 de los 5 fallos las variables responsables aparecen en las primeras posiciones del ranking, y en los otros dos fallos, aparecen en la primera y tercera posición. Además, se observa que las puntuaciones de las variables en las primeras posiciones son claramente más altas que las demás, facilitando de esa forma encontrar el grupo de variables candidatas. En los 5 casos, las variables realmente responsables estaban dentro del grupo de candidatas. Por contra, el PCA estándar no consigue un 100% de efectividad en ninguno de los 5 fallos, dando el peor resultado en el fallo 5, en el que no consigue colocar las variables que originaron el fallo en las primeas posiciones.

En los casos de moderada aversión al riesgo ($orness=0.5$, ver Tabla 5.17) en estos primeros fallos, sí que aparecen las variables 42 y 125 en las primeras posiciones del ranking, pero las puntuaciones obtenidas por las variables

Tabla 5.17: Comparativa para la identificación *Temprana*. Moderada aversión al riesgo ($orness = 0.5$)

Fallo	OWA-RIM			PCA	
	Retraso (instantes)	Ranking	Puntuación	Retraso (instantes)	Ranking
1	2	42 125 83 76 56 55	0.3170 0.2638 0.2604 0.2604 0.2604 0.2525	1	125 42 62
2	2	42 125 83 76 56 62	0.3589 0.3107 0.2784 0.2784 0.2784 0.2616	1	125 21 42
3	2	125 42 76 56 44 62	0.4061 0.3388 0.2740 0.2704 0.2484 0.2454	1	125 49 21
4	2	42 125 83 56 49 62	0.5973 0.3016 0.30162 0.30162 0.2879 0.2724	1	21 125 42 28 49 35
5	2	21 42 125 76 83 56	0.4382 0.3406 0.3058 0.1978 0.1978 0.1978	1	21 35 28 62 69 56
6	2	8 15 97 103 115 121	0.5547 0.5547 0.08241 0.08241 0.08241 0.08240	1	8 15
7	2	15 8 83 76 76 56 7 14	0.5514 0.5481 0.1271 0.1271 0.1271 0.1179	1	8 15
8	2	8 15 102 96 120 108	0.5412 0.5412 0.08052 0.08052 0.08052 0.0804	1	8 15
9	2	22 55 75 7 14 82	0.7179 0.5036 0.4621 0.4392 0.4393 0.4269	1	22
10	2	22 29 55	0.73798 0.4041 0.3239	1	22
10	2	7 14 75	0.3021 0.3021 0.2853		
11	2	22 29 36 89 83 76	0.9350 0.5960 0.2729 0.1778 0.1702 0.1702	1	22
15	2	86 80 95 110 116 104	0.8795 0.6652 0.1302 0.1301 0.1301 0.1301	1	86 80
16	2	86 80 8 15 95 104	0.8319 0.6585 0.1446 0.1446 0.13103 0.13044	1	86 80

son muy similares entre sí, y dificultan mucho la elección de un grupo de variables candidatas a responsables del fallo, aunque en algunos fallos todavía se podría hacer dicha selección. Ya en el caso de los resultados de aplicar baja aversión al riesgo ($orness=0.9$, ver Tabla 5.18), las variables 42 y 125 aparecen cerca de las primeras posiciones del ranking, pero mezcladas con otras variables no relevantes, en algún caso (fallo 3) una de esas variables ni siquiera aparece entre las 6 primeras variables. Además, ahora, es completamente imposible encontrar un grupo de variables candidatas que tengan una puntuación claramente más alta que las demás, ya que las puntuaciones obtenidas son muy similares para todas las variables.

En el caso de los demás fallos (del 6 al 16), los resultados son buenos en todos los casos: las diferentes políticas de aversión al riesgo consiguen colocar en las primeras posiciones a las variables responsables del fallo. La única diferencia que hay entre los distintos niveles de $orness$ es que los valo-

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

Tabla 5.18: Comparativa para la identificación *Temprana*. Baja aversión al riesgo (*orness* = 0.9)

Fallo	OWA-RIM			PCA	
	Retraso (instantes)	Ranking	Puntuación	Retraso (instantes)	Ranking
1	2	55 59 62 69 42 125	0.825 0.8204 0.8141 0.8116 0.8089 0.7819	1	125 42 62
2	2	55 59 62 125 79 42	0.8264 0.8211 0.8174 0.8149 0.8096 0.7877	1	125 21 42
3	2	62 55 69 59 83 42	0.8417 0.8408 0.8392 0.830 0.8264 0.8263	1	125 49 21 125 49 21
4	2	55 59 62 42 79 125	0.8223 0.8162 0.8162 0.8138 0.8049 0.7941	1	21 125 42 28 49 35
5	2	62 69 42 76 125 79	0.7812 0.7791 0.7588 0.7588 0.7588 0.6158	1	21 35 28 62 69 56
6	2	8 15 115 97 103 109	0.6444 0.6443 0.4078 0.4078 0.4078 0.4078	1	8 15
7	2	15 8 76 83 56 95	0.6388 0.6383 0.4788 0.4787 0.4787 0.4774	1	8 15
8	2	8 15 95 98 104 110	0.6358 0.6358 0.3900 0.3898 0.3897 0.3897	1	8 15
9	2	22 55 75 59 79 7	0.9469 0.9019 0.8302 0.8220 0.8121 0.8045	1	22
10	2	22 55 59 79 75 7	0.9558 0.8471 0.7870 0.7783 0.7735 0.7649	1	22
11	2	22 29 89 56 83 76	0.9878 0.7877 0.7303 0.7205 0.7204 0.7205	1	22
15	2	86 80 95 116 104 110	0.9461 0.8462 0.6884 0.6877 0.6877 0.6877	1	86 80
16	2	86 80 95 104 110 116	0.9396 0.8397 0.6792 0.6762 0.6762 0.6762	1	86 80

res más bajos de este parámetro (baja aversión al riesgo) consiguen otorgar unas puntuaciones a las variables de las primeras posiciones del ranking claramente más altas que el resto de variables, y esto facilita mucho la elección del grupo de candidatas, formado justamente por las variables responsables del fallo. Si se adopta una política de aversión al riesgo baja (*orness*=0.9) si sigue consiguiendo colocar a las variables realmente responsables en las primeras posiciones, pero con unas puntuaciones más próximas al resto de variables. En cualquier caso, en estos fallos todavía las puntuaciones de las primeras variables son lo suficientemente altas como para poderlas separar del resto. En estos fallos, el PCA estándar también consigue colocar en las primeras posiciones a las variables responsables, pero el método propuesto da una información muy valiosa para escoger a las variables candidatas de entre todas las demás.

En resumen, el uso de la agregación de métodos de diagnosis en una ventana de tiempo proporciona una diagnosis más robusta, encontrando las variables responsables de los fallos y discriminándolas más claramente de las demás, evitando perturbaciones instantáneas, etc. siempre y cuando el método adopte una estrategia de alta o moderada aversión al riesgo. Por otro lado, como el PCA estándar utilizado aplica la contribución a Q , que es uno de los 7 métodos de identificación usados en el método ORAFI, si dicha contribución no funciona bien en un fallo, el ORAFI es capaz de compensarlo utilizando la identificación proporcionada por los 6 métodos restantes, y esto hace al ORAFI más robusto.

De cara a cuantificar el desempeño de cada diagnosis, la Tabla 5.19 muestra la tasa de éxito de cada una de ellas. Esta tasa depende de la inclusión de las variables responsables del fallo en el grupo de candidatas encontradas, y de la capacidad de separar dichas candidatas del resto de variables (usando las puntuaciones obtenidas).

Tabla 5.19: Tasas de éxito de cada diagnosis

Calificación	Puntuación	Condiciones
Excelente	100	Sólo variables responsables entre las candidatas
Buena	67	Las variables responsables entre las candidatas
Regular	33	No todas las responsables entre las candidatas o no hay grupo de candidatas, pero las responsables están en las primeras posiciones
Mala	0	No hay variables responsables entre las candidatas o no hay grupo de candidatas

Tabla 5.20: Tasas de éxito en la diagnosis *Temprana*

Fallo	OWA-RIM			PCA
	Aversión al riesgo			
	Alta	Moderada	Baja	
1	100	33	33	67
2	100	100	33	67
3	100	33	33	33
4	67	33	33	33
5	67	67	67	0
6	100	100	100	100
7	100	100	100	100
8	100	100	100	100
9	100	100	100	100
10	100	100	100	100
11	100	100	100	100
15	100	100	100	100
16	100	100	100	100
Tasa media	94.5	80.5	74.92	76.92

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

La Tabla 5.20 muestra que el método propuesto da un mejor resultado que el PCA estándar cuando se aplican políticas de alta o moderada aversión al riesgo. La discriminación de las variables candidatas al tomar baja aversión al riesgo es peor que en los dos casos anteriores, y en el caso de los 5 primeros fallos, las variables responsables del fallo ni siquiera aparecen en las primeras posiciones del ranking, dando como resultado una mala diagnosis.

Identificación de fallos *Estacionaria* La diagnosis en la ventana *Estacionaria* es la segunda diagnosis del método propuesto. El retardo respecto a la detección del fallo en ofrecer dicha diagnosis se muestra en la Tabla 5.21 y se obtiene cuando se ha detectado que el sistema ha llegado a un punto de trabajo estacionario después de aparecer el fallo. Se ha hecho una comparativa de tiempos de diagnosis entre la identificación *Estacionaria* y otros métodos como son los propuestos por [Liu, 2012, Liu and Chen, 2014], que utilizan todos los datos disponibles. Y el resultado es que el método propuesto, ORAFI, es capaz de dar la diagnosis mucho antes que los demás. Esto es así porque se dispone, en este caso de estudio, de 58464 muestras disponibles, apareciendo el fallo en el instante 18800, y como los métodos de [Liu, 2012, Liu and Chen, 2014] usan todos los datos disponibles, su diagnosis se demora 39664 instantes, mientras que el método ORAFI trata de encontrar un instante a partir del cual ya no haya cambios en el sistema y no merezca la pena seguir tomando datos.

Tabla 5.21: Retardo en la diagnosis *Estacionaria* (en instantes). BSM2

Fallo	Aversión al riesgo (<i>orness</i>)		
	Alta (0.2)	Moderada (0.5)	Baja (0.9)
1	759	211	13630
2	113	4782	8282
3	396	6042	8281
4	15001	210	27743
5	1236	788	788
6	82	62	72
7	76	64	286
8	64	64	86
9	670	210	2476
10	278	122	288
11	1348	110	1344
15	104	100	146
16	150	130	286
Media	1559.77	991.92	4900.65
Desv. Típ.	4062.73	1987.5	8109.9

En el caso de aplicar una política de moderada aversión al riesgo (*orness* = 0.5) se obtienen los menores valores medios y de desviación típica del retardo, mientras que usando un valor de baja aversión al riesgo (*orness* = 0.9) se obtiene el peor resultado. Esto es debido a que un mayor número de variables aparecen como candidatas en el ranking durante el proceso de

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

detección del estado estacionario, y, por tanto, de un instante a otro sea más fácil que pueda haber cambios en el ranking. Esto hace que el retardo en detectar dicho estado estacionario sea mayor. En cualquier caso, el retardo en ofrecer la diagnosis usando el método ORAFI es considerablemente menor que si se utilizasen todos los datos disponibles, como hacen los métodos de [Liu, 2012, Liu and Chen, 2014].

Los resultados para esta diagnosis *Estacionaria* se muestran en las Tablas 5.22, 5.23 y 5.24, en función del valor de aversión al riesgo escogido. Además se muestra una comparativa con el método RCI ([Liu and Chen, 2014]), que es uno de los siete métodos de identificación usados en la agregación ORAFI, que, como se ha dicho, emplea todos los datos a partir del fallo contenidos en el dataset para hacer la identificación.

Tabla 5.22: Diagnosis *Estacionaria*. Alta aversión al riesgo ($orness = 0.2$)

Fallo	OWA-RIM							RCI
	Ranking	Puntuaciones						Variabes
1	42 125 62 69 56 76	0.6730	0.6730	0.6190	0.3389	0.3089	0.3089	42 125 21
2	125 42 62 69 56 76	0.9425	0.8653	0.4239	0.3859	0.35856	0.3586	125 42
3	125 42 113 107 101 49	0.7182	0.4253	0.3779	0.3364	0.3103	0.1717	125 42
4	125 28 42 21 89 56	0.2187	0.1679	0.1183	0.1043	0.0465	0.0244	42 125
5	56 76 83 62 69 49	0.7761	0.7761	0.7761	0.6982	0.693	0.4796	21 28 56 76 83 62
6	8 15 90 36 22 43 63	0.9698	0.9698	0.0340	0.0073	0.0068	0.0054	8 15
7	8 15 90 121 109 103	0.9025	0.9025	0.0211	0.0026	0.0026	0.0026	8 15
8	8 15 90 29 102 120	0.9320	0.9320	0.0268	0.0023	0.0022	0.0022	8 15
9	50 70 63 43 36 84	0.7272	0.72588	0.7242	0.7191	0.7180	0.6917	63 70 84 50 70 36
10	36 43 50 63 70 77	0.8274	0.8211	0.8003	0.7882	0.7876	0.7506	50 77 43 63 70 36
11	84 77 57 63 70 36	0.9069	0.9069	0.9069	0.8969	0.8915	0.8359	84 77 57 63 70 36
15	80 86 113 101 107 64	0.6917	0.0982	0.0417	0.0417	0.0417	0.0087	28 80
16	80 86 107 113 101 90	0.6528	0.2410	0.0347	0.0347	0.0347	0.0254	80 28

Tabla 5.23: Diagnosis *Estacionaria*. Moderada aversión al riesgo ($orness = 0.5$)

Fallo	OWA-RIM							RCI
	Ranking	Puntuaciones						Variabes
1	62 69 56 76 83 21	0.4508	0.4028	0.3550	0.3550	0.3550	0.3482	42 125 21
2	62 69 56 76 83 21	0.5367	0.4584	0.4027	0.4027	0.4027	0.3022	125 42
3	109 115 97 103 121 96	0.2817	0.2817	0.2817	0.2817	0.2817	0.2812	125 42
4	56 83 76 62 69 7	0.4148	0.3951	0.3951	0.3649	0.3643	0.2739	42 125
5	21 56 76 62 69 83	0.8140	0.6709	0.6709	0.6245	0.6234	0.6047	21 28 56 76 83 62
6	8 15 115 103 109 97	0.9813	0.9813	0.1339	0.1339	0.1339	0.1339	8 15
7	8 15 102 96 108 120	0.9385	0.9385	0.1353	0.1353	0.1353	0.1353	8 15
8	8 15 97 103 109 115	0.9734	0.9734	0.1440	0.1440	0.1440	0.1440	8 15
9	36 43 50 84 77 57	0.8018	0.80	0.7776	0.7609	0.7609	0.7609	63 70 84 50 70 36
10	36 43 22 50 29 84	0.7879	0.7706	0.7360	0.7262	0.7136	0.5620	50 77 43 63 70 36
11	84 77 57 36 63 70	0.9028	0.9028	0.9028	0.9005	0.8905	0.8860	84 77 57 63 70 36
15	80 86 113 101 107 95	0.8941	0.2863	0.1986	0.1986	0.1986	0.1465	28 80
16	80 107 101 113 86 95	0.9013	0.2396	0.2396	0.2396	0.2394	0.1646	80 28

CAPÍTULO 5. FDI MULTIMÉTODO BASADA EN OPERADORES DE AGREGACIÓN

Tabla 5.24: Diagnósis *Estacionaria*. Baja aversión al riesgo ($orness = 0.9$)

Fallo	OWA-RIM							RCI
	Ranking	Puntuaciones						
1	62 69 76 56 83 80	0.8552	0.8542	0.8020	0.8018	0.8016	0.6534	42 125 21
2	101 107 113 50 97 103	0.8182	0.8182	0.8182	0.8005	0.7607	0.7607	125 42
3	101 107 113 97 103 109	0.8207	0.8207	0.8207	0.8052	0.8052	0.8052	125 42
4	69 62 56 76 83 49	0.7557	0.7554	0.7288	0.7280	0.7257	0.7033	42 125
5	21 56 76 62 69 83	0.8140	0.6709	0.6709	0.6245	0.6234	0.6047	21 28 56 76 83 62
6	8 15 115 97 103 109	0.9911	0.9911	0.6566	0.6566	0.6566	0.6566	8 15
7	8 15 98 104 110 116	0.9905	0.9905	0.7534	0.7534	0.7534	0.7534	8 15
8	8 15 103 109 115 121	0.9906	0.9906	0.7788	0.7788	0.7788	0.7788	8 15
9	84 77 57 63 36 70	0.9229	0.9229	0.9229	0.9050	0.9029	0.9013	63 70 84 50 70 36
10	84 77 57 63 70 36	0.9721	0.9721	0.9721	0.9623	0.9574	0.9496	50 77 43 63 70 36
11	70 63 84 77 57 50	0.9643	0.9640	0.9626	0.9626	0.9626	0.9440	84 77 57 63 70 36
15	80 95 110 116 104 98	0.9897	0.7542	0.7511	0.7511	0.7511	0.7507	28 80
16	80 95 110 104 116 98	0.9918	0.7483	0.7443	0.7443	0.7443	0.7436	80 28

Una vez que el sistema ha entrado en modo estacionario después de aparecer el fallo, éste puede afectar a más variables aparte de las responsables de dicho fallo debido a que determinados tipos de fallos se expanden, llegando a afectar a varias áreas del sistema. En este estudio, los fallos 1 a 5 y 9 a 11 tienen dicha característica de expansión del fallo, al contrario que los fallos 6 a 8 y 15 y 16, en los que las variables afectadas por el fallo no cambian a lo largo del tiempo.

En el caso de aplicar una aversión al riesgo alta ($orness=0.2$, ver Tabla 5.22), algunas variables no relacionadas con el fallo aparecen como candidatas en los 5 primeros fallos (el origen de estos fallos está en las variables 42 y 125, relacionadas con el sensor de O_2). Concretamente, las variables que aparecen son aquellas que miden el nivel de O_2 en diferentes puntos de la instalación (variables: 21, 49, 56, 62, 69, 76, 83, 89) o el nivel de Demanda Química de Oxígeno (DQO) (variables: 101, 107, 113), que está relacionado también con el O_2 . Este resultado es similar al obtenido por el método RCI. Analizando las puntuaciones obtenidas, se ve que ya no hay tanta diferencia entre ellas como en el caso de la identificación *Temprana*, esto nos indica que el fallo está afectando a cada vez más variables. Aún así, es posible hacer una diagnóstico de los fallos excepto para el quinto, donde todas las variables candidatas tienen puntuaciones similares.

El fallo afecta a otras partes de la planta en el caso de los fallos 9 a 11, y el resultado de la diagnóstico refleja este hecho ya que las puntuaciones son muy similares y hay muchas variables candidatas. Ahora el aislamiento de las variables que originaron el fallo no es posible, al igual que ocurre con el método RCI. En cualquier caso, al igual que antes, nos sirve para constatar la propagación del fallo por la instalación.

Los otros fallos no sufren un destacable cambio en esta diagnóstico, ya que analizan datos de unos tipos de fallo que no se expanden. Se observa,

además, que las diferencias entre las puntuaciones obtenidas por las variables candidatas y las demás se han incrementado con respecto al caso de la identificación *Temprana*, en la mayoría de casos. Las variables candidatas para los fallos 15 y 16, se identifican claramente al estar en las primeras posiciones del ranking y tener una puntuaciones más altas que las del resto. Esto confirma que el fallo sólo afecta al punto de origen del mismo y no se propaga a otras partes de la planta. Las diagnósicos del método ORAFI coinciden con lo que realmente sucede en la planta, mientras que los resultados del método RCI no colocan a las variables responsables (80 y 86) en las primeras posiciones e identifican a otras variables como responsables. Esto confirma la robustez del método ORAFI.

Si se analizan los resultados de aplicar moderada aversión al riesgo ($orness=0.5$, ver Tabla 5.23) y baja aversión al riesgo ($orness=0.9$, ver Tabla 5.24) se observa que ocurre algo similar al caso anterior, las puntuaciones, en general, se han igualado y esto hace difícil la selección de variables candidatas. Y en los fallos que se expanden por la planta, cada vez aparecen más variables entre las candidatas, de hecho, en los fallos 1 a 5 ninguna de las variables realmente responsables aparece entre las seis primeras. En los fallos 6 a 8 sí se identifica correctamente a las responsables y se puede extraer un grupo de candidatas. Mientras que en los fallos 15 y 16 las responsables están mezcladas con más variables y obtienen puntuaciones similares entre ellas, dificultándose la identificación.

Resumiendo, el hecho de trabajar con datos pertenecientes a un intervalo de tiempo largo (todos los datos disponibles, por ejemplo) no conduce a tener una diagnósico confiable. Esto sucede en los casos de fallos que se expanden por la planta y terminan alterando los valores de otras variables. Por tanto, en el caso de fallos expansivos, el origen del fallo sólo puede ser localizado si la diagnósico se realiza durante los primeros instantes que siguen a la aparición del fallo, que es lo que hace la identificación *Temprana* del método ORAFI.

En estas pruebas, la identificación *Estacionaria* para los fallos 1 a 5 y 9 a 11 no ha servido para encontrar las causas del fallo, ya que nos encontramos con muchas variables candidatas y no todas ellas son realmente las responsables, y, en algunos casos, ni siquiera es posible extraer un grupo de candidatas. En los fallos 15 y 16, como el fallo no se extiende por la planta, la diagnósico ORAFI es capaz de localizar el origen del fallo, al contrario que con el método RCI que no da una buena diagnósico. Como en una situación real, no se puede saber si el próximo fallo que va a suceder va a ser de tipo expansivo o no, se requiere aplicar dos identificaciones: una inicial (*Temprana*), que servirá para encontrar el origen del fallo, y una tardía (*Estacionaria*), cuando el sistema alcanza un nuevo estado estacionario, que servirá para confirmar la diagnósico anterior o dará un resultado diferente y por tanto, constatará que el fallo se ha expandido y cómo se ha propagado por la planta.

5.6. Resumen

En este capítulo se ha presentado un método para hacer la diagnosis distribuida de una planta, donde en cada nodo se calcula un método de identificación de fallos FDI: Diagnosis de Fallos basada en Agregación OWA-RIM (ORAFI). Las principales aportaciones del método propuesto son: la agregación de las diagnosis de varios métodos FDI mediante una toma de decisiones multicriterio, y la realización de dichas diagnosis en dos instantes.

La agregación de diferentes métodos permite crear una sinergia entre ellos, de modo que las debilidades de unos métodos se compensen con las fortalezas de los otros. Por tanto, se puede obtener una identificación más robusta.

Además, el método propuesto, permite adoptar diferentes políticas en cuanto al grado de riesgo que se quiere asumir. En el ranking final de variables, según el riesgo asumido, se tendrán más o menos en cuenta las diagnosis de los distintos métodos.

En el caso de que no se quiera asumir un gran riesgo, el método sólo colocará en las primeras posiciones del ranking aquellas variables seleccionadas como candidatas por casi todos los métodos (aunque esto puede hacer que se pierda alguna variable en el proceso). Pero si se quiere asumir un riesgo mayor en la identificación, el método puede incluir en las primeras posiciones del ranking las variables que cuenten con buena puntuación en solo unos pocos métodos (con el riesgo de introducir variables poco relevantes en la identificación).

Por otro lado, se ha establecido que la diagnosis se haga en dos pasos: el primero se ejecuta justo después de la detección del fallo, y facilita la identificación del origen del fallo, mientras que el segundo diagnóstico se realiza cuando el sistema alcanza un punto de operación estacionario después de aparecer el fallo, y brinda información al usuario sobre las nuevas condiciones de trabajo en las que se encuentra el sistema y permite decidir si se puede continuar operando la planta o es necesario llevar a cabo reparaciones. Ambos diagnósticos se desarrollan usando ventanas de tiempo, que buscan evitar los efectos del ruido, perturbaciones, etc., haciendo que el método sea más robusto.

El método ORAFI se ha probado en dos de los casos de estudio abordados en esta tesis: una planta de tratamiento de aguas residuales y la planta Tennessee Eastman Process, mostrando su eficacia en la realización de los diagnósticos y confirmando que ambos diagnósticos (*Temprana* y *Estacionaria*) son necesarios para realizar correctamente el diagnóstico de la planta.

Capítulo 6

FDI BASADO EN EL ANÁLISIS DE SERIES TEMPORALES Y CONTROL ESTADÍSTICO MULTIVARIANTE. DISTRIBUCIÓN DEL CÁLCULO

6.1. Introducción

En este capítulo presentaremos una nueva propuesta de detección de fallos basado en datos que tiene en cuenta algunas características de los datos industriales: son dinámicos, están correlacionados (no todas las variables son independientes) y además son no-lineales. Para tener en cuenta todas estas consideraciones presentaremos una nueva metodología de detección de fallos dinámica y no-lineal, que tiene en cuenta explícitamente las relaciones dinámicas de la planta, utilizando una selección dinámica de variables, así como la naturaleza no lineal del proceso a través del modelado de las variables mediante el análisis de series temporales.

Se va a proceder a hacer un modelo de cada variable medida del proceso en función de las demás. Como los procesos industriales tienen muchas variables, es imposible modelar todas ellas al mismo tiempo en un único procesador, por lo que en este caso para disminuir el tiempo de procesamiento se distribuirá el cálculo de dichos modelos entre varios nodos. Luego se calcula la desviación del modelo obtenido para cada variable respecto a los datos reales medidos en la instalación, calculando los residuos, que serán procesados mediante técnicas de control estadístico de procesos. Por tanto, en este caso, la descentralización se aplica a los cálculos necesarios para aplicar el método FDI.

Luego cada módulo envía sus resultados a un nodo central que es el encargado de analizarlos en su conjunto y decidir sobre la situación en la

que se encuentra la planta: normal o con fallo.

Esta metodología, por lo tanto, consta de tres pasos: selección dinámica de variables, modelado mediante el análisis de series temporales y detección de fallos usando las herramientas del SPC. Vamos ahora a explicar brevemente que herramientas pueden utilizarse para resolver estos tres pasos.

6.2. Fundamentos teóricos

Para llevar a cabo esta propuesta, si en la instalación industrial considerada se toman medidas de varias variables, se pueden construir modelos de series temporales de cada una de ellas. Si hay m variables, se hace necesario identificar m modelos en el sistema. Concretamente, para cada variable $x_i, i = 1, \dots, m$ del sistema, se quiere encontrar un modelo de predicción basado en una selección del resto de variables que consiga la mejor predicción posible de la variable i -ésima. Los modelos de series temporales deben reflejar la influencia de los estados previos del sistema en el estado actual del mismo, por tanto, el modelo de predicción debe tener un carácter dinámico. Para ello, los modelos deben incluir un número suficiente de datos con retardo de las m variables del sistema, buscando siempre tener el mejor modelo de predicción posible. Por tanto, las entradas de cada modelo serán un conjunto de datos actuales de variables además de otro conjunto de datos pasados de dichas variables. Por tanto, para una variable i , el modelo será:

$$x_i(t+1) \sim [x_1(t), \dots, x_1(t-k_1), x_2(t), \dots, x_2(t-k_2), \dots, x_m(t), \dots, x_m(t-k_m)] \quad (6.1)$$

donde $\{k_1, k_2, \dots, k_m\} \subseteq \{0, \dots, L\}$. L es el máximo número de retardos que se utilizará para el modelado y debe ser ajustado mediante un proceso de selección dinámica de características. Dicho lo cual, en las entradas de los modelos nos podemos encontrar con datos de variables sin retardo, datos con diferentes retardos de la misma variable y/o datos con diferentes retardos de distintas variables.

6.2.1. Selección dinámica de características

Como se ha dicho, el modelo de predicción de una variable va a tener como entradas datos pasados de dicha variable y datos pasados o no del resto de variables del sistema. La selección de qué variables y retardos usar se puede llevar a cabo mediante una selección dinámica de características basada en la autocorrelación y la correlación cruzada con diferentes retardos de tiempo. Para ello, a partir de la matriz \mathbf{X} (n muestras \times m variables) que contiene las variables sin retardo en condiciones normales de operación, se

genera la matriz aumentada, \mathbf{X}_a , tomando para cada variable datos actuales y datos con retardo de hasta L instantes y se apilan de la siguiente manera:

$$\mathbf{X}_a = [\mathbf{X}_t | \mathbf{X}_{t-1} | \dots | \mathbf{X}_{t-L}] \quad (6.2)$$

donde $|$ es el operador de concatenación de matrices, la matriz aumentada $\mathbf{X}_a \in \mathbb{R}^{(n-L) \times (m(L+1))}$, \mathbf{X}_t es la matriz de datos \mathbf{X} sin retardo y \mathbf{X}_{t-d} es la matriz de datos con un retardo de d instantes. Realmente, la forma de construir esta matriz aumentada consiste en unir $(L+1)$ matrices \mathbf{X} , una al lado de la otra, desplazando cada matriz una fila con respecto a la anterior, y luego se eliminan las filas incompletas, como se ve en la figura 6.1.

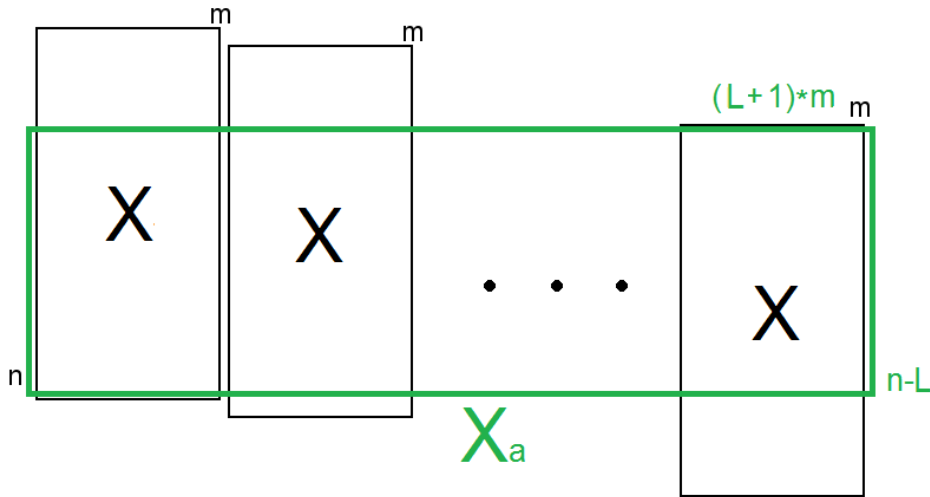


Figura 6.1: Esquema de la matriz aumentada.

Para implementar la selección dinámica de características, la relación entre dos variables x_t^i y x_t^j en dos instantes de tiempo diferentes se calcula como el valor absoluto del coeficiente de correlación:

$$\mathfrak{R}_i(\mathbf{x}_t^i, \mathbf{x}_t^j) = \left| x_t^{iT} x_t^j / \|x_t^i\| \|x_t^j\| \right| \quad (6.3)$$

donde $i, j = 1, \dots, m$ y $t, l = 1, \dots, L$. Este coeficiente es una medida directa de la correlación entre variables y, después del cálculo del vector \mathfrak{R}_i para la variable i -ésima, la selección dinámica de características se realiza eligiendo las variables de la matriz aumentada \mathbf{X}_a que tengan valores altos de correlación. Por ejemplo, las variables usadas para modelar la variable i se corresponderán con aquellos elementos del vector \mathfrak{R}_i que verifiquen que $\mathfrak{R}_i > \delta_i$, siendo δ_i un umbral establecido por el usuario para cada una de las variables y que dependerá de los valores de correlación obtenidos.

Esta selección dinámica de características se puede hacer también utilizando el conocido método *Partial Least Squares* (PLS) [James et al., 2013]. El método PLS desarrolla un modelo lineal, basado en mínimos cuadrados, que obtiene una serie de nuevas variables (o características), que son combinaciones lineales de los datos de partida, como se explicó en la Sección 4.2.1.2.

Una vez completada la creación del modelo se obtiene la ecuación de regresión PLS, que es:

$$\mathbf{Y} = \mathbf{X} * \mathbf{B}_{PLS} + \mathbf{F} \quad (6.4)$$

donde \mathbf{F} es la matriz de residuos y B_{PLS} es el vector de coeficientes de regresión. De modo que ahora, se construye un modelo PLS para cada variable i en el que la matriz de predictores es la matriz aumentada \mathbf{X}_a (Ecuación 6.2), sin la correspondiente variable i -ésima, y la de respuestas es $\mathbf{Y} = \mathbf{x}_i$, $i = 1, \dots, m$. La selección dinámica de variables se lleva a cabo escogiendo las variables de la matriz \mathbf{X}_a con un coeficiente de regresión grande, es decir, con un alto valor en el vector \mathbf{B}_{PLS} .

Una tercera forma de realizar la selección dinámica de características es mediante la combinación de ambos métodos, es decir, primero, realizar una selección dinámica de características con el coeficiente de correlación entre las variables y, después de eso, refinar la selección utilizando el método PLS. Entonces, las variables y sus valores retardados con mayor coeficiente de correlación se seleccionan y se incluyen en la matriz de datos \mathbf{X}_i , creando una para cada variable i del sistema, con $i = 1, \dots, m$. Después, se genera un modelo de regresión PLS para la variable i -ésima con la matriz de datos anterior, y aquellas variables con un alto valor de coeficiente de regresión en valor absoluto se eligen como variables de entrada finales del modelo i -ésimo.

En el caso particular de aplicar modelos ARIMA, en los que una variable se modela sólo usando datos retardados de dicha variable, la selección dinámica de variables se realiza usando las funciones de Autocorrelación y de Autocorrelación Parcial [Shumway and Stoffer, 2006].

6.2.2. Modelado de series temporales

El modelado de series temporales ha sido y es objeto de estudio de varios investigadores [Gooijer and Hyndman, 2006, Hyndman and Athanasopoulos, 2017], que buscan el método más adecuado para describir y predecir dichas series. Uno de los métodos más ampliamente utilizado es el modelo Autorregresivo Integrado de Media Móvil (ARIMA) o metodología *Box-Jenkins* [Box and Jenkins, 1976, Hyndman and Khandakar, 2008].

ARIMA. Este modelado se basa en dos modelos más sencillos: el modelo Autorregresivo (AR) y el modelo de Media Móvil (MA), que, combinados,

dan lugar al modelo Autorregresivo de Media Móvil (ARMA). En el caso de que la serie temporal a modelar no sea estacionaria, se incluye una parte de diferenciación, dando lugar al modelo ARIMA ([Box and Jenkins, 1976, Shumway and Stoffer, 2006, Hyndman and Athanasopoulos, 2017]). Si se da el caso de trabajar con una serie temporal que muestra comportamiento estacional (en inglés, *seasonal*) se utiliza un modelo *Seasonal* ARIMA (SARIMA), que modela, también, la parte estacional de la serie.

El modelo AR tiene la siguiente forma:

$$x_t = c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + e_t \quad (6.5)$$

siendo c una constante, ϕ_j es el coeficiente del dato retardado j instantes de la serie temporal, y e_t es el error cometido en el instante t . Por tanto, la predicción se obtiene a partir de los valores previos de la variable.

En el caso de un modelo MA, su expresión es la siguiente:

$$x_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} \quad (6.6)$$

siendo c una constante, e_t es el error cometido en tiempo t , θ_j es el coeficiente del error cometido al predecir x_j con el modelo MA. En este caso, se hace la predicción en función de los errores de predicción de los instantes anteriores.

Si se combinan los dos anteriores se obtiene el modelo ARMA, cuya expresión es:

$$x_t = c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} + e_t \quad (6.7)$$

Si la serie no es estacionaria, se aplica una diferenciación, que, al ser incluida, da lugar al modelo ARIMA:

$$x_t^d = c + \phi_1 B x_{t-1}^d + \dots + \phi_p B x_{t-p}^d + \theta_1 B e_{t-1} + \dots + \theta_q B e_{t-q} + e_t \quad (6.8)$$

siendo B el operador desplazamiento que cumple que: $Bx_t = x_{t-1}$ y $B^2 x_t = B(Bx_t) = x_{t-2}$. Además, x_t^d es la serie diferenciada un cierto número d de veces.

El orden del modelo ARIMA se define por la terna de parámetros: (p, d, q) . Siendo p el número de términos de la parte AR, d es el número de diferenciaciones realizadas y q es el orden de la parte MA del modelo [Hyndman and Athanasopoulos, 2017].

Los valores de p y q se determinan a través del análisis de las gráficas de Autocorrelación y Autocorrelación parcial, mientras que el orden de la parte diferencial, d , es el necesario para conseguir que la serie sea estacionaria [Hyndman and Athanasopoulos, 2017]. La forma más usual de determinar si una serie temporal o su expresión diferenciada es estacionaria es mediante

el uso de diferentes test estadísticos, como el test Dickey-Fuller [Dickey and Fuller, 1979].

La metodología más difundida para ejecutar el modelado de una serie temporal es la *Box-Jenkins* [Box and Jenkins, 1976], cuyo esquema se presenta en la Figura 6.2) y consta de cuatro pasos:

1. El primer paso es la **identificación** del modelo ARIMA que sigue la serie, esto requiere:
 - Comprobar el carácter estacionario (mediante tests) de la serie, y decidir las transformaciones necesarias para hacerla estacionaria en caso de que no lo sea (diferenciación).
 - Obtener un modelo ARMA para la serie ya estacionaria: órdenes p y q , usando las gráficas de autocorrelación y autocorrelación parcial.
2. El segundo paso es la **estimación** de los parámetros de los modelos AR y MA del modelo. Estos se pueden estimar usando el criterio de máxima verosimilitud [Hyndman and Athanasopoulos, 2017], el criterio de Información de *Akaike* (AIC) [Akaike, 1974], etc. Finalmente, se obtienen los errores estándar y los residuos del modelo.
3. El tercer paso, de **validación**, sirve para comprobar que los residuos son independientes y tienen estructura de ruido blanco. Si no fuera así, se modifica el modelo y se repiten las etapas anteriores hasta obtener un modelo adecuado.
4. Finalmente, el último paso es el de **predicción**. Ya con el modelo adecuado, se pueden obtener estimaciones del valor futuro de la variable.

Se puede dar el caso de que la serie presente estacionalidad, es decir, el comportamiento de la serie se repite cada cierto número de observaciones m . Esto se puede modelar con un modelo *Seasonal* ARIMA (SARIMA). En este caso, se modela una parte no estacional y otra parte estacional, por lo que hay que seleccionar los valores para los órdenes de cada una de las partes del modelo: p , d y q , para la no estacional, y P , D y Q , para la estacional [Hyndman and Athanasopoulos, 2017].

Se pueden usar otros modelos no lineales para predecir la serie temporal en función de los valores actuales y pasados del resto de variables del sistema, como pueden ser los basados en Redes Neuronales [Sainz et al., 2004, Khashei and Bijari, 2010, Kuhn and Johnson, 2013, Heidari et al., 2016], Bosques Aleatorios (RF) [Tsagkrasoulis and Montana, 2018, Jog et al., 2017] o Regresión de Máquinas de Vector Soporte (SVR) [Vapnik, 1995, Zhou et al., 2011, Zaidi, 2012, Nieto et al., 2013, Zhang and Gao, 2018], ya explicados en el Capítulo 4, todos ellos siguen la misma metodología explicada en la figura 6.2.

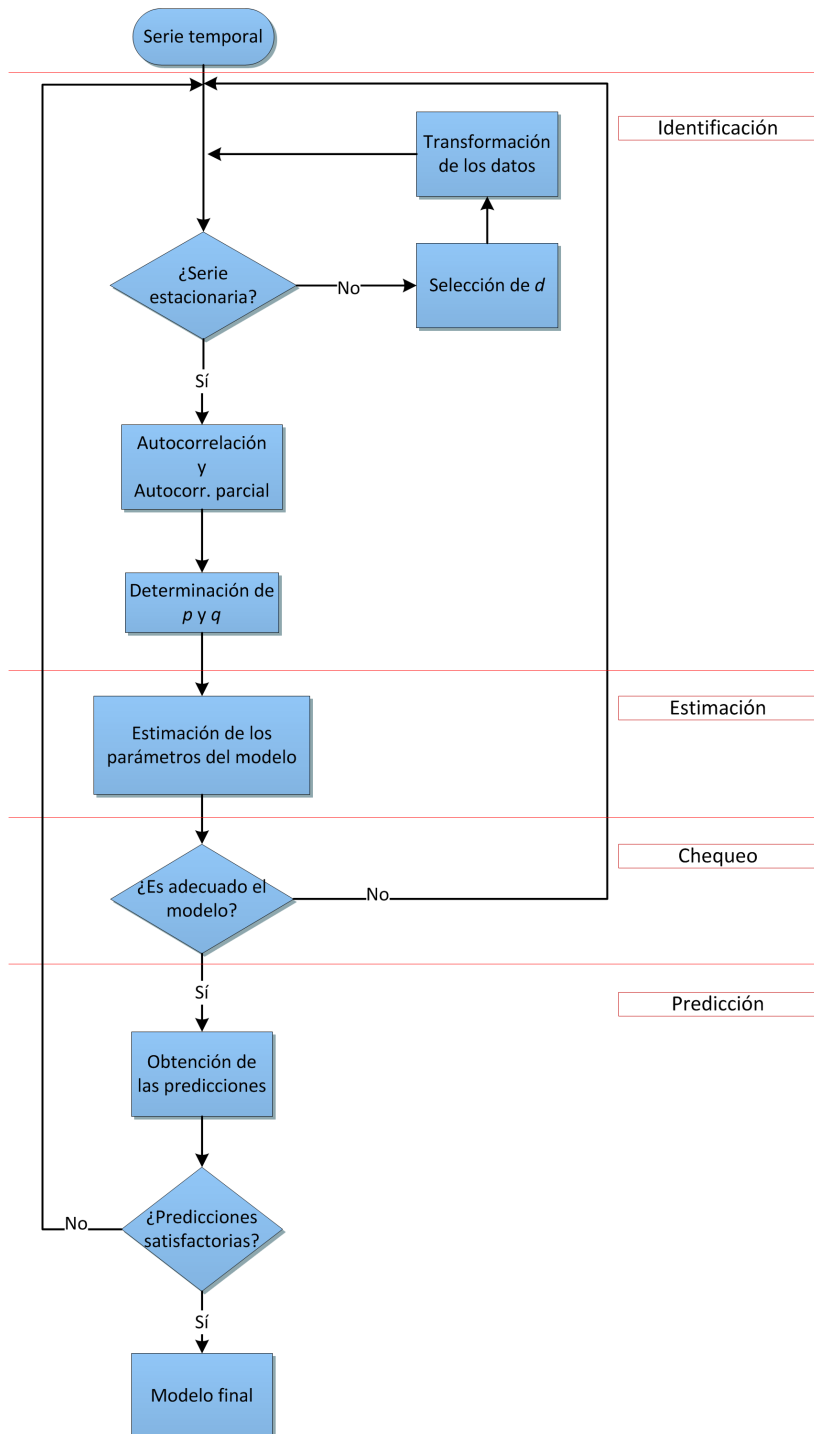


Figura 6.2: Metodología *Box-Jenkins*

6.2.3. Detección de fallos basada en las gráficas de control de SPC

Una vez que se han obtenido los modelos de cada una de las variables, es posible hacer predicciones con dichos modelos y compararlas con los valores reales medidos en la planta. De esta forma se obtienen los residuos en cada uno de los nodos, que corresponden a cada una de las variables. Es posible ir evaluando los valores de dichos residuos para determinar si existe o no un fallo, utilizando gráficas de control como las mostradas en el Capítulo 2, pero en una planta compleja nos encontramos con muchas variables y puede ser complejo realizar dicha tarea, así que en este capítulo se proponen dos alternativas: por un lado, filtrar los residuos y comprobar si alguno supera su correspondiente umbral y, por otro lado, procesar esos residuos mediante un modelo PCA, que entregue unos pocos indicadores (T^2 y Q) que simplifiquen la tarea de la detección de fallos.

6.3. Método FDI propuesto: Detección de fallos basada en series temporales y control estadístico de procesos: TSR-FDI

El objetivo principal del método es obtener una técnica de detección de fallos que tenga en cuenta el carácter dinámico de la planta monitorizada, así como su no linealidad. Características éstas que son esperables en casi cualquier planta industrial de cierto nivel de complejidad. La dinámica de la planta va a ser analizada mediante un proceso de selección dinámica de características, mientras que el comportamiento no lineal será tratado a través de un modelado no lineal basado en series temporales.

Por tanto, inicialmente se cuantificará lo que se influyen las variables entre sí, y luego se utilizará esta información para crear un modelo, basado en series temporales, de cada una de las variables, utilizando aquellas que tengan mayor influencia.

Posteriormente, se utilizarán los modelos para hacer predicciones que serán comparadas con los datos medido en la planta. Los residuos obtenidos se enviarán a un procesador central para ser analizados. Será este procesador el que determine si existe o no un fallo.

En la Figura 6.3 y en el Algoritmo 4 se muestran esquematizadas las diferentes etapas de las que consta el método que se pasan a detallar a continuación.

Etapas 1. Selección dinámica de características. Como se ha comentado, las variables en una planta industrial no suelen ser independientes entre si. Existe correlación cruzada, pero además es esperable que exista correlación temporal entre ellas, es decir, el estado actual de una variable va

CAPÍTULO 6. FDI BASADO EN ANÁLISIS DE SERIES TEMPORALES

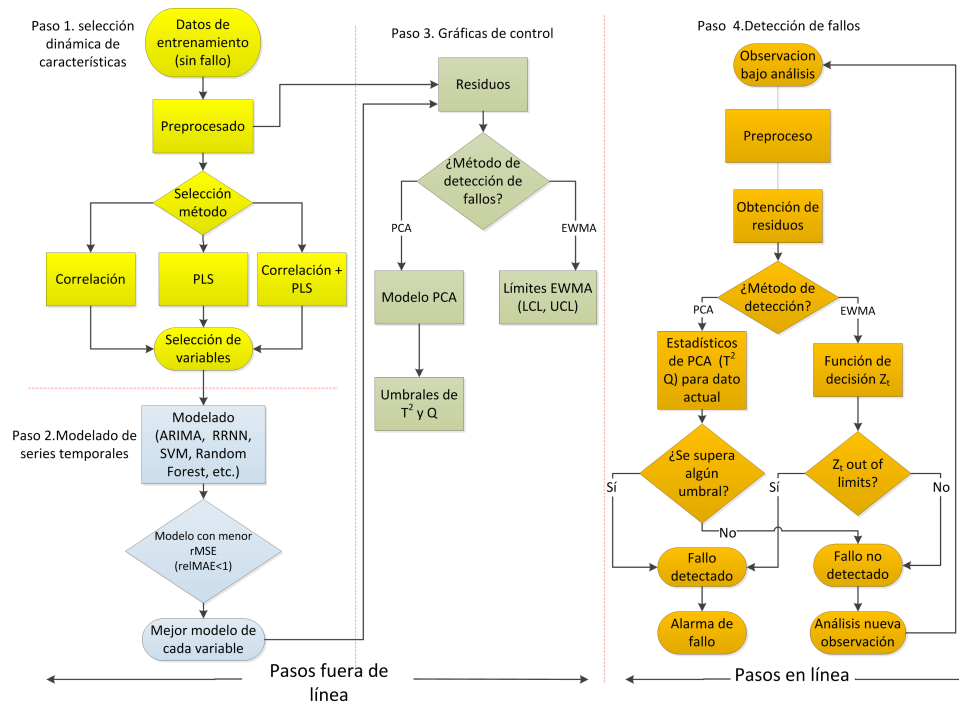


Figura 6.3: Diagrama de flujo del método propuesto

a depender no sólo del estado actual de la planta y sus variables sino de los estados pasados. Por tanto, se presenta un método de selección dinámica de características que analiza la correlación temporal y espacial, teniendo en cuenta diferentes retardos de tiempo. Los pasos que componen esta primera etapa son:

1. **Recopilación de datos de entrenamiento:** tomados en condiciones normales de operación de la planta.
2. **Preprocesado de los datos:** que consta de un análisis previo para eliminar valores atípicos, y, después, un escalado de los datos al rango adecuado.
3. **Selección de los retardos de tiempo:** se toman los más adecuados para modelar cada variable, es decir, la selección dinámica de características, para ello se utilizarán los diferentes métodos presentados en la Sección 6.2.1, es decir, correlación, PLS o una combinación de ambos métodos

Etapa 2. Modelado de las series temporales. En la segunda etapa se construye un modelo descriptivo de la planta utilizando el conjunto de

Algoritmo 4 *Detección de fallos basada en series temporales: STR-FDI*

```

1: for Datos sin fallo, off-line do
2:   Etapa 1. Selección dinámica de características
3:   Normalizar datos.
4:   Hacer la selección dinámica de características para cada variables mediante
   3 métodos: correlación, PLS y una combinación de ambos.
5:   Etapa 2. Modelado de series temporales
6:   Modelar las series temporales de cada variable con: ARIMA, ARMAX, NN,
   SVR, random Forest, etc.
7:   Escoger el mejor modelo para cada variable, mediante rMSE y relMAE.
8:   Etapa 3. Obtención de los límites de las gráficas de control
9:   Obtener predicciones de cada variable. ▷ Usando cada modelo seleccionado
10:  Obtener residuos sin fallo:
11:     $Residuo^{SF}(t) = Observación(t) - Predicción(t)$ .
12:    Entrenar  $PCA_{residuos}$ . ▷ Modelo PCA con  $Residuos^{SF}(t)$ 
13:    Definir los límites UCL y LCL para las gráficas de control.
14:    Definir los umbrales  $T_\alpha^2$  y  $Q_\alpha$  para los estadísticos  $T^2$  y  $Q$ .
15: end for
16:
17: Etapa 4. Detección de fallos
18: Analizar nuevos datos, on-line.
19: for t=1 to n do ▷ Para cada instante
20:   Obtener y normalizar una nueva observación:  $x_n(t)$ .
21:   for i=1 to m do ▷ Para cada variable m
22:    Hacer la selección dinámica de características de la etapa 1.
23:     $Predicción^i(t) : \hat{x}_n^i(t) = f(modelo^i, (t, t-1, t-2, \dots))$ 
24:     $Residuo^i(t) = x_n^i(t) - \hat{x}_n^i(t)$ 
25:   end for
26:
27:   Detección de fallos basada en EWMA:
28:   for i=1 to m do ▷ Para cada variable m
29:    Calcular la función de decisión  $z^i(t) = \lambda Residuo^i(t) + (1 - \lambda)z^i(t-1)$ .
30:    if  $z^i(t) > UCL$  then
31:      $Fallo = TRUE$  en instante  $t$ .
32:    end if
33:   end for
34:
35:   Detección de fallos en PCA de residuos:
36:   Calcular  $T^2(t)$  y  $Q(t)$  para  $Residuo(t)$ .
37:   if  $T^2(t) > T_\alpha^2$  or  $Q(t) > Q_\alpha$  then
38:     $Fallo = TRUE$  en instante  $t$ .
39:   end if
40: end for

```

datos de entrenamiento, en condiciones normales de la planta, definido en la primera etapa. Inicialmente se realiza otro preprocesado de los datos, que consiste en el análisis (de tendencias y de estacionalidad de la serie temporal) y la normalización de los datos. Luego, estos datos se dividen en

dos grupos, creando unos datos de entrenamiento y unos datos de prueba. Después, se procede a construir diferentes modelos para cada variable del sistema usando los datos de entrenamiento, con la idea de hallar el mejor modelo descriptivo usando las variables seleccionadas en la etapa 1 como predictoras. Para evaluar el nivel de ajuste del modelo se van a utilizar algunas medidas del error cometido al hacer predicciones. Estas medidas de error son:

- **rMSE:** es la raíz cuadrada del error cuadrático medio (MSE).
- **sMAPE:** es el valor medio del valor absoluto del porcentaje de error.

$$sMAPE = \frac{2}{n} \sum_{t=1}^n \frac{|F_t - A_t|}{(|F_t| + |A_t|)} * 100\% \quad (6.9)$$

donde n es el número de muestras, F_t es la predicción obtenido en el instante t y A_t es valor real de la variable en ese instante.

- **relMAE:** es el error medio absoluto relativo, y se obtiene dividiendo el valor medio del error absoluto por el valor medio del error absoluto obtenido por otro modelo utilizado como referencia. En este trabajo, el modelo de referencia usado ha sido el modelo llamado *predictor ingenioso*, que supone que la próxima observación será igual a la observación actual.

$$relMAE = \frac{MAE}{MAE_p} \quad (6.10)$$

donde MAE es el error medio absoluto del modelo probado y MAE_p es la misma medida, pero del modelo usado como referencia.

El modelo con el valor rMSE más bajo será el seleccionado para hacer la predicción de cada variable, siempre que el relMAE sea inferior a 1, es decir, que el modelo en cuestión es mejor que el modelo de referencia. La otra medida de error: sMAPE se utilizará para conocer el rendimiento del modelo seleccionado.

Etapla 3. Obtención de los límites de las gráficas de control. Se calculan los límites de cada gráfica de control utilizada. El procedimiento se basa en analizar los residuos, que son las diferencias entre los valores predichos para cada variable y los valores reales de las variables. Se plantea la posibilidad de usar dos técnicas de control estadístico distintas:

- **Gráficas de control EWMA:** estas gráficas trabajan con las variables individualmente, es decir, se obtiene una gráfica de control para cada residuo de cada variable. En cada gráfica se establecen unos valores límite: límite de control superior (UCL) y límite de control inferior

(LCL), que no deben ser superados cuando el sistema trabaja sin fallos. Estos límites se calculan mediante la Ecuación 2.2.

- **PCA:** todos los residuos se procesan juntos, y se utilizan dos gráficas para detectar anomalías. Primero, se calcula un modelo de PCA utilizando los residuos en condiciones normales, y, después se obtienen los umbrales para los estadísticos (T^2 y Q) según se explica en la Sección 2.5.

Las tres etapas anteriores se ejecutan fuera de línea (*off-line*).

Etapla 4. Detección de fallos. Esta etapa, mostrada en el Algoritmo 4, se calcula en línea (*on-line*) y se subdivide en los siguientes pasos:

1. **Toma de datos de la planta bajo análisis:** buscando determinar si trabaja en condiciones de fallo o no.
2. **Preprocesado de los datos:** que se hace de la misma forma que con los datos de entrenamiento. Además, se ejecuta la selección dinámica de características explicada en el etapa 1 de este proceso.
3. **Generación de los residuos para cada variable:** que son la diferencia entre las mediciones y las predicciones de los modelos de series temporales construidos en la etapa 2.
4. **Calculo de la función de decisión $z_i(t)$:** tal como se indica en la Ecuación 2.1, para cada residuo, que se usará con la gráfica de control EWMA. Calculo de los estadísticos T^2 y Q con las Ecuaciones 2.9 y 2.12, respectivamente, para las gráficas de control del PCA.
5. **Detección de fallos:** se comparan los valores obtenidos en cada gráfica de control con sus umbrales correspondientes. Si en algún caso ($z_i(t)$, T^2 o Q) se superan los correspondientes umbrales durante una serie de observaciones consecutivas (*Obs*), se avisa de la presencia de un fallo. El establecer un número consecutivo de observaciones antes de declarar una alarma se hace para evitar falsas alarmas. Si no se detecta un fallo (es decir, ninguna de las tres gráficas de control excede sus límites), se toma una nueva observación y se repite esta cuarta etapa del proceso de monitorización.

Por tanto, el método propuesto basa la detección de fallos en los residuos de los modelos de las variables. Estos residuos son cercanos a cero cuando el comportamiento del sistema monitorizado es normal, siempre que se disponga de buenos modelos. Sin embargo, cuando ocurre una situación anormal o un fallo, los residuos se desvían significativamente de cero, lo que indica la presencia de un nuevo estado de la planta. En esta propuesta, el control

de dichos residuos se lleva a cabo de dos formas: gráficas EWMA y PCA. Tal como se ha comentado, al usar este método, las técnicas dinámicas de selección de características así como el modelado no lineal de las variables, se elimina tanto el carácter dinámico como la no linealidad de la instalación. Esto hace que esta propuesta pueda ser aplicada en gran cantidad de plantas e instalaciones industriales, que presentan dichos caracteres dinámico y no lineal.

6.4. Caso de estudio I: Tennessee Eastman Process

La primera planta en la que se ha puesto a prueba esta metodología es la *Tennessee Eastman Process* [Downs and Vogel, 1993], explicada en el Apéndice A.3 y utilizada en capítulos anteriores.

6.4.1. Metodología experimental

Las operaciones fuera de línea presentadas en el Algoritmo 4 se han realizado usando los datos de entrenamiento sin fallo, con los que se construye la matriz de datos: $\mathbf{X} \in \mathbb{R}^{500 \times 52}$.

1. **Selección dinámica de características.** El objetivo es seleccionar qué variables con y sin retardo deben ser usadas para modelar las variables en la etapa siguiente. Esto se realizó usando los tres métodos explicados en la Sección 6.2.1: análisis de correlación, extracción de características con PLS y una combinación de los dos anteriores. Para ello se analizaron las variables sin retardo y las variables con retardos desde 1 a 15 instantes. Por tanto, en la matriz ampliada, (Ecuación 6.2), se estableció el valor $L = 15$. Una vez hecho el análisis, se seleccionaron, para cada variable, las 10 variables con retardo y/o sin retardo, que más influencia tuvieron en dicho análisis. Este valor se eligió, después de varias pruebas, buscando evitar sobreajustes en los modelos de la etapa siguiente, especialmente al modelar con redes neuronales.
2. **Modelado de series temporales.** Se usaron los datos sin fallo de entrenamiento (500 observaciones) para modelar cada variable con las variables (con y/o sin retardo) seleccionadas en el paso anterior. Dichos datos se normalizaron según el tipo de modelo: para las redes neuronales se normalizaron al rango $[0,2, 0,8]$, y no al más común $[0,1]$, para dejar cierto margen para observaciones extremas que puedan tomarse luego en los datos de test; para modelar con ARIMA, se normalizó a media cero, y se diferencié la serie en el caso de no ser estacionaria, etc. Finalmente, se generaron varios modelos para cada variable, con los siguientes parámetros:

- **Modelos ARIMA:** cuya estructura es $ARIMA(p,q,i)$. Para hacer pruebas, se hicieron variar estos valores entre 0 y 15 para p y q , y entre 0 y 2, para i , con la limitación de que: $p + q \leq 20$.
- **Modelos de Redes Neuronales:** el tipo de red usado fue el de red Perceptrón Multicapa (MLP) con 3 capas, la primera capa con 10 entradas, correspondientes a las 10 variables seleccionadas en la etapa anterior, y la capa de salida con una sola neurona. El número de neuronas en la capa oculta se fue variando entre los valores: $neuronas\ en\ capa\ oculta = [5, \dots, 6 \cdot (2 \cdot entradas)]$ hasta dar con el mejor modelo.
- **Regresión SVM con kernel Gaussiano:** ($k(u, v) = \exp(-\gamma|u - v|^2)$). Aquí es necesario ajustar dos parámetros C y γ , que se hicieron variar entre los siguientes valores: $C = [0,1, 1, 10, 100, 1000]$ y $\gamma = [0,01, 0,1, 1, 10, 100]$.
- **Regresión SVM con kernel sigmoide:** ($k(u, v) = \tanh(\gamma * u^T v + coef)$). Para este modelo los parámetros C y γ se eligieron entre los valores: $C = [0,1, 1, 10, 100, 1000]$ y $\gamma = [0,01, 0,1, 1, 10, 100]$.
- **Modelos Random Forest:** en este caso el número de variables muestreadas aleatoriamente oscilaron entre los siguientes valores: $[200, 300, 400]$, y para el número de árboles se probó con los valores $[1500, 2000]$. Este último valor no debe ser muy bajo para asegurarse que cada variable de entrada es predicha un número suficiente de veces.

Una vez que se han generado todos estos modelos para cada variable, se elige aquel que presente el menor valor de error rMSE. En la Tabla 6.1 se muestran los modelos finalmente elegidos para cada una de las 52 variables. Los modelos del tipo ARIMA se muestran con sus órdenes p , q e i . Los modelo identificados como ANN son redes neuronales con selección de variables basada en correlación y los PLS ANN son también redes neuronales pero haciendo la selección con PLS. Existen algunas variables con modelo ARIMA y orden $(0,0,0)$, que son variables consideradas como ruido blanco y que no se han podido modelar, de modo que el modelo usado para obtener residuos es una constante cuyo valor es la media de la variable en condiciones sin fallo.

3. **Obtención de los límites de las gráficas de control.** Una vez se dispone de un modelo para cada variable se pueden hacer predicciones para cada una de ellas y generar unos residuos al evaluar la diferencia entre la predicción y el valor real de cada variable. Inicialmente se generan estos residuos para los datos sin fallo, que servirán para establecer los límites de los gráficos de control que se van a usar:

CAPÍTULO 6. FDI BASADO EN ANÁLISIS DE SERIES
TEMPORALES

Tabla 6.1: Modelo de serie temporal seleccionado para cada variable

Variable	Modelo	Variable	Modelo
1	ANN: Capa oculta:26	27	ARIMA: (0,0,0)
2	ANN: Capa oculta: 13	28	ARIMA: (0,1,0)
3	ANN: Capa oculta: 6	29	ARIMA: (2,2,0)
4	ANN: Capa oculta: 13	30	ARIMA: (1,1,0)
5	ANN: Capa oculta: 13	31	ARIMA: (1,3,0)
6	ANN: Capa oculta: 8	32	PLS ANN: Capa oculta: 3
7	ANN: Capa oculta: 23	33	ARIMA: (2,0,0)
8	ANN: Capa oculta: 5	34	ARIMA: (1,1,0)
9	ANN: Capa oculta: 5	35	ARIMA: (1,1,0)
10	ANN: Capa oculta: 12	36	ARIMA: (1,0,0)
11	ANN: Capa oculta: 5	37	ARIMA: (0,0,0)
12	ANN: Capa oculta: 10	38	ARIMA:(1,2,0)
13	ANN: Capa oculta: 13	39	ARIMA: (0,1,0)
14	ANN: Capa oculta: 7	40	ARIMA: (0,0,0)
15	ANN: Capa oculta: 5	41	ARIMA: (0,0,0)
16	PLS ANN: Capa oculta: 3	42	ANN: Capa oculta: 20
17	ANN: Capa oculta: 24	43	ANN: Capa oculta: 16
18	ARIMA: (2,2,1)	44	ANN: Capa oculta: 7
19	PLS ANN:Capa oculta: 8	45	ANN: Capa oculta: 24
20	ARIMA: (2,2,0)	46	ANN: Capa oculta: 13
21	ANN: Capa oculta: 7	47	ANN: Capa oculta: 11
22	ANN: Capa oculta: 15	48	ANN: Capa oculta: 11
23	ARIMA: (0,1,0)	49	ANN: Capa oculta: 9
24	ARIMA: (1,1,0)	50	ARIMA: (1,2,1)
25	ARIMA:(3,1,0)	51	ANN: Capa oculta: 21
26	PLS ANN: Capa oculta: 4	52	ANN: Capa oculta: 15

a) **Gráfica de control EWMA sobre residuos (EWMAres).**

A partir de la Ecuación 2.2 se calculan los umbrales, que se ajustan posteriormente de forma experimental probando varios valores λ y obteniendo el mínimo número de observaciones anómalas consecutivas (*Obs*) para activar la alarma de fallo. Los valores seleccionados después de algunos experimentos fueron $\lambda = 0,7$ y $Obs = 6$. Con esta combinación se obtiene el mejor resultado en términos de tasa de falsas alarmas (FAR), tasa de no detección de fallos (MDR) y tiempo de detección mínimo, como se puede ver en los resultados.

b) **Detección de fallos basada en PCA de los residuos (PCAres).** Inicialmente se crea un modelo PCA utilizando los residuos calculados. Este PCA contiene un 85 % de variación de los datos

contenida en los componentes principales seleccionados (es un valor al que se ha llegado después de varias pruebas y representa el compromiso entre maximizar la información contenida en los componentes principales y minimizar el número de ellos). A continuación se calculan los umbrales para los estadísticos T^2 y Q , teniendo en cuenta que es necesario un cierto número consecutivo de observaciones que excedan el umbral (Obs) para detectar un fallo. Los umbrales de estos estadísticos se calculan teóricamente como se explica en [Jolliffe, 2002] y, después, se ajustan experimentalmente para un nivel de significación o α de 1%. Este valor representa el porcentaje esperado de alarmas para el sistema en condiciones normales de operación. El valor de $Obs = 3$ se ajustó experimentalmente buscando un nivel de falsas alarmas de 0% en los estadísticos T^2 y Q al analizar los datos de entrenamiento sin fallo.

6.4.2. Análisis de resultados

Se presentan a continuación los resultados obtenidos en la etapa 4 de la metodología: Detección de fallos. Para evaluar el desempeño del método se ha hecho una comparativa del mismo con varios métodos estadísticos multivariantes aplicados a la planta Tennessee Eastman Process: PCA, PCA dinámico (DPCA), Análisis de Variables Canónicas (CVA) (usando los estadísticos T_s^2 , T_r^2 y Q) [Russell et al., 2000], y *Kernel PCA* (KPCA) ([Zhang, 2008]). La comparativa se realiza en términos de los índices de rendimiento utilizados en todos los métodos desarrollados en esta tesis: número de fallos detectados (mejor cuantos más se detecten), tasa de falsas alarmas (FAR), tasa de no detección de fallos (MDR) (mejor cuanto más bajo sea este valor) y retardo en la detección de cada fallo (también es mejor cuanto más bajo).

En la Tabla 6.2 están contenidos el número de fallos detectados por cada uno de los métodos. Al analizar los resultados se comprueba que el método propuesto es el que da los mejores resultados, ya que la alternativa basada en EWMA (*EWMAres*) es capaz de detectar 20 de los 21 fallos considerados, a la vez que la propuesta basada en un modelo PCA de los residuos (*PCAsres*) es capaz de detectar los 21 fallos, utilizando el índice Q . Por el contrario, el resto de métodos de la comparativa dan un peor resultado, ya que sólo son capaces de detectar entre 16 y 18 fallos.

La robustez de los métodos se puede comprobar mediante la tasa de falsas alarmas (FAR), también conocida como error de tipo I. En este caso de estudio se ha calculado la tasa FAR a partir de los datos de entrenamiento sin fallo (500 observaciones), ajustando los umbrales para la detección de fallos, para todos los métodos, a un nivel igual al parámetro α : 1%. Posteriormente, se obtienen las tasas de falsas alarmas (FAR) con los datos de test sin fallo (que constan de 960 muestras), que se muestran en la Tabla 6.3.

Tabla 6.2: Fallos detectados por cada método. TEP

	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2
Fallos detectados	16	18	17	18	18	18
	CVA Q	KPCA T^2	KPCA Q	EWMA- res	PCAs T^2	PCAs Q
Fallos detectados	16	18	18	20	17	21

Tabla 6.3: Tasa de falsas alarmas (FAR), en porcentaje. TEP

Método y estadístico	Datos de entrenamiento	Datos de test
PCA T^2	0.2	1.4
PCA Q	0.4	1.6
DPCA T^2	0.2	0.6
DPCA Q	0.4	28.1
CVA T_s^2	1.3	8.3
CVA T_r^2	0	12.6
CVA Q	0.9	8.7
KPCA T^2		1.5
KPCA Q		2
EWMA s	0.94	1.6
PCAs T^2	0.3	0.39
PCAs Q	1.04	1.06

Se puede observar que la tasa mínima de falsas alarmas para los datos de test la consigue *PCAs* con T^2 con un valor de 0,39%, mientras que el método *DPCA* con *Q* consigue el valor más alto de la comparativa: 28,1%, que es un valor realmente alto. En el caso del método *CVA* las tasas de falsas alarmas obtenidas son muy altas con cualquiera de sus estadísticos, ya que oscilan entre 8,3% que da usando T_s^2 y el 12% que entrega el método *CVA* con T_r^2 . Quedándose en un dato de 8,7% para el estadístico *Q*.

Es conveniente recalcar que para que una alarma de fallo se active, tanto con *EWMA*s como con *PCAs*, se requieren de 3 a 6 observaciones anómalas consecutivas, por tanto, los valores de error de Tipo I obtenidos por nuestra propuesta son aceptables, ya que el FAR se calcula contando observaciones individuales, y cuando no hay fallo, estas observaciones anómalas no son consecutivas y, por tanto, no se avisa del fallo.

De hecho los valores de *Obs* y de los umbrales se han ajustado para que no haya avisos de falsas alarmas en los datos sin fallo de test. Por lo tanto, la tasa efectiva de falsas alarmas de los métodos propuestos es 0 %.

En la Tabla 6.4 se muestran, en porcentaje, los valores de la tasa de no detección de fallos (MDR), para cada uno de los métodos y para cada uno de los fallos. Los resultados recogidos en dicha tabla muestran que los métodos propuestos (*EWM*Ares, *PC*Ares usando el estadístico Q) son capaces de superar a los otros métodos: PCA, DPCA, KPCA y CVA (con T_s^2 y Q) en la mayoría de fallos. El método CVA con T_r^2 también ofrece buenos resultados respecto a las tasas MDR en muchos de los fallos, no quedando muy lejos de los métodos *EWM*Ares y *PC*Ares cuando usan Q , exceptuando los fallos 5, 10 y 19, en los que los resultados que se consiguen son muy diferentes.

Sin embargo, en los fallos 8, 11, 13 y 18, el rendimiento de detección obtenido es superior usando los métodos propuestos. Incluso se obtiene un buen rendimiento en los fallos 3, 9 y 15, que son fallos muy difíciles de detectar. Esto indica que el método propuesto parece ser más capaz de detectar los fallos que los demás métodos. Además, hay que tener en cuenta, como se ha dicho anteriormente, que el método CVA da niveles de falsas alarmas bastante altos, y esto puede ser consecuencia de tener unos umbrales de detección muy bajos.

Este último hecho puede facilitar la detección de fallos y la obtención de tasas bajas de MDR, al hacer que el método sea muy sensible a pequeños cambios. La contrapartida es que la presencia de cualquier pequeña perturbación, ruido, etc. va hacer que se avise de la presencia de fallos cuando realmente no los hay. Mientras que los métodos propuestos, obtienen buenos resultados en todos los índices testados: MDR, fallos detectados y retardo en la detección, pero entregando mejores resultados de tasas de falsas alarmas (FAR).

Hay que tener en cuenta que, como se indica en [Russell et al., 2000], se han cambiado los umbrales para realizar esta comparación de MDR para los métodos de PCA, DPCA y CVA con respecto a los utilizados para calcular las FAR para obtener mejores resultados. Mientras que en KPCA y los métodos propuestos en este documento, los umbrales aplicados para obtener los valores de MDR y de FAR son los mismos.

CAPÍTULO 6. FDI BASADO EN ANÁLISIS DE SERIES
TEMPORALES

Tabla 6.4: Tasa de no detección de fallos (MDR), en porcentaje. TEP

Fallo	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2	CVA Q	KPCA T^2	KPCA Q	EWMares	PCARES T^2	PCARES Q
1	0.8	0.3	0.6	0.5	0.1	0	0.3	0	0	0	0.625	0.125
2	2	1.4	1.9	1.5	1.1	1	2.6	2	2	0	2	1.5
3	99.8	99.1	99.1	99	98.1	98.6	98.5	96	92	89.1	99.875	93.8
4	95.6	3.8	93.9	0	68.8	0	97.5	91	0	0.25	81.5	6.625
5	77.5	74.6	75.8	74.8	0	0	0	75	73	38	76.125	65.3
6	1.1	0	1.3	0	0	0	0	1	0	0	0	0
7	8.5	0	15.9	0	38.6	0	48.6	0	0	0	0.75	0
8	3.4	2.4	2.8	2.5	2.1	1.6	48.6	3	4	0	2.75	1.6
9	99.4	98.1	99.5	99.4	98.6	99.3	99.3	96	96	88.4	99.75	93.8
10	66.6	65.9	58	66.5	16.6	9.9	59.9	57	49	28.9	66.625	43.3
11	79.4	35.6	80.1	19.3	51.5	19.5	66.9	76	19	7.0	42.75	27.375
12	2.9	2.5	1	2.4	0	0	2.1	3	2	0	1.5	0.5
13	6	4.5	4.9	4.9	4.7	4	5.5	6	5	0.25	6.5	4.0
14	15.8	0	6.1	0	0	0	12.2	21	0	0	0	0
15	98.8	97.3	96.4	97.6	92.8	90.3	97.9	95	93	85.5	99.5	90.9
16	83.4	75.5	78.3	70.8	16.6	8.4	42.9	70	48	15.9	64.375	33.75
17	25.9	10.8	24	5.3	10.4	2.4	13.8	26	5	2.6	15.25	2.625
18	11.3	10.1	11.1	10	9.4	9.2	10.2	10	10	0	11.5	9.0
19	99.6	87.3	99.3	73.5	84.9	1.9	92.3	97	51	56.6	98.75	41.8
20	70.1	55	64.4	49	24.8	8.7	35.4	59	45	16.1	43.625	20.1
21	73.6	57	64.4	55.8	44	34.2	54.7	65	47	51.5	69	62.1

La Tabla 6.5 contiene los resultados del tiempo de retardo en la detección para cada método y cada fallo. Se comprueba claramente cómo el método *PCAres*, con el estadístico Q , consigue una gran reducción del retardo de detección (número de instantes) en comparación con los métodos PCA, DPCA y CVA, dando a la vez un valor bajo de falsas alarmas (es decir, es sensible y robusto al mismo tiempo). El retraso de detección de la propuesta es similar al obtenido por el método KPCA en 10 de los 21 fallos considerados. Sin embargo, nuestro método detecta los fallos 3, 9 y 15, mientras que el KPCA no lo hace. Además, *PCAres*, con Q , es el mejor en 7 de los restantes fallos, mientras que KPCA es el mejor en los otros 4 fallos. Al mismo tiempo, el método *PCAres*, con Q , ofrece el menor retardo en la detección en los fallos 4, 5, 6, 7 y 14, donde los fallos se detectaron sin retardo alguno. Finalmente, se comprueba que el método *EWMAres* logra mejores o similares resultados a los de PCA, DPCA y CVA (con Q).

En resumen, se ha comprobado la efectividad de los métodos propuestos para llevar a cabo la tarea de detección. Concretamente, el método que aplica EWMA sobre residuos ha destacado por su sensibilidad a observaciones anómalas, dando los niveles más bajos de tasas MDR. También ha conseguido buenos resultados el método de PCA sobre residuos (con el estadístico Q). En ambos casos, en la mayoría de fallos han sido capaces de superar a los demás métodos de la comparativa.

En cuanto a los retardos en la detección, el método que ha destacado por su rapidez es el de PCA sobre residuos, utilizando el estadístico Q . Esta propuesta funciona mejor que la del método basado en EWMA, que aunque no da mal resultado, es más lenta. Muy probablemente, esto se deba a que el PCA trabaja con los datos instantáneos, tal cual vienen de los bloques, mientras que el EWMA aplica un filtro y trabaja con datos actuales y pasados, por lo que es esperable que la detección se retrase algo más.

En lo referente a los índices de falsas alarmas, ambos métodos han conseguido valores bastante bajos, lo que da prueba de su robustez frente a pequeñas perturbaciones, ruido, etc.

Por tanto, estos resultados sirven para comprobar la validez de estas propuestas.

Tabla 6.5: Retardo en la detección (en instantes). TEP

Fallo	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2	CVA Q	KPCA	EWMares	PCAres T^2	PCAres Q
1	7	3	6	5	2	3	2	5	2	5	1
2	17	12	16	13	13	15	25	10	13	16	12
3	nd	nd	nd	nd	nd	nd	nd	nd	86	nd	50
4	nd	3	151	2	462	1	nd	3	1	0	0
5	16	1	2	2	1	1	0	1	1	10	0
6	10	1	11	1	1	1	0	1	1	0	0
7	1	1	1	1	1	1	0	1	1	0	0
8	23	20	23	21	20	20	21	25	22	22	15
9	nd	nd	nd	nd	nd	nd	nd	nd	nd	nd	233
10	96	49	101	50	25	23	44	20	42	108	38
11	304	11	195	7	292	11	27	23	12	10	13
12	22	8	3	8	2	2	0	3	3	7	2
13	49	37	45	40	42	39	43	41	36	47	35
14	4	1	6	1	2	1	1	1	2	0	0
15	nd	740	nd	nd	677	nd	nd	9	674	nd	654
16	312	197	199	196	14	9	11	9	12	33	12
17	29	25	28	24	27	20	23	19	20	27	19
18	93	84	93	84	83	79	84	74	80	94	79
19	nd	nd	nd	82		11	nd	nd	280	nd	10
20	87	87	87	84	82	66	72	59	71	86	75
21	563	285	522	286	273	511	302	252	475	563	510

nd=no detectado

6.5. Caso de estudio II: Depuradora de Aguas Residuales

Después de haber probado la metodología propuesta en la planta anterior, y comprobado que da un tiempo de detección bajo así como un reducido número de falsas alarmas y de tasa MDR, se procede ahora a probarla en otra de las plantas utilizadas en esta tesis: la depuradora de aguas residuales BSM2, cuyas características se describen en el Apéndice A.2.

Dispondremos de 141 variables medidas (se incluye el valor de la variable controlada del oxígeno). Se van a tomar datos cada 8 horas y se dispondrá de conjuntos de datos sin fallo y de conjuntos de datos con 16 fallos diferentes.

6.5.1. Metodología experimental

La metodología experimental es similar al caso de estudio anterior: el conjunto de datos de entrenamiento sin fallos ($\mathbf{X} \in \mathbb{R}^{1827 \times 141}$) se usa para ejecutar todas las etapas fuera de línea (ver Algoritmo 4), que se detallan a continuación:

1. **Selección dinámica de características.** Esta etapa se hizo usando los tres métodos explicados en la Sección 6.2.1 considerando variables sin retardo y con los primeros 15 retardos. Y se seleccionaron las 10 variables (con o sin retardo) que tenían más influencia en el modelo.
2. **Modelado de las series temporales.** Se partió de los datos de entrenamiento sin fallo (1827 observaciones) y se generaron modelos de cada variable a partir de la selección hecha en el paso anterior. Estos modelos, para cada variable, fueron de los tipos:
 - **Modelos ARIMA:** con estructura (p,q,i) . Estos parámetros se hicieron variar entre los siguientes valores: $p = [0, \dots, 15]$, $q = [0, \dots, 15]$ e $i = [0, 1, 2]$, con la condición: $p + q \leq 20$.
 - **Modelos de redes neuronales:** el modelo de red neuronal fue un Perceptrón Multicapa (MLP) con tres capas: la primera se compone de 10 entradas que corresponden con las 10 entradas seleccionadas en la etapa anterior, y la capa de salida tiene una neurona de salida. El número de neuronas en la capa oculta se hizo variar entre los valores: $[5, \dots, 6 \cdot (2 \cdot \text{entradas})]$.

Para cada variable, se seleccionó el modelo con el menor valor de error rMSE en los datos de test sin fallo. En la Tabla 6.6 se muestran los modelos de serie temporal seleccionados para algunas de las variables. Los modelos ANN se corresponden con redes neuronales con selección de variables basada en correlación y los modelos PLS ANN son redes neuronales con selección de variables basada en PLS. En estos modelos

con redes neuronales se muestra también el número de neuronas en la capa oculta.

Tabla 6.6: Modelos seleccionados para algunas variables. BSM2

Variable	Modelo	Variable	Modelo
1	PLS ANN: Capa oculta: 26	28	ARIMA: (2,2,1)
2	ANN: Capa oculta: 21	35	ARIMA: (4,2,1)
3	PLS ANN: Capa oculta: 26	42	ARIMA: (4,2,1)
4	ANN: Capa oculta: 26	56	ARIMA: (5,1,1)
5	PLS ANN: Capa oculta: 26	77	ARIMA: (2,1,1)
6	ANN: Capa oculta: 26	84	ARIMA (4,2,1)
7	PLS ANN: Capa oculta: 21	98	ARIMA: (2,2,1)
8	ANN: Capa oculta: 26	105	ARIMA: (2,2,1)
9	PLS ANN: Capa oculta: 21	126	ARIMA: (2,2,1)
10	PLS ANN: Capa oculta: 26	133	ARIMA: (2,2,1)
11	PLS ANN: Capa oculta: 21	140	ARIMA: (2,2,1)

Para la mayoría de las variables, los modelos seleccionados están basados en redes neuronales. Esto puede ser debido a que, al estar trabajando con una instalación claramente no lineal, como es una planta de tratamiento de aguas residuales, el mejor modelo encontrado es, precisamente, de tipo no lineal, como son las redes neuronales. Todas las variables en las que se ha seleccionado un modelo ARIMA aparecen en la Tabla 6.6.

3. **Cálculo de los límites de las gráficas de control:** se obtienen a partir de los residuos para ambas gráficas de control:

- a) **Gráfica de control EWMA sobre los residuos (*EWMA-res*):** a partir de los datos de entrenamiento sin fallo y buscando un valor de λ que diera lugar a un número mínimo de observaciones consecutivas fuera de los umbrales, los valores elegidos fueron $Obs = 5$ y $\lambda = 0,9$. Estos valores dieron el menor retardo posible sin falsas alarmas y consiguieron detectar todos los fallos.
- b) **Detección de fallos basada en PCA de los residuos (*PCA-res*):** Se construye un modelo PCA a partir de los residuos y se escoge un número de componentes principales que sea capaz de retener un 85 % de la varianza de los datos de partida. Los umbrales de T^2 y Q se calculan de forma teórica y luego se reajustan experimentalmente para un nivel de significación impuesto (ISL) de 5 % en condiciones normales de operación. Además para detectar fallos es necesario superar durante 5 instantes consecutivos los límites ($Obs = 5$). Esto evitará falsas alarmas.

6.5.2. Análisis de resultados

En este apartado se ejecuta la cuarta etapa de la metodología: detección en línea de fallos. El método propuesto se ha comparado con un PCA con un 70% de varianza retenida en sus componentes principales, y con umbrales de T^2 y Q ajustados de forma teórica a un valor del 5% de falsas alarmas.

La tasa de falsas alarmas (FAR) para los datos de test se muestra en la Tabla 6.7, donde se puede ver que el mejor resultado lo obtiene el método *EWMares* con un valor de 0,32%. Los otros métodos tienen valores comprendidos entre 2,6% y 4,8%, valores todos ellos por debajo del nivel de falsas alarmas impuesto (5%) y, por tanto, correctos.

Tabla 6.7: Tasa de falsas alarmas (FAR), en porcentaje. BSM2

Método y estadístico	Datos de test
PCA T^2	2.6
PCA Q	4.8
<i>EWMares</i>	0.32
PCAses T^2	3.2
PCAses Q	2

En la Tabla 6.8 se muestran los resultados de la tasa de no detección (MDR). Al igual que en los datos de tasa FAR, se ve que el método *EWMares* es el que ofrece el mejor resultado, ya que es capaz de detectar más observaciones con fallo que el resto de métodos, en algunos casos con una diferencia abultada. Los fallos 6 a 16 son fáciles de detectar y todos los métodos dan un resultado bueno, excepto PCA con T^2 , que es el peor de todos.

En cuanto a los fallos 1 a 5, que son los que presentan una mayor dificultad para su detección, el método *EWMares* obtiene el mejor resultado, a bastante distancia de los demás, dando el método *PCAses*, con el estadístico Q , un resultado aceptable, mejor que el del PCA convencional en 3 de estos 5 fallos, e igual en otro de ellos.

Los resultados de retardo en la detección se muestran en la Tabla 6.9. Claramente el mejor método es el *PCAses* con el estadístico Q , ya que obtiene el menor retardo en todos los fallos excepto el número 2. Además, todos los métodos son capaces de detectar todos los fallos, excepto *PCAses* con T^2 , que no detecta el 3 ni el 4, que son fallos difíciles de detectar. En general, se comprueba que los métodos basado en PCA funcionan mejor con el estadístico Q que con el T^2 .

Por tanto, siendo el más rápido en 15 de los 16 fallos, el método *PCAses*, con el estadístico Q , es el que consigue el mejor resultado en la detección. El método *EWMares*, da también un buen resultado, aunque es peor que el PCA sobre residuos, y consigue menores retardos que el PCA (con T^2 o Q)

Tabla 6.8: Tasa de no detección de fallos (MDR), en porcentaje. BSM2

Fallo	PCA T^2	PCA Q	EWMARes	PCARes T^2	PCARes Q
1	96.41	10.53	0.12	69.83	6.93
2	96.66	42.11	3.98	87.42	27.86
3	97.18	87.55	27.47	96.92	82.16
4	97.82	71.50	25.03	96.79	86.01
5	91.78	0.26	0.13	44.16	0.26
6	27.47	0.26	0.26	0.26	0.26
7	86.65	0.26	0.26	0.26	0.26
8	26.32	0.26	0.26	0.26	0.26
9	2.05	0.39	0.39	0.39	0.39
10	0.39	0.39	0.26	0.39	0.26
11	27.47	0.26	0.26	0.39	0.26
12	0.26	0.26	0.26	0.26	0.26
13	0.26	0.26	0.26	0.26	0.26
14	0.26	0.26	0.26	0.26	0.26
15	0.26	0.26	0.26	0.26	0.26
16	14.89	0.26	0.26	0.26	0.26

en 15 de los 16 fallos.

En resumen, los métodos propuestos consiguen menores tasas de falsas alarmas y, por tanto, son menos sensibles al ruido o a perturbaciones. Además, dan menores tasas de MDR, es decir, son más sensibles a observaciones anómalas. Concretamente, el método *EWMARes* es el que obtienen los mejores índices MDR.

Finalmente, ambos métodos son más rápidos en la detección, destacando *PCARes* con Q , que es el mejor. Estos resultados reafirman las conclusiones obtenidas en el caso de estudio anterior (planta TEP) y reafirman la validez de las propuestas realizadas.

Tabla 6.9: Retardo en la detección (en instantes). BSM2

Fallo	PCA T^2	PCA Q	EWMAres	PCAres T^2	PCAres Q
1	258	772	23	253	12
2	258	725	184	253	193
3	258	211	223	<i>nd</i>	194
4	259	278	223	<i>nd</i>	208
5	259	7	3	2	2
6	726	7	3	2	2
7	541	7	3	2	2
8	775	7	3	2	2
9	506	8	4	3	3
10	8	8	3	3	2
11	775	7	3	3	2
12	7	7	3	2	2
13	7	7	3	2	2
14	7	7	3	2	2
15	7	7	3	2	2
16	692	7	3	2	2

nd=no detectado

6.6. Resumen

En este capítulo se hace una propuesta de método de detección de fallos descentralizando el cálculo, es decir, en cada bloque se hace un procesamiento de los datos actuales y de datos con retardo de las variables correspondiente mediante técnicas de *análisis de series temporales*.

El proceso consiste en crear un modelo de serie temporal para cada variable cuando la planta está trabajando en condiciones normales de operación y usar estos modelos para hacer predicciones. Dichas predicciones se obtendrán cuando se esté analizando la planta en busca de posibles fallos. De la comparación entre los valores reales de las variables medidos en la planta y las predicciones hechas con los modelos se obtendrán unos residuos que serán analizados de dos posibles formas: por un lado, analizar los residuos de cada variable de forma individual mediante gráficas de control EWMA (método EWMA sobre residuos), y, por otro, crear un modelo PCA cuyos datos de entrada sean los residuos obtenidos en los bloques (método PCA sobre residuos).

En la primera alternativa se establece un umbral para cada residuo, de

modo que si en alguno de ellos se supera el correspondiente umbral se detectará un fallo. En la segunda alternativa se usarán los estadísticos propios del PCA (T^2 y Q) y sus correspondientes umbrales para hacer la detección de fallos.

Los resultados obtenidos de aplicar dichos métodos en las plantas TEP y BSM2 muestran que las dos alternativas propuestas son eficaces y mejoran los resultados de otros métodos de detección. Esto es así, tanto en términos de fallos detectados, como en tasas de no detección y en retraso en la detección de fallos. Concretamente, el método basado en EWMA destaca por su eficacia detectando observaciones anómalas (tasa de no detección MDR), mientras que el método PCA sobre residuos (usando el estadístico Q) consigue los tiempos de detección de fallos más bajos. En cuanto a las tasas de falsas alarmas (FAR) los dos métodos dan resultados similares o algo mejores que las de otros métodos con los que han sido comparados.

Este sistema de detección de fallos descentralizado es muy adecuado para plantas en las que se manejen gran cantidad de variables y no sea factible el enviar y procesar en cada instante todos los datos medidos en la instalación en un mismo procesador, ya que en este caso los datos se procesan en distintos nodos y sólo requiere transmitir información de la detección o no detección de un fallo en cada bloque al nodo central para obtener la diagnosis de la planta completa.

Parte III:

**CONCLUSIONES, APÉNDICES Y
BIBLIOGRAFÍA**

Capítulo 7

CONCLUSIONES FINALES Y TRABAJO FUTURO

En este trabajo se han presentado los resultados de una serie de investigaciones relacionadas con la diagnosis de fallos distribuida. A continuación se presentan las conclusiones derivadas de dichas investigaciones así como las líneas de trabajo futuras con las que se puede ampliar el trabajo realizado.

7.1. Contribuciones y publicaciones de la tesis

1. **Estudio de métodos distribuidos basados en PCA.** [Sánchez-Fernández et al., 2015a, Sánchez-Fernández et al., 2015b]. Se ha realizado un primer intento de aplicación distribuida del método PCA, en la que se ha dividido la planta usando criterios topológicos. Cada bloque cuenta con su propio modelo PCA y envía sus resultados a un procesador central para generar un único resultado global.

También se han puesto a prueba varios métodos de PCA distribuido (no pensados inicialmente para hacer diagnosis de fallos) propuestos por otros autores para comprobar su eficacia. Algunos han funcionado en la monitorización del proceso, pero han dado peor resultado que el PCA distribuido propuesto en esta Tesis.

Además, se ha comprobado que la aproximación distribuida ha obtenido mejores resultados que la aproximación centralizada. Pero distribuir la planta usando criterios topológicos es complejo de aplicar en el caso de plantas complejas y, por tanto, requiere buscar un método alternativo.

2. **Aplicación de métodos dinámicos de detección de fallos con un planteamiento distribuido. Propuesta de los métodos: MB-DDPCA y MB-DCVA.** [Sánchez-Fernández et al., 2018b, Sánchez-Fernández et al., 2018c, Sánchez-Fernández et al., 2019]. En esta se-

gunda contribución se presentan dos métodos de diagnóstico de fallos dinámicos ampliamente utilizados como métodos centralizados: DP-CA y CVA. El hecho de usar métodos dinámicos permite analizar la correlación temporal de las variables y no sólo la espacial. Teniendo en cuenta que las plantas industriales son instalaciones complejas con un claro comportamiento dinámico, es indispensable el uso de este tipo de métodos para hacer una adecuada diagnosis.

Dentro de cada una de las propuestas se han encontrado alternativas que ofrecen mejores resultados que la aplicación centralizada de ambos métodos, en términos de rapidez en la detección, sensibilidad a los fallos, tasas de falsas alarmas y número de fallos detectados.

- 3. Desarrollo de métodos para realizar la división de la planta en bloques.** [Sánchez-Fernández et al., 2018b, Sánchez-Fernández et al., 2018c, Sánchez-Fernández et al., 2019]. Esta contribución aborda la forma de hacer la división de la planta. Ante las dificultades que presentan los métodos basados en la topología o en el conocimiento previo de la planta, se ha optado por métodos basados en el análisis de datos. De esta forma únicamente se necesita disponer de datos medidos en la planta, trabajando ésta en condiciones normales.

Estos métodos tratan de encontrar qué variables están más relacionadas entre sí para incluirlas en el mismo bloque cuando se aplique PCA, o las menos relacionadas cuando se aplique CVA. Esto es así debido a que el CVA crea un modelo en espacio de estados de la planta, y, al modelar, es preferible incluir variables poco correlacionadas entre sí, que aportarán más información que las que están muy correlacionadas y, por tanto, se obtendrán mejores modelos.

Para esta tarea, se ha propuesto usar para la descentralización métodos basados en regresión lineal (LASSO, DLASSO, SPLS DSPLS), regresión no-lineal (redes neuronales, SVR, RF), métodos basados en información de la señal como correlación cruzada sin tendencias (DC-CA), métodos de agrupaciones de variables (como DBSCAN y EM). Estos métodos permiten operar de forma totalmente descentralizada, con un bloque por variable, o bien utilizando menos bloques que variables. En este último caso los bloques pueden estar solapados (es decir, hay variables que pertenecen a varios bloques) o no solapados, para poder estudiar también el efecto del número de bloques en la eficacia de la detección de fallos. Todos estos métodos de división de la planta se han utilizado tanto con el método DPCA como con el CVA, obteniendo buenos resultados, especialmente con DPCA donde cualquier método de descentralización usada es mejor que el DPCA centralizado.

- 4. Distribución de la identificación de fallos. Propuesta del método de identificación de fallos distribuida basada en opera-**

dores lingüísticos OWA-RIM: ORAFI [Sánchez-Fernández et al., 2018d, Sánchez-Fernández et al., 2018e]. La siguiente contribución aborda la descentralización desde otro punto de vista, en vez de dividir la planta en bloques de variables para ser procesados individualmente, se crea un sistema distribuido en el que cada centro de procesamiento trata los datos de todas las variables, pero usando métodos diferentes.

En este caso, esta distribución se refiere a la identificación de fallos. Se han dispuesto diferentes bloques de modo que en cada uno de ellos se ha implementado un método de identificación de entre los más usados en la literatura. Cada uno de ellos procesa los datos medidos en la planta y envía los resultados a un procesador central. Este procesador los fusiona mediante técnicas de toma de decisiones, permitiendo agregar resultados procedentes de diferentes criterios para generar un resultado único.

Como no existe un método de identificación completamente efectivo en todas las ocasiones, esta propuesta permite usar varios de los métodos de identificación más empleados, de modo que al darse una situación en la que uno de ellos no es efectivo, los demás pueden suplirle y el resultado de identificación final sea correcto. Esto hace que la identificación sea más robusta.

- 5. Realización de la identificación en dos instantes de tiempo.** [Sánchez-Fernández et al., 2018d, Sánchez-Fernández et al., 2018e]. En esta contribución se presenta la idea de hacer la identificación de fallos en dos instantes de tiempo. En general, se suele realizar la identificación nada más detectarse el fallo, aunque también hay autores que la ejecutan usando todos los datos tomados desde la aparición del fallo hasta muchos instantes después. La identificación realizada justo al detectarse el fallo sirve para conocer su origen.

Mientras que la identificación realizada varios instantes después, cuando la planta ha evolucionado en condiciones de fallo y ha llegado a un punto de trabajo estable, no va a servir para conocer el origen del fallo salvo que la planta, una vez aparezca el fallo, no modifique sustancialmente su estado. En cambio, si la planta pasa a otro estado después del fallo, esta segunda diagnosis servirá para conocer en qué condiciones está trabajando la planta de cara a tomar decisiones sobre las acciones a tomar.

Los resultados obtenidos muestran que de esta forma se consigue realizar un análisis más completo del sistema cuando sufre un fallo.

- 6. Detección de fallos distribuida basada en modelos de series temporales y métodos estadísticos multivariantes.** [Sánchez-Fernández et al., 2018a]. En esta última contribución se aborda la

detección distribuida con otro punto de vista. Ahora lo que se descentraliza son los cálculos. En cada bloque se modela una o varias variables en base a sus datos actuales y pasados usando técnicas de análisis de series temporales. Con estos modelos se hacen predicciones de las variables que son comparadas con las medidas reales, generándose así unos residuos que serán analizados posteriormente. Se han propuesto dos métodos de análisis de estos residuos: mediante gráficas de control del tipo EWMA y mediante un modelo PCA.

Los resultados obtenidos por ambos métodos son mejores que los de otros métodos de detección, ya que son capaces de detectar más fallos, son más sensibles a las observaciones anómalas y permiten detectar los fallos antes.

7. **Todos los métodos propuestos se han aplicado en varias plantas complejas obteniéndose buenos resultados en la detección e identificación de fallos.** Dichas propuestas han demostrado ser más eficaces que los métodos centralizados y, en muchos casos, más eficaces que otras propuestas descentralizadas encontradas en la literatura.

7.2. Conclusiones

A partir de la información, los resultados y las consideraciones expuestas a lo largo de esta tesis se llega a las siguientes conclusiones principales:

- Los métodos de diagnóstico de fallos distribuidos basados en análisis de datos presentan una mayor eficacia que los de tipo centralizado: consiguen detectar más fallos, más rápido, dan una mejor identificación, etc. Aunque requieren abordar la problemática relativa al método de división de la planta y a la forma de fusionar los resultados de cada uno de los bloques con los que se trabaja.
- Al ser las plantas reales de tipo no lineal y dinámicas, se hace imprescindible usar métodos de diagnóstico que permitan analizar las no linealidades y el carácter dinámico de la instalación: los métodos DPCA y CVA cumplen estos requisitos y son efectivos para la monitorización de la planta.
- Una opción para hacer la división de la planta son los métodos basados en datos, que permiten hacer la descentralización únicamente a partir del análisis de las medidas tomadas en la instalación, sin necesidad de poseer un conocimiento previo de la misma. Estos métodos consiguen buenos resultados en el posterior proceso de diagnóstico y facilitan el trabajo del operador de la planta. Además, hay diferentes variantes para conseguir distintos tipos de descentralizaciones.

- Ante el hecho de que no existe un único método de identificación que sea efectivo en todos los casos, se plantea la posibilidad de utilizar varios de estos métodos. Se ha comprobado que, usando simultáneamente los métodos que se consideren más eficaces y fusionando sus resultados, se consiguen mejores identificaciones. Además, de cara a obtener una identificación más completa del fallo de la planta, es preferible realizar la identificación tanto en el instante de detección del fallo como varios instantes después, cuando la planta ya se ha estabilizado.
- El uso de técnicas de modelado basadas en series temporales aplicado a una descentralización del procesamiento de la información permite obtener un método de detección que supera en eficacia a otros métodos de monitorización. Además, con este método se reduce la cantidad de información a transmitir, siendo adecuado para plantas con gran número de variables y con restricciones en las comunicaciones.

7.3. Líneas futuras de trabajo

El trabajo que se ha desarrollado en esta tesis abre nuevas alternativas que deben ser consideradas como punto de partida para el desarrollo de nuevos métodos de diagnóstico de fallos. A continuación se muestran algunas de estas alternativas:

- Al trabajar con plantas descentralizadas es crucial la elección del método para llevar a cabo dicha descentralización. Hay que tener en cuenta que no existe un único método que sea eficaz en todos los casos. Por tanto, en función de la planta con la que se esté trabajando habrá que optar por aquel método que mejor se pueda adaptar a dicha planta. Esto abre muchas posibilidades que deben ser consideradas ya que los métodos que se pueden aplicar son innumerables: lineales, no lineales, basados en regresión, etc.
- En el trabajo expuesto en esta tesis se han utilizado DPCA y CVA como métodos de diagnóstico, pero no son los únicos. Por tanto, otra línea de trabajo que se abre es la aplicación de más técnicas de monitorización basadas en datos, que pueden ser basadas en control estadístico de procesos como KPCA, KICA, etc. o en inteligencia computacional, como las redes neuronales de *deep learning*, o los métodos de clasificación no balanceados, etc. Esto también se debe hacer buscando aquellas técnicas que mejor se adapten a la planta. Incluso se puede plantear la posibilidad de combinar varios métodos de diagnóstico en cada bloque y fusionar sus resultados para dar un resultado único.
- Otra línea de trabajo futura puede ser explorar más posibilidades para la realización de la identificación. En este trabajo se han usado siete métodos

distintos, pero se puede ampliar esta selección.

- Además, cuando se han modelado las variable mediante técnicas de análisis de series temporales se ha optado por unos tipos de modelos. Se puede probar con más modelos, cada uno con diferentes características de linealidad o ausencia de esta, autocorrelación, etc. El objetivo es encontrar el mejor modelo posible para la variable que se esté modelando.
- Otra posibilidad que se puede explorar consiste en hacer una descentralización que no sea jerárquica, sin un nodo central que fusione toda la información de los bloques, sino que los bloques se comuniquen entre si para dar la solución global.

Apéndice A

MODELOS DE SIMULACIÓN

A.1. Red de distribución de agua potable

Se dispone de un modelo de red de distribución de agua potable implementada mediante el software de simulación EPANET [Rossman, 2000b], del que se pueden extraer datos de funcionamiento de las variables del sistema en diferentes instantes. El manual de dicho software se puede encontrar en [Rossman, 2000a].

En la Figura: A.1 se muestra el esquema de la red utilizada. Es una red de un tamaño medio que cuenta con varios puntos de entrada de agua y muchos puntos de consumo, además de la red de tuberías que transportan el agua por la instalación. Se realizarán simulaciones de la red en diferentes situaciones y se extraerán datos para realizar los análisis.

La red considerada está formada por varios elementos, que se detallan a continuación:

- **Nudos** (representados por puntos en el esquema), son la unión de varias tuberías. En estos nudos se puede extraer agua para abastecer un punto de consumo, o introducir agua en la red. En la red considerada, en todos los nudos se extrae agua, excepto en los nodos: 1, 17, 34, 36, y 38, por los que entra agua a la red. Por el nodo 1 es por el que entra mayor cantidad de agua a la red, mientras que los demás nodos mencionados se aportan menores cantidades de agua.
- **Tuberías** (representadas por líneas rectas), sirven para el transporte de agua entre los nodos de la red, son de distintos diámetros y longitudes, según la cantidad de agua que deban transportar.
- **Reservas de agua** (en el esquema está identificada como el elemento 1), son fuentes de agua teóricamente inagotables.
- **Bombas**, son los componentes que aportan la energía necesaria al agua para que pueda circular por la red y llegar a todos sus puntos (está representada sobre la tubería 1 de la red).

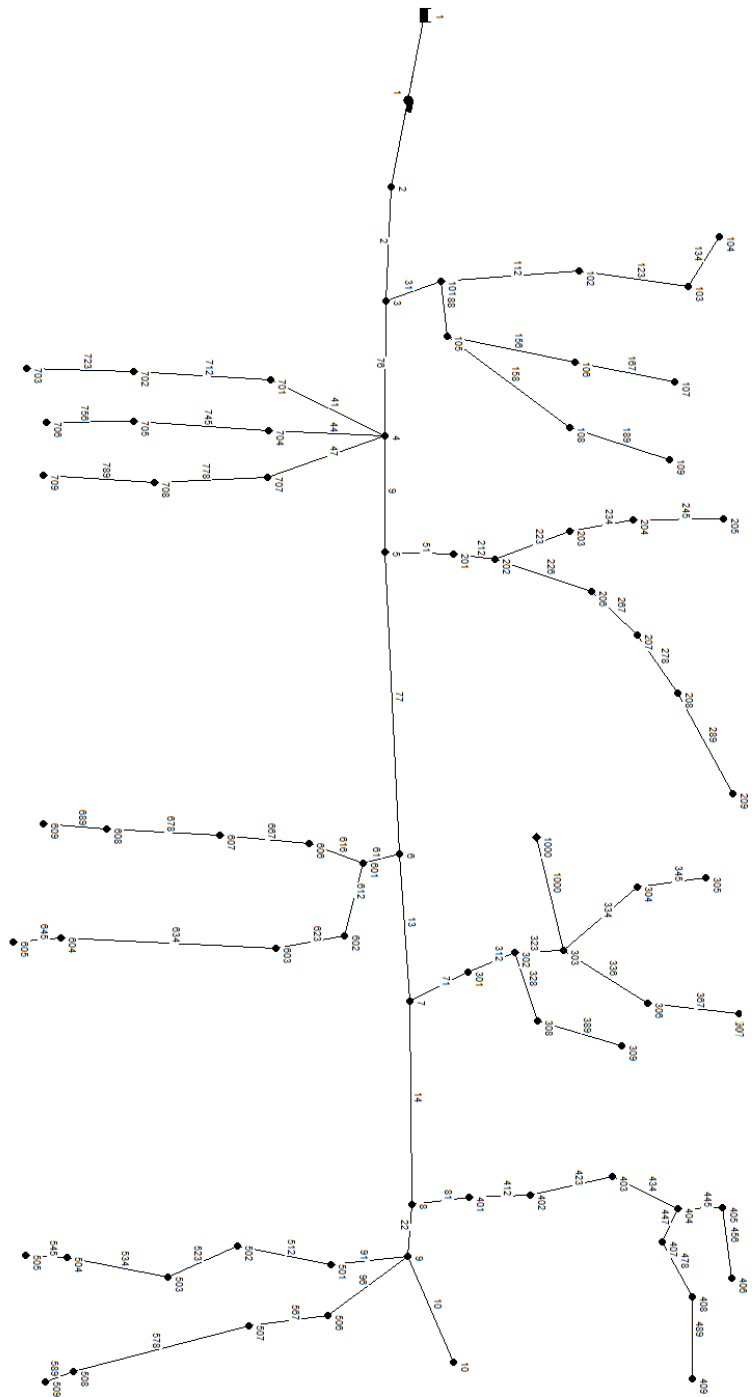


Figura A.1: Red de distribución de agua potable

Tanto en las tuberías como en los nodos se pueden tomar datos, concretamente, en los nodos se dispondrá de los siguientes datos:

- **Presión**
- **Demanda de agua** (cantidad de agua que se consume en ese nodo)
- **Altura piezométrica**
- **Concentración de aditivo** (en nuestro caso será flúor)

Y en las tuberías se tendrán datos de:

- **Caudal**
- **Velocidad del agua**
- **Pérdida de altura piezométrica** (pérdidas por unidad de longitud)
- **Concentración media de aditivo**

Tomando todos estos datos en todos los elementos de la red y en diferentes instantes se podrán crear tablas de datos para ser analizadas con los métodos de diagnóstico propuestos. En dichas tablas se hará un preprocesado de cara a eliminar aquellas variables que no aporten información (por ejemplo, las que no cambien de valor en toda la simulación o tengan una varianza próxima a cero).

Nota: para obtener las tablas completas de resultados, con datos de los valores de cada variable en cada instante, se ha recurrido a alternativas como son las aplicaciones *AWARE – P software suite* desarrolladas por el grupo BASEFORM [BASEFORM, 2014].

A.1.1. Fallos

Con esta planta se han realizado dos tipos de simulaciones, por un lado simulaciones sin ningún fallo en la instalación para tomar datos de funcionamiento en condiciones normales, y por otro lado simulaciones en la presencia de fallos. Estas últimas se usarán para comprobar si los métodos de diagnóstico propuestos son eficaces.

Los fallos que se han implementado son los siguientes:

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

En cada simulación se generarán datos de 200 instantes, tomando medidas cada hora. Para los análisis, se eliminarán las 100 primeras observaciones, que se consideran de estabilización de la red, y por tanto, se utilizarán las

100 observaciones restantes. En el caso de las simulaciones con fallo, éste aparecerá en el instante 50 (una vez descartadas las primeras 100 observaciones). Todas las simulaciones cuentan con un patrón de variación en la demanda para simular las variaciones normales de consumo que se pueden observar en una red real (ver Figura A.2), de tal forma que la red suministra una cantidad de agua que varía a lo largo del día.

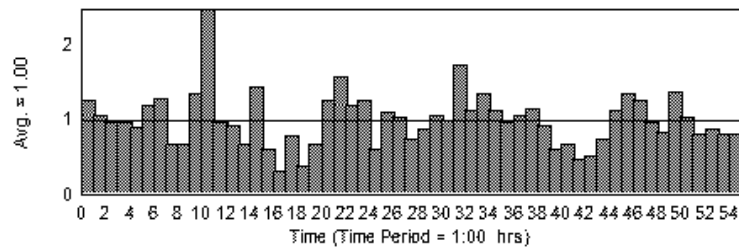


Figura A.2: Patrón de demanda de agua

A continuación se describen los fallos considerados:

- **Fallo en la bomba.** Aquí lo que se hace es introducir un cambio en la velocidad de giro de la bomba (simulando alguna problema en el motor de la bomba, suciedad, etc.), por tanto, la energía comunicada al agua disminuirá. Esto se ha conseguido alterando el parámetro de velocidad relativa de la bomba en EPANET.
- **Aparición de un contaminante.** El siguiente fallo simula la aparición de un producto químico nocivo en la red. Esto se implemente en EPANET colocando un deposito con agua que contiene una determinada concentración del contaminante (elemento 1000 en la Figura A.3) conectado a la red mediante una tubería inicialmente cerrada que se abre en el instante en el que se quiera que aparezca el fallo.
- **Fuga en una tubería.** Este fallo simula una fuga en una parte de la red, concretamente en el nodo 303 (ver Figura A.4). La forma de implementarla ha sido colocando una tubería conectada a ese nodo que, inicialmente, está cerrada. En el instante de aparición del fallo se permitirá el paso de agua a través de la tubería que llegará hasta el nodo terminal (nodo 1000). Este nodo se ha configurado para que funcione como un aspersor y, por tanto, la cantidad de agua que sale a través de él dependerá de la presión con la que le llega el agua. Se puede considerar que una pequeña fuga en la tubería se comporta de forma similar a un aspersor.

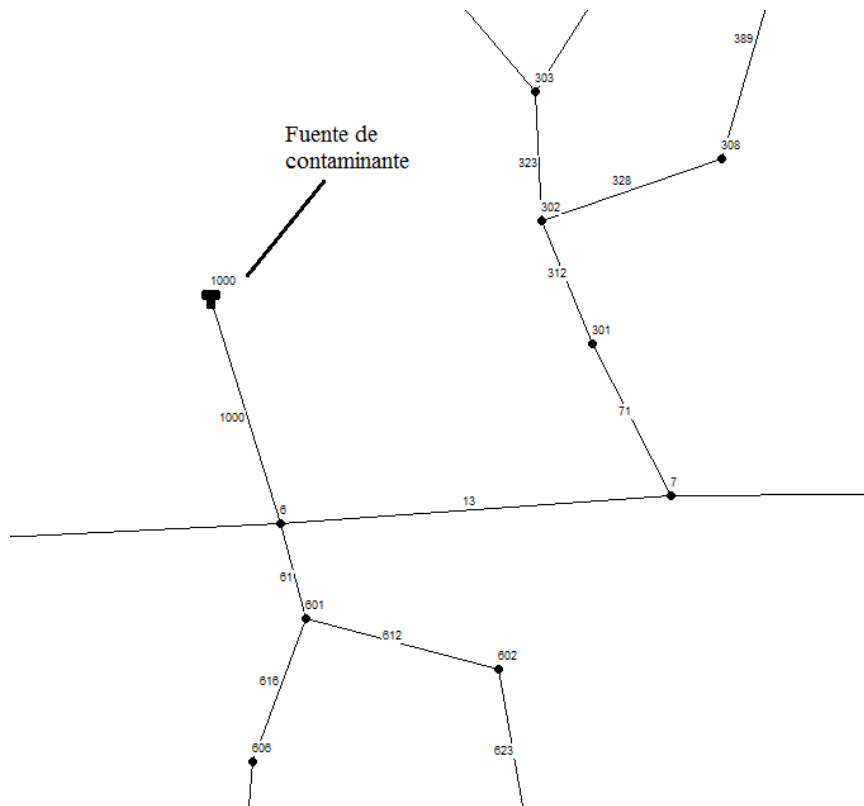


Figura A.3: Introducción de contaminante a la red

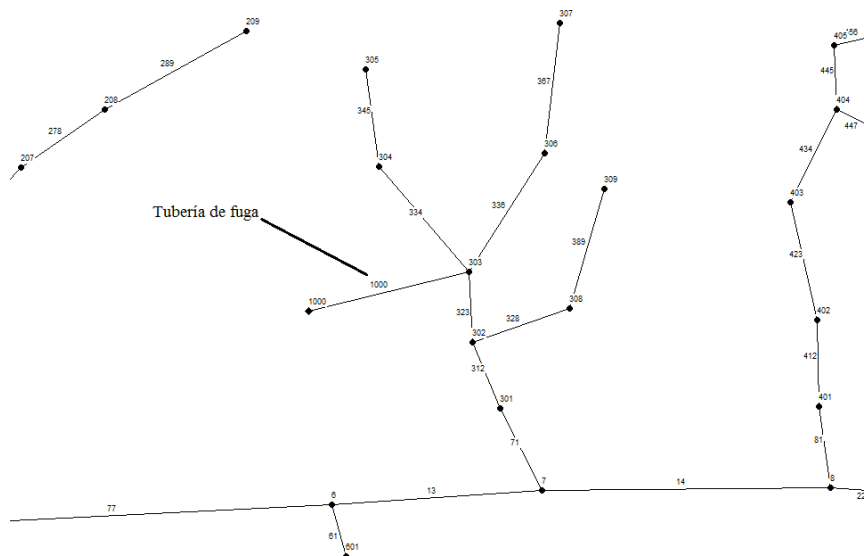


Figura A.4: Tubería de fuga de agua

A.2. Planta depuradora de aguas residuales

El modelo usado para simular y obtener datos de una depuradora de aguas residuales es el Modelo de Simulación BSM2 (en inglés, *Benchmark Simulation Model 2*) desarrollado por los Grupos de Trabajo *COST Action 682 y 624*, y el *IWA Task Group on Benchmarking of control Strategies for WWTP* ([Copp, 2001, Alex et al., 2008]).

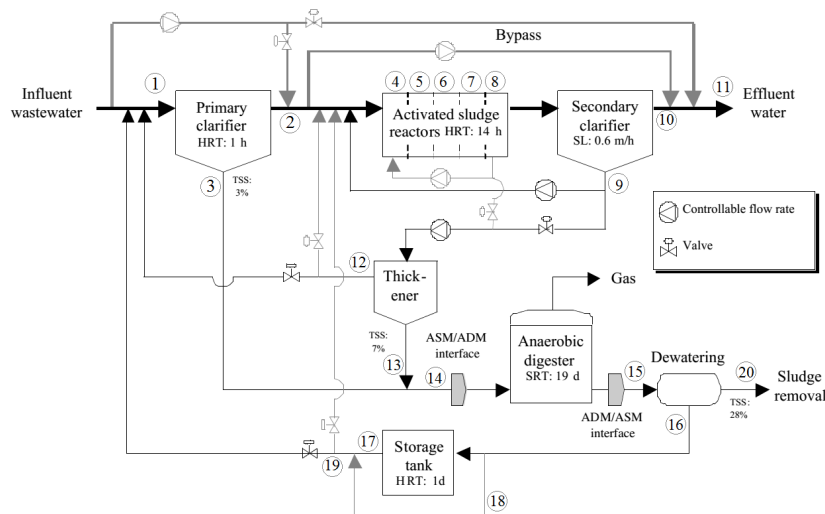


Figura A.5: Planta BSM2

Este tipo de plantas son unas instalaciones diseñadas para eliminar los productos químicos y biológicos nocivos presentes en el agua que se recoge de la red de aguas residuales de una población, esto permite que este agua, una vez tratada, se pueda usar para otros fines: riego, usos industriales, etc. o para ser vertida en cauces naturales con el mínimo impacto posible. Esta planta esta formada por distintas partes (ver figura A.5) como son un clarificador, varios reactores de fangos activados, un sedimentador (o clarificador secundario), dispositivos de tratamiento de fangos: espesante, digestor, tanque de almacenamiento, unidad de deshidratación, etc. Todos estos elementos están unidos por tuberías, y controlados por válvulas, bombas, etc.

Como se dispone de la implementación del modelo BSM2 en Simulink (Matlab), es posible hacer todas las simulaciones que se necesiten, cambiando incluso algunas partes de este modelo para producir deliberadamente anomalías, y así tener datos con fallo para probar los métodos de diagnosis.

Este modelo cuenta con un sistema de control que regula su funcionamiento. Dicho control mide el nivel de oxígeno disuelto en el líquido del cuarto reactor del bloque *Activated sludge reactors* y en función de ese dato ajusta la cantidad de oxígeno que inyecta en los reactores tercero, cuarto y quinto del citado bloque. 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 quinto reactor al primer reactor) y Q_w (caudal de entrada al espesador de lodos (*Thickener*)), en función de la composición del agua a la salida del quinto reactor.

La planta trabaja con 16 variables de estado como son: material inerte soluble, sustrato lentamente biodegradable, nivel de nitratos y nitritos, etc. [Alex et al., 2008] en cada uno de los 20 puntos de medición disponibles, por tanto hay un total de 320 variables en el sistema. Dichos puntos de medida están indicados en la figura A.5.

En una planta real de tratamiento de aguas residuales no es posible medir todo este conjunto de variables en cada instante de tiempo, por lo que en este trabajo se ha trabajado con una serie de variables mucho más fáciles de medir instantáneamente en una instalación real (ver primera columna de la Tabla A.1), que se han obtenido a partir de varias variables extraídas del modelo (segunda columna de la Tabla A.1), en algunos casos sumando varias variables para dar lugar a otra (por ejemplo, el nivel de nitrógeno es la suma del nivel de nitritos y nitratos, del nivel de amonio y amoniaco, y del nitrógeno soluble orgánico asociado a S_S). De modo que ahora se cuenta con 7 variables medidas en los 20 puntos de medición, dando lugar a 140 variables. Además, se ha tomado el valor que el sistema de control establece para ajustar el nivel de oxígeno en el cuarto reactor (variable KLa), quedando un total de 141 variables.

El modelo de simulación se ejecuta a lo largo de 609 días, tomándose medidas cada 15 minutos. Así que al final del proceso se obtiene una matriz de datos de 58.464 filas y 141 columnas.

Tabla A.1: Variables en la planta BSM2

Variables utilizadas	Variables del modelo
COD (Demanda química de Oxígeno) (en $g\ COD.m^{-3}$)	S_I (Material soluble inerte), S_S (Sustrato fácilmente biodegradable), X_I (Material inerte particulado), X_S (Substrato lentamente biodegradable)
O_2 (Oxígeno) (en $g\ (-COD).m^{-3}$)	S_O (Oxígeno disuelto)
Alk (Alcalinidad)	S_{alk} (Alcalinidad)
N (Nitrógeno) (en $g\ N.m^{-3}$)	S_{NO} (Nitratos y nitritos), S_{NH} (Amoniac y amonio), S_{ND} (Nitrógeno soluble orgánico asociado a S_S)
SS (Sólidos suspendidos) (en $g\ SS.m^3$)	TSS (Sólidos suspendidos totales)
$Caudal$ (en $m^3.d^{-1}$)	$Caudal$
$Temperatura$ (en $^{\circ}C$)	$Temperatura$

A.2.1. Fallos

En este caso, se han considerado tres tipos de fallo, que se han implementado con una magnitud que varía entre el 20% y 70%, dando como resultado un conjunto de 13 conjuntos de datos con los que probar los dife-

rentes métodos de detección:

- **Fallo en el sensor de oxígeno (F_1)**, consiste en simular un fallo en el sensor de oxígeno del cuarto reactor del bloque de fangos activados, dando medidas erróneas. El controlador de oxígeno, entonces, no introducirá la cantidad correcta de oxígeno en los reactores. Los experimentos llevados a cabo con los datos de este tipo de fallo muestran que es el más difícil de detectar e identificar de los tres.
- **Cambio del valor de alcalinidad del agua (F_2)**, consiste en cambiar el valor de la alcalinidad del agua en la entrada y, en otros casos, en el primer reactor de fangos activados.
- **Fallos en los caudales (F_3)**, se dispone de datos de simulación de una fuga en la salida inferior del decantador primario y, también de una fuga en la salida del tanque de almacenamiento de lodos. Además, se tienen datos de un mal funcionamiento de las válvulas de control que hay a la salida del clarificador secundario: el flujo denominado Q_u circula por una tubería que se divide en dos dando lugar a otros dos flujos: Q_w y Q_r , y el fallo consiste en simular un atasco en una de las tuberías de salida que provoca un aumento de caudal en la otra.

Los conjuntos de datos con fallo disponibles se muestran en la Tabla A.2, donde aparecen las intensidades de fallo y, también, cuáles de las 141 variables son las responsables del fallo.

Tabla A.2: Fallos disponibles en la planta BSM2

Fallo n°	Descripción	Variabes responsables
1 a 5	Fallo sensor O_2 , desde -50 % a 70 %	42 125
6 a 8	Alcalinidad Influyente, desde -50 % a 40 %	8 15
9 a 11	Alcalinidad Reactor 1, desde -30 % a 20 %	22
12 y 13	Fuga en flujo inferior Decantador, desde -50 % a -30 %	80 86
14	Fuga en salida Tanque almacenamiento, - 50 %	80 86
15 y 16	Reparto flujos Q_r y Q_w , desde -50 % a -25 % de Q_r	80 86

A.3. Tennessee Eastman Process

La planta *Tennessee Eastman Process* (TEP) es un modelo de simulación de un proceso químico [Downs and Vogel, 1993]. Es un caso de estudio ampliamente utilizado para probar nuevos métodos y estrategias de control y monitorización. En el campo de la monitorización ha sido usada por multitud de autores para comprobar el desempeño de métodos de detección y diagnosis de fallos [Grbovic et al., 2012, Liu, 2012, Jiang et al., 2015, Odiowei

and Cao, 2010, Ge and Song, 2013, Krishnannair and Aldrich, 2017, Hajihosseini et al., 2018].

Esta planta trabaja con cinco reactivos: A, B, C, D y E, que procesa para obtener dos productos: G y H. Consta de distintos elementos como un reactor, un condensador, un separador, etc. además de bombas, tuberías y válvulas. Se puede ver un esquema de la planta en la figura A.6. En la parte izquierda se encuentran las entradas y a la derecha la salida.

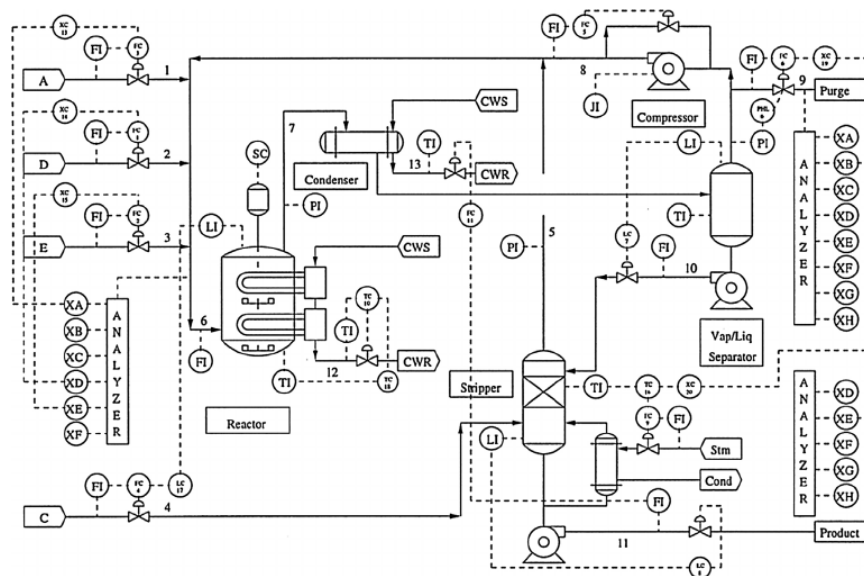


Figura A.6: Esquema de la planta Tennessee Eastman Process (TEP)

Los datos disponibles para esta planta corresponden a 52 variables, de las cuales 22 son mediciones continuas del proceso que se miden cada 3 minutos (variables $Xmeas1$ a $Xmeas22$), 19 variables remuestreadas del proceso que se miden cada 6 minutos (variables $Xmeas23$ a $Xmeas36$) y cada 15 minutos (variables $Xmeas37$ a $Xmeas41$), y, finalmente, las variables manipuladas del proceso que se miden cada 3 minutos (variables $xmv1$ a $xmv12$). La lista de variables se puede ver en la tabla A.3.

Para esta planta se dispone de dos conjuntos de datos: los de entrenamiento y los de prueba. Cada uno de ellos está formado por 22 grupos de datos: el primero de ellos corresponde a datos de funcionamiento sin fallo y los otros 21 corresponden a diferentes fallos, incluidos fallos de tipo escalón, cambios aleatorio en algunas variables, etc. Los fallos comienzan en la primera observación de los datos de entrenamiento y en la 161 de los datos de prueba. Cada uno de los 22 conjuntos de datos de entrenamiento contiene 500 observaciones mientras que cada uno de los 22 de test tiene 960 observaciones.

Tabla A.3: Variables en la planta TEP

Variable n°	Descripción	Nombre
Mediciones continuas del proceso		
1	A (entrada)	Xmeas1
2	D (entrada)	Xmeas2
3	E (entrada)	Xmeas3
4	A y C (entrada)	Xmeas4
5	Flujo de reciclado	Xmeas5
6	Reactor (tasa de alimentación)	Xmeas6
7	Reactor (presión)	Xmeas7
8	Reactor (nivel)	Xmeas8
9	Reactor (temperatura)	Xmeas9
10	Tasa de purga	Xmeas10
11	<i>Product separator</i> (temperatura)	Xmeas11
12	<i>Product separator</i> (nivel)	Xmeas12
13	<i>Product separator</i> (presión)	Xmeas13
14	<i>Product separator</i> (flujo inferior)	Xmeas14
15	<i>Stripper</i> (nivel)	Xmeas15
16	<i>Stripper</i> (presión)	Xmeas16
17	<i>Stripper</i> (flujo inferior)	Xmeas17
18	<i>Stripper</i> (temperatura)	Xmeas18
19	<i>Stripper</i> (flujo de vapor)	Xmeas19
20	Trabajo del compresor	Xmeas20
21	Reactor (refrigerante: temp. de salida)	Xmeas21
22	<i>Separator</i> (refrigerante: temp. de salida)	Xmeas22
Variables remuestreadas del proceso		
23	Flujo 6: Componente A	Xmeas23
24	Flujo 6: Componente B	Xmeas24
25	Flujo 6: Componente C	Xmeas25
26	Flujo 6: Componente D	Xmeas26
27	Flujo 6: Componente E	Xmeas27
28	Flujo 6: Componente F	Xmeas28
29	Flujo 9: Componente A	Xmeas29
30	Flujo 9: Componente B	Xmeas30
31	Flujo 9: Componente C	Xmeas31
32	Flujo 9: Componente D	Xmeas32
33	Flujo 9: Componente E	Xmeas33
34	Flujo 9: Componente F	Xmeas34
35	Flujo 9: Componente G	Xmeas35
36	Flujo 9: Componente H	Xmeas36
37	Flujo 11: Componente D	Xmeas37
38	Flujo 11: Componente E	Xmeas38
39	Flujo 11: Componente F	Xmeas39
40	Flujo 11: Componente G	Xmeas40
41	Flujo 11: Componente H	Xmeas41
Variables manipuladas del proceso		
42	D (flujo de alimentación)	xmv1
43	E (flujo de alimentación)	xmv2
44	A (flujo de alimentación)	xmv3
45	A y C (flujo de alimentación)	xmv4
46	Compresor (Válvula de reciclado)	xmv5
47	Válvula de purga	xmv6
48	<i>Separator</i> (flujo de líquido)	xmv7
49	<i>Stripper</i> (flujo de producto líquido)	xmv8
50	<i>Stripper</i> (válvula de vapor)	xmv9
51	Reactor (flujo de refrigerante)	xmv10
52	Condensador (flujo de refrigerante)	xmv11

En la tabla A.4 se puede ver una descripción de los fallos del proceso. Como se aprecia, hay distintos tipos de fallo: cambios abruptos (“Salto”), lentos (“Cambio lento”), válvulas pegadas (“Bloqueo”), cambios imprevisibles (“Variación aleatoria”), y otros desconocidos. Algunos de estos fallos son fáciles de detectar: fallos 5, 6, 7 y 14, etc., por contra, otros son más difíciles, y, de hecho, hay algunos que no suelen detectarse: fallos 3, 9 y 15 [Jiang et al., 2015].

Tabla A.4: Fallos disponibles en la planta TEP

Fallo n°	Descripción	Tipo
1	Alimentación A/C, composición B constante (Flujo 4)	Salto
2	Composición B, ratio A/C constante (Flujo 4)	Salto
3	Alimentación D (Flujo 2)	Salto
4	Temperatura del refrigerante, entrada del reactor	Salto
5	Temperatura del refrigerante, entrada del condensador	Salto
6	Pérdida alimentación A (Flujo 1)	Salto
7	Menor presión en tubería alimentación C (Flujo 4)	Salto
8	Composición A, B y C (Flujo 4)	Variación aleatoria
9	Temperatura entrada D (Flujo 2)	Variación aleatoria
10	Temperatura entrada C (Flujo 4)	Variación aleatoria
11	Temperatura del refrigerante, entrada del reactor	Variación aleatoria
12	Temperatura de refrigerante, entrada del condensador	Variación aleatoria
13	Cinética de la reacción	Cambio lento
14	Válvula del refrigerante del reactor	Bloqueo
15	Válvula del refrigerante del condensador	Bloqueo
16	Desconocido	-
17	Desconocido	-
18	Desconocido	-
19	Desconocido	-
20	Desconocido	-
21	Válvula del flujo 4	Bloqueo

Como se ha dicho, esta planta se ha usado profusamente para hacer detección y diagnóstico de fallos. En el primero caso, se tiene la total certeza del momento en el que empiezan los fallos y por tanto se puede comprobar si un método de detección es bueno o no simplemente comprobando si detecta rápidamente los fallos, si las observaciones con fallo las clasifica como datos con fallo, etc. En el caso de la diagnosis de fallos ya no existe tal certeza, ya que, aunque se conoce el tipo de fallo que se está usando (excepto los fallos 16 a 20), no sé sabe qué variables son las que originan el fallo. Esto dificulta la comprobación de la efectividad de un método de identificación. Algunos autores han recurrido a representar gráficamente las variables y ver cuáles sufren cambios al aparecer el fallo [Liu, 2012, Liu and Chen, 2014], y considerar a estas como originarias del fallo, pero no es una técnica demasiado precisa para identificar, y además no se analizan los cambios en la correlación entre variables.

BIBLIOGRAFÍA

- [Abiodun et al., 2018] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H. (2018). State-of-the-art in artificial neural network applications: a survey. *Heliyon*, 4.
- [Ahmad and Dey, 2007] Ahmad, A. and Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data and Knowledge Engineering*, 63(2):503–527.
- [Akaike, 1974] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723.
- [Alcala and Qin, 2009] Alcala, C. F. and Qin, S. J. (2009). Reconstruction-based contribution for process monitoring. *Automatica*, 45:1593–1600.
- [Alex et al., 2008] Alex, J., Benedetti, L., Copp, J., Gernaey, K., Jeppsson, U., Nopens, I., Pons, M., Rosen, C., Steyer, J., and Vanrolleghem, P. (2008). Benchmark simulation model no. 2.
- [Alvarez, 2013] Alvarez, D. G. (2013). *Monitoring, fault detection and estimation in processes using multivariate statistical techniques*. PhD thesis, Departamento de Ingeniería de Sistemas y automática, Universidad de Valladolid.
- [Angeli, 2010] Angeli, C. (2010). Diagnostic expert systems: from expert’s knowledge to real-time systems. In *Advanced knowledge based systems: model, applications & research*, volume 1, pages 50–73.
- [Azar, 2010] Azar, A. T. (2010). *Fuzzy systems*. IntechOpen.
- [Bai et al., 2005] Bai, Z.-J., Chan, R. H., and Luk, F. T. (2005). Principal component analysis for distributed data sets with updating. In Cao, J., Nejdil, W., and Xu, M., editors, *Advanced Parallel Processing Technologies*, pages 471–483, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Bakdi and Kouadri, 2017] Bakdi, A. and Kouadri, A. (2017). A new adaptive PCA based thresholding scheme for fault detection in complex systems. *Chemometrics and Intelligent Laboratory Systems*, 162(Supplement C):83 – 93.

- [BASEFORM, 2014] BASEFORM (2014). Baseform.org, platform of applications for networked infrastructures.
- [Behzadian et al., 2012] Behzadian, M., Otaghsara, S., Yazdani, M., and Ignatius, J. (2012). A state-of-the-art survey of TOPSIS applications. *Expert Systems with Applications*, 200:198–215.
- [Borsje, 1999] Borsje, H. J. (1999). Fault detection in boilers using canonical variate analysis. In *Proceedings of the 1999 American Control Conference, San Diego, CA*, volume 2, pages 1167–1170.
- [Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*. ACM Press.
- [Box and Jenkins, 1976] Box, G. E. P. and Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control (Revised Edition)*. Holden-Day.
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- [Bro et al., 2008] Bro, R., Kjeldahl, K., Smilde, A. K., and Kiers, H. A. L. (2008). Cross-validation of component models: A critical look at current methods. *Analytical and Bioanalytical Chemistry*, 390(5):1241–1251.
- [Candès et al., 2011] Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust Principal Component Analysis? *J. ACM*, 58(3):1–37.
- [Castillo and Edgar, 2008] Castillo, I. and Edgar, T. (2008). Model based fault detection and diagnosis. In *TWCCC Conference, Austin, Texas*.
- [Chen and Zhang, 2014] Chen, C. P. and Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: a survey on Big Data. *Information Sciences*, 275:314–347.
- [Chiang et al., 2001] Chiang, L. H., Russell, E. L., and Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems*. Springer-Verlag London.
- [Chun and Keles, 2010] Chun, H. and Keles, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25.
- [Colombani et al., 2012] Colombani, C., Croiseau, P., Fritz, S., Guillaume, F., Legarra, A., Ducrocq, V., and Robert-Granier, C. (2012). A comparison of partial least squares (PLS) and sparse PLS regressions in genomic

- selection in French dairy cattle. *Journal of Dairy Science*, 95(4):2120–2131.
- [Copp, 2001] Copp, J. (2001). The cost simulation benchmark-description and simulator manual. *Directorate-General for Research and Innovation (European Commission)*.
- [Cortez and Embrechts, 2013] Cortez, P. and Embrechts, M. J. (2013). Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, 225:1–17.
- [Cover and Thomas, 2006] Cover, T. M. and Thomas, J. A. (2006). *Elements of information theory*. Wiley John + Sons.
- [Dasarathy and Sheela, 1979] Dasarathy, B. V. and Sheela, B. V. (1979). A composite classifier system design: concepts and methodology. *Proceedings of the IEEE*, 67(5):708–713.
- [d'Aspremont et al., 2007] d'Aspremont, A., Ghaoui, L. E., Jordan, M. I., and Lanckriet, G. R. G. (2007). A direct formulation for Sparse PCA using semidefinite programming. *Society for Industrial and Applied Mathematics*, 49(3):434–448.
- [de Oña and Garrido, 2014] de Oña, J. and Garrido, C. (2014). Extracting the contribution of independent variables in neural network models: a new approach to handle instability. *Neural Computing and Applications*, 25(3-4):859–869.
- [Dean and Ghemawat, 2008] Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.
- [Detroja et al., 2007] Detroja, K., Gudi, R., and Patwardhan, S. (2007). Plant-wide detection and diagnosis using correspondence analysis. *Control Engineering Practice*, 15:1468–1483.
- [Dhanisetty et al., 2018] Dhanisetty, V. V., Verhagen, W., and Curran, R. (2018). Multi-criteria weighted decision making for operational maintenance processes. *Journal of Air Transport Management*, 68:152–164.
- [Dickey and Fuller, 1979] Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366):427–431.
- [Dorr et al., 1997] Dorr, R., Kratz, F., Ragot, J., Loisy, F., and Germain, J. . (1997). Detection, isolation, and identification of sensor faults in nuclear power plants. *IEEE Transactions on Control Systems Technology*, 5(1):42–60.

- [Downs and Vogel, 1993] Downs, J. J. and Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17:245–255.
- [Dunia and Qin, 1998] Dunia, R. and Qin, S. J. (1998). Subspace approach to multidimensional fault identification and reconstruction. *AIChE Journal*, 44:1813–1831.
- [Edwards et al., 1998] Edwards, S., Lees, A., and Friswell, M. (1998). Fault diagnosis of rotating machinery. *The Shock and Vibration Digest*, 30(1):4–13.
- [Elattar et al., 2010] Elattar, E. E., Goulermas, J., and Wu, Q. (2010). Electric load forecasting based on locally weighted support vector regression •. *IEEE Transactions on Systems, Man and Cybernetics.*, 40(4):438–447.
- [Elgamal and Hefeeda, 2015] Elgamal, T. and Hefeeda, M. (2015). Analysis of PCA algorithms indistributed environments. *Technical ReportarXiv:1503.05214*.
- [Escobet et al., 2019] Escobet, T., Bregon, A., Pulido, B., and Puig, V. (2019). *Fault diagnosis of dynamic systems*. Springer International Publishing.
- [Fassò and Cameletti, 2009] Fassò, A. and Cameletti, M. (2009). The EM algorithm in a distributed computing environment for modelling environmental space–time data. *Environmental Modelling & Software*, 24(9):1027–1035.
- [Ferré, 1995] Ferré, L. (1995). Selection of components in principal component analysis: a comparison of methods. *Computational Statistics & Data Analysis*, 19(6):669–682.
- [Frank et al., 2000] Frank, P. M., Ding, S. X., and Marcu, T. (2000). Model-based fault diagnosis in technical processes. *Transactions of the Institute of Measurement and Control*, 22(1):57–101.
- [Frisén, 2009] Frisén, M. (2009). Optimal sequential surveillance for finance, public health, and other areas. *Sequential Analysis*, 28(3):310–337.
- [Fuller and Majlender, 2001] Fuller, R. and Majlender, P. (2001). An analytic approach for obtaining maximal entropy OWA operator weights. *Fuzzy Sets and Systems*, 124:53–57.
- [Gao et al., 2010] Gao, D., Wu, C., Zhang, B., and Ma, X. (2010). Signed directed graph and qualitative trend analysis based fault diagnosis in chemical industry. *Chinese Journal of Chemical Engineering*, 18(2):265–276.

-
- [Garraway, 1999] Garraway, H. (1999). Parallel computer architecture: a hardware/software approach. *IEEE Concurrency*, 7(2):83–84.
- [Ge, 2017] Ge, Z. (2017). Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems*, 171:16–25.
- [Ge and Chen, 2016] Ge, Z. and Chen, J. (2016). Plant-wide industrial process monitoring: a distributed modeling framework. *IEEE Transactions on Industrial Informatics*, 12(1):310–321.
- [Ge and Song, 2013] Ge, Z. and Song, Z. (2013). Distributed PCA model for plant-wide process monitoring. *Industrial and Engineering Chemistry Research*, 52:1947–1957.
- [Ge et al., 2013] Ge, Z., Song, Z., and Gao, F. (2013). Review of recent research on data-based process monitoring. *Industrial & Engineering Chemistry Research*, 52:3543–3562.
- [Ghemawat et al., 2003] Ghemawat, S., Gobioff, H., and Leung, S.-T. (2003). The google file system. *ACM SIGOPS Operating Systems Review*, 37(5):29.
- [Ghosh et al., 2011a] Ghosh, K., Natarajan, S., and Srinivasan, R. (2011a). Hierarchically distributed fault detection and identification through Dempster–Shafer evidence fusion. *Industrial & Engineering Chemistry Research*, 50(15):9249–9269.
- [Ghosh et al., 2011b] Ghosh, K., Ng, Y. S., and Srinivasan, R. (2011b). Evaluation of decision fusion strategies for effective collaboration among heterogeneous fault diagnostic methods. *Computers and Chemical Engineering*, 35:342–355.
- [Gooijer and Hyndman, 2006] Gooijer, J. G. D. and Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473.
- [Grbovic et al., 2012] Grbovic, M., Li, W., Xu, P., Usadi, A., Song, L., and Vucetic, S. (2012). Decentralized fault detection and diagnosis via sparse PCA based decomposition and maximum entropy decision fusion. *Journal of Process Control*, 22:738–750.
- [Hajihosseini et al., 2018] Hajihosseini, P., Anzehaee, M. M., and Behnam, B. (2018). Fault detection and isolation in the challenging tennessee eastman process by using image processing techniques. *ISA Transactions*, 79:137–146.

- [Hansen and Salamon, 1990] Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001.
- [Harrou et al., 2013] Harrou, F., Fillatre, L., Bobbia, M., and Nikiforov, I. (2013). Statistical detection of abnormal ozone measurements based on constrained generalized likelihood ratio test. In *52nd IEEE Conference on Decision and Control*, pages 4997–5002.
- [Heidari et al., 2016] Heidari, E., Sobati, M. A., and Movahedirad, S. (2016). Accurate prediction of nanofluid viscosity using a multilayer perceptron artificial neural network (MLP-ANN). *Chemometrics and Intelligent Laboratory Systems*, 55:73–85.
- [Heizer and Render, 2013] Heizer, J. and Render, B. (2013). *Operations Management*. Pearson.
- [Heo and Lee, 2018] Heo, S. and Lee, J. H. (2018). Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine*, 51(18):470–475.
- [Ho, 2002] Ho, T. K. (2002). A data complexity analysis of comparative advantages of decision forest constructors. *Pattern Analysis & Applications*, 5(2):102–112.
- [Hunter, 1986] Hunter, J. (1986). The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210.
- [Hyndman and Athanasopoulos, 2017] Hyndman, R. J. and Athanasopoulos, G. (2017). Forecasting: principles and practice. <https://www.otexts.org/book/fpp>.
- [Hyndman and Khandakar, 2008] Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 27.
- [Isermann, 2006] Isermann, R. (2006). *Fault-Diagnosis Systems*. Springer Berlin Heidelberg.
- [Jackson and Mudholkar, 1979] Jackson, J. and Mudholkar, G. (1979). Control procedures for residuals associated with principal component analysis. *Technometrics*, 3(21):341–349.
- [Jackson, 1991] Jackson, J. E. (1991). *A user's guide to principal components*. Wiley New York.
- [James et al., 2013] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*. Springer New York.

-
- [Jiang and Braatz, 2017] Jiang, B. and Braatz, R. D. (2017). Fault detection of process correlation structure using canonical variate analysis-based correlation features. *Journal of Process Control*, 58:131–138.
- [Jiang et al., 2015] Jiang, B., Huang, D., Zhu, X., Yang, F., and Braatz, R. D. (2015). Canonical variate analysis-based contributions for fault identification. *Journal of Process Control*, 26:17–25.
- [Jiang and Huang, 2016] Jiang, Q. and Huang, B. (2016). Distributed monitoring for large-scale processes based on multivariate statistical analysis and Bayesian method. *Journal of Process Control*, 46(Supplement C):75–83.
- [Jiang and Yan, 2014] Jiang, Q. and Yan, X. (2014). Plant-wide process monitoring based on mutual information-multiblock principal component analysis. *ISA Transactions*, 53(5):1516–1527.
- [Jiang et al., 2016] Jiang, Q., Yan, X., and Huang, B. (2016). Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and bayesian inference. *IEEE Transactions on Industrial Electronics*, 63(1):377–386.
- [Jog et al., 2017] Jog, A., Carass, A., Roy, S., Pham, D. L., and Prince, J. L. (2017). Random forest regression for magnetic resonance image synthesis. *Medical Image Analysis*, 35:475–488.
- [Jolliffe, 2002] Jolliffe, I. (2002). *Principal component analysis*. Springer Verlag New York.
- [Kadri et al., 2016] Kadri, F., Harrou, F., Chaabane, S., Sun, Y., and Tahon, C. (2016). Seasonal ARMA-based SPC charts for anomaly detection: Application to emergency department systems. *Neurocomputing*, 173:2102–2114.
- [Kano et al., 2003] Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I., and Ohno, H. (2003). Monitoring independent components for fault detection. *AIChE Journal*, 49(4):969–976.
- [Kargupta et al., 2001] Kargupta, H., Huang, W., Sivakumar, K., and Johnson, E. (2001). Distributed clustering using collective principal component analysis. *Knowledge and Information Systems*, 3(4):422–448.
- [Kariwala et al., 2010] Kariwala, V., Odiowei, P.-E., Cao, Y., and Chen, T. (2010). A branch and bound method for isolation of faulty variables through missing variable analysis. *Journal of Process Control*, 20:1198–1206.

- [Khashei and Bijari, 2010] Khashei, M. and Bijari, M. (2010). An artificial neural network (p, d,q) model for timeseries forecasting. *Expert Systems with Applications*, 37:479–489.
- [Kou et al., 2014] Kou, G., Peng, Y., and Wang, G. (2014). Evaluation of clustering algorithms for financial risk analysis using MCDM methods. *Information Sciences*, 275(Supplement C):1–12.
- [Kourti and MacGregor, 1996] Kourti, T. and MacGregor, J. (1996). Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, 28:409–428.
- [Kourti and MacGregor, 1995] Kourti, T. and MacGregor, J. F. (1995). Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28(1):3–21.
- [Kriegel et al., 2011] Kriegel, H.-P., Kröger, P., Sander, J., and Zimek, A. (2011). Density-based clustering. *WIREs Data Mining and Knowledge Discovery*, 1(3):231–240.
- [Krishnannair and Aldrich, 2017] Krishnannair, S. and Aldrich, C. (2017). Fault detection in the tennessee eastman benchmark process with nonlinear singular spectrum analysis. *IFAC-PapersOnLine*, 50(1):8005–8010.
- [Ku et al., 1995] Ku, W., Storer, R., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 30(1):179–196.
- [Kuhn and Johnson, 2013] Kuhn, M. and Johnson, K. (2013). *Applied predictive modeling*. Springer-Verlag New York.
- [Larimore, 1996] Larimore, W. E. (1996). Statistical optimality and canonical variate analysis system identification. *Signal Processing*, 52:131–144.
- [Larimore, 1997a] Larimore, W. E. (1997a). *Canonical variate analysis in control and signal processing*, pages 83–120. Marcel Dekker, Inc. New York, NY, USA.
- [Larimore, 1997b] Larimore, W. E. (1997b). *Statistical methods in control and signal processing*. Marcel Dekker.
- [Li et al., 2010] Li, W., Gui, W. H., Xie, Y. F., and Ding, S. X. (2010). Decentralised fault detection of large-scale systems with limited network communications [brief paper]. *IET Control Theory Applications*, 4(9):1867–1876.
- [Li et al., 2018] Li, X., Duan, F., Loukopoulos, P., Bennett, I., and Mba, D. (2018). Canonical variable analysis and long short-term memory for

-
- fault diagnosis and performance estimation of a centrifugal compressor. *Control Engineering Practice*, 72:177–191.
- [Liu, 2012] Liu, J. (2012). Fault diagnosis using contribution plots without smearing effect on non-faulty variables. *Journal of Process Control*, 22:1609–1623.
- [Liu and Chen, 2014] Liu, J. and Chen, D.-S. (2014). Fault isolation using modified contribution plots. *Computers & Chemical Engineering*, 61:9–19.
- [Liu et al., 2017] Liu, J., Li, Y.-F., and Zio, E. (2017). A SVM framework for fault detection of the braking system in a high speed train. *Mechanical Systems and Signal Processing*, 87:401–409.
- [Liu et al., 2014] Liu, J., Wong, D., and Chen, D.-S. (2014). Bayesian filtering of the smearing effect: fault isolation in chemical process monitoring. *Journal of Process Control*, 24:1–21.
- [Liu and Han, 2008] Liu, X. and Han, S. (2008). Orness and parameterized RIM quantifier aggregation with OWA operators: a summary. *International Journal of Approximate Reasoning*, 48(1):77–97.
- [Livani and Abadi, 2010] Livani, M. and Abadi, M. (2010). Distributed PCA-based anomaly detection in wireless sensor networks. In *2010 IEEE International Conference for Internet Technology and Secure Transactions*, pages 1–8.
- [Mahadevan and Shah, 2009] Mahadevan, S. and Shah, S. L. (2009). Fault detection and diagnosis in process data using one-class support vector machines. *Journal of Process Control*, 19:1627–1639.
- [Majumder, 2015] Majumder, M. (2015). *Multi Criteria Decision Making*, pages 35–47. Springer Singapore.
- [Manali et al., 2009] Manali, K., Abou, S., and Stachowicz, M. (2009). Fault detection in hydraulic system using fuzzy logic. In *Proceedings of the World Congress on Engineering and Computer Science 2009*, volume II, pages 966–971.
- [Mardani et al., 2015] Mardani, A., Jusoh, A., and Zavadskas, E. (2015). Fuzzy multiple criteria decision-making techniques and applications – Two decades review from 1994 to 2014. *Expert Systems with Applications*, 42:4126–4148.
- [Miljković, 2011] Miljković, D. (2011). Fault detection methods: A literature survey. In *Proceedings of the 34th International Convention MIPRO*, pages 750–755.

- [Nakamura et al., 2017] Nakamura, T. A., Palhares, R. M., Caminhas, W. M., Menezes, B. R., de Campos, M. C. M., Fumega, U., de M. Bomfim, C. H., and Lemos, A. P. (2017). Adaptive fault detection and diagnosis using parsimonious gaussian mixture models trained with distributed computing techniques. *Journal of the Franklin Institute*, 354(6):2543–2572. Special issue on recent advances on control and diagnosis via process measurements.
- [Negiz and Cinar, 1997] Negiz, A. and Cinar, A. (1997). Statistical monitoring of multivariable dynamical processes with state-space models. *AIChE Journal*, 43(8):2002–2020.
- [Ng and Srinivasan, 2010] Ng, Y. S. and Srinivasan, R. (2010). Multi-agent based collaborative fault detection and identification in chemical processes. *Engineering Applications of Artificial Intelligence*, 23(6):934–949.
- [Nieto et al., 2013] Nieto, P. G., Combarro, E., del Coz Díaz, J., and Montañés, E. (2013). A SVM-based regression model to study the air quality at local scale in Oviedo urban area (Northern Spain): A case study. *Applied Mathematics and Computation*, 219(17):8923–8937.
- [Nomikos and MacGregor, 1995] Nomikos, P. and MacGregor, J. F. (1995). Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37(1):41–59.
- [Nor et al., 2017] Nor, N. M., Hussain, M. A., and Hassan, C. R. C. (2017). Fault diagnosis and classification framework using multi-scale classification based on kernel Fisher discriminant analysis for chemical process system. *Applied Soft Computing*, 61:959–972.
- [Odiowei and Cao, 2010] Odiowei, P. and Cao, Y. (2010). State-space independent component analysis for nonlinear dynamic process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 103:59–65.
- [Olden and Jackson, 2002] Olden, J. D. and Jackson, D. A. (2002). Illuminating the “black box”: a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling*, 154(1-2):135–150.
- [Page, 1954] Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1-2):100–115.
- [Page, 1961] Page, E. S. (1961). Cumulative Sum Charts. *Technometrics*, 3(1):1–9.
- [Park et al., 2014] Park, Y., Baek, S., Kim, S.-h., and Tsui, K.-L. (2014). Statistical process control-based intrusion detection and monitoring. *Quality and Reliability Engineering International*, 30(2):257–273.

-
- [Paul et al., 2016] Paul, S., Magdon-Ismael, M., and Drineas, P. (2016). Feature selection for linear SVM with provable guarantees. *Pattern Recognition*, 60:205–214.
- [Paz and Gomez, 2014] Paz, R. C. and Gomez, D. G. (2014). *Administración De Las Operaciones*. Nueva Librería.
- [Pineda et al., 2018] Pineda, P. J. G., Liou, J. J., Hsu, C.-C., and Chuang, Y.-C. (2018). An integrated MCDM model for improving airline operational and financial performance. *Journal of Air Transport Management*, 68:103–117.
- [Podobnik and Stanley, 2008] Podobnik, B. and Stanley, H. E. (2008). Detrended cross-correlation analysis: A new method for analyzing two nonstationary time series. *Physical Review Letters*, 100(8):084102.
- [Poli and Boudet, 2018] Poli, J.-P. and Boudet, L. (2018). A fuzzy expert system architecture for data and event stream processing. *Fuzzy Sets and Systems*, 343:20–34.
- [Puigjaner et al., 2006] Puigjaner, L., Ollero, P., Prada, C. D., and Jiménez, L. (2006). *Estrategias de modelado, simulación y optimización de procesos químicos*. Síntesis.
- [Qin, 2003] Qin, S. (2003). Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17:480–502.
- [Qin, 2012] Qin, S. (2012). Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36:220–234.
- [Qu et al., 2002] Qu, Y., Ostrouchov, G., Samatova, N., and Geist, A. (2002). Principal component analysis for dimension reduction in massive distributed data sets. In *IEEE International Conference on Data Mining*.
- [Rossman, 2000a] Rossman, L. A. (2000a). *Epanet 2. User's manual*. United States Environmental Protection Agency. (<http://nepis.epa.gov/Adobe/PDF/P1007WWU.pdf>).
- [Rossman, 2000b] Rossman, L. A. (2000b). EPANET software. United States Environmental Protection Agency. (<https://www.epa.gov/water-research/epanet>).
- [Russell et al., 2000] Russell, E. L., Chiang, L. H., and Braatz, R. D. (2000). Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 51(1):81–93.

- [Russo et al., 2019] Russo, S., Li, G., and Villez, K. (2019). Automated model selection in principal component analysis: A new approach based on the cross-validated ignorance score. *Industrial & Engineering Chemistry Research*, 58(30):13448–13468.
- [Sainz et al., 2004] Sainz, G. I., Fuente, M. J., and Vega, P. (2004). Recurrent neuro-fuzzy modelling of a wastewater treatment plant. *European Journal of Control*, 10:83–95.
- [Saludes-Rodil and Fuente, 2010] Saludes-Rodil, S. and Fuente, M. J. (2010). Fault tolerance in the framework of support vector machines based model predictive control. *Engineering Applications of Artificial Intelligence*, 23:1127–1139.
- [Sánchez-Fernández et al., 2018a] Sánchez-Fernández, A., Baldán, F. J., Sainz-Palmero, G. I., Benítez, J. M., and Fuente, M. J. (2018a). Fault detection based on time series modeling and multivariate statistical process control. *Chemometrics and Intelligent Laboratory Systems*, 182:57–69.
- [Sánchez-Fernández et al., 2015a] Sánchez-Fernández, A., Fuente, M. J., and Sainz-Palmero, G. I. (2015a). Fault detection in wastewater treatment plants using distributed PCA methods. In *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–7.
- [Sánchez-Fernández et al., 2015b] Sánchez-Fernández, A., Fuente, M. J., and Sainz-Palmero, G. I. (2015b). Fault detection with distributed PCA methods in water distribution networks. In *2015 23rd Mediterranean Conference on Control and Automation (MED)*, pages 156–161.
- [Sánchez-Fernández et al., 2018b] Sánchez-Fernández, A., Fuente, M. J., and Sainz-Palmero, G. I. (2018b). Decentralized and dynamic fault detection using PCA and bayesian inference. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 800–807.
- [Sánchez-Fernández et al., 2018c] Sánchez-Fernández, A., Fuente, M. J., and Sainz-Palmero, G. I. (2018c). Detección de fallos dinámica y descentralizada basada en métodos de regresión. In *XXXIX Jornadas de Automática*, pages 554–561.
- [Sánchez-Fernández et al., 2019] Sánchez-Fernández, A., Fuente, M. J., and Sainz-Palmero, G. I. (2019). Decentralized DPCA model for large-scale processes monitoring. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE.
- [Sánchez-Fernández et al., 2018d] Sánchez-Fernández, A., Fuente, M. J., Sainz-Palmero, G. I., and Benítez, J. M. (2018d). Operadores lingüísticos

-
- OWA-RIM para la diagnosis de fallos en plantas complejas. In *XVIII Conferencia de la Asociación Española para la Inteligencia Artificial*, pages 1327–1332.
- [Sánchez-Fernández et al., 2018e] Sánchez-Fernández, A., Sainz-Palmero, G. I., Benítez, J. M., and Fuente, M. J. (2018e). Linguistic OWA and two time-windows based fault identification in wide plants. *Computers & Chemical Engineering*, 115:412–430.
- [Senobari et al., 2018] Senobari, R. K., Sadeh, J., and Borsi, H. (2018). Frequency response analysis (FRA) of transformers as a tool for fault detection and location: a review. *Electric Power Systems Research*, 155:172–183.
- [Shewhart, 1938] Shewhart, W. (1938). Application of statistical methods to manufacturing problems. *Journal of the Franklin Institute*, 226(2):163–186.
- [Shumway and Stoffer, 2006] Shumway, R. H. and Stoffer, D. S. (2006). *Time series analysis and its applications*. Springer-Verlag Berlin.
- [Shvachko et al., 2010] Shvachko, K., Kuang, H., Radia, S., and Chansler, R. (2010). The hadoop distributed file system. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–10.
- [Smola and Schölkopf, 2004] Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222.
- [Song et al., 2010] Song, X., Yang, J., Wu, X., and Yang, X. (2010). An optimal symmetrical null space criterion of Fisher discriminant for feature extraction and recognition. *Soft Computing*, 15(2):281–293.
- [Stanimirova et al., 2004] Stanimirova, I., Walczak, B., Massart, D., and Simeonov, V. (2004). A comparison between two robust PCA algorithms. *Chemometrics and Intelligent Laboratory Systems*, 71(1):83–95.
- [Tian et al., 2019] Tian, Y., Hu, T., Peng, X., Du, W., and Yao, H. (2019). Decentralized monitoring for large-scale process using copula-correlation analysis and Bayesian inference-based multiblock principal component analysis. *Journal of Chemometrics*, 33(8):1–18.
- [Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society*, 58:267–288.
- [Tidriri et al., 2016] Tidriri, K., Chatti, N., Verron, S., and Tiplica, T. (2016). Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: a review of researches and future challenges. *Annual Reviews in Control*, 42:63–81.

- [Tong et al., 2017] Tong, C., Lan, T., and Shi, X. (2017). Fault detection and diagnosis of dynamic processes using weighted dynamic decentralized PCA approach. *Chemometrics and Intelligent Laboratory Systems*, 161(Supplement C):34 – 42.
- [Tong and Shi, 2016] Tong, C. and Shi, X. (2016). Decentralized monitoring of dynamic processes based on dynamic feature selection and informative fault pattern dissimilarity. *IEEE Transactions on Industrial Electronics*, 63(6):3804–3814.
- [Tsagkrasoulis and Montana, 2018] Tsagkrasoulis, D. and Montana, G. (2018). Random forest regression for manifold-valued responses. *Pattern Recognition Letters*, 101:6–13.
- [Vanhatalo et al., 2017] Vanhatalo, E., KulaHCI, M., and Bergquist, B. (2017). On the structure of dynamic principal component analysis used in statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 167:1–11.
- [Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer New york.
- [Venkatasubramanian et al., 2003a] Venkatasubramanian, V., Rengaswamy, R., and Kavuri, S. (2003a). A review of process fault detection and diagnosis. Part I: quantitative model-based methods. *Computers & Chemical Engineering*, 27:291–311.
- [Venkatasubramanian et al., 2003b] Venkatasubramanian, V., Rengaswamy, R., and Kavuri, S. (2003b). A review of process fault detection and diagnosis. Part II: qualitative models and search strategies. *Computers & Chemical Engineering*, 27:313–326.
- [Venkatasubramanian et al., 2003c] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S., and Yin, K. (2003c). A review of process fault detection and diagnosis. Part III: process history based methods. *Computers & Chemical Engineering*, 27:327–346.
- [Villacreses et al., 2017] Villacreses, G., Gaona, G., Martinez-Gomez, J., and Jijón, D. J. (2017). Wind farms suitability location using geographical information system (GIS), based on multi-criteria decision making (MCDM) methods: The case of continental Ecuador. *Renewable Energy*, 109(C):275–286.
- [Wang et al., 2016] Wang, B., Yan, X., and Jiang, Q. (2016). Independent component analysis model utilizing de-mixing information for improved non-Gaussian process monitoring. *Computers & Industrial Engineering*, 94:188–200.

-
- [Wang et al., 2012] Wang, W., Ma, X., and Kang, R. (2012). Fisher Discriminant Analysis for fault classification. *Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing)*, pages 1–5.
- [Wold, 1966] Wold, H. (1966). *Estimation of principal components and related models by iterative least squares*, pages 391–420. Academic Press, New York.
- [Wold, 1982] Wold, H. (1982). Soft modeling: the basic design and some extensions. In Wold, K. J. . H., editor, *System under indirect observations*. Amsterdam: North Holland.
- [Wold et al., 2001] Wold, S., Sjostrom, M., and Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58:109–130.
- [Woodward and Goldsmith, 1964] Woodward, R. and Goldsmith, P. (1964). *Cumulative sum techniques, mathematical and statistical techniques for industry*. Oliver and Boyd, Edinburgh.
- [Wu and Zhao, 2018] Wu, H. and Zhao, J. (2018). Deep convolutional neural network model based chemical process fault diagnosis. *Computers & Chemical Engineering*, 115:185–197.
- [Wu et al., 2018] Wu, S. X., Wai, H.-T., Li, L., and Scaglione, A. (2018). A review of distributed algorithms for principal component analysis. *Proceedings of the IEEE*, 106(8):1321–1340.
- [Wu et al., 2016] Wu, Z., Li, Y., Plaza, A., Li, J., Xiao, F., and Wei, Z. (2016). Parallel and distributed dimensionality reduction of hyperspectral data on cloud computing architectures. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(6):2270–2278.
- [Xue, 2019] Xue, W. (2019). Input analysis in neural network. In *2nd ANU Bio-inspired Computing conference, 20-21 July 2019, Canberra, Australia*.
- [Yager, 1993] Yager, R. (1993). Families of OWA operators. *Fuzzy Sets and Systems*, 59:125–148.
- [Yager, 1996] Yager, R. (1996). Quantifier guided aggregation using OWA operators. *International Journal of Intelligent Systems*, 11:49–73.
- [Yager, 1988] Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18:183–190.

- [Yager, 2020] Yager, R. R. (2020). On the fusion of multiple multi-criteria aggregation functions with focus on the fusion of OWA aggregations. *Knowledge-Based Systems*, 191:105216.
- [Yin et al., 2014] Yin, S., Ding, S. X., Xie, X., and Luo, H. (2014). A review on basic data-driven approaches for industrial process monitoring. *IEEE Transactions on Industrial Electronics*, 61(11):6418–6428.
- [Yoon and MacGregor, 2001] Yoon, S. and MacGregor, J. (2001). Fault diagnosis with multivariable statistical models, part I: Using steady state fault signatures. *Journal of Process Control*, 11:387–400.
- [Yue and Qin, 2001] Yue, H. and Qin, S. J. (2001). Reconstruction based fault identification, using a combined index. *Industrial & Engineering Chemistry Research*, 40(20):4403–4414.
- [Zaharia et al., 2010] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud’10, pages 10–10, Berkeley, CA, USA. USENIX Association.
- [Zaidi, 2012] Zaidi, S. (2012). Development of support vector regression (SVR)-based model for prediction of circulation rate in a vertical tube thermosiphon reboiler. *Chemical Engineering Science*, 69(1):514–521. cited By 23.
- [Zebende, 2011] Zebende, G. (2011). DCCA cross-correlation coefficient: Quantifying level of cross-correlation. *Physica A: Statistical Mechanics and its Applications*, 390(4):614–618.
- [Zhang and Ge, 2015] Zhang, F. and Ge, Z. (2015). Decision fusion systems for fault detection and identification in industrial processes. *Journal of Process Control*, 31(Supplement C):45–54.
- [Zhang and Gao, 2018] Zhang, H. T. and Gao, M. X. (2018). The application of Support Vector Machine (SVM) regression method in tunnel fires. *Procedia Engineering*, 211:1004–1011.
- [Zhang and Zhao, 2019] Zhang, S. and Zhao, C. (2019). Hybrid independent component analysis (H-ICA) with simultaneous analysis of high-order and second-order statistics for industrial process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 185:47–58.
- [Zhang, 2008] Zhang, Y. (2008). Fault detection and diagnosis of nonlinear processes using improved kernel independent component analysis (KICA) and support vector machine (SVM). *Industrial & Engineering Chemistry Research*, 47(18):6961–6971.

-
- [Zhang and Hu, 2011] Zhang, Y. and Hu, Z. (2011). On-line batch process monitoring using hierarchical kernel partial least squares. *Chemical Engineering Research and Design*, 89(10):2078–2084.
- [Zhang et al., 2010] Zhang, Y., Zhou, H., Qin, S., and Chai, T. (2010). Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares. *IEEE Transactions on Industrial Informatics*, 6(1):3–10.
- [Zhang et al., 2003] Zhang, Z., Chen, C., Sun, J., and Chan, K. (2003). EM algorithms for gaussian mixtures with split-and-merge operation. *Pattern Recognition*, 36(9):1973–1983. cited By 121.
- [Zhou et al., 2016] Zhou, F., Park, J. H., and Liu, Y. (2016). Differential feature based hierarchical PCA fault detection method for dynamic fault. *Neurocomputing*, 202:27–35.
- [Zhou et al., 2011] Zhou, J., Shi, J., and Li, G. (2011). Fine tuning support vector machines for short-term wind speed forecasting. *Energy Conversion and Management*, 52(4):1990–1998.
- [Zhou and Qiu, 2018] Zhou, Y. and Qiu, G. (2018). Random forest for label ranking. *Expert Systems with Applications*, 112:99–109.
- [Zhu et al., 2017] Zhu, J., Ge, Z., and Song, Z. (2017). Distributed parallel PCA for modeling and monitoring of large-scale plant-wide processes with big data. *IEEE Transactions on Industrial Informatics*, 13(4):1877–1885.
- [Zou et al., 2006] Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286.
- [Zumoffen and Basualdo, 2008] Zumoffen, D. and Basualdo, M. (2008). From large chemical plant data to fault diagnosis integrated to decentralized fault-tolerant control: pulp mill process application. *Industrial & Engineering Chemistry Research*, 47(4):1201–1220.