

Control predictivo no lineal tolerante a fallos en una planta de tratamiento de aguas residuales

A. Sánchez-Fernández, M. Francisco**, M.J. Fuente, P. Vega**

Departamento de Ingeniería de Sistemas y Automática. Universidad de Valladolid.

alvar.sanchez@uva.es, mjfuente@eii.uva.es

** Departamento de Informática y Automática. Universidad de Salamanca.

[mfs,pvega]@usal.es

Resumen

En este trabajo se pretende diseñar un control tolerante a fallos, es decir un sistema que sea capaz de detectar cambios en el sistema que puedan provocar que el proceso abandone su modo de funcionamiento normal, y reconfigurar el controlador para que el sistema siga funcionando a pesar de los fallos. El sistema está formado por un controlador base, en este caso un control predictivo no lineal, un módulo de detección e identificación de fallos basado en el análisis de componentes principales (PCA) y un método de reconfiguración del controlador, que en este caso consiste en sustituir los sensores que fallan por un sensor software calculado con información del resto de las variables del sistema. Este sistema se ha probado en una planta de tratamiento de aguas residuales, y los resultados obtenidos muestran un funcionamiento adecuado a pesar de los fallos.

Palabras Clave: Control predictivo, Control tolerante a fallos, Detección e identificación de fallos, Análisis de componentes principales distribuido, Estación depuradora de aguas residuales, Sensores software.

1. INTRODUCCIÓN

El control predictivo basado en modelo (MPC) [2] es una de las técnicas de control avanzado más extendidas tanto en el entorno industrial como en el entorno académico. Este tipo de controladores presentan varias ventajas con respecto al control clásico. Su planteamiento es sencillo e intuitivo y pueden aplicarse a un gran espectro de sistemas, con diferente número de entradas y salidas, dinámicas o perturbaciones.

En cualquier tipo de control es muy importante conseguir un funcionamiento seguro y eficiente de la planta. Un aspecto muy importante para conseguir ese funcionamiento es disponer de un sistema rápido y fiable de detección e identificación de fallos. Esto puede reducir el número de averías y de paradas no deseadas de la planta. La detección de fallos indica algún tipo de malfuncionamiento en

el proceso, y la identificación de fallos apunta a la variable o variables del proceso responsables de que aparezca el fallo y que son las que hay que analizar para ver qué ha ocurrido. Dentro de los métodos de detección e identificación de fallos pueden aparecer métodos basados en datos y métodos basados en modelos.

Dentro del primer tipo, se encuentran métodos como el Análisis de Componentes Principales (PCA) [10, 12], que crea un modelo estadístico implícito partiendo de datos del sistema en condiciones normales de funcionamiento y avisa de un fallo al encontrarse desviaciones del comportamiento normal. En los métodos basados en modelos [3, 14], se desarrolla un modelo de la planta muy preciso que simula el comportamiento del sistema en condiciones normales, y se observan las desviaciones que pueda haber entre el modelo y la planta real. Una vez detectado un fallo, se hace el diagnóstico para saber dónde está el problema que ha aparecido en la instalación. Para ello se pueden aplicar técnicas como el Análisis del discriminante de Fisher [5, 6], redes neuronales, diagrama de contribuciones [11], etc.

Una vez el fallo ha sido detectado e identificado, el siguiente paso es la reconfiguración del sistema para que siga funcionando aceptablemente a pesar de los fallos, es decir, diseñar un control tolerante a fallos. En este trabajo la tolerancia a los fallos se consigue sustituyendo los sensores que han fallado por sensores software. Estos permiten estimar los valores de algunas variables de la planta sin necesidad de medirlas, solamente usando datos del resto de variables de la instalación. Esto se puede hacer usando diferentes técnicas: redes neuronales, modelos matemáticos, etc, pero en este caso se ha optado por aplicar métodos de regresión múltiple y en concreto PLS (Mínimos cuadrados parciales). [9].

El documento se divide de la forma siguiente: la primera sección da una descripción general del control predictivo, más adelante se detallan los aspectos más importantes del método PCA utilizado para detectar fallos. A continuación se explica brevemente cómo se aplica un sensor software basado en PLS y después se dará una breve descripción de

la planta de tratamiento de aguas residuales. En el apartado siguiente se muestran los resultados obtenidos del control tolerante a fallos implementado en la planta, y finalmente, en la última sección, se presentan las conclusiones y las futuras líneas de trabajo.

2. NMPC

El MPC se enmarca dentro de los controladores óptimos, es decir, aquellos en los que las actuaciones responden a la optimización de un criterio. El criterio a optimizar, o función de coste, está relacionado con el comportamiento futuro del sistema, que se predice gracias a un modelo dinámico del mismo, denominado modelo de predicción (de ahí el término predictivo basado en modelo). El intervalo de tiempo futuro que se considera en la optimización se denomina horizonte de predicción.

Dado que el comportamiento futuro del sistema depende de las actuaciones que se aplican a lo largo del horizonte de predicción, son éstas las variables de decisión respecto a las que se optimiza el criterio. La aplicación de estas actuaciones sobre el sistema conducen a un control en bucle abierto. La posible discrepancia entre el comportamiento predicho y el comportamiento real del sistema crean la necesidad de imponer cierta robustez al sistema incorporando realimentación del mismo. Esta realimentación se consigue gracias a la técnica del horizonte deslizante que consiste en aplicar las actuaciones obtenidas durante un periodo de tiempo, tras el cual se muestrea el estado del sistema y se resuelve un nuevo problema de optimización.

El control predictivo está formado por los siguientes elementos [2]:

Modelo de predicción: es el modelo matemático que describe el comportamiento esperado del sistema. Este modelo puede ser lineal o no lineal, en tiempo continuo o en tiempo discreto, en variables de estado o en entrada salida. El modelo de predicción considerado tiene la forma:

$$x(k+1) = f(x(k), u(k)) \quad (1)$$

siendo $x(k)$ el estado y $u(k)$ la entrada de control al sistema en tiempo k .

El hecho de que el problema de optimización implicado se resuelva mediante el computador, así como la técnica de horizonte deslizante con la que se aplica la solución, hace que sea más natural considerar modelos discretos que continuos.

Función de coste: en esta función se recoge el criterio a optimizar. Compuesto por una función

definida positiva cuyo valor es el coste de una determinada evolución del sistema a lo largo del horizonte de predicción (h_p). Esta función suele tener la forma

$$\begin{aligned} J_N(x_k, u_F(k)) &= \\ &= \sum_{i=0}^{N-1} L(x(k+i|k), u(k+i|k)) + V(x(k+N|k)) \end{aligned} \quad (2)$$

siendo L la función de coste de etapa y V la función de coste terminal (suelen ir multiplicadas por unos pesos que ponderan cada término). Estas funciones son definidas positivas.

Dado que el coste considera el comportamiento del sistema hasta un horizonte N , éste depende del estado actual del sistema $x(k)$ y de la secuencia de N actuaciones que se aplican durante el horizonte de predicción $u_F(k)$. Siendo $u_F(k)$:

$$u_F(k) = \{u(k|k), u(k+1|k), \dots, u(k+N-1|k)\} \quad (3)$$

Restricciones: indican los valores límites entre los que debe permanecer el sistema a lo largo del tiempo. Estos límites se deben a que las variables del sistema no pueden tomar cualquier valor que se quiera, ya sea por límites físicos o bien por motivos de seguridad. Estas restricciones se incorporan a la formulación del problema, casi siempre por motivos económicos, ya que puede ser demasiado costoso modificar el sistema para tener un rango más amplio de las variables del mismo.

Estas restricciones se suelen expresar como conjuntos X y U , generalmente cerrados y acotados, en los cuales deben estar contenidos los estados del sistema y las actuaciones en cada instante, de forma que

$$\begin{aligned} x_k &\in X \forall k \\ u_k &\in U \forall k \end{aligned} \quad (4)$$

Es habitual imponer una restricción sobre el estado terminal del sistema llamada restricción terminal. Ésta viene dada por un conjunto $\Omega \subseteq X$ denominado conjunto o región terminal. Así, esta restricción tiene la forma:

$$x(k+N|k) \in \Omega \quad (5)$$

Finalmente, el problema de optimización a resolver en cada instante queda de la forma:

$$\begin{aligned} & \min_{u_F(k)} J_N(x_k, u_F(k)) \\ & \text{s.a.} \\ & u(k+j | k) \in U \quad j = 0, \dots, N-1 \quad (6) \\ & x(k+j | k) \in X \quad j = 0, \dots, N-1 \\ & x(k+N | k) \in \Omega \end{aligned}$$

Una vez resuelto, se aplican las acciones de control sobre el sistema y en el siguiente período de muestreo se vuelve a repetir la optimización.

3. PCA

Al extraer datos de una planta se obtiene una matriz de datos X (matriz de n observaciones y m variables observadas), que debe ser normalizada a media 0 y varianza 1: X^{norm} .

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

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

Y se descompone en valores singulares:

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

Siendo Λ ($m \times m$) una matriz diagonal que contiene los valores singulares de X^{norm} (que son, también, los valores propios de la matriz de covarianza, S), que representan los valores de varianza del conjunto de datos. Las columnas de V ($m \times m$) son los vectores propios. Si se elige un número a de elementos de la diagonal de Λ que representen un porcentaje suficientemente alto (usualmente un 60% o 70%) de la varianza total de los datos, obtenemos una matriz Λ_a . Tomando un número a de entre las primeras columnas de V , resulta una matriz P , llamada *Matriz de cargas*. La matriz T , de componentes principales de X , se calcula como:

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

Para detectar fallos se dispone de las herramientas estadísticas T^2 , Q y φ . El estadístico T^2 se calcula según la expresión siguiente:

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

Siendo z una nueva observación. Este estadístico da una medida de la variabilidad del proceso capturada por los componentes principales. Si el valor de este parámetro está por debajo de un umbral, se puede decir que el sistema trabaja en condiciones normales. Este umbral, para un nivel de

significación α , es:

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

donde $F_\alpha(a, n-a)$ es el valor crítico superior de la distribución F de Fischer-Snedecor, con n y $n-a$ grados de libertad.

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

$$Q = r^T r$$

donde

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

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

$$Q_\alpha = g * \chi^2(\alpha, h) \quad (7)$$

con:

$$g = \sigma_Q^2 / (2 * \bar{Q}), \quad h = (2 * \bar{Q}^2) / \sigma_Q^2 \quad (8)$$

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

En el caso del estadístico φ , éste es una combinación de los dos estadísticos anteriores, de la forma [8]:

$$\varphi = \frac{Q}{Q_\alpha} + \frac{T^2}{T_\alpha^2} \quad (9)$$

El umbral para éste estadístico se calcula así:

$$\varphi_\alpha = g^\varphi \chi^2(\alpha, h^\varphi)$$

con :

$$\begin{aligned} g^\varphi &= \frac{\frac{a}{(T_\alpha^2)^2} + \frac{\theta_2}{Q_\alpha^2}}{\frac{a}{T_\alpha^2} + \frac{\theta_1}{Q_\alpha}} \\ h^\varphi &= \frac{(\frac{a}{T_\alpha^2} + \frac{\theta_1}{Q_\alpha})^2}{\frac{a}{(T_\alpha^2)^2} + \frac{\theta_2}{Q_\alpha^2}} \end{aligned} \quad (10)$$

$$\theta_i = \sum_{j=a+1}^m \sigma_j^{2i}$$

Siendo σ_j^2 el elemento j de la diagonal de la matriz Λ .

La alarma saltará para T^2 , Q o φ cuando uno de ellos sobrepase su umbral un número determinado de observaciones consecutivas, indicando que el sistema está funcionando de forma anómala. El número de observaciones consecutivas para activar la alarma debe ser lo suficientemente alto para evitar falsas alarmas pero no ser tan elevado que provoque grandes retrasos en avisar del fallo.

Los valores de los umbrales calculados antes son teóricos, y pueden servir como una primera aproximación, pero es posible obtener unos umbrales más realistas con los datos disponibles. Para ello, se toman los datos del sistema funcionando sin fallo y se busca un valor umbral para cada estadístico, de modo que sólo un $\alpha\%$ de las observaciones superen ese umbral. Para este proceso se aplica validación cruzada dando a α un valor del 5%. Finalmente, se elegirá entre el valor teórico y el obtenido por validación cruzada, el de más alto valor.

Cuando se produzcan estas alarmas, interesa saber qué variable ha contribuido más para provocar el fallo. Esto se puede hacer aplicando el diagrama de contribución [5, 11].

4. SENSORES SOFTWARE

Los sensores software utilizados se basan en el método Partial Least Squares (PLS) [4, 7, 9, 13, 15]. Éste es un método para obtener modelos de predicción cuando existe un gran número de variables y hay una gran correlación entre ellas. Se parte de una matriz X de variables en la que se introducen los datos las variables que serán los predictores, y de otra matriz Y que contendrá las variables predichas, es decir las que queremos estimar, y se genera un modelo como el que sigue:

$$X = TP^T + E \quad (11)$$

$$Y = UQ^T + F \quad (12)$$

donde X es una matriz con n observaciones y m variables e Y es una matriz (n observaciones y p variables). T y U son, respectivamente, proyecciones de X y de Y . P y Q son matrices de cargas ortogonales. Finalmente, E y F son los términos de error, supuestos independientes. Estas descomposiciones se hacen buscando que exista una máxima covarianza entre T y U .

El esquema del proceso se representa en la figura 1.

Una vez hechas las descomposiciones se puede obtener una matriz B_2 que sirve para calcular una aproximación de las variables contenidas en Y de la forma:

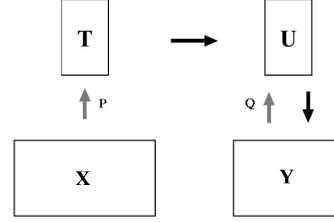


Figura 1: Representación del método PLS

$$\hat{y}(t) = x(t)B_2 \quad (13)$$

Siendo $x(t)$ la observación en el instante t de las variables contenidas en X y $\hat{y}(t)$ es la aproximación de las variables contenidas en la matriz Y .

Para calcular B_2 se aplica la fórmula:

$$B_2 = W(P^T W)^{-1} Q^T \quad (14)$$

W es la matriz de pesos de las variables latentes, calculada con el algoritmo PLS.

En el caso de fallo de un sensor se puede sustituir el valor enviado por éste por una aproximación obtenida a partir de los datos enviados por el resto de sensores (sensor software). Para ello se aplica el método PLS explicado antes, introduciendo en la matriz Y la variable medida por el sensor que se quiere aproximar, y en X el resto de variables.

En el caso de que se detecte un fallo en algún sensor en el estadístico Q (no se usa T^2 ya que no es muy fiable en la identificación, ni φ que, al contener a T^2 , se ve influenciado por éste) se calcula el sensor software correspondiente y se sustituye por el sensor que ha fallado para dotar de tolerancia al sistema y por tanto corregir la medida aportada por el sensor defectuoso.

5. PLANTA DE TRATAMIENTO DE AGUAS RESIDUALES

La propuesta del presente artículo se ha implementado sobre un modelo de una planta de tratamiento de aguas residuales simulada, figura 2. Dicho modelo es el Benchmark Simulation Model 1 (BSM1) desarrollado por el IWA Task Group on Benchmarking of control Strategies for WWTP [1].

Este tipo de instalaciones para el tratamiento de aguas residuales (EDAR) están diseñadas para eliminar los productos químicos y biológicos del agua, de forma que sea posible el uso de este agua tratada para otros fines. La instalación está formada por dos elementos diferentes (Fig 2): un con-

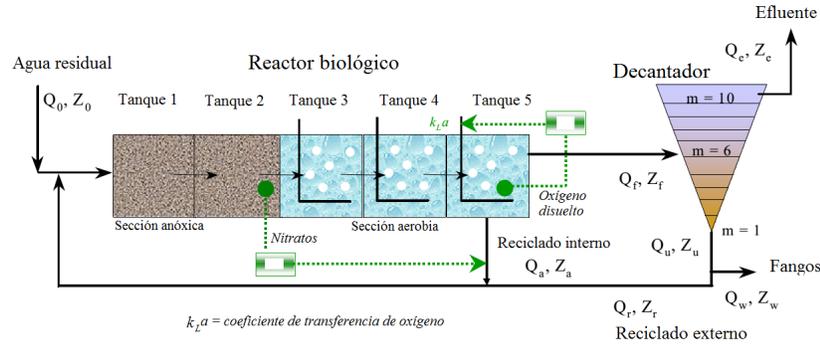


Figura 2: Planta BSM1

junto de reactores de fangos activados y un sedimentador o clarificador. En este sistema el agua residual pasa por los reactores de fangos, de los que los dos primeros son de tipo anóxico y los tres últimos son de tipo aerobio. Una vez el líquido ha pasado por estos reactores, una parte se recicla, enviándola al primer reactor (recirculación interna), y otra parte es introducida en un decantador en el que, por la parte superior, se obtiene agua depurada a la que ya se puede dar uso o verter al medio, y por la parte inferior de este decantador se extraerán unos fangos de los que una parte se introducirá en el primer reactor (recirculación externa) y otra parte se desecha (se reserva para ser tratada o se vierte en un vertedero controlado).

La planta cuenta con un controlador predictivo no-lineal (NMPC) que controla la cantidad de nitratos presentes en el segundo reactor actuando sobre la válvula de la recirculación interna, y la cantidad de oxígeno disuelto en el quinto tanque actuando sobre la tasa de transferencia de oxígeno en ese tanque.

El modelo de la EDAR utilizado para la simulación en lazo abierto sobre la que se aplican las acciones de control, y que simula el comportamiento de una planta real, está implementado en Simulink (Matlab), y algunas partes de este modelo se han cambiado para producir deliberadamente anomalías o fallos en la instalación.

El modelo matemático tiene 15 variables de estado como: material inerte soluble, sustrato lentamente biodegradable, nitrato y nitrito, etc. [1] en cada punto de medición (los puntos de medición incluyen la entrada a la planta, cada uno de los reactores, cada nivel del decantador, etc.). En la realidad no es posible conseguir datos instantáneos de todas esas variables en una planta de aguas residuales, por lo que se ha reducido el número de variables combinando varias de ellas que dan lugar a otras mucho más fáciles de medir instantáneamente en una planta real (Figura 3). El modelo de

simulación se ejecuta a lo largo de 1 día. El control NMPC recalcula las acciones de control cada 0,36 minutos, y el sistema de detección de fallos se ejecuta cada 15 minutos.

Used variables	Variables in the model
COD (Chemical Oxygen demand), g COD.m ⁻³	S _i (inert soluble material), S _r (readily biodegradable substrate), X _i (inert particulate material), X _s (slowly biodegradable substrate)
O ₂ g (-COD).m ⁻³	S _o (dissolved oxygen)
Alk (alkalinity)	S _{alk} (alkalinity)
N (nitrogen), g N.m ⁻³	S _{no} (nitrate and nitrite), S _{nh} (ammonia and ammonium), S _{no} (soluble organic nitrogen associated with S _s)
SS (solids suspended), g SS.m ⁻³	TSS (total suspended solids)
Flow, m ³ .d ⁻¹	Flow rate

Figura 3: Variables medidas en cada bloque de la instalación

5.1. Fallos

En este caso, se han considerado cuatro tipos de fallo, que se han implementado con una magnitud que varía entre el 15 % y 50 %, según el fallo, dando como resultado un conjunto de 4 pruebas con las que probar el método de detección de fallos.

- El primer fallo (F_1) consiste en simular una fuga de un 30 % del caudal de recirculación externa (recirculación de fangos del decantador)
- Otro fallo (F_2) consiste en simular un fallo en el sensor de nitratos presentes en el tanque 2, que dará un valor un 50 % inferior al valor real, lo cual afectará al sistema de control de la planta
- El tercer tipo de fallo (F_3) simula un fallo en el sensor de oxígeno disuelto en el reactor 5, que enviará un valor un 15 % inferior al valor real. Esto también alterará las acciones de control obtenidas por el control predictivo
- El último fallo (F_4) consiste en simular una descenso del 30 % en la alcalinidad del agua

residual que entra en la planta (debido, por ejemplo, a un vertido).

6. RESULTADOS EXPERIMENTALES

Se ha trabajado con un sistema de control basado en NMPC al que se le ha añadido un sistema de detección de fallos, que incluye la implementación de unos sensores software que se usarán en caso de detectar un fallo en alguno de los dos sensores usados por el control (sensor de NO en el reactor 2 y sensor de O_2 en el reactor 5).

Se han hecho 4 simulaciones, cada una de ellas con un fallo implementado, y duran una día cada una, con un tiempo de muestreo de 0,36 minutos, esto da lugar a obtener 4000 datos en cada simulación. Por lo tanto el sistema de control NMPC realizará 4000 cálculos de acciones de control en cada una de las simulaciones. Los parámetros utilizados en el control han sido: horizonte de predicción: 20, horizonte de control: 1, pesos de la función coste de etapa: [1 1] y pesos de la función de coste terminal: [1 0,0001]. (Las salidas de la planta son dos: el valor del NO en el reactor 2 y del O_2 en el reactor 5; mientras que las acciones de control son dos también: el caudal de recirculación interna y factor de aireación del tanque 5). Los valores indicados se han tomado después de realizar diversas pruebas y comprobar que son los que mejor resultado ofrecen.

Los datos para la detección de fallos se toman en 10 puntos de la planta (Fig. 4), en los que se recogen datos para ser analizados, de tal manera que se obtienen 6 variables en los 10 puntos de medida que dan lugar a una matriz de datos con 60 variables. Además se han añadido 4 variables más, por un lado las acciones de control calculadas por el control predictivo, y por otro, los errores de los dos sensores (serán los valores enviados por cada sensor a los que se les resta la referencia de cada uno de ellos: 1 para el sensor de NO y 2 para el sensor de O_2). Ésta toma de datos se realiza cada 15 minutos, por lo tanto se tendrán 96 datos en cada simulación. Los fallos aparecen en el instante 200 de la simulación ($200 \cdot 0,36 = 72$ minutos después de iniciar), como el sistema de fallos toma datos cada 15 minutos, el fallo aparece justo antes de que este sistema de fallos haga la quinta toma de datos, es decir, si todo funciona correctamente, al menos alguno de los estadísticos debería superar el umbral en esa quinta toma de datos.

Se establecido un valor de 15 valores anómalos consecutivos de alguno de los tres estadísticos para que se active la alarma de fallo, es decir, a partir de la quinta muestra si se mantiene algún estadístico

15 muestras más por encima del umbral salta la alarma. Éste valor se ha tomado después de hacer pruebas y se ha visto que es el valor más bajo con el que se evitan falsas alarmas y no se retrasa en exceso la detección. Además, en los casos en los que se detecten fallos en los sensores de los que depende el sistema de control se aplicarán los sensores software. Esto se aplicará con un cierto retraso para poder observar claramente el efecto del sensor software en las gráficas. Se ha tomado un plazo de 100 observaciones ($100 \cdot 0,36 = 36$ minutos después de saltar la alarma de fallo) después de detectarse el fallo para aplicar el sensor software.

Los resultados de la detección e identificación se pueden ver en la tabla 1.

El estadístico T^2 no parece funcionar muy bien, ya que sólo detecta 2 fallos y estos los detecta muy tarde y no los identifica correctamente aunque en el cuarto fallo sí que identifica dónde está el fallo (influyente) pero no la variable correcta.

El estadístico Q detecta los 4 fallos y lo hace muy rápido, ya que el fallo se produce justo antes de la quinta toma de datos y como la alarma necesita 15 observaciones consecutivas, lo más pronto que se puede detectar un fallo es en el instante 19, y Q lo hace así en 3 de los 4 fallos, y en el otro sólo tarda una observación más. En cuanto a la diagnosis identifica correctamente los 4 fallos (en el caso del cuarto fallo identifica un fallo en la alcalinidad del reactor 1, que es justo dónde llega el influente con la alcalinidad inadecuada). El estadístico φ detecta 3 de los 4 fallos, y lo hace rápido, aunque no tanto como Q . En cuanto a la identificación, hace una identificación correcta en 2 de esos 3 fallos.

En cuanto al efecto de los sensores software, estos van a actuar en las simulaciones de los fallos 2 y 3, ya que es dónde hay y se identifican fallos en los sensores de NO y O_2 (con los estadísticos Q y φ). En el caso del fallo 2 el efecto del fallo y del sensor software se observa en la figura 5. Aquí se puede ver cómo aparece el fallo en el instante 200 (que se corresponde con 0,05 días) y cómo se desvía el valor del sensor de la referencia de valor 1. Una vez detectado e identificado el fallo se aplica, 100 observaciones después (0,025 días después a los que se añade el retraso ocasionado por la menor frecuencia de muestreo del sistema de detección de fallos), el sensor software, que aproxima el valor del sensor usando el resto de variables, y se comprueba que el valor estimado del sensor retorna a valores próximos a la referencia.

De forma similar, para el fallo 3 en la figura 6 se observa el efecto del fallo y del sensor software. El fallo aparece a los 0,05 días y el valor del sen-

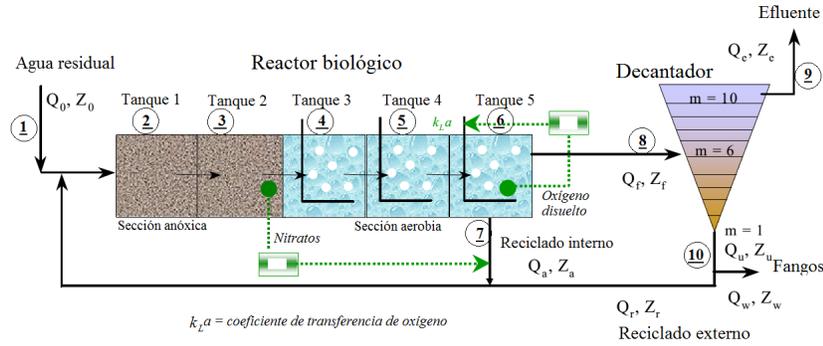


Figura 4: Puntos de medida para la detección de fallos

Cuadro 1: Fallos detectados e identificados

Tipo de fallo	T^2		Q		φ	
	Instante alarma	Variable responsable	Instante alarma	Variable responsable	Instante alarma	Variable responsable
Recirculación externa	50	Influente, Nitrógeno	19	Reactor 1, O_2	20	Error sensor NO
Sensor NO reactor 2	-	-	19	Error sensor NO	-	-
Sensor O_2 reactor 5	-	-	19	Error sensor O_2	19	Error sensor O_2
Alcalinidad influente	61	Influente, Nitrógeno	20	Reactor 1, Alcalinidad	21	Reactor 1, Alcalinidad

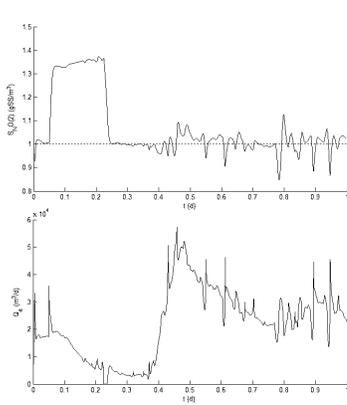


Figura 5: Sensor NO y acción de control asociada, con sensor software

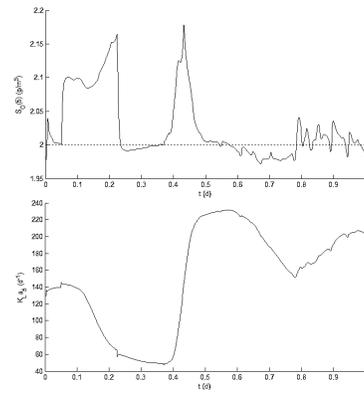


Figura 6: Sensor O_2 y acción de control asociada, con sensor software

El sensor se desvía del valor de referencia (2 g/m^3). Al detectar e identificar el fallo se aplica, con cierto retraso, el sensor software, y se ve que ahora la estimación calculada por el sensor software regresa valores próximos al de referencia (hay un salto unos instantes después de la entrada en funcionamiento del sensor software, pero rápidamente se vuelve a aproximar al valor de referencia).

7. CONCLUSIONES

En este trabajo se muestra un control tolerante a fallos que consiste en un control NMPC con un sistema de detección de fallos y una reconfiguración del controlador que consiste en sustituir el sensor que falla por un sensor software, aplicado a una depuradora de aguas residuales.

Esta detección se ve que funciona razonablemente

bien, teniendo además la ventaja de que se ha incorporado un sistema de sensores software que son capaces de suplir la función de dos sensores que son los que usa el control predictivo para calcular las acciones de control futuras.

Por un lado, se ha visto que el sistema de detección de fallos funciona bien en cuanto a que es rápido y detecta e identifica correctamente todos los fallos (usando el estadístico Q).

Por otro lado, los sensores software han usado los datos del resto de variables de la planta para obtener una estimación del valor medido por los sensores defectuosos. Según lo visto anteriormente, el uso de estos sensores permite que continúe el funcionamiento de la planta en condiciones adecuadas mientras se efectúa la reparación o sustitución del sensor o sensores averiados.

Agradecimientos

Los autores desean agradecer a la Comisión Europea y al Ministerio de Economía y Competitividad de España por su apoyo financiero a través de los proyectos DPI20015-67341-C2-1-R y DPI20015-67341-C2-2-R

Referencias

- [1] J. Alex, L. Benedetti, J. Copp, K.V. Gernaey, U. Jeppsson, I. Nopens, M.N. Pons, C. Rosen, J.P. Steyer, and P. Vanrolleghem. Benchmark Simulation Model no. 1 (BSM1). Technical report, Dept. of Industrial Electrical Engineering and Automation. Lund University., 2008.
- [2] Eduardo F. Camacho and Carlos Bordons Alba. *Model Predictive Control*. Springer-Verlag London, 2007.
- [3] S.X. Ding. *Model based Fault Diagnosis Techniques*. Springer, 2008.
- [4] A. Ferrer, D. Aguado, S. Vidal-Puig, J.M. Prats, and M. Zarzo. Pls: A versatile tool for industrial process improvement and optimization. *Applied Stochastic Models in Business and Industry* 24, 2008.
- [5] D. Garcia-Alvarez, G. I. Sainz-Palmero, M.J. Fuente, P. Vega, and G.I. Sainz. Fault detection and diagnosis using multivariate statistical techniques in a wastewater treatment plant. In *ADCHEM*, 2009.
- [6] Q. He, S. Qin, and J. Wang. A new fault diagnosis method using fault directions in fisher discriminant analysis. *AIChE Journal*, 51(2):555–571, 2005.
- [7] A. Höskuldsson. Pls regression methods. *Journal of Chemometrics* 2, 1988.
- [8] H.Yue and S. J. Qin. Reconstruction based fault identification using a combined index. *Industrial and Engineering Chemistry Research*, 2001.
- [9] Petr Kadleca, Bogdan Gabrys, and Sibylle Strandt. Data-driven soft sensors in the process industry. 2009.
- [10] T. Kourti. Process Analysis and Abnormal Situation Detection: From Theory to Practice. *IEEE Control System Magazine*, pages 10–25, 2002.
- [11] T. Kourti and J.F. MacGregor. Multivariate SPC Methods for Process and Product Monitoring. *Journal of Quality Technology*, 28:409–428, 1996.
- [12] W. Ku, R.H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems*, 30:179–196, 1995.
- [13] Prats-Montalbán, J.M., A. Ferrer, J.L. Malo, and J. Gorbeña. A comparison of different discriminant analysis techniques in a steel industry welding process. *Chemometrics and Intelligent Laboratory Systems*, 80, 2006.
- [14] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A review of process fault detection and diagnosis. Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27:291–311, 2003a.
- [15] D. Zhou, G. Li, and S.J. Qin. Total Projection to Latent Structures for Process Monitoring. *AIChE Journal*, 56(1):168–178, 2010.