DETECCIÓN DE FALLOS CON VALIDACIÓN PROBABILÍSTICA

J. Blesa. Eng.Sistemes, Automàtica i Inf.Ind. U. Politecnica de Catalunya (joaquim.blesa@upc.edu)

A. Luque, T. Alamo. Dpto. Ingeniería de Sistemas y Automática. U. Sevilla (amalia,alamo@cartuja.us.es)

F. Dabbene. CNR-IEIIT. Politecnico di Torino. Italia. (fabrizio.dabbene@polito.it)

Resumen

Presentamos una estrategia general para el diseño de un bloque de detección de fallos con validación probabilística (PCV- Procesado, clasificación, validacion). Se propone un esquema general de PCV, que permite diseñar un bloque de detección de fallos con validación probabilística en el porcentaje máximo de fallos no detectados (impuesto como condición de diseño) y en el porcentaje de falsas alarmas (obtenido a posteriori). En cada iteración del algoritmo secuencial, una solución candidata se valida probabilísticamente mediante un conjunto de muestras generadas aleatoriamente. Presentamos un marco general en el que la solución candidata puede violar las restricciones para un reducido número de elementos del conjunto de validación. Este esquema generalizado muestra significativas ventajas, en particular en términos de la obtención de la solución probabilística.

Palabras clave: algoritmos aleatorios, detección de fallos, clasificadores, validación probabilística

1 INTRODUCCIÓN

Se entiende como fallo todo cambio en el comportamiento de alguno de los componentes del sistema (desviación no permitida de alguna de sus propiedades o parámetros característicos) de manera que éste ya no puede satisfacer la función para la cual ha sido diseñado ([4]). Además de los fallos, existen otros factores que alteran el comportamiento normal del sistema, como las perturbaciones y el ruido. Las perturbaciones son entradas no conocidas que pueden manifestarse en el sistema en cualquier momento pero que se han tenido en cuenta a la hora de diseñar el lazo de control convencional. Cualquier perturbación que no se haya tenido en cuenta en este diseño será considerada como un fallo. El ruido también es una entrada no conocida que se manifiesta en el sistema pero, a diferencia de las perturbaciones, tiene media nula y, además, a priori se puede tener conocimiento de cual es su amplitud. Un sistema de detección de fallos ha de reaccionar frente a los fallos y ser inmune (robusto), en la medida de lo posible, a los otros factores presentes en el sistema que generan incertidumbre. Por otro lado, muchos de los métodos de detección de fallos se basan en un modelo (matemático o cuantitativo) del sistema a monitorizar que nunca podrá describir de manera exacta el comportamiento del sistema real y por lo tanto presentará un error de modelado que también se deberá tener en cuenta.

El objetivo de un bloque de detección de fallos es, una vez se ha producido un fallo en un instante T_F, detectarlo en un intervalo de tiempo menor o igual a $T_{D_{max}}$ fijado previamente. Dependiendo de la magnitud e incidencia de los fallos que se deseen detectar y de la presencia de otros factores de incertidumbre en el sistema, no siempre será posible diseñar un bloque de detección que detecte todos los fallos sin que en situaciones de no fallo se activen falsas alarmas. Así que siempre existirá un compromiso entre la proporción de fallos que no se detecten (MF "Missed Faults") y la proporción de veces que se active el bloque detector sin la presencia de fallos debido a los factores de incertidumbre presentes en el sistema (FA "False alarms"). En este compromiso que se deberá tener en cuenta en el proceso de diseño del bloque detector de fallos es lógico priorizar la minimización de fallos no detectados respecto a la minimización de falsas alarmas.

La naturaleza aleatoria de los fallos y las incertidumbres inherentes del sistema convierten el problema de diseño del bloque de detección en un problema de robustez.

Típicamente, para un problema de robustez, los parámetros de diseño, así como diferentes variables auxiliares, son descritos en términos de un vector de variables de decisión θ, que se denota como "parámetro de diseño", y es restringido al conjunto Θ. Por otro lado, la incertidumbre w está acotada en el conjunto $\mathcal{W}$. Es decir, cada elemento $w \in \mathcal{W}$ representa una de las realizaciones admisibles de la incertidumbre, con probabilidad $\Pr_{\mathcal{W}}$. En nuestro contexto de detección de fallos, θ corresponde a las variables de decisión que determinan el bloque de detección de fallos. Dicho bloque permite determinar si hay un fallo o no en

un determinado escenario, por lo tanto tendremos dos conjuntos de incertidumbre $\mathcal{W}_F$ y $\mathcal{W}_N$ que consisten en todos los posibles escenarios de funcionamiento del sistema a monitorizar con fallo y sin fallo respectivamente. Por otro lado, w_F y w_N representan una realización de un escenario con fallo y sin fallo. $\mathcal{W}_F$ y $\mathcal{W}_N$ tienen asociados unos espacios de probabilidad $\Pr_F$ y $\Pr_N$ respectivamente.

Además consideramos también dos funciones binarias medibles:

$$g(\theta, w) := \begin{cases} 0 & \text{si } \theta \text{ detecta fallo} \\ 1 & \text{en otro caso.} \end{cases}$$

$$h(\theta, w) := \begin{cases} 0 & \text{si } \theta \text{ no detecta fallo} \\ 1 & \text{en otro caso.} \end{cases}$$

Al aplicar estas dos funciones sobre los espacios $\mathcal{W}_F$ y $\mathcal{W}_N$ se obtienen las siguientes esperanzas

$$E_g(\theta) := \Pr_F\{w_F \in \mathcal{W}_F : g(\theta, w_F) = 1\}$$

$$E_h(\theta) := \Pr_N\{w_N \in \mathcal{W}_N : h(\theta, w_N) = 1\}.$$

Donde $E_g(\theta)$ y $E_h(\theta)$ son el tanto por uno de fallos no detectados (MF) y falsas alarmas (FA) respectivamente. La utilidad de los algoritmos aleatorios surge del hecho de poder tratar el siguiente problema de diseño

$$\min_{\theta \in \Theta} E_h(\theta) \text{ sujeto a } E_g(\theta) \leq \eta_F \qquad (1)$$

donde η_F es el tanto por uno máximo de fallos no detectados impuesto como requerimiento del bloque detector.

En este marco, se pueden extraer N_N y N_F i.i.d. muestras (independientes e identicamente distribuidas) $\{w_N^{(1)}, \ldots, w_N^{(N_N)}\}$ de $\mathcal{W}_N$ y $\{w_F^{(1)}, \ldots, w_F^{(N_F)}\}$ de $\mathcal{W}_F$ de acuerdo a la probabilidad $\Pr_N$ y $\Pr_F$ respectivamente y con una proporción entre escenarios de fallo y no fallo $F_N = \frac{N_F}{N_N}$ determinada por la probabilidad de fallo del sistema a monitorizar. De esta manera se puede resolver el siguiente problema de optimización muestreado

$$\min_{\theta \in \Theta} \sum_{\ell_N=1}^{N_N} h(\theta, w_N^{(\ell_N)}) \qquad (2)$$

$$\text{sujeto a } \sum_{\ell_F=1}^{N_F} g(\theta, w_F^{(\ell_F)}) \leq \eta_F N_F$$

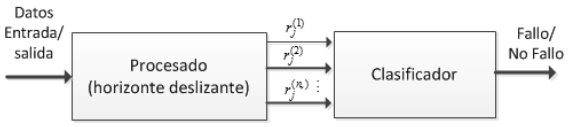

Figura 1: Esquema del detector de fallos en linea

La idea de permitir algunas violaciones de las restricciones no es nueva y puede encontrarse, por ejemplo, en el contexto de identificación [3].

En este artículo se propone un método de diseño del bloque detector de fallos basado en la utilización de históricos o simulaciones de episodios reales con fallo y sin fallo evitando la dificultad del análisis, que no siempre es posible, debido a la complejidad del problema.

El resultado así obtenido, mediante un test de validacion probabilística, garantiza que la solución propuesta se comporta de la manera deseada con una cierta probabilidad, fijada a priori. Se garantiza asimismo la satisfacción probabilística de las restricciones. Esta técnica resulta muy adecuada para el abordaje de problemas complejos.

Finalmente este artículo ilustra la metodología propuesta con la presentación de una aplicación, en este caso, al diseño de un detector de fallos con garantía probabilística en un depósito virtual que modela el comportamiento de un colector de una red de alcantarillado tal y como se detalla en [8].

2 ESQUEMA DE DISEÑO PROPUESTO

Tal y como se ha comentado anteriormente, el objetivo de este artículo es el de proponer un esquema que permita diseñar un bloque de detección de fallos con validación probabilística en el porcentaje máximo de fallos no detectados (impuesto como condición de diseño) y en el porcentaje de falsas alarmas (obtenido a posteriori). El funcionamiento de este bloque de detección de fallos se describe en el esquema de la Figura 1. De los datos de entrada/salida del sistema a monitorizar cada instante de tiempo j se extraen unos atributos o indicadores $(r_j^{(1)}, r_j^{(2)}, ..., r_j^{(n_r)})$ de éste mediante un bloque de procesado que actúa en horizonte deslizante en el tiempo. Estos indicadores que pueden ser de distinta naturaleza son sensibles a la presencia de fallos pero también al ruido, perturbaciones y otros factores como errores de modelado. Por esta razón son introducidos en un clasificador que determina si se ha producido un fallo o no en el sistema.

Por otro lado, el algoritmo del proceso fuera de linea para lograr diseñar el bloque detector se des-

cribe en el esquema de la Figura 2. En primer lugar se diseña el bloque de procesado o lo que es lo mismo se determina que indicadores serán útiles con el fin de detectar los fallos presentes en el sistema. Para este paso se debe utilizar el conocimiento de la planta y técnicas básicas de detección de fallos [5], [6]. La naturaleza de los indicadores a utilizar puede ser muy diversa : residuos entre medidas y estimaciones, señales características para detectar fallos como vibraciones, señales sonoras, etc..., magnitudes que determinen un cambio de operación de la planta como la temperatura, humedad, etc...

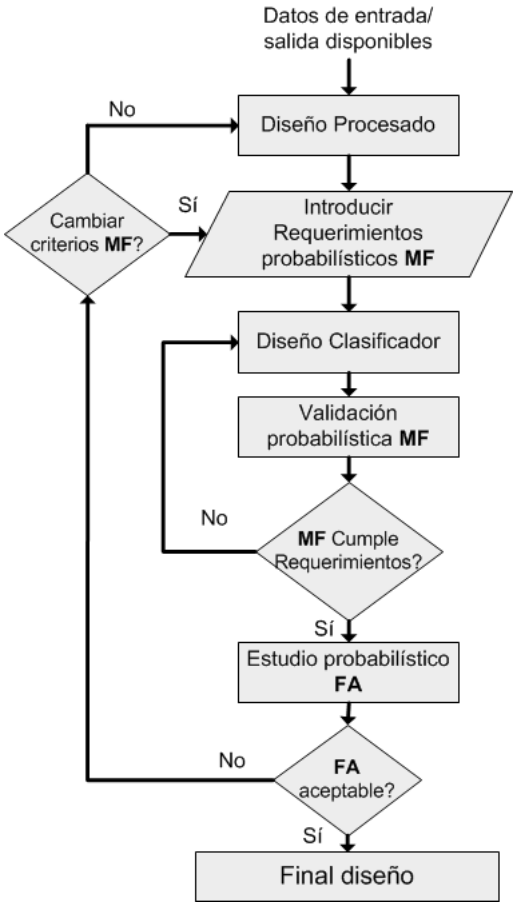

Figura 2: Esquema diseño fuera de linea del bloque de detección de fallos

Una vez determinados los indicadores que se quieren extraer de la información disponible de la planta, se deben especificar los requerimientos probabilísticos de la cota máxima en porcentaje de fallos no detectados (MF) y la garantía probabilística mínima de cumplimiento. A continuación se procede a diseñar un clasificador al que se le suministran datos de la planta en situaciones de funcionamiento con y sin fallo (reales o simuladas). Mediante un proceso de optimización se minimiza el número de falsas alarmas (FA) y se impone (como restricción) un número máximo de fallos no detectados (MF) se calibra adecuadamente el clasificador. A continuación se evalúa el clasificador obtenido mediante un test de validación probabilística con datos diferentes a los utilizados en el diseño de calibración del clasificador. Si no se pasa el test de validación probabilística se vuelve a la etapa de diseño del clasificador. Si se pasa el test de validación probabilística, el clasificador cumple las especificaciones de MF con las garantía probabilísticas impuestas y se procede a un estudio probabilístico del número de falsas alarmas. Si el resultado no és satisfactorio (número de falsas alarmas elevado) hay dos opciones: cambiar (relajar) los requerimientos probabilísticos de MF o mejorar el diseño de procesado (obtener mejores indicadores).

3 DISEÑO DEL PROCESADO

Como se ha comentado anteriormente, los indicadores extraídos del sistema a monitorizar pueden ser de diferente naturaleza. En este apartado se ponen dos ejemplos de posibles indicadores que se utilizan en los métodos de detección de fallos basada en modelos matemáticos. En este tipo de métodos se comprueba la consistencia o no consistencia del modelo del sistema a monitorizar con las medidas obtenidas de dicho sistema.

Por ejemplo, si consideramos que el sistema a monitorizar se puede describir mediante el modelo de regresión en tiempo discreto

$$y(j) = \varphi^T(j)\beta_0 + e(j), \;\; j = 1, \ldots, M \qquad (3)$$

donde

- $y(j)$ es la medida de la salida
- $\varphi(j)$ es el vector de regresión de dimensión n_β función de las entradas $u(j)$ y salidas $y(j)$
- β_0 es el vector de parámetros nominales de dimensión n_β
- $e(j)$ es el error aditivo que contiene el ruido de las medidas y el error de modelado

Entonces esta consistencia se puede evaluar calculando en cada instante j el residuo $\Delta(j)$ entre la salida medida y la estimación dada por el modelo

$$\Delta(j) = y(j) - \varphi^T(j)\beta_0. \qquad (4)$$

En un caso ideal, el residuo debería de ser diferente de cero solo en el caso de que un fallo estuviera

presente en el sistema. Sin embargo, debido a la presencia de ruido y error de modelado el residuo puede ser diferente de cero cuando no haya presencia de fallo y el método de detección debe de ser robusto. Una manera de abordar este problema es aplicar las técnicas Set-Membership que consideran el error $e(j)$ desconocido pero acotado ([7]), esto es

$|e(j)| \leq \sigma \; j = 1, \ldots, M.$

De esta manera se puede aplicar el siguiente test de detección de fallos

$$\begin{cases} \text{Si } |\Delta(j)| \leq \sigma \Rightarrow \text{NoFallo} \\ \text{Si } |\Delta(j)| > \sigma \Rightarrow \text{Fallo.} \end{cases}$$

Y por lo tanto el diseño del diagnosticador consiste en elegir una cota σ adecuada. A este test se le denomina test directo.

Otra manera de mirar la consistencia del modelo con las medidas en el espacio de parámetros, mediante una ventana temporal de N muestras se puede hacer una estimación de parámetros

$$\hat{\beta}(j) = \left(\Phi^T(j)\Phi(j)\right)^{-1}\Phi^T(j)Y(j) \qquad (5)$$

donde $\Phi(j) = \begin{pmatrix} \varphi^T(j-N) \\ \vdots \\ \varphi^T(j) \end{pmatrix}$ y

$Y(j) = \begin{pmatrix} y(j-N) \\ \vdots \\ y(j) \end{pmatrix}$

y de la misma manera que en el residuo temporal, se puede calcular un residuo de parámetros respecto a un modelo nominal

$\Delta\beta(j) = \hat{\beta}(j) - \beta_0$

y así definir el siguiente test de detección

$$\begin{cases} \text{Si } \Delta\beta(j) \in \text{B} \Rightarrow \text{NoFallo} \\ \text{Si } \Delta\beta(j) \notin \text{B} \Rightarrow \text{Fallo.} \end{cases}$$

donde B es el conjunto de incertidumbre de parámetros debido al error aditivo $e(j)$ y a la poca riqueza de los datos usados en la identificación.

Tanto el test directo como el test inverso se pueden incluir en el esquema general 1 incluyendo los términos procesando formular en forma de clasificador elegiendo $r_j^{(1)} = \Delta(j)$ para el test directo y $r_j^{(2)} = \Delta\beta(j)$ para el test inverso. Además tests básicos como valores máximos, fijos o variaciones máximas en los datos obtenidos en los sensores se pueden implementar escogiendo los indicadores adecuados.

4 DISEÑO DEL CLASIFICADOR

Con el fin de conseguir una buena discriminación entre los escenarios sin fallo y los escenarios con fallo, se propone utilizar un clasificador basado en dos etapas: una estática y otra dinámica. Tal y como se muestra en la Figura 3.

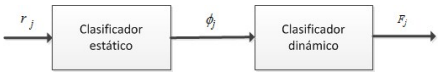

Figura 3: Esquema del clasificador propuesto

El objetivo del clasificador estático es determinar en un instante de tiempo j con el vector de indicadores en ese mismo instante de tiempo r_j si la situación es sintomática de fallo o no mediante una señal binaria ϕ_j. Si la situación es sintomática de fallo $\phi_j = 1$ y si no $\phi_j = 0$. Esta operación se podría realizar mediante una función de entrada vectorial analógica y salida binaria h_{est}

$\phi_j = h_{est}\left(r_j\right)$

donde $r_j^T = (r_j^{(1)}, r_j^{(2)}, ..., r_j^{(n_r)})^T$.

El objetivo del clasificador dinámico de orden T_p es el de determinar en el instante j si se ha producido un fallo o no con las últimas T_p señales sintomáticas. Esto es:

$F_j = h_{din}\left(\phi_j, \phi_{j-1}, \cdots, \phi_{j-T_P+1}\right).$

Donde h_{din} es una función de entrada y salidas binarias y, por la definición de detectabilidad, se cumple que $T_P \leq T_{D_{\max}}$. El motivo de dividir en dos partes el clasificador es el de hacerlo lo más robusto posible respecto a las falsas alarmas sin empeorar las prestaciones de detectabilididad (fallos no detectados).

Para diseñar estos dos clasificadores se dispondrá de escenarios sin fallos y con fallos $\{w_N^{(1)}, \ldots, w_N^{(N_N)}\}$ de $\mathcal{W}_N$ y $\{w_F^{(1)}, \ldots, w_F^{(N_F)}\}$ de $\mathcal{W}_F$ respectivamente. Cada escenario consiste en una secuencia de indicadores ${r_1}^{(i)}, {r_2}^{(i)}, \cdots, {r_{T_{Si}}}^{(i)}$ donde T_{Si} es el número de instantes de tiempo que dura el escenario i. El conjunto de todos indicadores de todos los escenarios se puede dividir en dos suconjuntos: el conjunto de los indicadores que están contaminados por un fallo y el conjunto de indicadores que están libres de fallos $\Re_F$ y $\Re_N$ respectivamente. Se cumple que todos los indicadores que pertenecen a un escenario libre de fallo pertenecen al conjunto $\Re_N$, mientras que en los indicadores que pertenecen a un escenario con fallo parte de ellos están contaminados por el efecto de un fallo (y por lo tanto pertenecen a $\Re_F$) y parte de ellos están libres de efectos de fallo (y por lo tanto pertenecen a $\Re_N$).

El objetivo del clasificador estático es discriminar lo máximo posible entre los indicadores pertenecientes a los dos conjuntos ($\Re_F$ y $\Re_N$). Esto se puede conseguir mediante la elección adecuada de un vector λ que cumpla lo mejor posible las siguientes restricciones

$$f^T(r)\lambda > 0 \Rightarrow r \in \Re_F$$
$$f^T(r)\lambda \leq 0 \Rightarrow r \in \Re_N \ ,$$

donde $f(r)$ es una expresión vectorial de r. El valor de λ se puede hallar mediante el siguiente problema de optimización convexa

$$\min_{\lambda} \left(\sum_{r \in \Re_F} e^{-\tau f^T(r)\lambda} + \sum_{r \in \Re_N} e^{f^T(r)\lambda} \right) \qquad (6)$$

donde $\tau \in (0, \infty)$ es una constante que se ha determinado previamente a la resolución del problema de optimización y permite penalizar en mayor o menor medida los errores de clasificación de los indicadores $r \in \Re_F$ respecto a los errores de clasificación de indicadores $r \in \Re_N$ y por lo tanto priorizar más o menos el comportamiento del sistema respecto a los fallos no detectados o respecto a las falsas alarmas. En concreto, cuanto mayor sea τ más penalización se dará a los indicadores clasificados como situación normal, o sea fallos no detectados, y por lo tanto menos peso dará a las falsas alarmas.

Como se ha comentado anteriormente, en el proceso de diseño del detector de fallos se impondrá un valor máximo de fallos no detectados (MF definido por η_F) y esto se conseguirá eligiendo un τ adecuado. El problema del clasificador estático es que al tomar la decisión en función de los indicadores en un determinado instante, la imposición de bajo número de fallos implique una gran sensibilidad a las incertidumbres del sistema y se traduzca en un elevado número de falsas alarmas. El segundo clasificador tiene como objetivo filtrar el efecto de estas incertidumbres y permitir un diagnóstico lo más robusto posible utilizando las T_P últimas salidas del diagnosticador estático. Como la función del diagnosticador h_{din} és una función de entrada binaria y salida binaria con 2^{T_P} combinaciones, éstas se pueden probar de manera exhaustiva y encontrar cual presenta mejores prestaciones entre todas.

Por lo tanto el diseño del clasificador se reduce en hallar el mínimo valor de τ que cumple con el máximo número de falsas alarmas permitidas. Al bloque de detección de fallos ebtenido lo denominaremos θ.

5 VALIDACIÓN

5.1 Algoritmos secuenciales con validación probabilística en MF

En esta sección se presenta una familia general de algoritmos aleatorios, que llamamos algoritmos SPV (del inglés "**S**equential **P**robabilistic **V**alidation algorithms"), ver [1].

La principal característica de este tipo de algoritmos es que están basados en un paso de validación probabilística.

Cada iteración de un algoritmo SPV se compone del cálculo de una solución candidata para el problema y un paso de validación. Los resultados son fundamentalmente independientes de la estrategia concreta elegida para obtener las soluciones candidatas.

El objetivo de utilizar estos algoritmos en este contexto es el de garantizar un cota máxima en de fallos no detectados (MF) definidos por η_F con una la garantía probabilística mínima de cumplimiento $1 - \delta_F$.

La precisión $\eta_F \in (0, 1)$ y confianza $\delta_F \in (0, 1)$ requeridas para la solución probabilística juegan un papel relevante en la determinación del tamaño muestral (número de escenarios con fallo a evaluar) para cada paso de iteración. Se puede proponer un esquema de validación que garantice que para una precisión η_F y confianza δ_F dadas (impuestas en el esquema de diseño), todas las soluciones probabilísticas obtenidas mediante el algoritmo SPV tienen una probabilidad de violación no mayor que η_F con probabilidad no menor que $1 - \delta_F$.

Cada iteración del algoritmo es enumerada mediante el entero k. Llamamos m_k al número de violaciones permitidas en el paso de validación de iteración k. Asumimos que m_k es dado como función de k, es decir, $m_k = m(k)$ donde la función $m : \mathbb{N} \to \mathbb{N}$ es dada. Denotamos M_k el tamaño muestral para el paso de validación de la iteración k. Asumimos que M_k es dado como función de k, η_F y δ_F. Es decir, $M_k = M(k, \eta_F, \delta_F)$ donde $M : \mathbb{N} \times \mathbb{R} \times \mathbb{R} \to \mathbb{N}$ tiene que ser apropiadamente diseñada para garantizar las propiedades probabilísticas del algoritmo. Denotamos las funciones $m(\cdot)$ y $M(\cdot, \cdot, \cdot)$ como *función de nivel* y *función de cardinalidad* respectivamente.

Estructura de un algoritmo SPV:

(i) Fijar $\eta_F \in (0, 1)$ y $\delta_F \in (0, 1)$ a los valores deseados. Poner $k = 1$.

(ii) Obtener una solución candidata $\hat{\theta}_{F_k}$ al problema de optimización robusta (1).

(iii) $m_k = m(k)$ y $M_k = M(k, \eta_F, \delta_F)$.

(iv) Obtener un conjunto de validación $\mathcal{V}_k = \{v^{(1)}, \ldots, v^{(M_k)}\}$ extrayendo M_k i.i.d muestras de validación de $\mathcal{W}_F$ según la probabilidad $\Pr_{\mathcal{W}_F}$.

(v) Si $\sum_{\ell=1}^{M_k} g(\hat{\theta}_{F_k}, v^{(\ell)}) \leq m_k$, entonces $\hat{\theta}_{F_k}$ es una solución probabilística.

(vi) Salir si se satisface la condición de salida.

(vii) $k = k + 1$. Ir a (ii).

La Figura 4 muestra la parte del esquema de diseño general propuesto que corresponde al algoritmo SPV donde se indica los pasos del algoritmo que corresponden a cada bloque.

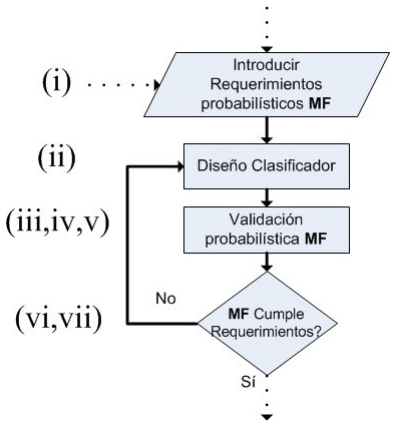

Figura 4: Esquema validación probabilística fallos no detectados

Aunque la condición de salida del algoritmo puede ser bastante general, una estrategia razonable es salir después de un número dado de soluciones candidatas hayan sido clasificadas como soluciones probabilísticas o si se supera un cierto tiempo computacional desde el comienzo del algoritmo. Después de salir se puede elegir entre las soluciones probabilísticas aquella que optimice un determinado índice de desempeño. En la siguiente subsección se propone una estrategia para elegir la cardinalidad del conjunto de validación en la iteración k de modo que, con probabilidad no menor que $1 - \delta_F$ todas las soluciones clasificadas como soluciones probabilísticas por el algoritmo satisfagan la precisión η_F.

5.2 Tamaño muestral

Consideremos un algoritmo SPV con precisión $\eta_F \in (0, 1)$, confianza $\delta_F \in (0, 1)$ y función de nivel $m(\cdot)$ dadas. Entonces, la función de cardinalidad

$$M(k, \eta_F, \delta_F) = \left\lceil \frac{1}{\eta_F} \left(m(k) + \ln \frac{1}{\delta_F \mu(k)} + \sqrt{2m(k) \ln \frac{1}{\delta_F \mu(k)}} \right) \right\rceil,$$

donde

$$\mu(k) = \frac{1}{\xi(\alpha) k^\alpha},$$

donde $\xi(\cdot)$ es la función zeta de Riemann, y $\alpha > 1$,

garantiza que, con probabilidad mayor que $1 - \delta_F$ todas las soluciones probabilísticas obtenidas mediante el algoritmo SPV tienen una probabilidad de violación (fallos no detectados) no mayor que η_F [1]. La función $\mu(k)$ puede adoptar otras expresiones, ver [1].

5.3 Falsas alarmas

Una vez diseñado un clasificador que cumple los criterios probabilísticos respecto a los fallos no detectados, se puede utilizar el resultado obtenido en [2] donde se proporciona la complejidad muestral que caracteriza cómo una media empírica converge en probabilidad a la verdadera probabilidad de violación (en este caso falsas alarmas), para determinar la precisión $\eta_N \in (0, 1)$ y confianza $\delta_N \in (0, 1)$ en términos de falsas alarmas (FA).

Dados $\hat{\theta} \in \Theta$ hallado en el proceso de diseño del bloque de detección y un conjunto de N_N escenarios libres de fallo, se cumple

$$N_N \geq \frac{\ln \frac{1}{\delta_N}}{(\sqrt{\eta_N} - \sqrt{\rho_N})^2}$$

con $0 \leq \rho_N < \eta_N < 1$ donde ρ_N es la proporción de falsas alarmas obtenidas al aplicar el bloque de detección diseñado a los N_N escenarios libres de fallo disponibles

$$\rho_N = \frac{\text{Número de falsas alarmas}}{N_N}.$$

entonces

$\Pr_{\mathcal{W}^N}\{\mathrm{w} \in \mathcal{W}^N \ : \ \hat{E}(\hat{\theta}, \mathrm{w}) \leq \rho_N \text{ y } E(\hat{\theta}) > \eta_N\} \leq \delta_N$.

Por lo tanto, si nuestra estimación de falsas alarmas $\hat{E}(\hat{\theta}, \mathrm{w})$ está dada por el valor ρ_N, se garantiza que la discrepancia entre este valor y la probabilidad de que el número real de falsas alarmas $E(\hat{\theta})$

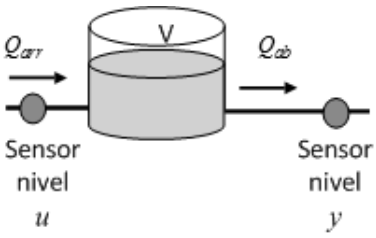

Figura 5: Depósito virtual

sea mayor que otro valor $\eta_N > \rho_N$ fijado a priori, está acotada por δ_N.

Si imponemos una δ_N deseada, la garantía estadística η_N queda determinada por las inecuaciones anteriores. En concreto

$$\eta_N = \left(\left(\frac{\ln(\frac{1}{\delta_N})}{N_N} \right)^{\frac{1}{2}} + \sqrt{\rho_N} \right)^2 \qquad (7)$$

6 RESULTADOS

Para demostrar la eficacia de la metodología presentada en este artículo se ha aplicado al diseño de un bloque de detección de fallos con garantía probabilística en el depósito virtual representado en la Figura 5 que modeliza el comportamiento de un colector de una red de alcantarillado tal y como se detalla en [8].

Se han tenido en cuenta fallos en los sensores de entrada y de salida ($f_u(t)$, $f_y(t)$) y fallos paramétricos ($f_a(t)$, $f_b(t)$) tal y como se indica en la Figura 6. El comportamiento del sistema real considerando estos fallos se puede describir mediante el siguiente modelo en tiempo discreto

$$\tilde{y}(j) = (\tilde{a} + f_a(j-1))\,\tilde{y}(j-1) + \left(\tilde{b} + f_b(j-1)\right)\tilde{u}(j-1) + e_d(j)$$

donde

- $\tilde{a}$, $\tilde{b}$ son los parámetros reales del sistema
- $\tilde{u}(j)$, $\tilde{y}(j)$ son la entrada y salida reales del sistema
- $e_d(j)$ es el error de discretización que dependerá del tiempo de muestreo
- $e_u(j)$, $e_y(j)$ son los errores aditivos introducidos por los sensores

Así pues, con las medidas disponibles

$$y(j) = \tilde{y}(j) + f_y(j) + e_y(j)$$
$$u(j) = \tilde{u}(j) + f_u(j) + e_u(j)$$

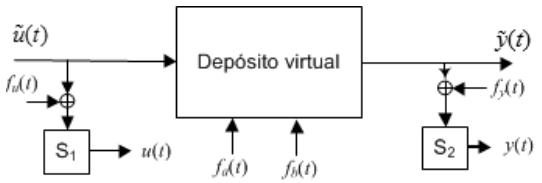

Figura 6: Posibles fallos en el sistema depósito virtual

se puede utilizar el siguiente modelo para describir el comportamiento del sistema real

$$y(j) = a_0 y(j-1) + b_0 u(j-1) + e(j) \qquad (8)$$

donde

- a_0, b_0 son los parámetros del modelo obtenidos con datos de entrada salida sin fallo del sistema
- $e(j)$ es el error del modelo que tiene en cuenta los errores $e_d(j)$, $e_u(j)$, $e_y(j)$ y la posible discrepancia entre los parámetros reales ($\tilde{a}$,$\tilde{b}$) y del modelo (a_0,b_0)

Con el fin de obtener los datos del funcionamiento normal y con fallo se ha implementado un simulador en el entorno Matlab-Simulink, donde se han introducido escenarios (con y sin fallo) basados en situaciones reales, teniendo posibles errores entre el modelo y sistema real. El objetivo es diseñar un detector de fallos que cumpla $MF < 1\%$ con una probabilidad no menor del $1-10^{-6}$ ($\eta_F = 0.01$ $\delta_F = 10^{-6}$) con un retardo máximo en la detección del fallo de $T_{D_{max}} = 5$ muestras.

En primer lugar, mediante los datos obtenidos de entrada/salida del proceso se ha estudiado cuales son los indicadores que nos permitirían distinguir entre las situaciones de fallo y de funcionamiento normal. Las Figuras 7 y 8 muestran los indicadores de error en la identificación de parámetros $r_j^{(1)} = (\Delta a(j), \Delta b(j))^T$ ($\Delta a(j) = \hat{a}(j) - a_0$ y $\Delta b(j) = \hat{b}(j) - b_0$) tomando una ventana temporal de 50 muestras, para escenarios con fallo y sin fallo respectivamente. Además, en estas gráficas también se muestra la región $f^T(r)\lambda \leq 0$ obtenida al diseñar un clasificador óptimo. Sólo con este indicador se pueden detectar el 65% de los fallos con un nivel de Falsas alarmas muy bajo (0.001%).

Una de las ventajas del uso de clasificadores es que, con el fin de mejorar la discriminación entre escenarios con y sin fallo, se pueden añadir otros indicadores. En nuestro caso además de indicador descrito anteriormente han añadido otros indicadores como el mismo error en la estimación de

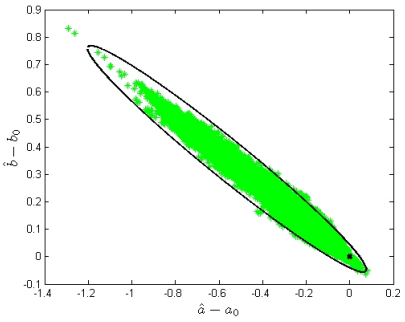

Figura 7: Indicador residuo de parámetros en escenarios sin fallo y conjunto $f^T(r)\lambda \leq 0$

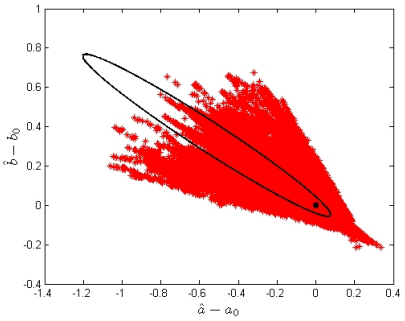

Figura 8: Indicador residuo de parámetros en escenarios con fallo y conjunto $f^T(r)\lambda \leq 0$

parámetros pero con ventanas de 15 y 100 muestras, el residuo temporal (3) utilizando el modelo dado por (6) y acumulando este residuo con ventanas de 15, 50 y 100 muestras. Así como valores máximos en las medidas determinados por el calado máximo admisible en el colector.

Una vez decididos los indicadores a utilizar por el clasificador, se ha utilizado el Algoritmo 2 para hallar el clasificador que cumpla los requisitos probabilísticos en MF definidos anteriormente (a priori) obteniendo $\tau = 7.1$. Al aplicar el clasificador diseñado al conjunto de escenarios sin fallo del sistema se obtiene mediante (7), a posteriori, un nivel de falsas alarmas $FA < 4.5\%$ con una probabilidad no menor del $1-10^{-6}$ ($\eta_N = 0.045$ $\delta_N = 10^{-6}$)

7 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha propuesto una metodología general para el diseño de detectores de fallos con garantía probabilística. La gran ventaja de la metodología propuesta es, por un lado su flexibilidad en introducir herramientas de diferente tipo en la detección de fallos y por otro la garantía probabilística certificada del detector propuesto. El funcionamiento de esta metodología se ha ilustrado con su aplicación a un ejemplo de depósito virtual. Como trabajo futuro, siguiendo la filosofía del esquema propuesto, sería interesante abordar el problema de diseñar un diagnosticador que permita determinar, una vez que se ha detectado un fallo, que tipo de fallo se ha producido con una determindad garantía probabilística.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el proyecto DPI-2011-26243. Los autores quieren agradecer a Vicenç Puig y a Roberto Tempo por sus aportaciones a la investigación presentada en este trabajo.

Referencias

[1] T. Alamo, A. Luque, D. R. Ramirez and R. Tempo. Randomized control design through probabilistic validation. *Proceedings of the American Control Conference*, Montreal, Canada, 2012.

[2] T. Alamo, R. Tempo and A. Luque. On the sample complexity of randomized approaches to the analysis and design under uncertainty. *Proceedings of the American Control Conference*, Baltimore, USA, 2010.

[3] E. Bai, H. Cho, R. Tempo and Y. Ye. Optimization with few violated constraints for linear bounded error parameter estimation. *IEEE Transactions on Automatic Control*, 47(7):1067–1077, 2002.

[4] Blanke, M. Fault-tolerant control systems. *New Trends in Advanced Control* Springer-Verlag, 1999

[5] R. Isermann. Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. Berlin, Germany: Springer, 2006

[6] R. Isermann Fault-Diagnosis Applications. Springer-Verlag, Berlin, Heidelberg, 2011

[7] M. Milanese, J. Norton, H. Piet Lahanier, E. Walter. Bounding approaches to system identification. *Plenum Press*. New York, USA, London, UK, 1996

[8] Puig, V.; Blesa, J. Limnimeter and Rain Gauge FDI in Sewer Networks using an Interval Parity Equations based Detection Approach and an Enhanced Isolation Scheme. *Control engineering practice* 21 (2), 146-170.