

# Integración de técnicas basadas en modelos para la determinación de la diagnosis mínima de un sistema

R. Ceballos, M.T. Gómez-López, R. M. Gasca y C. Del Valle

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla  
Escuela Técnica Superior de Ingeniería Informática,  
Avenida Reina Mercedes s/n 41012 Sevilla (Spain)

## Resumen

Actualmente son dos las comunidades que trabajan en la diagnosis basada en modelos, FDI y DX. Aunque sus investigaciones se desarrollan en paralelo, suelen trabajar por separado. Este trabajo propone una integración de las técnicas provenientes de ambas metodologías. En un primer paso el sistema es dividido en conjuntos de componentes de forma que el proceso de diagnosis puede ser realizado de forma independiente en cada conjunto. A su vez, cada uno de estos conjuntos será dividido en nodos para lograr agrupar aquellos componentes donde no resulta posible diferenciar fallos simples. De esta forma se consigue aumentar de forma automática la eficiencia del proceso de diagnosis, sin disminuir la diagnosticabilidad del sistema, ya que la información relevante para el proceso de diagnosis es mantenida. Combinando adecuadamente los nodos detectados previamente, se obtienen nuevas restricciones donde sólo aparecerán variables asociadas a sensores del sistema. Estas restricciones permiten generar una matriz de firmas que confiere la base para la obtención de la diagnosis del sistema. La diagnosis mínima se genera usando la matriz de firmas y una serie de reglas establecidas. La evaluación de la matriz de fallos y del conjunto de reglas se hace en línea, el resto del proceso puede hacerse de forma automática y fuera de línea.

**Palabras clave:** Diagnosis basada en modelos, FDI, DX, diagnosis mínima.

## 1. Introducción

La diagnosis basada en modelos permite determinar por qué un sistema diseñado correctamente no funciona conforme a las medidas captadas en el proceso de monitorización de las entradas y salidas. Se parte de un modelo explícito del sistema a diagnosticar, y a partir de él se razonan, aíslan e identifican los subsistemas que pueden generar fallos. La diagnosis permite tener un control sobre los fallos en componentes o procesos, lo que conlleva a los sistemas que la incorporan una mayor seguridad y reducción de costos.

La diagnosis es un campo muy activo de inves-

tigación. En este trabajo nos centraremos en la diagnosis basada en modelos. Existen dos comunidades que trabajan en paralelo y normalmente de forma aislada: la metodología FDI (Fault Detection and Isolation) proveniente del campo del control automático, y la metodología DX (acrónimo inglés de la palabra Diagnosis) emergida del campo de la Inteligencia Artificial. En los últimos años hay una apuesta por la integración de ambas metodologías. La integración de las teorías de FDI con DX y las pruebas de sus equivalencias se han mostrado para varios supuestos en [6] y [2].

En la metodología de localización y detección de fallos, FDI se usan relaciones dentro del modelo de comportamiento [14][13]. La metodología

permite un análisis fuera de línea (off-line) de parte del trabajo, frente a la metodología DX donde el trabajo es casi totalmente en línea (on-line). El trabajo fuera de línea se basa en el análisis estructural del modelo del sistema a diagnosticar. En los trabajos [4][18] se presenta la formalización del análisis estructural para la obtención de las relaciones de redundancia analítica de un sistema, denominadas ARRs (Analytical Redundancy Relation). Estas relaciones (obtenidas fuera de línea) permiten, aplicando los valores recibidos de los sensores (trabajo en línea), obtener cual es la diagnosis del sistema.

En la metodología DX el primer trabajo relevante fue [7], que presentaba un modelo para el diagnóstico de sistemas de componentes basado en la estructura y comportamiento de los componentes. La formalización de la diagnosis se concretó en [17] y [9], donde se propone una teoría general para el problema de explicar las discrepancias entre los comportamientos observados y los definidos como correctos para los componentes. Los modelos centrados en componentes describen los sistemas mediante relaciones entrada/salida. La mayoría de las aproximaciones para componentes caracterizan la diagnosis de un sistema como una colección de conjuntos mínimos de componentes que fallan, para explicar los comportamientos observados (síntomas). Este tipo de modelos permite diagnosticar rápidamente partes importantes de los sistemas de componentes. Las primeras implementaciones para diagnosis fueron DART[11] y GDE[10], que usaban diferentes componentes de inferencia para detectar los posibles fallos.

En este trabajo se presenta una integración de las teorías de las comunidades FDI y DX, que permite aumentar la eficiencia en la determinación de la diagnosis mínima de un sistema de componentes. La estructura que en este trabajo se seguirá es la siguiente. En primer lugar se establecen varias definiciones previas y se proponen dos ejemplos para demostrar los beneficios de la metodología. A continuación se explican las ventajas del tratamiento estructural previo, y la forma de generar el modelo usado en el proceso de diagnosis. Una vez obtenido el modelo se describe la forma de generar la diagnosis mínima a partir de las observaciones recogidas de los sensores. Para terminar se describen las conclusiones y el trabajo futuro.

## 2. Definiciones y notación

En este trabajo usaremos conceptos provenientes de las metodologías de diagnosis basada en modelos. Por tanto, se hace necesario establecer algunas definiciones previas.

**Definición 2.1** *Modelo del sistema*: Se define como el conjunto de ecuaciones que determinan el comportamiento de un sistema. Este objetivo se consigue por medio de las relaciones entre las variables asociadas a las restricciones. Estas variables podrán ser observables (están asociadas a un sensor) o no observables (sin un sensor asociado).

**Definición 2.2** *Modelo observacional*: Será una tupla de valores asociados a las variables observables. Estos valores serán obtenidos por los sensores colocados en el sistema.

**Definición 2.3** *Problema de diagnosis*: Se considera un problema de diagnosis a la tupla formada por el modelo del sistema y el modelo observacional. La solución al problema de diagnosis es el conjunto de posibles componentes que están fallando.

## 3. Ejemplos

Para clarificar la metodología propuesta, se han escogido los siguientes ejemplos, extraídos de la bibliografía asociada a las publicaciones relacionadas con la diagnosis basada en modelos:

- **Sistema de intercambiadores de calor**

Este sistema fue inicialmente propuesto en [12]. Consiste en seis intercambiadores de calor junto con tres flujos  $f_i$  de entrada que tienen asociados diferentes temperaturas  $t_i$ . En la figura 1 se muestra el sistema especificado, las zonas marcadas con un círculo sombreado indican que se ha colocado un sensor de flujo y temperatura en dicho punto. Es posible diferenciar tres subsistemas diferentes, cada uno formado por dos intercambiadores:  $\{E_1, E_2\}$ ,  $\{E_3, E_4\}$  y  $\{E_5, E_6\}$ . Cada uno de los seis intercambiadores, y cada uno de los ocho cruces ( $C_{ij}$ ) del sistema, son considerados como componentes en el proceso de diagnosis. El comportamiento del sistema puede ser descrito mediante las siguientes restricciones provenientes de los tres diferentes posibles tipos

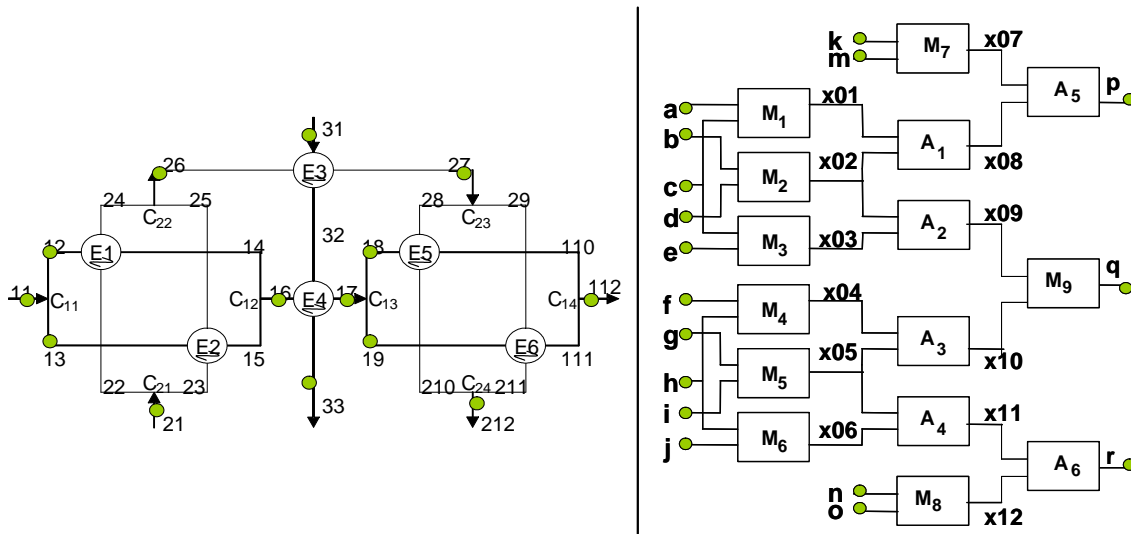


Figura 1: Ejemplos. Sistema de intercambiadores de calor, y el sistema polybox extendido

de balances:

$$\sum_i f_i = 0: \text{balance de masa en cada cruce}$$

$$\sum_i f_i \cdot t_i = 0: \text{balance térmico en cada cruce}$$

$$\sum_{in} f_i \cdot t_i - \sum_{out} f_j \cdot t_j = 0: \text{balance de entalpía para cada intercambiador de calor}$$

■ **Sistema polybox extendido**

Este sistema es una extensión del ejemplo más utilizado en la bibliografía relativa a la diagnosis basada en modelos [16][9]. En este caso aparecen quince componentes: nueve multiplicadores y seis sumadores, tal como se presenta en la figura 1. Los multiplicadores están representados como  $M_i$ , y los sumadores como  $A_j$ . Para detectar qué componentes fallan, sólo se conocen los valores de las variables observables que aparecen marcadas con un círculo sombreado sobre la figura.

**4. Pretratamiento estructural**

El pretratamiento estructural es la primera fase de la metodología propuesta en este trabajo. Como se explicará en esta sección, en esta fase de la metodología se consigue, en primer lugar, dividir el problema de diagnosis en subproblemas independientes, y, en segundo lugar, reducir el número de variables no-observables del sistema.

**4.1. Identificación de los clusters de componentes**

El primer paso será dividir el sistema en subconjuntos de componentes que puedan ser diagnosticados de forma independiente, y así reducir la complejidad de cómputo. La división en subsistemas permite la resolución en paralelo de estos problemas, y reduce el espacio de búsqueda.

**Definición 4.1** *Cluster de componentes (CC):* Un conjunto de componentes  $T$  es un cluster, si no existe ninguna variable no observable perteneciente a una restricción de un componente del cluster, que sea común con otra restricción de un componente que esté fuera del cluster de componentes. Además, para todo  $T' \subset T$ , se debe cumplir que  $T'$  no es un cluster de componentes.

Todas las variables no-observables comunes entre los componentes del mismo cluster pertenecen al cluster, por tanto, todas las conexiones con los componentes que están fuera del cluster deben estar monitorizadas. Por esta razón, un cluster de componentes está externamente monitorizado, y de ahí que la detección de fallos dentro del cluster sea posible sin la información de otros componentes que no pertenezcan al cluster. Cuanto más pequeños sean los clusters de componentes, mayor será la reducción computacional, ya que será posible dividir el problema en problemas más pequeños. Por este motivo, a través de la definición de cluster, se obliga que el cluster tenga el

menor número de componentes posible. En [5] se detalla el proceso para poder obtener el conjunto de clusters de un sistema de componentes.

**Ejemplo:** En el sistema de intercambiadores de calor se pueden obtener cinco clusters de componentes:  $\{C_{11}\}$ ,  $\{C_{13}\}$ ,  $\{C_{12}, C_{21}, C_{22}, E_1, E_2\}$ ,  $\{C_{14}, C_{23}, C_{24}, E_5, E_6\}$ , y  $\{E_3, E_4\}$ . Estos clusters permiten poder generar la diagnosis del sistema como combinación de las diagnosis obtenidas para cada cluster.

El número de diagnosis posibles para un sistema es inicialmente  $2^{nComp}$  [9], donde  $nComp$  es el número de componentes del sistema. Por tanto, la complejidad computacional del proceso de diagnosis es siempre menor para el sistema equivalente dividido en clusters, debido a la reducción del conjunto de posibles diagnosis.

En el sistema de intercambiadores de calor se reduce de  $2^{15} - 1$  a  $(2^1-1)+(2^1-1)+(2^5-1)+(2^5-1)+(2^2-1) = 67$  posibles diagnosis.

## 4.2. Obtención de relaciones sin variables no observables

En la metodología de diagnosis que proponemos se producirán nuevas relaciones sin ciertas variables. Estas relaciones permitirán, usando solamente un modelo observacional, supervisar el comportamiento y detectar los fallos de un sistema. Para ello se ha diseñado la función *NuevasRelaciones(NR)*.

**Definición 4.2** Función *NuevasRelaciones(NR)*: se define como una función que recibe dos argumentos, un conjunto de restricciones y un conjunto de variables; y obtiene, cuando es posible, un sistema de nuevas restricciones pero sin las variables incluidas en el conjunto que llegó como segundo argumento.

Por ejemplo  $NR(\{x-a \cdot c, y-b \cdot d, f-x-y\}, \{x, y\})$  genera la restricción  $\{a \cdot c + b \cdot d - f = 0\}$  donde han desaparecido las variables  $\{x\}$  e  $\{y\}$ . Esta función puede ser implementada usando diferentes técnicas. En este caso se han escogido las bases de Gröbner [3]. Las bases de Gröbner son el origen de muchos algoritmos de manipulación simbólica de restricciones polinómicas. Se trata de una combinación de la eliminación de Gauss y el algoritmo de Euclides.

Si tenemos un sistema de restricciones de igual-

dad  $P = 0$ , el algoritmo de las bases de Gröbner produce un sistema equivalente  $G = 0$ , con el mismo conjunto de soluciones que el original, pero sin ciertas variables. El algoritmo de las bases de Gröbner permite también eliminar ecuaciones redundantes. Las bases permiten trabajar sólo con restricciones de igualdad y polinómicas. Si las restricciones del sistema no permiten el uso de las bases de Gröbner, la implementación de la función *NuevasRelaciones* debería cambiar para incluir otras técnicas de cálculo simbólico. Por ejemplo, para poder utilizar restricciones de desigualdad se puede utilizar la descomposición cilíndrica [1], y encontrar así los valores de las variables en función de otras variables.

## 4.3. Obtención de los nodos de cada cluster

En este trabajo supondremos sistemas en los cuales cada componente tiene asociado una única restricción. Si fuera necesario aplicar esta metodología a componentes con  $n$  restricciones asociadas (donde  $n > 1$ ), se debería descomponer cada componente  $x$  en  $n$  componentes virtuales  $x_i$ , cada uno con una restricción asociada.

La aproximación que presentamos obtiene el menor número de conjuntos de restricciones que permiten detectar todos los posibles fallos de un sistema. Antes de seguir, resultan necesarias algunas definiciones para poder clarificar la forma en que se generan estos conjuntos de restricciones:

**Definición 4.3** *Variable prescindible*: Una variable no observable  $v_i$  es prescindible si existen sólo dos componentes  $x_i$  y  $x_j$  que incluyen esta variable en sus restricciones asociadas.

En el sistema polybox extendido de la figura 1 existen varias variables prescindibles, por ejemplo las variables  $x04$  y  $x08$  son prescindibles.

**Definición 4.4** *Nodo de componentes*: Un componente es un nodo de componentes si ninguna de sus variables no observables es prescindible. Dos componentes, o, un componente y un nodo de componentes, pertenecen a un mismo nodo de componentes si tienen en común al menos una variable prescindible.

El siguiente algoritmo (figura 2) permite obtener el conjunto de nodos asociado a un cluster.

**Algoritmo:** El algoritmo recoge  $T$ , el conjunto de

```

nodesIdentification(T) return N , S
Ev = Lista de componentes asociados a la variable v // Fusión de los nodos
Ni = Lista de componentes del nodo i while ∃ Ek ∈ E where |Ek| = 2
Si = Lista de variables prescindibles del nodo i ∧ Ek = {Ni,Nj} ∧ i<j
// Generación de todos los nodos iniciales Ni = Ni ∪ Nj
foreach x ∈ T N = N \ Nj
Ni = {x} Si = Si ∪ Sj ∪ {k}
endforeach foreach Eq ∈ E where Nj ∈ Eq
// Detección de los componentes asociados Eq = Eq \ Nj
// a una variable Eq = Eq \ Nj
foreach Ni ∈ N if ¬ ( Ni ∈ Eq )
foreach vark ⊂ nonObsVar(Ni) Eq = Eq ∪ Ni
foreach Ek = Ek ∪ {Ni} endif
endforeach endforeach
endforeach endwhile

```

Figura 2: Algoritmo para la detección de los nodos de un cluster de componentes

componentes de un cluster, y devuelve N, la lista de nodos, y S, la lista de variables prescindibles de cada nodo. Comienza creando  $n$  nodos, donde  $n$  es el número de componentes del cluster. Todos estos nodos comienzan teniendo un componente. Mas tarde, cada lista  $E_k$  almacenará todos los nodos que tienen una restricción que incluya la variable no observable  $k$ . La función auxiliar *nonObsVar* devuelve el conjunto de variables no observables de un conjunto de componentes.

Las variables prescindibles son detectadas cuando una de las listas  $E_k$  (asociada a la variable  $k$ ) incluye sólo dos nodos. Es este caso, los dos nodos,  $N_i$  y  $N_j$ , son fusionados en un único nodo  $N_i$ . Lógicamente las listas  $S_i$  y  $S_j$  serán también fusionadas. El resultado de la fusión será una lista  $S_i$  que incorporará las variables prescindibles del nodo, incluyendo la variable  $k$ . Después será necesario actualizar todos los elementos de cada lista  $E_q$ , para poder garantizar que el nodo  $N_i$  aparece en todos los elementos de la lista  $E_q$  en lugar del nodo eliminado  $N_j$ .

Aplicando la función *NuevasRelaciones* al conjunto de restricciones asociadas a cada nodo, se podrá obtener un nuevo conjunto de restricciones

sin las variables prescindibles. De esta forma será posible eliminar todas las variables prescindibles, y por tanto el sistema obtenido será menos complejo en cuanto al número de variables no observables.

Para obtener la diagnosis de un sistema, nuestra metodología necesita obtener conjuntos de nodos que contengan sistemas sobre-determinados de restricciones, que posibiliten la detección de fallos en los clusters. Antes de seguir avanzando vamos a definir el concepto de contexto.

**Definición 4.5 Contexto:** Es una colección de nodos del sistema de componentes. Las restricciones de un contexto se corresponden con la unión de las restricciones asociadas a cada uno de los nodos del contexto.

Las restricciones asociadas a estos contextos serán tratadas para obtener nuevas restricciones sin variables no observables que permitan detectar conflictos en un cluster. El número de posibles contextos, en el peor de los casos, es  $2^n - 1$ , donde  $n$  es el número de nodos del sistema. Los contextos se construyen basándose en los nodos de componentes en lugar de usar simplemente com-

Cuadro 1: Nodos para el sistema polybox extendido

Nodos	Componentes	Restricciones	Var. dispensables	Var. no obs.
N <sub>1</sub>	M <sub>6</sub> M <sub>8</sub> A <sub>4</sub> A <sub>6</sub>	h·j + n·o - r + x05	{x06, x11, x12}	{x05}
N <sub>2</sub>	M <sub>5</sub>	g·i - x05	{}	{x05}
N <sub>3</sub>	M <sub>1</sub> M <sub>7</sub> A <sub>1</sub> A <sub>5</sub>	a·c + k·m - p + x02	{x01, x07, x08}	{x02}
N <sub>4</sub>	M <sub>2</sub>	b·d - x02	{}	{x02}
N <sub>5</sub>	M <sub>3</sub> M <sub>4</sub> M <sub>9</sub> A <sub>2</sub> A <sub>3</sub>	q - (f·h + x05)·(x02 + c·e)	{x03, x04, x09, x10}	{x02, x05}

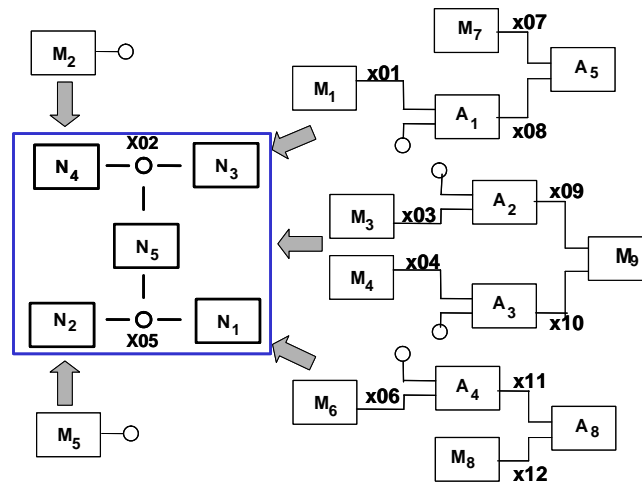


Figura 3: Nodos del sistema polybox extendido

ponentes, ya que es imposible generar nuevas restricciones usando solamente un subconjunto de componentes pertenecientes a un nodo, ya que será imposible sustituir alguna de las variables prescindibles del nodo de componentes.

Si un conjunto  $Q$  no contiene todos los componentes de un nodo  $N$ , entonces existirá una variable prescindible  $v$  que no tendrá pareja dentro del conjunto  $Q$ , ya que sólo aparecerá en un componente del conjunto  $Q$ . Las restricciones del conjunto  $Q$  que provienen del nodo  $N$  no permiten generar nuevas restricciones sin variables no observables. Por tanto, no será posible generar nuevas restricciones usando todas las restricciones del conjunto  $Q$ , ya que parte de los componentes que forma el nodo  $N$ , no pertenecen al conjunto  $Q$ . Por tanto, por razones de eficiencia, los contextos se deben generar a partir de nodos y no de componentes.

**Ejemplos:** La figura 3 muestra la división del polybox extendido en nodos. La tabla 1 muestra la lista de nodos del sistema polybox extendido. El número de variables no observables se ha reducido de 12 a 2. En la tabla 2 aparecen los resultados que se han obtenido para los dos ejemplos propuestos. En dicha tabla, las diferentes colum-

nas tienen el siguiente significado. La columna *Nodos* muestra la suma de todos los nodos incluidos en el cluster del sistema. La columna *Vars* muestra el número inicial y final (después del pretratamiento estructural) de variables no observables. La columna *Ctxs* muestra el número total de posibles contextos para el sistema con o sin análisis estructural. La columna *Tiempo* muestra el tiempo necesario para procesar los posibles contextos si se supone que el tiempo para procesar cada contexto es de 1 milisegundo.

### 5. Determinación de la red de contextos

En la comunidad DX, la diagnosis del sistema se obtiene a través de la detección de conflictos en el modelo especificado. Los métodos de pre-compilación usan la descripción estructural del sistema para detectar conflictos fuera de línea. Por ejemplo, el trabajo [15] propone el concepto de los Posibles Conflictos como técnica de pre-compilación. Cada Posible Conflicto representa un subsistema de restricciones capaz de almacenar la mínima información para obtener un con-

Cuadro 2: Mejoras obtenidas usando el pretratamiento estructural en los ejemplos

Ejemplo	Clusters	Nodos	Sin pretratamiento			Con pretratamiento		
			Vars.	Ctxs.	Tiempo	Vars.	Ctxs.	Tiempo
I. Calor	5	14	26	$2^{14}-1$	16 segundos	0	14	14 milisegundos
Polybox	1	5	12	$2^{15}-1$	32 segundos	2	31	31 milisegundos

Cuadro 3: CARCs obtenidas del sistema polybox extendido

Index	Context	CARC
1	$N_1N_2$	$h \cdot j + n \cdot o - r + g \cdot i$
2	$N_3N_4$	$a \cdot c + k \cdot m - p + b \cdot d$
3	$N_1N_3N_5$	$q - (f \cdot h - h \cdot j - n \cdot o + r) \cdot (-a \cdot c - k \cdot m + p + c \cdot e)$
4	$N_1N_4N_5$	$q - (f \cdot h - h \cdot j - n \cdot o + r) \cdot (b \cdot d + c \cdot e)$
5	$N_2N_3N_5$	$q - (f \cdot h + g \cdot i) \cdot (-a \cdot c - k \cdot m + p + c \cdot e)$
6	$N_2N_4N_5$	$q - (f \cdot h + g \cdot i) \cdot (b \cdot d + c \cdot e)$

flicto en el modelo del sistema. En la metodología de FDI, la redundancia analítica a través de las relaciones (ARRs)[18], se puede considerar también como una técnica de pre-compilación.

La metodología que proponemos permite obtener el conjunto mínimo de contextos necesarios para detectar un fallo en un cluster de componentes. El objetivo de minimalidad no está garantizado en la obtención de las ARRAs cuando hablamos de la metodología FDI, pero sí está garantizado en este trabajo para la obtención de las restricciones tipo CARC. En el trabajo [15] los Posibles Conflictos son obtenidos directamente usando grupos de componentes, en nuestra aproximación se usarán nodos en lugar de componentes, por tanto el número de posibles sistemas sobre-determinados se reduce de  $2^c$ , donde  $c$  es el número de componentes, a  $2^n$ , donde  $n$  es el número de nodos.

El siguiente paso en la metodología propuesta es la generación de una red de contextos, de acuerdo a la forma ATMS[8]. Esta red permitirá obtener todos los contextos relevantes en el proceso de diagnóstico. Antes de seguir, son necesarias algunas definiciones:

**Definición 5.1** *Red de contextos*: Será un grafo dirigido formado por todos los contextos de un sistema mediante la forma ATMS propuesta en [8].

**Definición 5.2** *Contexto estructural*: Será un contexto donde todos los nodos forman un único grafo conexo. La función que se usará para saber si un contexto es estructural se llama *isStructural*, toma un contexto  $C$  y devuelve el valor verdadero si es un contexto estructural.

**Definición 5.3** *Contexto completo minimal*: Un contexto estructural  $C$  es un contexto completo minimal si el conjunto de restricciones de los nodos que forman el contexto forman un sistema sobre-determinado, y, si es posible generar nuevas restricciones sin variables no observables usando

el conjunto de restricciones del contexto. Un contexto completo es minimal si no existe ningún contexto  $C' \subset C$  tal que  $C'$  sea un contexto completo.

Sea  $C$  un contexto completo minimal y sea  $C'$  un contexto tal que  $C \subset C'$ , todas las restricciones sin variables no observables que se pueden obtener con  $C'$  ya habrán sido generadas usando un contexto que contenga menos nodos que  $C'$ . Por tanto, si un contexto  $C$  es completo y minimal, no será necesario procesar los contextos  $C'$  tales que  $C \subset C'$ , ya que no es posible generar nuevas restricciones relevantes para el proceso de diagnóstico.

El algoritmo que genera los contextos de cada cluster tiene  $n$  etapas: primero los contextos con sólo un componentes son procesados, luego los contextos con 2 nodos, y así hasta llegar a los contextos con  $n$  nodos. La función *NuevasRelaciones* se usa sólo en aquellos contextos que resultan ser contextos estructurales. Cuando un contexto completo y minimal es detectado, las nuevas restricciones sin variables no observables se almacenan, y ningún contexto  $C'$ , tal que  $C \subset C'$ , será procesado. Estas nuevas restricciones se llamarán restricciones de contextos con redundancia analítica (Context Analytical Redundancy Constraint).

**Definición 5.4** *Restricción de contexto con redundancia analítica (CARC)*: Será una restricción obtenida de un contexto completo minimal de tal forma que sólo aparezcan variables observables.

**Ejemplo**: En la tabla 3 se muestran las CARCs obtenidas para sistema polybox extendido. El sistema incluye sólo un cluster de 15 componentes. Usando el pretratamiento estructural el número de posibles contextos se reduce de  $2^{15}-1$  a  $2^5-1$ , ya que como aparece en la tabla 1 sólo existen 5 nodos. Aplicando las reglas propuestas, sólo 14 contextos de los 31 ( $2^5-1$ ) posibles son generados,

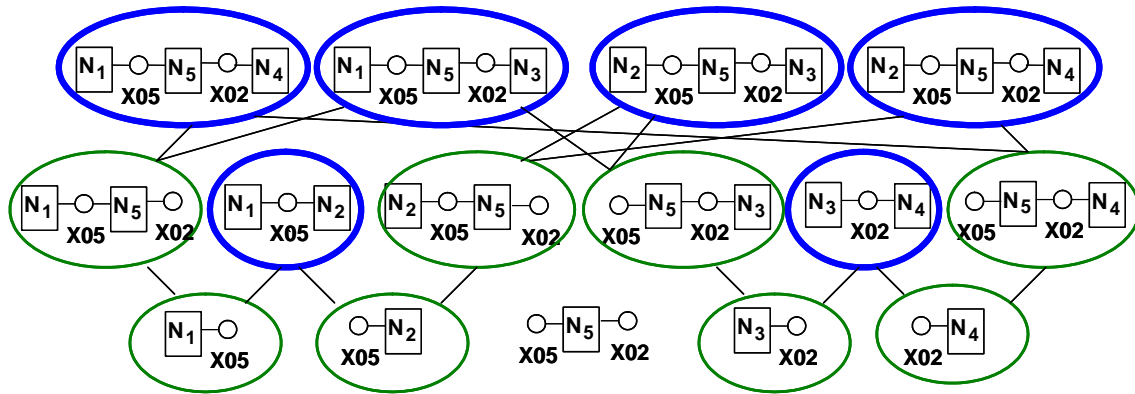


Figura 4: Red de contextos del ejemplo polybox extendido

y sólo 6 son contextos completos y mínimos. En la figura 4 se muestran los contextos tratados del total de la red de contextos. Los contextos completos y mínimos aparecen marcados con una elipse en negra.

## 6. Determinación de la diagnosis mínima

El último paso será determinar la diagnosis mínima del sistema usando el conjunto de restricciones de tipo CARC. Para clarificar la metodología, supondremos que las observaciones de los sensores son correctas. De forma muy parecida a como se realiza en la metodología FDI, proponemos usar una matriz de firmas.

**Definición 6.1** *Firma de fallo:* Dado un conjunto de  $n$  relaciones CARCs,  $CARC = \{CARC_1, CARC_2, \dots, CARC_n\}$ , y un conjunto de  $m$  fallos  $F = \{F_1, \dots, F_m\}$ , la firma de fallo  $F_j$  viene dada por  $FS_j = [s_{1j}, \dots, s_{nj}]^T$  en la cual  $s_{ij} = 1$  si el contexto que genera la restricción  $CARC_i$  contiene los nodos incluidos en el fallo  $F_j$ , y  $s_{ij} = 0$  en otro caso.

**Definición 6.2** *Matriz de firmas:* Todas las firmas para el conjunto de posibles fallos constituye lo que llamaremos matriz de firmas.

**Definición 6.3** *Firma de un modelo observacional:* Vendrá dada por  $OS = [OS_1, \dots, OS_n]$  donde  $OS_i = 0$  si cada restricción  $CARC_i$  es satisfecha, y  $OS_i = 1$  en otro caso.

**Definición 6.4** *Conjunto de diagnosis:* Será el conjunto de fallos cuyas firmas son consistentes

con la firma del modelo observacional. En nuestra metodología se supone que una firma de un modelo observacional OS es consistente con otra firma  $FS_j$  si  $OS_i = s_{ij}$  para todo  $i$ .

**Definición 6.5** *Diagnosis mínima:* Un fallo  $F_j$  es una diagnosis mínima si para todo  $F_k \subset F_j$ ,  $F_k$  no es una diagnosis.

La tabla 4 muestra la matriz de firmas para el ejemplo del polybox extendido. La firma  $OK = [0, \dots, 0]^T$  representa el caso de no fallo. La matriz de firmas generada es muy similar a la que se obtendría al aplicar la metodología FDI, pero en nuestro caso, los fallos implican nodos en lugar de componentes.

En el ejemplo presentado resulta necesario ampliar el número de columnas de la matriz de firmas para poder almacenar las firmas de los fallos múltiples. Cada fallo  $F_j$ , que incluye  $n$  nodos, se obtendrá usando el fallo  $F_k$ , que incluye  $n-1$  nodos, y un fallo simple  $F_s$  que no está incluido en el fallo  $F_k$ . Las firmas de fallos múltiples  $F_j$  vendrán dadas por  $FS_j = [s_{1j}, \dots, s_{nj}]^T$  en donde  $s_{ij} = 0$  si  $s_{ik} = s_{is} = 0$ , y  $s_{ij} = 1$  en otro caso. La firma de un fallo múltiple  $F_j$  no será añadida a la matriz de fallo si  $\forall s_{ij} : s_{ij} = 1 \rightarrow s_{ij} = s_{ik}$ , ya que ello implicaría que ese nuevo fallo múltiple es un superconjunto de un fallo que incluye menos nodos, y por tanto, aunque es una diagnosis, no puede ser una diagnosis mínima. La generación de la matriz de firmas parará cuando resulte imposible generar nuevas firmas de fallos que incluyan  $n$  nodos basándose en los fallos que incluyen  $n-1$  nodos.

La metodología FDI lleva implícito el principio de exoneración [6], es decir, dado un modelo observacional, cada componente que forma parte del



Cuadro 4: Matriz de firmas del sistema polybox extendido

CARC	OK	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>12</sub>	F <sub>13</sub>	F <sub>14</sub>	F <sub>15</sub>	F <sub>23</sub>	F <sub>24</sub>	F <sub>25</sub>	F <sub>34</sub>	F <sub>35</sub>	F <sub>45</sub>	F <sub>xxx</sub>
1	0	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	1
2	0	0	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1
3	0	1	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1
4	0	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1
5	0	0	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1
6	0	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1

$$F_{xxx} \subset \{F_{123}, F_{124}, F_{134}, F_{135}, F_{145}, F_{234}, F_{235}, F_{245}\}$$

	OK	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>12</sub>	F <sub>13</sub>	F <sub>14</sub>	F <sub>15</sub>	F <sub>23</sub>	F <sub>24</sub>	F <sub>25</sub>	F <sub>34</sub>	F <sub>35</sub>	F <sub>45</sub>	F <sub>xxx</sub>
VC	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	0	0
MD	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0

Valores de VC y MD para la firma de fallo OS = [0, 0, 1, 1, 1, 1]<sup>T</sup>

soporte de una restricción CARC satisfecha se considera que funciona correctamente, es decir, se exonera. En la metodología DX, por defecto el principio de exoneración no se utiliza, sin embargo en la metodología FDI viene implícito.

En la metodología que proponemos queremos tener la posibilidad de aplicar el principio de exoneración o no, es decir, queremos que ambas posibilidades puedan satisfacerse. Para obtener soluciones sin que el principio de exoneración de cumpla es necesario aplicar una nueva definición de consistencia. En el caso de no aplicar el principio de exoneración, una firma de un modelo observacional es consistente con una firma FS<sub>j</sub> si para todo OS<sub>i</sub> = 1 se cumple que s<sub>ij</sub> = 1. Es decir, sólo se usarán las restricciones tipo CARC que no puedan ser satisfechas, y además se debe cumplir que F<sub>j</sub> contiene el valor 1 en cada restricción tipo CARC no satisfecha.

Con esta nueva definición de consistencia de firmas, queremos además obtener de forma automática sólo las diagnosis que son mínimas diagnosis. Para ello son necesarias una serie de reglas para detectar cuáles de los fallos son diagnosis mínimas, ya que muchos fallos serán consistentes con el modelo observacional aunque no sean diagnosis mínimas. El siguiente algoritmo permite generar estas reglas.

**Algoritmo:** Sea CS(OS,FS) una función que permite evaluar si la firma OS es consistente con la firma FS. Para cada posible fallo F<sub>j</sub> de la matriz de firmas, sea MD<sub>F<sub>j</sub></sub> una variable booleana que almacena si el fallo F<sub>j</sub> es una diagnosis mínima, y sea VC<sub>F<sub>j</sub></sub> una variable booleana que almacena si un fallo F<sub>j</sub> es un candidato válido para la gen-

eración de nuevos fallos que puedan llegar a ser diagnosis mínimas. Para cada posible fallo F<sub>j</sub> se supondrá inicialmente que VC<sub>F<sub>j</sub></sub> = verdadero.

El primer paso consiste en validar si OK (caso de no fallo) puede ser la diagnosis mínima. OK es una diagnosis mínima si MD<sub>OK</sub> = CS(OS,OKS). Para todo fallo simple F<sub>j</sub>, la igualdad VC<sub>F<sub>j</sub></sub> = ¬ MD<sub>OK</sub> debe ser satisfecha, para garantizar que no existen más diagnosis mínimas.

Si OK no es una diagnosis mínima, las siguientes reglas deben ser evaluadas para todo fallo posible (excepto OK) en el mismo orden secuencial en el que aparecen en la matriz de firmas. Estas reglas garantizan la detección correcta de la diagnosis mínima para un modelo observacional:

- Para cada fallo F<sub>j</sub> con la firma FS<sub>j</sub>, la igualdad MD<sub>F<sub>j</sub></sub> = C<sub>F<sub>j</sub></sub> ∧ CS(OS,FS<sub>j</sub>) debe satisfacerse.
- Para cada fallo F<sub>k</sub> que incluye n + 1 nodos, donde n ≥ 0, y que puede ser obtenido usando un fallo F<sub>j</sub> (que incluya n nodos) y un fallo simple F<sub>s</sub> (que no este incluido en F<sub>j</sub>) entonces MD<sub>F<sub>j</sub></sub> ⇒ VC<sub>F<sub>k</sub></sub> = falso.

**Ejemplo:** La tabla 5 muestra un resumen de las reglas generadas para el ejemplo del sistema polybox extendido. La generación de las reglas para verificar si un fallo es una diagnosis mínima, o no, puede hacerse fuera de línea, ya que estas reglas son siempre las mismas para cualquier modelo observacional. La parte baja de la tabla 4 muestra los resultados asignados a las variables VC y MD tras evaluar las reglas para la firma del modelo observacional OS = [0, 0, 1, 1, 1, 1]<sup>T</sup>. Sólo

Cuadro 5: Resumen de las reglas generadas para el sistema polybox extendido

$MD_{OK} = CS(OS,OKS)$	$MD_{F1} \Rightarrow VC_{F14} = \text{falso}$	$MD_{F13} \Rightarrow VC_{F123} = \text{falso}$
$VC_{F1} = \neg MD_{OK}$	...	$MD_{F13} \Rightarrow VC_{F134} = \text{falso}$
...	$MD_{F1} \Rightarrow VC_{F15} = \text{falso}$	$MD_{F13} \Rightarrow VC_{F135} = \text{falso}$
$VC_{F5} = \neg MD_{OK}$	$MD_{F2} = VC_{F2} \wedge CS(OS,FS_2)$	$MD_{F14} = VC_{F14} \wedge CS(OS,FS_{14})$
$MD_{F1} = VC_{F1} \wedge CS(OS,FS_1)$	...	...
$MD_{F1} \Rightarrow VC_{F12} = \text{falso}$	$MD_{F13} = VC_{F13} \wedge CS(OS,FS_{13})$	$MD_{F245} = VC_{F245} \wedge CS(OS,FS_{245})$

la evaluación de las reglas debe hacerse en línea. La evaluación del modelo observacional es simplemente una propagación de valores booleanos.

La evaluación de la matriz de fallos es muy similar a la forma propuesta en la metodología FDI. Sin embargo en nuestra propuesta, los fallos incluyen nodos en lugar de componentes. Por tanto, el último paso será la sustitución de cada nodo por alguno de sus componentes. En el ejemplo del sistema polybox extendido, el fallo  $F_3$  (fallo en el nodo 3) sería equivalente a los fallos en los componentes  $\{\{M_1\}, \{M_7\}, \{A_1\}, \{A_5\}\}$ ; el fallo  $F_{12}$  sería equivalente a los fallos  $\{\{M_6M_5\}, \{M_8M_5\}, \{A_4M_5\}, \{A_6M_5\}\}$ ; y así con todos las diagnosis mínimas.

La información de todas las posibles diagnosis mínimas se encuentran almacenadas en esta matriz y el conjunto de reglas asociadas. Por tanto, sólo es necesario calcular esta matriz y las reglas asociadas una vez. Tal como sucede en la metodología FDI, este trabajo se puede realizar fuera de línea, sólo la evaluación de la matriz de firmas y las reglas asociadas se debe hacer en línea. Nuestra aproximación genera de forma automática sólo las diagnosis que son mínimas para un modelo observacional concreto, pero en la metodología FDI esto no esta garantizado, ya que sólo se usa la matriz de firmas para generar las posibles diagnosis.

El coste del cálculo de la matriz de firmas es exponencial con respecto al número de componentes en el caso de FDI, y en nuestra metodología es exponencial con respecto al número de nodos. El calculo de la matriz de firmas para todo tipo de fallos (simples, dobles...) es complejo, y a veces incluso inviable. En estos casos es necesario ceñirse a fallos simples o dobles para que el mantenimiento de la matriz de firmas sea viable. Aunque este problema existe, hay que tener en cuenta que el uso de los nodos en la generación de la matriz de firmas, hace que la metodología que proponemos sea mas eficiente y sea aplicable a más tipos de problemas que la metodología FDI, ya que el nu-

mero de nodos siempre será menor o igual al número de componentes. La metodología DX no requiere el almacenaje de la matriz de firmas, pero debe realizar prácticamente todo el proceso en línea. Además, aun recogiendo los conflictos fuera de línea, es necesario aplicar técnicas de *hitting set*, que son de orden exponencial, para detectar la diagnosis mínima.

## 7. Conclusiones y trabajos futuros

En este trabajo se ha propuesto una metodología para la automatización y mejora del proceso de determinación de la diagnosis mínima de un sistema. Esta aproximación está basada en técnicas provenientes de las teorías de las comunidades FDI y DX. El pretratamiento estructural en una primera fase, y la reducción del modelo en una segunda fase, permiten mejorar la eficiencia y reducir la necesidad de cálculos del proceso de diagnosis. Todas las posibles mínimas diagnosis son representadas en una matriz de firma de fallos y un conjunto de reglas asociadas. La diagnosis mínima se obtiene aplicando un modelo observacional sobre la matriz de firma de fallos y las reglas asociadas. Sólo la evaluación de la matriz de firmas y las reglas asociadas se realiza en línea, el resto del proceso se realiza fuera de línea.

La metodología propuesta se ha aplicado a ejemplos usuales en la diagnosis basada en modelos, y en ellos se ha conseguido mejorar la eficiencia del calculo de la diagnosis mínima. Como trabajos futuros queremos extender la metodología a sistemas dinámicos y tratar problemas reales donde la aplicación de la metodología sea más compleja.

## Agradecimientos

Este trabajo ha sido financiado por el M. de Ciencia y Tecnología (DPI2003-07146-C02-01).

## Referencias

- [1] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition i: the basic algorithm. *SIAM J. Comput.*, 13(4):865–877, 1984.
- [2] G. Biswas, M.-O. Cordier, J. Lunze, L. Travé-Massuyés, and M. Staroswiecki. Diagnosis of complex systems: Bridging the methodologies of the FDI and DX communities. *IEEE Transactions on Systems, Man and Cybernetics*, 34(5):2159–2162, 2004.
- [3] B. Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. *Multidimensional Systems Theory*, N. K. Bose, ed., pages 184–232, 1985.
- [4] J. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *IFAC-IFIP-IMACS Conf. on Control of Industrial Processes, Belfort, France, 1997*.
- [5] R. Ceballos, M. T. Gómez-López, R.M. Gasca, and S. Pozo. Determination of possible minimal conflict sets using components clusters and Grobner Bases. In *DX04, 15th International Workshop on Principles of Diagnosis*, pages 21–26, Carcassonne, France, June 2004.
- [6] M. Cordier, F. Lévy, J. Montmain, L. Travé-Massuyés, M. Dumas, M. Staroswiecki, and P. Dague. A comparative analysis of AI and control theory approaches to model-based diagnosis. In *14th European Conference on Artificial Intelligence*, pages 136–140, 2000.
- [7] R. Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410, 1984.
- [8] J. de Kleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 2(28):127–161, 1986.
- [9] J. de Kleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 2-3(56):197–222, 1992.
- [10] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.
- [11] M. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24:411–436, 1984.
- [12] C. Guernez. Fault detection and isolation on non linear polynomial systems. In *15th IMACS World Congress on Scientific, Computation, Modelling and Applied Mathematics, 1997*.
- [13] R. Isermann. Supervision, fault-detection and fault-diagnosis methods, an introduction. *Control Engineering Practice*, 5(5):639–652, 1997.
- [14] R. J. Patton and J. Chen. A review of parity space approaches to fault diagnosis. *IFAC-SAFEPROCESS Symposium*, 1991.
- [15] B. Pulido and C. Alonso González. Possible conflicts: A compilation technique for consistency-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(5):2192–2206, October 2004.
- [16] R. Reiter. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence 24*, pages 347–410, 1984.
- [17] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence 32*, 1:57–96, 1987.
- [18] M. Staroswiecki and P. Declerk. Analytical redundancy in non linear interconnected systems by means of structural analysis. In *IFAC Advanced Information Processing in Automatic Control (AIPAC-89)*, pages 51–55, Nancy, France, June 1989.