



Computación y Sistemas

ISSN: 1405-5546

computacion-y-sistemas@cic.ipn.mx

Instituto Politécnico Nacional

México

Abad, Pedro J.; Suárez, Antonio J.; Gasca, Rafael M.; Ortega, J. A.  
Diagnosis de Sistemas Dinámicos Basada en Aprendizaje Supervisado Off-Line  
Computación y Sistemas, vol. 5, núm. 3, enero-marzo, 2002, pp. 180-191  
Instituto Politécnico Nacional  
Distrito Federal, México

Available in: <http://www.redalyc.org/articulo.oa?id=61550303>

- ▶ How to cite
- ▶ Complete issue
- ▶ More information about this article
- ▶ Journal's homepage in redalyc.org

# Diagnosis de Sistemas Dinámicos Basada en Aprendizaje Supervisado Off-Line

*Diagnosis of Dynamic Systems Using Supervised Learning Off-Line*

Pedro J. Abad<sup>1</sup>, Antonio J. Suárez<sup>1</sup>, Rafael M. Gasca<sup>2</sup> y J. A. Ortega<sup>2</sup>

<sup>1</sup>Departamento de Ing. Electrónica, Sist. Informáticos y Automática, Universidad de Huelva  
Ctra Huelva-Palos de la Frontera s/n. Huelva, España

<sup>2</sup>Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, España  
e-mail : {abadhe, asuarez}@uhu.es, {gasca, ortega}@lsi.us.es

*Artículo recibido en mayo 30, 2001; aceptado en septiembre 25, 2001*

## Resumen

*Este trabajo presenta una aproximación para la diagnosis de sistemas dinámicos basándose en reglas de decisión generadas mediante aprendizaje supervisado. Para comenzar el proceso se deberá disponer, bien de la evolución experimental del sistema, o bien de la colección de simulaciones que muestren como evoluciona el sistema ante el fallo de sus distintos componentes. La información de dichas simulaciones será enriquecida antes de proceder a procesarlas mediante una herramienta de clasificación, que nos proporcionará el conjunto de reglas de decisión sobre el comportamiento del sistema ante las distintas situaciones con las que se ha probado. Todo el trabajo de la obtención de las reglas de decisión será realizada previamente a la diagnosis. El conjunto de reglas obtenido será el que nos permita discernir, durante el funcionamiento del sistema, si existe algún fallo, y en qué componente se localiza, mediante la asociación del comportamiento observado con el proporcionado por las reglas.*

**Palabras Clave:** Diagnosis, Aprendizaje, Clasificación, Reglas de decisión, Detección de Fallos.

## Abstract

*This paper presents an approach to diagnose dynamic systems using decision rules generated by supervised learning. The methodology can start with real system data or with a set of simulations of the dynamic system model, both of them showing the system evolution when different components are faulty. Before processing the obtained information with a classification tool, it will be treated. After that, classification tool will provide us the decision rules corresponding with the behaviour of the system under trained situations. Tasks to obtain the decision rules are made before making diagnosis. When system is running, diagnosis can be made by associating the observed behaviour with the obtained decision rules. This way it is possible to detect when a fault occurs and the corresponding faulty component.*

**Keywords:** Diagnosis, Learning, Classification, Decision Rules, Fault Detection.

## 1 Introducción

La tarea de diagnosis determina por qué un sistema diseñado correctamente no está funcionando como se esperaba. La explicación para dicho comportamiento erróneo representa una discrepancia en alguna forma con el diseño. Una de las subtareas de la diagnosis es determinar los elementos del sistema que podrían causar el erróneo comportamiento de acuerdo con unas observaciones realizadas. Por ello la monitorización es un proceso clave, que debe evitar que la detección de pequeñas alteraciones en los valores de las variables, debido generalmente a la dinámica de los sistemas, induzca fallos no reales.

La detección de fallos consiste en determinar, a partir de las observaciones realizadas, cuando existe un funcionamiento incorrecto del sistema sujeto a observación. Una vez detectado el fallo la diagnosis se encargará de encontrar cuales son las causas de ese comportamiento incorrecto.

La detección y la diagnosis del funcionamiento anómalo de mecanismos es importante desde el punto de vista estratégico de las empresas, debido a las demandas económicas y de conservación del medio ambiente que se requieren para permanecer en los mercados competitivos. En parte esto conduce a que sea un campo de investigación muy activo. Los fallos producidos en los componentes y procesos pueden provocar paradas indeseables y deterioro de los sistemas, con el consiguiente aumento de costos y la disminución de la producción. Por tanto para mantener los sistemas en niveles de seguridad, producción y fiabilidad deseados se necesita desarrollar mecanismos que permitan la detección y diagnosis de esos fallos que se producen en los sistemas.

En este entorno de trabajo, la monitorización debe representar un fiel reflejo del sistema y de las desviaciones que se producen del comportamiento esperado. La diagnosis permite identificar las partes que fallan. Generalmente los sistemas de diagnosis integran la monitorización y la diagnosis (detección e identificación de fallos).

El concepto de Diagnóstico es muy amplio y cubre prácticamente todos los aspectos de la vida cotidiana: medicina, dispositivos electrónicos, economía, ecosistemas, etc.

Nosotros nos vamos a centrar en el diagnóstico automático, entendiendo como tal la utilización de los ordenadores para reducir o eliminar la intervención humana en el proceso.

Desde hace más de una década se ha manifestado un interés creciente en la comunidad científica por la diagnosis de sistemas dinámicos. Dos comunidades científicas diferentes han desarrollado dos aproximaciones diferentes. Por una parte la comunidad bajo el nombre de *Identificación y Detección de Fallos(FDI)* que utiliza técnicas procedentes de la teoría de control y análisis estadístico. De entre ellas pueden destacarse la estimación de estados (Frank, 87), de parámetros (Iserman, 87) y la técnica de redundancia analítica (Gertler, 91). En los últimos años han aparecido diferentes monografías sobre el tema (Frank, 96), (Iserman, 97) y (Patton, 00 et al., 00). En esta última cita se reflejan algunos de los trabajos desarrollados por la otra comunidad científica, que surgió más recientemente y que se basa principalmente en técnicas derivadas de la Inteligencia Artificial(DX).

Dentro de esta comunidad y principalmente en la década de los 80 se produce una evolución de los sistemas expertos que hasta entonces llevaban a cabo la diagnosis de los sistemas a una nueva aproximación llamada diagnosis basada en modelos(DBM), (Reiter,87), (De Kleer, et al.,92). El motor de diagnosis general, GDE (De Kleer & Willians, 87) determinó un marco de trabajo para la diagnosis, que subsecuentemente influenciará casi todos los trabajos en el campo de la diagnosis basada en modelos. El dominio de aplicación principal de este tipo de diagnosis era sobre sistemas "estáticos", esto es, sistemas que no evolucionaban en el tiempo.

Los primeros trabajos que aparecen sobre sistemas que evolucionan en el tiempo extienden los algoritmos de los sistemas estáticos (GDE) y considerando QSIM [Kuipers86] como herramienta para detectar los conflictos en estos sistemas(Ng, 90), (Ng,91). En trabajos posteriores se ha seguido utilizando QSIM (Dvorak&Kuipers,92) y se ha mostrado (Dressler&Freitag, 94) como ATMS sirve como un elemento para tratar computacionalmente etiquetas temporales y permite integrarlas con las etiquetas lógicas. También se ha utilizado el conocimiento de las consecuencias de los diferentes modos de comportamiento de los componentes y un modelo de evolución de tales modos (Console et al., 94). Una caracterización general para la diagnosis basada en modelos teniendo en cuenta el tiempo puede encontrarse en (Brusconi et al. , 98). Otras aproximaciones han usado los grafos causales para llevar a cabo la diagnosis (Bousson&Trave-Massuyes, 92) y los grafos causales temporales (Mosterman,97). Para una revisión de las técnicas desarrolladas en el área de la Inteligencia Artificial puede consultarse (Price,99). Las tendencias sobre la diagnosis basada en modelos y los retos en el futuro se recogen en un trabajo reciente (Console&Dressler, 99).

Dos de las discusiones más significativas que se han observado en los últimos años son:

- En primer lugar sobre si es suficiente para llevar a cabo la diagnosis de los sistemas dinámicos considerar únicamente los estados del sistema e ignorar lo que sucede entre ellos (Malik & Struss,96), (Dressler, 96). En (Struss,97) se han presentado las condiciones generales de equivalencia de la diagnosis basada en estado y en la simulación:

\*Un comportamiento continuo del sistema

\*Las restricciones temporales son las mismas para todos los estados

Un trabajo posterior (Panati & Drupé, 00) discute como los resultados de la simulación basada en estados en contextos diferentes no produce los mismos resultados y muestra como se puede razonar a partir de la causalidad del sistema y usando la simulación.

- En segundo lugar es el uso de los modelos cuantitativos, que tiene una serie de inconvenientes pues cuando son usados existe una complejidad del modelado y simulación, además que si hay imprecisión o incertidumbre se necesita redefinir el concepto de consistencia. Pero sin embargo estos modelos pueden evitar el problema del bucle realimentado (Dressler & Struss, 94). En esta línea un reciente trabajo usa estos modelos y registros de dependencias off-line (Pulido et al., 01)

El presente trabajo se centra en modelos cuantitativos y usa las técnicas de aprendizaje supervisado para obtener un modelo basado en reglas que permita mejorar la eficiencia de la diagnosis de modelos dinámicos. Se han presentado últimamente algunos trabajos sobre diagnosis usando técnicas basadas en aprendizaje como son métodos estocásticos (Poulezos&Stavrakakis, 94), técnicas de tipo conexiónista (Venkatusubramanian&Chan, 95) y sistemas de clasificación (Leonhart & Ayoubi , 97). Nuestra aproximación usa aprendizaje supervisado para las trayectorias de los modelos con comportamiento correcto y modelos de fallos, e induce un árbol de decisión (Quinlan, 86), mediante la herramienta C4.5 (Quinlan, 93), que se utiliza para realizar la diagnosis del sistema dinámico, que sujeto a observación, se comporta adecuadamente o se adapta a cualquiera de las reglas de fallos obtenidas. Las implementaciones de esta aproximación muestran una notable eficiencia y el grado de acierto es bastante alto, sobre todo en fallos que están suficientemente alejados del comportamiento correcto.

El artículo ha sido organizado de la siguiente manera: en la siguiente sección se describe un problema ejemplo de aplicación para la aproximación desarrollada. A continuación expondremos la metodología empleada y la forma de realizar la diagnosis. Para ilustrar el funcionamiento de estas técnicas se presenta un amplio juego de pruebas desarrolladas, que nos permiten comprobar los resultados ofrecidos por la aproximación empleada. Por último se discuten algunas mejoras que están en proceso de desarrollo.

## 2 Descripción del Problema

Como problema ejemplo de sistema dinámico cuyo comportamiento debe ser diagnosticado se ha elegido el presentado en (Panati & Drupé, 00), que es una modificación del presentado en un trabajo previo (Malik & Struss, 96).

El modelo del sistema dinámico considerado se representa en la Figura 1. Dicho sistema consta de un Motor Eléctrico (M),

que estará girando a una determinada velocidad angular ( $w$ ). La velocidad de giro del motor ( $w$ ) será recogida por un Sensor (S) que proporcionará la velocidad angular medida ( $w_m$ ) a un controlador (C). El controlador (C) actuará proporcionando un voltaje (V) al motor (M) para adecuar la velocidad de giro del mismo a la velocidad angular deseada ( $d$ ) en función de la velocidad angular medida ( $w_m$ ).

En (Malik & Struss, 96) se considera un controlador proporcional (P) para el modelo, mientras que en el trabajo de (Panati & Drupé, 00) se considera un sistema dinámico con un controlador proporcional integral (PI), que será el mismo que consideraremos para nuestro ejemplo.

Las ecuaciones que rigen el modelo son:

$$\text{Motor : } T * \frac{dw}{dt} = c_m * v - w \quad (1)$$

$$\text{Controlador PI : } \frac{dv}{dt} = cc * (d - w_m) \quad (2)$$

$$\text{Sensor : } w_m = c_s * w \quad (3)$$

Las causas de fallo que vamos a considerar en nuestro modelo son provocadas por el funcionamiento anómalo de alguno de los componentes. Esto es provocado principalmente por la desviación del valor nominal de las constantes relacionadas con el componente. Estas constantes se desvían del rango de los valores considerados como correctos. Alguno de dichos fallos hace que esas constantes tengan valores que estén por encima de los valores correctos y otros harán que las constantes estén por debajo. Al efectuar la diagnosis, pues, no solo se deberá indicar el componente que está fallando, sino también si la constante que rige dicho componente se sitúa en valores por encima o por debajo del rango correcto.

Las posibles causas de fallos que deseamos identificar son por tanto:

- Que la constante del motor se sitúe por encima de los valores que consideramos como correctos. A este comportamiento lo identificaremos como: *CmAlta*.
- Que la constante del motor se sitúe por debajo de los valores que consideramos como correctos. A este comportamiento lo identificaremos como: *CmBaja*.
- Que la constante del sensor se sitúe por encima de los valores que consideramos como correctos. A este comportamiento lo identificaremos como: *CsAlta*.
- Que la constante del sensor se sitúe por debajo de los valores que consideramos como correctos. A este comportamiento lo identificaremos como: *CsBaja*.

- Que la constante del controlador se sitúe por encima de los valores que consideramos como correctos. A este comportamiento lo identificaremos como: *CcAlta*.
- Que la constante del controlador se sitúe por debajo de los valores que consideramos como correctos. A este comportamiento lo identificaremos como: *CcBaja*.

Para describir el funcionamiento correcto del sistema vamos a considerar que los valores de las constantes de los componentes no van a tener un único valor correcto, sino que vamos a permitir que dichos valores puedan oscilar dentro de un rango de valores que consideraremos correcto para cada una de las constantes. De esta forma se permite una flexibilidad de funcionamiento al sistema que se acerca más a lo que será un comportamiento real, donde no existe un único valor correcto, sino que dicho valor puede oscilar entre unos márgenes de corrección flexibles. Esto provoca que nuestro sistema no tendrá un único comportamiento correcto, sino una familia de comportamientos correctos que se corresponderán con todas aquellas combinaciones de las constantes de los componentes que se encuentren dentro de los límites de tolerancia considerados.

Esto dificulta la diagnosis al no tener un valor de referencia fijo y único que represente un comportamiento correcto sobre el que medir las posibles desviaciones que se puedan producir, pero por el contrario proporciona una visión más realista del sistema.

Otras consideraciones que haremos al llevar a cabo la diagnosis, y que se consideran presentes en nuestro sistema son:

1. El fallo está presente desde el principio y no evoluciona en el tiempo.
2. Si el sistema evoluciona en el tiempo el cambio de comportamiento se supone que se realiza instantáneamente y a partir de aquí no evoluciona en el tiempo.
3. Una vez que se ha indicado la velocidad angular deseada, no se vuelve a alterar hasta que no se alcance dicha velocidad angular, si el sistema funciona correctamente, o se detecte el fallo en caso de un mal funcionamiento.

De esta forma la diagnosis se realizará en el momento en que se cambie la velocidad angular deseada ( $d$ ), comprobando la forma en que el sistema va a alcanzar dicha velocidad. Hay que tener en cuenta que a pesar de la existencia de un fallo en alguno de los componentes, el controlador PI es capaz de actuar sobre el motor para alcanzar la velocidad final requerida, aunque la forma en la que se consigue, por supuesto, varía de la forma en la que se conseguiría con el sistema funcionando correctamente. Dicha diferencia de comportamiento nos permitirá realizar la diagnosis.

### 3 Generación de Reglas de Decisión

Para realizar la diagnosis de los sistemas dinámicos debemos generar las reglas de decisión a partir de las trayectorias que se conozcan del sistema o las simulaciones que generemos a partir de un modelo. Para llevar a cabo esto, necesitamos un conjunto de definiciones y una notación para la formalización de la misma.

#### 3.1 Definiciones y Notación

**Definición 1: Familia de Comportamientos.** Es un conjunto finito de trayectorias que tienen un comportamiento similar desde el punto de vista del diagnóstico y que se determinan de acuerdo con unos determinados rangos de valores de los parámetros.

**Definición 2: Comportamiento Correcto.** Es el conjunto finito de trayectorias que se consideran son evoluciones del sistema sin ningún tipo de fallo.

**Definición 3: Comportamiento Perfecto.** La trayectoria que describe el sistema cuando los valores de los parámetros toman todos los valores centrales de los rangos que tienen definidos como correctos.

**Definición 4: Observación.** Trayectoria real del sistema dinámico que contiene los valores de las variables que son observables en el sistema.

**Definición 5: Diagnosis.** Es la identificación del comportamiento del sistema, a partir de las reglas de decisión, como perteneciente a una familia de comportamiento determinado (etiqueta de diagnóstico).

La aproximación que se propone en este trabajo, puede ser generada de dos formas diferentes:

1. Las reglas son generadas a partir de un conjunto de modos de funcionamiento diferentes del modelo. Es decir, sometemos al modelo a distintos modos de funcionamiento, que serán posteriormente los modos de funcionamiento del sistema que se quieren diagnosticar.

✓ *Modelo(modo) ⇒ Reglas de decisión*

2. Las reglas son generadas a partir de un conjunto de trayectorias experimentales del sistema dinámico para el comportamiento correcto y para los posibles fallos.

✓ *Trayectorias(modo) ⇒ Reglas de decisión*

A partir de las reglas de decisión generadas mediante la

herramienta de aprendizaje supervisado correspondiente, se puede obtener la diagnosis del sistema dinámico, que consiste en reconocer la observación a diagnosticar como perteneciente a una determinada familia de comportamiento según dichas reglas de decisión.

*Clasifica(Reglas, Observación)-> Etiqueta de diagnóstico*

Donde a partir de una observación del sistema y las reglas de decisión se genera una etiqueta de diagnóstico.

#### 3.2 Propuesta Metodológica

La metodología que se propone, con objeto de llevar a cabo la diagnosis, consiste en obtener un conjunto de reglas que indiquen, para una observación dada, a que *familia de comportamientos* se ajusta.

Para poder hacer esto necesitamos obtener un conjunto de *familias de comportamiento* que definen el comportamiento del sistema bajo las diferentes situaciones que puedan darse. Estas situaciones deberán ser: cualquiera de los posibles fallos o el comportamiento correcto. Todo este proceso se llevará a cabo antes de intentar realizar la diagnosis, de forma off-line, y se tendrá disponible para cuando sea necesario.

De esta forma cada posible situación de funcionamiento, con fallo o sin fallo, se caracteriza por la forma en que el sistema se comporta en dicha situación. La diagnosis vendrá dada, por tanto, al identificar el comportamiento observado como perteneciente a una determinada *familia de comportamientos*. Los pasos para llevar a cabo esta aproximación se representan en la figura 2.

Inicialmente partiremos de una descripción del problema, que para nuestro caso será el presentado con anterioridad. A partir de dicha descripción debemos modelar dicho sistema para poder comenzar a realizar simulaciones con él. En el caso que nos ocupa el sistema se ha modelado como un diagrama de Forrester (Forrester, 68), para poder realizar simulaciones usando la herramienta de simulación VEMSIM®. El diagrama de Forrester generado para el sistema de ejemplo que nos ocupa es el que se presenta en la figura 3. Las variables *f* y *f2* del diagrama de Forrester son variables de flujo, y la función 'INTEG' representa la integral. De esta forma representamos formalmente el sistema de ejemplo de la figura 1.

Las ecuaciones que relacionan las diferentes variables que aparecen en el diagrama son las que siguen:

$$Wm = W * Cs \quad (4)$$

$$V = \text{INTEG}(f2) \quad (5)$$

$$W = \text{INTEG}(f) \quad (6)$$

$$f2 = Cc * (d - Wm) \quad (7)$$

$$f = \frac{Cm * V - W}{T} \quad (8)$$

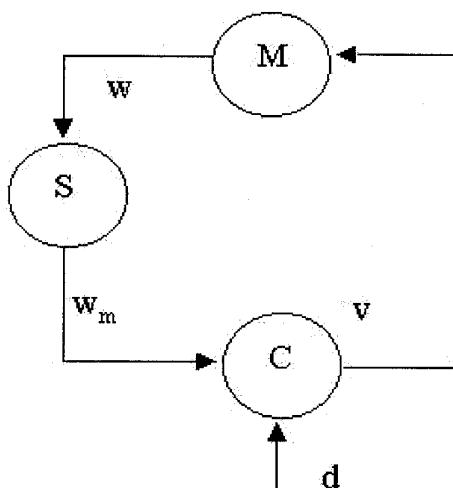


Figura 1: Sistema de ejemplo

Una vez obtenido el modelo debemos simular las distintas situaciones en las que se puede encontrar el sistema. En nuestro caso y como ya se comentó con anterioridad estas situaciones podrán ser: funcionamiento correcto, la constante  $Cm$  está demasiado alta o baja, la constante  $Cs$  está demasiado alta o baja, la constante  $Cc$  está demasiado alta o baja. A partir de ahora nos referiremos a estas situaciones como: *OK*, *CmAlta*, *CmBaja*, *CsAlta*, *CsBaja*, *CcAlta* y *CcBaja* respectivamente.

Cada una de estas situaciones vendrán representadas por un conjunto de trayectorias que se corresponderán a distintas simulaciones con los valores de las constantes de cada componente que se encuentran dentro del rango de la situación elegida.

Se considerará que los valores correctos para el sistema serán los de la tabla 1:

T	3
d	10
W	5
Cm	[0.98-1.02]
Cc	[0.98-1.02]
Cs	[0.98-1.02]

Tabla 1: Valores correctos

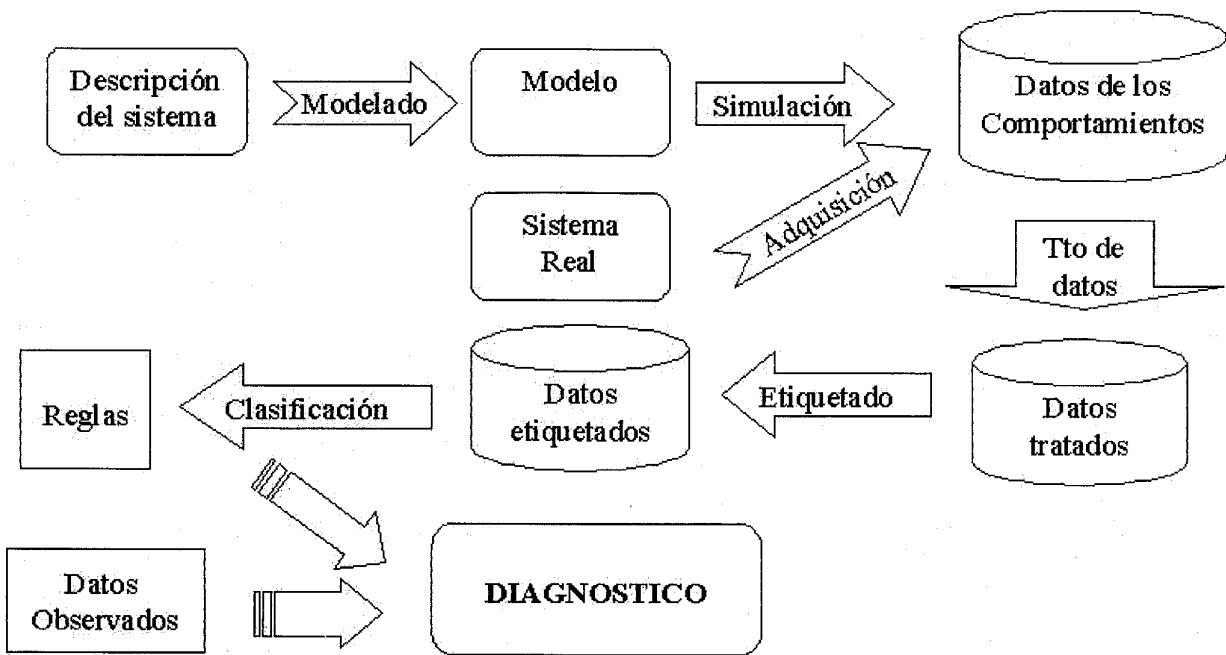


Figura 2: Metodología Propuesta

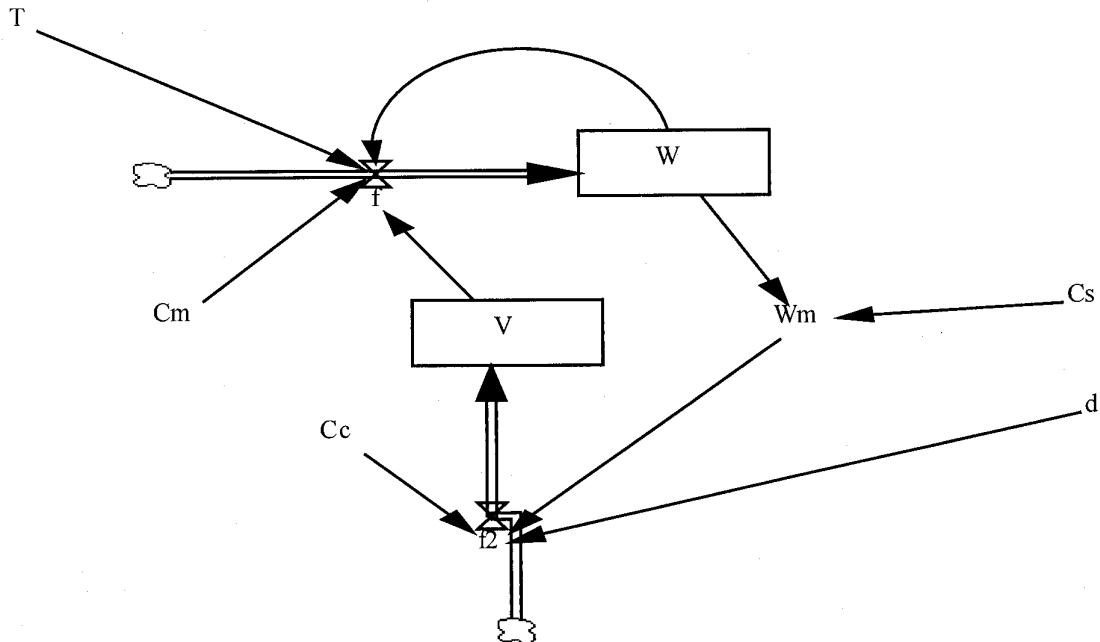


Figura 3: Modelado mediante diagrama de Forrester

Con esto suponemos que nuestro sistema tiene un comportamiento correcto mientras que los valores de las constantes de cada componente oscilen en el rango [0.98-1.02], y que la situación de estudio se corresponde con el motor subiendo de 5 rad/seg a 10 rad/seg, teniendo el motor elegido una inercia ( $T$ ) igual a 3.

Como podemos ver las constantes de los distintos componentes tienen un rango de tolerancia, lo cual simula los posibles ruidos y pequeñas desviaciones que se producen en los sistemas reales, mientras que consideramos que la inercia del motor es fija y tiene un valor de 3.

Para este caso se generarán aleatoriamente un conjunto de simulaciones donde todas las variables de las variables se encuentren dentro de los rangos considerados como correctos. Esos valores serán elegidos por sorteo, según el método Monte Carlo, siguiendo la distribución uniforme. El número de simulaciones dependerá del grado de fiabilidad que se pretenda conseguir, según se demuestra en [Ortega00]. Para nuestro ejemplo vamos a realizar 100 simulaciones para valores de las constantes de los componentes dentro de su rango correcto. Estas 100 simulaciones que vamos a obtener se etiquetarán como OK, y representan la familia de comportamientos correctos para nuestro sistema.

De la misma forma se genera el conjunto de simulaciones (100 para cada caso) correspondientes a las etiquetas  $CmAlta$ ,  $CmBaja$ ,  $CcAlta$ ,  $CcBaja$ ,  $CsAlta$  y  $CsBaja$ , donde los valores

de las constantes variarán entre [0-0.97] para los valores en los que estén bajas y [1.03-5] para los valores en los que estén altas. Cada una de estas simulaciones se etiquetará a continuación con la etiqueta correspondiente a su comportamiento. Obteniendo por tanto un total de 700 simulaciones.

Una vez hecho esto dispondremos de una base de datos donde a cada comportamiento le corresponde su etiqueta para cada simulación. Si en vez del modelo tuviésemos una base de trayectorias etiquetadas de esta forma, el modelado no sería necesario y podríamos continuar a partir de este momento de la misma forma en que vamos a hacerlo.

Los comportamientos vendrán definidos por la forma de la curva que forma la variable  $Wm$  con respecto al tiempo, y que se define por el valor de dicha variable en cada instante de la simulación. Se elige la trayectoria formada por  $Wm$  y no por  $W$  porque esta variable será la única observable en el funcionamiento real del sistema.

Así pues la base de datos de simulaciones estará formada por todas las trayectorias simuladas entre el instante inicial de la simulación y el instante final, y etiquetadas según su comportamiento.

A continuación se muestran algunos ejemplos de simulaciones del sistema para distintos comportamientos.

En la figura 4 podemos apreciar la forma en la que el sistema evolucionará para pasar de 5 rad/seg a 10 rad/seg si todas las constantes de los componentes permanecen dentro del rango de valores de las variables considerados como correctos.

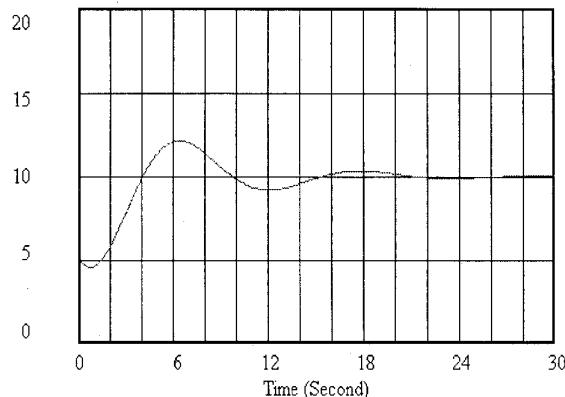


Figura 4: Comportamiento OK

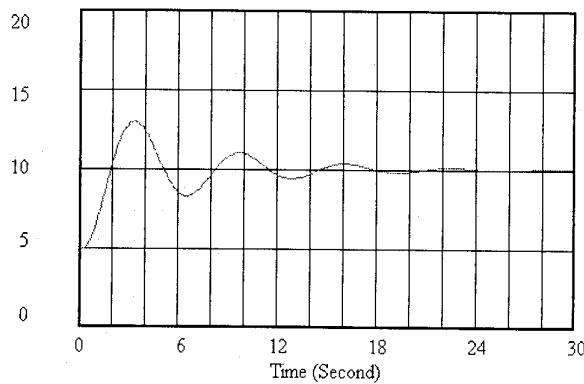


Figura 5: Comportamiento CmAlta

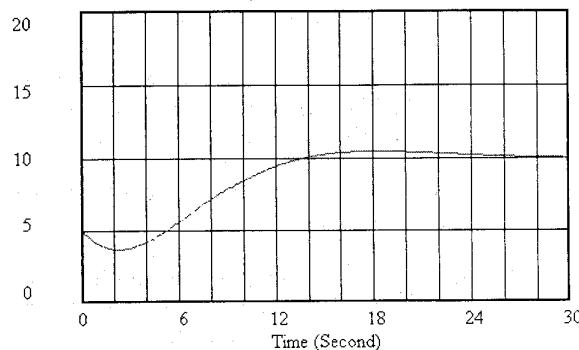


Figura 6: Comportamiento CcBaja

La figura 5 muestra el comportamiento del sistema para el caso en que la constante del motor esté por encima del rango de valores considerado como correctos, mientras que la figura 6 representa el comportamiento del sistema cuando la constante del controlador esté por debajo de los valores correctos.

Una vez generada la base de datos de simulaciones debemos proceder al tratamiento de esa información. Este tratamiento vendrá dado en dos sentidos, en primer lugar la información recogida por la variables  $W_m$  es muy pobre como para diagnosticar posibles comportamientos futuros, y en segundo lugar la cantidad de puntos que forman una trayectoria es excesivamente grande como para poder trabajar con ellos. Por tanto el tratamiento de los datos consistirá en reducir el número de puntos que forman la trayectoria y enriquecerlos con más información.

Para reducir el número de puntos que forman la trayectoria realizamos un muestreo de la misma. Es necesario seleccionar los puntos que forman la muestra de forma constante, puesto que cualquier otro método elegido, por ejemplo un método estadístico, modificaría la forma de la curva, base para nuestra diagnosis. El único motivo para reducir el número de puntos de cada trayectoria es la imposibilidad de manejar trayectorias excesivamente grandes por parte del clasificador. Mientras menos puntos tenga la muestra más fácil será de trabajar con ella, pero por el contrario perdemos parte de la información sobre la trayectoria. En nuestro caso el muestreo se realiza tomando uno de cada tres puntos de la trayectoria original.

Para cada uno de esos puntos seleccionados se calculan las siguientes magnitudes, con objeto de obtener mayor información del comportamiento de la trayectoria de la que proporcionan únicamente las magnitudes medidas.

1. Inclinación relativa ( $IR$ ). Determina el grado de inclinación con que la trayectoria llega a ese punto, independientemente de su posición absoluta. Con esto conocemos la pendiente que forma la recta que une dicho punto con el punto inmediatamente anterior. Será una medida de lo rápido que aumenta la velocidad angular medida ( $w_m$ ) en dicho punto. Se calcula como:

$$IR(i) = \frac{W_m[i] - W_m[i-1]}{\Delta t} \quad (9)$$

El punto  $W_m[i-1]$  será el punto anterior al seleccionado en la trayectoria original, y no el anterior punto de la muestra tomada. De esta forma aumentamos la precisión de dicha magnitud, puesto que si se elige el punto anterior de la muestra estamos eligiendo dos puntos muy distantes, entre los cuales la curva puede variar su forma significativamente, y dicha información se estaría perdiendo. Sin embargo al elegir el

punto más cercano posible la información que obtenemos es más precisa.

2. Distancia al comportamiento perfecto (*DP*). Determina lo lejos o cerca que nos encontramos del comportamiento perfecto. Con esta magnitud se pretende obtener una medida de proximidad o alejamiento, del comportamiento actual, al comportamiento que tendría el sistema si todos los valores de las constantes de los componentes tomaran el valor central del rango en que se considera que el comportamiento del componente es correcto. Se calcula como:

$$DP(i) = Wm[i] - Wmpf[i] \quad (10)$$

Donde *Wmpf*[i] es el valor de *Wm* en el punto i para la trayectoria perfecta.

3. Inclinación relativa a la trayectoria perfecta(*IRP*). Determina la inclinación de acercamiento o alejamiento a la trayectoria perfecta. Esta medida proporciona, para cada uno de los puntos de la muestra seleccionada., la pendiente que tendrá, en cada punto, la trayectoria formada por los puntos que conforman la trayectoria de la distancia al comportamiento perfecto. Se define como:

$$IRP(i) = \frac{DP(i) - DP(i-1)}{\Delta t} \quad (11)$$

*DP*(i-1) es referida al punto anterior en la trayectoria original, y no al punto anterior de la muestra seleccionada, con lo cual el punto *DP*(i-1) debe calcularse como la distancia entre el punto anterior en la trayectoria original y el punto anterior en la trayectoria perfecta, antes de realizar el muestreo.

De esta forma tendremos que hemos pasado de una base de datos de trayectorias donde cada trayectoria se representa de la forma:

*Wm*[1], *Wm*[2], *Wm*[3], ..., *Wm*[n], ..., ETIQUETA

A otra donde cada trayectoria se representa de la forma:

*Wm*[i], *IR*(i), *DP*(i), *IRP*(i), *Wm*[j], *IR*(j), *DP*(j), *IRP*(j), ..., ETIQUETA

Así pues tenemos ahora una base de datos con todas las trayectorias simuladas, representadas por una serie de puntos de cada una y por un conjunto de magnitudes que las caracterizan. Cómo nuestro objetivo final es clasificar los distintos comportamientos que nos vienen representados por un conjunto de curvas, estas magnitudes nos aportan información valiosa sobre cada uno de los citados comportamientos.

Si en vez de tener un modelo del sistema tuviésemos datos recogidos a partir del sistema real el tratamiento a realizar sería exactamente el mismo.

Esta base de datos servirá como entrada para realizar el aprendizaje supervisado. El aprendizaje se llevará a cabo mediante una herramienta de clasificación, que en nuestro caso será el C4.5(Quinlan, 93), que será capaz de clasificar, en función de la etiqueta, los valores que deben tomar cada uno de los datos aportados para que se correspondan con el comportamiento asociado a la etiqueta. Lo que se consigue con esta herramienta es caracterizar cada una de las *familias de comportamiento* según los valores de los atributos que se les ha proporcionado. Para los datos de nuestro ejemplo el tamaño del clasificador generado es de 37 nodos.

La salida del clasificador estará compuesta por un conjunto de reglas de clasificación que serán las que nos sirvan para llevar a cabo la diagnosis.

## 4 Diagnosis

La realización de la diagnosis se llevará a cabo de la forma que se expone a continuación:

Cada vez que el sistema alcance uno de los instantes de tiempo, para los cuales se han seleccionado datos de muestra, se calculan las ecuaciones (9) (10) y (11) para dicho valor observado.

Estas magnitudes calculadas, junto con el valor observado, se comparan con las reglas de decisión obtenidas al realizar la clasificación de las simulaciones. En el momento en que se cumpla una de las reglas, sabremos a que *familia de comportamiento* pertenece la forma de actuar del sistema, y por tanto la etiqueta asociada a dicho comportamiento nos indica el tipo de fallo que existe u *OK* si la evolución del sistema se corresponde con el comportamiento correcto. Si el valor observado actual no proporciona ningún tipo de conclusión con el conjunto de reglas de que se dispone, se actúa de la misma forma con el siguiente punto observado que se corresponda con un instante de muestreo.

Hay que resaltar que para esta forma de diagnosis no sólo nos fijamos en el valor de los puntos finales de la simulación, es decir en el estacionario, sino que se comprueba, en el transitorio, la forma en que la trayectoria evoluciona, viendo así, según la forma de evolución del sistema, a que tipo de comportamiento corresponde.

Esta forma de diagnosis es poco costosa, puesto que el trabajo ‘duro’ se ha realizado previamente *off-line*, y la diagnosis consiste únicamente en comparar el valor observado, junto con sus magnitudes asociadas, con los valores que aparecen en cada una de las reglas de decisión que definen los comportamientos.

Para cada modo de funcionamiento del sistema disponemos del conjunto de reglas de decisión que le corresponda, previamente generadas. Así pues, si el motor gira a 10 rad/seg. y se pretende reducir su velocidad a 6 rad/seg., deberemos

diagnosticar con las reglas generadas para esas condiciones, pero en cualquier caso la simulación ya estaría hecha con anterioridad.

Hay ocasiones en que las reglas basan su clasificación en valores de los atributos de instantes de tiempo muy tardíos. Si se pretende obtener una diagnosis temprana, debemos generar simulaciones a ‘tiempo parcial’, es decir, simulaciones que tengan como momento final instantes de tiempo más tempranos, con objeto de generar reglas de decisión que puedan diagnosticar la evolución parcial del sistema hasta el instante de tiempo elegido. Superado dicho instante se deberá seguir diagnosticando con un conjunto de reglas que hayan sido generadas para instantes de tiempo posteriores, o en último caso con el conjunto de reglas que se han generado según las trayectorias completas hasta el instante en que se alcanza nuevamente el equilibrio en el sistema, es decir, cuando se alcanza la velocidad final deseada.

## 5 Resultados

Para evaluar la metodología vamos a llevar a cabo un conjunto de pruebas que nos permitan ver lo acertado de las diagnosis obtenidas.

Para los casos de ejemplo se han utilizado los siguientes valores fijos:  $T=3$ ,  $W=5$ ,  $d=10$ .

Esto indica que nuestro motor deberá tener una inercia fija de 3, y que se pretende acelerar desde 5 rad/seg hasta 10 rad/seg. Los datos de la velocidad angular medida se obtendrán cada 0,1 unidades de tiempo, y para la generación de reglas y la diagnosis vamos a quedarnos con una muestra de 1 de cada 3 datos obtenidos.

Los datos de observación del sistema se obtienen realizando la simulación del mismo en diferentes condiciones de funcionamiento. De esta forma conocemos la diagnosis que deberíamos obtener y lo podemos comparar con la diagnosis obtenida a partir del conjunto de reglas.

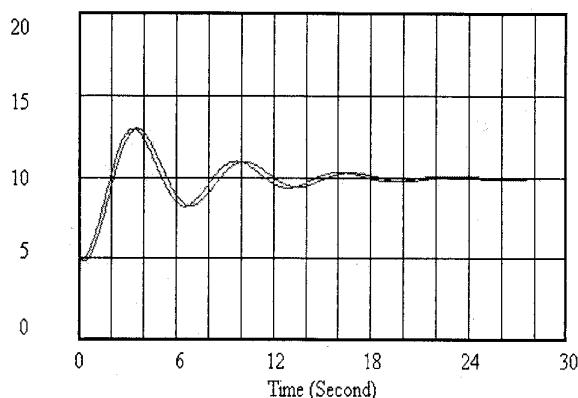


Figura 7: Comportamiento CcAlta vs CmAlta

El resultado de la diagnosis de este caso se puede ver en la siguiente tabla de pruebas:

Cm	Cc	Cs	Valores de funcionamiento		Diagnosis Correcta	Diagnosis Obtenida
			Cm	Cc		
1	1	1.03	Cs Alta		<b>OK</b>	
1	1	1.07	Cs Alta		Cs Alta	
1	1	1.1	Cs Alta		Cs Alta	
1	1	1.5	Cs Alta		Cs Alta	
1	1	2	Cs Alta		Cs Alta	
1	1	3	Cs Alta		Cs Alta	
1	1.03	1	Cc Alta		<b>OK</b>	
1	1.07	1	Cc Alta		Cc Alta	
1	1.1	1	Cc Alta		Cc Alta	
1	1.5	1	Cc Alta		Cc Alta	
1	2	1	Cc Alta		Cc Alta	
1	3	1	Cc Alta		Cc Alta	
1.03	1	1	Cm Alta		<b>OK</b>	
1.07	1	1	Cm Alta		<b>Cc Alta</b>	
1.1	1	1	Cm Alta		<b>Cc Alta</b>	
1.5	1	1	Cm Alta		Cm Alta	
2	1	1	Cm Alta		Cm Alta	
3	1	1	Cm Alta		Cm Alta	
1	1	0.97	Cs Baja		Cs Baja	
1	1	0.93	Cs Baja		Cs Baja	
1	1	0.89	Cs Baja		Cs Baja	
1	1	0.85	Cs Baja		Cs Baja	
1	1	0.5	Cs Baja		Cs Baja	
1	1	0.1	Cs Baja		Cs Baja	
1	0.97	1	Cc Baja		<b>OK</b>	
1	0.93	1	Cc Baja		Cc Baja	
1	0.89	1	Cc Baja		Cc Baja	
1	0.85	1	Cc Baja		Cc Baja	
1	0.5	1	Cc Baja		Cc Baja	
1	0.1	1	Cc Baja		Cc Baja	
0.97	1	1	Cm Baja		<b>OK</b>	
0.93	1	1	Cm Baja		Cm Baja	
0.89	1	1	Cm Baja		Cm Baja	
0.85	1	1	Cm Baja		Cm Baja	
0.5	1	1	Cm Baja		Cm Baja	
0.1	1	1	Cm Baja		Cm Baja	
0.99	0.98	1.02	OK		OK	
1	1.02	1.02	OK		OK	
0.98	1	0.98	OK		OK	
0.98	1.02	1.02	OK		OK	
0.99	1.01	1.01	OK		OK	
1.01	1	0.99	OK		OK	

Tabla 2: Resultados de las pruebas

Observamos que en los casos en los que los fallos vienen dados por valores de las constantes muy cercanas a la de los valores correctos, la diagnosis no es acertada, indicando que el comportamiento es correcto. Esto se debe a la relación existente entre las constantes, y a permitir un rango de valores para los comportamientos correctos, lo que provoca que esa pequeña desviación en una de las constantes pueda ser ‘compensado’ por un valor de otra de las constantes en sentido contrario pero dentro de sus márgenes de tolerancia.

La diagnosis también es errónea para ciertos valores de *CmAlta*, diagnosticándose *CcAlta*. Esto se debe a que la estrecha relación existente entre las constantes de los componentes para este sistema provoca que alteraciones complementarias en estas constantes haga que el sistema evolucione de una misma forma. En la figura 7 podemos ver como para los mismos valores de *Cs* y con valores de *Cm* y *Cc* complementados el comportamiento del sistema es prácticamente idéntico. En este caso las simulaciones se han realizado con *Cs*=1, *Cc*=1 y *Cm*=3 para *CmAlta* y *Cs*=1, *Cc*=2.8 y *Cm*=1 para *CcAlta*.

Actualmente se están desarrollando técnicas que permitan paliar, en la medida de lo posible, estos errores, aunque como puede observarse en la figura 7, hay casos en los que será muy difícil distinguir el comportamiento del sistema, puesto que la salida que se genera es prácticamente la misma.

No obstante podemos comprobar que en general la diagnosis es bastante acertada, sobre todo cuando la constante que falla se aleja significativamente de sus valores correctos.

Hay que resaltar también que mientras los valores de las constantes se mantienen en el rango que se considera de funcionamiento correcto, la metodología propuesta funciona correctamente, aunque los valores de dichas constantes estén en sus extremos.

Para el juego de pruebas que se ha presentado se consigue diagnosticar el 80.55% de los fallos de forma correcta. Se diagnostica un fallo distinto al existente en un 5.55% de los casos, y no se encuentra el fallo en un 13.88% de los casos, teniendo un 0% de falsos positivos, es decir, de indicar un fallo sin que exista. No obstante con un juego de pruebas más exhaustivo estos porcentajes pueden mejorar bastante.

## 6 Conclusiones y Trabajos Futuros

La metodología presentada lleva a cabo diagnosis en sistemas dinámicos independientemente del tipo de sistema de que se trate y aislando de las características del mismo. Para ello nos centramos fundamentalmente en los posibles comportamientos que toma el sistema para cada uno de los casos de fallo o para el comportamiento correcto. De esta forma, y puesto que cada uno de los posibles fallos del sistema conlleva un comportamiento distinto del mismo, podemos deducir el fallo que se produce comparando el comportamiento observa-

do con todos los comportamientos simulados. Esta comparación se puede realizar desde el instante inicial y fijándonos no solamente en la situación a la que se llega sino en la forma en que se llega a dicha situación.

Tendremos una batería de simulaciones preparadas para cada forma de trabajo del sistema, de tal forma que en el momento de la diagnosis tan solo habrá que seleccionar el conjunto de reglas que se corresponda a la forma de trabajo que hemos elegido. Por ejemplo el motor acelerando de 5 a 10 rad/seg. o decelerando de 10 a 4 rad/seg.

La obtención de reglas de decisión, que caracterizan las distintas evoluciones según el valor de los atributos en ciertos instantes de tiempo, nos facilita la diagnosis en tiempo real, comprobando la forma de evolución en cada uno de los instantes de tiempo que vamos observando.

En cierto tipo de sistemas, como en el ejemplo que hemos tratado, hay ocasiones en que la relación existente entre las constantes que rigen el comportamiento puede provocar que una misma trayectoria se corresponda con simulaciones de casos distintos. Esto ocurre porque el sistema puede comportarse de la misma forma para fallos de distinto tipo, que provoquen la misma evolución con alteraciones de constantes distintas en sentido contrario. Puesto que este tipo de fallos no es diagnosticable, al obtenerse una evolución del sistema igual para fallos distintos, lo que pretendemos hacer en futuros trabajos es asociar el comportamiento a cualquiera de los fallos, de tal forma que se identifique la evolución como que el fallo es provocado por un componente u otro, sin poder especificar cuál de ellos es.

Otro campo de actuación que se encuentra actualmente en curso es el de poder diagnosticar combinaciones de fallos de componentes, es decir, identificar los comportamientos del sistema cuando hay más de un componente que está fallando al mismo tiempo.

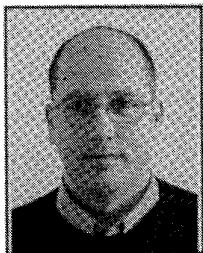
## 7 Reconocimientos

Este trabajo ha sido parcialmente financiado por la Comisión Interministerial de Ciencia y Tecnología (DPI2000-0666-c02-02) y por el Grupo de Investigación de Modelización Matemática, Redes y Multimedia de la Universidad de Huelva.

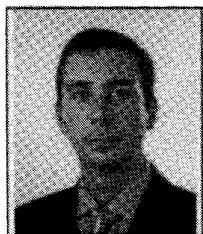
## Referencias

- K. Bousson, y L. Trave-Massuyes** A computational causal model for process supervision. Technical Report 92147, LAAS-CNRS, Toulouse, France, (1992).
- V. Brusoni, L. Console, P. Terenziani, D. T. Dupré** A spectrum of definitions for temporal model-based diagnosis. En Artificial Intelligence 102 39-79, (1998).

- L. Console., L. Portinale, D. Theseider Durpé, Y P. Torasso,** *Diagnosing time-varying misbehavior: an approach based on model decomposition.* En *Annals of Mathematics and Artificial Intelligence, special issue on Model-based diagnosis.* Console, L. and Friedrich, G.(eds.) 1994.
- L. Console and Dressler.** *Model-based diagnosis in the real world lessons learned and challenges remaining.* En Proceedings de IJCAI'99 Stocolmo. 1999
- J. de Kleer, A. Mackworth, and R. Reiter,** *Characterizing diagnoses and systems* Artificial Intelligence. 1992.
- J. De Kleer y B. Williams.** *Diagnosing multiple faults.* Artificial Intelligence 32, 97-130, 1987
- O. Dressler y H. Freitag.** *Prediction Sharing Across Time and Contexts.* En actas del 8<sup>th</sup>. Interntation Workshop on Qualitative Reasoning about Physical Systems(QR'94) 63-68, Japón, 1994.
- O. Dressler y P. Struss,** *Model-based diagnosis with the Default-based Diagnosis Engine: effective control strategies that work in practice.* En Proceedings of the 11<sup>th</sup> European Conference on Artificial Intelligence (ECAI-94), 677-681, 1994
- Dressler.** *On-Line Diagnosis and Monitoring of Dynamic Systems based on Qualitative Models and Dependency-recording Diagnosis Engines.* Proc. 12<sup>th</sup>. European Conference on Artificial Intelligence(ECAI-96) 461-465, 1996.
- D. Dvorak y B. Kuipers** *Mode Based monitoring of dynamic systems.* En W. Hamscher, L. Console, y De Kleer eds. Reading in model-based diagnosis. San Mateo CA Morgan Kaufmann 249-254.
- J. W. Forrester** *Principles of systems.* Wright-Allen Press 1968
- P. Frank** *Fault diagnosis in dynamic systems via state estimation: a survey.* System Fault Diagnostics Reliability and Related Knowledge-based Approaches Vol I(Tzafestas, Singh and Schmidt(Eds.)). 1987
- P. Frank.** *M. Analytical and qualitative model-based fault diagnosis -a survey and some new results.* European Journal of Control2. 1996
- J. Gertler** *Analytical redundancy methods in failure detection and isolation in complex plants.* Proc. of the IFAC/IMACS Symposium SAFEPROCESS-91. 1991
- R. Iserman** *Experiences with process fault detection methods via parameter estimation. .* En *System Fault Diagnostics Reliability and Related Knowledge-based Approaches* Vol I(Tzafestas, Singh and Schmidt(Eds.)) 35-98. 1987
- R. Iserman.** *Supervision, fault detection and fault-diagnosis -an introduction* Control Engineering Practice 5(5). 1997.
- B.J. Kuipers.** *Qualitative Simulation.* Artificial Intelligence 29. 1986
- S. Leonhart y M. Ayoubi.** *Methods of fault diagnosis.* Control Engineering Practice, 5. 1997.
- A. Malik and P. Struss.** *Diagnosis of Dynamic Systems does not necessarily require simulation.* Workshop Notes of the Seventh International Workshop on Principles of Diagnosis DX-96 Montreal. 1996
- P. Mosterman** *Hybrid dynamic systems: a hybrid bond graph modeling paradigm and its applications in diagnosis.* Tesis Doctoral Vanderbilt University, Nashville, Tennessee, USA
- H.T. Ng.** *Model-based, multiple fault diagnosis of time-varying , constinuous physical devices.* Proceedings 6<sup>th</sup> IEEE Conf. On AI Applications, Santa Barbara, CA . 1990
- H.T. Ng.** *Model-based, multiple fault diagnosis of dynamic, constinuous physical devices.* IEEE Expert 6(6). 1991.
- J. Ortega Ramírez.** *Patrones de comportamiento temporal en modelos semicualitativos con restricciones.* Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. 2000
- A. Panati and D. T. Drupé** *Stated based vs simulation-based diagnosis of dynamic system.* ECAI2000. 14<sup>th</sup> European Conference on Artificial Intelligent. 2000.
- R.J. Patton, P. M. Frank and R.N. Clark (eds).** *Issues of Fault Diganosis for Dynamic Systems* Springer-Verlag. 2000.
- A.D. Pouliezos y G.S. Stavrakakis.** Real time fault monitoring of industrial process. Microprocessor-based systems engineering. Kluwer Academic Publishers, Dordrecht, 1994.
- C. Price.** *Computer-based diagnositic systems.* Springer Verlag. Nueva York, 1999.
- B. Pulido, C. Alonso. Y F. Acebes.** *Consistency-based diagnosis of dynamic systems using quantitative models and off-line dependency-recording.* Proc. DX2001.
- J. Ross Quinlan.** *Induction of decision trees.* Machine learning, 1986
- J. Ross Quinlan.** *C45:Program for Machine Learning.* Morgan Kaufman, 1993
- R. Reiter** *A theory of diagnosis from first principles.* Artificial Intelligence, 32(1). 1987
- P. Struss.** Fundamentals of model-based diagnosis of dynamic systems. Proc. IJCAI'97. 1997.
- V. Venkatusubramanian and K. Chan.** *A neural network methodology for process fault diagnosis.* Journal of Artificial Intelligence in Chemical Engineering, 35:1993-2001. 1995



**Pedro José Abad Herrera**, recibió el Diplomado en Informática en 1991 en la Universidad de Sevilla, España en 1999 obtuvo el grado de Ingeniero en Ciencias de la Computación por la misma Universidad. Desde 1992 trabaja en el Departamento de Ingeniería Electrónica y Ciencias de la Computación de la Universidad de Huelva. Actualmente realiza su Doctorado con el tema Diagnosis Automática.



**Antonio J. Suárez Fabrega**, recibió el Diplomado en Informática en 1991 en la Universidad de Sevilla, España en 1999 obtuvo el grado de Ingeniero en Ciencias de la Computación por la misma Universidad. Desde 1992 trabaja en el Departamento de Ingeniería Electrónica y Ciencias de la Computación de la Universidad de Huelva. Actualmente realiza su Doctorado con el tema Razonamiento Cualitativo.



**Rafael M. Gasca**, Obtuvo el título de Doctor en Informática en 1998 en la Universidad de Sevilla, España. Desde 1991 es profesor del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla. Sus principales áreas de investigación son: la programación con restricciones y el razonamiento semicualitativo.



**Juan Antonio Ortega**, nació en 1968 y es Ingeniero en Informática desde 1992 y Doctor en Informática desde el año 2000. ambos títulos los obtuvo en la Universidad de Sevilla, España. Actualmente es profesor del Departamento de Lenguajes y Sistemas Informáticos de la misma Institución. Sus principales áreas de investigación son: la simulación de sistemas dinámicos con conocimiento semicualitativo y la obtención de sus patrones de comportamiento temporal.

