Computación y Sistemas Vol. 6 No. 2 pp. 116 - 129
© 2002, CIC - IPN. *ISSN 1405-5546* Impreso en México

Diagnosis en Fases Tempranas de Sistemas Dinámicos

Early Diagnosis of Dynamic Systems

Pedro J. Abad[1], Antonio J. Suárez[1], Juan Antonio Ortega[2] y Rafael M. Gasca[2]
[1]Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática
Escuela Politécnica Superior La Rábida, Universidad de Huelva, Campus de la Rábida.
Carretera Huelva-Palos de la Frontera s/n 21071, Palos de la Frontera, Huelva, España
[2]Departamento de Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática, Universidad de Sevilla
Avda. Reina Mercedes s/n 41012, Sevilla, España
E-mail: {abadhe, asuarez}@uhu.es {ortega, gasca}@lsi.us.es

Artículo recibido en Marzo 17, 2002; aceptado en Octubre 01, 2002

Resumen

La diagnosis es una de las áreas de investigación a la que las instituciones europeas y las empresas privadas dedican más recursos. Esto es debido a la pérdida de productividad y deterioro de equipos que provocan fallos no diagnosticados a tiempo. Por estos motivos, nuestra investigación se centra en diagnosis sobre sistemas dinámicos, usando aprendizaje supervisado. En la presente aproximación tratamos la detección de fallos y diagnosis en momentos tempranos del transitorio entre dos estados del sistema dinámico. Realizamos, por tanto, la diagnosis durante los cambios de estado debidos al funcionamiento normal del sistema, o provocados como pulsos de control generados para la diagnosis.
La diagnosis se realizará en dos fases: la primera de ellas, la de aprendizaje, que es la más costosa computacionalmente, se realizará off-line, la segunda, la diagnosis, se realizará durante el funcionamiento del sistema. En la primera fase obtendremos 3 árboles de decisión en instantes de tiempo que se corresponden con un tercio, dos tercios y el final del transitorio, y en la segunda detectaremos y diagnosticaremos el error usando los citados árboles de decisión.
Palabras clave: Detección de Fallos, Diagnosis, Aprendizaje Supervisado, Reglas de Decisión.

Abstract

Diagnosis is one of the research areas in which European institutions and private companies dedicate more resources. This is due to the loss of productivity and machine deterioration that are caused because don't diagnose on time For these reasons, our investigation is centered in diagnosis of dynamic systems, by using supervised learning tools. In the present approach, the fault detection and diagnosis, in two early moments of the transitory between two states of the dynamic system, will be treated. Therefore, the diagnosis will be carried out during the changes of the state in the normal operation of the system, or in a provoked control pulse which are generated for the diagnosis
The diagnosis will be carried out in two phases: the first one, the learning task, is the longest in time and it will be carried out off-line. The second phase, the diagnosis task, will be carried out on-line, during the operation of the system. After the first phase, 3 decision trees will be obtained. They will correspond to instants of time in 1/3,2/3 and the end of the transitory. In the second phase, the possible fault will be detected and diagnose using the previous decision trees.
Keywords: Fault Detection, Diagnosis, Supervised Learning, Decision Rules.

1 Introducción

Entendemos por *detección de fallos* la tarea de determinar, a partir de las observaciones realizadas, cuándo existe un funcionamiento incorrecto del sistema sujeto a observación, y por *diagnosis* cuáles son las causas de ese comportamiento incorrecto.

La detección y la diagnosis del funcionamiento anómalo de mecanismos y procesos son importantes 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 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 costes 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.

Cuando se pretende, además, realizar diagnosis de sistemas dinámicos la complejidad aumenta. La característica común a este tipo de sistemas es su comportamiento dinámico, donde el tiempo es un factor muy importante a tener en cuenta, y donde la dinámica de los síntomas puede ser comparable a la del propio proceso, de tal forma que no se detectará de la misma forma una pequeña fuga de un tanque que la obstrucción total de una válvula. En (Struss, 97) se presentan los fundamentos de las *diagnosis basada en modelos* para sistemas dinámicos. En este trabajo se discuten los fundamentos teóricos y los aspectos prácticos de la aplicación de la *diagnosis basada en modelos* (particularmente la *diagnosis basada en consistencia*) para sistemas dinámicos

Una de las discusiones más significativas que se han observado en los últimos años versa 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, o es mejor realizar la simulación del sistema para tener en cuenta los cambios entre estados. En el primer caso, la *diagnosis basada en estados*, sólo se razona sobre los estados simples del sistema, sin tener en cuenta los cambios de estado, mientras que en el segundo caso, la *simulación*, el comportamiento del sistema se trata de forma continua, tanto en los estados como en las transiciones entre ellas.

Los defensores de la propuesta basada en estados (Malik & Struss, 96), (Dressler, 96) argumentan que la simulación numérica no es aplicable si sólo tenemos información parcial o cualitativa del sistema y sus condiciones iniciales. En estos casos la simulación puede llegar a ser muy compleja, debido a la ambigüedad existente en el conjunto de predicciones de comportamiento.

Un trabajo posterior (Panati & Drupa, 00) presenta unos resultados opuestos a los presentados (Malik & Struss, 96), para el mismo ejemplo, mostrando que la simulación puede ser útil para restringir el uso de posibles diagnósticos.

En el presente artículo se pretende usar la simulación para, mediante técnicas de *aprendizaje supervisado*, demostrar que se pueden obtener buenos resultados en la diagnosis de sistemas dinámicos.

Existen dos enfoques fundamentales en el campo del *aprendizaje*: el primero incorpora nueva información al conocimiento ya adquirido (aprendizaje mediante el análisis de diferencias o aprendizaje mediante la aplicación de experiencias), mientras que el segundo extrae conocimiento de la regularidad de los datos (programación lógica, técnicas conexionistas, programación genética o clasificación automática). Este segundo enfoque será el que se sigue en el presente artículo.

Con estas técnicas, la diagnosis se consigue mediante la comparación de la evolución del sistema a diagnosticar con el conocimiento aprendido que se tiene del mismo, evaluando de ésta forma el comportamiento del sistema. Para la adquisición del conocimiento se suele recurrir a experiencias previas almacenadas o a la simulación del comportamiento del sistema a diagnosticar en las distintas situaciones que se pueden presentar. Generalmente, no suelen estar disponibles tantos ejemplos de situaciones reales de fallo, por lo que deben recurrir a simulaciones [Balakrishnan&Jonavar98] de los mismos.

Desde hace algún tiempo las técnicas de aprendizaje vienen siendo aplicadas al campo de la diagnosis de muy diversas formas, como pueden ser los métodos estocásticos (Pouliezos&Stavrakakis, 94), técnicas de tipo conexionistas (Venkatusubramanian & Chan, 95) y sistemas de clasificación (Leonhart & Ayoubi, 97). Más recientes son los trabajos de (Fuente, 01), donde se exponen tres formas distintas de aplicar las Redes Neuronales Artificiales al campo de la detección y diagnóstico de fallos, (Saludes et al., 01) que utiliza una red neuronal SOM, o (Simón et al., 01), donde podemos encontrar técnicas conexionistas combinadas con lógica borrosa.

En cuanto a los sistemas de clasificación, dentro del campo del aprendizaje supervisado, son procedimientos automáticos, basados en operaciones lógicas, que aprenden una tarea a partir de un conjunto de ejemplos. En el campo de la clasificación la atención se ha centrado, concretamente, en aproximaciones con árboles de decisión (Quinlan, 86) donde la clasificación es el resultado de una serie de pasos lógicos. Estas aproximaciones son capaces de representar los sistemas más complejos si tienen datos suficientes. Aplicados a la diagnosis, podemos encontrar aplicaciones de estos métodos usados para la clasificación de patrones temporales (Rodríguez et. al, 01), o en trabajos previos al presente (Suárez et al. 01), (Abad et al., 01).

En el presente artículo veremos como usando *aprendizaje supervisado*, realizado a partir de las simulaciones del sistema dinámico, podemos llegar detectar y diagnosticar, con cierta precisión, diversas situaciones de fallo. El aprendizaje se realizará, además, de forma incremental, permitiendo obtener un conjunto de reglas de clasificación en instantes muy tempranos, lo que posibilita tener un diagnóstico en los primeros instantes de funcionamiento del sistema.

El artículo ha sido organizado de la siguiente manera: en la siguiente sección se introducen las definiciones y notaciones necesarias para el desarrollo de la metodología de trabajo. A continuación expondremos la metodología empleada y la forma de realizar la diagnosis. La siguiente sección nos presenta el caso de estudio seleccionado para validar la propuesta. 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 Definiciones y Notaciones

Para realizar la diagnosis de sistemas dinámicos es necesario establecer ciertas definiciones sobre las cuales basaremos el modo de operación de nuestra metodología. Estas definiciones son las siguientes:

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, rangos que determinan las distintas categorías de comportamiento.

Definición 2: Comportamientos correctos. Es el conjunto finito de trayectorias que son consideradas como evolución del sistema en ausencia de fallo.

Definición 3: Comportamiento perfecto. Definimos comportamiento perfecto como 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, o aquellos valores determinados por el fabricante, es decir, aquellos en los que un sistema ideal debería permanecer.

Definición 4: Observación. Datos de las variables observables, del sistema real, que forman una trayectoria del sistema dinámico para ese comportamiento concreto. Son datos que se obtiene on-line.

Definición 5: Detección. Es la identificación de la presencia de fallo en una observación del sistema real a partir de los árboles de decisión obtenidos durante el aprendizaje.

Definición 6: Diagnosis. Tarea de determinar cual es la causa del fallo, identificando la observación del sistema real como perteneciente a la familia de comportamiento de ese fallo.

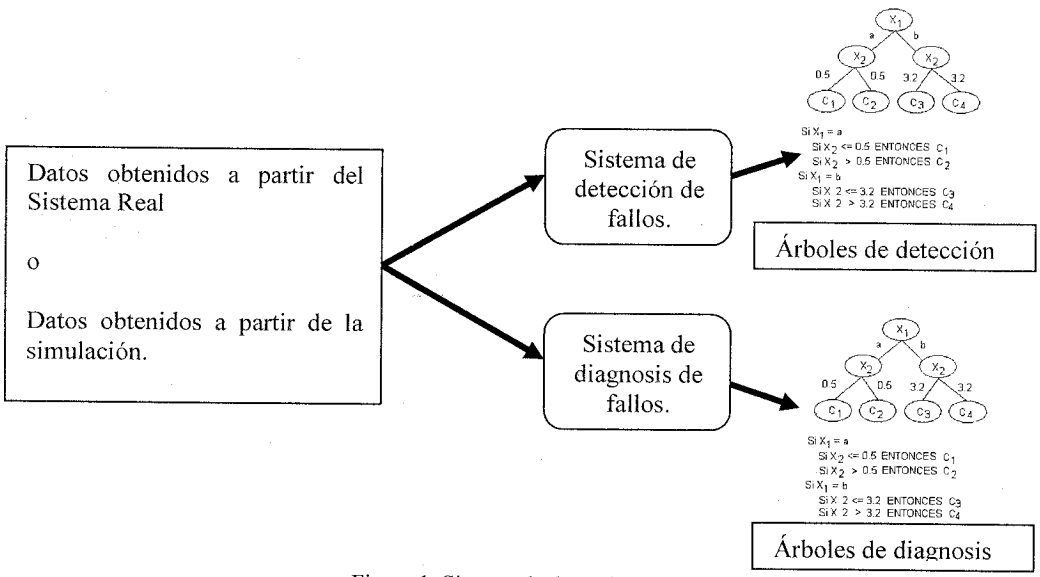

Figura 1: Sistema de detección y diagnosis.

3 Propuesta Metodológica

La metodología que se propone con objeto de llevar a cabo la diagnosis consiste en obtener un conjunto de reglas o árboles de decisión que indiquen, para una observación dada, si ésta presente o no algún fallo sin identificar cual. A continuación, y del mismo modo, se propone obtener un conjunto de reglas o árboles de decisión que identifiquen el error detectado con anterioridad. La metodología parte del sistema real o de una simulación de éste. A continuación, mediante el sistema de detección de fallos primero y posteriormente el sistema de diagnosis, se procede a la obtención de los árboles de detección y de diagnosis respectivamente, tal y como se indica el la figura 1.

Pasemos a describir por tanto el sistema de obtención del árbol de detección de fallos y del árbol de identificación de fallos o de diagnosis.

3.1 Sistema de Detección

El sistema de detección, como ya se ha indicado, tiene como objetivo obtener un árbol de decisión que detecte la presencia o ausencia de fallo, indicando si el comportamiento del sistema es correcto o incorrecto. El funcionamiento del sistema de detección se puede ver en la figura 2. Hay que resaltar que el sistema de obtención de árboles de detección y diagnosis se realiza off-line, de tal forma que eliminamos el tratamiento computacionalmente duro del funcionamiento en tiempo real.

El sistema de detección puede partir de dos puntos distintos: del sistema real o, en su ausencia, de un modelo del mismo. A partir de la adquisición de datos del sistema real o de la simulación del modelo incluyendo el ruido, sea cual sea el punto de partida, obtenemos una base de datos de trayectorias, las cuales han sido obtenidas simulando fallos de distintos tipos, si partimos de la simulación, o provocando estos fallos en el sistema real si es el caso. Conocemos, por tanto, si cada trayectoria contiene o no fallos, siendo la base de datos obtenida de la siguiente forma:

Trayectoria 1: Dato[1],Dato[2],Dato[3], ...,Dato[m]
Trayectoria 2: Dato[1],Dato[2],Dato[3], ...,Dato[m]
:
Trayectoria n: Dato[1],Dato[2],Dato[3], ...,Dato[m]

Estos datos serán las medidas de las variables observables dentro del sistema.

Estas trayectorias serán definidas como comportamiento en ausencia de fallo mediante la etiqueta *OK* o en presencia de fallo mediante la etiqueta *KO* quedando la base de datos como sigue:

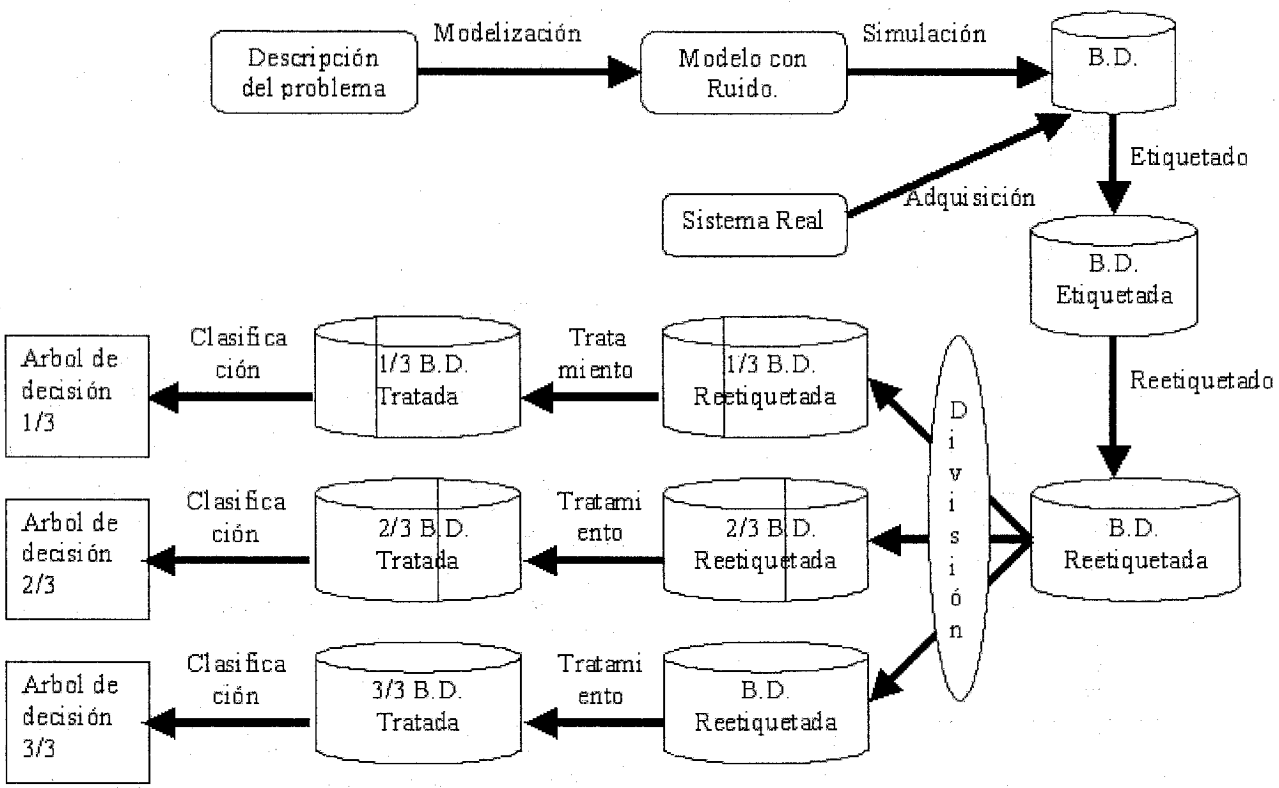

Figura 2: Sistema de detección de fallos.

Trayectoria 1: Dato[1],Dato[2], ...,Dato[m] Etiqueta{OK, KO}
Trayectoria 2: Dato[1],Dato[2], ...,Dato[m] Etiqueta{OK, KO}

Trayectoria n: Dato[1],Dato[2], ...,Dato[m] Etiqueta{OK, KO}

El siguiente paso será dividir la base de datos en tantas partes como puntos de chequeo temporales queramos obtener, en nuestro caso hemos definido tres puntos que nos dividirán la base de datos en tres partes iguales. Estas partes corresponden a un tercio, dos tercios y el total de la trayectoria, de tal forma que podamos utilizar o trabajar con datos muy tempranos, con datos al sesenta y seis por ciento de la transición y con la base de datos completa, como podemos ver en la figura 3.

A cada una de estas porciones de la base de datos original se le realizará un tratamiento, obteniéndose una serie de atributos, que enriquecerán la base de datos para facilitar la clasificación, puesto que los obtenidos directamente del sistema son pobres para realizar el aprendizaje supervisado y los resultados son poco satisfactorios. Estos atributos se describen a continuación:

Distancia al comportamiento perfecto (DP[i]). Indica el grado de similitud entre una trayectoria concreta y la trayectoria del comportamiento perfecto que fue definida con anterioridad. Este atributo se calcula restando el dato de la trayectoria del comportamiento perfecto al dato de la trayectoria en curso en cada instante de tiempo que se obtuvo una medida. Esto podemos observarlo en la ecuación (1).

$$DP(i) = Dato[i] - Datopf[i] \quad (1)$$

Donde *Dato[i]* es el punto tratado de la trayectoria en curso y *Datopf[i]* es el correspondiente punto de la trayectoria del comportamiento perfecto.

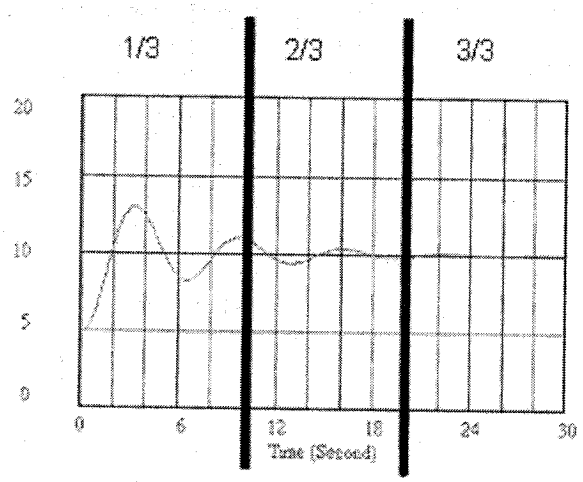

Figura 3: Puntos de chequeo.

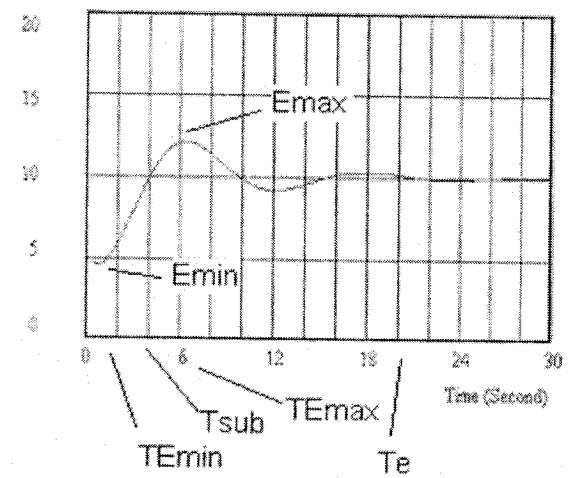

Figura 4: Atributos para cada trayectoria.

Integral(I[i]). Es la magnitud que resulta de la integración numérica entre el punto actual y el precedente, representando esto el área encerrada entre el punto actual y el precedente. Se calcula mediante la aproximación siguiente:

$$I(i) = Ts \times \frac{p[i] + p[i-1]}{2} \qquad (2)$$

Donde *Ts* es el paso de tiempo de la simulación o frecuencia con que se van adquiriendo los datos del sistema real, *p[i]* el punto actual tratado y *p[i-1]* el precedente.

Además calcularemos una serie de atributos para cada trayectoria, es decir, un único atributo para cada una de las trayectorias, intentando aumentar todavía más la información disponible a la hora de clasificar, estos atributos que podemos ver en la figura 4 son:

Tiempo de subida (Tsub). Es el momento en el cual se alcanza por primera vez el punto al que aspiramos llegue y se estabilice el sistema, es decir, es cuando el sistema iguala su estado al indicado por la consigna a alcanzar.

Tiempo de establecimiento (Te). Es el punto temporal en el que sistema alcanza el estado deseado definitivamente, es decir, lo alcanza y permanece en el de forma estable.

Estado máximo. (Emax). Es el estado máximo que el sistema alcanza en el proceso oscilatorio de acercamiento al estado deseado.

Tiempo de estado máximo. (TEmax). Es el momento en el que se alcanza el estado máximo.

Estado mínimo. (Emin). Es el estado mínimo que el sistema alcanza en el proceso oscilatorio de acercamiento al estado deseado.

Tiempo de estado mínimo. (TEmin). Es el momento en el que se alcanza el estado mínimo.

Una vez calculados todos estos atributos, obtenemos tres bases de datos, una por cada punto de detección elegido, es decir, un tercio del transitorio, dos tercios o al final de éste. Dichas bases de datos tendrán la forma siguiente:

Trayectoria 1: Tsub, Te, Emax, TEmax, Emin, TEmin Dato[1],DP[1],I[1],Dato[2],DP[2],I[2],……,Dato[n],DP[n],I[n] {KO, OK}
Trayectoria 2: Tsub, Te, Emax, TEmax, Emin, TEmin Dato[1],DP[1],I[1],Dato[2],DP[2],I[2],……,Dato[n],DP[n],I[n] {KO, OK}

Trayectoria n: Tsub, Te, Emax, TEmax, Emin, TEmin Dato[1],DP[1],I[1],Dato[2],DP[2],I[2],……,Dato[n],DP[n],I[n] {KO, OK}

Tras realizar este tratamiento, sobre cada base de datos, éstas nos servirán de entrada para realizar el aprendizaje supervisado.

El aprendizaje se llevará a cabo mediante una herramienta de clasificación, que en nuestro caso será

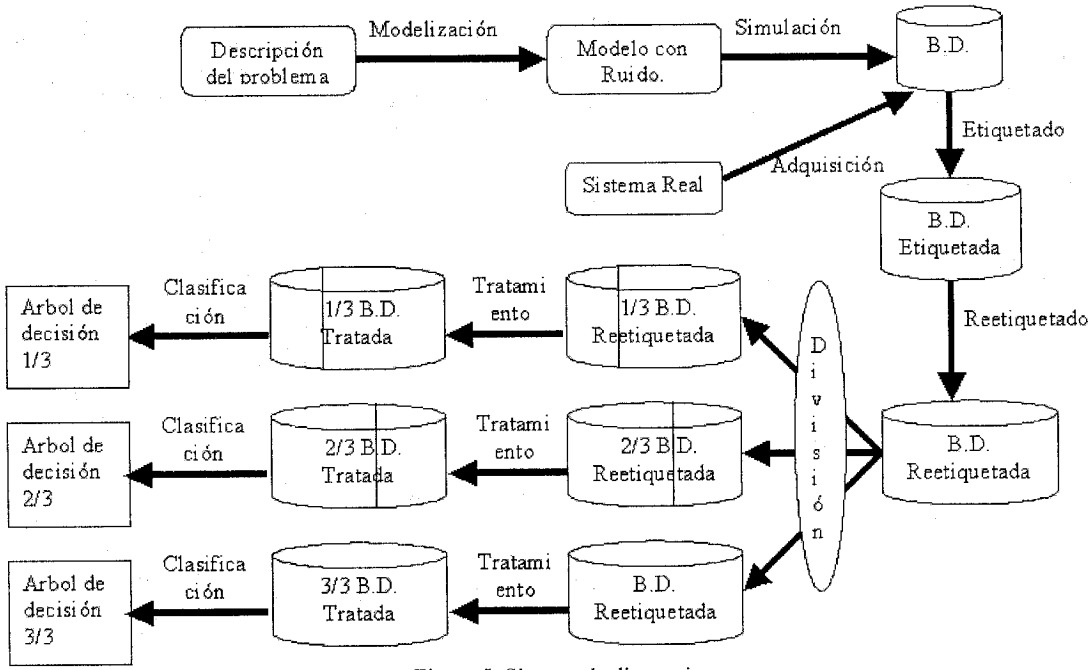

Figura 5: Sistema de diagnosis.

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 dicha 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. La salida del clasificador estará compuesta por un conjunto de reglas de clasificación que serán las que nos sirvan para llevar a cabo la detección. Estas reglas serán del tipo:

SI <Condición_en_atributos>
ENTONCES <clase>

Una vez explicado el tratamiento off-line que se realiza para obtener los árboles de decisión para la detección de errores, pasaremos a describir el tratamiento necesario para la obtención del árbol de diagnosis.

3.2 Sistema de Diagnosis

El sistema de diagnosis tendrá una forma de obtención similar al del sistema de detección, la cual puede intuirse en la figura 5.

Al igual que ocurría con el sistema de obtención del árbol de detección, el sistema de obtención del árbol de diagnosis también se realizará off-line, es decir, todo el tratamiento computacionalmente costoso se realiza antes de la instalación en el sistema real.

El sistema de diagnosis, encaminado a encontrar un árbol de decisión que se utilizará para realizar diagnosis, consistirá en una serie de pasos. De estos, los iniciales son idénticos a los del sistema de detección, partiendo desde el sistema real o en su ausencia de la simulación incluyendo el ruido.

A partir de una de estas opciones obtenemos una base de datos de trayectorias con comportamientos conocidos y cuyas etiquetas representan el tipo de comportamiento. Como podemos ver, los pasos hasta este punto son iguales a los del sistema de detección, con la salvedad que esta base de datos es etiquetada de forma distinta, no se etiquetará como *OK, KO*, si no que se etiquetara según el tipo de fallo al que pertenezca la trayectoria o que la trayectoria está en ausencia de fallo, es decir, comportamiento correcto (*OK*). Obtenemos por tanto una base de datos de la siguiente forma:

Trayectoria 1: Dato[1], ... , Dato[n]. {Error tipo 1}
Trayectoria 2: Dato[1], ... , Dato[n]. {Error tipo 1}
:
Trayectoria n: *Dato[1], ... , Dato[n]. {Error tipo 1}*

Trayectoria 1: Dato[1], ... , Dato[n]. {Error tipo 2}
Trayectoria 2: Dato[1], ... , Dato[n]. {Error tipo 2}
:
Trayectoria n: *Dato[1], ... , Dato[n]. {Error tipo 2}*
:
Trayectoria 1: Dato[1], ... , Dato[n]. {Error tipo n}
Trayectoria 2: Dato[1], ... , Dato[n]. {Error tipo n}
:
Trayectoria n: *Dato[1], ... , Dato[n]. {Error tipo n}*

El siguiente paso es el que denominamos reetiquetado. Este paso tiene lugar debido a que existe trayectorias pertenecientes a familias de comportamientos tan similares que es imposible discernir a que comportamiento pertenece cada una.

Con objeto de solucionar este problema, los conjuntos de trayectorias similares, que pertenezcan a familias de comportamientos distintos, serán reetiquetadas con ambas etiquetas, creando por tanto una nueva familia de comportamientos.

El problema consiste en definir cuando dos o más trayectorias son similares. La decisión que se ha tomado es que dos trayectorias son similares cuando la distancia entre ambas sea menor que una determinada magnitud, que es dependiente del sistema tratado. La distancia que hemos usado es la distancia Euclídea.

Los siguientes dos pasos a realizar son: la división de la base de datos y el tratamiento de los datos, que se realiza en base a enriquecer la base de datos y facilitar la tarea de clasificación, y la clasificación, que ha sido realizada utilizando C4.5 (Quinlan, 93). Dichos pasos son idénticos a los ya expuestos en el sistema de detección de fallos.

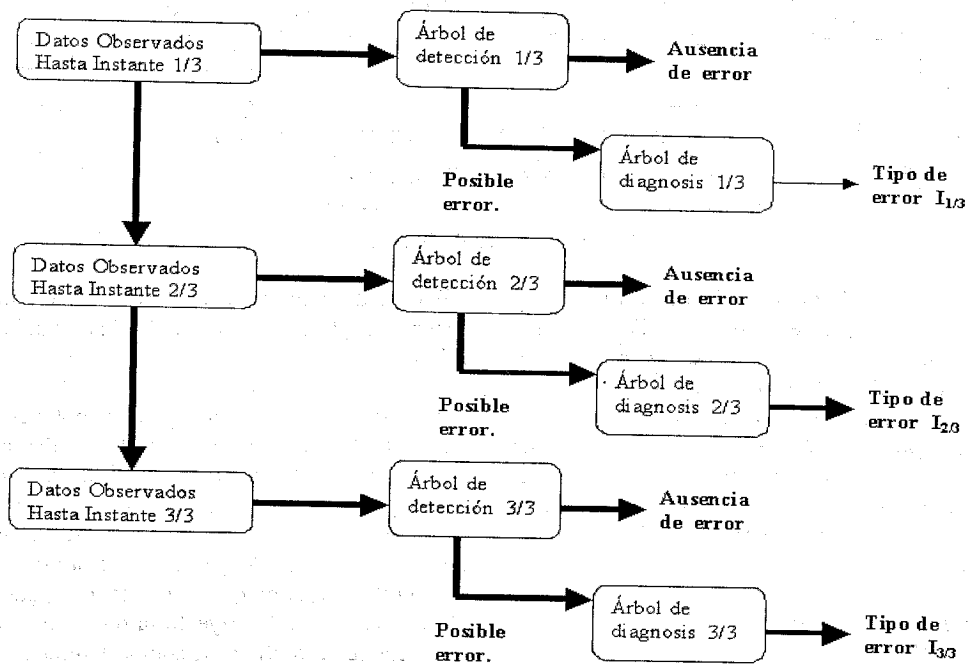

Figura 6: Sistema de detección e identificación de fallo en tiempo real.

Con lo cual se obtiene 3 árboles de decisión para diagnosis uno por cada momento en los que deseamos hacer el diagnostico.

4 Detección y Diagnosis en Tiempo Real

Una vez se ha descrito como se obtienen los árboles de decisión tanto de detección como de diagnosis, vamos a dar la explicación de cómo son utilizados en tiempo real, de lo cual tenemos un esquema en la figura 6.

En el momento que se produce una petición de cambio de consigna, seleccionaremos el conjunto de árboles obtenidos para esa situación concreta y comenzaremos la adquisición de datos del sistema.

El cambio de consigna puede ser origen del funcionamiento cotidiano del sistema o porque introduzcamos pulsos periódicos de control.

Una vez alcanzado el primer punto de diagnosis, en el instante 1/3 del tiempo total utilizado para el entrenamiento, se le pasa esos datos observables al árbol de detección de errores seleccionado previamente, del cual existirá uno diferente para cada cambio de funcionamiento del sistema. Existe en este punto dos alternativas, la primera es que el sistema no detecte ningún error y por lo tanto mostrará mensajes de funcionamiento correcto, la segunda es la detección de un error, en este caso los datos de la observación se pasan a través del árbol de decisión de diagnosis pertinente, mostrándonos éste la predicción del fallo o de la de la familia de fallos detectados.

El método de actuación se repite en el siguiente punto de diagnosis, en el instante 2/3 del tiempo total utilizado para entrenamiento. Es de destacar que el instante de detección y diagnosis es más tardío pero la detección y diagnosis es más fiable, ya que la información de la que se dispone es

mayor, puesto que disponemos de la observada en el instante 1/3 más la observada desde el instante 1/3 hasta el instante 2/3.

De nuevo, el método de actuación se repite para todo el periodo de transición. La información ahora es la del periodo total y por tanto la más enriquecida, con lo cual la detección y diagnosis en este punto será la más fiable.

5 Caso de Estudio

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

El modelo de sistema dinámico considerado se representa en la Figura 7. Consta de un motor eléctrico (M), cuya velocidad angular (W) se controla mediante el voltaje (V) proporcionado por el controlador ©. El controlador © actúa basándose en la velocidad angular deseada (d) y el valor de la velocidad angular medida (W_m) por el sensor (S).

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

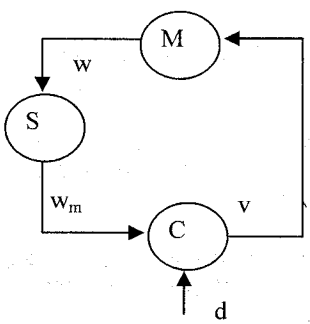

Figura 7: Sistema ejemplo.

Las ecuaciones que rigen el modelo son:

$$Motor: T * \frac{dw}{dt} = c_m * v - w \qquad (3)$$

$$Controlador\ I: \frac{dv}{dt} = c_c * (d - w_m) \qquad (4)$$

$$Sensor: w_m = c_s * w \qquad (5)$$

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 uno de los componentes. Ellas 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, no sólo 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 serán denominadas:

- *CmAlta.* Que la constante del motor (Cm) se sitúe por encima de los valores que consideramos como correctos.
- *CmBaja.* Que la constante del motor (Cm) se sitúe por debajo de los valores que consideramos como correctos.
- *CsAlta.* Que la constante del sensor (Cs) se sitúe por encima de los valores que consideramos como correctos.
- *CsBaja.* Que la constante del sensor (Cs) se sitúe por debajo de los valores que consideramos como correctos.
- *CcAlta.* Que la constante del controlador (Cc) se sitúe por encima de los valores que consideramos como correctos.
- *Ccbaja* Que la constante del controlador (Cc) se sitúe por debajo de los valores que consideramos como correctos.

Para describir el funcionamiento correcto del sistema, vamos a considerar que los valores de las constantes de los componentes podrán oscilar dentro de un rango que consideraremos correcto para cada una de dichas constantes, estos valores serán proporcionados por el fabricante. De esta forma se le permite una flexibilidad de funcionamiento al sistema que se acerca más a lo que será un comportamiento real. Esto dificulta la diagnosis al no tener un valor de referencia fijo y unívoco 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 sufre un cambio de comportamiento, este cambio se supone que ocurre instantáneamente y a partir de aquí no evoluciona en el tiempo.

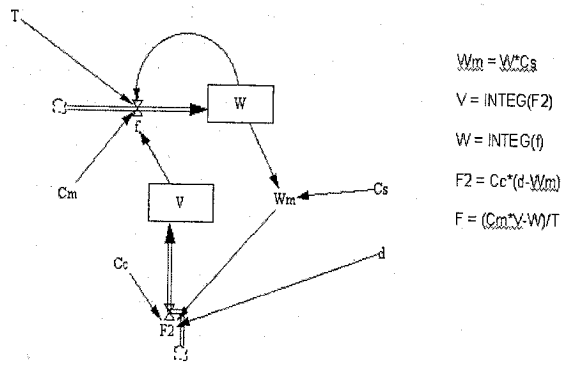

Figura 8: Diagrama de Forrester del modelo.

Apliquemos el funcionamiento de nuestra metodología al citado caso de ejemplo, comenzando con el sistema de detección.
Como hemos dicho con anterioridad se puede partir del modelo real o de una simulación, que es el caso, hemos simulado el sistema en Vemsin® con presencia de un ruido blanco. El sistema simulado en Vemsin queda como se muestra en la figura 8.

Los datos obtenidos tendrán por tanto la forma:

Wm[1], Wm[2], Wm[3], ..., Wm[m]

Las cuales serán etiquetadas como {OK, KO}, el siguiente paso será dividir la base de datos en tres partes conteniendo cada una de ellas un tercio, dos tercios y el total de los datos obtenidos en la simulación obteniendo por tanto 3 bases de datos que quedarán etiquetadas como sigue:

Wm[1], Wm[2], Wm[3], ..., Wm[m]. ETIQUETA(ok, ko)

Una vez calculados los atributos que se describen en la sección 3.1. Obteniendo la siguiente base de datos.

Tsub, Te, Vmax, TVmax, Vmin, TVmin
Wm[1],DP[1],I[1], Wm[2],DP[2],I[2], ……
Wm[n],DP[n],I[n] {OK, KO}

Esta base de datos es utilizada para el aprendizaje supervisado mediante la herramienta C4.5 [Quinlan 93] obteniendo el árbol de decisión que se utilizará para la detección de fallos, obteniéndose reglas del tipo:

SI <Condición_en_atributos>
ENTONCES {OK, KO}

Una vez obtenido el árbol de decisión para la detección se procede a la obtención del árbol de decisión para la diagnosis. Como se ha dicho con anterioridad los pasos iniciales son idénticos con la diferencia que ahora las simulaciones realizadas en presencia de fallo se etiquetarán con la etiqueta correspondiente a la familia de comportamiento en presencia de fallos (*CmAlta, CmBaja, CsAlta, CsBaja, CcAlta, CcBaja*) obteniendo una base de datos como la que se muestra a continuación:

Wm[1], Wm[2], Wm[3], ..., Wm[m] CmAlta
:
Wm[1], Wm[2], Wm[3], ..., Wm[m]. CmBaja
:
Wm[1], Wm[2], Wm[3], ..., Wm[m]. CcAlta
:
Wm[1], Wm[2], Wm[3], ..., Wm[m]. CcBaja
:
Wm[1], Wm[2], Wm[3], ..., Wm[m]. CsAlta
:
Wm[1], Wm[2], Wm[3], ..., Wm[m]. CsBaja
:
Wm[1], Wm[2], Wm[3], ..., Wm[m]. OK
:

El siguiente paso es el reetiquetado que como hemos dicho dependerá de la distancia Euclídea entre las trayectorias.

A continuación se realiza el tratamiento de la base de datos para enriquecerla con los atributos descritos en la sección 3.1, quedándonos por tanto la siguiente base de datos.

Trayectoria 1: Tsub, Te, Wmax, TWmax, Wmin, TWmin
Wm[1],DP[1],I[1], Wm[2],DP[2],I[2], ……
Wm[n],DP[n],I[n] {}*
Trayectoria 2: Tsub, Te, Wmax, TWmax, Wmin, TWmin
Wm[1],DP[1],I[1], Wm[2],DP[2],I[2], ……
Wm[n],DP[n],I[n] {}*
:
Trayectoria n: Tsub, Te, Wmax, TWmax, Wmin, TWmin
Wm[1],DP[1],I[1], Wm[2],DP[2],I[2], ……
Wm[n],DP[n],I[n] {}*

Donde * representa cualquier combinación de errores entre los posibles. Una vez realizado el aprendizaje supervisado sobre esta base de datos obtendremos los tres árboles de decisión para diagnosis descritos con anterioridad, dispuestos para ser aplicados al sistema real.

4.1 Resultados

Para poder validar la metodología hemos realizado nuevas simulaciones con unos parámetros de funcionamiento concretos. De esta forma conocemos de antemano la diagnosis correcta y podemos comprobar la exactitud de los resultados obtenidos mediante los árboles de decisión y diagnosis obtenidos.
Las condiciones de test a las que nos referimos están en la tabla 1.

Tabla 1: Condiciones del test.

T (inercia)	3
D	10
W inicial	5
Paso de tiempo	0.1
Ruido	±5%
Valores para OK	[0.98 - 1.02]
Valores para CmAlta, CsALta, CcAlta	[1.02 - 5]
Valores para CmBaja, CsBaja, CcBaja	[0 – 0.98]

Como podemos ver las constantes de los distintos componentes tienen un rango de tolerancia, lo cual simula con más detalle las 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. La situación que se pretende modelar consiste en que el motor está girando a una velocidad de 5 rad/seg. y se quiere incrementar hasta 10 rad/seg. Como ya indicamos en la metodología debe existir un conjunto de árboles de detección y de diagnosis para cada cambio de consigna en el sistema.

Para este caso se generarán aleatoriamente un conjunto de simulaciones donde todos los valores de las variables se encuentren dentro de los rangos considerados correctos. Esos valores serán elegidos por sorteo, según el método Monte Carlo, siguiendo una 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 la etiqueta, OK que representan la familia de comportamientos correctos para nuestro sistema, como podemos ver en la figura 9.

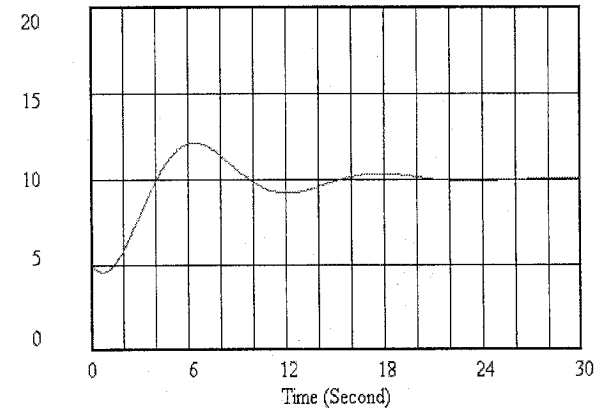

Figura 9: Comportamiento OK.

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. Obtenemos por tanto un total de 700 simulaciones. Podemos ver algún ejemplo de esto en las figuras 10 y 11.

A estas trayectorias se le calcularán los atributos de igual forma que se hizo en el sistema de detección o diagnosis, esto es necesario puesto que en los árboles de decisión se discriminará a veces por estos atributos.

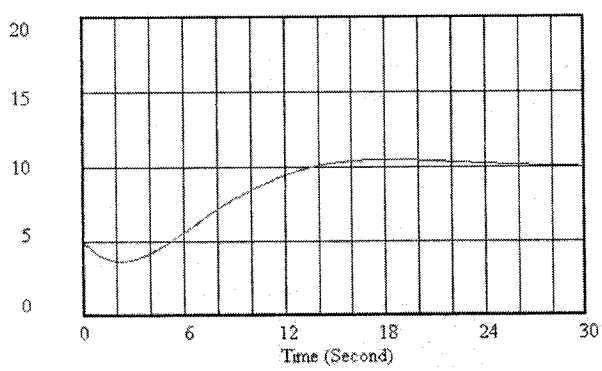

Figura 10: Comportamiento CcBaja.

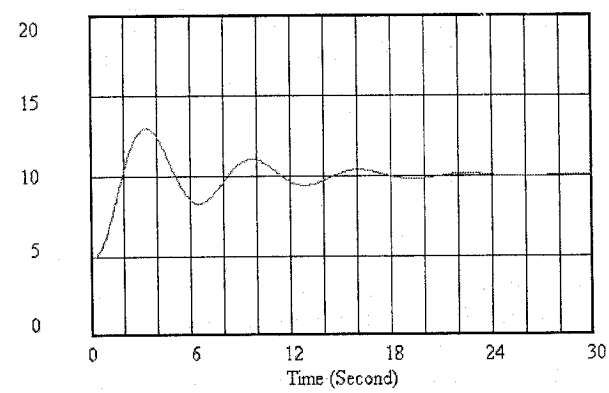

Figura 11: Comportamiento CmAlta.

Tabla 2: Resultados de detección de errores.

Valores de las constantes			Detección Correcta	Detección Instante 1/3	Detección Instante 2/3	Detección Instante 3/3
Cm	Cc	Cs				
1	1	1.07	KO	KO	~~OK~~	~~OK~~
1	1	1.1	KO	KO	KO	KO
1	1	3	KO	KO	KO	KO
1	1.07	1	KO	KO	OK	KO
1	1.1	1	KO	KO	KO	KO
1	3	1	KO	KO	KO	KO
1.07	1	1	KO	KO	KO	KO
1.1	1	1	KO	KO	KO	~~OK~~
3	1	1	KO	KO	KO	KO
1	1	0.93	KO	KO	KO	KO
1	1	0.85	KO	KO	KO	KO
1	1	0.1	KO	KO	KO	KO
1	0.93	1	KO	KO	KO	KO
1	0.85	1	KO	KO	KO	KO
1	0.1	1	KO	KO	KO	KO
0.93	1	1	KO	KO	KO	KO
0.85	1	1	KO	KO	KO	KO
0.1	1	1	KO	KO	KO	KO
0.99	0.98	1.02	OK	KO	OK	OK
1	1.02	1.02	OK	~~KO~~	OK	OK
0.98	1	0.98	OK	KO	OK	OK
0.98	1.02	1.02	OK	~~KO~~	OK	OK
0.99	1.01	1.01	OK	~~KO~~	OK	OK
1.01	1	0.99	OK	~~KO~~	OK	OK

Los resultados tanto para la detección de fallos como para diagnosis de familias de fallos son mostrados en las tablas 2 y 3. Los resultados de la detección de fallos se muestran en la Tabla 2 y en la tabla 3 la diagnosis cuando se detecta el fallo.

Como podemos apreciar en la tabla 2, sólo en dos casos se produce la no detección de errores. Cabe destacar también que los comportamientos OK en el instante primero de detección son reconocidos como posibles fallos, esta tasa baja sólo a 2 de 6 pruebas realizadas en el instante 2/3 y en el último instante la tasa baja a cero. Teniendo en cuenta que en el último instante es cuando se tiene mayor información y por tanto al que le daremos un valor más significativo la tasa de error en la detección es francamente baja.

Tabla 3: Resultados de la diagnosis.

Valores de las constantes			Diagnosis Correcta	Diagnosis Instante 1/3	Diagnosis Instante 2/3	Diagnosis Instante 3/3
Cm	Cc	Cs				
1	1	1.07	CsAlta	CsAlta, CmAlta, CcAlta	CsAlta, CmAlta, CcAlta	CmAlta, CcAlta
1	1	1.1	CsAlta	CsAlta, CmAlta, CcAlta	CsAlta, CmAlta, CcAlta	CsAlta, CmAlta, CcAlta
1	1	3	CsAlta	CsAlta	CsAlta	CsAlta
1	1.07	1	CcAlta	Todos	CsAlta, CmAlta, CcAlta	CmAlta, CcAlta
1	1.1	1	CcAlta	CsAlta, CmAlta, CcAlta	CmAlta, CcAlta	CmAlta, CcAlta
1	3	1	CcAlta	CmAlta, CcAlta	CsAlta, CmAlta, CcAlta	CsAlta, CmAlta, CcAlta
1.07	1	1	CmAlta	CsAlta, CmAlta, CcAlta	CmAlta, CsAlta	CmAlta, CsAlta
1.1	1	1	CmAlta	CsAlta, CmAlta, CcAlta	CmAlta, CsAlta	CsAlta
3	1	1	CmAlta	CmAlta, CsAlta	CmAlta	CmAlta
1	1	0.93	CsBaja	CsBaja, CmBaja,CcBaja	CsBaja, CmBaja,CcBaja	CsBaja, CmBaja,CcBaja
1	1	0.85	CsBaja	CsBaja, CmBaja,CcBaja	CsBaja, CmBaja,CcBaja	CsBaja
1	1	0.1	CsBaja	CsBaja	CsBaja	CsBaja
1	0.93	1	CcBaja	CsBaja, CmBaja,CcBaja	CsBaja, CmBaja,CcBaja	CmBaja,CcBaja
1	0.85	1	CcBaja	CsBaja, CmBaja,CcBaja	CmBaja,CcBaja	CmBaja,CcBaja
1	0.1	1	CcBaja	CmBaja,CcBaja	CcBaja	CcBaja
0.93	1	1	CmBaja	CsBaja, CmBaja,CcBaja	CsBaja, CmBaja,CcAlta, CsAlta, CmAlta	CsBaja, CmBaja,CcBaja
0.85	1	1	CmBaja	CsBaja, CmBaja,CcBaja	CmBaja,CcBaja	CmBaja,CcBaja
0.1	1	1	CmBaja	CmBaja,CcBaja	CmBaja	CmBaja

Por otra parte en la tabla 3 (la de identificación de fallos) observamos que en el instante 3, excepto en los dos casos en los que no se obtiene una detección del error, que por otro lado jamás podría haber sido diagnosticado puesto que no fue detectado, en el resto de los casos probados la identificación del error es absoluta. Es de reconocer que en el instante 2/3 y en el instante 1/3 aunque se identifica el error, no se identifica sólo a éste, sino a una familia de fallos, la cual va decreciendo al avanzar los instantes de identificación siguientes.

5 Conclusiones y Trabajos Futuros

La metodología presentada lleva a cabo diagnosis en sistemas dinámicos independientemente del tipo de sistema de que se trate y aislándonos 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 observado 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 sólo 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 posibilita 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 cual 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.

6 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

Pedro J. Abad, Antonio J. Suárez, Rafael M. Gasca y Juan A. Ortega. *Using Supervised Learning Techniques for Diagnosis of Dynamic Systems*. In Proceedings of the DX'02 (13th International Workshop on Principles of Diagnosis).

K. Balakrishnan and V. Jonavar. *Intelligent diagnosis systems.* Journal of Intelligent Systems, 8, 1998.

O. Dressler. *On-Line Diagnosis and Monitoring of Dynamic Systems based on Qualitative Models and Dependency-recording Diagnosis Engines*. Proc. 12th. European Conference on Artificial Intelligence(ECAI-96) 461-465, 1996.

Fuente, M. J. *Detección y diagnóstico de fallos usando redes neuronales*. Diagnosis, Razonamiento Cualitativo y Sistemas Socioeconómicos. Carlos Alonso y Juan Antonio Ortega, Editores. 2001

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

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. !4Th European Conference on Artificial Intelligent. 2000

A.D. Pouliezos y G.S. Stavrakakis. *Real time fault monitoring of industrial process.* Microprocessor-based systems engineering. Kluwer Academic Publishers, Dordrecht, 1994.

J. Ross Quinlan. *Induction of decision trees.* Machine learning, 1986

J. Ross Quinlan. *C45:Program for Machine Learning.* Morgan Kaufman, 1993

Juan J. Rodríguez, Carlos J. Alonso y Q. Isaac Moro. *Clasificación de patrones temporales en sistemas dinámicos mediante Boosting y Alineamiento dinámico temporal.* In proceedings of the I Jornadas de Trabajo sobre Diagnosis. Valladolid 2001.

Saludes, A. Vargas y J. R. Perán. *Aplicación de la red neuronal SOM para la detección de fallos desconocidos en un grupo hidroeléctrico.* In proceedings of the I Jornadas de Trabajo sobre Diagnosis. Valladolid 2001.

A. Simón, L. Alonso & A. Antón. *Sistema Híbrido Borroso para la ayuda del Diagnóstico del Glaucoma.* I Jornadas de Trabajo sobre Diagnosis. Valladolid 2001.

P. Struss. *Fundamentals of model-based diagnosis of dynamic systems*. Proc. IJCAI'97. 1997.

Antonio J. Suárez, Pedro J. Abad, Rafael M. Gasca y Juan A. Ortega. *Aplicación de Técnicas de Aprendizaje a la diagnosis de Sistemas Dinámicos con Etiquetado Múltiple*. In proceedings of the IX CAEPIA –TTIA. 2001.

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, y en 1999 obtuvo el grado de ingeniero en Informática por la misma Universidad. Desde 1992 viene trabajando en el Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática de la Universidad de Huelva. Actualmente realiza su Tesis Doctoral en el campo de la Diagnosis Automática.*

Antonio J. Suárez Fabrega, *recibió el Diplomado en Informática en 1991 en la Universidad de Sevilla, España, y en 1999 obtuvo el grado de ingeniero en Informática por la misma Universidad. Desde 1992 viene trabajando en el Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática de la Universidad de Huelva. Actualmente realiza su Tesis Doctoral en el campo del Razonamiento Cualitativo..*

Juan Antonio Ortega, *nacido en 1968 es Ingeniero en Informática desde 1992 y Doctor en Informática desde el año 2000, ambos títulos obtenidos en la Universidad de Sevilla (España). Es profesor del Departamento de Lenguajes y Sistemas Informáticos de la citada Universidad desde el año 1992. Su principal campo de investigación se centra en la patrones cualitativos de sistemas dinámicos, diagnosis y domótica.*

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