

## Reconstrucción de la dinámica de los tiempos de ejecución de tareas en tiempo real empleando filtrado digital difuso

### Reconstruction of the execution times dynamics of real-time tasks by fuzzy digital filtering

*Jorge Salvador Valdez Martínez\**, *Gustavo Delgado Reyes*, *Pedro Guevara López*, *Juan Carlos García Infante*

Sección de Estudios de Posgrado e Investigación. Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad “Culhuacán”. Instituto Politécnico Nacional. Av. Santa Ana No. 1000 Col. San Francisco Culhuacán del Coyoacán, México D.F., México. C.P. 04430.

(Recibido en 13 de diciembre de 2012. Aceptado el 15 de octubre de 2013)

#### Resumen

Los sistemas en tiempo real (STR) procesan sus actividades a través de tareas que a su vez están formadas por un conjunto de instancias, las cuales tienen seis restricciones temporales: tiempo de arribo, tiempo de inicio, tiempo de ejecución, tiempo de finalizado, tiempo excedente y plazo. De estas seis restricciones, el tiempo de ejecución depende directamente del software y hardware de la computadora, originando que la variación de esta restricción temporal esté vinculada con diversos factores externos e internos, esta variación puede ocasionar que algunas instancias no cumplan su plazo; por este motivo es importante proponer un modelo para reconstruir el comportamiento de los tiempos de ejecución, con el fin de: dimensionar de manera correcta el sistema en tiempo real, proponer técnicas adecuadas de tolerancia a fallos y mejorar sus módulos de operación. Por lo cual en este trabajo, se propone un modelo para reconstruir los tiempos de ejecución obtenidos de la medición hecha a un algoritmo de programación elaborado en el sistema operativo en tiempo real QNX Neutrino 6.5. Dicha reconstrucción se hace a través de un modelo tipo autoregresivo de promedios móviles integrado con un estimador de parámetros construido con un procedimiento de filtrado digital difuso (FDD). Se usa el error cuadrático medio para validar la reconstrucción, el cual converge a una región cercana a cero y permite concluir que la reconstrucción es buena.

----- **Palabras clave:** Tarea en tiempo real, Tiempo de ejecución, Estimador, Filtro difuso, Reconstrucción

---

\* Autor en correspondencia: teléfono: + 52 + 55 + 56242000 ext. 73253, correo electrónico: jsvaldezmtz@yahoo.com.mx (J. Valdez)

## Abstract

Real-time systems (RTS) process their activities through tasks which in turn consist of a set of instances, each one of this real-time task (RTT) have six temporary constraints: arrival time, start time, execution time, end time, over time and deadline. Of these six constraints, the execution time depends directly on the software and hardware of the computer, because of that this temporary constraints varies due to external and internal factors, this variation may cause at least some instance does not satisfy the deadline, in this sense it is important to propose a model to reconstruct the behavior of the execution times for the purpose to determine the proportions of the Real-time System, propose appropriate techniques of fault tolerance and improve operating modules. Therefore in this paper, is proposed a model to reconstruct the execution times from measurements made at a program algorithm developed in the real-time operating system QNX Neutrino 6.5. This reconstruction is done by a model type autoregressive–moving-average integrated with parameter estimator constructed with a digital filtering procedure diffuse (FDD). To validate the reconstruction is used the mean square error which converges to a region close to zero and indicate that the reconstruction is good.

----- **Keywords:** Real-time task, execution time, estimator, fuzzy filtering, reconstruction

## Introducción

De acuerdo con [1], un sistema en tiempo real implantado en un sistema operativo como QNX Neutrino está compuesto por un conjunto de tareas en tiempo real (TTR)  $\{J_i\}$ , donde cada tarea está formada por un conjunto de instancias  $\{j_{i,k}\}$ , las que a su vez están formadas por un conjunto de restricciones temporales de las cuales la más significativa es el tiempo de ejecución  $c_{i,k}$  pues el tiempo de respuesta del sistema en tiempo real depende en gran parte del comportamiento de los tiempos de ejecución de cada instancia de cada tarea, por esa razón modelar las instancias y las tareas no es sencillo ya que cada una de ellas necesita una representación de acuerdo a sus propias dinámicas; de esta manera es necesaria la medición y estimación de los tiempos de ejecución  $c_{i,k}$ . Anteriormente los modelos estaban limitados al solo considerar a los tiempos de ejecución fijos [2]; sin embargo las variaciones de los tiempos de ejecución originan divergencias entre lo identificado y lo que ocurría en el sistema de cómputo; dichas variaciones están relacionadas

con factores internos y externos del equipo de cómputo tal y como son el *caching*, *pipeline*, la búsqueda de la ruta de ejecución considerando a la exclusión mutua, número de lazos, predicciones, planificación y otras interacciones [3, 4]; sin embargo, esta variación hace que los Sistemas en Tiempo Real puedan no cumplir con sus plazos  $d_{i,k}$  provocando que sean poco predecibles, mal dimensionados o poco tolerantes a fallos. Para dar solución a este problema se ha planteado reconstruir el comportamiento de la dinámica de los tiempos de ejecución; diversos autores han planteado varias técnicas, algunas basadas en filtrado digital como la utilizada en [5] donde se presentó un modelo estocástico autoregresivo de promedios móviles (ARMA) de orden (1,1) junto con la estimación de los parámetros del modelo utilizando un filtro digital. Una alternativa a esta técnica es la presentada en este trabajo, donde el procesamiento difuso resulta de gran utilidad para aplicaciones en donde se deba obtener una descripción del sistema en base a sus restricciones en tiempo real, ya que a partir de este método

algorítmico con propiedades difusas es posible obtener el comportamiento de un sistema en tiempo real en base a sus restricciones de tiempo medibles. Esto permitirá obtener un mayor uso de los recursos, mejor rendimiento de los procesos de control, ahorro de energía y puede permitir la adaptación a entornos dinámicos así como de situaciones de sobrecarga. Permitiendo obtener información adicional acerca del comportamiento del sistema, márgenes de planificación y variabilidad. Además se pueden obtener mejores resultados en el modelado de las instancias o de las tareas en tiempo real, incluso en el diseño de criterios para asignar tareas en tiempo real a sistemas computacionales en sistemas con tolerancia a fallos.

### Desarrollo

#### Tiempos de ejecución para Tareas en Tiempo Real (TTR): Descripción y Modelado

Una TTR  $J_i$  se describe en [1] como una entidad ejecutable de trabajo que al menos es caracterizada por un tiempo de arribo y una restricción temporal asociada a un conjunto de instancias  $\{j_{i,k}\}$ , donde  $i$  es el índice de la tarea,  $k$  representa el índice de la instancia que debe ser ejecutada dentro del plazo máximo asociado a un intervalo de tiempo de acuerdo con el criterio de Nyquist definido en [6]. Cada instancia  $j_{i,k}$  está definida en (1) por la siguiente séxtupla [5]:

$$(l_{i,k}, s_{i,k}, c_{i,k}, f_{i,k}, r_{i,k}, d_{i,k} \mid i, k \in Z^+) \quad (1)$$

Donde  $l_{i,k}$  es el tiempo de arribo,  $s_{i,k}$  es el tiempo de inicio,  $c_{i,k}$  es el tiempo de ejecución,  $f_{i,k}$  es el tiempo de finalizado,  $r_{i,k}$  es el tiempo excedente y  $d_{i,k}$  es el plazo máximo, todos en unidades temporales (U. T.). Estas restricciones se aprecian en la figura 1.

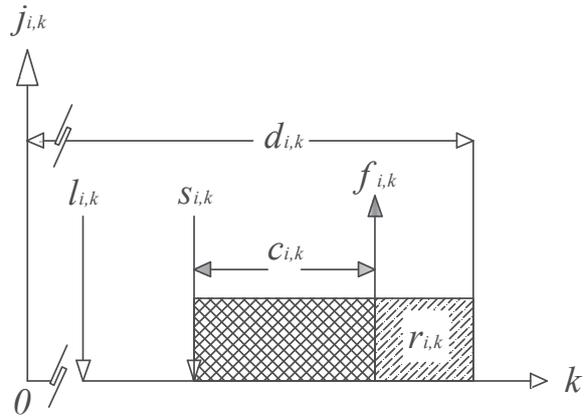


Figura 1 Restricciones temporales para una tarea en tiempo real  $J_i$

El tiempo de ejecución  $c_{i,k}$  es el tiempo en que la instancia con índice  $k$  de una TTR  $J_i$  lleva a cabo sus operaciones, sin considerar los desajustes de ésta en el procesador, puede variar en cada índice  $k$  debido a los factores mencionados en [3] y [4]. En la figura 2, se puede observar la variación del tiempo de ejecución  $c_{i,k}$  debido a los desajustes provocados por otras tareas.

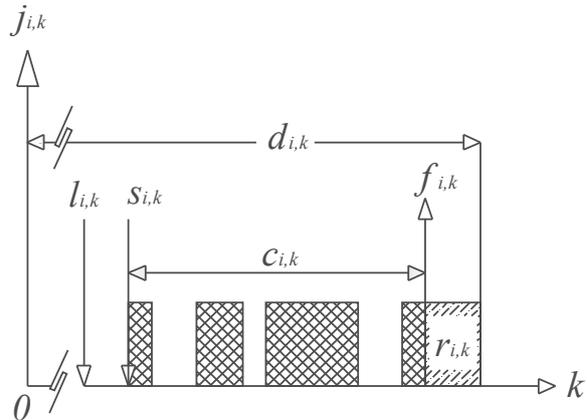


Figura 2 Tiempos de ejecución  $c_{i,k}$  para  $n$  instancias de una tarea de tiempo real

Para el modelado del tiempo de arribo  $l_{i,k}$  de las instancias de una tarea en tiempo real, en [7] propusieron un modelo estocástico autoregresivo de promedios móviles (ARMA) de orden (1,1) para determinar su comportamiento y estimar

los parámetros del modelo utilizando un filtro digital. Para el modelado del tiempo de ejecución  $c_{i,k}$  de las instancias de una TTR  $J_i$  se han encontrado referencias bibliográficas en donde lo consideran fijo y como el peor caso de tiempos de ejecución (*WCET* por sus siglas en inglés) tal y como lo expresaron [3] y [4]. Los autores de [8] describieron a los tiempos de ejecución  $c_{i,k}$  como periodos de duración de una TTR variantes con comportamiento probabilístico.

Entonces, tomando en consideración a [8], los autores de [5] utilizaron el modelo ARMA de orden (1,1) presentado en [7] para modelar y reconstruir a los tiempos de ejecución de tres tareas en tiempo real, el cual se muestra a continuación en (2) y (3).

$$x_{i,k+1} = \hat{a}_{i,k} x_{i,k} + v_{i,k} \quad (2)$$

$$c_{i,k} = x_{i,k} + h_{i,k} + w_{i,k} \quad (3)$$

En donde  $c_{i,k}$  es el tiempo de ejecución de la  $k$ -ésima instancia de la  $i$ -ésima tarea en tiempo real,  $\hat{a}_{i,k}$  es el parámetro del sistema,  $v_{i,k}$  es el ruido interno de la computadora,  $w_{i,k}$  es el ruido externo a la computadora,  $h_{i,k}$  es el tiempo de ejecución de referencia y  $x_{i,k}$  es el estado interno del modelo. Para el caso del trabajo de [5],  $\hat{a}_{i,k}$  fue estimado usando un filtro digital basado en la variable instrumental, con lo cual se obtuvieron resultados favorables, sin embargo en este trabajo se propone usar filtrado difuso para realizar el proceso de estimación del parámetro del sistema  $\hat{a}_{i,k}$ .

### **Reconstrucción de la Dinámica de los Tiempos de Ejecución de TTR usando Filtrado Difuso**

En la bibliografía se pudieron encontrar algunos ejemplos de aplicación de lógica difusa a otras áreas de investigación. Por ejemplo en [9], se propuso la aplicación de la Lógica Difusa en el reconocimiento de patrones de señales en la medicina (caso específico de monitoreo de señales cardiovasculares). Mientras que en [10], trataron de evaluar y controlar los estados cognitivos

y psicológicos de un usuario en un sistema de tutoría multimedia inteligente (Intelligent Multimedia Tutoring System IMTS) al utilizar algebra difusa. Otro ejemplo de aplicación es mostrado en [11], en donde se propuso una red neurodifusa para la clasificación de documentos emulando características de la percepción Visual humana. Otra aplicación del mismo autor es la mostrada en [12], en donde se presentó un nuevo enfoque basado en lógica difusa para determinar la complejidad de una imagen mediante el análisis de los porcentajes de niveles de bordes de la imagen. En el trabajo de [13] se mostró otra aplicación enfocada a la representación grafica de modelos complejos de interrelaciones entre conceptos (mapas cognitivos difusos) basada en el modelo de redes neuronales y así describir implicaciones en el modelado de procesos dinámicos. En [14] se abordó un conjunto de algoritmos para la simulación de la locomoción libre de un robot de 6 patas, en donde la toma de decisiones es hecha a partir de técnicas de lógica difusa, encontrándolo adecuado para lograr el control en la simulación del movimiento del robot. En la aplicación basada en Lógica Difusa de [15] se replica la comprobación de la adecuación de Modelos de estimación del esfuerzo de desarrollo de software personal y lo comparan contra el modelo de estimación basado en regresión lineal encontrando que el desarrollado bajo el método difuso es ligeramente mejor. En el resumen de tesis indicado en [16] se presentó una variante de filtrado digital en tiempo real utilizando Lógica Difusa para la estimación de parámetros a partir del modelo del filtro de Kalman y se concluye que su desarrollo ofrece una solución más robusta para diversos sistemas, da una mayor capacidad de servicios y funcionalidades, menor uso de procesamiento, reducción de tamaño de sistemas con múltiples funciones así como el procesamiento basado en la evolución natural del sistema de referencia. El filtrado difuso cuenta con la capacidad de interpretar dinámicamente las variables de un sistema, así como sus respectivas variaciones, y a partir de esta interpretación dar una respuesta natural de acuerdo a los requerimientos de un proceso,

garantizando el funcionamiento adecuado del sistema [17]. Por este motivo en este trabajo se propone un modelo estocástico autorregresivo de promedios móviles (ARMA) de orden (1,1) que se integró con la estructura del filtro difuso para reconstruir el comportamiento de la dinámica del tiempo de ejecución  $c_{i,k}$ .

El filtro difuso es un filtro de tipo adaptativo con una estructura que clasifica sus respuestas en diferentes niveles de operación para la selección del parámetro más adecuado descrito como  $\hat{a}_{i,k}$  de una base de conocimiento, utilizando para ello los conectores lógicos si-entonces [18-19]. Esto con la finalidad de actualizar los pesos del filtro para describir por medio de un modelo ARMA a la dinámica de una TTR. Siendo el problema de este trabajo la estimación del parámetro  $\hat{a}_{i,k}$  de la ecuación (3). Las funciones de membresía de la base del conocimiento del filtro difuso, están limitadas por el error cuadrático medio descrito en la ecuación (4) con la finalidad de disminuir el error en la estimación del parámetro y obtener su desempeño de acuerdo a la aproximación del parámetro para cada instancia [20, 21]:

$$G_{i,k} = E\{(e_{i,k})^2\}^{1/2} \quad (4)$$

Donde el error  $e_{i,k}$  en la estimación del parámetro se muestra en (5):

$$e_{i,k} = \hat{a}_{i,k} - a_{i,k} \quad (5)$$

Y la aproximación del parámetro para cada instancia es mostrada en (6):

$$\hat{a}_{i,k} \xrightarrow[k \rightarrow \delta]{} a \quad (6)$$

Donde  $\delta$  es el número total de instancias de un proceso. El objetivo en la selección de los parámetros es obtener una disminución dinámica entre los tiempos de ejecución reales  $c_{i,k}$  del sistema en tiempo real y la reconstrucción de los tiempos  $\hat{c}_{i,k}$  por el modelo ARMA. Los diferentes niveles de operación dentro del filtro difuso deben estar dentro del criterio de error descrito en (4) de la siguiente forma mostrada en (7):

$$G_{i,k} = \left[ \frac{1}{k} \left( (k-1)G_{i,k-1} + e_{i,k}^2 \right) \right]^{1/2} \quad (7)$$

Para extraer los parámetros de forma difusa se emplea el mecanismo de inferencia de la figura 3, con los conectores *si-entonces*, para seleccionar el valor más adecuado de una base de conocimiento (BC) dinámicamente.

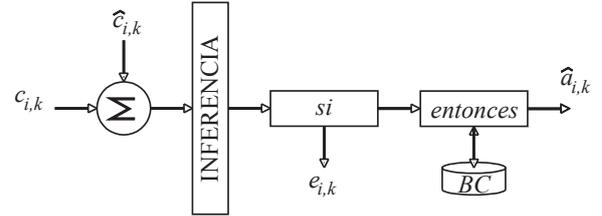


Figura 3 Mecanismo de inferencia difuso

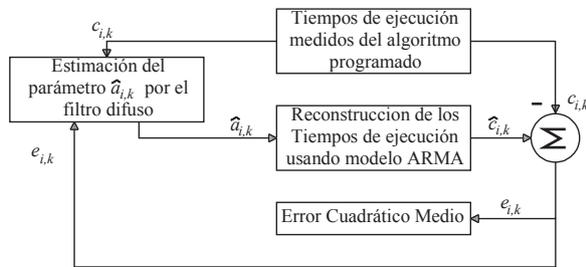
El proceso difuso con base en el criterio de  $\lim_{k \rightarrow N_i} G_{i,k} \rightarrow G_{i, min}$  con  $i, k, N \in \mathbb{N}^+$ , realiza la selección del parámetro en la base de conocimiento y permite aproximar la salida del modelo ARMA  $\hat{c}_{i,k}$  a los tiempos de ejecución  $c_{i,k}$ . En el procesamiento difuso para realizar la selección de los parámetros de manera heurística, en base a las propiedades probabilísticas del sistema, se establecen sus valores dentro del error cuadrático medio, en donde a partir del rango  $G_{i,k} \subseteq [0, \varepsilon)$  con  $\varepsilon$  como radio de  $N$ , el filtro tiene tres niveles de operación. Para cada nivel, es seleccionado un valor específico de  $\hat{a}_{i,k}$ , logrando obtener de esta forma la mejor aproximación de  $\hat{c}_{i,k} \cong c_{i,k}$ .

Para realizar la selección de parámetros de la base de conocimientos se toma en cuenta la respuesta del error cuadrático medio al elegir un valor de parámetro; cuando el error cuadrático llega a un valor mínimo se obtiene la aproximación más cercana de  $\hat{c}_{i,k}$  hasta  $c_{i,k}$ , donde existe la mínima diferencia entre ambos, teniendo un mínimo error de aproximación entre la dinámica de los tiempos de una tarea y la salida del modelo del filtrado difuso [17, 21].

## Resultados

Para el desarrollo de esta sección primeramente se pretende obtener las medidas de los tiempos

de ejecución de un algoritmo de programación, para después estimar el parámetro  $\hat{a}_{i,k}$  usando filtrado difuso y aplicarlo al modelo autoregresivo de promedios móviles ARMA de orden (1, 1), para obtener la reconstrucción de los tiempos de ejecución. Se usará el error cuadrático medio como una medida descriptiva de la convergencia del filtro en función de la estimación del parámetro  $\hat{a}_{i,k}$  para el filtro difuso y en función de la reconstrucción de los tiempos de ejecución. Este procedimiento se puede apreciar en la figura 4.



**Figura 4** Descripción gráfica del proceso de estimación del parámetro y reconstrucción de los tiempos de ejecución

**Medición de los tiempos de ejecución  $c_{i,k}$  del algoritmo programado**

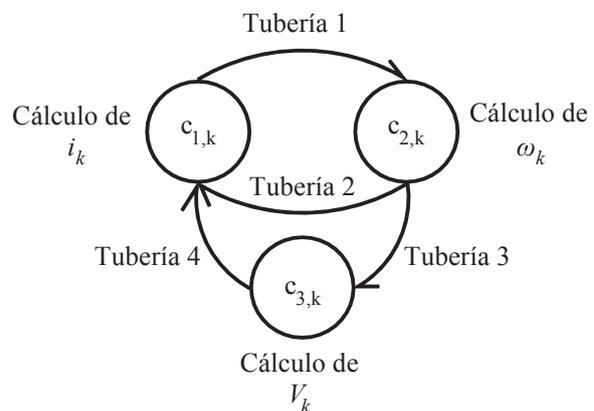
Para esta sección, los tiempos de ejecución  $c_{i,k}$  que fueron medidos experimentalmente en el sistema operativo de tiempo real QNX Neutrino 6.5 sobre un procesador Intel Centrino Dúo de 2,53 GHz, fueron los de la simulación de un motor de corriente continua descrito por las ecuaciones (8), (9) y (10) obtenidas a partir del modelo mostrado en [22], donde  $i_k$  es la corriente de armadura en [A],  $w_k$  es la velocidad angular en [rad/seg],  $R_a$  es la resistencia de armadura en [Ω],  $L_a$  es la inductancia de armadura en [Hy],  $k_a$  es la constante de construcción del motor,  $i_f$  es la corriente de campo en [A],  $b$  es la constante de viscosidad de fricción,  $V_k$  es el voltaje de corrección en [Volts],  $T_D$  es el par debido a la carga del motor en [Nm] y  $k$  la ganancia en el lazo de retroalimentación que corresponde a un controlador proporcional.

$$i_k = \frac{-k_a i_f w_k + V_k + i_{k-1} L_a}{L_a + R_a} \tag{8}$$

$$w_k = \frac{k_a i_f i_k + T_D + w_{k-1} J}{J + b} \tag{9}$$

$$V_k = V_{ref} - k w_k \tag{10}$$

Para ello, en el sistema operativo QNX se programaron 3 procesos concurrentes para la simulación de la dinámica del motor, una TTR designada como  $J_1$  estuvo a cargo del cálculo de la corriente de armadura  $i_k$ , una TTR  $J_2$  a cargo del cálculo de la velocidad angular  $w_k$  y una tercera TTR  $J_3$  a cargo del Voltaje de corrección (controlador proporcional)  $V_k$ . La comunicación entre tareas se realizó usando la lectura y escritura de tuberías FIFO de acuerdo al esquema de comunicación entre procesos concurrentes que se observa en la figura 5 con  $l_{i,k}$  creciente monótonicamente, es decir con  $l_{i,k} = T \pm l_{i,k-1}$ , con  $T = 10$  mseg, programándose T con un *temporizador de tiempo real* a través de las funciones `timer_create()` y `timer_settime()` propias de la biblioteca `timer.h` de QNX Neutrino. Para la planificación de las tareas se utilizó el *planificador adaptivo* con prioridades bajas de nivel 1 [23].



**Figura 5** Esquema de TTR concurrentes y su mecanismo de comunicación en el Sistema Operativo de Tiempo Real QNX

En lo que respecta a la experimentación realizada, se midieron los tiempos de ejecución de 1000 instancias

de las tareas  $J_1, J_2$  y  $J_3$ , haciendo uso de la función `clock_gettime()`, la cual es una herramienta presente en la biblioteca `time.h` de los sistemas operativos basados en POSIX usada para obtener mediciones con en el rango de los nanosegundos, cuya forma de aplicación es mostrada en el segmento de código mostrado en la figura 6.

```

/* Programa para medir tiempos de ejecución */
/* de las tres tareas (cálculos) usando */
/* la función clock_gettime() */

clock_gettime(CLOCK_REALTIME, &inicio);

/* Algoritmo o cálculo a medir */


/* Cálculo de J1 */ /* Cálculo de J2 */ /* Cálculo de J3


clock_gettime(CLOCK_REALTIME, &fin);

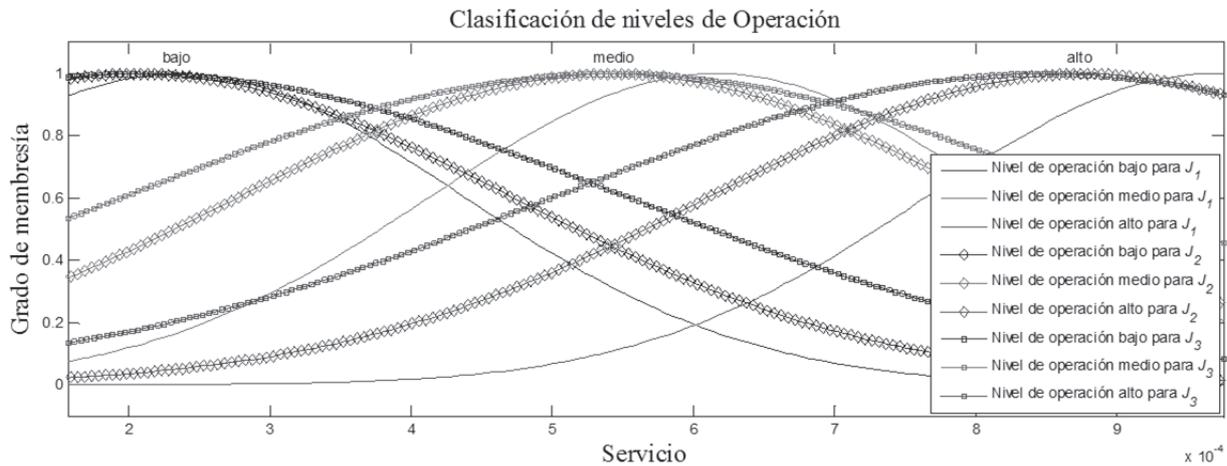
/* Valor medido obtenido */
acu1 = ( fin.tv_sec - inicio.tv_sec );
acu2 = (double)( fin.tv_nsec - inicio.tv_nsec )/
(double)BILLION;
acu = acu1+acu2;
printf("Ci,k es: %lf\n", acu);

```

**Figura 6** Código para la medición de los tiempos de ejecución usando la función `clock_gettime()`

### Estimación del parámetro difuso $\hat{a}_{i,k}$ mediante el empleo del filtrado difuso

En el filtro difuso son empleados tres niveles de operación representados por funciones de tipo gaussiano, ya que de acuerdo con [21], por medio de ellas se tiene una mejor aproximación a un proceso de referencia en comparación con otro tipo de funciones como las triangulares. En la figura 7 se presentan los niveles de operación (bajo, medio y alto) para las TTR  $J_1, J_2$  y  $J_3$  que permiten obtener la estimación del parámetro  $\hat{a}_{i,k}$ . El filtro difuso va seleccionando de manera dinámica los parámetros que ajustan al modelo ARMA y de esta forma lograr la mejor aproximación a los tiempos de ejecución de acuerdo a [20].

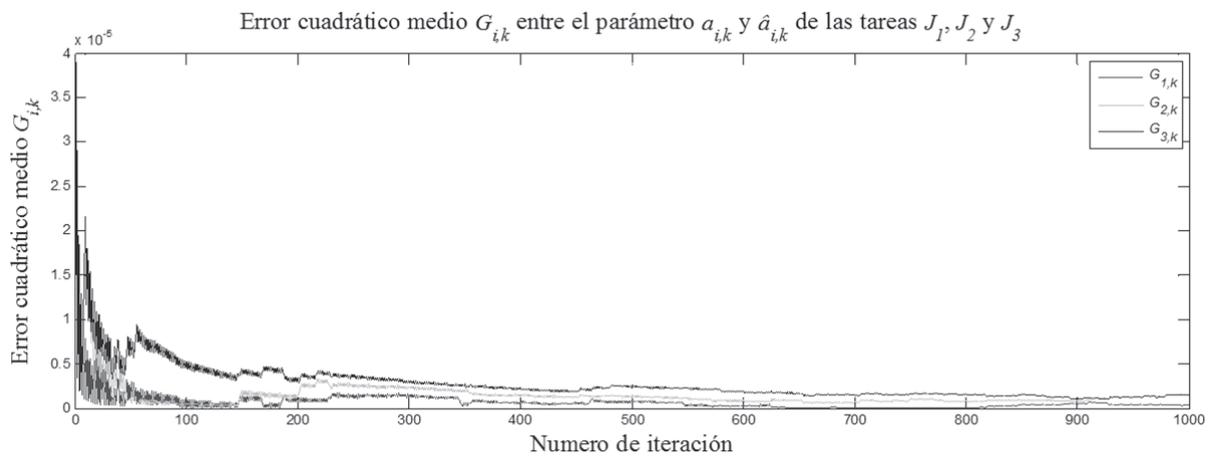


**Figura 7** Clasificación de los niveles de operación para las TTR  $J_1, J_2$  y  $J_3$  por el Filtro Difuso

**Error cuadrático medio  $G_{i,k}$  de la estimación del parámetro  $\hat{a}_{i,k}$**

Por medio del error cuadrático medio se describirá la convergencia del filtro hasta encontrar el mínimo error de  $G_{i,k}$  entre el parámetro del sistema  $a_{i,k}$  y el parámetro estimado del sistema reconstruido  $\hat{a}_{i,k}$  para las tres tareas  $J_1$ ,  $J_2$  y  $J_3$ . En la figura 8 se aprecia el comportamiento asintótico de las curvas  $G_{i,k}$  y su convergencia a un valor muy cercano a

ceros, demostrando que el filtro difuso funciona de manera adecuada. En los tres casos pertenecientes a las tres tareas en tiempo real, el error entre el parámetro del sistema  $a_{i,k}$  y el parámetro estimado del sistema reconstruido  $\hat{a}_{i,k}$  es de aproximadamente  $1 \times 10^{-3}$  [Segundos]; esto muestra que la reconstrucción a través del filtro difuso es satisfactoria y asegura una reconstrucción de los tiempos de ejecución exitosa.

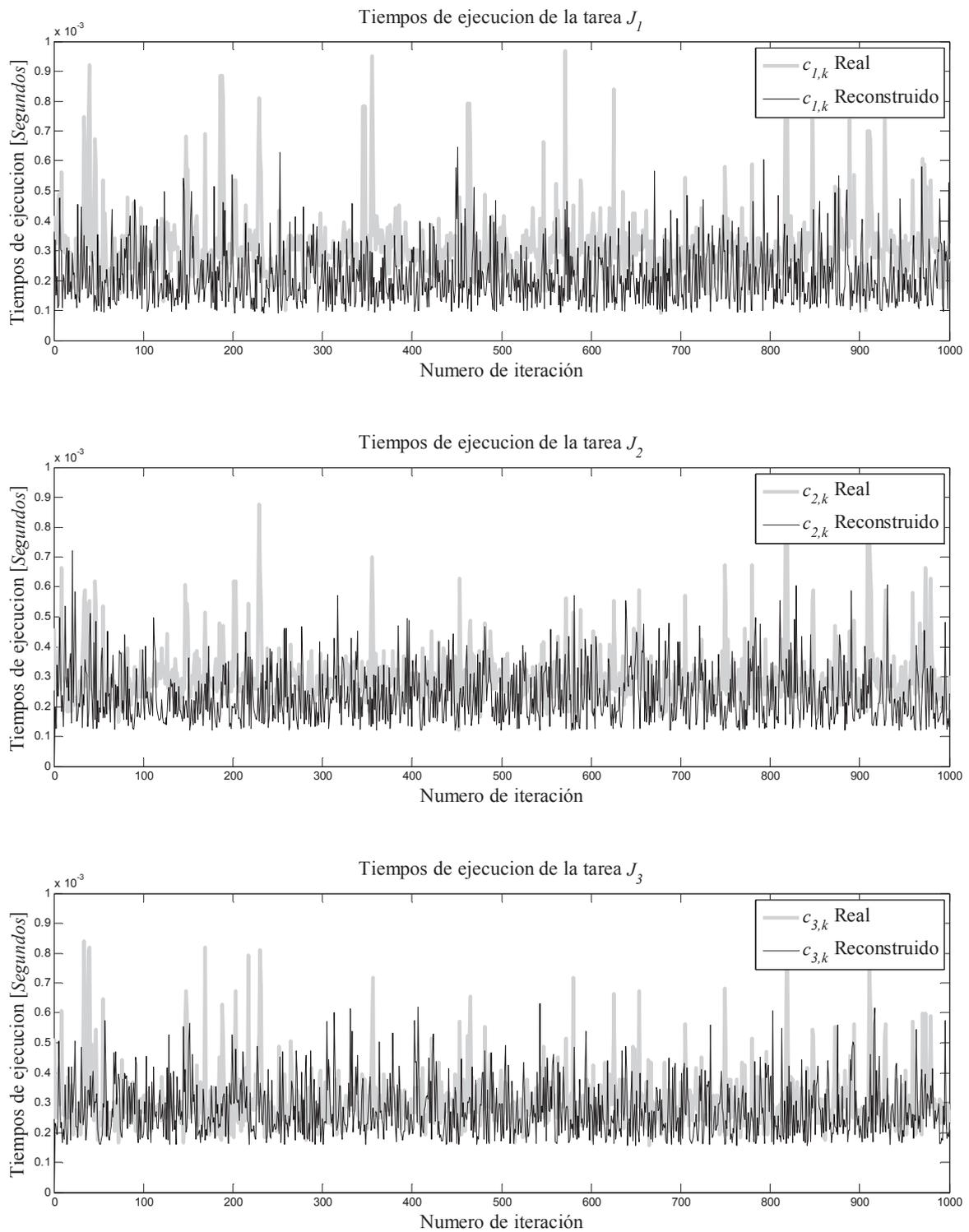


**Figura 8** Error cuadrático medio  $G_{i,k}$  calculado entre el parámetro del sistema  $a_{i,k}$  y el parámetro estimado del sistema reconstruido  $\hat{a}_{i,k}$  para las TTR  $J_1$ ,  $J_2$  y  $J_3$

**Reconstrucción de los tiempos de ejecución  $\hat{c}_{i,k}$  del algoritmo programado**

En la figura 9 se puede apreciar los tiempos de ejecución  $c_{i,k}$  medidos en color claro y los tiempos de ejecución  $\hat{c}_{i,k}$  reconstruidos en color oscuro. El comportamiento de los tiempos de ejecución  $c_{i,k}$  medidos se observan con variaciones entre muestras contiguas, originadas por factores internos y externos del equipo de cómputo como las descritas en [3] y [4]. Por ello se planteó la necesidad de utilizar un modelo un estocástico

autoregresivo de promedios móviles ARMA de orden (1, 1) descrito en (2) y (3) considerando a [5], integrado con un estimador de parámetros construido con un procedimiento de filtrado digital difuso. En la figura 9 se observa que a partir de la estimación del parámetro  $\hat{a}_{i,k}$  a través del filtro difuso, los tiempos de ejecución reconstruidos  $\hat{c}_{i,k}$  son muy próximos a los tiempos de ejecución reales  $c_{i,k}$  para las TTR  $J_1$ ,  $J_2$ , y  $J_3$  en todos los valores de  $k$ , pero se considera que este argumento no es suficiente, lo que conlleva a usar medidas descriptivas tal y como el error cuadrático medio para validar este resultado.



**Figura 9** Tiempos de ejecución medidos y reconstruidos por medio del filtro difuso, para las Tareas en Tiempo Real  $J_1$ ,  $J_2$ , y  $J_3$  con 1000 instancias

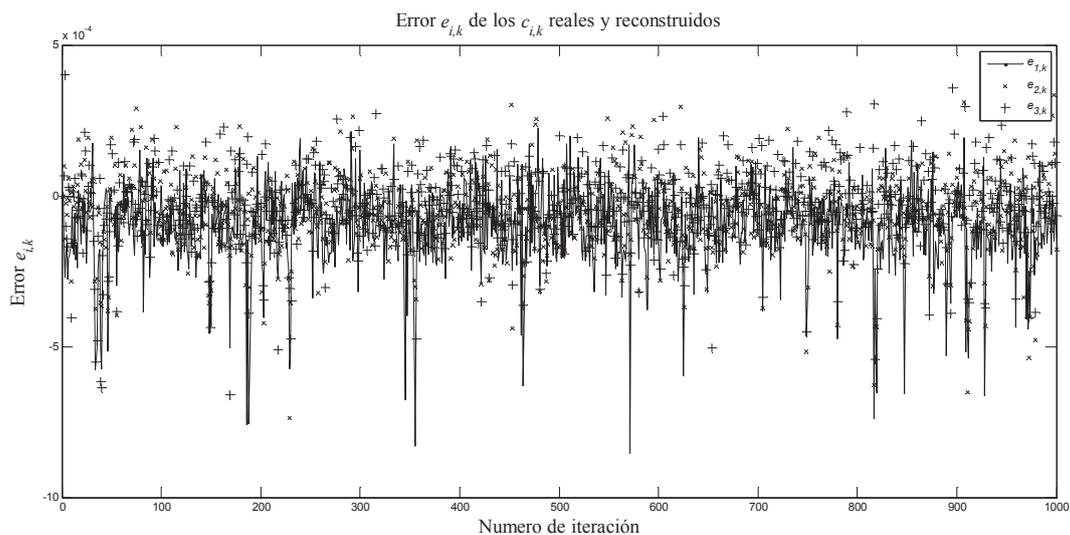
**Error cuadrático medio  $H_{i,k}$  de los tiempos de ejecución medidos  $c_{i,k}$  y reconstruidos  $\hat{c}_{i,k}$**

Para el caso del error cuadrático medio entre los tiempos de ejecución reales  $c_{i,k}$  y tiempos de ejecución reconstruidos  $\hat{c}_{i,k}$  para las TTR  $J_1$ ,  $J_2$  y  $J_3$  se va describiendo la convergencia del filtro mediante el empleo de (11) y (12) hasta encontrar el mínimo error de  $H_{i,k}$ .

$$e_{j,k} = \hat{c}_{i,k} - c_{i,k} \tag{11}$$

$$H_{i,k} = \left[ \frac{1}{k} ((k-1)H_{i,k-1} + e_{j,k}^2) \right]^{\frac{1}{2}} \tag{12}$$

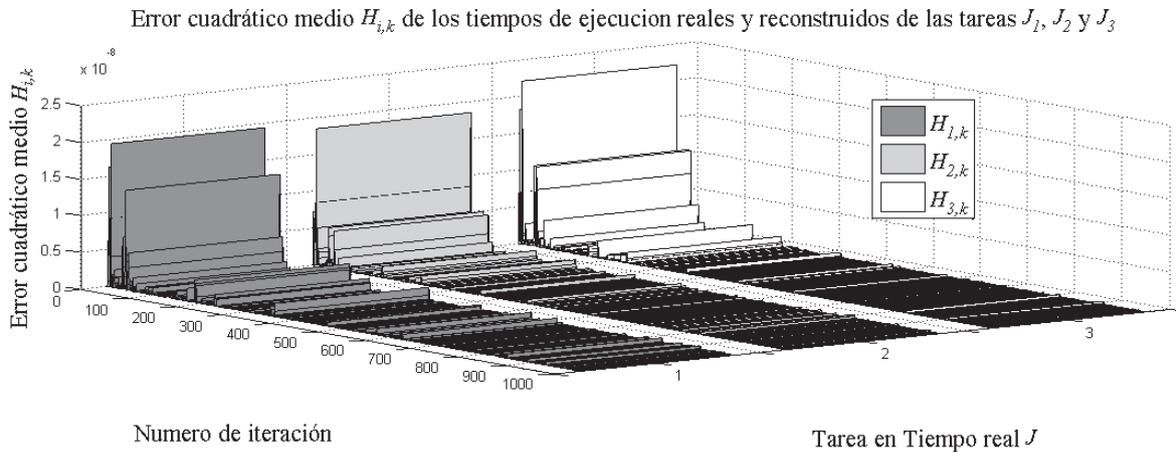
En este sentido los errores definidos por la ecuación (11) se muestran en la figura 10, ya que estos son empleados en la obtención del error cuadrático medio  $H_{i,k}$ . En esta figura se observa que el error para  $c_{1,k}$ ,  $c_{2,k}$  y  $c_{3,k}$  es aproximadamente  $\pm 5 \times 10^{-4}$  [Segundos] en cada caso.



**Figura 10** Error  $e_{j,k}$  entre los tiempos reales y reconstruidos para las TTR  $J_1$ ,  $J_2$  y  $J_3$

Finalmente en la figura 11 y de acuerdo a la ecuación (6), el filtro tiene una convergencia en casi todos los puntos; en ésta se observan las respuestas de los errores  $H_{1,k}$ ,  $H_{2,k}$  y  $H_{3,k}$ , destacándose que estos convergen a valores muy pequeños:  $1,398 \times 10^{-7}$  [Segundos],  $1,774 \times 10^{-7}$  [Segundos] y  $6,525 \times 10^{-8}$  [Segundos] para  $c_{1,k}$ ,  $c_{2,k}$  y  $c_{3,k}$  respectivamente, lo que permite describir la calidad de la reconstrucción pues a medida

que se obtengan valores cada vez más pequeños de  $H_{i,k}$  durante el proceso de filtrado difuso, la diferencia descrita por la ecuación (11), converge a valores cercanos a 0, lo que permite establecer que la dinámica de los tiempos de ejecución reconstruidos es similar a la de los tiempos de ejecución medidos, con lo que se valida la reconstrucción lograda a través del algoritmo computacional utilizado en este trabajo.



**Figura 11** Error cuadrático medio  $H_{i,k}$  calculado entre los tiempos de ejecución reales y tiempos de ejecución reconstruidos para las Tareas en Tiempo Real  $J_1$ ,  $J_2$  y  $J_3$  con 1000 instancias

### Conclusiones

De los resultados mostrados, así como de su análisis, se pueden obtener las siguientes conclusiones, sobre la reconstrucción de la dinámica de los tiempos de ejecución de TTR empleando filtrado difuso. 1) Debido a que la mayoría de los autores de sistemas en tiempo real consideran a los tiempos de ejecución  $c_{i,k}$  como el peor caso medido (WCET), no estudian la dinámica del comportamiento de los tiempos de ejecución, entonces como aportación del presente artículo es la propuesta de modelado y reconstrucción de la dinámica variante de los tiempos de ejecución de las TTR utilizando un modelo ARMA y estimando el parámetro interno del sistema con un filtro digital difuso en lugar de usar un filtro estocástico. 2) El filtrado difuso cuenta con la capacidad de interpretar dinámicamente las variables de un sistema, así como sus respectivas variaciones, y a partir de esta interpretación dar una respuesta natural de acuerdo a los requerimientos de un proceso, garantizando un funcionamiento adecuado. 3) Para la estimación del parámetro del sistema  $\hat{a}_{i,k}$  para cada tarea, el filtro difuso se diseñó con tres

funciones de membresía (bajo, medio y alto) del tipo gaussiano obteniéndose buenos resultados reflejados en el comportamiento del error cuadrático medio entre el parámetro estimado y el real. 4) La reconstrucción a partir de del modelo autoregresivo de promedios móviles ARMA de orden (1, 1) basado en filtrado difuso, es suficientemente buena ya que los tiempos de ejecución reconstruidos  $\hat{c}_{i,k}$  son muy próximos a los tiempos de ejecución reales  $c_{i,k}$  para las tres tareas y para todos los valores de  $k$ , obteniéndose una media aritmética del error entre el valor real y estimado del tiempo de ejecución, de aproximadamente de  $1 \times 10^{-3}$  segundos en cada caso, comprobándose que el uso del modelo ARMA con el filtro difuso son una excelente herramienta para lograr la reconstrucción de la dinámica de tiempos de ejecución, pues a pesar de las características estadísticas de estos y de su dinámica cambiante, las magnitudes de los errores cuadráticos obtenidos convergen de manera asintótica a 0. 5) El método de medición propuesto para obtener los tiempos de ejecución del algoritmo programado integrado por tres tareas concurrentes, fue satisfactoria, pues permitió obtener valores con mucha precisión.

## Referencias

1. J. Medel, P. Guevara, D. Cruz. "Temas selectos de sistemas en tiempo real." *Tareas en Tiempo Real*. 1ª ed. Ed. Politécnico. México DF., México. 2007. pp.50-54
2. S. Manolache, P. Eles, Z. Peng. "Schedulability Analysis of Applications with Stochastic Task Execution Times". *ACM Trans. on Embedded Computing Systems*. Vol. 3. 2004. pp. 706-735.
3. F. Stappert, P. Altenbernd. "Complete Worst-Case Execution Time Analysis of Straight-line Hard Real-Time Programs". *Journal of Systems Architecture*. Vol. 46. 2000. pp. 339-355.
4. G. Bernat, A. Colin, S. Petters. *pWCET: A tool for probabilistic Worst-Case Execution Time Analysis of Real-Time Systems*. Technical Report YCS-2003-353. Department of Computer Science, University of York. York, UK. 2003. pp. 1-18.
5. P. Guevara, J. Medel, G. Delgado. "Description of execution time dynamics for a set of concurrent real-time tasks". *Revista Facultad de Ingeniería: Universidad de Antioquia*. N.º. 61. 2011. pp. 123-131.
6. T. Söderström, P. Stoica. "On some system identification techniques for adaptive filtering". *IEEE Trans. Circuits and Systems*. Vol. 35. 1988. pp. 457-461.
7. J. Medel, P. Guevara, D. Cruz. *Matricial estimation for start times with stochastic behavior by periodic real time tasks in a concurrent system*. Proc. of the 7<sup>th</sup> International Conference on Mathematical Methods and Computational Techniques In Electrical Engineering. Sofia, Bulgaria. 2005. pp. 214-217.
8. C. Liu, J. Layland. "Scheduling algorithms for multiprogramming in hard-real-time environment". *Journal of the ACM*, Vol. 20. 1973. pp. 273-250.
9. P. Felix "Fuzzy Temporal Profiles: A Model for the representation and Recognition of signal patterns". *Computación y Sistemas*. Vol. 4. 2001. pp. 385-388.
10. L. Di Lascio, E. Fischetti, A. Gisolfi. "Fuzzy Modeling and Evaluation of an IMTS User". *Computación y Sistemas*. Vol. 5. 2002. pp. 247-255.
11. M. Chacón "A Fuzzy Neural Network Approach for Document Region Classification Using Human Visual Perception Features". *Computación y Sistemas*. Vol. 6. 2002. pp. 083-093.
12. M. Chacón, A. Corral, R. Sandoval. "A Fuzzy Approach on Image Complexity Measure". *Computación y Sistemas*. Vol. 10. 2007. pp. 268-284
13. J. Aguilar. "Dynamic Random Fuzzy Cognitive Maps". *Computación y Sistemas*. Vol. 7. 2004. pp. 260-270.
14. E. Gorrostieta, E. Vargas. "Algoritmo Difuso de Locomoción Libre para un Robot Caminante de Seis Patas". *Computación y Sistemas*. Vol. 11. 2008. pp. 260-287.
15. C. López, C. Yáñez, A. Gutiérrez, E. Felipe. "Adequacy Checking of Personal Software Development Effort Estimation Models Based upon Fuzzy logic: A Replicated Experiment". *Computación y Sistemas*. Vol. 11. 2008. pp. 333-348.
16. J. García. "Filtrado Digital Difuso en Tiempo real". *Computación y Sistemas*. Vol. 11. 2008. pp. 390-401.
17. J. García, J. Medel, P. Guevara. "Filtrado Difuso en Tiempo Real". *Computación y Sistemas*. Vol. 11. 2008. pp. 390-401.
18. J. Medel, J. García, J. Sánchez. "Real-time Fuzzy Digital Filters (RTFDF) Properties for SISO Systems". *Automatic Control and Computer Sciences*. Vol. 42. 2008. pp. 26-34.
19. J. García, J. Medel, J. Sánchez. "Filtrado Neuronal Difuso: Caso MIMO". *Revista Ingeniería e Investigación*. Vol. 31. 2011. pp. 184-192.
20. J. Valdez, C. García, P. Guevara, J. Sánchez. *Filtrado Difuso para la reconstrucción de la dinámica de tiempos de ejecución para n instancias en una tarea en tiempo real*. Memorias del CIINDET 2011. Morelos, México. 2011. pp. 1-8.
21. S. Vázquez, J. García, J. Sánchez. *Descripción del Filtrado Adaptativo Difuso en un DSK TMS320C6713*. Memorias del CIINDET 2009. Morelos, México. 2009. pp. 1-6.
22. G. Delgado, P. Guevara, J. Falcón. *Simulación Concurrente en Tiempo Real de un motor de Corriente Continua Sobre la Plataforma QNX*. Memorias del XIV Congreso Latinoamericano de Control Automático, XIX Congreso de la Asociación Chilena de Control Automático ACCA. Santiago de Chile, Chile. 2010. pp. 1-6.
23. R. Krten. *QNX Neutrino RTOS Getting Started with QNX Neutrino: A Guide for Real-time Programmers*. 1ª ed. Ed. QNX Software Systems International Corporation. Canada. 2008. pp. 191-283.