

# Distribución del tiempo ocioso orientado al ahorro de energía en tareas de control de tiempo real

## Slack time distribution oriented energy aware in real-time control task

Alfonsi Sebastiani, Alfonso Salvador<sup>1\*</sup>; Yáñez Martínez, Raiza del Carmen<sup>2</sup>; Pérez Rodríguez, Jesús Alberto<sup>3</sup>

<sup>1</sup> Grupo de Investigación Arquitecturas de Sistemas de Control (GASC), Escuela de Ingeniería y Cs. Aplicadas, Universidad de Oriente, Núcleo de Anzoátegui, Barcelona 6001, Venezuela

<sup>2</sup> Dpto. Sistemas Industriales, Escuela de Ingeniería y Cs. Aplicadas, Barcelona, Universidad de Oriente, Núcleo de Anzoátegui, Barcelona 6001, Venezuela

<sup>3</sup> Universidad Politécnica Territorial del Estado Aragua "Federico Brito Figueroa", La Victoria 2121, Venezuela

\* alfonso\_alfonsi@udo.edu.ve

### Resumen

*Un criterio que sostiene las tecnologías sustentables es mejorar la eficiencia energética en los procesadores que contienen los sistemas empotrados, incidiendo en su operatividad. El objetivo del trabajo es definir el tiempo ocioso multifrecuencia o tiempo verde, desde las causas que lo producen y como permite distribuir los tiempos ociosos debidos la ejecución de las tareas de control  $i$  ( $T_i$ ) intra/interTarea, aprovechando las operaciones del control multifrecuencia y la variabilidad de los tiempos de cómputo, para ajustar la energía consumida por  $T_i$ . Se apeló a la integración del control multifrecuencia, la planificación de tiempo real y técnicas de escalamiento dinámico de voltaje y frecuencia. Para validar el funcionamiento se tomaron un conjunto de tareas para pruebas comparativas, variando los tiempos de cómputo consumidos. El comportamiento global intertarea y local inter/intraTarea, arrojan un consumo de energía del 44,70% al 90,00% y 42,00% al 86,66%, respectivamente. El tiempo verde permite fijar límites en cuanto al tiempo ocioso disponible inter e intraTarea. Este aporte promociona el desarrollo de tecnologías sustentables.*

**Palabras clave:** Ahorro de energía, DVFS, tiempo ocioso dinámico, tiempo real.

### Abstract

*An approach that maintains the sustainable technologies is to improve energy efficiency in the processors that contain embedded systems, stressing its operability. The objective of this work is to define the multirate slack time or green time, from the causes that produce and how to distribute the slack time due the execution of the control tasks  $i$  ( $T_i$ ) intra/interTask, taking advantage of the operations of the multirate control and variability of computation time, to adjust the energy consumed by it. It appealed to the integration of the multirate control, real-time scheduling and dynamic voltage and frequency scaling techniques. To validate the operation took a set of tasks for benchmark, by varying the computation time consumed. The interTask overall behavior and inter/intratarea local, throw a power consumption of the 44.70 % to 90.00 % and 42.00 % to 86.66 %, respectively. The green time allows you to set time limits on the slack time available inter/intraTask. This contribution promotes the development of sustainable technologies.*

**Palabras clave:** Energy aware, DVFS, dynamic slack time, real-time.

### 1 Introducción

Un Sistema Empotrados de Control de Tiempo Real Autónomo (SECTRA) es una combinación de hardware (procesadores, memoria, dispositivos de entrada y salidas),

software (núcleos de control o sistemas operativos, y software de aplicación) y algunas otras partes, como sensores y/o actuadores, destinadas a desempeñar funciones dedicadas de control, con autonomía energética y características de tiempo real (Pont 2011, Noergaard 2013). Lo anterior se materializa con tareas que ejecutan leyes de control y algo-

ritmos de procesamiento de señal para la manipulación del entorno. El algoritmo del o los controladores está almacenado en la memoria del sistema, e interactúa directamente con los procesos a los que este está asociado. Estos sistemas se alojan en un producto más grande y normalmente no son visibles para el usuario, y pueden estar en sitios alejados y a la intemperie. Además, el software empotrado es vital pues los convierte en sistemas de procesamiento computacional.

Actualmente, el consumo de energía de los SECTRA es un tema de apertura a nivel tecnológico de especial relevancia que repercute en el tiempo de operación debido a la dependencia de baterías para su alimentación, también incide en la temperatura lo que ocasiona comportamientos indeseados.

Al enfocar la atención al consumo de energía en los SECTRA se destaca una vía relacionada a la capa intermedia de software (núcleo de control, sistema operativo), llamada Escalamiento Dinámico de Voltaje y Frecuencia (DVFS= Dynamics Voltage and Frequency Scaling), con la cual se puede reducir la energía consumida ajustando el voltaje de alimentación y frecuencia de operación de un procesador con características de bajo consumo de energía (Hu y Quan, 2007, Piguet 2006).

El DVFS usa dos estrategias para distribuir el tiempo ocioso (ST= slack time) llamadas inter e intraTarea (Hu y Quan, 2007, Piguet 2006, Pillai, Shin, 2001). El ST se produce cuando una tarea se ejecuta consumiendo menos de su tiempo de cómputo de peor caso (WCET= Worst Case Execution Time). Merece destacarse que en las aplicaciones de los SECTRA la posibilidad de que una tarea requiera un tiempo de cómputo igual a su WCET es muy baja.

El ST se utiliza en planificadores de tiempo real para ejecutar tareas aperiódicas junto con las periódicas con el objetivo de no violar sus restricciones temporales (Buttazzo 2011), así como también puede orientarse a darle mayor tiempo a las tareas para ejecutarse, de modo que consuma menos energía sin vulnerar sus características temporales, convirtiéndolos en planificadores con ahorro de energía (Scordino y col., 2007, Xia, Sun, 2008, Abdelzaher y col., 2008), los cuales disponen de estrategias estáticas y dinámicas.

Existen propuestas que integran el manejo de la energía por hardware, el DVFS (Hu y Quan, 2007, Piguet 2006), a la planificación realimentada (Hellerstein y col. 2004), las cuales son conocidas como planificación realimentada con ahorro de energía (Xia, Sun, 2008). Algunas explotan técnicas para la utilización del ST (Chantem y col., 2009, Niu 2011, Alfonsi y col., 2013).

El objetivo de este trabajo es presentar el tiempo ocioso multifrecuencia o tiempo verde, desde las causas que lo producen y como se utiliza para distribuir los ST debidos a la ejecución de las tareas de control  $i$  que pertenecen a sus instancias  $k$  ( $\tau_{i,k}$ ) inter e intraTarea, aprovechando las operaciones naturales de las técnicas de control multifrecuencia, observando la variabilidad de los tiempos de cómputo y así ajustar el voltaje y frecuencia de operación consumida por

$\tau_{i,k}$ , que se reflejará en la energía requerida por éste.

Además se está dando un aporte en cuanto a la reducción de energía en los equipos de cómputo, al implementar mecanismos en procesadores para tal fin, cuyos efectos están acordes con el uso eficiente de los recursos computacionales minimizando el impacto ambiental, maximizando su viabilidad económica y asegurando deberes sociales, como lo mantiene la Tecnología Verde (López-Vallejo y col., 2011). También se atiende a la invitación que se hace a los desarrolladores, investigadores e innovadores de otras ramas del conocimiento, distintas a las tradicionales involucradas con el Desarrollo Sustentable, a formar un frente multi y transdisciplinario, que ayude a fomentar el compromiso con la sostenibilidad mediante productos e innovaciones inspiradas en ésta temática, cada vez más pertinente, orientado a contribuir en la disminución de la crisis socioambiental (Yáñez y col., 2013).

La organización del trabajo está estructurada como sigue. En la sección dos, se presenta el desarrollo y definición de tiempo ocioso multifrecuencia o tiempo verde. La sección tres, presentan el funcionamiento bajo un conjunto de tareas para pruebas comparativas, variando los tiempos de cómputo consumidos, haciendo una discusión de los resultados. Finalizando con las conclusiones del trabajo.

## 2 Desarrollo

### 2.1 Modelo de la Tarea de Control de Tiempo Real

En la planificación realimentada con ahorro de energía la gestión de recursos es formulada como un problema de control de lazo cerrado, tratando los sistemas de cómputo como un proceso controlado (Pillai y Shin, 2001, Xia, Sun, 2008, Abdelzaher y col., 2008), reflejado en el compromiso entre las características temporales y la señal de control obtenida, que se manifiesta en las demandas al procesador, ya sea por manipulación de los períodos de muestreo o los tiempos de cómputo de cada tarea. Segundo, tiene la flexibilidad de poder ajustar políticas de planificación de tiempo real a las tareas, de tal forma que consuman una cierta cantidad de energía, la cual dependerá de la velocidad seleccionada para su ejecución, indicando el cambio en el nivel de voltaje de forma que se minimice el consumo de energía y no afecte sus restricciones temporales.

Las tareas de control de tiempo real consideradas en este trabajo son críticas, periódicas, aperiódicas, independientes, apropiables, no tienen restricciones de precedencia, ni presentan armonicidad en sus períodos de activación. Su modelo se define como una entidad ejecutable  $T_i$ , siendo  $i$  la identificación para cada tarea, formada por un conjunto de instancias o unidades de trabajo  $\tau_{i,k}$ , que se manifiesta en la activación  $k$ , esto es:

$$T_i = \{\tau_{i,k}\} \forall i \in (1,2,\dots,N) \wedge k \in (1,2,\dots,M) \quad (1)$$

$$\tau_{i,k} = (C_{i,k}, D_{i,k}, P_{i,k}, \alpha_{i,k}) \quad (2)$$

donde  $C_{i,k}$  es el tiempo de cómputo,  $D_{i,k}$  el plazo de finalización,  $P_{i,k}$  el período de activación y,  $\alpha_{i,k}$  es el factor de escalamiento que representa la normalización de la velocidad a frecuencia actual ( $f_{i,k}$ ) de operación del procesador, dado por

$$\alpha_{i,k} = \frac{f_{i,k}}{f_{max}} \quad \forall i \in (1,2,\dots,N) \wedge k \in (1,2,\dots,M) \quad (3)$$

acotada entre las frecuencias máxima ( $f_{max}$ ) y mínima ( $f_{min}$ ). También se cumple en este trabajo

$$D_{i,k} = D_i \wedge P_{i,k} = P_i \quad \forall i \in (1,2,\dots,N) \wedge k \in (1,2,\dots,M) \quad (4)$$

$$C_i^{f_{max}} \leq C_{i,k} \leq C_i^{f_{min}} \quad (5)$$

además,

$$C_i^{f_{max}} = WCET_i. \quad (6)$$

Vale la pena acotar, que de aquí en adelante, el superíndice de la variable involucrada indica la magnitud a la cual está referida. El subíndice sigue identificando la tarea.

Ejemplo,  $C_i^{f_{max}}$  es el tiempo de cómputo de  $T_i$  a  $f_{max}$ .

La reducción del consumo de energía en un procesador DVFS se enfoca en la utilización de los ST que se producen en la ejecución de las tareas de tiempo real.

El ST (Chantem y col., 2009, Niu 2011, Zhu, Muller, 2006, Urriza y col., 2010) está orientada a satisfacer la holgura cuando una instancia  $\tau_{i,k}$ , no consume su  $WCET_i$ . Está dado por

$$ST_{i,k} = WCET_i - C_{i,k} \quad (7)$$

Vale la pena acotar que este  $ST_{i,k}$  tomará el nombre de  $STWCET_{i,k}$  por razones de la estructura utilizada.

En la Fig. 1 se presentan las características enunciadas de una  $T_i = \{\tau_{i,k}\}$ .

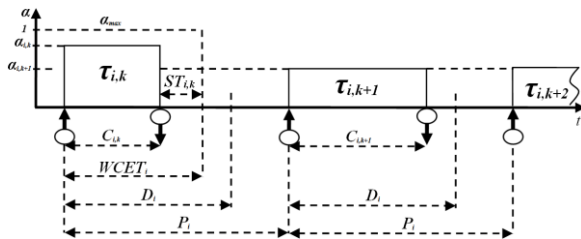


Fig. 1. Tarea de Control de Tiempo Real.

## 2.2 Arquitectura Orientada al manejo del ST para el Ahorro de Energía

Para manejar el ST se necesita de algún método que calcule y contabilice de manera exacta o aproximada, directa e indirecta, cuanto ST se puede disponer para ajustar el consumo de energía sin que el sistema de tiempo real viole sus restricciones temporales. Además es importante la forma de distribuir el ST (Shin y col., 2000), siendo estas, el intraTarea que aporta el ST ocasionado por una tarea ejecutada a su próxima activación, y la interTarea que distribuye el ST a las próximas tareas planificadas.

Para tal fin se recurre al Lazo de Planificación Realimentado Multifrecuencia con Ahorro de Energía (LPRMFAE) establecido en Alfonsi y col. (2013). En la Fig. 2 se muestra la arquitectura del LPRMFAE, que mantiene una estructura funcional con enfoque de control local y entrada de referencia local, incorporando un control con muestreo no convencional o multifrecuencia del factor de carga del procesador ( $U_T$ ) calculada por (8), el cual será muestreado cada período global o hiperperíodo  $H$  (mínimo común múltiplo de los  $P_i$ ), de tal forma que permita regular dinámicamente el tiempo de cómputo, de forma global, a todas las instancias de una tarea, o local, a cada instancia de ejecución, sumándole características de ajuste intra e inter-Tarea, con la intención de escalar la velocidad del procesador, y por ende, el consumo de energía. La política de planificación empleada es la EDF (Earliest Deadline First), planificador dinámico que selecciona las tareas de acuerdo a su plazo absoluto, utiliza adelanto y puede ser usado para planificación de tareas periódicas como aperiódicas.

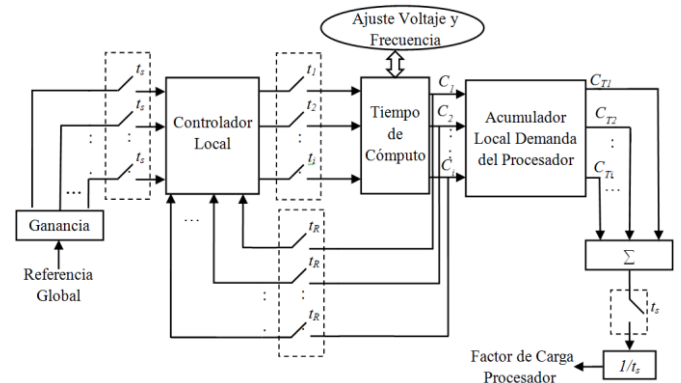


Fig. 2. Arquitectura Lazo de Planificación Realimentado Multifrecuencia con Ahorro de Energía (LPRMFAE).

El factor de carga del procesador se determina por

$$U_T = \frac{\sum_{i=1}^N \sum_{k=1}^{N_{T_i}} C_{i,k}}{H} \leq 1 \quad (8)$$

donde  $N$  es el número de tareas, y  $N_{T_i}$  es el número de  $\tau_{i,k}$

involucradas en  $H$ , definida así

$$N\tau_i = \frac{H}{P_i} \quad \forall i \in (1, 2, \dots, N) \rightarrow N\tau_i \in \mathbb{Z}^+ \quad (9)$$

El  $ST$  puede ser explotado por algoritmos de planificación dinámica para cambiar la planificación en tiempo de ejecución, y además reducir la velocidad del procesador. En su mayoría utilizan para ello, por lo menos una de las siguientes técnicas de estimación del  $ST$  (Scordino 2007): Estirar al Próximo Tiempo de Llegada, Uso del Tiempo Ocioso por otras Tareas, Actualización de la Utilización.

Siguiendo la estructura del LPRMFAE (Alfonsi y col. 2013), las técnicas dinámicas anteriormente mencionadas pasan a llamarse:  $ST$  debido a las Próximas Activaciones ( $STEPA$ ),  $ST$  debido al  $WCET$  ( $STWCET$ ) y  $ST$  debido a la Actualización del Factor de Carga del Procesador ( $STU$ ).

Por razones de la arquitectura el  $STU$  es la técnica base a utilizar, la cual ajusta  $C_{i,k}$  del sistema respecto a un Factor de Carga de Referencia ( $U_{ref}$ ). Entonces el  $ST$  debido a la actualización del Factor de Carga  $STU$ , dado por:

$$STU_{i,k} = C_{i,k}^{f_{max}} - C_{i,k}^{U_{ref}f_i} \quad (10)$$

En la Fig. 3 se muestra la distribución intra e interTarea usando como base LPRMFAE, donde se aprecia que  $\tau_{1,1}$  y  $\tau_{2,1}$  inician ejecutándose a  $\alpha_{i,k}=1$ , lo que indica máxima frecuencia de operación, ofreciendo  $ST$ . En  $\tau_{1,2}$  se aprovecha la distribución del  $ST$  intraTarea, que toma dicho  $ST$  de  $\tau_{1,1}$  ocasionado por  $STU$ . Para  $\tau_{2,2}$  se muestra su ejecución debido al  $STU$  contabilizado en  $\tau_{2,1}$ , siguiendo la distribución del  $ST$  intraTarea. O, la distribución del  $ST$  interTarea activando para tal fin  $STEPA$ . En ambas activaciones se ajusta los  $\alpha_{i,k}$  que se traduce en ahorros significativos de la energía consumida por las tareas involucradas. Siendo  $P_{abi}$  el período absoluto requerido por el  $EDF$ .

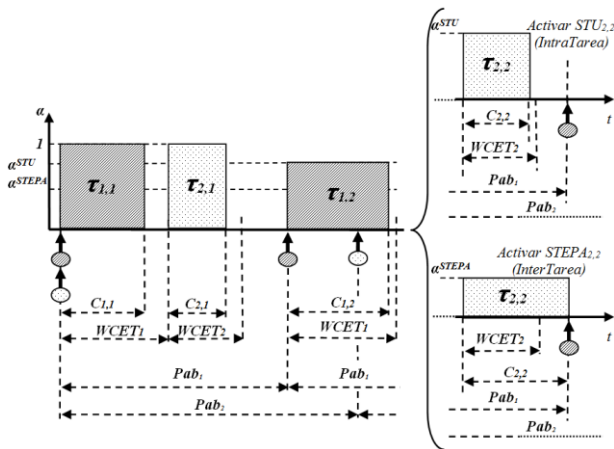


Fig. 3. Distribución del  $ST$  con LPRMFAE intra e interTarea.

De lo anterior, al considerar que el voltaje y frecuen-

cia de operación de las tareas son variables, lo que trae por consecuencia que si una instancia de una tarea al variar la velocidad sufrirá un cambio en la escala temporal, se producirá una diferencia cuando se contabilice el tiempo de ejecución a la frecuencia mínima permitida por el procesador y la impuesta por la aplicación. Bajo la premisa anterior, a continuación se introduce el concepto de tiempo ocioso multifrecuencia o tiempo verde.

### 2.3 Tiempo Ocioso Multifrecuencia o Tiempo Verde

Si una  $\tau_{i,k}$  tiene sus  $STWCET_{i,k}$ ,  $STU_{i,k}$  y  $STEPA_{i,k}$ , éstos serán contabilizados para proporcionar un  $ST$  que podrá ser consumido por el planificador para ajustar la velocidad de operación de la próxima  $\tau_{i,k}$ , si son causados por  $STWCET_{i,k}$  y  $STU_{i,k}$ , y/o en la actual ejecución si es debida por  $STEPA_{i,k}$ .

Pero por restricciones de planificación es posible que las  $\tau_{i,k}$  no logren ser ejecutadas en todas las frecuencias ofrecidas por el procesador, por ésta razón, las  $T_i$  solo pueden ser retrasadas ciertas unidades de tiempo (beneficiándose del  $STWCET_{i,k}$ ,  $STU_{i,k}$  y  $STEPA_{i,k}$ ), para poder garantizar las restricciones temporales, y considerar, que dicho retraso es suficiente para poder reducir la frecuencia de operación hasta una cierta frecuencia.

Entonces se define el tiempo verde u ocioso multifrecuencia,  $TV_{i,k}$ , a la variación de la carga de trabajo que sucede cuando una  $\tau_{i,k}$ , operando  $f_{i,k}$ , se ejecuta por debajo del  $C_{i,k}$  a  $f_{min}$ , calculada como sigue

$$TV_{i,k} = C_{i,k}^{f_{min}} - C_{i,k}^{f_{i,k}} \quad (11)$$

En la Fig. 4 se muestra el modelo de  $T_i$  con el  $TV_{i,k}$ .

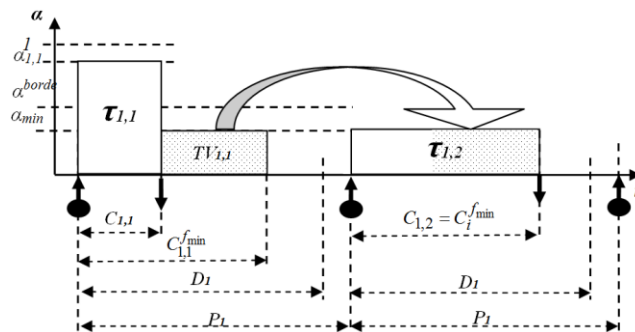


Fig. 4. Modelo de la Tarea de Control  $T_i$  con  $TV_{i,k}$  menor que las restricciones temporales.

El  $TV_{i,k}$  contribuye a disponer de una frecuencia de operación que retrasará el  $C_{i,k}$  según la restricción de la aplicación que debe ser menor o igual a  $f_{min}$ . Es evidente sin embargo, que la magnitud  $TV_{i,k}$ , ocasionalmente, pudiera violar las características de tiempo real, lo que lleva a establecer una frecuencia de borde ( $f_i^{borde}$ ), que se define a

continuación.

Un procesador puede ajustar dinámicamente su frecuencia de trabajo en un rango de  $f_{min} \leq f_{i,k} \leq f_{max}$ .

El suministro de esta frecuencia de operación puede ser escalado con el factor  $\alpha$ , magnitud que representa la relación de la velocidad de operación actual del procesador referida a la velocidad máxima posible. Además se puede relacionar el  $C_{i,k}$ , a  $f_{i,k}$ , y el obtenido a  $f_{max}$  (Xia y Sun, 2008, Scordino 2007, Xia y col., 2009) mediante

$$\alpha_{i,k} = \frac{WCET_i}{C_{i,k}} \quad (12)$$

Ahora bien, en sistemas con restricciones temporales pretender operar en el rango, de voltaje y frecuencia, suministrado por el procesador es obviar que las tareas cuando se ejecutan a velocidad menor, ocupan menos espacio temporal que la misma ejecutada a mayor velocidad. Por tal motivo, hay ciertas frecuencias donde las tareas empiezan a incumplir con las restricciones temporales, como el plazo de finalización y/o el período.

Como consecuencia de lo anterior, se debe buscar cual es la frecuencia límite menor,  $f_i^{lim}$ , a la que se puede operar la aplicación, respetando sus restricciones temporales. Empleando (8) se define la relación que existe entre el factor de carga que trabaja a frecuencia límite menor  $U_i^{f lim}$ , y el que trabaja a máxima frecuencia  $U_T^{f max}$ , llamada  $\alpha^{f lim}$ . Por razones de planificación, el factor de carga a frecuencia límite menor  $U_i^{f lim}$  es igual al factor de carga de referencia  $U_{ref}$  impuesta por la aplicación, y  $\alpha^{f lim}$ , bajo esta premisa se convierte en  $\alpha^{borde}$ , calculada así

$$\alpha^{borde} = \frac{U_T^{f max}}{U_{ref}} \quad (13)$$

Entonces, teniendo  $\alpha^{borde}$  se calcula el tiempo de cómputo de borde ( $C_i^{borde}$ ) definido como el tiempo de cómputo relacionado a la frecuencia mínima de operación, que garantiza las restricciones temporales y la utilización del procesador y obtener ahorro de energía, utilizando

$$\alpha^{borde} = \frac{C_i^{f max}}{C_i^{borde}} \quad (14)$$

Con  $\alpha^{borde}$  se establece el rango real de operación para una aplicación, que propondrá un rango de  $C_{i,k}$  para ejecutarse sin violar las restricciones temporales, permanecer dentro de la utilización de referencia del procesador.

### 2.3 Ejemplo de la influencia del $TV_{i,k}$ calculado fuera de línea

Se tienen los parámetros temporales de tres  $T_i$  (Alfonsi y Pérez, 2009), tabla 1. Así como también la característica de voltaje y frecuencia de un procesador PXA270 de Marvell Technology Group (Marvell, 2009), tabla 2.

Tabla 1. Parámetros temporales

$T_i$	$WCET_i$ (ms)	$D_i$ (ms)	$P_i$ (ms)
1	14,30	55,00	58,00
2	10,45	30,00	58,00
3	6,60	20,00	58,00

Tabla 2. Característica de frecuencia/voltaje de un procesador XScale de Marvell Technology Group

$f$ (Mhz)	150	400	600	800	1.000
$V_{dd}$ (V)	0,75	1,00	1,30	1,60	1,80

El suministro de voltaje y frecuencia de operación, puede ser escalado con el factor  $\alpha$  tomando como  $f_{min} = 150$  Mhz y  $f_{max} = 1.000$  Mhz, quedando

$$\alpha = \{0,15 \quad 0,40 \quad 0,60 \quad 0,80 \quad 1,00\}$$

De (12) se calcula

$$C_i^{f min} = \{95,33 \quad 69,66 \quad 44,00\} \text{ ms}$$

Como se observa de la tabla 1 las magnitudes de  $C_i^{f min}$  violan las restricción impuestas por sus plazos de finalización y períodos de activación. Por tal motivo se debe calcular los tiempos de cómputo de borde  $C_i^{borde}$  aplicando (13) y (14).

El factor  $\alpha^{borde}$  se calcula a partir de  $U_T^{f max}$ , como sigue

$$H = m.c.m(58,58,58) = 58;$$

$$N_1 = N_2 = N_3 = \frac{H}{P_i} = \frac{58}{58} = 1$$

$$U_T^{f max} = \frac{\sum_{i,k \in S_l} C_{i,k}}{t_2 - t_1} = \frac{14,30 + 10,45 + 6,60}{58} = 0,54$$

$$\alpha^{borde} = \frac{U_T^{f max}}{U_{ref}} = \frac{0,54}{0,8} = 0,68$$

De (14) se calculan los  $C_i^{borde}$  cuyos resultados se presentan en la tabla 3.

Tabla 3. Resultados del  $C_i^{borde}$ 

Tarea	$C_i^{borde}$ (ms)
$T_1$	21,03
$T_2$	15,37
$T_3$	9,70

$$U_T^{fborde} = \frac{\sum_{i,k \in S_l} C_{i,k}}{t_2 - t_1} = \frac{21,03 + 15,37 + 9,70}{58} = 0,80$$

Al verificar nuevamente los parámetros de la tabla 1, las magnitudes de  $C_i^{borde}$  no violan las restricción impuestas por sus plazos de finalización y períodos de activación.

### 3 Discusión de los Resultados

Como punto de partida es establecer los ambientes temporales de la aplicación y del procesador. De la aplicación, se toma un conjunto de tareas para realizar pruebas comparativas (*benchmark*), propuestos por Shin y col. (2001), utilizadas para evaluar los algoritmos propuestos en (Alfonsi y col., 2013, Moncusí 2005, Choi y col., 2007, Rakhmatov 2008, Chang y col., 2009). En la Tabla 4 se muestra los parámetros temporales de las  $T_i$ , los cuales serán destinados a imponer las restricciones de la aplicación. Así como también se establece  $U_{ref}$ , siendo en este caso,  $U_{ref} = 0,95$ . Respecto al procesador se dispone de las características de voltaje y frecuencia del procesador *PXA270* de *Marvell Technology Group* (Marvell 2009) presentadas anteriormente en la Tabla 2.

Tabla 4. Pruebas Comparativas

Tarea	$P$ (ms)	$D$ (ms)	$WCET$ (ms)
$T_1$	50	50	10
$T_2$	80	80	20
$T_3$	100	10	40

La experiencia se desarrolló para validar el comportamiento de las  $T_i$  a nivel de la planificación (tomando en cuenta el  $TV_{i,k}$ ) y la velocidad a la cual se ajustan las instancias usando el LPRMFAE. Se toma en consideración la variabilidad de los  $C_{i,k}$  en forma global (cada hiperperíodo) y local (a cada instancia de las tareas) al ser ejecutados a valores de  $\alpha$ .

Se han considerado los efectos de la variabilidad del porcentaje de los  $C_{i,k}$  consumidos en cada instancia, ya que muestra el funcionamiento del criterio y comportamiento de las  $T_i$ , y los consecuencias de  $U_T$ , ligadas a características de la aplicación y restricciones operativas del procesador, condiciones establecidas en Kim y col. (2003) y Bhatti y col. (2010).

Se procede a calcular, fuera de línea, lo siguiente:

1. El hiperperíodo  $H$  (mínimo común múltiplo de los  $P_i$ ).

2. El número de instancias  $N_{ti}$  que se deberían ejecutar durante  $H$  con (9).
3. Con (8) el factor de carga a máxima frecuencia.
4. El factor  $\alpha^{borde}$  con (14).

El LPRMFAE está desarrollado en C++, tomando ventaja de la programación orientada objeto y librerías de aplicación para integrar clases, de la estructura funcional del control multifrecuencia, la política de planificación *EDF*, el factor de escalamiento de borde y el procedimiento para activar el STEPA. Entonces se incorpora el método *CambioFV()* que tiene las características dadas en la Tabla 2, siendo su definición:

```
float Tarea::CambioFV()
Inicio_ definicion clase
    //Activar variador de voltaje y frecuencia
    //segúnProcesador
Fin_definicion
```

De la misma manera se incorpora el método *TVerde()*, el parámetro  $C$  es el tiempo de cómputo, siendo su definición:

```
float Tarea::TVerde()
Inicio_ definicion clase
    float TV
    TV = C_{i,k}^f \min - C
Fin_definicion
```

Se realizaron varias experiencias que arrojan resultados en un diagrama de planificación de la  $T_i$  con escalamiento dinámico  $\alpha_{i,k}$ , teniendo un rango de  $150 \text{ MHz} \leq f_{i,k} \leq 1000 \text{ MHz}$  y haciendo variaciones aleatorias de  $T_1$ ,  $T_2$  y  $T_3$  desde el consumo del 100% de todos sus  $C_{i,k}$ , alternando el 50% de los mismos, hasta llegar entre el 50% y 100% de sus  $C_{i,k}$  de forma aleatoria.

En la Fig. 5 se muestra el resultado en un diagrama de planificación de la  $T_i$  con escalamiento dinámico por tarea. La variación de  $T_1$ ,  $T_2$  y  $T_3$  está entre el 50% y 100% de sus  $C_{i,k}$ .

En su primera activación ( $t=0$ ),  $\tau_{1,1}$  consume todo  $WCET_1$ ,  $\tau_{2,1}$  consume el 50% del  $WCET_2$  y  $\tau_{3,1}$  consume todo  $WCET_3$ . Estas primeras activaciones se ejecutan a  $\alpha = 1$ , lo que trae como consecuencia un consumo de energía del 100%.

Después de un retraso debido a la planificación,  $\tau_{1,2}$  se le otorga una velocidad  $\alpha_{1,3} = 0,50$ , pudiendo ejecutar esta instancia con  $C_{i,k}=20$  ms, aun cuando por restricción de  $U_{ref}$ , debería ser 11,175 ms. Lo anterior se realiza debido a que se calcula el STEPA de  $\tau_{2,2}$  da como resultado que su próxima activación es en 80 ms, pudiendo conceder hasta este punto su ejecución. Logrando consumir un 25% de energía. Esto es un ejemplo del uso del criterio interTarea.

Por otro lado, el  $\tau_{1,3}$  usa el criterio intratarea, al adjudi-

carle un  $\alpha_{1,3} = 0,8947$  que implica un  $STU_{1,2}$  con  $C_{1,3} = 11,175$  ms, que por restricciones de la prueba lo terminará ejecutando a un 50%, dando 5,58 ms. Logrando consumir un 80% de la energía suministrada.

Las siguientes  $\tau_{i,k}$  irán ejecutándose a los  $\alpha_{i,k}$  correspondientes usando intra e interTarea respectivamente.

Los resultados arrojan que cada tarea actualiza sus  $C_{i,k}$  ya sea en la próxima instancia de ejecución o, al finalizar el hiperperíodo. Manifestándose el manejo del  $TV_{i,k}$  y su contribución al planificador realimentado.

Permite observar, que cada tarea está controlada por un lazo local, que podrá actualizar sus cómputos, ya sea en la próxima instancia de ejecución o al finalizar el hiperperíodo, manifestándose el manejo del  $TV_{i,k}$ .

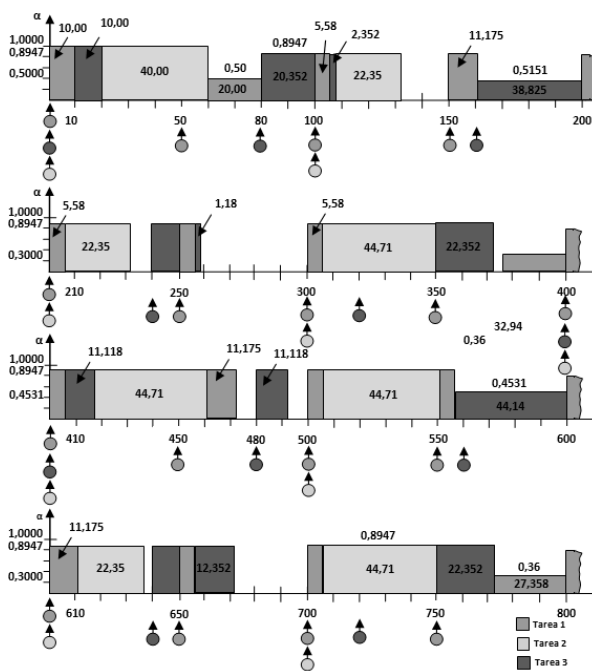


Fig. 5. Diagramas de planificación de las tareas con LPMFAE, escalamiento dinámico por tarea.  $T_1$ ,  $T_2$  y  $T_3$  consumen por restricción entre el 50% y 100% y de sus  $C_{i,k}$ .

El  $\alpha_{i,k}^{STU}$  está alrededor de 0,8947 siendo este el que garantiza la  $U_{ref}$  igual a 0,95. Lo que permite señalar que los lazos de control trabajan con sus referencias, dejando la flexibilidad de que localmente sean ajustados por el planificador en caso de existir condiciones para aprovechar el  $ST$ , teniendo un tope que es el  $TV_{i,k}$ . Lo anterior expresado, tiene un efecto en el  $U_T$ , debido a la contabilización de magnitudes que podrían alejarse de su referencia.

Se destaca que el comportamiento global interTarea y local inter e intraTarea, arrojan un consumo de energía del 44,70% al 90,00%, y 42,00% al 86,66%, respectivamente. Vale la pena destacar que el cálculo del consumo de la energía se realiza según lo estipulado en (Piguet 2006, Xia

y Sun, 2008, Zhu y Muller, 2006, Xia y col., 2009).

## 4 Conclusiones

Para el cálculo dinámico de la velocidad del procesador hay que tener presente que no todas las frecuencias de operación otorgadas por el procesador podrán ser usadas debido a restricciones de la aplicación y las características de tiempo real que posee. Por tal motivo, al definir el tiempo ocioso multifrecuencia o tiempo verde ( $TV_{i,k}$ ) desde las causas que lo producen, ayuda a acotar la distribución de los  $ST$  debido la ejecución de las tareas de control con criterios intraTarea e interTarea desde la variabilidad de los  $C_{i,k}$ , puntualizando que existe un límite de tiempo al cual están sometidas las  $\tau_{i,k}$  por restricciones de la aplicación caracterizado por una frecuencia de borde, que debe ser calculada, estando aquí la formalidad para hacerlo.

Se aprovecha las operaciones naturales de la planificación realimentada con técnicas de control multifrecuencia para la variabilidad de los tiempos de cómputo de  $\tau_{i,k}$ .

Como próximo paso es diseñar primero un núcleo de control con estas especificaciones y empotrarlas de manera real en el *PXA270*, para así medir su incidencia en el sistema y poder calcular el rendimiento a nivel de ejecución. Segundo, modificar un núcleo existente como el RTAI o Linux Empotrado, entre otros.

## 5 Agradecimientos

Los autores quieren agradecer a los entes involucrados que hicieron posible llevar a cabo los proyectos de investigación CI-020402-1739-11 y CmEPG 195-2010.

## Referencias

- Abdelzaher T, Diao Y, Hellerstein J, Lu C, Zhu X, 2008, Introduction to control theory and its application to computing systems. Performance modeling and engineering, Liu & Xia (eds.), Springer Science+Business Media. New York.
- Alfonsi A, Pérez J y Dunia E, 2013, Arquitectura de control multifrecuencia para el ajuste dinámico del consumo de energía en tareas de tiempo real, SABER, Vol. 25, N° 2, pp. 202-209.
- Alfonsi A, Pérez J, 2009. Sistema de control de tiempo real a una planta piloto compacta usando software libre, Universidad, Ciencia y Tecnología, Vol. 13, N° 52, pp. 189-198.
- Bhatti MK, Belleudi C, Auguin M, 2010, An inter-task real time DVFS scheme for Multiprocessor embedded systems, Proceedings of the Int. Conf. Design and Architectures for Signal and Image Processing. Edinburgh, UK, pp. 136-143.
- Buttazzo G, 2011, Hard real-time computing systems: Predictable scheduling algorithms and applications, 3 ed., Vol. 24, Springer Science+Business Media, Nueva York.
- Chang HW, Chang WH, Tsai CH, 2009, Integrated single-inductor buck-boost or boost-boost DC-DC converter with power-distributive control, International Conference on

- Power Electronics and Drive Systems, Taipei, pp. 1184-1187.
- Chantem TT, Hu XS, Lemmon MD, 2009, Generalized elastic scheduling for real-time tasks. *IEEE Transactions on Computers*, Vol. 58, N° 4, pp. 480-495.
- Choi Y, Chang N, Kim T, 2007, DC-DC converter-aware power management for low-power embedded systems, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, pp. 1367-1381.
- Hellerstein J, Diao Y, Parekh S, Tilbury D, 2004, *Feedback control of computing systems*, Wiley-Interscience, Hoboken.
- Hu X, Quan G, 2007, *Designing embedded processors – A low power perspective*, Henkel & Parameswaran eds., Springer, Berlin, pp. 219-229.
- Kim W, Shin D, Yun H-S, Kim J, Min SL, 2003, Performance evaluation of dynamic voltage scaling algorithms for hard real-time systems, *Low Power Electronics*, Vol. 1, pp. 1-11.
- López-Vallejo M, Huedo E, Sopena J, 2011, Informe de vigilancia tecnológica Madrid “Green IT: tecnologías para la eficiencia energética en los sistemas TI”, Fundación Madrid para el Conocimiento Velázquez, 76. E-28001, Madrid.
- Marvell, 2009, PXA270 Processor Electrical, Mechanical, and Thermal Specification, Marvell International Ltd.
- Moncusí M, 2005, Ahorro energético en la planificación de sistemas en tiempo real. Tesis de Doctorado. Universidad Politécnica de Cataluña, Barcelona.
- Niu L, 2011, Energy efficient scheduling for real-time embedded systems with QoS guarantee, *Real-Time Systems*, Vol. 47, N° 2, pp. 75-108.
- Noergaard T, 2013, *Embedded systems architecture: A comprehensive guide for engineers and programmers*. 2 ed., Elsevier Inc, Florida.
- Piguet C, 2006, *Low-power CMOS circuits: Technology, logic design and CAD tools*, CRC Press Taylor & Francis Groups, Boca de Ratón.
- Pillai P, Shin K, 2001, Real-time dynamic voltage scaling for low-power embedded operating systems. *Proceedings of the 18th ACM Symposium on Operating Systems Principles*. Banff, Alberta, Canada, pp. 89-102.
- Pont M, 2011, *Patterns for time-triggered embedded systems*, Association for Computing Machinery and Addison-Wesley, London.
- Rakhmatov D, 2008, Energy budget approximations for battery-powered systems with a fixed schedule of active intervals, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 16, pp. 985-998.
- Scordino C, 2007, *Dynamic voltage scaling for energy-constrained real-time systems*, Ph.D Thesis, Università Di Pisa. Pisa, Italy.
- Scordino C, Lipari G, 2007, A Resource Reservation Algorithm for Power Aware Scheduling of Periodic and Aperiodic Real-Time Tasks, *IEEE Transactions on Computers*, Vol. 55, N° 12, pp. 1509-1522.
- Shin D, Kim J, Lee S, 2000, Intra-task voltage scheduling for low energy hard real-time applications, *Special Issue of IEEE Design and Test of Computers*, October 2000.
- Shin Y, Choi K, Sakurai T, 2001, Power-conscious scheduling for real-time embedded systems design. *VLSI design*, Overseas Publisher Association, Vol. 12, pp.139-150.
- Urriza J, Páez F, Cayssials R, Orozco J, Schorb L, 2010, Low cost slack stealing method RM/DM, *International Review in Computers and Software*, Vol. 5, N° 6, pp. 660-667.
- Xia F, Sun Y, 2008, *Control and scheduling codesign-flexible resource management in real-time control systems*, Springer, Berlin.
- Xia F, Ma L, Zhao W, Sun Y, Dong J, 2009, Enhanced energy-aware feedback scheduling of embedded control systems, *Journal of Computers*, Vol. 4, N° 2, pp. 103-111.
- Yáñez R, Briceño MA, Alfonsi A, Yáñez J, 2013, Aproximación al redimensionamiento de la ingeniería desde la perspectiva del desarrollo sustentable y el pensamiento complejo, *Ingeniería Industrial, Actualidad y Nuevas Tendencias*, Vol. 3, N° 11, pp. 87-94.
- Zhu Y, Muller F, 2006, Exploring synchronous and asynchronous DVS for Feedback EDF Scheduling on an embedded platform, *ACM Transaction Embedded Computing Systems*, Vol. 5, June 2006, pp. 1-24.

**Recibido:** 01 de marzo de 2016

**Aceptado:** 20 de diciembre del 2016

**Alfonsi S. Alfonso S.:** Profesor Titular de la Escuela de Ingeniería y Ciencias Aplicadas, Departamento de Computación y Sistemas de la Universidad de Oriente (UDO, Venezuela). Ingeniero Electricista (UDO). Magister Scientiarum y Doctor en Ciencias mención Instrumentación de la Universidad Central de Venezuela. Investigador en el área de Sistemas de Tiempo Real y Tecnologías para la Sustentabilidad.

**Yáñez M. Raiza M.:** Profesora Titular de la Escuela de Ingeniería y Ciencias Aplicadas, Departamento de Sistemas Industriales de la Universidad de Oriente (UDO, Venezuela). Ingeniera Industrial y Magister Scientiarum en Ciencias Administrativas mención Gerencia General (UDO). Doctora en Gerencia de la Universidad de Yacambú. Investigadora y experta en el área de Desarrollo Sustentable y Pensamiento Complejo. Correo electrónico: raiza\_yanezm@udo.edu.ve

**Pérez R., Jesús A.:** Profesor Titular de la Universidad Politécnica Territorial del Estado Aragua "Federico Brito Figueroa" (Venezuela). Ingeniero Electricista de la Universidad de Carabobo. Magister Scientiarum y Doctor en Ciencias Mención Instrumentación de la Universidad Central de Venezuela. Investigador en sistemas de adquisición de datos y robótica. Correo electrónico: jesusaperez@gmail.com