

## **Gestión del tiempo ocioso dinámico para ajustar el consumo de energía en tareas de tiempo real integrando control multifrecuencia**

**Alfonso Salvador Alfonsi Sebastiani**  
Grupo de Investigación Arquitecturas de Sistemas de Control, Universidad de Oriente - Barcelona - Venezuela  
**alfonso\_alfonsi@udo.edu.ve**

**Jesús Alberto Pérez Rodríguez**  
Universidad Politécnica Territorial del Estado Aragua  
"Federico Brito Figueroa" - Venezuela,  
**jesusaperez@gmail.com**

**Emery Richard Dunia Amair**  
Postgrado en Instrumentación, Facultad de Ciencias,  
Universidad Central de Venezuela - Caracas - Venezuela  
**edunia@fisica.ciens.ucv.ve**

**Resumen:** Este trabajo tiene como objetivo ajustar la energía consumida por tareas de tiempo real críticas, producto de la variabilidad de sus tiempos de cómputo usando la integración del control multifrecuencia y la planificación realimentada. Se adaptaron e integraron técnicas dinámicas que manejan el tiempo ocioso debido al tiempo de ejecución en el peor caso, al factor de carga del procesador y el aprovechamiento del tiempo ocioso por estiramiento a la(s) próxima(s) activación(es), en una Técnica Dinámica Multifrecuencia para el Manejo del Tiempo Ocioso, alojada en un planificador realimentado

para el ahorro de energía, dirigido a procesadores que varían el voltaje de alimentación y frecuencia de operación. Además se tomó ventaja de las técnicas de control multifrecuencia dado que la gestión de recursos es formulada como un problema de control de sistemas de cómputo que especifica a cada tarea por un lazo de control que trabaja a su propio periodo de activación, diferentes a los requeridos en la referencia y respuesta del sistema (factor de carga total del procesador). Los resultados arrojan que el tiempo ocioso debido a variabilidad de los tiempos de cómputo se distribuye de forma natural por los lazos de control multifrecuencia, global y localmente, pudiendo llegar a un ahorro de energía del 61,04%, dando un valor agregado Intra e InterTarea. Además, sugiere un buen desempeño al contrastarlo con otras estrategias.

**Palabras clave:** ahorro de energía, control multifrecuencia, planificación de tiempo real, tiempo ocioso.

## **Dynamic slack time management to adjust the power in real time task integrating multirate control**

**Abstract:** This work aims to adjust the energy consumed by hard real-time tasks, product variability of their execution times using integration of multirate control and feedback scheduling. Adapt and integrate dynamic techniques that handle the slack time due to the worst case execution time, update processor load factor and stretching of task to the next activation were adapted and integrated into Multirate Dynamic Technique for Management of Slack Time, lodged in power aware real-time scheduler, to processors which vary the power voltage and operation frequency. In addition take advantage of multirate control techniques because resource management is formulated as a computer systems control problem that specifies each task by a control loop that works at their own activation period, other than those required in the reference and system response (processor total load factor). Results show that the slack time

due to variability of the execution time is distributed naturally by multirate control loop, globally and locally, and can reach an energy saving of 61.04%, giving intra- interTask added value. It suggests a good performance to contrast it with other strategies.

**Keywords:** multirate control, power aware, real-time scheduling, slack time.

## 1. Introducción

En la actualidad los sistemas empujados se encuentran desde los más simples aparatos domésticos hasta dispositivos móviles de alta complejidad. La mayoría tienen en común capacidades limitadas de operación, ya que dependen del uso de baterías para su funcionamiento, además de poseer características de tiempo real, lo que puede llegar a ser un punto crítico y limitar su rendimiento.

Frente a las restricciones de energía y tiempo existe una vía relacionada a la capa intermedia de software (núcleo de control, sistema operativo) llamada Escalamiento Dinámico de Voltaje y Frecuencia (*DVFS: Dynamics Voltage 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 & Quan, 2008; Piguet, 2006). Además, para explotar el DVFS se han desarrollado técnicas que manejan el tiempo ocioso (*ST: Slack Time*) originado cuando se ejecuta una tarea consumiendo menos de su tiempo de cómputo de peor caso.

Si bien es cierto que 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 (Sha, Abdelzaher, Arzén, Cervin, Baker, Burns, Buttazzo, Caccamo, Lehoczky & Mok, 2006), no obstante subyace que puede orientarse a planificadores con ahorro de energía (Scordino & Lipary, 2007; Xia

& Sun, 2008; Abdelzaher, Diao, Hellerstein, Lu & Zhu, 2008), conjugando características para el ajuste de energía sin vulnerar las temporales.

Los planificadores con ahorro de energía disponen de estrategias estáticas y dinámicas. En las estáticas el planificador debe seguir los pasos marcados, ejecutando cada tarea a la velocidad indicada sujetándose a un plan concebido fuera de línea. Entre las estrategias estáticas se encuentran: velocidad constante máxima de la tarea, velocidad fija de la tarea, métodos estocásticos y las basadas en trayectorias. Por otro lado, las dinámicas cambian la planificación en tiempo de ejecución o en línea. En su mayoría, emplean una de las siguientes técnicas: uso del ST por otras tareas, actualización de la utilización del procesador, estiramiento de la tarea al próximo tiempo de llegada y métodos de predicción.

Dentro de los planificadores dinámicos con ahorro de energía existe una tendencia que refuerzan las funcionalidades combinando técnicas e integrando disciplinas, como las suministradas por la teoría de control realimentado, llamada planificación realimentada con ahorro de energía (Xia, Ma, Zhao, Sun, Dong, 2009; Niu, 2011; Chantem, Hu & Lemmon, 2009; Zhu & Muller, 2006).

La planificación realimentada con ahorro de energía permite formular la gestión de recursos como un problema de control de lazo cerrado, tratando los sistemas de cómputo como un proceso controlado (Xia & Sun, 2008; Abdelzaher et al., 2008; Xia et al., 2009; Niu, 2011; Chantem et al., 2009; Zhu & Muller, 2006). Esta planificación manifiesta el compromiso entre las características temporales del sistema y la señal de control hacia las demandas al procesador, que dependerá del nivel del voltaje de alimentación y frecuencia de operación requerida por las tareas involucradas en el sistema, de forma que se minimice el consumo de energía y no afecte sus restricciones temporales.

En atención a lo anteriormente expuesto este trabajo tiene como objetivo la adaptación e integración de las técnicas dinámicas que manejan el ST, como son uso del ST por otras tareas, la actualización de la utilización del procesador

y el estiramiento de la tarea al próximo tiempo de llegada, en una llamada técnica dinámica multifrecuencia para el manejo del ST. La cual toma ventaja de las técnicas de control multifrecuencia (Cimino & Prabhakar, 2010; Salt, Casanova, Cuenca & Pizá, 2008; Salt & Albertos, 2005), dado que la gestión de recursos es formulada como un problema de control de sistemas de cómputo (Xia et al., 2009), que especifica a cada tarea por un lazo de control que trabaja a su propio periodo de activación, diferentes a los requeridos en la referencia y respuesta del sistema. Además, combina criterios de la distribución del ST aportada por una tarea ejecutada, a la misma (Intratarea) y a diferentes tareas (InterTarea), cada vez que sea planificada (Scordino & Lipary, 2007).

La organización del trabajo es la siguiente: en la Sección 2 se expone la especificación del sistema, definiendo la tarea de control necesaria para conceptualizar la técnica dinámica para el manejo del ST, mostrando el enfoque multifrecuencia adoptado, para luego presentar en la sección 3 las experiencias que demuestran el funcionamiento de la técnica unificada. En la sección 4, se encuentra el agradecimiento. Y, en la sección 5 y 6 se pueden encontrar las conclusiones y las respectivas referencias.

## **2. Especificación del Sistema**

Desde la óptica de los sistemas de control para los sistemas de cómputo se adopta una estructura con enfoque de control local y entrada de referencia local (Abdelzaher et al., 2008), incorporando técnicas multifrecuencia (Cimino & Prabhakar, 2010; Salt et al., 2008; Salt & Albertos, 2005), destacando que los controladores locales representan un conjunto finito de tareas de control (críticas, periódicas, aperiódicas, independientes, apropiables, no tienen restricciones de precedencia, ni presentan armonicidad en sus periodos de activación) que proporcionará resultados parciales guiados a diferentes

periodos, que a su vez serán totalizados, dando como resultado el factor de carga del procesador en un tiempo finito.

A continuación se define el modelo de la tarea de control que cumple con las especificaciones de tiempo y energía. Así como también la conceptualización, el enfoque basado en control multifrecuencia y la operacionalización de la técnica dinámica multifrecuencia para el manejo del ST.

## 2.1 Modelo de la Tarea de Control con las Especificaciones de Tiempo y Energía.

El modelo de la tarea de control 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  $k$ . Siendo estos:

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

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

donde  $C_{i,k}$  es el tiempo de cómputo,  $D_{i,k}$  el plazo de finalización y  $P_{i,k}$  el periodo de activación. Parámetros comunes en la literatura de tiempo real (Sha et al., 2006).

El parámetro  $\alpha_{i,k}$  es el factor de escalamiento que representa la normalización de la velocidad. Está acotado entre la frecuencia máxima ( $f_{max}$ ) y mínima ( $f_{min}$ ) de operación del procesador, calculada por:

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

donde  $f_{i,k}$  es la frecuencia actual de operación del procesador cuando ejecuta a  $\tau_{i,k}$ . El  $\alpha_{i,k}$  será igual a uno cuando la frecuencia de operación es máxima ( $\alpha^{fmax}$ ).

Además, en este trabajo se cumple:

$$C_i^{fmax} \leq C_{i,k} \leq C_i^{fmin} \quad (4)$$

$$D_{i,k} < P_{i,k} \perp D_{i,k} = D_i \quad \forall i \in (1, 2, \dots, N) \perp k(1, 2, \dots, M) \quad (5)$$

$$P_{i,k} = P_i \quad \forall i \in (1,2,\dots,N) \perp k(1,2,\dots,M) \quad (6)$$

donde los superíndices expresan el cargo de la variable. Por ejemplo  $C_i^{f_{\max}}$  se lee como el tiempo de cómputo de la tarea  $i$  ejecutado a frecuencia máxima del procesador  $f_{\max}$ . De aquí en adelante este  $C_i^{f_{\max}}$  es igual al tiempo de cómputo de peor caso de la tarea  $i$  (WCET <sub>$i$</sub> : Worst Case Execution Times).

Tomando en consideración lo que se establece en trabajos previos (Scordino & Lipary, 2007; Xia & Sun, 2008; Chantem et al., 2009) se logra relacionar el tiempo de cómputo actual a  $f_{i,k}$ , y el obtenido a  $f_{\max}$ , para obtener el factor de escalamiento mediante:

$$\alpha_{i,k} = WCET_i / C_{i,k} \quad (7)$$

Ahora bien, con todo lo anterior, se busca calcular el consumo de energía normalizado por  $\tau_{i,k}$  en un procesador tomando un rango de tiempo fijo. Para lo cual se dispone de la siguiente aproximación (Piguet, 2006; Xia & Sun, 2008; Zhu & Muller, 2006):

$$E_{i,k}(\alpha) = \alpha_{i,k}^2 \quad (8)$$

## 2.2 Conceptualización del Técnica Dinámica Multifrecuencia para el Manejo del ST

A continuación se desarrollan y clasifican las tres técnicas dinámicas consideradas para aprovechar el ST, definiendo la distribución en las  $\tau_{i,k}$ , combinando criterios IntraTarea e InterTarea, que son integradas en una sólo, llamada Técnica Dinámica Multifrecuencia para el Manejo del ST (TDMFMTO).

La primera, uso del ST debido al WCET, (Sha et al., 2006; Niu, 2011; Zhu & Muller, 2006; Urriza & Orozco, 2005), está orientada a calcular y entregar el ST ocasionado por la ejecución de una  $\tau_{i,k}$ , a su próxima ejecución. De aquí en adelante este ST se representa por las siglas *STWCET*, y está dado por:

$$STWCET_{i,k} = WCET_i - C_{i,k} \quad (9)$$

La segunda técnica es la debida a la actualización de la utilización del procesador (Scordino & Lipary, 2007; Xia & Sun, 2008; Abdelzaher et al., 2008). No obstante, como lo que se persigue es integrar con técnicas multifrecuencia, en este trabajo se emplea el factor de carga del procesador en vez de la utilización del mismo. De aquí en adelante se identifica con *STU*.

Por tanto, se define el factor de carga del procesador  $U_T$  como el parámetro indicativo de requerimiento del procesador para que se ejecuten un conjunto de  $\tau_{i,k}$  en un intervalo finito de tiempo  $IM$ . Dado por:

$$U_T = \frac{\sum_{i=1}^N \sum_{k=1}^{N_i} C_{i,k}}{IM} \leq 1 \text{ para } \alpha = 1 \quad (10)$$

donde  $N$  es el número máximo de tareas, y  $N_{\tau_i}$  es el número de  $\tau_{i,k}$  involucradas en  $IM$ . El  $N_{\tau_i}$  es definida por:

$$N_{\tau_i} = IM / P_i \quad \forall i \in (1, 2, \dots, N) \rightarrow N_{\tau_i} \in \mathbb{Z}^+ \quad (11)$$

La condición menor o igual a uno de (10), proviene de la prueba de planificabilidad, donde la menor cota para el  $U_T$  es uno, por lo tanto las  $T_i$  pueden utilizar el 100% del procesador y aún ser planificable. Esta prueba de planificabilidad está asociada al planificador dinámico *EDF (Earliest Deadline First)* (Liu y Layland, 1973) ampliamente conocido en la literatura de los sistemas de tiempo real (Sha et al., 2006).

Al hacer  $C_{i,k} = WCET_i$ , en (10), se obtiene el factor de carga total del procesador máxima  $U_T^{fmax}$ . También se necesita el factor de carga del procesador máxima debido a las instancias  $\tau_{i,k}$  de cada tarea, a ejecutarse, en su periodo de activación  $P_i$ , identificado como  $U_i^{fmax}$ . Estos parámetros ayudan a definir el factor de contribución normalizado ( $\sigma_i$ ) por cada  $T_i$ , como sigue:

$$\sigma_i = U_i^{fmax} / U_T^{fmax} \quad (12)$$



Además, debido al enfoque de control realimentado se debe asignar un factor de carga del procesador de referencia ( $U_{ref}$ ), con el cual se obtienen, de la ecuación (13), los factores de carga del procesador por cada tarea ( $U_{refi}$ ), y de (14) los tiempos de cómputo asociados a estos  $U_{refi}$ .

$$U_{refi} = \sigma_i U_{ref} \quad (13)$$

$$C_{i,k}^U = U_{refi} IM / N\tau_i \quad \forall i \in (1, 2, \dots, N) \perp N\tau_i \in \mathbb{Z}^+ \quad (14)$$

Definiendo al  $STU_{i,k}$  por:

$$STU_{i,k} = C_{i,k}^{f_{max}} - C_{i,k}^U \quad (15)$$

La tercera técnica no sólo aborda la condición básica que consiste en medir cuál es el intervalo desde la finalización de una  $\tau_{i,k}$  hasta su próxima activación o hasta su  $D_i$ , para adjudicársela a su próxima  $\tau_{i,k}$  planificada (Scordino & Lipary, 2007), con la condición de consumir todo su  $WCET_i$ .

Sino también se extiende a otra situación adicional que toma en consideración los próximos arribos de todas las  $\tau_{i,k}$  planificadas. El  $ST$  debido a estas dos causas se define como el  $ST$  por Estiramiento a la(s) Próxima(s) Activación(es), o por siglas *STEPA*.

Entonces, tomando en cuenta la primera condición, el *STEPA* puede ser calculado como:

$$STEPA_{i,k} = I_{i,k} \quad (16)$$

siendo  $I_{i,k}$  la distancia al plazo de finalización de  $\tau_{i,k}$ , dado por

$$I_{i,k} = D_i - C_{i,k} \quad (17)$$

recordando que  $D_i$  es el plazo de finalización de una tarea y  $C_{i,k}$  el respectivo tiempo de cómputo.

Merece destacar que el *STEPA* debe cumplir con la condición  $0 \leq STEPA_{i,k} \leq I_{i,k}$ .

Ahora bien, cuando se considera la segunda condición referente a los próximos arribos de las tareas planificadas, siendo  $D_i=P_i$ , se debe tener claro que distancia al próximo arribo más cercano de una instancia planificada, está dada por  $I_{\tau_{i,k}}$ , entonces:

$$\exists STEPA_{i,k} \in [T_{Fi,k}, P_{abz}] \mid STEPA_{i,k} = I_{\tau_{i,k}} \wedge I_{\tau_{i,k}} = P_{abz} - T_{Fi,k} \quad \forall z \in (1, 2, \dots, N) \\ \wedge k \in (1, 2, \dots, M) \quad (18)$$

$T_{Fi,k}$  es el tiempo de finalización de  $\tau_{i,k}$ , calculada por

$$T_{Fi,k} = r_{i,k} + C_{i,k} \quad (19)$$

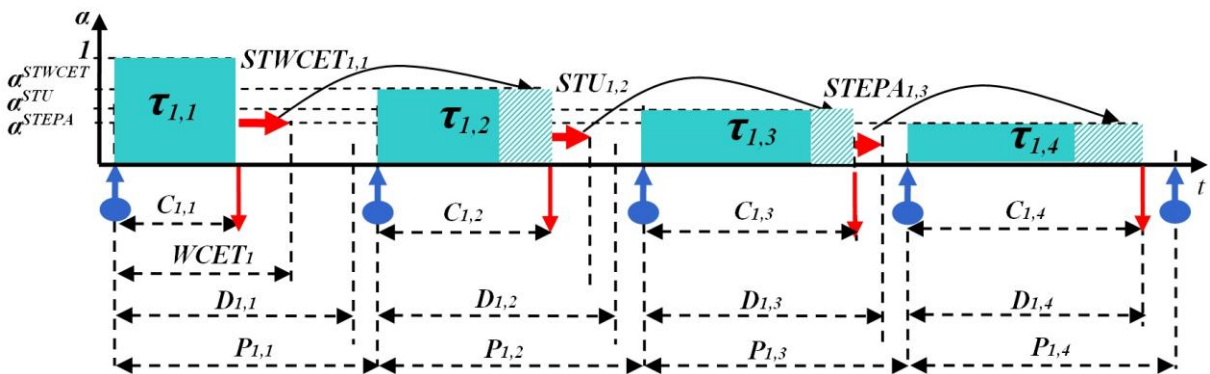
$P_{abz}$  es el próximo tiempo de arribo más cercano de una  $\tau_{i,k}$  planificada o el próximo período absoluto de

$\tau_{z,k}$ , siendo el subíndice  $z$  la identificación de la tarea planificada lista para su ejecución.

$$P_{abz} = (k - 1)P_z + P_z \quad (20)$$

Se debe cumplir la siguiente condición  $T_{Fi,k} < P_{abz}$

En la Figura 1 se presenta un ejemplo de como se distribuye el ST según el criterio IntraTarea, con las técnicas definidas hasta el momento.



**Figura 1.** Tarea de Control  $T_i$  aprovecha el ST para variar sus  $\tau_{i,k}$  según el STWCET, STU y STEPA tomando en cuenta la próxima activación o su  $D_i$

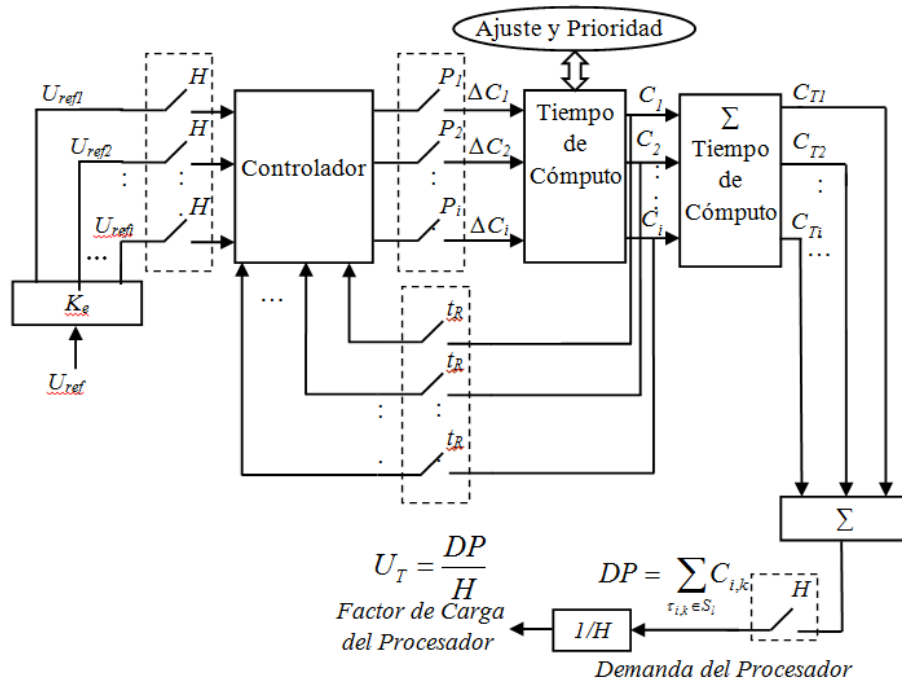
**Nomenclatura:**  $\alpha^{STWCET}$ : factor de escalamiento debido al *STWCET*.  $\alpha^{STU}$ : factor de escalamiento debido al *STU*.  $\alpha^{STEPA}$ : factor de escalamiento debido al *STEPA*

Inicia con  $\tau_{1,1}$  ejecutándose a  $\alpha^{fmax}$ , y a un  $C_{1,1}$  menor que su  $WCET_{1,1}$ , por lo que un  $STWCET_{1,1}$  es calculado. En  $\tau_{1,2}$  se utiliza este *ST*, incrementando su  $C_{1,2}$ , causado por el cambio del factor de escalamiento debido al *STWCET* ( $\alpha^{STWCET}$ ); a su vez, existe un  $STU_{1,2}$ , que es considerado por  $\tau_{1,3}$ , pudiendo ajustar al factor de escalamiento debido al *STU* ( $\alpha^{STU}$ ). En  $\tau_{1,3}$  se activa el *STEPA*, ocasionando que  $\tau_{1,4}$  se ejecute a un factor de escalamiento debido al *STEPA* ( $\alpha^{STEPA}$ ), por consiguiente varíe su  $C_{1,4}$ .

## 2.3 Enfoque de Control Multifrecuencia Realimentado del Factor de Carga del Procesador total $U_T$

Para sistematizar la combinación de las técnicas dinámicas para el manejo del *ST*, detalladas en secciones anteriores, se introduce una estructura funcional con enfoque de control local y entrada de referencia local (Figura 2) (la constante  $K_e$  divide la señal de error de acuerdo a ganancias por cada lazo de control local) incorporando un control con muestreo no convencional o multifrecuencia (MF) del factor de carga del procesador total ( $U_T$ ), el cual será muestreado cada periodo global o hiperperiodo  $H$ , dado por

$$H = m.c.m(P_i) \quad (21)$$



**Figura 2.** Estructura funcional del control multifrecuencia para el  $U_T$ , donde el lazo de control local, las entradas locales ( $U_{refi}$ ) están muestreada a  $H$ , el cambio en los tiempos de cómputo ( $\Delta C_i$ ) a  $P_i$ . Los  $C_{i,k}$ , son realimentados a un tiempo de muestreo  $t_R$ , y totalizados ( $C_{Ti}$ ) para obtener la demanda del procesador ( $DP$ ), para luego calcular el  $U_T$  a  $H$ .

La estructura funcional del control MF permite ajustar la velocidad a un conjunto de instancias de modo global (cada hiperperíodo  $H$ ) y local (cada período de ejecución  $P_i$ ), ambas acotadas en  $H$ .

Para el primer caso, global cada  $H$ , se definen los muestreadores  $t_R$  ubicados en los lazos de realimentación local igual a  $H$ , y se emplea a la entrada del controlador una operación multifrecuencia de expansión o *Expand* (*OPMF\_Expand*), ya que se necesita que la señal de salida del controlador aumente su tiempo de muestreo y que se disponga cada  $H$ , dejando las otras secuencias a  $P_i$  en cero. A la salida del sumador, una operación multifrecuencia de salto o *Skip* (*OPMF\_Skip*), debido a que se necesita la demanda del procesador ( $C_{Ti}$ ) ocasionada por las entregas parciales de cada tarea a  $H$ , no importando las totalizaciones parciales a  $P_i$ .

El segundo caso, local cada período de ejecución  $P_i$ , admite variar la velocidad en cada instancia  $\tau_{i,k}$ . Es decir, las instancias se pueden ejecutar a velocidades diferentes durante  $H$ , al definir los muestreadores  $t_R$  de los lazos de realimentación local igual a los  $P_i$ , aplicando a la referencia una operación multifrecuencia de retención u *Hold* (*OPMF\_Hold*), ya que se necesita que esta señal sea mantenida y disponible cada  $P_i$ . En la salida del sumador se aplica una operación multifrecuencia de salto, como se establece en el primer caso.

Con lo anterior es posible, darle a las  $\tau_{i,k}$  la capacidad de variar sus  $C_{i,k}$  de manera local y global, ya que el *STU* y *STWCET* se calculan de forma natural debido a la acción del lazo de control local multifrecuencia. No obstante, la estructura funcional *MF* debe ser reforzada por una entidad funcional llamada “Ajuste y Prioridad” para sacar provecho de la técnica *STEPA*, junto al algoritmo de planificación de tiempo real EDF.

## **Módulo Ajuste y Prioridad**

En este módulo se establecen la política de planificación que dará permiso a las tareas para su ejecución, el factor de escalamiento de borde, y el procedimiento para activar el *STEPA*.

Como política de planificación se elige el *EDF* (Liu y Layland, 1973) el cual ejecuta en cada momento la tarea cuyo plazo de finalización (absoluto) está más próximo. Se justifica ésta selección porque puede ser usado para la planificación de tareas periódicas como aperiódicas ya que no hace uso de la periodicidad de las tareas para la planificación. Adicionalmente, su desempeño y uso en las arquitecturas realimentadas (Hu & Quan, 2007; Scordino & Lipari, 2007; Zhu & Muller, 2006; Xia & Sun, 2008; Xia, , Ma, Zhao, Sun & Dong, 2009) está comprobado, y se adapta a las características funcionales establecidas en este trabajo.

Es importante señalar, la adopción aquí del *EDF* no restringe usar otras políticas de planificación, ya sean estáticas o dinámicas.

Otro punto a tener en consideración son las debidas a restricciones de planificación, ya que es posible que las  $\tau_{i,k}$  no logren ser ejecutadas en todas las velocidades ofrecidas por el procesador. Por tal motivo se define el factor de escalamiento de borde  $\alpha^{borde}$  por la relación que existe entre los factores de carga del procesador total a frecuencia máxima ( $U_T^{fmax}$ ) y el de referencia ( $U_{ref}$ ).

El  $\alpha^{borde}$  establece el rango real de operación para una aplicación, que propondrá un rango de  $C_i$  para ejecutarse sin violar las restricciones temporales, permanecer dentro de la utilización de referencia del procesador y ajustar el consumo de energía. Está dado por:

$$\alpha^{borde} = U_T^{fmax} / U_{ref} \quad (22)$$

Con respecto al procedimiento para activar el STEPA, desarrollado en la sección 2.2, se entrega en la próxima sección, en el algoritmo 3 el método *DeterSTEPA()* para su implementación.

## 2.4 Operacionalización de TDMFMTO

Para implementar la técnica se toma ventaja de codificación de programas en lenguajes con capacidad orientada objeto y librerías de aplicación para integrar clases en la estructura funcional del control MF

Lo primero es establecer los ambientes temporales de la aplicación y del procesador. De la aplicación, son conocidos los tiempos de cómputo a máxima frecuencia  $C_i^{fmax}$  o  $WCET_i$ , los  $P_i$ , y sus  $D_i$ . Así como también el Factor de Carga de referencia  $U_{ref}$ .

Se calcula, fuera de línea, lo siguiente:

1. El hiperperiodo  $H$  con (21).
2. El número de instancias  $N_{Ti}$  que se deberían ejecutar durante  $H$  con (11).
3. Con (12) el factor de carga de contribución de la referencia local  $\sigma_i$ .
4. Con (10) el factor de carga a máxima frecuencia,  $U_T^{fmax}$ .
5. El factor de escalamiento del procesador de borde  $\alpha^{borde}$  con (22).

Por parte del procesador se conocen los parámetros operación como los diferentes niveles de frecuencia y voltaje de alimentación.

A continuación se presentan los Algoritmos 1 y 2 que detallan los prototipos de las clases *Sistema* y *Tarea*. También se describen los términos que no han sido especificados, hasta ahora.

**Términos:**

**nea**

Niveles de escalado dependiente de la aplicación.

**errorlocal**

Error del lazo realimentado local.

**Listo**

Estado de Listo.

**Ejec**

Estado de Ejecución.

**C<sub>st</sub>**

Tiempo de cómputo sintético o analítico.

**D<sub>ab</sub>**

Plazo de finalización absoluto de  $T_i$ .

**r<sub>i,k</sub>**

Tiempo de invocación o activación de  $\tau_{i,k}$ .

**CTL**

Sumatoria de los tiempos de cómputo de cada  $T_i$ .

**deltaC**

Cambio del tiempo de cómputo por instancia.

**alfa**

Factor de escalamiento sintético o analítico.

**Cant**

Tiempo de cómputo anterior o  $C_{i,k-1}$ .

**STEPA**

ST próxima activación.

Algoritmo 1: Prototipo clase Sistema	
1	Inicio_clase
2	Sistema(float q=0,float n=0) //q=Uref y n=H
3	float Factor_Carga() //Ecuación (11)
4	Fin_clase Sistema

Algoritmo 2: Prototipo clase Tarea	
1	Inicio_clase
2	//Definición de variables
3	float C,W CET,P,D,Ut,nea,Cant,Uref,H,Cref,errorlocal,Cst,STEPA
4	boolean Listo,Ejec
5	Tarea(float a=0,float b=0,.....)
6	float referlocal() //Referencia local
7	float error_local() //Establece error local para cada lazo i
8	float TareaCtrol() //Controlador local
9	float Compst() //Cálculo local $C_{i,k}$
10	float CambioFV() //Ajuste velocidad depende del procesador
11	float EjecutarComp() //Ejecución $T_i$ depende planificador
12	float CalculoCompT() //Módulo Acumulador
13	float DeterSTEPA() //ST próxima activación
14	Fin_clase Tarea

El Algoritmo 3 están los métodos que definen al TDMFMTO, error\_local(), Compst(), CalculoCompT() y DeterSTEPA().

Algoritmo 3: Definición clase Tarea con TDMFMTO	
1	float Tarea::referlocal(flota m, flota o) //Método referencia local
2	Inicio_definicion clase
3	$N_{Ti}=m; U_T^{fmax}=0$
4	//Significado de notaciones
5	float $\sigma, K_e, U_{refi}$
6	Cálculo $\sigma_i$ //Factor contribución normalizado $xT_i$ ecuación (12)
7	$U_{refi} = \sigma_i U_{refi}$ //Referencia local ecuación (13)
8	$K_e = \sigma_i H / N_{Ti}$ //Ganancia de referencia local
9	$C_{refi} = U_{refi} K_e$ //Tiempo de cómputo $x T_i$ de referencia
10	Fin_definición
11	
12	float Tarea::errorlocal() //Método de la clase tarea error local
13	Inicio_definicion clase
14	Si Op=global entonces //Operación Global
15	errorlocal=OpMF_Expand( $C_{ref}^H - C_{st}$ )
16	de lo contrario //Operación Local
17	$C_{ref}^{Pi} = OpMF\_Hold(C_{ref}^H)$
18	errorlocal= $C_{ref}^{Pi} - C_{st}$
19	Fin_Si
20	Fin_definicion
21	
22	float Tarea::Compst() //Método de la clase tarea local $C_{i,k}$



23	Inicio_definicion clase
24	flota deltaC,alfa
25	deltaC = this → TareaCtrl
26	$C_{st}^{Pi} = Cant + deltaC$
27	STEPA <sub>st</sub> = this → DeterSTEPA
28	$C_{st}^{Pi} = C_{st}^{Pi} + STEP A$
29	alfa=WCET/ $C_{st}^{Pi}$ //Factor de escalamiento analítico
30	Si Op=global entonces //Operación Global a H
31	$C_{st} = OpMF\_Skip(C_{st}^{Pi})$
32	de lo contrario //Operación Local a Pi
33	$C_{st} = C_{st}^{Pi}$
34	Fin_Si
35	Fin_definicion
36	
37	float Tarea::DeterSTEPA() //Tiempo ocioso próxima activación
38	Inicio_definicion clase
39	float Tf <sub>ik</sub> ,r <sub>i,k</sub>
40	$Tf_{ik} = r_{i,k} + C_{st}$ //Tiempo de finalización ecuación (19)
41	Si Tf <sub>i,k</sub> < Dab entonces STEP A = Tf <sub>i,k</sub> - Dab
42	de lo contrario STEP A = 0,00
43	Fin_Si
44	Fin_definicion
45	
46	float Tarea::CalculoCompT()
47	Inicio_definicion clase //Módulo Acumulador
48	float CTL
49	$CTL = Cant + C_{st}$
50	$Cant = CTL$
51	Return CTL //Desde programa principal se aplica Skip a H
52	Fin_definicion

Las otras definiciones de la clase Tarea, como *TareaCtrl()*, *CambioFV()*, *EjecutarComp()* son realizadas a partir de las características del algoritmo de control, procesador utilizado, y la política de planificación, que para este caso es el *EDF* (Liu y Layland, 1973).

### 3. Discusión de los Resultados

Para evaluar el funcionamiento y el comportamiento de las tareas a nivel de la planificación y nivel energético se materializó del TDMFMTO en códigos desarrollados en C++, tomando ventaja de la programación orientada objeto y librerías de aplicación para integrar clases, de la estructura funcional del control MF, la política de planificación *EDF*, el factor de escalamiento de borde y el procedimiento para activar el STEPA. Así como también, el cálculo del consumo de energía normalizado ( $E_{i,k}$ ) con la aproximación dada por (8).

Con respecto al procesador se toma como base a un procesador hipotético que varía linealmente su voltaje y frecuencia de operación, cuyas características se presentan en la Tabla 1. Vale la pena mencionar que el tiempo de conmutación de una frecuencia a otra, es típicamente en microsegundos (entre 30  $\mu$ s y 150  $\mu$ s), mientras los tiempos de cómputo de las tareas es en milisegundo. Por lo tanto, comparando éstos tiempos, el tiempo de conmutación puede ser ignorado (Rizvandi, Venkatachalam & Franz, 2005; Piguet, 2006; Rakhmatov, 2008; Chang, Chang & Tsai, 2009; Taheri, & Zomaya, 2011).

Parámetros	Mínimo			Máximo	
<b>Frecuencia (MHz)</b>	150,00	400,00	600,00	800,00	1.000,00
<b>Voltaje (V)</b>	0,750	1,00	1,30	1,60	1,80
<b><math>\alpha</math></b>	0,150	0,40	0,60	0,80	1,00
<b>Energía = <math>\alpha^2</math></b>	0,023	0,16	0,36	0,64	1,00

**Tabla 1.** Características del Procesador variable de voltaje y frecuencia Hipotético

Para el análisis de algoritmos de planificación de tareas con ahorro de energía existen cuatro condiciones a saber: debido a efectos de la variabilidad de los tiempos de cómputo, al factor de carga local o total del procesador, los producidos por el número de tareas, y los cambios en el procesador variable de voltaje y frecuencia (Shin, Kim, Jeon, Kim & Min, 2003; Kim, Shin, Yun, Kim & Min, 2003; Bhatti, Belleudy & Auguin, 2010).

Hay trabajos que se pronuncian y tienen un consenso respecto a la utilización de estas condiciones (Shin, Choi, & Sakurai, 2001; Moncusí, 2005; Choi, Chang & Kim, 2007; Xian, Lu & Li, 2008; Bhatti, Belleudy & Auguin, 2010). Una de ellas, el efecto que produce variar el número de tareas involucradas en el sistema no presenta diferencias significativas. Si bien es cierto, que con mayor número de tareas hay oportunidades para generar más ST y la posibilidad de utilizarlos, esto no cambiará que la utilización o factor de carga del procesador debe ser menor e igual a uno para garantizar la planificación y funcionamiento temporal de la aplicación. Esto permite señalar que el principal determinante de las variaciones en el consumo de energía sigue siendo la carga de trabajo real. Otro pronunciamiento referido al rendimiento respecto al ahorro de energía, es que tiende a ser mejor cuanto mayor sea el número de niveles de voltaje y frecuencia disponibles de operación por parte del procesador, ocurriendo lo contrario cuando ese número disminuye. Por tanto estos efectos son dirigidos por el procesador y no por el enfoque usado en éste trabajo. Por estas razones para las experiencias se tomaron las dos primeras condiciones, los efectos de la variabilidad de los tiempos de cómputo y los debidos al factor de carga local o total del procesador.

Se desarrollan dos experiencias que muestran Una con  $\alpha = 1$  en su primer  $H$ , para luego variar  $\alpha$  en los sucesivos hiperperiodos (modo global); la segunda haciendo que  $\alpha$  varíe cada vez que se instancie una tarea (modo local).

En la Tabla 2 se muestra un conjunto de tareas para realizar pruebas comparativas (*benchmark*), propuestas por Shin, Choi & Sakurai (2001), utilizadas para evaluar los algoritmos de Moncusí (2005), Choi, Chang & Kim (2007), Rakhmatov (2008) y Chang, Chang & Tsai (2009).

Tarea	P (ms)	D (ms)	WCET (ms)
<b>T1</b>	50	50	10
<b>T2</b>	80	80	20
<b>T1</b>	100	100	40

**Tabla 2.** Pruebas Comparativas (Shin, Choi & Sakurai, 2001)

Para una mejor comprensión de las experiencias realizadas se usa el diagrama de planificación y escalado de velocidad, cuyo objetivo es mostrar el tiempo de dedicación previsto para las diferentes  $\tau_{i,k}$  a lo largo de un tiempo determinado y el factor de escalamiento de velocidad  $\alpha$  a la cual es sometida. Se destacan los tiempos de activación de cada  $\tau_{i,k}$  a lo largo del tiempo.

### 3.1 Prueba Uno: Convencional $\alpha = 1$ Sin Escalamiento, y Con Escalamiento Global de Velocidad Lineal

En la Figura 3 se observa que en  $t = 0$  con  $U_{ref} = 0,95$ , y aplicando la *OPMF\_Expand* ( $H$  a  $t_i$ ), las tareas son ejecutadas, arrojando los siguientes valores:

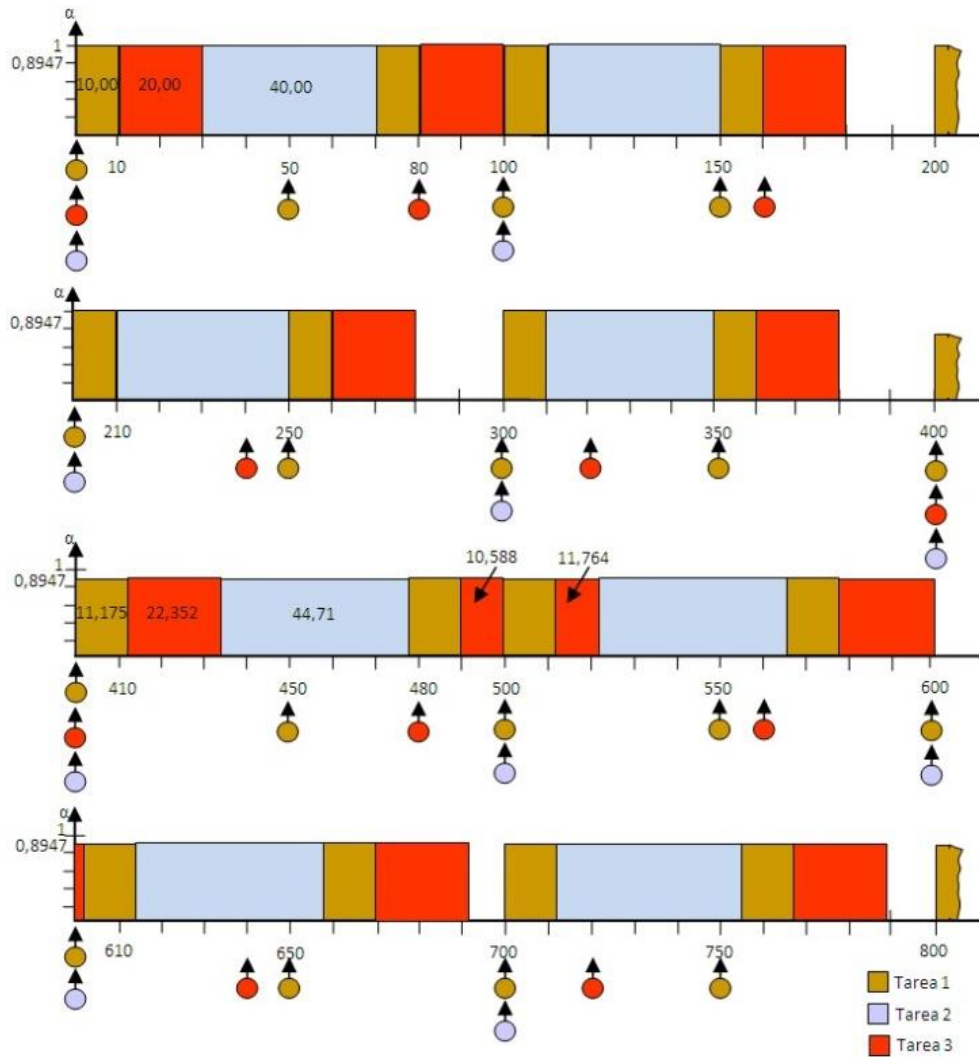
$$C^{st} = \{ 10,00 \ 20,00 \ 40,00 \} \text{ ms a } \alpha = 1$$

Vale la pena mencionar que el subíndice *st* indica que los cálculos son sintéticos o analíticos.

A 400 ms se aplica la *OPMF\_Skip*, resultando:

$$E(\alpha) = \alpha^2 = 1 \ C_{\tau_i} = \{ 80,00 \ 100,00 \ 160,00 \} \text{ ms}$$

De (10) se calcula, su numerador la ms, arrojando



**Figura 3.** Comportamiento de las tareas bajo EDF con escalamiento global,  $\alpha=1$  y variable lineal. Consumen el 100% de sus  $C_{i,k}$ .

A partir de 400 ms inicia el segundo hiperperiodo, la  $U_{ref}=0,95$ , donde el tiempo de cómputo de referencia local  $C_{refi}$  toma los siguientes valores { 11,175 22,352 44,710 }ms.

Aunque estas magnitudes son causa de la  $U_{ref}=0,95$ , por razones de las características temporales de las tareas, para este segundo hiperperiodo se ejecutarán las tareas a:

$$C_i^{st} = \{ 11,175 \ 22,352 \ 44,710 \}ms \text{ a } \alpha^{st} = 0,8947$$

Al aplicar *OPMF\_Skip* a 800 ms, da como resultado:

$$E(\alpha) = \alpha^2 = 0,80 \ C_{Ti} = \{ 89,40 \ 111,76 \ 178,84 \}ms$$

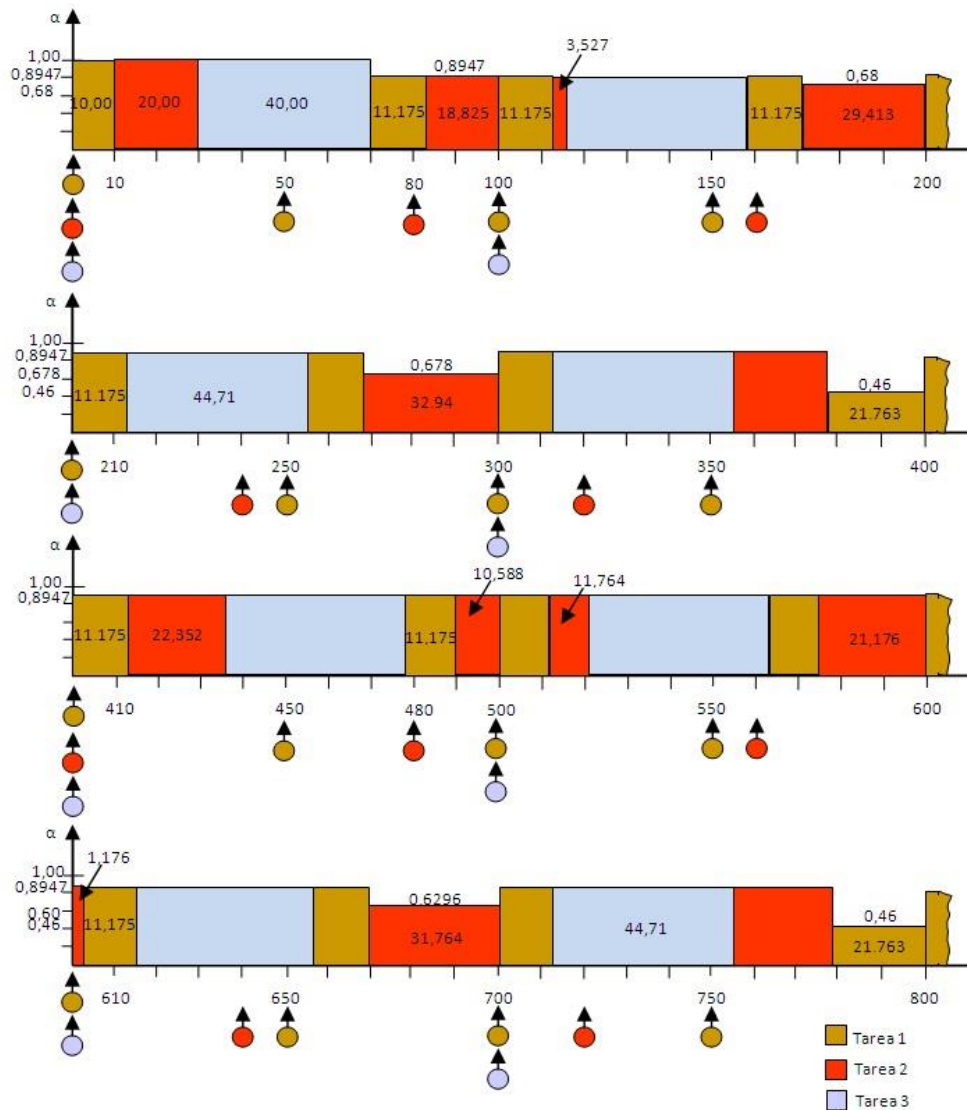
$$DP = 380,00 \ ms \ U_T = 0,95$$

Se alcanza la referencia, por lo tanto, el comportamiento del sistema es similar en el resto de los hiperperiodos, quedando una energía normalizada de  $E(\alpha)=0,8$ .

### **3.2 Prueba Dos: Con Escalamiento Local de Velocidad Lineal**

Aquí, la ejecución de cada instancia de una tarea se ejecuta durante un hiperperiodo a diferentes velocidades. En la Figura 4 se observa que en  $t=0$  con  $U_{ref}=0,95$  y aplicando la *OPMF\_Hold* ( $H$  a  $t_i$ ), las tareas son ejecutadas, manifestándose los siguientes valores:

$$C^{st} = \{ 10,00 \ 20,00 \ 40,00 \}ms \text{ a } \alpha = 1$$



**Figura 4.** Comportamiento de las tareas bajo EDF con escalamiento local variable lineal, y consumen el 100% de sus  $C_{i,k}$ .

Para 70 ms las tareas se les adjudica el  $STU$ , y se ejecutan a  $\alpha=0,8947$  que garantiza la  $U_{ref} = 0,95$ . En 170,587 ms inicia ejecución  $\tau_{2,3}$ , produciéndose lo siguiente:

$$e_{12,3} = C_{ref2} - C_{2,2} = 0,00 \text{ ms}; \Delta C_{2,3} = 0,00 \text{ ms}; C_{2,3}^{st} = 22,352 \text{ ms}$$

Luego se ejecuta el criterio de la próxima activación (STEPA), quedando ahora:

$$C_{2,3}^{st} = 29,413 \text{ ms}; \alpha_{2,3}^{st} = 0,68 \rightarrow \text{Ee ejecuta } \tau_{2,3}; E_{2,3} = 0,4624$$

Lo anterior se repetirá en las instancias  $T_{2,4}$ ,  $T_{1,8}$ ,  $T_{2,9}$  y  $T_{1,16}$

El comportamiento en el primer hiperperiodo se resume así:

$$C_{T_1}^H = \{ 98,813 \ 127,057 \ 174,13 \}; DP^H = 400,00; U_T = 1,00$$

$$E_{i,k}(\alpha) = \left\{ \begin{array}{cccccccc} 1,00 & 0,80 & 0,80 & 0,80 & 0,80 & 0,80 & 0,80 & 0,212 \\ 1,00 & 0,80 & 0,4624 & 0,4624 & 0,80 & & & \\ 1,00 & 0,80 & 0,80 & 0,80 & & & & \end{array} \right\}$$

El comportamiento en el segundo hiperperiodo arroja lo siguiente:

$$C_{T_2}^H = \{ 99,99 \ 121,17 \ 178,84 \}; DP^H = 400,00; U_T = 1,00$$

$$E_{i,k}(\alpha) = \left\{ \begin{array}{cccccccc} 0,80 & 0,80 & 0,80 & 0,80 & 0,80 & 0,80 & 0,80 & 0,212 \\ 0,80 & 0,80 & 0,80 & 0,396 & 0,80 & & & \\ 0,80 & 0,80 & 0,80 & 0,80 & & & & \end{array} \right\}$$

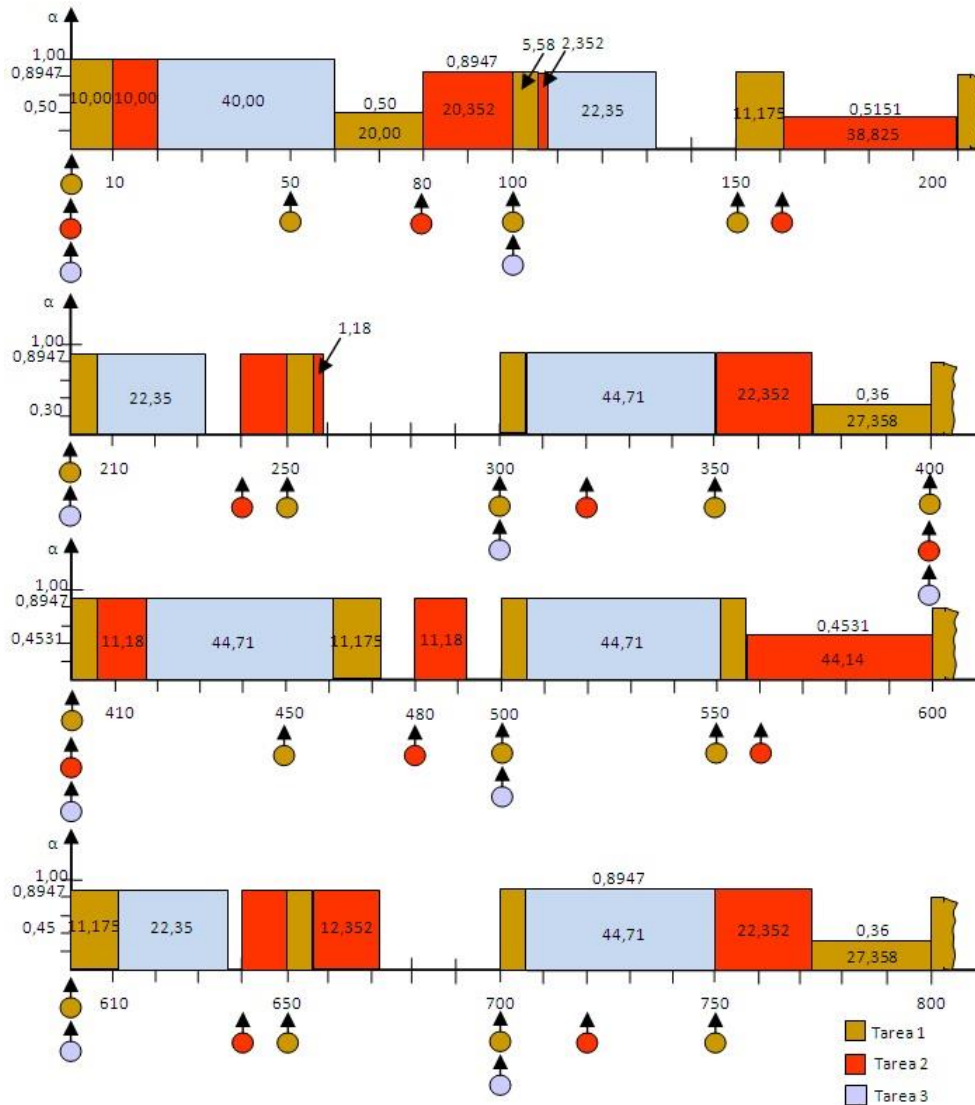
Se destaca que la actividad del *STU* y *STWCET* es operada naturalmente por el lazo realimentado, activando la utilización del *STEPA* cuando las condiciones se cumplen. Cada tarea está controlada por un lazo local que podrá actualizar sus tiempos cómputos, ya sea en la próxima instancia de ejecución o al finalizar el hiperperiodo. El  $\alpha_{i,k}^{st}$  está alrededor de 0,8947, siendo este el que garantiza la  $U_{ref} = 0,95$ .

La Figura 5 muestra el comportamiento de las tareas cuando consumen entre 50% y 100% de sus  $C_{i,k}$ , con escalamiento lineal local. 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 un ST.

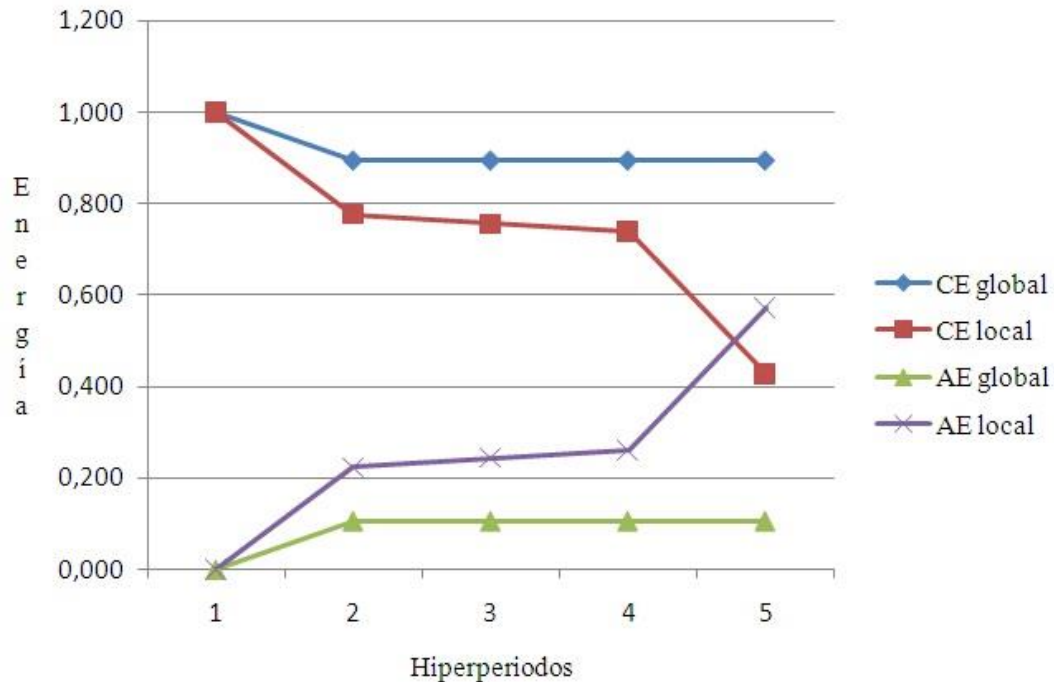
Adicionalmente, en la Figura 6, se visualiza el comportamiento de los modos de trabajo de la técnica, global y local, ejecutando las tareas variando sus  $C_{i,k}$  aleatoriamente, al entrar en cada hiperperiodo, con escala lineal. Arroja, que el ahorro de energía global (*AE global*), desde el segundo hiperperiodo en



adelante, alcanza su referencia, quedando en 10,60%, con escala lineal. El comportamiento del ahorro local ( $AE_{local}$ ), es creciente desde su inicio, esto es debido al aumento progresivo de las demandas del procesador o  $DP$ , al utilizar los  $STEPS$  para alargar sus  $C_{i,k}$ . Entonces, a medida que aumentan los  $ST$  estos son consumidos por las tareas en sus próximas instancias debido al  $STWCET$  y  $STU$ , o en la ejecución actual de la instancia de la tarea como lo refiere  $STEPS$ .



**Figura 5.** Diagramas de planificación de las tareas con EDF, escalamiento lineal local, T1, T2 y T3 consumen entre 50% y 100% de sus  $C_{i,k}$ .



**Figura 6.** Comportamiento de los modos de trabajo, global y local, variando los tiempo de cómputo aleatoriamente, en cada hiperperiodo, con escala lineal.

### 3.3 Comparación de TDMFMTO con otros planificadores

Adicionalmente, se comparó el ahorro de energía obtenido con TDMFMTO alojado en estructura funcional del control MF (Figura 2), y otras técnicas que usan los planificadores de Sha et al. (2006), Shin, Choi & Sakurai (2001), Moncusí (2005) y Zhu & Muller (2006), registrándose en la Tabla 3 los resultados arrojados,

Planificador	Ahorro de Energía Consumo 100% $C_{i,k}$	Ahorro de Energía Consumo 50% $C_{i,k}$
<b>EDF (Sha et al., 2006)</b>	0,000	0,000
<b>DVS-EDF (Zhu &amp; Muller, 2006)</b>	0,108	0,600
<b>PLMDP (Moncusí, 2005)</b>	0,082	0,700
<b>LPFPS (Shin, Choi &amp; Sakurai, 2001)</b>	0,076	0,548
<b>Estructura funcional del control MF +TDMFMTO</b>	0,133	0,583

**Tabla 3.** Comparación estructura funcional del control *MF+TDMFMTO* con otros planificadores, consumiendo el 100% y 50% de sus  $C_{i,k}$  con escala lineal.

Se observa, cuando se consume el 100% de sus  $C_{i,k}$ , la estructura funcional del control *MF+TDMFMTO* y el *DVS-EDF*, planificadores realimentados, arrojan el mayor ahorro de energía de 13,30% y 10,82%, debido a la búsqueda de su referencia, manejando de forma natural el *STWCET* y *STU*, ocasionada por el lazo realimentado, dejando algunos ajustes para el *STEPA*.

Sin embargo, cuando consume el 50% de sus  $C_{i,k}$ , el ahorro de la estructura funcional del control *MF+TDMFMTO* es del 58,30%, por encima del *DVS-EDF* y *PLMDP*. Esto sucede debido a que la demanda del procesador que se necesita para ejecutar las tareas asignadas en el hiperperiodo, con la restricción impuesta del 50% de sus  $C_{i,k}$ , ocasiona que existan más cálculos del *ST* y a su vez buscan cumplir con la referencia indicada. En cambio el *PLMDP*, aunque su característica de planificación es estática, trabaja en la aplicación en forma dinámica, lo que aprovecha para realizar cálculos a favor.

## 4. Conclusiones

En este trabajo se ha presentado una estrategia para aprovechar el tiempo ocioso dinámico dirigido al ahorro de energía usando lineamientos de la planificación realimentada. Materializándose en lazos de control multifrecuencia para el factor de carga de un procesador, acompañado de un módulo de ajuste y prioridad.

Se adaptaron e integraron tres técnicas dinámicas como son: el uso del tiempo ocioso debido al *WCET* (*STWCET*), factor de carga del procesador (*STU*) y el uso del tiempo ocioso por estiramiento a la(s) próxima(s) activación(es) (*STEPA*). En una en una llamada Técnica Dinámica Multifrecuencia para el

Manejo del Tiempo Ocioso (*TDMFMTO*), la cual se alojó en un planificador realimentado para el ahorro de energía, basado en *DVFS*.

Las pruebas realizadas demuestran que el enfoque adoptado deja la flexibilidad de que tanto a nivel global o local el planificador actúa en caso de existir condiciones para aprovechar tiempos ociosos, siendo capaces de compensar la variabilidad y garantizar la operatividad del sistema. Además sugiere un buen desempeño al contrastarlo con otros propuestos en la literatura.

## Agradecimiento

Los autores agradecen al Postgrado en Instrumentación de la Facultad de Ciencias de la Universidad Central de Venezuela, por propiciar el proyecto CmEPG 195-2010, donde convergen varias disciplinas. Así como también, al Consejo de Investigación de la Universidad de Oriente – Venezuela, por cofinanciar el proyecto CI-020402-1739-11.

## 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*. USA: Springer Science+Business Media.

Bhatti M. K., Belleudy C., & Auguin M. (2010). *An inter-task real time DVFS scheme for Multiprocessor embedded systems*. (pp. 136-143). *Proc. Int. Conf. Design and Architectures for Signal and Image Processing*, Edinburgh, UK.: Electron Chips & Syst Design Init.

Chang, H-W., Chang, W-H., & Tsai, C-H. (2009). *Integrated single-inductor buck-boost or boost-boost DC-DC converter with power-distributive control*. (pp. 1184 – 1187). *Proc. 8vo Int. Conf. Power Electro. Drive Syst*. Taipei, Taiwan: IEEE Ind Electron Soc.

Chantem, T.T., Hu, X.S., & Lemmon, M.D. (2009). Generalized Elastic Scheduling for Real-Time Tasks. *IEEE Trans. on Comput.*, 58(4): 480-495.

Choi, Y., Chang, N., & Kim, T. (2007). DC-DC converter-aware power management for low-power embedded systems. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 26(8): 1367-1381.

Cimino, M., & Prabhakar, R. (2010). Design of linear time-invariant controllers for multirate systems. *Automatica*, 46(8):1315-1319.

Hu, X., & Quan, G. (2007). Fundamentals of power-aware scheduling. In J. Henkel y S. Parameswaran (Eds), *Designing Embedded Processors: A Low Power Perspective* (pp. 219-229). Dordrecht, Netherlands: Springer.

Kim W., Shin D., Yun H-S, Kim J., & Min S L. (2003). Performance evaluation of dynamic voltage scaling algorithms for hard real-time systems. *J. Low Power Electronics*, 1(3): 1-11.

Liu, C.L., & Layland, J.W. (1973). Scheduling Algorithms for Multiprogramming in a Hard Real-Time. *J. ACM*, 20(1): 40-61.

Moncusí, M. (2005). Ahorro energético en la planificación de sistemas en tiempo real. [Tesis Doctoral en línea]. Universidad Politécnica de Cataluña. España. Consultado el 19 de junio de 2013 en: <http://www.tesisenxarxa.net/TDX-0601106-122458/index.html>

Niu, L. (2011). Energy efficient scheduling for real-time embedded systems with QoS guarantee. *Real-Time Syst.*, 47(2): 75-108.

Piguet, C. (2006). *Low-power CMOS Circuits: Technology, Logic Design and CAD Tools*. Boca de Ratón, USA: CRC Press Taylor & Francys Groups.

Rakhmatov, D. (2008). Energy budget approximations for battery-powered systems with a fixed schedule of active intervals. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 16(8): 985 – 998.

Rizvandi, N.B., Taheri, J., & Zomaya, A.Y. (2011). Some observations on optimal frequency selection in DVFS-based energy consumption minimization. *J. Parallel Distrib. Comput.*, 71(8): 1154-1164

Salt, J., Casanova, V., Cuenca, A., & Pizá, R. (2008). Sistemas de control basados en red. Modelado y diseño de estructuras de control. *Rev. Iberoam. Autom. In.*, 5(3): 5-20.

Salt, J., & Albertos, P. (2005). Model-based multirate controllers design. *IEEE Trans. on Control System Technology*, 6(3): 988-997.

Scordino, C., & Lipari, G. (2007). A resource reservation algorithm for power aware scheduling of periodic and aperiodic real-time tasks. *IEEE Trans. Comput.*, 55(12):1509-1522.

Sha, L., Abdelzaher, T., Arzén, K.-E., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., & Mok, A. (2006). Real time scheduling theory: a historical perspective, *Real-Time Syst.*, 28 (2-3): 101-155.

Shin, Y., Choi, K., & Sakurai, T. (2001). Power-conscious scheduling for real-time embedded systems desing. *J. VLSI Desing.* 12(2): 139-150.

Shin, D., Kim, W., Jeon, J., Kim J., & Min S L. (2003). SimDVS: An integrated simulation environment for performance evaluation of dynamic voltage scaling algorithms. In Falsafi and T.N. Vijaykumar B. (Eds.), *PACS 2002* (pp. 141-156). Berlin Heidelberg: Springer-Verlag.

Urriza, J., & Orozco, D. (2005). Fast Slack Stealing methods for Embedded Real Time System. *Proc. IEEE 26th Int. Real-Time Syst. Symp.* (pp. 12-16), Miami, USA: IEEE Com Soc Tech. Committee on Real-Time Syst.

Venkatachalam V., & Franz M. (2005). Power Reduction Techniques for Microprocessor Systems. *ACM Comput. Survey*, 37(3): 195-237.

Xia, F., & Sun, Y. (2008). *Control and scheduling codesing-flexible resource management in real-time control systems.* Heidelberg, Germany: Springer.

Xia, F., Ma, L., Zhao, W., Sun, Y., & Dong, J. (2009). Enhanced energy-aware feedback scheduling of embedded control systems. *J. Comput.*, 4(2): 103-111.

Xian, C., Lu Y-H, & Li, Z. (2008). Dynamic voltage scaling for multitasking real-time system with uncertain execution time. *IEEE Trans. Computer-Aided Design of Integrate Circuits and Systems*, 27(8): 1467-1478.

Zhu, Y., & Muller, F. (2006). Exploiting synchronous and asynchronous DVS for Feedback EDF Scheduling on an Embedded Platform. *ACM T. Embed. Comput. S.*, 7(1): 1-26.

## Notas biográficas:



**Alfonso Salvador Alfonsi Sebastiani** Recibió el grado de Ingeniero Electricista de la Universidad de Oriente (UDO) - Barcelona - Venezuela, en 1994, Magister Scientiarum Mención Instrumentación de la Universidad Central de Venezuela (UCV) - Caracas - Venezuela, en 2005. Profesor Titular adscrito al Grupo de Investigación Arquitecturas de Sistemas de Control (GASC) del Departamento de Computación y Sistemas de la Escuela de Ingeniería y Ciencias Aplicadas, UDO, Núcleo Anzoátegui, Barcelona, Venezuela. Profesor del Postgrado en Automatización e Informática Industrial de la UDO. Su interés investigador se centra en las arquitecturas de sistemas de control empotrados de tiempo real con ahorro de energía y tecnologías para la sustentabilidad. Es autor varias publicaciones en revistas y congresos. Acreditado al Programa Estímulo a la Investigación e Innovación (PEII) de Venezuela.



**Jesús Pérez** Recibió el grado de Ingeniero Electricista de la Universidad de Carabobo (UC) – Valencia – Venezuela, en 1988, el grado de Magister Scientiarum y Doctor en Ciencias Mención Instrumentación de la Universidad Central de Venezuela (UCV) - Caracas - Venezuela, en el 2002 y 2005, respectivamente. Profesor Titular de la Universidad Politécnica Territorial del Estado Aragua "Federico Brito Figueroa", La Victoria, Venezuela. Profesor del Postgrado en Instrumentación de la Facultad de Ciencias en la UCV. Su interés investigador se centra en aplicaciones de sistemas de adquisición de datos y robótica. Es autor varias publicaciones en revistas y congresos. Acreditado al Programa Estímulo a la Investigación e Innovación (PEII) de Venezuela.



**Emery Dunia** Recibió el grado de Físico de la Universidad Central de Venezuela (UCV) – Caracas - en 1969 y Docteur es Sciences de la Université Pierre et Marie Curie, Paris VI – France - en 1980.

Profesor Titular adscrito a la Escuela de Física de la Facultad de Ciencias, UCV, Caracas, Venezuela. Coordinador del Postgrado en Instrumentación de la Facultad de Ciencias. Su interés investigador se centra en la Enseñanza de las Ciencias. Es autor varias publicaciones en revistas y congresos. Acreditado al Programa Estímulo a la Investigación e Innovación (PEII) de Venezuela.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.