XXV Jornadas de Automática
Ciudad Real, del 8 al 10 de septiembre de 2004

ASIGNACION DINAMICA DE RECURSOS A TAREAS DE CONTROL EN SISTEMAS DE TIEMPO REAL

Pau Martí, Manel Velasco, José Yépez, Jordi Ayza, Ricard Villà y Josep M. Fuertes
Dept. Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya
Pau Gargallo, 5, 08028 Barcelona, Spain
{pau.marti,manel.velasco,jose.yepez,jordi.ayza,ricard.villa josep.m.fuertes@upc.es}

Resumen

En este trabajo mostramos que, en los sistemas de tiempo real, el rendimiento de control proporcionado por un conjunto de tareas de control que se ejecutan de forma concurrente mejora significativamente si los recursos de cómputo se asignan dinámicamente según el estado de los sistemas controlados. Además presentamos la política de asignación de recursos óptima en el sentido que maximiza el rendimiento de control proporcionado por el conjuto de tareas teniendo en cuenta la restricción en la limitación de los recursos de cómputo.

Palabras clave: Sistemas de tiempo real, sistemas controlados por computador, asignación de recursos, optimización.

1 INTRODUCCION

En los sistemas de control de tiempo real, la asignación de recursos a tareas de control determina el rendimiento de los lazos de control. Las técnicas tradicionales de tiempo real de asignación de recursos a tareas de control están basadas en estrategias que asignan estáticamente una porción de procesador a cada tarea de control, independientemente de la dinámica que cada lazo de control tenga en tiempo de ejecución. Pero si nos fijamos en el comportamiento de los lazos de control y en la relación existente entre rendimiento de control y la frecuencia de ejecución del controlador, estas técnicas parecen ser no óptimas. Un contralador puede no necesitar toda la frecuencia de ejecución asignada si el sistema controlado está en equilibrio. Cuando el sistema controlado sufre una perturbación y su estado es desplazado lejos del punto de equilibrio, un incremento en la frecuencia de ejecución de la tarea de control puede acelerar su recuperación [7].

Teniendo en cuenta esta observación, en este trabajo mostramos que asignando dinámicamente recursos a tareas de control según el *estado* de los sistemas controlados, el rendimiento de control proporcionado por el conjunto de tareas de control se puede mejorar significativamente. Concretamente, presentamos una política de asignación de recursos óptima para tareas de control (en el sentido de ser la solución a un problema de optimización lineal con restricciones), que llamaremos *óptima*.

La correcta implementación de la política óptima, así como la implementación de otra política de asignación de recursos en lazo cerrado, llamada *proporcional* y desarollada para el estudio comparativo, pasa por considerar dos apectos. Primero, estas políticas requieren la implementación de tareas de control capaces de ejecutarse con distintas frecuencias, proporcionando mejor rendimiento de control cuanto mas proporción de procesador se les asigna [8]. Segundo, la implementación requiere el soporte de un sistema de tiempo real flexible, capaz de cambiar dinámicamente los recursos asignados (via modificación de los periodos de las tareas, por ejemplo), garantizando así mismo las restricciones temporales de las tareas, como ofrece el sistema de tiempo real RBED (Rate-Based Earliest Deadline) [2].

Presentamos resultados experimentales que muestran que dada una misma secuencia de perturbaciones generadas de forma aleatoria sobre un conjunto de lazos de control, la política óptima mejora significativamente el rendimiento de control (proporcionado por el conjunto de tareas de control) con respecto a la implementaion traditonal (política que llamaremos *estática*), y maximiza el rendimiento de control comparado con la política proporcional.

El resto de este trabajo se estructura como sigue. Sección 2 repasa el estado del arte. La assignación de recursos como problema de optimización se formula en la Sección 3 y se soluciona en la Sección 4. Sección 5 presenta los resultados experimentales y Sección 6 concluye el trabajo.

2 ESTADO DEL ARTE

Las técnicas que presentamos son similares a las técnicas de asignacin de recursos presentadas en [11] y [6]. El objetivo del trabajo [11] es en el diseño de controladores para satisfacer criterios de

estabilidad exponencial y asintótica con periodos constantes. Nuestro objetivo no es el diseño de controladores, sino el estudio de como la adaptabilidad de los controladores a distintos periodos puede ser usado para mejorar el rendimiento de control en un sistema de control de tiempo real multitarea. El trabajo presentada en [6] se centra en controladores predictivos con modelo, donde la adaptación se consigue variando los tiempos de ejecución. Nuestra arquitectura apunta a una clase más amplia de lazos de control con controladores diseñados con técnicas clásicas de control para sistemas lineales, y la adaptación se basa en la variación de la frecuencia de ejecución de los controladores.

Cabe destacar que la política de asignación de recursos óptima que presentamos soluciona el problema de planificación con calidad de control (QoC, Quality-of-Control) formulado en [7], pero desde un punto de vista de asignación de recursos.

3 PROBLEMA

El sistema de control de tiempo real que consideramos está formado por un conjunto de n tareas de control $\tau_1, \tau_2, \ldots, \tau_n$, cada una de ellas encargada de controlar un sistema físico[1].

Cada sistema controlado por una tarea τ_i puede describirse con las ecuaciones en espacio de estados (1) y (2) que describen su dinámica lineal e invariante en el tiempo.

$$\dot{\vec{x}}_i(t) = A_i\vec{x}_i(t) + B_i\vec{u}_i(t) \quad t \in R^+ \qquad (1)$$

$$\vec{y}_i(t) = C_i\vec{x}_i(t) + D_i\vec{u}_i(t) \qquad (2)$$

Si consideramos 0 el punto de equilibrio de cada sistema, entonces el error e_i (3) mide lo lejos que cada sistema está de su punto de equilibrio, para cualquier $t > 0$. Este error es la información que cada tarea envia al gestor de recursos para que este re-asigne recursos en tiempo de ejecución.

$$e_i = |\vec{x}_i(t)| \qquad (3)$$

Además, para cada tarea de control especificamos un índice de rendimiento de control p_i que relaciona, para cada lazo de control, el rendimiento según la proporción de procesador (*factor parcial de utilización*, r_i en (4)) asignada a la tarea de control. Como se indica en [3], esta relación, usando índices estándares lineales o cuadráticos, se puede aproximar con una relación lineal (4), donde c_i y h_i son, respecivamente, el peor tiempo de ejecución de la tarea i y su periodo.

$$p_i(r_i) = \alpha_i \frac{c_i}{h_i} = \alpha_i r_i \qquad (4)$$

[1]De aquí en adelante, el subíndice "i" identifica un lazo de control, identificado la tarea de control, y/o el sistema físico controlado o proceso.

En (4), teniendo en cuenta que el peor tiempo de ejecución de la tarea es constante para cada tarea, cualquier variación en la asignación de recursos a una tarea, r_i, implica variar su periodo h_i.

Las tareas de control en el sistema de tiempo real implementan leyes de control adaptables a distintos valores del periodo de muestreo (que coiniciden con los periodos de las tareas), siguiendo las técnicas presentadas en [10] y [1]. Para la clase de sistemas que se pueden describir con las ecuaciones (1) y (2), las señales de control $\vec{u}_i(t)$ vienen dadas por leyes de control discretas L_i que se diseñan con metodos estándares. Para cada ley de control, que depende paramétricamente del period de muestreo $L_i(h_i)$, se le especifica un rango de periodos de muestreo $h_i \in [h_i^{max} \ldots h_i^{min}]$ de tal manera que para todos ellos las especificaciones de rendimiento se cumplen.

Cada ley se implementa en una tarea de control de tiempo real, que secuencialmente muestrea, ejecuta la ley de control y envia la actuación. Las tareas de control se especifican con un conjunto de periodos (en lugar de especificar un solo valor por el periodo) que corresponde al conjunto de periodos de muestreo utilizados en la fase de diseño del controlador. En tiempo de ejecución, cada tarea de control se ejecuta con un valor de periodo perteneciente al rango especificado, adpatando las ganancias, $L(h_i)$, apropiadamente (para mas detalles, ver [8]). La estabilidad de cada lazo de control en el sistema de tiempo real se analiza mediante las técnicas descritas en [5]. A nivel de sistema operativo, cada tarea de control τ_i se caracteriza por el factor parcial de utilización r_i, su índice de rendimiento p_i, i el error del sistema controlado e_i, tal y como se representa en (5).

$$\tau_i = \{r_i, p_i, e_i\} \qquad (5)$$

3.1 ENUNCIADO

Dado un sistema de tiempo real multitarea, para un conjunto de n tareas de control especificadas como en (5), el problema es determinar (en tiempo de ejecución) los factores parciales de utilización r_i, $i = 1, \ldots, n$ que se tienen que asignar a cada tarea de control, de tal manera que todas las tareas sean planificables y el rendimiento de control proporcionado por el conjunto de tares sea maximizado en cada instante de tiempo.

4 ASIGNACION OPTIMA

El problema de asignación de recursos se puede formular como un problema genérico de optimización con restricciones, como se muestra a con-

tinuación,

$$\begin{aligned} maximize \quad & g(p_i(r_i), e_i) && (6) \\ subjectto \quad & \textstyle\sum_{i=1}^{n} r_i \leq U_d && (7) \end{aligned}$$

donde la solución es un vector $\vec{r} = [r_1, r_2, \ldots, r_n]$ que maximiza el rendimiento de control global proporcionado por el conjunto de tareas de control, rendimiento representado por la función (vector) objetivo g en (6), tenindo en cuenta las restricciones de planificabilidad dadas por el factor de utilizacion deseado por el conjunto de tareas, U_d en (7).

El máximo absoluto $\vec{r}$ se encontrará en el interior, en la frontera, o en los puntos extremos del conjunto de puntos definido por (7). Un algoritmo genérico para encontrar la solución se puede resumir en cuatro pasos [4]:

Paso 1: Buscar máximos relativos en el interior del conjunto de puntos definido por (7), solucionando el sistema de ecuaciones especificado por (8),

$$\frac{\partial g}{\partial r_1} = 0,\ \frac{\partial g}{\partial r_2} = 0,\ \ldots,\ \frac{\partial g}{\partial r_n} = 0 \qquad (8)$$

y guardar aquellos $\vec{r}$ que maximizan g.

Paso 2: Buscar máximos relativos en la frontera del conjunto de puntos definido por (7), solucionando el sistema de ecuaciones especificado por (9),

$$\begin{aligned} &\frac{\partial g([U_d - r_2 - r_3 - \ldots - r_n, r_2, r_3, \ldots, r_n])}{\partial r_i} = 0, i \neq 1 \\ &\frac{\partial g([r_1, U_d - r_1 - r_3 - \ldots - r_n, r_3, \ldots, r_n])}{\partial r_i} = 0, i \neq 2 \qquad (9) \\ &\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ &\frac{\partial g([r_1, r_2, \ldots, U_d - r_1 - r_2 - \ldots - r_{n-1}])}{\partial r_i} = 0, i \neq n \end{aligned}$$

y guardar aquellos $\vec{r}$ que maximizan g.

Paso 3: Buscar los valores de g para los extremos del conjunto de puntos definido por (7), como se especifica en (10),

$$\begin{aligned} &g([U_d, 0, \ldots, 0]) \\ &g([0, U_d, \ldots, 0]) \qquad (10) \\ &\qquad\qquad \vdots \\ &g([0, 0, \ldots, U_d]) \end{aligned}$$

y guardar aquellos $\vec{r}$ que maximizan g.

Paso 4: Escoger el $\vec{r}$ entre los obtenidos en *Paso 1, 2* y *3* que maximiza g.

Según como sea la función objetivo g en (6), solucionar el problema de optimización en tiempo de ejecución puede no ser viable debido al coste computacional. Pero como explicamos a continuación, para la arquitectura que presentamos, el problema se puede simplificar de tal manera que 1) tiene solución y 2) es viable encontrarla en tiempo de ejecución.

4.1 SIMPLIFICACION

La simplificación del problema de optimización está basada en las siguientes obsevaciones:

- Cada tarea de control se considera independiente, en el sentido de estar encargada de controlar un sistema físico (sección 3). Entonces la función objetivo $g(\cdot)$ en (6) que enlaza los rendimientos p_i de cada tarea de control se puede considerar como la suma (posiblemente ponderada) de los rendimientos proporcionados por cada tarea de control (tal y como también se definió en los procedimientos de optimización presentados en [9] o [3]). Cada índice de rendimiento puede ser ponderado con un peso w_i, proporcionando así un mecanismo que permita comparar los distintos lazos de control del sistema.
- La solución $\vec{r}$ tiene que tener en cuenta que para cualquier lazo de control, a mayor error e_i y dada una asignación r_i, mayor tiene que ser el rendimiento proporcionado por la tarea de control. Esta lógica se consigue escalando el rendimiento p_i con el error e_i del sistema controlado, $e_i p_i$.

En resumen, el problema de optimización se puede re-escribir como:

$$\begin{aligned} maximize \quad & \textstyle\sum_{i=1}^{n} w_i e_i p_i(r_i) && (11) \\ subjectto \quad & \textstyle\sum_{i=1}^{n} r_i \leq U_d && (12) \end{aligned}$$

La complejidad de la solución del problema planteado en (11) y (12) depende de cada función p_i por el hecho que las ecuaciones (8) y (9) se han simplificado respectivamente al conjunto de ecuaciones especificados en (13)

$$\frac{\partial b_1}{\partial r_1} = 0, \frac{\partial b_2}{\partial r_2} = 0, \ldots, \frac{\partial b_n}{\partial r_n} = 0 \qquad (13)$$

y (14), donde $b_i = w_i e_i p_i(r_i)$.

$$\begin{aligned} &\frac{\partial b_1(U_d - r_2 - r_3 - \ldots - r_n)}{\partial r_i} = 0,\ i \neq 1 \\ &\frac{\partial b_2(U_d - r_1 - r_3 - \ldots - r_n)}{\partial r_i} = 0,\ i \neq 2 \qquad (14) \\ &\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ &\frac{\partial b_n(U_d - r_1 - r_2 - \ldots - r_{n-1})}{\partial r_i} = 0,\ i \neq n \end{aligned}$$

Si los índices de rendimiento p_i son lineales (como se definió en (4)), el problema de optimización es

lineal y la solución $\vec{r} = [r_1, r_2, \ldots, r_n]$ se encuentra haciendo una simple búsqueda (correspondiente al *Paso 3*) ya que las ecuaciones (13) y (14) correspondientes a los *Pasos 1* y *2* quedan indeteminadas. Por ejemplo, si p_i son lineales ($p_i = \alpha_i r_i$) en (13), obtenemos el siguiente conjunto de ecuaciones (15)

$$\begin{aligned} \frac{\partial b_1}{\partial r_1} &= \frac{\partial w_1 e_1 \alpha_1 r_1}{\partial r_1} = w_1 e_1 \alpha_1 = 0 \\ \frac{\partial b_2}{\partial r_2} &= \frac{\partial w_2 e_2 \alpha_2 r_2}{\partial r_2} = w_2 e_2 \alpha_2 = 0 \qquad (15) \\ &\vdots \\ \frac{\partial b_n}{\partial r_n} &= \frac{\partial w_n e_n \alpha_n r_n}{\partial r_n} = w_n e_n \alpha_n = 0 \end{aligned}$$

que quedan indeterminadas. Y lo mismo ocurre con el conjunto de ecuaciones especificado en (14).

En consecuencia, la solución se reduce en realizar la búsqueda del *Paso 3*, ajustada al problema planteado en (11) y (12). Así, evaluando

$$\begin{aligned} g([U_d, 0, \ldots, 0]) &= b_1(U_d) + b_2(0) + \ldots + b_n(0) \\ &= w_1 e_1 \alpha_1 U_d \\ g([0, U_d, \ldots, 0]) &= b_1(o) + b_2(U_d) + \ldots + b_n(0) \\ &= w_2 e_2 \alpha_2 U_d \qquad (16) \\ &\vdots \\ g([0, 0, \ldots, U_d]) &= b_1(0) + b_2(0) + \ldots + b_n(U_d) \\ &= w_n e_n \alpha_n U_d \end{aligned}$$

encontraremos la asignación óptima de recursos. Cabe destacar que (16) equivale a buscar el máximo $w_i e_i \alpha_i$, $\forall i = 1 \ldots n$.

En consecuencia, la solución óptima $\vec{r} = [r_1, r_2, \ldots, r_n]$ del problema de optimización especificado en (11) y (12) es $\vec{r} = [0, 0, \ldots, 0, r_i = U_d, 0, \ldots, 0]$, $i \in [1, \ldots, n]$ tal que $w_i e_i \alpha_i$ es máximo $\forall i \in [1, \ldots, n]$, si el conjunto de tareas de control se especifican como en (5).

En términos de asignación de recursos, la solución óptima determina que todo el procesador disponible (U_d) debe de asignarse a la tarea de control que tenga mayor $w_i e_i p_i$. Si todos los índices p_i son iguales y todos los w_i son iguales, todo el procesador disponible debe de asignarse a la tarea de control cuyo sistema controlado tenga el mayor error e_i. En la prácitca, a cada tarea de control τ_i, se le tiene que asignar un factor de utilización mínimo dado por su h_i^{max} para garatizarle una frecuencia mínima de ejecución.

5 RESULTADOS

Hemos implementado y evaluado la arquitectura en lazo cerrado para tareas de control de tiempo real descrita en la sección 3 en el sistema de tiempo real integrado RBED.

Aparte de implementar en RBED la política óptima (obtenida en la Sección 4), y para demostrar sus beneficios y evaluar su rendimiento, también hemos imlementado una política de asignación de recursos en lazo cerrado, llamada *proporcional*, que asigna recursos a las tareas en proporción al error. Además, para poder comparar directamente el rendimiento con aquel que se obtiene con implementacines tradicionales, también hemos implementado una política de base, llamada *estática*, en la cual las tareas de control siempre comparten los recursos disponibles de forma fija (sin usar asignación dinámica de recursos). La política estática implementa leyes de control *tradicionales* (no adaptativas).

La implementación de las políticas dinámicas (óptima y proporcional) requiere 1) que las tareas de control puedan enviar al gestor de recursos del sistema operativo, en cada ejecución, el error de su sistema controlado y 2) que conozcan el periodo que se les va aplicar para poder agustar las ganancias de forma adecuada. Esto se consigue con una llamada al sistema con el error como parámetro de entrada, y el periodo (calculado por la política dinámica) como parámetro de salida.

La políticas de asignación de recuros para tareas de control se han integrado en RBED y se han implementado, por razones de senzillez, en Linux (kernel 2.4.20). Todos los experimentos se realizaron en un PC estándar (Pentium III, 1Ghz, 512MB RAM y disco duro de 40GB)

En los experimentos, ejecutamos un conjunto de tres tareas de control, cada una de ellas controlando un *péndulo invertido* simulado. Cada tarea de control implementa la misma ley de control paramétrica, obtenida usando técnicas estándares de asignación de polos. Hemos definido todas las tareas iguales ya que facilita el anánlisis de rendimiento. Cabe destacar que en este caso, w_i y p_i son iguales para todas las tareas, lo que significa que el error e_i sera el factor principal a tener en cuenta en cada política.

El factor de utilización deseado para las tres tareas de control es $U_d = 97\%$, dejando un 3% de CPU a tareas de propósito general. El peor tiempo de ejecución de cada tarea es $0.0135s$. Para las políticas dinámicas (óptima y proporcional), cada tarea puede ejecutarse con un periodo dentro del rango $[0.03s \ldots 0.05s]$. Para la política estática, cada tarea tiene un periodo fijo proporcioanl al tercio de U_d, esto es, un period de 0.042s.

Para cada política, ejecutamos las tres tareas durante diez minutos y generamos perturbaciones

aleatorias (para cada *péndulo invertido*). Figura 1 (parte superior) muestra durante un periodo de 60s las distintas desviaciones sobre el ángulo que afectan los tres péndulos controlados por las tres tareas de control, debidas a las perturbaciones.

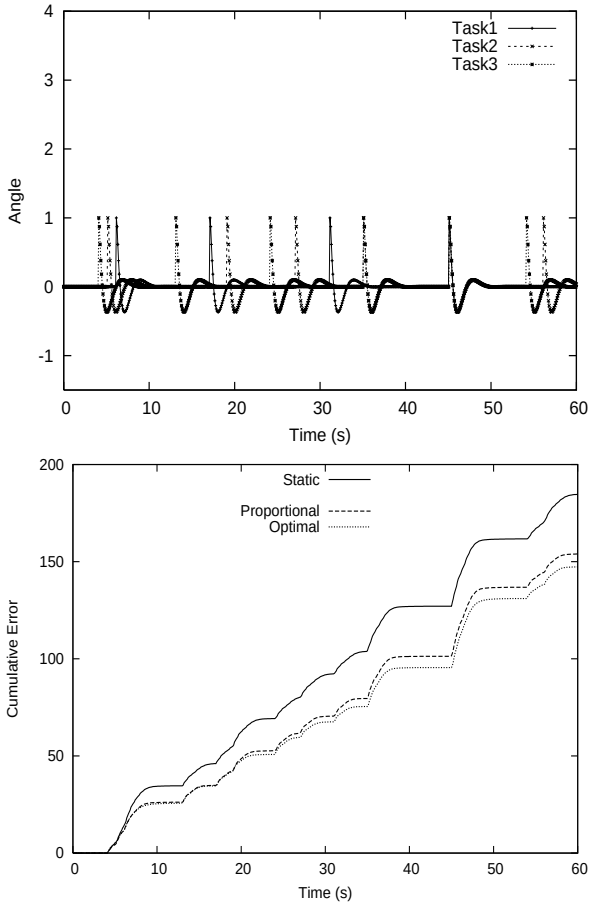

Figura 1: Parte superior: secuencia de errores (ángulos) de los tres pndulos (debidos a las perturbaciones). Parte inferior: error acumulado de los tres péndulos para las tres políticas de asignación de CPU (estática, proporcional y óptima).

Figura 1 (parte inferior) detalla el análisis de rendimiento de las políticas óptima, proporcional y estática durante un periodo de 60s. En el eje vertical se mide el error acumulado total de los tres *péndulos invertidos*, que se ha calculado como sigue

$$\int_0^{t_e} \sum_{i=1}^{3} |\vec{x}_i(t)| dt \qquad (17)$$

donde t_e es el tiempo que duró cada experimento (diez minutos) y $\vec{x}_i(t)$ es el vector d'estado de cada péndulo.

De la Figura 1 (parte inferior) se concluye nque

- la política óptima es la que proporciona mejor rendimiento comparado con la proporcional y la estática , ya que es la que presenta un menor error acumulado
- las políticas dinámicas (óptima y proporcional) dan un mejor rendimiento que la estática

Además, comparando la Figura 1 parte superior e inferior, se puede observar la relación que existe entre cada perturbación y el correspondiente incremento en el error acumulado de los tres péndulos.

Figura 2 completa y corrobora las conclusiones extraidas de la Figura 1 (parte inferior) pero para un intervalo de diez minutos. Como se puede observar, la diferencia de rendimiento entre la política estática y las dinámicas incrementa significativamente, echo que también justifica que asignar recursos dinámicamente en función de los estados (dinámica de los sistemas controlados es una buena estrategia.

También se puede observar en la Figura 2 que la diferencia entre la política óptima y la proporcional no es tan significativa. Una posible explicación tendria su origen en la secuencia aleatoria de perturbaciones. Si el intervalo entre perturbaciones es corto, la probabilidad de que todas las tareas tengan error es alta, y por tanto, la política óptima y la proporcional asignaran recursos de forma muy diferente. Pero si el intervalo entre perturbacions incrementa, la probabilidad de que todas las tareas tengan error disminuye, con lo cual, la política proporcional asignaria recursos como óptima, proporcionando entonces un rendimiento de control muy parecido (futuras lineas de trabajo se centraran en investigar este punto).

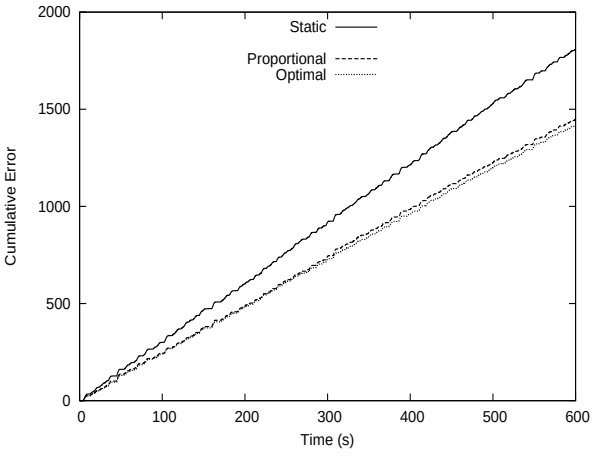

Figura 2: Error acumulado de los tres péndulos para las tres políticas de asignación de CPU (estática, proporcional y óptima) para un intervalo de 600s.

Cabe destacar que las políticas dinámicas, en contraposición a la tradicional, ofrecen un ahorro de CPU. En la política estática, el 97% de la CPU (o cualquiera que sea el factor de utilización deseado U_d para el conjuto de tareas de control) siempre está ocupado por las tareas de control. En las políticas dinámicas, cuando no hay presencia de error en los sistemas controlados, las tareas de control se ejecutan con su periodo máximo, proporcionando un ahorro de CPU que puede ser usado por otras tareas.

6 CONCLUSIONES

En este trabajo hemos presentado políticas de asignación de recursos para tareas de control de tiempo real basadas en la dinámica de los sistemas controlados. Hemos planeado el problema de asignación de recursos como un problema de optimización lineal con restricciones, cuya solución ofrece la política óptima en terminos de rendimiento de control.

Los resultados experimentales confirman los resultados teóricos, y demuestran que la asignación dinámica de recursos ofrece un rendimiento superior a la política estática de asignación de recursos que tradicionalmente se aplica a las tareas de control en un sistema de tiempo real.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por la CICYT del Ministerio de Ciencia y Tecnología Espaol DPI2002-01621. Los autores también desean agradecer el apoyo recibido por el Dr. Scott A. Brandt y Mr. Caixue Lin en la implementación de las políticas de asignación de recursos sobre RBED.

Referencias

[1] P. Albertos and J. Salt. Digital regulators redesign with irregular sampling. In *11th IFAC World Congress (Preprints)*, volume 8, pages 157–161, 1990.

[2] S. A. Brandt, S. Banachowski, C. Lin, and T. Bisson. Dynamic integrated scheduling of hard real-time, soft real-time and non-real-time processes. In *Proceedings of the 24th IEEE Real-Time Systems Symposium*, pages 396–407, Dec. 2003.

[3] A. Cervin, J. Eker, B. Bernhardsson, and K.-E. Årzén. Feedback-feedforward scheduling of control tasks. *Real-Time Systems*, 23:25–53, July 2002.

[4] E. Chong and S. Zak. *An Introduction to Optimization.* John Villey and Sons, Inc, 1996.

[5] M. Dogruel and U. Özgüner. Stability of a set of matrices: A control theoretic approach. In *Proceedings of the 34th IEEE Conference on Decision and Control*, Sept. 1995.

[6] D. Henriksson, A. Cervin, J. Åkesson, and K.-E. Årzén. On dynamic real-time scheduling of model predictive controllers. In *Proceedings of the 41th IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[7] P. Marti, G. Fohler, K. Ramamritham, and J. M. Fuertes. Improving quality-of-control uisng flexible time constraints: Metric and scheduling issues. In *Proceedings of the 23nd IEEE Real-Time Systems Symposium*, Dec. 2002.

[8] P. Marti, J. M. Fuertes, G. Fohler, and K. Ramamritham. Jitter compensation for real-time control systems. In *Proceedings of the 22nd IEEE Real-Time Systems Symposium*, Dec. 2001.

[9] D. Seto, J. Lehoczky, L. Sha, and K. Shin. On task schedulability in real-time control systems. In *Proceedings of the 17th IEEE Real-Time Systems Symposium*, Dec. 1996.

[10] B. Wittenmark and K. J. Åström. Simple self-tuning controllers. *Unbehauen, Ed. Methods and Applications in Adaptive Control, Lecture Notes in Control and Information Sciences*, 24:21–29, 1980.

[11] Q. Zhao and D.-Z. Zheng. Stable and real-time scheduling on a class of perturbed hybrid dynamic systems. In *IFAC World Congress*, pages 91–96, 1999.