

Diseño y análisis de tiempo real de un controlador para motor de corriente directa

Reinier Torres Labrada

Correo electrónico: reinier@electrica.cujae.edu.cu

Mailén Sarmiento Fong

Correo electrónico: mailen@electrica.cujae.edu.cu

Instituto Superior Politécnico José Antonio Echeverría, Cujae, La Habana, Cuba

Resumen

Se presenta el análisis de tiempo real para un controlador de motores de corriente directa (CD). Durante el análisis funcional se obtuvieron los plazos y períodos de las tareas, cuyos valores son utilizados posteriormente como parámetros en el análisis de tiempo de respuesta (ATR). Para obtener el caso peor de tiempo de ejecución (CPTE) el sistema es estimulado con un vector aleatorio de entrada para las tareas analizadas, se mide el tiempo de computación para un número grande de iteraciones y se realiza el modelado con el uso de la distribución de probabilidad de Gumbel. Con los resultados obtenidos para el CPTE y los parámetros del conjunto de tareas que se obtuvieron durante el análisis funcional, se evalúa un modelo de ATR y sus resultados son discutidos. Al modelar el CPTE mediante la teoría de valores extremos (TVE) se evita el análisis del software a bajo nivel y la necesidad de un modelo preciso para el hardware. Este método es útil para el análisis de sistemas de tiempo real blandos y puede ser fácilmente aplicado a otras plataformas de desarrollo hardware/software.

Palabras clave: sistema de tiempo real, tiempo de respuesta, sistema operativo de tiempo real, procesamiento digital de señales, PID

Recibido: 20 de junio del 2011

Aprobado: 29 de agosto del 2011

INTRODUCCIÓN

Usualmente se tiende a asociar el concepto de tiempo real con sistemas que trabajan muy rápido. Es cierto que una parte importante de los sistemas en tiempo real (STR) operan a alta velocidad aunque no siempre ocurre así, [1] para un STR la velocidad no es el único aspecto determinante.

Se considera que un sistema es de tiempo real si responde a los estímulos, elabora y entrega la respuesta correcta antes del cumplimiento de los plazos temporales establecidos; una respuesta tardía carece de valor. [1- 3] El concepto que se debe aplicar a un STR no es operar muy rápido sino hacer el trabajo en tiempo.

Las investigaciones han extendido el concepto a sistemas en los que se acepta la pérdida de plazos y, como consecuencia, se reconocen al menos cuatro posibles subdivisiones según la tolerancia respecto a la inexactitud temporal:

1. *STR estricto*: La pérdida de un plazo implica el fallo total del sistema. [4]

2. *STR estricto-debilitado*: La tasa de fallos durante un intervalo de tiempo es conocida y está acotada. La pérdida de plazos dentro del intervalo es aceptable aunque debilita el desempeño del sistema.[5]

3. *STR firme*: Se acepta la pérdida de algunos plazos, pasado el umbral de plazos permisibles el sistema falla con consecuencias catastróficas.[1]

4. *STR blando*: La pérdida de plazos no implica el fallo total y catastrófico del sistema.[1]

Constituye una práctica relativamente frecuente en ingeniería realizar análisis funcional de la aplicación. Sin embargo, el análisis temporal normalmente se centra en la respuesta media del sistema y no tiene en cuenta los casos peores. [6] Por otro lado, varios autores refieren que cuando se realiza análisis de tiempo real, muchas veces se sustentan en la experiencia de los desarrolladores y no sobre resultados obtenidos en investigaciones académicas. [7] Las causas para la falta de aplicabilidad de resultados académicos en el análisis de tiempo real son varias, según

refiere [6] dos de las principales son: la falta de métodos suficientemente generales que puedan ser aplicados a diferentes plataformas de hardware/software y ejemplos que describan el uso de estas técnicas desde una perspectiva ingenieril.

En el presente trabajo se describe el diseño funcional y análisis de tiempo real de un controlador para motores de corriente directa (CD) como subsistema de un sistema digital (SD). Durante el análisis y diseño funcional, hasta las pruebas de comportamiento de TR, se aplican resultados académicos en el campo de sistemas de control aplicado a motores de CD, teoría de planificación en tiempo real, obtención del caso peor de tiempo de ejecución con el uso de la teoría de valores extremos, modelado de tiempo real y desarrollo de aplicaciones con sistemas operativos de tiempo real.

MATERIALES Y MÉTODOS

Las técnicas de control convencional han probado su eficacia a lo largo de la historia de la ingeniería de control. Un sistema bien descrito en la literatura es el control de motores de CD *shunt* o de excitación separada. Gracias a la característica lineal del comportamiento de su relación torque/velocidad, los motores de CD son procesos apropiados para aplicar control PID. [8], [9] Esta aplicación es factible de utilizar de conjunto con las técnicas de análisis de TR con intenciones demostrativas debido a que su nivel de complejidad y amplia aplicabilidad lo permiten.

El procesamiento digital de señales se aproxima a las cuatro décadas de desarrollo. Se pueden encontrar aplicaciones en el campo del entretenimiento, las comunicaciones, la exploración espacial y la medicina. [10], [11] Todo el desarrollo obtenido en este campo está ligado al desarrollo de la computación.

La aparición en la década de 1970 de los microprocesadores, su rápida evolución, así como la aplicación de teorías básicas como la transformada rápida de Fourier (FFT), permitieron el surgimiento del procesamiento digital de señales. [10], [11]

En los años 1980 el panorama para el procesamiento digital de señales cambia sustancialmente. Se logran implementar muchas de las teorías para el procesamiento digital; las cuales se habían establecido en la década anterior. Esto fue posible gracias al avance de la tecnología de fabricación de circuitos integrados y la electrónica digital, así como de las herramientas de desarrollo de software. [11] En esta década se desarrollan aplicaciones dedicadas al campo de las comunicaciones y la instrumentación industrial. [10], [11]

Los orígenes del análisis de TR se remontan a la década de 1940, [1] sin embargo, no es hasta 1973 con el trabajo de Liu y Layland [12] que esta disciplina despegue con fuerza. Uno de los resultados significativos en la teoría de sistemas de TR es el método de análisis del tiempo de respuesta (ATR) desarrollado en la década de 1980 [13], [14] cuyas ecuaciones fundamentales son (1) y (2).

$$R_i = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_j}{T_j} \right\rceil C_j \quad (1)$$

$$R_i^{k+1} = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_j^k}{T_j} \right\rceil C_j \quad (2)$$

donde:

$hp(i)$: Conjunto de tareas con prioridad mayor que τ_i .

R_j : Tiempo de respuesta de τ_j .

B_i Bloqueo que sufre τ_i .

C_i, C_j : Tiempo de computación de τ_i y τ_j .

La ecuación (1) permite obtener el tiempo de respuesta de una tarea desde el momento en que es liberada hasta completar el trabajo. Sin embargo, la presencia del parámetro R_j en ambos miembros impide evaluar la expresión. La ecuación (2), derivada de la anterior, permite obtener mediante recursión, el tiempo de respuesta R_j de cualquier tarea. La ecuación (2) tiene dos condiciones de parada. Si $R_i^{k+1} > D_i$ la tarea τ_i no puede ser planificada debido a que el tiempo de respuesta supera su plazo. Por otro lado si $R_i^{k+1} = R_i < D_i$ el tiempo de respuesta de τ_i ha sido encontrado con valor R_i^{k+1} . El resultado del ATR ha sido extendido para incorporar otros fenómenos como los bloqueos y las demoras [1], [3], [15].

El CPTe es de importancia fundamental en el análisis de TR debido a que se utiliza como el parámetro tiempo de computación (C) en cualquier prueba de planificación. Desde el trabajo de Liu y Layland [12], el cálculo del CPTe ha contribuido a forjar una rama de la teoría de sistemas de tiempo real, completamente dedicada al estudio y desarrollo de métodos para estimarlo.

La teoría de valores extremos (TVE) es uno de los métodos de reciente introducción en el análisis de tiempo real. Inicialmente aplicada en la predicción de eventos meteorológicos extremos. [16], [17] La TVE ha demostrado su aplicabilidad en otros campos del conocimiento científico y actualmente se utiliza en muchas ramas de la ciencia donde se necesite modelar el comportamiento estadístico de sistemas, que no operan sobre la media de la población, sino en alguno de sus extremos. Investigaciones recientes han aplicado la TVE a la estimación del caso peor de tiempo

de ejecución (CPTE) como alternativa a los métodos puramente analíticos. [18], [19]

Existen tres distribuciones fundamentales en la TVE: Frechet, Weibull y Gumbel [16]. La de Gumbel, como distribución limitante de la máxima de una población es la que se aplica en la estimación del CPTE [18], [19], [20]. Para el análisis se debe muestrear al sistema y obtener suficientes datos que permitan desarrollar su modelo de Gumbel. El modelo se construye procesando los datos con (3) y (4) y graficando las mediciones de tiempo de computación contra las probabilidades acumuladas empírica y de Gumbel. Una buena correspondencia entre ambas curvas es prueba de que el modelo refleja bastante bien el comportamiento temporal del proceso medido. El análisis del modelo permite decidir si este se puede utilizar para estimar el CPTE con una probabilidad aceptable para el sistema. [19], [20]

$$\bar{F}(z) = \begin{cases} 0, & z \leq x_0 \\ \frac{i}{n+1}, & x_i < z \leq x_{i+1} \\ 1, & z > x_{n+1} \end{cases} \quad (3)$$

$$G_{(0,\mu,\beta)}(x) = e^{-e^{-\frac{x-\mu}{\beta}}} \quad (4)$$

Siendo:

$$\beta = \frac{\sqrt{6}}{\pi} \hat{\sigma}, \quad \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad \mu = \bar{x} - \beta \lambda,$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \lambda = 0,5772$$

donde:

$\bar{F}(z)$: Función de probabilidad empírica.

$G_{(0,\mu,\beta)}$: Función de probabilidad de Gumbel.

β : Parámetro de escala para Gumbel.

μ : Parámetro de localización para Gumbel.

$\hat{\sigma}^2$: Desviación estándar de la muestra.

\bar{x} : Media de la muestra.

λ : Constante de Euler.

El modelo obtenido debe ser validado con otras mediciones y si el resultado demuestra que este es capaz de estimar el

CPTE con cierto nivel de pesimismo se puede considerar que es seguro. Los valores obtenidos de CPTE se pueden aplicar a la determinación del tiempo de respuesta según (2).

ANÁLISIS FUNCIONAL Y DISEÑO

Generalidades

El sistema digital (SD) está dividido en dos capas funcionales. La primera agrupa a todos los elementos de hardware (hw), y la segunda al firmware (fw). Esta última se encuentra dividida en dos niveles, el más cercano al hw incluye al SOTR y todas las subrutinas de atención a interrupción o manejadores de dispositivos. Las interfaces hacia el nivel superior son exportadas como servicios básicos del SD, haciendo accesible el hw a través de una interfaz única. Esta organización facilita la portabilidad de la aplicación hacia otro soporte hw o reutilizar parte del código de aplicación existente [3]. La figura 1 muestra algunos de los elementos que conforman el sistema digital.

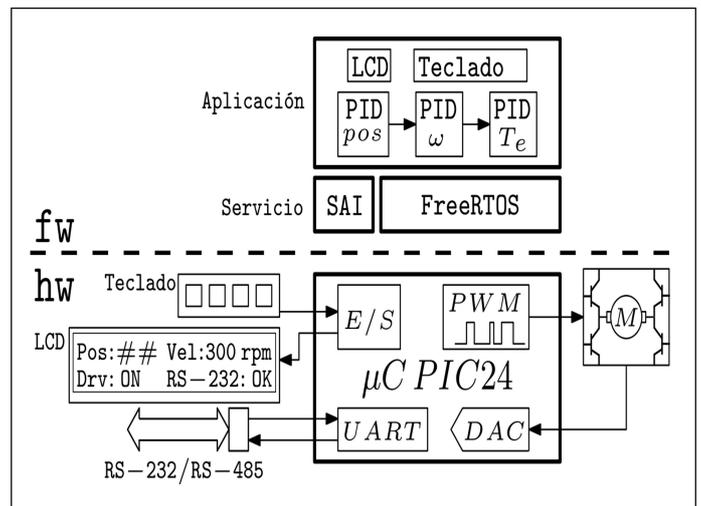


Fig. 1. Diagrama simplificado del sistema digital.

En este trabajo solo se aborda el análisis de TR para los componentes fw del subsistema de control del motor de CD. Este se encuentra formado por las implementaciones de los tres controladores con acción PID y el filtro digital IIR. El SOTR FreeRTOS también es analizado debido a su influencia en el comportamiento de TR del sistema.

Una estructura típica de control de motores de CD está formada por tres lazos en cascada, donde se controlan torque (T_e), velocidad (ω) y posición. [8], [9] La figura 2 muestra el esquema de control implementado. Cada celda PID incluye control Feedforward y para la realimentación se miden dos variables en el motor de CD, T_e y ω . [9], [21], [22] La velocidad se obtiene con un encoder incremental acoplado al eje del motor, y el torque con un sensor de corriente de *shunt* con salida analógica.

El *driver* es un *chopper* formado a partir de un puente H controlado con el módulo CCP1 en modo PWM y otros pines

de E/S del microcontrolador (μC). El período para el PWM se obtiene de (5) según [8].

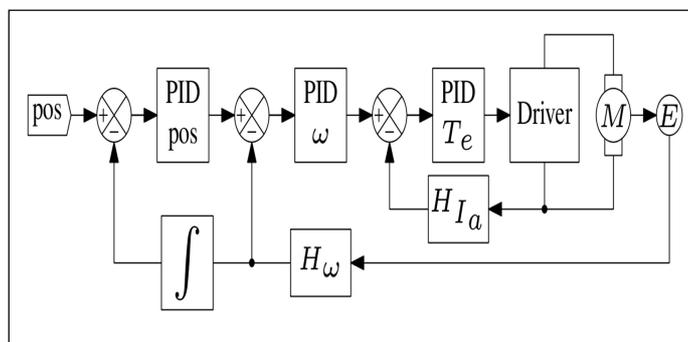


Fig. 2. Esquema del lazo de control en cascada.

$$f_{ch} = \frac{R_a}{-2L_a \ln\left(1 - \frac{F_p}{100}\right)} \quad (5)$$

donde:

R_a : Resistencia de armadura del motor.

L_a : Inductancia de armadura del motor.

F_p : Factor porcentual del ciclo útil.

La salida del encoder incremental se conecta a un circuito que convierte las fases y el índice en tres señales. La primera es el índice sin rebotes para la indicación de una revolución completa. La segunda es la codificación en cuadratura de las fases, se entrega como un tren de pulsos indicativo de la velocidad. Por último, las fases decodificadas son la referencia del sentido de giro. La velocidad es procesada con el módulo de captura CCP3, el índice está conectado a INT0, y la dirección a IC0 del módulo de interrupción por cambio de estado, del μC .

Según la expresión (6) dada por [8], en los motores de CD el torque (T_e) es función de la corriente de armadura (I_a).

$$T_e = K\Phi I_a \quad (6)$$

donde:

K : Constante del motor.

Φ : Flujo magnético en el estator.

Durante la operación del motor, los inductores de su armadura son conmutados a alta frecuencia por el circuito de potencia y a más baja frecuencia por las escobillas, ambos procesos provocan la aparición de transitorios de alta frecuencia en la señal medida de I_a [8], [22], [9]. Para tener una referencia del torque es preciso digitalizar y realizar el procesamiento digital de la señal muestreada, quien impondrá la máxima velocidad de respuesta sobre el sistema [23], [24].

Según [9] el período del lazo de control de torque es usualmente 10 ms y es el lazo con el período más corto en un sistema de control como el que aquí se presenta. El resto de las celdas de control deben tener entre cuatro y cinco veces el período de la celda externa más próxima en el circuito en cascada.

Filtro digital

El filtro digital (FD) se diseñó teniendo en cuenta los requerimientos de TR del controlador. Este hecho implica analizar la señal a procesar, y las relaciones que se establecen entre el FD y el controlador. [23], [24] Atendiendo a la relación que tiene el orden de un FD con los recursos de computación que demanda, las implementaciones IIR resultan menos costosas. [23] Sin embargo, en el filtro IIR se tuvo en cuenta, además, la fase y la estabilidad. [10], [11]

Durante el proceso de diseño del filtro se trabajó en las siguientes direcciones:

1. El orden del filtro (N): seleccionar el mínimo que brinde una respuesta de amplitud adecuada.

2. La frecuencia de muestreo (f_s) de la señal: seleccionarla tal que elimine el *aliasing* y evite el sobremuestreo.

3. Fijar el plazo (D_{IIR}) para el cálculo de la salida del filtro: de modo que las tareas dependientes la obtengan sin retardos apreciables.

La máxima componente de frecuencia (f_m) de I_a , se obtiene de la constante de tiempo del inductor de la armadura según la ecuación (7) dada por [8] y [22].

$$\tau_a = \frac{L_a}{R_a} \quad (7)$$

donde:

R_a : Resistencia de armadura del motor.

L_a : Inductancia de armadura del motor.

Para el caso

$\tau_a = 100 \text{ mH}/12,5\Omega = 8,10^{-3} \text{ s} \Rightarrow f_m = 125\text{Hz}$. Con f_m se fijó la frecuencia de muestreo según [10], [11] y [23] como $f_s = 4f_m \Rightarrow T_s = 2\text{ms}$. La frecuencia de esquina de la banda de paso (f_c) y la frecuencia de esquina de la banda de atenuación f_a , se obtienen con f_m y el ancho de la banda de transición, de modo que: $f_c = f_m$ y $f_a = f_m + \Delta f$ en este caso $\Delta f = 75\text{Hz} \Rightarrow f_m = 200\text{Hz}$ [23], [24]. La figura 3 (a) muestra las especificaciones del filtro analógico equivalente.

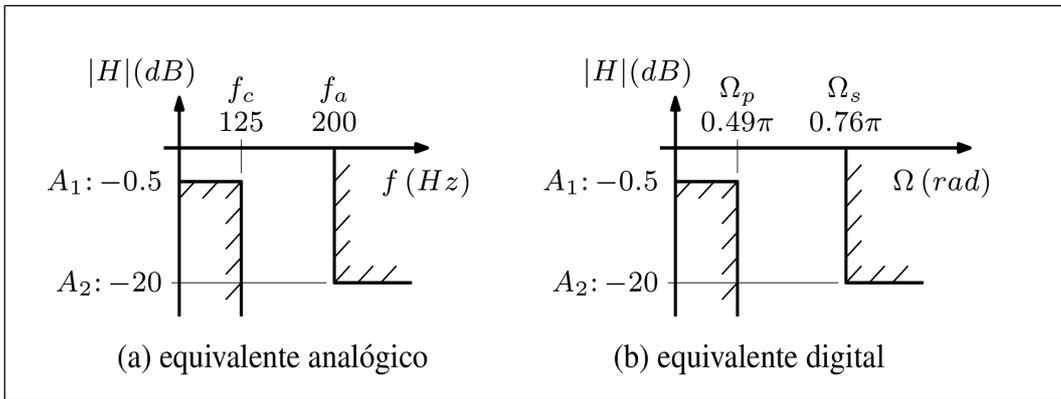


Fig. 3. Especificaciones del filtro.

Para la transformación de la frecuencia analógica a digital se aplica la expresión (8) dada por. [11]

$$\Omega = 2 \arctan\left(\frac{(2\pi fT)}{2}\right) \quad (8)$$

donde:

- Ω : Frecuencia digital.
- f : Frecuencia analógica.
- T : Periodo de la señal.

Para obtener Ω_p se utilizó f_c y para Ω_s se utilizó f_a . Las especificaciones del filtro digital resultante se muestran en la figura 3 (b). Los valores de A_1 y A_2 fueron seleccionados siguiendo criterios prácticos. Los valores de las especificaciones de la figura 3 (b) se introducen en GNU/Octave o MatLab y se aplica el procedimiento para sintetizar un filtro IIR con función de aproximación de Butterworth y Transformación Bilineal. Los valores de A_1 y A_2 son ajustados hasta obtener una respuesta apropiada del filtro y con el menor valor posible de N . La figura 4 muestra las respuestas de filtros con diferentes valores de N , para el caso se seleccionó $N = 4$. Con el aproximante de Butterworth se obtiene una respuesta libre de oscilaciones a ambos lados de la frecuencia de corte mientras que la transformación Bilineal previene los efectos del aliasing [10], [11].

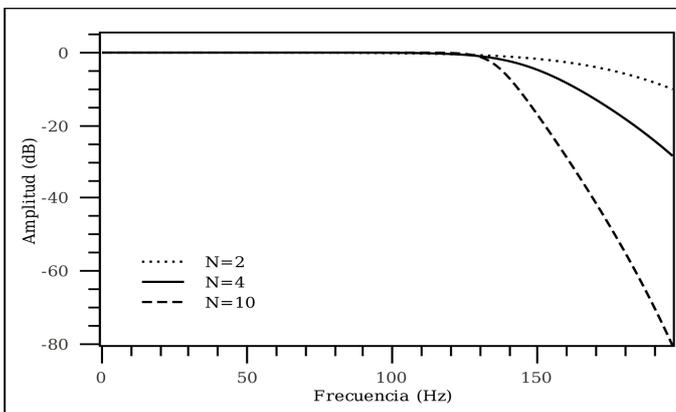


Fig. 4. Respuestas del filtro IIR para diferentes N.

El gasto computacional para este filtro es $2N + 1 = 9$ multiplicaciones y $2N = 8$ sumas en cada iteración. [10], [11] El tiempo requerido para el cálculo de una iteración debe ser estimado después de implementado en el fw. El plazo del FD puede establecerse como el periodo de I_a , por ello $D_{IIR} = T_s$; pues el filtro requiere que la salida precedente esté lista un instante antes de cada iteración [23], [24]. Los plazos para los procesos de control pueden establecerse de la misma forma que para el FD, es decir, tomar como plazos a los periodos ($D_i = T_i$). Al determinar el plazo para el filtro ($D_{IIR} = 2ms$), se pueden asignar a partir de aquí los periodos para el resto de las tareas del proceso de control según [9], los resultados se muestran en la tabla 1.

Tarea	Relación	T_i, D_i (ms)
FD-IIR I_a	$1 \times T_{IIR}$	2
PID T_e	$2 \times T_{IIR}$	4
Sensor ω	$1/5 \times T_{PID-\omega}$	4
PID ω	$5 \times T_{PID-Te}$	20
PID pos	$5 \times T_{PID-\omega}$	100

Sistema operativo

FreeRTOS es el SOTR utilizado como soporte fw para la capa de aplicación. El subsistema de control del motor de CD tiene tres tareas periódicas de FreeRTOS y dos subrutinas de atención a interrupción (SAI) que se modelan como tareas periódicas por la naturaleza síncrona de las interrupciones. Estas interrupciones son producidas por temporizadores con período constante; todas ellas se incluyen en el análisis de TR.

La tabla 2 resume los diferentes parámetros del sistema obtenidos del análisis funcional y que deben ser cumplidos por la implementación del fw del sistema durante su operación. Las prioridades (P_i) para FreeRTOS se han asignado según la política del período monotónico, es decir: las prioridades más altas se asignan a las tareas de período más corto y solo se permite una tarea por nivel de prioridad. Además, los plazos son iguales a los períodos de la tarea ($T_i = D_i$). [1], [12]

τ_i	Nombre	P_i	T_i (ms)	D_i (ms)	Tipo
1	FD-IRR I_a	$K + 2$	2	2	SAI
2	Sensor ω	$K + 1$	4	4	SAI
3	Tick FreeRTOS	K	4	<1	SAI
4	PID T_e	$K - 1$	4	4	Task
5	PID ω	$K - 2$	20	20	Task
6	PID pos	$K - 1$	100	100	Task

Es importante destacar que el plazo asignado al núcleo del SOTR se ha establecido en menos de un milisegundo con la intención de imponer una restricción al impacto producido por el planificador sobre el resto de las tareas del sistema. Las SAI asociadas al FD y al cálculo de velocidad a partir de la señal del encoder están programadas con prioridades superiores al núcleo. Estas tareas pueden liberar a otras a través de la API de FreeRTOS sin la necesidad de que intervenga el planificador. Las prioridades se han establecido relativas al núcleo porque el sistema tiene otras tareas con prioridades menores a las aquí analizadas.

Para completar el ATR del sistema es necesario obtener el CPTe y los retardos que sufren las tareas como consecuencia de la atención a interrupciones, el trabajo de FreeRTOS y las interferencias provocadas por la expropiación del procesador cuando tareas de prioridad superior lo necesitan. El CPTe es cardinal en el proceso de comprobación del comportamiento de TR de un sistema. La estimación del CPTe y el resto de las demoras debe ser realista, puesto que si el resultado incluye un alto grado de pesimismo el sistema quedará subutilizado; mientras que una estimación demasiado optimista puede provocar fallos por incumplimiento de plazos.

RESULTADOS Y DISCUSIÓN

Recolección de datos

El sistema fue implementado con MPLAB+C30 y simulado en ISIS-Proteus. El simulador permitió utilizar las

herramientas para análisis gráfico en la obtención de las muestras de tiempo de computación. Cada programa a prueba fue anotado con código en ensamblador para generar un pulso con frente de subida en la primera instrucción y frente de caída en la última. Los resultados se plotearon en un gráfico de análisis digital, desde donde se exportaron a un fichero de texto y posteriormente a otro de base de datos de Paradox. Una aplicación desarrollada en C++ Builder fue utilizada para el procesamiento de los datos y obtención del modelo de Gumbel.

Durante el procesamiento de los datos se descartó el efecto negativo de las instrucciones de trazado y se obtuvieron los tiempos de ejecución para cada instancia de trabajo, el resto del procesamiento consiste en la evaluación de las ecuaciones para la obtención del modelo de Gumbel y en la representación gráfica de los resultados.

En el proceso de medición se obtuvieron muestras del tiempo de ejecución para cada tarea del subsistema de control. Se muestreó al sistema un total de cuatro veces y en cada muestreo se obtuvieron más de cien mil muestras por conjunto muestral. Para la obtención del CPTe se ejecutaron los programas de modo que no existiese interferencia por parte de otros módulos del fw. Además, el código a prueba se estimuló para que transitara por el camino más largo.

En este caso no se realizó análisis estático para determinar un vector de estímulo que obligase a los programas a transitar por su camino más largo, y no se realizó análisis para identificar el camino más largo como se realiza en las técnicas descritas en [25] y [26]. Para obligar a los programas a tomar el camino peor se estimuló cada fragmento de código bajo prueba con un vector de estímulo formado por datos aleatorios en el rango de la variable de entrada para cada caso. Al ejecutar la aplicación durante tiempo suficiente, se espera obtener valores de tiempo de ejecución cercanos al CPTe que se puedan utilizar para modelar el comportamiento estadístico del sistema en la zona extrema del tiempo de computación.

El FD es dependiente de la salida del convertidor análogo digital (ADC) del μC por lo que se estimuló la entrada del ADC con ruido blanco para obtener un vector de estímulo con alta probabilidad de ocurrencia de todos los valores de entrada al FD. La celda PID se estimuló con valores generados aleatoriamente para la señal medida, la referencia y los parámetros de ajuste. Es de destacar que para la estimación del CPTe no es importante disponer de un sistema de control estable sino tratar que el sistema consuma el mayor tiempo posible durante su ejecución. Se utilizó el mismo código para la implementación de las tres celdas de control PID, por lo que el CPTe es el mismo para cada una de ellas, aunque su tiempo de respuesta dependerá de las demoras e interferencias asociadas a cada una.

Cada fragmento de código analizado se muestreó en aislamiento total, de modo que no hubiese interferencia por

parte del núcleo del SOTR, de alguna SAI o de otra tarea. [26] De esta forma se puede garantizar que las muestras temporales obtenidas para cada fragmento de código a prueba, no contengan tiempo de ejecución de otro código. Alimentar los programas a prueba con valores aleatorios en sus parámetros de entrada persigue aumentar el nivel de independencia de la variable aleatoria tiempo de computación para cada instancia de trabajo.

Desarrollo del modelo

Con uno de los resultados muestrales se desarrolló el modelo de Gumbel de cada tarea bajo análisis. Para ello los datos se procesaron según (3) y (4) y se ploteó el tiempo de computación contra las probabilidades acumuladas de \bar{F} y $G_{(0,\mu,\beta)}$. Una buena correspondencia entre ambas curvas reafirma la idea de que el modelo representa al proceso. La gráfica de cuantiles también se utiliza para comparar el modelo de Gumbel con los resultados experimentales; en este caso el modelo debe ajustarse a una línea recta y los resultados medidos deben seguir de cerca al modelo. [16], [20]

La figura 5 muestra el tiempo de computación observado contra las distribuciones de probabilidad empírica $\bar{F}(x)$ y de Gumbel $G_{(0,\mu,\beta)}(x)$ para el código del filtro digital. En general se observa una buena correspondencia entre ambas curvas. Hacia los valores más altos del tiempo de computación ambas funciones de probabilidad tienden a converger. El hecho de que F se aproxime antes a 1, refuerza la idea de que el modelo de Gumbel podrá estimar con pesimismo el tiempo de computación más allá de los valores observados. [20] Este comportamiento es deseado puesto que en los STR es muy importante que el CPTe se estime con pesimismo más que con exactitud. [26]

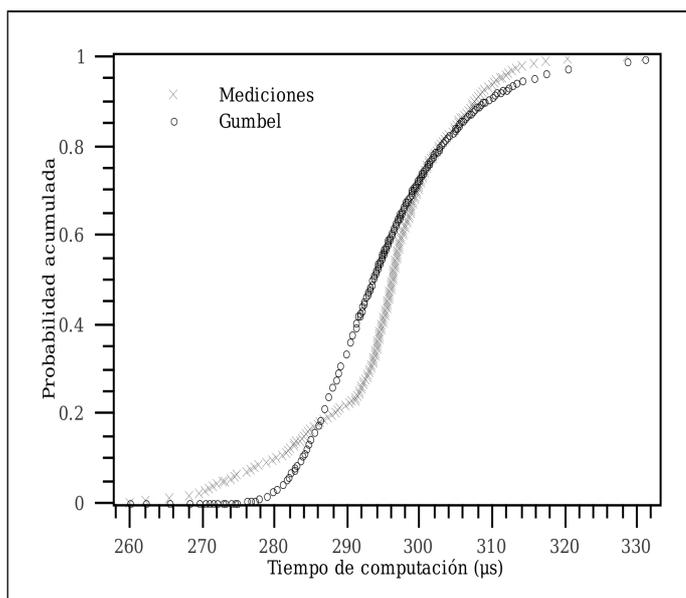


Fig. 5. Tiempo de computación vs $F, G_{(0,\mu,\beta)}$ para el FD.

En la gráfica de cuantiles de la figura 6 se observa una correspondencia cercana entre el modelo y los tiempos de computación observados. Aunque en los extremos, el modelo y los datos de experimento no coinciden, la cercanía entre ambas curvas para la mayor parte de las muestras, refuerza la idea de que el modelo es representativo del proceso.

En los valores medidos se observa una tendencia hacia el decrecimiento del número de muestras para los tiempos de computación más altos, mientras que el modelo de Gumbel se mantiene con una pendiente mayor que el mejor ajuste lineal para las muestras y por encima de los valores observados. La tasa de crecimiento en tiempo de computación del modelo respecto de las mediciones empíricas, es prueba de que este podrá estimar el CPTe con pesimismo.

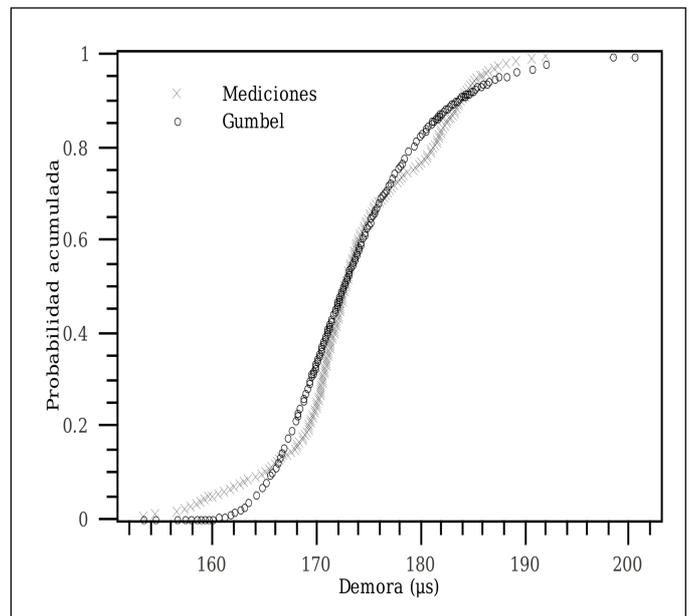


Fig. 6. Tiempo de espera (TE) del PID vs $F, G_{(0,\mu,\beta)}$.

Las figuras 7 y 8 muestran el comportamiento del tiempo de espera del lazo PID en la cola de mensajes que se utiliza para obtener las muestras desde el FD. El tiempo se mide desde el momento en que la muestra está lista, hasta que la tarea del PID es liberada para que comience a procesarla. El análisis del comportamiento se realiza de manera similar al explicado para el CPTe. La única diferencia es que los tiempos estimados en este caso son utilizados como el parámetro demora (J_i) en la estimación del tiempo de respuesta de la tarea analizada.

Los análisis del comportamiento temporal para cada una de las tareas arrojaron resultados similares, siendo los mostrados aquí los casos en que la correspondencia entre el modelo y los datos de muestreo tienen una relación menos cercana.

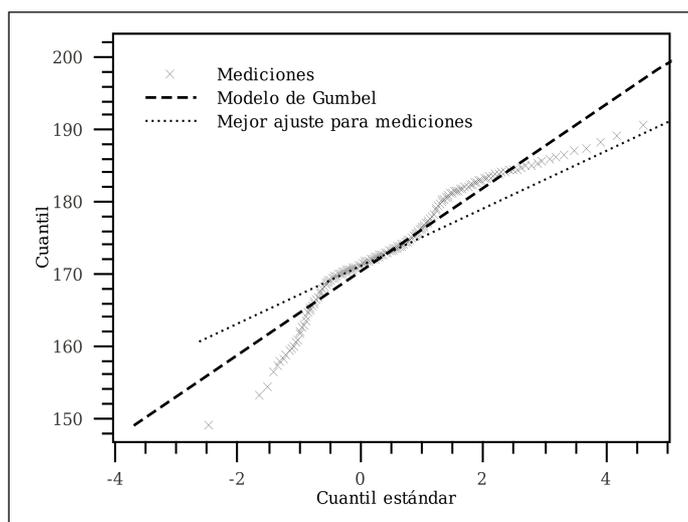


Fig. 7. Gráfico de cuantiles de $F, G_{(0,\mu,\beta)}$ TE del PID.

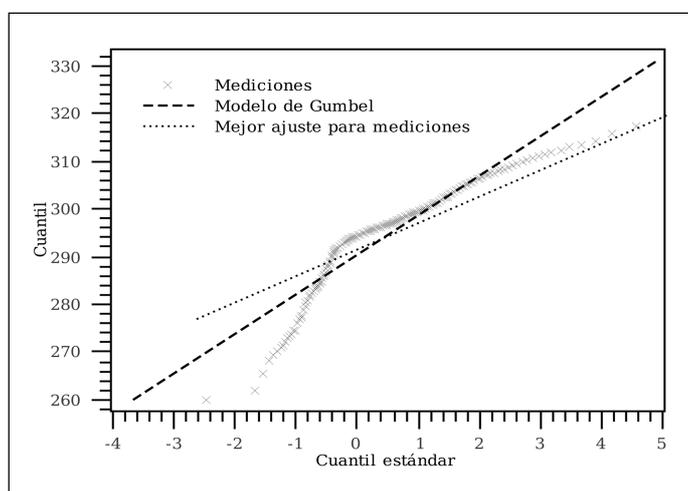


Fig. 8. Gráfico de cuantiles de $F, G_{(0,\mu,\beta)}$ para el FD.

Comprobación del modelo

Una vez obtenido un modelo representativo del tiempo de computación se pueden utilizar los valores de los parámetros estadísticos μ y β de (4) para estimar el tiempo de computación (w) para un valor de probabilidad de excedencia (ε). El proceso se realiza evaluando la inversa $G_{(0,\mu,\beta)}^{-1}(1-\varepsilon)$ de (4). La ecuación resultante para w es (9). Al escoger suficientemente pequeño se puede estimar el más allá de los resultados observados y por consiguiente utilizar estos valores de w como CPTe para el proceso analizado. [18], [20]

$$w = G_{(0,\mu,\beta)}^{-1}(1-\varepsilon) = \mu - \beta \ln(-\ln(1-\varepsilon)) \quad (9)$$

donde:

w : Estimado del tiempo de computación.

ε : Probabilidad de excedencia para w .

La comprobación del modelo se realiza construyendo una tabla de w para diferentes valores de ε y comparando los resultados obtenidos con valores de mediciones realizadas de manera independiente. Si los resultados observados y estimados están lo suficientemente cercanos, se puede considerar que el modelo construido es correcto y se pueden utilizar los w_i que se encuentran más allá de los valores observados como CPTe para una probabilidad de excedencia, del tiempo de computación de la tarea.

La tabla 3 muestra diferentes w_i para la tarea del FD y los datos medidos para comprobar su modelo. En la primera columna se encuentra la probabilidad de excedencia (ε) para los w_i , y los percentiles (per) expresados como $per = 1 - \bar{F}$ de la función de probabilidad acumulada para los conjuntos muestrales de comprobación $T2$, $T3$ y $T4$. Se puede notar que los w_i están siempre por encima del resto de los valores observados en los conjuntos de comprobación, los cuales presentan además una variabilidad muy baja. Para $\varepsilon < 10^{-5}$ no existen resultados en los conjuntos de comprobación, pero gracias al modelo se pueden extrapolar estimados del tiempo de computación. De este comportamiento no puede concluirse que el proceso de estimación siempre será tan satisfactorio. En este caso se podría afirmar que se debe a que la variabilidad del proceso es menor para esta aplicación porque el procesador utilizado no dispone de mecanismos de aceleración de velocidad como memorias *cache* o *pipeline*; por lo que solamente existe aquella asociada a la dependencia del código respecto a sus parámetros de entrada.

ε / per	$w_i (\mu s)$	$T2 (\mu s)$	$T3 (\mu s)$	$T4 (\mu s)$
10^{-1}	309,00	307,68	307,80	307,88
10^{-2}	328,45	317,37	317,60	317,50
10^{-3}	347,54	324,50	324,50	324,50
10^{-4}	366,60	329,50	330,30	329,50
10^{-5}	385,66	331,20	334,00	331,13
10^{-6}	404,71
10^{-7}	423,77
10^{-8}	442,82
10^{-9}	461,88

Selección del CPTE

Seleccionar un valor apropiado de w como CPTE es el próximo paso en el proceso del análisis de TR. El valor elegido de w debe ser tal que se garantice el correcto desempeño del sistema durante su explotación. El problema está relacionado con la selección de un valor apropiado de ε que permita obtener un w considerado seguro. Para ello es importante conocer con qué nivel de confianza w es probablemente el CPTE. Basado en el modelo de Gumbel para la estimación de w no es posible especificar un valor de confianza.

La ecuación (10) es la definición de θ en términos del modelo de Gumbel y el máximo observado en el conjunto muestral desde el que se construyó el modelo [20]. La función θ es la función de densidad de excedencia de pertenencia y representa la confianza en la probabilidad de exceder el tiempo de computación estimado a partir del máximo tiempo de ejecución observado con distribución de probabilidad $G_{(0,\mu,\beta)}$ cuyo máximo se denota (m) [19], [20].

$$\theta_{(\mu,\beta)}(x) = G_{(0,\mu,\beta)}^{(m)}(x) = \frac{G(x) - G(m)}{1 - G(m)} \quad x > m \quad (10)$$

donde:

θ : Probabilidad de exceder el tiempo de computación estimado.

m : Máximo valor observado en el conjunto muestral.

Al evaluar (10) para los valores del tiempo de ejecución observado se obtiene como resultado el nivel de confianza θ para un estimado del tiempo de ejecución. Para completar el análisis se requiere una ecuación que permita estimar el tiempo de ejecución referido a θ ; la ecuación (11) presentada por Edgar en [20] permite calcular este estimado a partir del máximo (m) observado, una probabilidad de excedencia de pertenencia especificada ε y los estadísticos del modelo de Gumbel μ y β .

$$W = \mu - \beta \ln(-\ln(\varepsilon \cdot G(m) + (1 - \varepsilon))), \quad \varepsilon \in [0,1) \quad (11)$$

donde:

W : Tiempo de computación estimado y referido a θ .

ε : Probabilidad de excedencia de pertenencia para W .

La figura 9 muestra el modelo de Gumbel para el FD y su función θ correspondiente. Se puede observar que θ siempre producirá estimados de CPTE más pesimistas para el mismo valor de ε : que los que se pueden obtener del modelo. Sin embargo, los estimados por θ tienen asociada la confianza acerca de W .

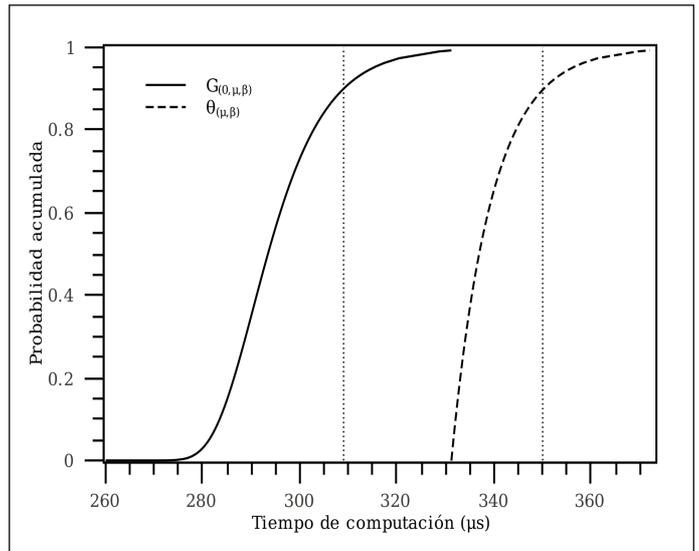


Fig. 9. Tiempo de computación vs $G_{(0,\mu,\beta)}$, $\theta_{(\mu,\beta)}$ para el FD.

En este caso se han marcado los estimados para el modelo de Gumbel del proceso (w) y el de su θ correspondiente (W) , ambos con $\varepsilon = 0,1$. Sobre w se puede decir que el 90 % de las veces el tiempo de computación será menor que $309 \mu s$, mientras que para W se puede decir que existe evidencia para afirmar que $350 \mu s$ es el CPTE con un 90 % de confianza.

La decisión de utilizar w o W como CPTE dependerá de la calidad percibida del modelo y de los requisitos para la tarea en cuestión. Si una tarea permite utilizar un CPTE para el cual no se requiere establecer un nivel de confianza, entonces utilizar w con valores de ε suficientemente pequeños bastará, siempre que el modelo se perciba confiable. Por otro lado si se exige que el CPTE tenga asociado un nivel de confianza, entonces lo más acertado es utilizar estimados W . En general, para STR blandos se pueden utilizar estimados w , mientras que para el resto es apropiado utilizar estimados W .

Debido a la gran variedad de problemas que se presentan en la práctica y a los diferentes requerimientos temporales que estos pueden demandar, determinar con qué nivel de confianza e incluso que tipo de estimado utilizar constituye un problema en sí mismo. Es por ello que Edgar y Burns proponen en [19] un procedimiento para optimizar en alguna medida el nivel de pesimismo con que se seleccionarán los estimados W . Posteriormente, Edgar en [20] describe este proceso en detalle. La dificultad de utilizar el procedimiento anteriormente citado radica en que se impone al sistema un conjunto de restricciones que no se cumplen en esta aplicación y por tanto su aplicabilidad no sería acertada.

Esta sección se ha centrado en la selección del caso peor de tiempo de ejecución (CPTE). Sin embargo, el mismo procedimiento se aplica a la determinación del caso peor de

demora que será utilizado como el parámetro J_i en el análisis del tiempo de respuesta (ATR).

Para seleccionar un CPTe seguro se utilizó un criterio simple pero efectivo. Este consiste en asignar a todas las tareas del subsistema de control una relativamente muy baja probabilidad de fallo a partir de un elevado nivel de confianza en para cada tarea. La tabla 4 resume los W con nivel de confianza del 99,99% $\Rightarrow \varepsilon = 10^{-4}$ para todas las tareas del subsistema de control, mientras que la tabla 5 expone los estimados W pero para la demora sufrida por cada una de las celdas de control PID. Las tablas contienen el máximo valor observado durante el proceso de toma de muestras (m) y el incremento porcentual de W respecto de m . Con estos casos peores tanto para tiempo de ejecución como para retardo se aplica el método del ATR, presentado al principio del trabajo, para determinar si las tareas cumplen con las exigencias de tiempo real.

τ_i	Nombre	m_i (μs)	C_i (μs)	Incremento
1	FD-IRR I_a	331,20	407,45	23 %
2	Sensor ω	147,30	178,40	21 %
3	Tick FreeRTOS	173,41	205,23	18 %
4, 5, 6	Celda PID	307,60	389,15	27 %

τ_i	Nombre	m_i (μs)	J_i (μs)	Incremento
4	PID T_e	200,50	253,79	27 %
5	PID w	273,20	340,60	25 %
6	PID ρ_{os}	320,40	402,52	26 %

En este análisis se ha considerado despreciable la latencia del proceso de interrupción que sufren las tareas que se implementan como SAI, este valor de $10 \mu s$ es fijo para todas, y es el tiempo que necesita el procesador para almacenar en memoria los registros que se requieren para restablecer el contexto en el momento de la interrupción. Como se puede observar, es un tiempo relativamente corto en comparación con los tiempos de computación de las tareas o los retardos que sufren las tareas de los lazos de control.

Índice de utilización

El índice de utilización de un sistema de cómputo es una de las ecuaciones publicadas por Liu y Layland en 1973 [12],

actualmente se utilizan como una medida del gasto computacional bruto de un conjunto de tareas. Aún cuando las restricciones que impone el modelo de Liu y Layland sobre el sistema no son las mismas que las de la aplicación aquí analizada. Evaluar (12) permite tener una idea general de la demanda sobre la capacidad de procesamiento de las tareas del sistema en el momento en que todas son liberadas al mismo tiempo, instante de tiempo conocido como hiperperíodo.

$$U = \sum \frac{C_i}{T_i} \leq 1 \quad (12)$$

donde:

U : Tiempo de la tasa de utilización del procesador.

C_i : Tiempo de computación de τ_i .

T_i : Período de τ_i .

El resultado obtenido indica que durante el hiperperíodo del subsistema de control, la demanda de esta parte de la aplicación sobre el procesador es del 42,03 % de su capacidad total. Sin embargo, este dato no debe ser motivo de confusión, en el sentido de suponer que existe una disponibilidad del 57,97 % de la capacidad de procesamiento, Otro de los resultados del trabajo de Liu y Layland indica que un sistema que sobrepase el límite teórico inferior que se calcula según (13) podría ser no planificable. [12], [27]

$$U = \sum \frac{C_i}{T_i} \leq n \left(2^{\frac{1}{n}} - 1 \right) \quad (13)$$

donde:

n : Número de tareas.

En este caso y con el conjunto de tareas analizado, la disponibilidad real del procesador para que no exista sobrecarga se limita al 73,48 %, y en la medida que se agreguen tareas al sistema esta disponibilidad disminuirá.

Análisis del tiempo de respuesta (ATR)

El análisis del tiempo de respuesta fue el método que permitió solucionar, en parte, las limitaciones que se imponían a un sistema para que pudiese ser analizada su planificabilidad según el método anteriormente mostrado. Este método fue presentado al inicio de este trabajo, sin embargo, la ecuación básica ha sido mejorada para ser aplicable a sistemas donde se utilice un planificador expropiativo como es el de FreeRTOS. Para el ATR se utilizó la ecuación (14) [14], [27], la cual es una extensión de (2) [13]. Esta ecuación incluye en el análisis bloqueos y retardos, además es apropiada para sistemas con planificadores expropiativos de prioridades fijas, y plazos menores o iguales a los períodos.

$$R_i^{k+1} = B_i + C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_j^k + J_j}{T_j} \right\rceil C_j \quad (14)$$

donde:

$hp(i)$: Conjunto de tareas con prioridad mayor que τ_i .

R_i : Tiempo de respuesta de τ_i .

B_i : Bloqueo que sufre τ_i .

C_i, C_j : Tiempo de computación de τ_i y τ_j .

J_i, J_j : Demora o retardo (*jitter*) que sufre τ_i y τ_j .

Para evaluar la ecuación (14) se comienza por asignar valores a R_i^0 , como norma $R_i^0 = C_i + B_i$. El proceso se detiene cuando $R_i^{k+1} = R_i^k \leq D_i - J_i$ en cuyo caso el tiempo de respuesta es o cuando se obtiene como resultado $R_i^{k+1} > D_i - J_i$ situación que convierte a la tarea en una no planificable. Se considera que el sistema es planificable si todas sus tareas son planificables.

La tabla 6 muestra los tiempos de respuesta para todas las tareas analizadas, en todos los casos las tareas son planificables por lo que el sistema es planificable y puede considerarse de tiempo real.

τ_i	Nombre	R_i	D_i (μs)	J_i (μs)	$D_i - J_i$ (μs)
1	FD-IRR I_a	407,45	2 000	0	2 000
2	Sensor	585,85	4 000	0	4 000
3	Tick FreeRTOS*	791,08	1 000	0	1 000
4	PID T_e	1 180,23	4 000	253,79	3 746,21
5	PID w	1 569,38	20 000	340,60	19 659,40
6	PID pos	1 958,53	1000 000	402,52	99 597,48

* Única tarea con período diferente al plazo: T3 4000 μs .

CONCLUSIONES

El análisis funcional de la aplicación permite definir parámetros que se utilizan en la obtención del tiempo de respuesta del sistema. En este caso se obtuvieron los períodos de las tareas y los plazos para la entrega de la respuesta. Durante esta etapa se pueden determinar características de componentes a implementar en sw que pueden tener un impacto determinante en el gasto computacional del sistema. El orden del FD escogido permitió obtener una buena respuesta en la salida del sistema con una reducción del gasto computacional de su implementación sw.

La obtención del CPTE mediante mediciones y el modelo de Gumbel permiten determinar el parámetro tiempo de computación para la evaluación del modelo de ATR escogido. Tradicionalmente el CPTE se ha obtenido mediante análisis estático, lo que requiere el modelado del hw y el análisis del sw a bajo nivel. El método aquí expuesto no requiere modelar el hw, ni analizar el sw a bajo nivel, lo que permite su fácil aplicación en otras plataformas de desarrollo hw/sw.

Una vez obtenidos los parámetros del sistema y el CPTE se puede evaluar directamente un modelo de ATR. El modelo de ATR debe ser cuidadosamente escogido, en este caso se seleccionó uno relativamente sencillo, con la intención expresa de demostrar la aplicabilidad de la teoría de análisis de tiempo real.

Para el subsistema analizado se demostró que se comporta en tiempo real y que existe disponibilidad en los recursos de computación. Este resultado permite el desarrollo del resto de los componentes del sistema y la realización del análisis de tiempo real de los mismos en la medida en que se completan.

El método de análisis presentado puede ser aplicado a sistemas de tiempo real firmes y blandos, ya que aunque se pueden imponer restricciones altas sobre la probabilidad de fallo en el cumplimiento de plazos no existen garantías para un sistema de tiempo real estricto. Tampoco se puede obtener la tasa de fallos probables durante un período de tiempo, lo que impide que esta técnica pueda ser utilizada en el desarrollo de sistemas de tiempo real estricto-debilitados.

RECONOCIMIENTOS

Los autores agradecen al doctor Alejandro Cabrera Sarmiento por la revisión del manuscrito y sus oportunos consejos, los cuales contribuyeron a mejorar la calidad de este trabajo.

REFERENCIAS

1. **LAPLANTE, P. A.** *Real-Time Systems Design and Analysis*. IEEE Press, 2004, 530 pp. ISBN 0-471-22855-9.
2. **BURNS, A. and WELLINGS, A. J.** *Concurrent and Real-Time Programming in Ada*. Cambridge University Press, 2007, 477 pp. ISBN 978-0-521-86697-2.
3. **LI, Q. and YAO, C.** *Real-Time Concepts for Embedded Systems*. Elsevier, 2003. 321 pp. ISBN 1-57820-124-1.
4. **BURNS, A.** "Scheduling Hard Real-Time Systems: a Review". *Software Engineering Journal*, 1991, pp. 116-128.
5. **BERNAT, G.; BURNS, A. and LLAMOSÍ, A.** "Weakly Hard Real-Time Systems". *IEEE Transactions on Computers*, 2001, vol. 50, núm 4, pp. 308-321.
6. **SINGHOFF, F. et al.** "Investigating the Usability of Real-Time Scheduling Theory with the Cheddar Project". *Real-Time Systems*. 2009, vol 43, núm 3, pp. 259-295. doi 10.1007/s11241-009-9072-y.
7. **BERNAT, G. and BURNS, A.** "Three Obstacles to Flexible Scheduling". In *13th Euromicro International Conference on Real-Time Systems*, 2001, pp. 11-18.

8. KRISHNAN, R. *Electric Motor Drives. Modeling, Analysis and Control*. Prentice Hall, 2001, 652 pp. ISBN 0-13-0910147.
9. SUH, S.H. *et al. Theory and Design of CNC Systems*. Springer - Verlag, 2008, 466 pp. ISBN 978-1-84800-335-4.
10. MITRA, S.K. *Digital Signal Processing A Computer-Based Approach*, 1st ed. McGraw-Hill, 1998, 880 pp, ISBN 0-07-042953-7.
11. OPPENHEIM, A.V.; SCHAFFER R. W. and BUCK, J. R. *Discrete-Time Signal Processing*, 2nd ed. Prentice-Hall, 1999, 870 pp., ISBN 0-13-754920-2.
12. LIU, C.L. and LAYLAND, J. W. "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment". *Journal of the ACM*, 1973, vol. 20, núm 1, pp. 46-61.
13. JOSEPH, M.; "Finding Response Times in a Real-Time System". *The Computer Journal*, 1986, vol. 29, pp. 390-395.
14. AUDSLEY, N.C.; BURNS, A. and WELLINGS, A. J. "Deadline Monotonic Scheduling Theory and Application". *Control Eng. Practice*, 1993, vol 1, pp. 71-78.
15. CHENG, A. M. K. *Real-Time Systems. Scheduling Analysis and Verification*. Wiley Interscience, 2002, 550 pp, ISBN 0-471-18406-3.
16. CASTILLO, E. *et al. Extreme Value and Related Models with Applications in Engineering and Science*. Wiley-Interscience, 1998, 367 pp.
17. KOTZ, S. and NADARAJAH, S. *Extreme Value Distributions. Theory and Applications*. Imperial College Press, 2000, 190 pp., ISBN 1-86-094224-5.
18. BURNS, A. and EDGAR, S. "Predicting Computation Time for Advanced Processor Architectures". In: *12th EUROMICRO Conference on Real-time Systems*, 2000.
19. EDGAR, S.; BURNS, A. "Statistical analysis of WCET for Scheduling". In *IEEE RTSS 2001*, 2001.
20. EDGAR, S. F. "Estimation of Worst-Case Execution Time Using Statistical Analysis". Thesis, University of York, September 2002.
21. TIPSUWAN, Y. and CHOW, M.-Y. "Fuzzy Logic Microcontroller Implementation for DC Motor Speed Control". In *25th Annual Conference of the IEEE, Industrial Electronics Society*, 1999.
22. HUGES, A. *Electric Motors and Drives. Fundamentals Types and Applications*, 3rd ed. Elsevier-Newnes, 2006, 431 pp., ISBN 978-0-7506-4718-2.
23. JAMES, K. *PC Interfacing and Data Acquisition: Techniques for Measurement, Instrumentation and Control*. Newnes, 2000, 471 pp, ISBN 0-7506-4624-1.
24. PARK, J. and MACKAY, S. *Practical Data Acquisition for Instrumentation and Control Systems*. Newnes, 2003, 425 pp. ISBN 0-7506-5796-0.
25. PUSCHNER, P.; SCHEDL, A. "Computing Maximum Task Execution Times a Graph-Based Approach". *Real-Time Systems*, 1995, pp. 67-91.
26. PUSCHNER, P. and BURNS, A. "A Review of Worst-Case Execution Time Analysis". *Real-Time Systems*, 2000, pp. 115-128.
27. DAVIS, R. and BURNS, A. "Response time upper bounds for fixed priority real-time systems". RT: Real-Time Systems Research Group, Department of Computer Science. University of York, 2008.

AUTORES

Reinier Torres Labrada

Ingeniero en Automática, Instructor, Departamento de Automática y Computación, Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría, Cujae, La Habana, Cuba

Mailén Sarmiento Fong

Ingeniera en Automática, Instructora, Departamento de Automática y Computación, Facultad de Ingeniería Eléctrica, Instituto Superior Politécnico José Antonio Echeverría, Cujae, La Habana, Cuba.

Desing and Real-Time Analysis of a Direct Current Motor Controller

Abstract

The real time analysis of a DC motor controller is addressed. During the functional analysis, deadline and period of tasks were extracted to be used as parameters for Response Time Analysis (RTA). To get the Worst Case Execution Time (WCET) each task is stimulated with a random input vector. A great number of computation time samples were recorded and the WCET computed using the Gumbel statistical distribution. With the WCET, deadlines and periods of the task set under analysis a RTA model is evaluated and its results discussed. By modeling the WCET using the Extreme Values Theory (EVT) the need of low level analysis for software and an accurate model for hardware is avoided. The method used here is useful for soft real-time systems and could be applied to other hw/sw platforms easily.

Key words: real-time system, response time, real-time operating system, digital signal processing, PID