

ANÁLISIS Y COMPENSACIÓN DE LOS RETARDOS DE PLANIFICACIÓN EN SISTEMAS DE CONTROL

Patricia Balbastre, Manuel Lluesma, Ismael Ripoll

*Departamento de Informática de Sistemas y Computadores
Universidad Politécnica de Valencia, Valencia, España*

Resumen: Existe una separación entre la fase de diseño del regulador y su implementación en el computador de forma que, cada fase asume simplificaciones sobre la otra fase, lo cual lleva a un comportamiento inesperado o incorrecto del sistema a controlar. Desde hace algunos años, han surgido diversos trabajos que han apuntado la falta de integración de las dos fases de control. Esta línea de investigación se conoce como integración del control y la planificación. El estudio de los retardos que sufre un sistema de control es una parte importante de esta línea de investigación. El objetivo de este trabajo es presentar la revisión y análisis tanto de los retardos de un sistema de control, centrándonos en los retardos de control, como de las técnicas utilizadas para reducir estos retardos. *Copyright © 2006 CEA-IFAC*

Palabras clave: Sistemas de tiempo real, planificación, compensación de retardos

1. INTRODUCCIÓN

Actualmente, las aplicaciones de control de tiempo real juegan un papel muy importante en la tecnología moderna. Por ejemplo, el sistema de antibloqueo de frenada en los coches de hoy en día son controlados por sistemas embebidos de tiempo real. Conforme aumenta la capacidad de procesamiento de los microcontroladores y disminuye su coste, la funcionalidad realizada por software es mayor. Por lo tanto, en los sistemas de control embebidos, una tarea de control es típicamente ejecutada junto a otras tareas. Por ejemplo, el sistema de control embebido que controla el sistema de frenada de un coche, podría constar de tres tareas con diferentes prioridades y funcionalidad: la tarea más prioritaria controlando el proceso de antibloqueo, la segunda tarea, con una prioridad

media, controlando la presión realizada sobre el pedal de freno y finalmente, la última tarea y menos prioritaria controlando la degradación de las pastillas de freno. De esta manera, este planteamiento (varias tareas compartiendo el mismo microcontrolador), necesita del concepto de la planificación. Es decir, qué tarea es la elegida para ejecutarse en un momento determinado. Por otra parte, la teoría de control (en la etapa de diseño) normalmente asume que los controladores son ejecutados en un único bucle y en un computador dedicado.

Esta separación entre las fases de diseño del controlador y planificación en un sistema de tiempo real conlleva a obtener una solución no óptima. Por lo tanto, para alcanzar las mejores prestaciones posibles en el sistema de control es necesaria-

rio el codiseño entre la plataforma computacional (donde serán implementados y planificados los diferentes controladores) y el sistema de control.

De esta manera, la motivación principal de este trabajo es analizar una de las causas de la degradación de la respuesta de un sistema de control: los retardos que aparecen en un sistema de control de tiempo real, prestando especial atención a los retardos introducidos por el planificador. Se estudiará y analizará cómo estos retardos afectan a las prestaciones finales del sistema a controlar, utilizando un sistema clásico de control: el péndulo invertido.

El resto del trabajo se divide de la siguiente forma: En la sección 2, se describen los retardos que pueden aparecer en un sistema de control de tiempo real. Seguidamente, en la sección 3, se analiza la respuesta del sistema en el caso ideal y real cuando se trata de controlar un sistema formado por tres péndulos invertidos. En la sección 4, se estudian y comparan dos técnicas para reducir los retardos de control y finalmente, en la última sección, se presentan las conclusiones y el trabajo futuro.

2. NATURALEZA DE LOS RETARDOS

En el esquema básico de un bucle de control el regulador calcula la acción de control que será aplicada sobre el proceso. Tras la reacción del sistema ante esta entrada se obtiene la respuesta del sistema. El error, definido como la diferencia entre la salida real del proceso y el valor ideal que se desea que tenga esta salida, entra como parámetro al regulador, el cual vuelve a calcular la acción de control en una nueva iteración. Lo más habitual es implementar este bucle de control como una tarea periódica en el computador. El periodo de muestreo T es la distancia temporal entre dos acciones del bucle de control. Es decir, cada T instantes de tiempo, se tomarán datos de la salida, se calculará la acción de control y se enviará al proceso. Desde el punto de vista de control, es deseable que el periodo de muestreo sea lo más pequeño posible, para conseguir que el control digital (discreto) se aproxime lo más posible al control continuo. Sin embargo, desde el punto de vista del computador, el periodo de muestreo viene restringido por la velocidad de los sensores y actuadores y por el tráfico introducido en la red de comunicaciones al aumentar la frecuencia.

Muchos sistemas de control por computador se implementan como sistemas multiprocesador o distribuidos, donde pueden aparecer problemas de tiempo. Incluso en sistemas monoprocesador, pueden aparecer retardos, entre que se reciben datos del proceso y se envía la acción de control.

En (Torngren, 1998) se presenta una perspectiva general de los requerimientos temporales de los sistemas de control, además de un estudio de los parámetros que afectan a la respuesta de estos sistemas. Törngren (Torngren, 1995) clasifica los retardos que pueden aparecer en un sistema de control en tres tipos:

- Retardos en el sistema de control. Causados por sistemas cuya dinámica hace muy difícil obtener medidas sin retrasos.
- Retardos inherentes al muestreo. Debido, por ejemplo, a la utilización de retenedores de orden cero, interfaces de los sistemas de adquisición (convertidores A/D y D/A), etc.
- Retardos debidos a los recursos computacionales. Son los retardos introducidos por el computador.

Dentro de los retardos introducidos por los recursos computacionales, se pueden identificar los siguientes tipos de retardos:

- Retardo de comunicación entre el sensor y el controlador.
- Retardo computacional. Es lo que se tarda en calcular la acción de control. En este retardo, no solo debe tenerse en cuenta el tiempo empleado por la tarea en calcular la acción de control sino también la política de planificación utilizada, o planificador de tiempo real. Cuando una tarea está en ejecución y se activa una tarea más prioritaria, la tarea en ejecución es expulsada, hasta que la más prioritaria acabe. Sólo cuando esta primera tarea expulsada sea la más prioritaria entre todas las tareas activas del conjunto podrá ejecutarse. El tiempo que está en espera de ejecución por expulsiones de tareas más prioritarias es el llamado *retardo introducido por el planificador*.
- Retardo de comunicación entre el controlador y el actuador.

En la Figura 1 se muestran gráficamente estos retardos. El retardo de comunicación entre el sensor y el controlador se produce desde que se realiza la petición al sensor, hasta que el dato se recibe. Con este dato, el controlador calcula la acción de control. Este tiempo es el retardo computacional que, como mínimo será igual que el propio tiempo de cómputo de la tarea encargada de realizar el control. Pueden afectar tiempos de espera de recursos compartidos, expulsiones de otras tareas, etc, haciendo mayor este retardo. Finalmente, la acción de control se envía, sufriendo un retardo hasta que llega al actuador.

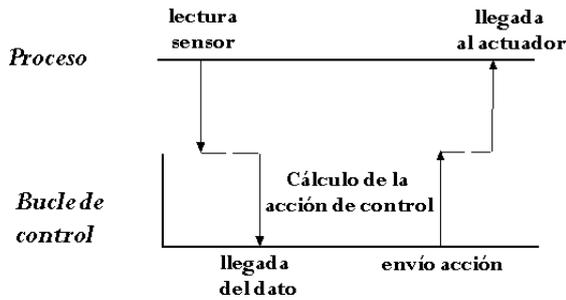


Figura 1. Retardos en el bucle de control

2.1 Retardos fijos y variables

Existe otra clasificación en función de la variabilidad de los retardos a lo largo de la ejecución del sistema. De esta forma, en el bucle de control, los retardos también pueden clasificarse como:

- *Retardo fijo*: Es aquel que no va a poder minimizarse, y debe tenerse en cuenta en el diseño del regulador cuando su valor no pueda despreciarse. El tiempo de cómputo de una tarea es un retardo fijo.
- *Retardo variable*: Es aquel retardo que varía de una activación a otra y es debido a diversas causas: expulsiones debido a distintas prioridades entre tareas, bloqueos por acceso a recursos compartidos en uso, exactitud del reloj de tiempo real, etc.

En la ecuación del tiempo de respuesta de una tarea T_i en una activación cualquiera k se pueden identificar las partes fijas y variables. La expresión es:

$$r_{ik} = O_i + C_i + \sum_{j \in hp(i)} C_j \left[\frac{r_{jk}}{P_j} \right]$$

En la ecuación anterior se observa como el tiempo de respuesta de una tarea está compuesto por el desplazamiento de la tarea, su tiempo de cómputo y las interferencias de tareas más prioritarias (el término del sumatorio). Tanto el desplazamiento O_i como el tiempo de cómputo C_i son parámetros que se consideran constantes en todas las activaciones (se recuerda que el tiempo de cómputo C_i se mide en términos de peor caso), por lo que suponen retardos fijos para la tarea T_i . Sin embargo, el término de interferencia de otras tareas puede variar de una activación a otra. Éste es el retardo variable.

3. INFLUENCIA DE LOS RETARDOS EN EL CONTROL

En el análisis y diseño de sistemas de control por computador, en general, se supone que estos retardos son despreciables. Por lo tanto, no son tenidos en cuenta. El regulador digital proporciona una acción de control óptima para el periodo

T para el que ha sido diseñado. Si se produce cualquier retraso en el muestreo, o en el envío de la acción de control, el periodo real al que trabaja el sistema puede diferir mucho del periodo nominal para el que ha sido diseñado el regulador, degradándose de esta forma la respuesta del sistema (Mita, 1985). Por tanto, estas interferencias deben ser eliminadas o minimizadas en la medida de lo posible.

Para mostrar la influencia de los retardos en los sistemas de control se utilizará el siguiente ejemplo.

3.0.1. Definición del proceso El proceso utilizado es el conocido péndulo invertido. El sistema simulado se muestra en la figura 2.

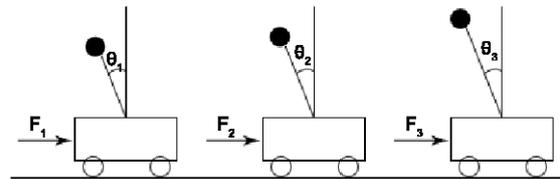


Figura 2. Sistema de tres péndulos invertidos

El objetivo es estudiar la respuesta del sistema analizando la estabilidad, el input jitter, la latencia de entrada-salida y el CAI. Para ello se utilizará el toolbox de MATLAB/Simulink llamado TrueTime desarrollado en la Universidad de Lund (Henriksson and Cervin, 2003).

Como se muestra en la figura 2, el sistema de control se compone de tres péndulos invertidos. Cada uno de ellos se caracteriza por una función de transferencia de la siguiente forma:

$$G(s) = \omega_0/s^2 - \omega_0^2$$

donde ω_0 es la frecuencia de cada péndulo invertido.

La figura 3 muestra el modelado de un péndulo invertido en MATLAB/Simulink. La posición, fricción y masa del carrito no han sido modelados.

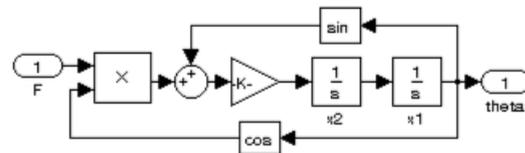


Figura 3. Modelo simulado de un péndulo invertido MATLAB/Simulink

En la tabla 1 se muestran las diferentes longitudes (L) y frecuencias (ω) para cada péndulo. La relación entre ambos parámetros es $\omega^2 = g/L$.

Además, tanto el proceso como las medidas sufren la perturbación de un ruido blanco continuo con varianza igual a la unidad.

	Longitud (L)	Frecuencia (ω)
Péndulo 1	0.1	9.9
Péndulo 2	0.2	7.0
Péndulo 3	0.3	5.7

Tabla 1. Características de los tres péndulos

El controlador diseñado es un controlador LQG (Linear-Quadratic-Gaussian), cuya estructura se muestra en la Figura 4. Este controlador ha sido diseñado utilizando la herramienta Jitterbug (Cervin and Lincoln, 2003), en concreto con el comando *lqgdesign* para cada péndulo.

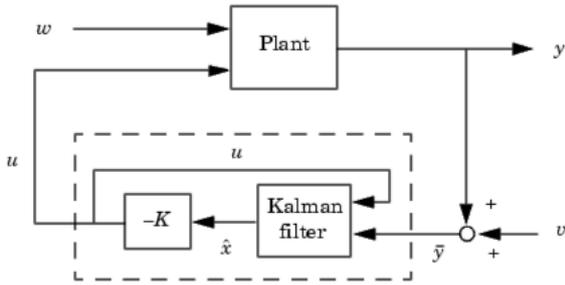


Figura 4. Regulador LQG

Cada controlador es ejecutado periódicamente. El periodo de muestreo ha sido escogido según establece (Åström and Wittenmark, 1997) de la siguiente forma:

$$\omega_c * T = [0,2, 0,6]$$

donde ω_c es el ancho de banda del sistema en bucle cerrado. Si el producto $\omega_c * T$ se fija al valor 0.4 (menores valores producen menos jitter pero una alta utilización) y se considera la frecuencia de Nyquist como:

$$\omega_c = 2 * \omega_0$$

entonces, el periodo de muestreo para cada controlador es T=20, 29, 35 ms, respectivamente.

La Figura 5 muestra el modelo en TrueTime para estos tres péndulos.

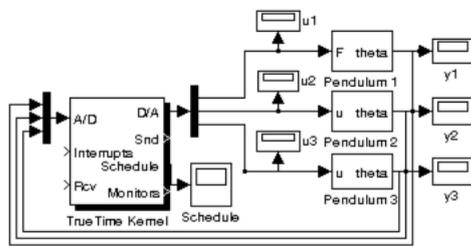


Figura 5. Modelo Truetime

3.0.2. Respuesta ideal y respuesta real En el caso ideal, se considera que la plataforma computacional no afecta en la ejecución del sistema, que no existen retardos, los tiempos de cómputos de las tareas son despreciables, etc. En la figura 6

se muestra la respuesta ideal de los tres péndulos en bucle cerrado.

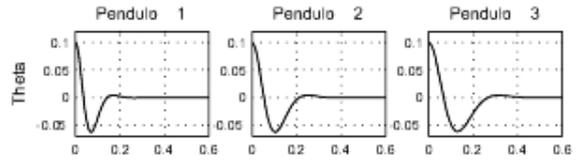


Figura 6. Respuesta ideal de los tres péndulos

Sin embargo, cuando el sistema es ejecutado en un sistema de tiempo real, podremos observar como la respuesta de los péndulos no es la esperada. Para ejecutar el proceso se considerarán tres tareas, una por cada controlador para cada péndulo. Los parámetros de estas tareas se muestran en la tabla 2. Los periodos corresponden a los periodos de muestreo obtenidos para cada péndulo. Además, el tiempo de cambio de contexto no se ha considerado despreciable, sino que se establece en 500 ns. El algoritmo de planificación utilizado es Deadline Monotonic (DM)(Liu and J.W.Layland., 1973).

	C	D	P	Prioridad
T_1	7	20	20	1
T_2	7	29	29	2
T_3	7	35	35	3

Tabla 2. Modelo de tareas con planificación DM

La respuesta de los tres péndulos no es la esperada, sino que sufre una degradación con respecto a la respuesta ideal esperada por los diseñadores del regulador. Todos ellos alcanzan la referencia demasiado tarde. La figura 7 muestra la respuesta para el péndulo 3 (resultados muy similares de han obtenido para los péndulos 1 y 2). En este caso la referencia se alcanza tras 1 segundo, cuando en el caso ideal se alcanza en 0.4 segundos, aproximadamente.

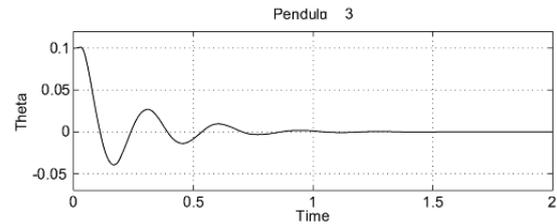


Figura 7. Respuesta real para el Pendulo 3

3.1 Trabajos relacionados

Un estudio completo de los retardos introducidos por el planificador es presentado por Albertos y Crespo en (Albertos and Crespo, 1999). Además, se introduce de forma experimental el concepto de *Intervalo de Acción de Control (CAI)* e *Intervalo de Adquisición de Datos (DAI)*. Se definen de la siguiente forma:

Definición 1. El *DAI* es el Intervalo de Adquisición de Datos¹. Expresa el porcentaje de variación respecto del periodo de la adquisición de datos. Para una tarea T_i sigue la siguiente expresión:

$$DAI_i = \frac{WCAT_i - BCAT_i}{P_i} 100$$

siendo $WCAT_i$ y $BCAT_i$ los mejores y peores tiempos de activación de la tarea T_i , respectivamente.

Definición 2. El *CAI* es el Intervalo de Acción de Control² y expresa el porcentaje de variación respecto del periodo del envío de la acción de control. El CAI proporciona información sobre el retardo que va a sufrir una tarea y permite comparar las respuestas de un sistema de control. Para una tarea T_i se expresa como:

$$CAI_i = \frac{WCRT_i - BCRT_i}{P_i} 100$$

siendo $WCRT_i$ y $BCRT_i$ los mejores y peores tiempos de respuesta de la tarea T_i , respectivamente.

En realidad, estos dos parámetros miden el retardo variable que sufre una tarea en el inicio de su ejecución (*DAI*) y en el final de la misma (*CAI*). Si el *CAI* de una tarea es cero, entonces la tarea envía la acción de control de forma síncrona, es decir, entre dos envíos consecutivos pasan T unidades de tiempo. Si el *DAI* de una tarea es cero, la toma de datos también se realizará de forma síncrona, lo cual no quiere decir que suceda lo mismo con el envío de la acción de control, medido por el *CAI*. Intuitivamente, si una tarea es siempre la más prioritaria, tanto su *CAI* como su *DAI* serán cero, ya que ninguna otra tarea la expulsara en su ejecución. Además, aunque una tarea tenga un *CAI* o un *DAI* nulo, esto no significa que tenga un retardo fijo nulo. Al menos, cada tarea tendrá un retardo fijo igual a su tiempo de cómputo en el peor caso. Como ya se ha comentado anteriormente, esto no es excesivamente preocupante, ya que un retado fijo puede ser compensado en el regulador.

Es importante hacer notar que es más prioritario reducir el *CAI* que el *DAI*. En primer lugar, porque de nada sirve recibir los datos del proceso sin retardo si luego la acción de control se ve retrasada y no se puede enviar a tiempo. En segundo lugar, si la toma de datos tiene retardo, éste retardo puede ser tenido en cuenta en el cálculo de la acción de control. En otras palabras, el *DAI* puede ser compensado en el algoritmo de

control, de forma que el algoritmo tenga en cuenta que los datos tienen una marca de tiempo mayor de la debida. Sin embargo, si el retraso se produce al enviar la acción de control (*CAI* no nulo), éste retardo no puede ser tenido en cuenta, ya que la acción de control ya ha sido calculada. Por tanto, es más importante reducir el *CAI*, ya que el *DAI* puede ser compensado.

Respecto al ejemplo del sistema de tres péndulos invertidos, en la Tabla 3 se puede observar el *CAI* de las tres tareas que controlan cada péndulo. La tarea T_1 tiene un *CAI* prácticamente nulo debido a que siempre es la tarea más prioritaria del sistema. Sin embargo, las tareas T_2 y T_3 sufren mayor variabilidad en el retardo, llegando la tarea T_3 a tener un *CAI* del 60%, lo cual causa la degradación en el comportamiento del péndulo 3. En la Figura 8 se puede ver la distribución del *CAI* para la tarea T_3 , donde se puede observar que, la mayoría de las veces la tarea sufre una variabilidad del 40% pero llega al 60% en muchas ocasiones.

	C	D	P	CAI(%)	Prioridad
T_1	7	20	20	0.005	1
T_2	7	29	29	24.14	2
T_3	7	35	35	60	3

Tabla 3. Parámetros y *CAI* de las tres tareas bajo planificación DM

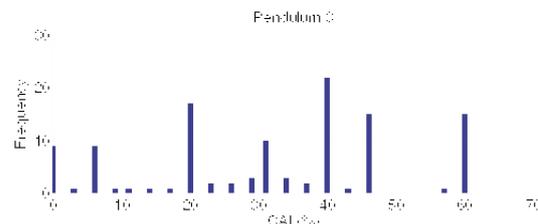


Figura 8. Distribución del *CAI* para la tarea T_3

Existen otros trabajos relativos al estudio e influencia de los retardos de control. Según Cervin (Cervin and Ecker, 2000) los parámetros temporales más importantes de una tarea de control son:

- *Periodo actual* P_{ik} : Es la distancia entre dos envíos consecutivos de la acción de control para la tarea T_i . Se puede expresar como la diferencia entre los tiempos de respuesta absolutos en la activación k y la activación $k + 1$.

$$P_{ik} = kP_i + rt_{ik+1} - (k-1)P_i - rt_{ik} = P + rt_{ik+1} - rt_{ik}$$

Se desea que sea constante, para que este periodo sea lo más parecido posible al periodo de muestreo para el que ha sido diseñado el regulador. Por tanto, será necesario que el tiempo de respuesta en todas las activaciones sea el mismo.

¹ Data Acquisition Interval

² Control Action Interval

- *Jitter* J_{ik} : Es el tiempo transcurrido entre el *release time* r_{ik} y el instante en el que se leen los sensores en la activación k para la tarea T_i . Se desea que sea nulo, o al menos constante para todas las activaciones.
- *Latencia* L_{ik} : También conocida como retardo de control o retardo computacional. La *latencia* se refiere a la variación entre el comienzo y la finalización de las activaciones de una tarea. Se puede expresar como:

$$L_{ik} = rt_{ik} - J_{ik}$$

Al igual que el *jitter*, se desea que sea nulo, y, si esto no es posible, que sea constante.

En la Figura 9 se representan estos parámetros. La *latencia* y el *jitter* son considerados como retardos, mientras que la diferencia entre los periodos actuales es consecuencia de estos retardos. Nótese que el *Periodo actual* depende de los tiempos de respuesta y estos, a su vez, dependerán de la *latencia* y el *jitter*.

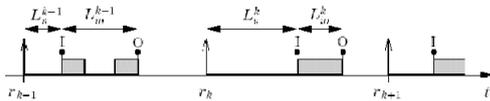


Figura 9. Representación de los retardos de una tarea

Los valores de latencia máximos y mínimos para el sistema de los tres péndulos bajo DM utilizados como ejemplo en este trabajo se muestran en la Tabla 4. En la Tabla se puede observar como la tarea T_1 presenta el mismo valor máximo y mínimo, por lo tanto su retardo variable es nulo, no siendo así para el resto de tareas, las cuales sufren interferencias de las tareas más prioritarias.

	L_{io}^{min}	L_{io}^{max}
T_1	7	7
T_2	7	14
T_3	7	28

Tabla 4. Valores de latencia para el ejemplo de los tres péndulos

4. TÉCNICAS PARA LA REDUCCIÓN DE LOS RETARDOS DE CONTROL

En este apartado se hace un breve repaso de los métodos que propone cada autor para minimizar los retardos definidos en el apartado anterior.

Según Árzén (Arzén, 2000) cuatro posibles soluciones pueden ser tomadas para compensar el retardo de control:

- Implementar el sistema de forma que se minimice el retardo de control. De este modo, puede ser ignorado a la hora de diseñar el regulador.

- Tratar de asegurar de que este retardo es constante. En este caso, puede tenerse en cuenta en el diseño del regulador, ya que existen técnicas de diseño de reguladores para sistemas con retardos fijos.
- Diseñar el regulador para que sea robusto frente al *jitter*. Esta solución es bastante más compleja que las dos anteriores.
- Diseñar el regulador para que, en ciertas situaciones y en cada periodo de muestreo sea capaz de compensar el retardo de control o parte de él.

Respecto a la segunda solución, van a presentarse dos trabajos que proponen un modelo de tareas para minimizar los retardos de control, en concreto los causados por el planificador de tiempo real. A su vez, se irán presentando los resultados de estos dos trabajos aplicados al sistema de tres péndulos invertidos que se ha utilizado en este artículo. El objetivo es comparar ambas técnicas sobre un sistema de control. El tercer apartado de este punto es una variación del modelo de particionado IMF con compensación del retardo ya que, como se comentará posteriormente, éste modelo introduce un retardo fijo considerable.

4.1 Particionado en dos tareas (modelo de Cervin)

En (Cervin and Ecker, 2000) se propone un método para minimizar la *latencia* dividiendo cada tarea de control en dos partes:

- Cálculo de la acción de control (T_{CO})³. Contiene solo la parte de cálculos que necesitan datos del actual intervalo de muestreo. También se encarga de enviar la acción de control.
- Actualización del estado (T_{US})⁴. Realiza la actualización del estado del controlador, así como la preparación para el siguiente instante de muestreo.

Ambas partes tienen el periodo de la tarea original T . Asignando un deadline menor a CO se consigue minimizar el tiempo de respuesta de esta subtarea, utilizando como política de planificación el RM. Es importante minimizar el retardo de la tarea T_{CO} , ya que es la encargada de enviar la acción de control. La asignación del deadline de la tarea T_{CO} se realiza con un método iterativo, comenzando por un valor inicial de $D_{CO} = T - C_{US}$ y asignando en cada iteración el tiempo de respuesta de T_{CO} tras planificar el sistema con RM. En la Figura 10 se representa este modelo de tareas.

Sin embargo, este método solo minimiza la *latencia* de la parte inicial pero no de la final, con lo que no se mejora la *latencia* total de la tarea sin

³ Calculate_Output

⁴ Update.State

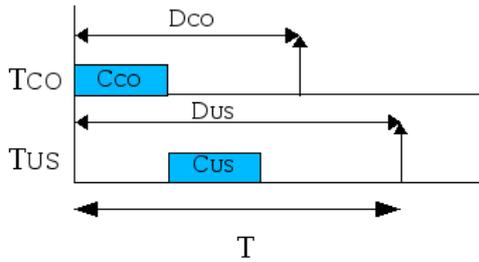


Figura 10. Modelo de particionado en dos tareas (Cervin)

particionar, sino que sólo se hace un poco más constante. En este trabajo se comenta la posibilidad de que, si no es viable reducir por completo la *latencia*, por lo menos mantenerla constante. De este modo, este retardo puede ser tenido en cuenta y, por tanto, compensado en el diseño del regulador. Esta misma idea es introducida en (Crespo *et al.*, 1999), con un modelo de particionado más completo, el cual se comenta en el siguiente punto.

El modelo de Cervin aplicado al sistema de tres péndulos, consiste en el particionado de las tres tareas que controlan cada péndulo en dos sub-tareas. Los parámetros de estas sub-tareas, así como los valores del CAI, se muestran en la Tabla 5. Además, en la Figura 11 se representa la distribución del CAI para la tarea T_3 , en el cual se observa que, aunque la mayoría de las veces el CAI es 0, hay ocasiones en las cuales llega hasta el 17.4%.

Como se puede observar, los valores de CAI se han reducido respecto del modelo de tareas sin particionar, por lo que es de esperar que la respuesta del sistema mejore. Esta respuesta, para el péndulo 3 se muestra en la Figura 12.

		C	D	P	O	Prioridad	CAI (%)
A_1	T_{CO}	3	3	20	0	1	0.005
	T_{US}	4	20	20	0	4	
A_2	T_{CO}	3	6	29	0	2	10.35
	T_{US}	4	29	29	0	5	
A_3	T_{CO}	3	9	35	0	3	17.14
	T_{US}	4	35	35	0	6	

Tabla 5. Parámetros del modelo de tareas de Cervin

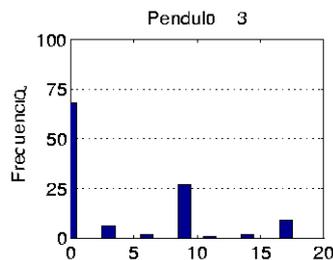


Figura 11. Distribución del CAI del péndulo 3 con el modelo de Cervin

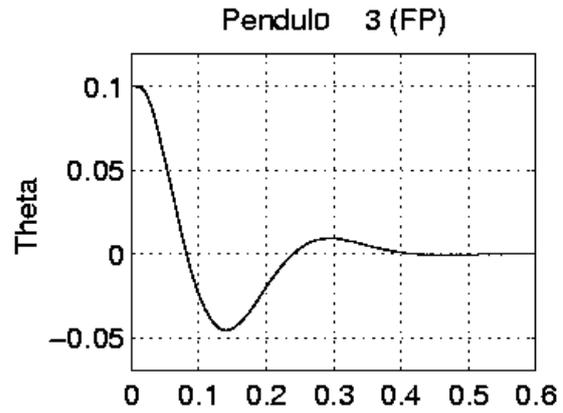


Figura 12. Respuesta del péndulo 3 con el modelo de Cervin

4.2 Particionado en tres tareas (modelo IMF)

El modelo de tareas propuesto en (Crespo *et al.*, 1999) y (Balbastre *et al.*, 2004) consta de n actividades de control $\mathcal{A} = (\mathcal{A}_1, \dots, \mathcal{A}_n)$. Una actividad de control está caracterizada como $\mathcal{A}_i = (C_i, D_i, P_i, O_i)$. Cada actividad de control \mathcal{A}_i da lugar a las siguientes tareas:

- T_{ii} Correspondiente a la adquisición de datos de la actividad \mathcal{A}_i . También se hará referencia a esta tarea como tarea inicial de \mathcal{A}_i .
- T_{im} Correspondiente al cálculo de la acción de control de la actividad \mathcal{A}_i . También se llamará tarea obligatoria⁵ de \mathcal{A}_i .
- T_{if} Correspondiente al envío de la acción de control dentro de la actividad \mathcal{A}_i . También se hará referencia a esta tarea como tarea final de \mathcal{A}_i .

No se han incluido otras tareas propias de una actividad de control como *Mejora de la solución*, *Evaluación del estado global* y *Actualización de la información global*. El motivo es que son partes de la actividad de control que no es necesario que se ejecuten en todas las activaciones, ya que son operaciones de mejora o mantenimiento del estado, no siendo críticas para el sistema.

Por tanto, el modelo de tareas completo consta de $3n$ tareas $\mathcal{T} = (T_{1i}, T_{1m}, T_{1f}, \dots, T_{ni}, T_{nm}, T_{nf})$ caracterizadas como:

$$T_{ii} = (C_{ii}, D_{ii}, P_{ii}, O_{ii})$$

$$T_{im} = (C_{im}, D_{im}, P_{im}, O_{im})$$

$$T_{if} = (C_{if}, D_{if}, P_{if}, O_{if})$$

Una relación entre este nuevo modelo propuesto y el modelo de tareas básico se presenta en la Figura 13.

⁵ El subíndice m viene de la palabra inglesa *mandatory* que significa obligatorio. El cálculo de la acción de control es una parte obligatoria dentro de la actividad de control.

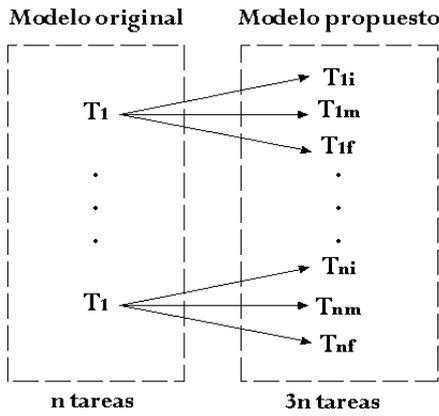


Figura 13. Relación entre el modelo de tareas original y el propuesto

Con este nuevo modelo de tareas, es necesario redefinir el CAI y el DAI. Ahora, la lectura de sensores la realizan las tareas iniciales para cada activación de control. Por lo tanto, el DAI se referirá a la variación en la activación de las tareas iniciales. No tiene sentido, por tanto, hablar de DAI para tareas finales u obligatorias. Del mismo modo, el CAI sólo será aplicable a tareas finales, ya que expresa la variación en el envío de la acción de control. Por lo tanto, para una actividad de control A_i los valores de CAI y DAI serán los siguientes:

$$DAI_i = \frac{WCAT_{ii} - BCAT_{ii}}{P_i} 100$$

$$CAI_i = \frac{WCRT_{if} - BCRT_{if}}{P_i} 100$$

Para minimizar tanto el CAI como el DAI, hay que asignar prioridades de forma adecuada. Para ello se consideran tres niveles de prioridad: en el mayor nivel de prioridad se sitúan las tareas finales. En el nivel intermedio la tareas iniciales y, por último, las tareas menos prioritarias son las obligatorias. Según esta asignación, las tareas finales se ejecutarían antes que las iniciales, por tener mayor prioridad que éstas. Para evitarlo, las tareas finales se ejecutan con un desplazamiento (*offset*) respecto del inicio de la activación de su actividad de control. La representación de este modelo se puede observar en la figura 14.

Aplicando el modelo IMF al proceso del ejemplo, se tienen las tareas con los parámetros y el CAI mostrado en la Tabla 6. En este caso, el CAI se ha reducido drásticamente. La distribución del mismo para la tarea T_3 se muestra en la Figura 15. Es más, la mayor parte de las activaciones la tarea T_3 presenta un CAI nulo, y en muy pocas ocasiones se presenta el máximo CAI del 1.45 %.

Con estos valores es de esperar que la respuesta del sistema mejore considerablemente. Ésta se muestra en la Figura 16 pero, contrariamente a lo esperado, la respuesta presenta mayor oscilación,

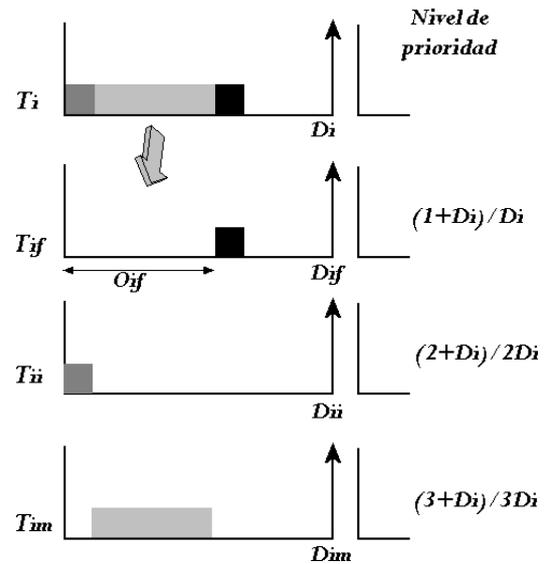


Figura 14. Modelo de tareas bajo planificación DM

		C	D	P	O	Prioridad	CAI
A_1	T_{1i}	0.5	20	20	0	4	0.005
	T_{1m}	6	20	20	0	7	
	T_{1f}	0.5	20	20	8.5	1	
A_2	T_{2i}	0.5	29	29	0	5	1.74
	T_{2m}	6	29	29	0	8	
	T_{2f}	0.5	29	29	14	2	
A_3	T_{3i}	0.5	35	35	0	6	1.45
	T_{3m}	6	35	35	0	9	
	T_{3f}	0.5	35	35	26.5	3	

Tabla 6. Parámetros del modelo IMF con planificación DM.

alcanzando la referencia mucho más tarde de lo deseado. El motivo es que, aunque se ha reducido en gran medida el CAI (retardo variable), ha aumentado considerablemente el retardo fijo (tiempo de cómputo más *offset* aplicado a la tarea T_{3f} . Este retardo fijo es el que causa la degradación de la respuesta del sistema, por lo que es conveniente aplicar la solución 4 comentada al inicio de este apartado, es decir, compensar el retardo fijo. Este es el objetivo del siguiente apartado.

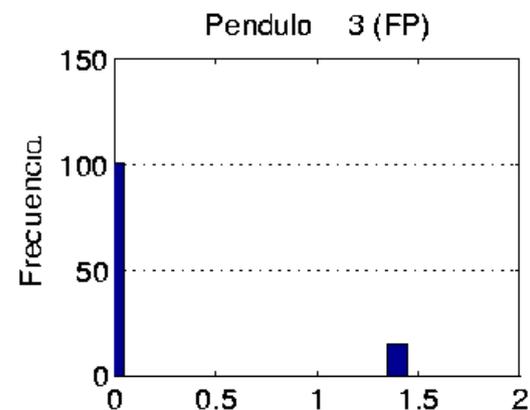


Figura 15. Distribución del CAI del péndulo 3 con el modelo IMF

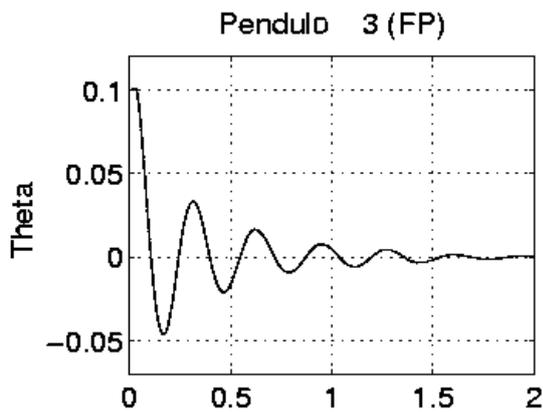


Figura 16. Respuesta del péndulo 3 con el modelo IMF

4.3 Particionado en tres tareas (modelo IMF) con compensación

Para mejorar la respuesta del modelo IMF, se ha diseñado un controlador LQG utilizando el comando de Matlab *lqgdesign*, el cual recibe como parámetro de entrada el retardo fijo de cada tarea. Estos retardos son los *offsets* para las tareas finales. La respuesta del péndulo 3, una vez se ha compensado el retardo fijo, y con el mismo CAI que en el modelo IMF, se muestra en la Figura 17. Como se puede observar, la respuesta del sistema mejora considerablemente, siendo muy similar a la respuesta ideal.

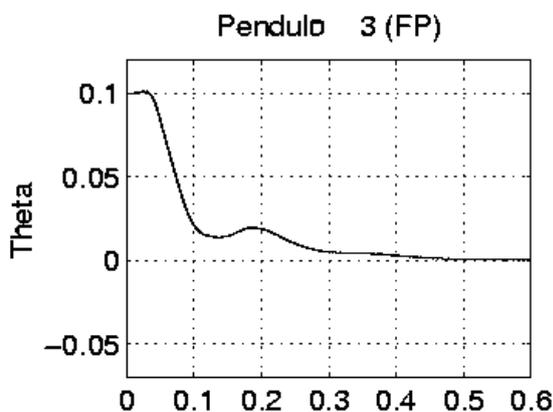


Figura 17. Respuesta del péndulo 3 con el modelo IMF con compensación

5. CONCLUSIONES Y TRABAJO FUTURO

Como se puede desprender de los resultados obtenidos en las distintas simulaciones realizadas bajo distintos modelos, los retardos introducidos por el planificador afectan drásticamente a las prestaciones finales del sistema a controlar. Esta influencia negativa es fácilmente visible comparando la respuesta del sistema en el caso ideal (respuesta esperada por los diseñadores de control) con la

respuesta obtenida cuando el sistema es planificado en un sistema de tiempo real. Por lo tanto, se ha comprobado que el codiseño entre la etapa de diseño del sistema de control y planificación en un sistema de tiempo real es necesaria.

De este modo, este artículo ha analizado y estudiado los trabajos que tratan de integrar dichas fases, desde el punto de vista de la reducción de los retardos de control. El objetivo principal de los dos trabajos comentados es reducir los retardos introducidos por el planificador basándose en el particionamiento de cada tarea de control en subtareas. Con esta división se consigue transformar los retardos variables en retardos fijos (que pueden ser tenidos en cuenta en el diseño del regulador). Principalmente se han analizado dos modelos de tareas de tiempo real, el modelo de Cervin y el modelo IMF.

De los resultados obtenidos se puede indicar que la respuesta del sistema del modelo de Cervin es muy cercana a la respuesta esperada por los diseñadores (caso ideal) aunque la acción de control no se envíe de forma sincrónica. Este comportamiento se debe a que la distancia entre la lectura de los datos y el envío de la acción de control es muy pequeña. Mientras que en el modelo IMF, aunque todas las tareas finales envían la acción de control de forma sincrónica, la respuesta del sistema no es buena y es muy diferente que la del caso ideal e incluso peor que la presentada en el modelo estándar (sin particionar). Básicamente esto es debido a que el retardo fijo introducido (equivalente al offset de la subtask final) es muy grande. Esto provoca que la distancia entre la lectura de los datos y el envío de la acción de control sea muy grande, de manera que, como la dinámica del péndulo cambia muy rápido la acción de control calculada no se corresponde con la situación actual del péndulo.

Del análisis anterior, se deduce que un retardo fijo muy grande en un sistema de control muy cambiante afecta a las prestaciones finales del sistema. Por lo tanto, este retardo fijo debe ser compensado. Para ello, se ha presentado una ampliación del modelo IMF con compensación del retardo fijo, obteniendo unos resultados muy buenos tanto para la variabilidad que presenta el envío de la acción de control (casi sincrónica para todas las tareas) como para la respuesta del sistema muy cercana al caso ideal.

Se podría concluir que el último modelo analizado presenta el mejor comportamiento respecto al modelo de Cervin y el IMF sin compensación, y evidentemente, que el modelo estándar (modelo sin particionar) obtenido una respuesta del sistema poco oscilante y con un error en el transitorio despreciable acercándola mucho a la obtenida por el sistema ideal.

Un punto de partida para seguir investigando en este área sería tratar de reducir el retardo fijo (offset) que se obtiene en el modelo IMF.

En la Figura 18 se muestra una nueva propuesta de particionamiento de las tareas donde la idea es retrasar la ejecución de la parte de la tarea obligatoria que se encarga de la actualización del estado y ejecutarla después de la tarea final, tal y como se hace en el modelo de Cervin. De esta forma, para el sistema de los tres péndulos invertidos, el modelo IMF quedaría con una tarea inicial con un tiempo de ejecución de 0.5 ms (adquisición de datos), una tarea obligatoria que será dividida a su vez en dos subtareas: una que sólo realizaría los cálculos necesarios para obtener la acción de control (tiempo de ejecución 2 ms) y que debe ejecutarse antes de la tarea final y otra que realizará la actualización del estado (tiempo de ejecución 4 ms) y que se ejecutará cuando finalice la subtaska final, y finalmente, una tarea final de 0.5 ms encargada de enviar la acción de control y que deberá disponer de un offset (retardo fijo) que será menor que el planteado en el modelo IMF inicial.

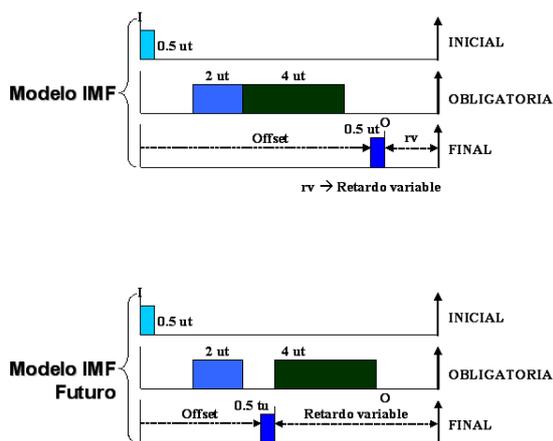


Figura 18. Modelo IMF con particionamiento de la tarea obligatoria

El objetivo es obtener un sistema con un CAI muy similar al modelo IMF, pero con un retardo fijo considerablemente menor, de forma, que no sea necesario compensar este retardo.

REFERENCIAS

- Albertos, P. and A. Crespo (1999). Real-time control of non-uniformly sampled systems. In: *Control Engineering Practice*. pp. 1–13.
- Arzén, Karl-Erik (2000). Sistemas de control en tiempo real: Control y planificación integrados. Technical report. Curso de especialización en automática. Almería. Cea-Ifac.
- Aström, K. and B. Wittenmark (1997). *Computer-Controlled Systems*. Prentice Hall.

- Balbastre, Patricia, Ismael Ripoll, Josep Vidal and Alfons Crespo (2004). A task model to reduce control delays. *The Journal of Real-time Systems* **27**(3), 215–236.
- Cervin, A. and B. Lincoln (2003). Jitterbug 1.1-reference manual. Technical report isrn lutfd2/tfirt-7604-se. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- Cervin, A. and J. Ecker (2000). Feedback scheduling of control tasks. In: *Proceedings of the 39th IEEE Conference on Decision and Control*.
- Crespo, A., I. Ripoll and P. Albertos (1999). Reducing delays in rt control: the control action interval. *IFAC World Congress Beijing*.
- Henriksson, D. and A. Cervin (2003). Truetime 1.13-reference manual. Technical report. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- Liu, C.L. and J.W. Layland. (1973). Scheduling algorithms for multiprogramming in a hard real-time environment. *JACM* **23**, 46–68.
- Mita, T. (1985). Optimal digital feedback control systems counting computation time of control laws. *IEEE Transaction on Aut. Control* **30**(6), 542–548.
- Torngren, M. (1995). Modeling and design of distributed real-time control applications. *Ph.D. thesis, Dept. of Machine Design, The Royal Institute of Technology, Stockholm, Sweden*.
- Torngren, M. (1998). Fundamentals of implementing real-time control applications in distributed computer systems. *The Journal of real-time systems* **14**(3), 3–34.