

## MUESTREO, CONTROL Y COMUNICACIÓN BASADOS EN EVENTOS

S. Dormido<sup>\*1</sup>, J. Sánchez\*, E. Kofman\*\*

*\*Dpto. Informática y Automática. UNED  
c/. Juan del Rosal 16, MADRID 28040, España  
EMAIL: [sdormido, jsanchez]@dia.uned.es*

*\*\*Laboratorio de Sistemas Dinámicos. FCEIA-UNR-CONICET  
c/. Riobamba 245 bis, ROSARIO 2000, Argentina  
EMAIL: kofman@fceia.unr.edu.ar*

Resumen: Los sistemas de control periódicos o activados por tiempo han dominado casi de manera exclusiva la investigación en ingeniería de control. El control basado en eventos es una alternativa muy prometedora particularmente cuando se consideran sistemas con capacidades reducidas de computación y de comunicación. En un sistema de control basado en eventos es la ocurrencia de un evento, en lugar del paso del tiempo lo que decide cuándo se debe efectuar el muestreo. En este trabajo se presenta una revisión panorámica de la situación actual del campo de los sistemas de muestreo basados en eventos. Se presentan los principales esquemas de muestreo basados en eventos y se analizan las diferentes estrategias de diseño de controladores que funcionan con este tipo de muestreo. Finalmente se analiza la implicación que tendrá su aplicación en el campo de los nuevos sistemas de control en red. *Copyright © 2008 CEA-IFAC*

Palabras Clave: Muestreo basado en eventos, Control basado en eventos, Control en red.

### 1. INTRODUCCIÓN

En el lenguaje coloquial un *evento* es simplemente algo que acontece y que provoca un cambio por el simple hecho de suceder. Esta idea tan intuitiva también es cierta en el contexto de los sistemas de control. En el estudio de los sistemas muestreados la actualización de la señal de control cada período de muestreo  $h$  supone la aparición de un *evento temporal* cuya frecuencia de ejecución está preestablecida de antemano. Hay sin embargo otra categoría de eventos que tienen también una gran importancia y que no pueden ser planificados a priori ya que su activación está relacionada con el instante de tiempo en que se cumple una determinada relación entre las variables de estado del sistema. A esta clase de eventos se les conoce como *eventos de estado*.

La mayor parte de la investigación que se lleva a cabo en control automático considera sistemas de control periódico o activados por tiempo donde las señales de tiempo continuo se representan por sus valores muestreados con período de muestreo  $h$  constante. A estos sistemas se les designa de forma genérica como *sistemas de control basados en tiempo*. Sin embargo cuando es la ocurrencia de un evento, en lugar del paso del tiempo, lo que decide cuándo se debe muestrear al sistema dinámico hablamos de un *sistema de control basado en eventos*.

En un sistema de control basado en tiempo es la progresión autónoma del tiempo lo que dispara la ejecución de las acciones mientras que en los

<sup>1</sup> Trabajo financiado por CICYT Contrato DPI 2007-61068

sistemas de control basados en eventos es la propia evolución dinámica del sistema la que decide cuándo se ejecutará la próxima acción de control. Para ilustrar la diferencia entre ambos tipos, consideremos el ejemplo de un repartidor que tiene que entregar unos paquetes a sus clientes. En la situación del muestreo basado en tiempo cada cliente dispone de un reloj para ir a comprobar cada 10 minutos si ha llegado un nuevo paquete a su puerta. Si no ha llegado puede continuar su trabajo. Sin embargo, en el caso del muestreo basado en eventos, el repartidor llama a la puerta del cliente específico a quien tiene que entregar el paquete quien abre la puerta y lo recoge. Los otros clientes pueden continuar realizando sus tareas sin necesidad de ser interrumpidos. Este ejemplo ilustra claramente uno de los posibles beneficios del control basado en eventos que es una reducción de la carga de trabajo ya que los clientes no tienen que abrir innecesariamente sus puertas. En una aplicación de control esto se traduce en una reducción, de por ejemplo, la carga del bus de comunicación y del uso del procesador. Desde un punto de vista de comportamiento del control, la ventaja real del control basado en eventos estriba en que se reducen los tiempos de respuesta. Cuando se entrega un paquete se avisa al cliente que puede abrir la puerta inmediatamente. En la situación activada por tiempos pueden transcurrir hasta 10 minutos después de la entrega para que el cliente realice una acción.

En la figura 1 se observa la diferencia entre el muestreo basado en tiempo y en eventos en la adquisición de una señal. En el primer caso la señal se muestrea con un período  $h$ , mientras que en el segundo lo hace cuando cruza por valores múltiples de un umbral  $\delta$ .

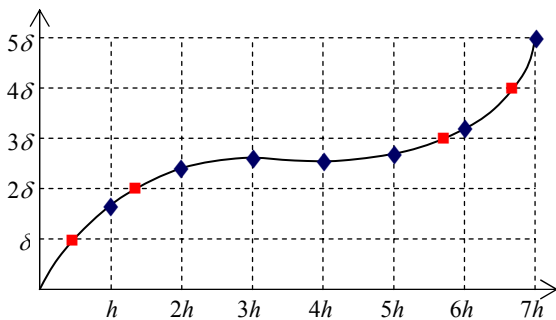


Figura 1. Muestreo basado en tiempo  $\blacklozenge$  ó eventos  $\blacksquare$

La razón fundamental para la predominancia de los sistemas de control basados en tiempo ha estado fundamentada en la existencia de una teoría bien establecida y madura para los sistemas de control muestreados (Åström and Wittenmark, 1997). Tal como se muestra en la figura 2a, las acciones que ocurren a esta frecuencia fija ( $1/h$ ) son *síncronas* en el tiempo.

Los actuales sistemas de control distribuidos también imponen restricciones de arquitectura al sistema que

dificultan la adopción del paradigma de eventos activados por tiempo. Esto es especialmente el caso de los lazos de control que se cierran utilizando redes de computadores o buses, tal como ocurre con los buses de campo, las redes de área local, las redes ATM o incluso la propia Internet (Nilsson, 1998). La comunidad de control asume que las plataformas de tiempo real utilizadas para implementar los controladores son capaces de garantizar estos intervalos de muestreo deterministas. En realidad esto se logra en los sistemas de control en red (Kawka and Alleyne, 2005; Ploplys *et al.*, 2004; Zhang *et al.*, 2001) en muy raras ocasiones a causa de los retardos de cálculo y/o comunicación que son inevitables y que son al mismo tiempo difíciles de predecir debido de una parte a la presencia de las memorias caches y de las instrucciones segmentadas que hay en las plataformas hardware modernas y de otra a la interferencia de las diferentes tareas que compiten en su ejecución por los recursos limitados de los procesadores. Esto provoca acciones de naturaleza *semisíncrona* tal como se observa en la figura 2b. Por *semisíncrona* se quiere expresar que las acciones ocurren con una cierta velocidad media pero los intervalos de muestreo varían dentro de ciertos límites.

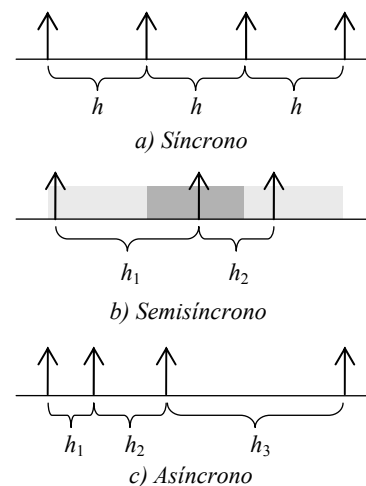


Figura 2. Clasificación de la distribución de las acciones en función del instante de su activación

Los retardos introducidos por el computador pueden degradar significativamente el comportamiento del control (Berge *et al.*, 2006). Para estudiar estos efectos en los lazos de control se pueden utilizar las herramientas Jitterbug y Truetime (Cervin *et al.*, 2003). Estas herramientas se basan en realizar un análisis estocástico de las señales y simulaciones extensivas del comportamiento del sistema. Otros trabajos adaptan la teoría de los sistemas muestreados de naturaleza periódica para incorporar la presencia del retardo en el diseño del control (Balluchi *et al.*, 2005; Cloosterman *et al.*, 2006; Kao and Lincoln, 2004; Lincoln, 2002; Zhang *et al.*, 2001). La variación en el “disparo del evento” se considera una perturbación y el objetivo es diseñar un compensador que sea robusto frente a dichos cambios.

Una alternativa a estos enfoques consiste en utilizar controladores basados en eventos que no estén restringidos a la ocurrencia sincrónica de las acciones del controlador. El empleo de un período de muestreo sincrónico es una de las condiciones más severas que imponen los ingenieros de control en la implementación del software. Es posible relajar estos requerimientos con los controladores basados en eventos. En este caso las acciones pueden ocurrir de forma *asíncrona* tal como se representa en la figura 2c. Existen muchas situaciones prácticas donde resulta interesante y ventajoso considerar sistemas de control basados en eventos en lugar del tradicional sistema de control basado en tiempo. El mecanismo de activación por eventos puede variar según los casos (Árzén, 1999). Algunos ejemplos significativos son los siguientes:

- El control de los motores de combustión interna es un ejemplo donde aparecen de forma natural intervalos de muestreo variables con el tiempo ya que se muestrea con respecto a la velocidad de la máquina (Alberoni *et al.*, 2005). Otro caso de este tipo es un sistema de fabricación donde el muestreo está relacionado con la tasa de producción.
- La naturaleza del muestreo basado en eventos puede ser intrínseca al método de medida utilizado o a la naturaleza física del proceso que se está controlando. Por ejemplo, cuando se usan sensores codificadores para medir la posición angular de un motor (Heemels *et al.*, 1999). Otros sensores activados por eventos incluyen sensores de nivel para medir la altura de un fluido en un tanque, unidades de disco magneto/ópticos con dispositivos de medida tipo codificador (Phillips and Tomizuka, 1995) y sistemas de transporte donde la posición longitudinal de un vehículo se conoce solo cuando se pasan determinadas marcas (de Bruin and van den Bosch, 1998). La cuantificación de señales en la cual el muestreo se induce por el cruce de los distintos niveles de cuantificación tienen un efecto análogo (Kofman and Braslavsky, 2006).
- Los sistemas de control que incorporan relés son otro ejemplo que puede ser considerado como un caso especial de muestreo basado en eventos (Tsytkin, 1984; Åström, 1995).
- El muestreo basado en eventos también puede ser una característica incorporada en un dispositivo sensor inteligente. En muchos casos resulta natural como por ejemplo cuando se usan como sensores codificadores o cuando los actuadores son de naturaleza de tipo *on-off*, tal como sucede en el control de satélite con impulsores (Dodds, 1981), o en sistemas con modulación de pulsos de frecuencia (Sira-Ramírez, 1989).
- El muestreo basado en eventos también se usa en la industria de procesos cuando se emplea en lazo cerrado el concepto de control de proceso estadístico (SPC). Para no perturbar el proceso, una nueva acción de control se calcula solo cuando ocurre una desviación estadísticamente significativa.
- Los moduladores  $\Delta$ - $\Sigma$  o los convertidores A/D de 1 bit que se utilizan normalmente en los sistemas de telefonía móvil y en audio son también casos especiales de muestreo basado en eventos (Norsworthy *et al.*, 1996). Es interesante observar que, a pesar de su uso extendido, no hay una buena teoría desarrollada para el diseño de sistemas con moduladores  $\Delta$ - $\Sigma$ .

Además de las diferentes fuentes naturales de disparo de eventos y su relevancia en la práctica, existen muchas otras razones de por qué la utilización de un control basado en eventos presenta interés:

- El control basado en eventos está mucho más cercano a la forma en que los seres humanos actuamos como un controlador. En realidad, cuando un operario efectúa control manual su conducta está guiada por eventos en lugar de por el tiempo. Ninguna acción de control se toma hasta que la señal de medida se ha desviado suficientemente del punto de consigna. El sistema de control del movimiento humano está basado en eventos o en pulsos en lugar de activarse por tiempo (Mead, 1989).
- Otra razón importante de por qué un sistema de control basado en eventos resulta de interés viene de la mano de la utilización de los recursos de cálculo. Un controlador empotrado se implementa típicamente sobre un sistema operativo en tiempo real. El tiempo de CPU disponible se comparte entre las tareas de manera tal que parece como si cada una estuviese ejecutándose de forma independiente. Tener ocupada a la CPU para que realice cálculos de control cuando nada significativo ha ocurrido en el proceso es claramente un uso innecesario de los recursos disponibles.
- El mismo argumento se aplica también a los sistemas de comunicación. El ancho de banda de comunicación que está disponible en un sistema distribuido es limitado. Utilizarlo para enviar datos mediante un esquema de control activado por tiempo se traduce en una pérdida de ancho de banda. Si se reduce el número de actualizaciones de la señal de control que se envían a través de la red, esto lleva aparejado directamente una reducción en el número de mensajes que se transmite y por consiguiente una disminución de la carga media del bus.

- Otro ejemplo es una red de sensores inalámbricos (Zhao and Guibas, 2004), donde cada uno de los nodos sensores está alimentado por una batería. Las experiencias realizadas (Feency and Nilsson, 2001) demuestran que comparativamente las transmisiones inalámbricas consumen significativamente más energía que la requerida por los propios cálculos internos y es por lo tanto un factor limitante de su autonomía (El Gamal *et al.*, 2002; Stark *et al.*, 2002). Así pues para reducir el consumo, sería deseable un método de muestreo basado en eventos que requiriese menos transmisiones de datos.
- También desde un punto de vista tecnológico los controladores basados en eventos se están haciendo más viables, particularmente en la adquisición de las variables medidas y la actualización de la señal de control en sistemas de tiempo real de naturaleza distribuida. Por ejemplo, TinyOS es un sistema operativo conducido por eventos que ya se está utilizando en algunas aplicaciones de redes de sensores distribuidos (Levis *et al.*, 2004).

Aunque el análisis anterior indica que en muchas situaciones es lógico estudiar e implementar controladores basados en eventos, sus aplicaciones son todavía escasas tanto en el mundo académico como industrial, al menos al nivel de los lazos de control por realimentación. Estos lazos se encuentran localizados en el nivel inferior de la jerarquía de un sistema de control y en general su conducta está dominada por dinámicas rápidas. En el nivel supervisor de la jerarquía de control si resulta más común emplear controladores activados por eventos que se adaptan muy bien (después de eliminar la dinámica continua) a la teoría de sistemas de eventos discretos (Cassandras and Lafortune, 1999). El motivo de por qué en los lazos de realimentación del nivel inferior todavía predomina el control activado por tiempos es la dificultad que entraña desarrollar una teoría de sistemas que se adapte a esta clase de sistemas activados por eventos en los cuales las dinámicas continuas no se pueden despreciar. Los tradicionales controladores basados en tiempo se diseñan con el objetivo de garantizar el comportamiento deseado del proceso controlado. La finalidad del control basado en eventos es obtener un mejor balance entre este comportamiento del control y otros aspectos del sistema (tales como la carga del procesador y del sistema de comunicación y el coste del sistema).

Tradicionalmente los sistemas con muestreo basado en eventos fueron implementados como sistemas analógicos. Hoy día es común realizarlos como sistemas discretos con un muestreo rápido. Este planteamiento, sin embargo, no hace sino esconder el problema de la detección del evento con el fin de usar todavía la teoría de los sistemas muestreados. Lo natural sería reconocer la naturaleza del problema y atacarlo directamente. Aparte de su interés intrínseco

los sistemas con muestreo basado en eventos pueden ser también una alternativa para tratar sistemas que utilizan muestreo multifrecuencia (Araki and Yamamoto, 1986). Un ejemplo de este enmascaramiento se produce con un tipo particular de sensores basados en eventos donde los datos de medida llegan no equidistantes en el tiempo. Para esta clase de sensores muy a menudo se diseñan observadores asíncronos que proporcionan estimas de las variables de estado de la planta a período fijo. En esta línea se han utilizado enfoques basados en el filtrado de Kalman (de Bruin and van den Bosch, 1998; Glad and Ljung, 1984; Krucinski *et al.*, 1998) o un observador tipo Luenberger (Phillips and Tomizuka, 1995). De esta forma como las estimas del estado si que están ahora disponibles a una velocidad de muestreo constante se pueden aplicar los métodos clásicos de análisis y diseño de sistemas de control (p. ej. realimentación del estado).

En algunos trabajos se consideran periodos de muestreo variables con el tiempo, aunque solo se permiten un número finito de posibles instantes de muestreo, y se diseñan controladores y observadores que emplean ganancias de realimentación y del observador que dependen del tiempo de muestreo conocido (Schinkel *et al.*, 2002 y 2003). La estabilidad del sistema en lazo cerrado se garantiza a través de la existencia de una función de Lyapunov de tipo cuadrático. En determinados casos que incluyen al control basado en eventos, sin embargo, conocer los tiempos de muestreo futuros es poco realista. Más aún, en el contexto de los sistemas de control basados en eventos se puede asegurar que no hay una función de Lyapunov ya que no se puede lograr estabilidad asintótica. Lo más que se puede conseguir es una acotación final (Blanchini, 1994 y 1999) con pequeñas cotas (un tipo de estabilidad práctica).

En la Tabla 1 se muestran esquemáticamente algunas de las características del muestreo basado en eventos en comparación con el muestreo basado en tiempos.

Tabla 1 Características de los sistemas muestreados basados en el tiempo y en eventos

<i>Basado en el tiempo</i>	<i>Basado en eventos</i>
$\{y(kh)\}$	$\{t_i   y(t_i) = n\delta\}$
Inyectividad si $h < 1/f_N$	Actuadores adaptados
LTI $\Rightarrow$ Sist. periódicos	Sensores inteligentes
Teoría madura	Modulador $\Delta$ - $\Sigma$ , IPFM
$\exists$ métodos de diseño	Supervisión de procesos
Implementación segura	Procesos con buffers
Formación orientada	Sistemas biológicos

En este trabajo se presenta una visión panorámica de los sistemas de control basados en eventos que es un tema que está teniendo un interés creciente en los últimos años dentro de la comunidad de control. En primer lugar se analizan los compromisos que plantean la implementación de los sistemas de control

basados en eventos desde el punto de vista del software. Este es un aspecto que requiere una mayor atención por parte de la comunidad de control y un mejor conocimiento de las exigencias que demandan el desarrollo de los sistemas de tiempo real que con mucha frecuencia no se tienen en cuenta. A continuación se hace un análisis de los tipos de muestreo basados en eventos que se han considerado hasta la fecha y para los cuales empiezan ya a aparecer en el mercado dispositivos sensores y actuadores adaptados a este enfoque. Se revisan las diferentes alternativas de diseño de controladores basados en eventos y finalmente el papel cada vez más importante que tiene en el control de sistemas en red. Uno de los mecanismos más importantes para evitar la congestión de una red de comunicación es transmitir información que sea verdaderamente significativa para el objetivo que se persigue. En este sentido los sistemas de control basados en eventos representan el paradigma natural para la formulación de estos sistemas y están llamados a jugar un papel cada vez más importante en el futuro inmediato.

## 2. COMPROMISOS SOFTWARE-CONTROL

¿Cuál es entonces el motivo de por qué todavía domina la escena el control activado por tiempo? En primer lugar la dificultad asociada con el desarrollo de una teoría formal de sistemas de control basados en eventos. Otra razón, que no se toma con la consideración que se merece, es que los sistemas de control basado en eventos necesitan colaboración y sinergia entre las comunidades de control y de tiempo real para un desarrollo eficaz del campo.

El ingeniero de control asume típicamente que las plataformas de tiempo real utilizadas para implementar controladores son capaces de garantizar intervalos de muestreo deterministas. En realidad, esto se logra en muy raras ocasiones. El mercado presiona en la dirección de utilizar sistemas comerciales, como por ejemplo Windows NT, para aplicaciones de tiempo real lo que significa que se disminuye el determinismo, se degrada el comportamiento del peor-caso, aunque se mejora drásticamente el rendimiento del tiempo promedio.

Por otra parte, el especialista de tiempo real supone que todos los lazos de control son periódicos, con períodos fijos, que presentan restricciones duras. Para esta clase de sistemas se ha establecido la teoría de planificación con velocidad monotónica y sus extensiones (Buttazzo, 1997). Usando esta teoría es posible decidir formalmente si una tarea dada cumplirá con sus restricciones de cálculo. Sin embargo, el punto débil de la teoría es la necesidad de tener una cota superior del peor tiempo de ejecución (WCET). En la práctica esto es muy difícil de obtener. De los ejemplos anteriores es obvio que hay algunos lazos de control que no son periódicos. También es fácil encontrar ejemplos de controladores híbridos o heterogéneos que conmutan entre un

conjunto de algoritmos de control cada uno, posiblemente con un intervalo de muestreo diferente. Algunos lazos de control tienen restricciones temporales duras, por ejemplo, controladores que controlan procesos inestables en lazo abierto. Hay un conjunto de aplicaciones donde las restricciones temporales no son tan estrictas. Con tal de que las restricciones no se violen a menudo o bajo ciertas condiciones operativas, se puede diseñar el controlador para que sea robusto frente a esta situación o bien para compensarlo.

Para poder garantizar un cierto comportamiento global del sistema, el experto en control debe ser capaz de reconocer las dificultades que surgen en la implementación de los algoritmos de control con requisitos estrictos de tiempo real en un entorno de software donde recursos tales como capacidad de procesamiento y ancho de banda de comunicación se comparten con otros procesos en ejecución. Es bastante usual que los especialistas de control verifiquen sus algoritmos sobre un hardware dedicado con suficiente potencia como para que cumpla todos los requerimientos. Sin embargo, no son conocedores de las técnicas de planificación que se emplean para implementar el algoritmo sobre el producto final. Esto significa que no toman en cuenta las necesidades del planificador durante el diseño ni hacen uso de sus propiedades (Sanz and Arzén, 2003).

Desgraciadamente también lo inverso suele ser cierto ya que muchos diseñadores de software de control tienen un conocimiento insuficiente de las exigencias que se pueden relajar en un controlador para que se adapte mejor a la arquitectura empleada. Un ejemplo de esto se puede encontrar en la selección de la frecuencia de muestreo. Desde un punto de vista de comportamiento de control, a muchos algoritmos les afecta poco una pequeña variación en la frecuencia de muestreo, pero este hecho podría facilitar mucho la implementación del software. También conocer los retardos que puede manejar el controlador es una información que el ingeniero de control no suele comunicar a los desarrolladores del software. Resulta pues evidente que una comunicación efectiva es esencial ya que es muy difícil que una única persona tenga un conocimiento amplio de ambas disciplinas. Adicionalmente se deberían disponer de métodos de diseño de sistemas de control que específicamente incorporasen en su propia formulación estos aspectos multidisciplinarios.

No obstante es conveniente considerar que existen algunos problemas que dificultan la implementación de los controladores en sistemas empotrados con recursos limitados tanto desde la perspectiva del ingeniero de control como del diseñador del software. Estos temas plantean una serie de compromisos importantes que afectan tanto al comportamiento del control (seguimiento de la referencia, estabilización, rechazo de perturbaciones, etc) como del software (utilización del procesador, tiempo de respuesta, etc)

que deben ser considerados conjuntamente en la fase de diseño del sistema. Los principales compromisos a tener en cuenta son:

1. *Selección de la frecuencia de muestreo.* La frecuencia de muestreo de las actualizaciones de la señal de control tiene una fuerte relación sobre el ancho de banda que se necesita en el controlador (la frecuencia hasta la cual se pueden seguir las señales de referencia y suprimirse las perturbaciones) (Franklin *et al.*, 2005). Cuando se aumenta la frecuencia de muestreo, también lo hace el ancho de banda que se puede obtener mediante una sintonía cuidadosa del controlador. Sin embargo, muy a menudo la frecuencia de muestreo es considerablemente mayor que la estrictamente necesaria. Típicamente, la selección de la frecuencia de muestreo se basa en reglas heurísticas (Åström and Wittenmark, 1997); por ejemplo, cuando se diseñan controladores de tiempo discreto puede ser de 6 veces el ancho de banda en lazo cerrado deseado. En la práctica industrial la frecuencia de muestreo se llega incluso a escoger como 40 veces el ancho de banda deseado en lazo cerrado para situarnos en el “lado seguro”.

La razón para reducir la frecuencia de muestreo es obvia en aplicaciones donde se tiene limitada la capacidad de procesamiento. Cada vez que se ejecuta el algoritmo de control, se necesita que se procesen un conjunto de tareas que tardan un cierto tiempo. Cuanto más complejo es un algoritmo, menor es generalmente la frecuencia máxima de muestreo que está permitida, ya que se emplea más tiempo en ejecutar dichos algoritmos. Por lo tanto, el compromiso principal que se puede identificar en la selección de la frecuencia de muestreo del controlador está entre el comportamiento de control de una parte y la carga del procesador de otro. Los esfuerzos se centran en relajar los requisitos de la frecuencia de muestreo con el diseño del controlador en tiempo discreto teniendo en cuenta la conducta de tiempo continuo del sistema controlado (Lindgärde and Lennartson, 1998).

Junto a la potencia de procesamiento disponible para el algoritmo de control, un factor limitante para la selección de la frecuencia de muestreo es el número reducido de temporizadores de interrupción en las plataformas de cálculo para disparar las actualizaciones de la señal de control. Esto suele ser el caso cuando hay más tareas que se ejecutan concurrentemente sobre el mismo procesador. En este caso puede resultar ventajoso tener tareas periódicas que se ejecutan con frecuencias de muestreo diferentes que son múltiplos unas de otras. Esto da una cierta indicación de las demandas que los diseñadores de software podrían imponer sobre la frecuencia de muestreo y que en la práctica habitual no se

toman en consideración cuando se diseña un controlador.

2. *Retardos variables con el tiempo.* Los retardos en la implementación de un sistema de control son inevitables a causa de que la ejecución de las tareas en un procesador simplemente llevan su tiempo. Cuando la capacidad de procesamiento es pequeña hay que analizar si la complejidad del algoritmo de control seleccionado es la adecuada. En muchas situaciones, los algoritmos de control que son complejos tienen la ventaja de ser capaces de efectuar tareas más exigentes que dan un mejor comportamiento de control. Sin embargo, cuando dicha complejidad aumenta, también lo hace el tiempo de ejecución, con lo cual también aumenta el período de actualización de la señal de control. Este incremento en el retardo suele influir de forma negativa sobre el rendimiento global del sistema. En la práctica, rara vez se toma en cuenta este compromiso. En la comunidad de software en tiempo real se refieren a los retardos fijos como *latencias* y el término *jitter* se utiliza para la parte variable del retardo.

Otras razones importantes para la variación de los retardos originados por la implementación software son: la comunicación de datos en una red (Zhang *et al.*, 2001), el análisis de cómo afectan los diferentes planificadores y la gestión de la memoria. Todas estas cuestiones contribuyen al comportamiento del control y del sistema en su globalidad y hacen que en último término la implementación del controlador sea un problema difícil de naturaleza multidisciplinar. Hay en estos momentos un interés renovado en estudiar cómo afectan estos retardos al control del sistema (Normey and Camacho, 2006). Este tema es particularmente importante cuantas más y más redes inalámbricas con ancho de banda limitado se utilizan para comunicar a distancia señales provenientes de los sensores y actuadores.

3. *Cuantificación.* La cuantificación se origina debido a la resolución limitada de los sensores, pero también la implementación del software y los mecanismos de comunicación pueden ser razones importantes. Cuando se realiza un controlador sobre una plataforma específica hay que tratar con una resolución limitada para la representación de las variables. Este efecto tiene su origen en la longitud finita de palabra de los computadores. Dependiendo del procesador, los cálculos requerirán más tiempo para longitudes de palabras más grandes. Por lo tanto, aumentar la longitud de palabra es una ventaja para el comportamiento del control ya que se reduce la cuantificación pero podría también traducirse en una desventaja porque se incrementa el tiempo de cálculo. Para comunicar datos en una red con capacidad limitada, también resulta válido el mismo razonamiento.

En teoría de control, la cuantificación es un efecto que se ha considerado desde los años 50, estudiándose algunos métodos de cómo afecta su inclusión al comportamiento del sistema (Franklin *et al.*, 2005). Sin embargo, desde un punto de vista práctico los cuantificadores se suelen analizar en simulación para observar su efecto sobre la conducta del sistema. Este es un aspecto importante, ya que la cuantificación provoca errores de seguimiento y puede introducir conductas adversas, tales como la aparición de ciclos límites en el sistema controlado. Desgraciadamente es un tema que con frecuencia no se toma en consideración en las fases iniciales del diseño (Ishii and Francis, 2002).

Se han mencionado diferentes compromisos entre comportamiento del control y el diseño del software en el contexto de la selección de la frecuencia de muestreo, los retardos y la cuantificación. Estos son, sin embargo, un pequeño subconjunto de las cuestiones que tienen que considerarse en el diseño e implementación de sistemas empotrados de control en tiempo real. Aún más, para los sistemas actuales se requiere considerar diferentes disciplinas en el proceso de garantizar estos múltiples compromisos. Todas ellas son responsables de aspectos específicos del diseño electrónico, mecánico y software, mientras que los fundamentos físicos, químicos y biológicos constituyen también restricciones de un nivel más básico. Los diseños se llevan a cabo a menudo en paralelo por grupos de personas con un perfil multidisciplinar. Considerando la complejidad de estos sistemas debido al tamaño que tienen (se puede pensar típicamente en millones de líneas de código y miles de componentes), producir una solución técnicamente correcta no es una tarea sencilla y constituye un reto de ingeniería no trivial. Como en la práctica resulta imposible analizar completamente cada compromiso debido a un tiempo y esfuerzo de diseño limitado, hay una necesidad de concentrarse sobre las cuestiones más cruciales y no perder tiempo en las otras. Por lo tanto, deberían de proveerse métodos que soporten la focalización de los temas más importantes y críticos en un diseño.

### 3. MUESTREO BASADO EN EVENTOS

El muestreo basado en eventos no es en realidad una idea nueva y sus orígenes se remontan a finales de los años 50 cuando (Ellis, 1959) planteaba que el método de muestreo más apropiado consiste en transmitir solamente datos cuando existe un cambio significativo en la señal que justifique la adquisición de una nueva muestra. El muestreo basado en eventos viene caracterizado así por el hecho de que existe una relación funcional entre los instantes de muestreo y la conducta de la señal y en este sentido se dice que es un *muestreo dependiente de la señal* (Miskowicz, 2007). Durante las décadas de los 60 y 70 en el dominio del control se propuso un esquema de

muestreo no convencional que se denominó *muestreo adaptativo*. Fue introducido en un trabajo por (Dorf *et al.*, 1962) que justificaba el nombre porque el sistema de control en lazo cerrado se hacía inherentemente adaptativo frente a variaciones en sus parámetros ya que en ese caso la frecuencia de muestreo variable compensaría dichos cambios mediante una modificación en la velocidad de muestreo de la señal de error. El muestreo adaptativo es pues en esencia un muestreo dependiente de señal, de naturaleza heurística, donde se permite que el período de muestreo cambie en tiempo real de intervalo a intervalo. El objetivo que persigue es reducir el número de muestras, sin que se produzca una degradación en la respuesta del sistema respecto de la que se obtendría con un sistema muestreado equivalente con periodo constante. En la figura 3 se muestra el diagrama de bloques de un sistema de este tipo, donde ZOH es un retenedor de orden cero.

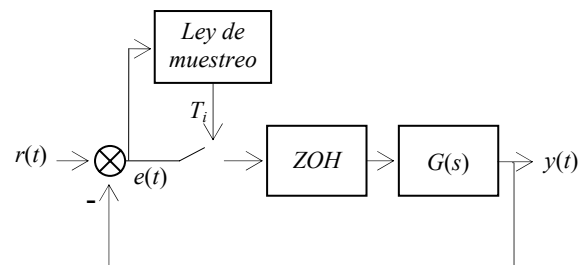


Figura 3. Sistema de control con muestreo adaptativo

Cabe preguntarse qué es lo que diferencia a un sistema de muestreo basado en eventos de un sistema de muestreo adaptativo. En el primer caso la adquisición de una nueva muestra viene definida por una *relación implícita* entre variables del sistema que cuando se cumple se activa un evento que marca el instante en que se produce el muestreo. Por el contrario en el segundo lo que se proporciona es una *relación explícita* que define el nuevo periodo de muestreo. En su origen todas las leyes de muestreo adaptativo se formularon de forma implícita. Es siempre posible, mediante una aproximación conveniente, transformarlas en una forma explícita. En la figura 4 se muestra una ley genérica de muestreo adaptativo de tipo implícito (a través de la función  $\phi$ )

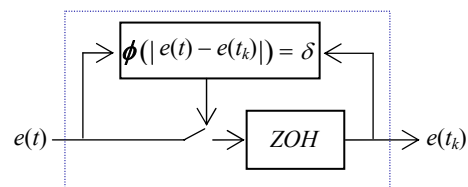


Figura 4. Ley implícita de muestreo adaptativo

Para poner de manifiesto la forma de obtener las leyes de muestreo considérense los dos casos particulares siguientes que han recibido una considerable atención en los trabajos sobre muestreo basado en eventos ya que son los esquemas más naturales en los que cabría inicialmente pensar.

1.  $\int_{t_k}^{t_{k+1}} |e(t) - e(t_k)| dt = \delta$ . Este ley de muestreo fue originalmente propuesta por (Dorf *et al.*, 1962). Si se aproxima el error  $e(t)$  por un desarrollo en serie de Taylor truncado en el primer término

$$e(t) \cong e(t_k) + \dot{e}(t_k)(t - t_k)$$

y se sustituye en la ley de muestreo se obtiene la siguiente expresión para el periodo  $T_k$ :

$$T_k = \frac{\sqrt{2\delta}}{\sqrt{|\dot{e}(t_k)|}}$$

2.  $|e(t) - e(t_k)| = \delta$ . Siguiendo un procedimiento análogo de aproximación se obtiene:

$$T_k = \frac{\delta}{|\dot{e}(t_k)|}$$

Esta ley se la denominó *criterio de diferencias de amplitudes constante* (Mitchell and McDaniel, 1969; Dormido *et al.*, 1973, 1977)

Si el valor de  $\dot{e}(t_k) = 0$ , entonces  $T_k = \infty$ . Para evitar esta singularidad se fija un intervalo de muestreo máximo  $T_{max}$  que se escoge tomando en cuenta requisitos de control tales como estabilidad, ancho de banda, etc. (Gupta, 1963; Bekey and Tomovic, 1966; Tomovic and Bekey, 1966; Ciscato and Mariani, 1967; Tait, 1966; Smith, 1971; Hsia, 1972b, 1974) estudian nuevos criterios de muestreo adaptativo formulados todos ellos de forma heurística. Un estudio unificado de leyes de muestreo adaptativo fue hecho por (Hsia, 1972a; Dormido *et al.*, 1978; de la Sen, 1996). Su interés es que proporcionan una base analítica a todas las leyes que habían sido propuestas.

### 3.1 Muestreo por cruce de nivel.

En la figura 5 se muestra una señal muestreada por cruce de nivel. En este caso, los niveles se encuentran equiespaciados y además hay histéresis (los cruces sucesivos por un mismo nivel no provocan muestras).

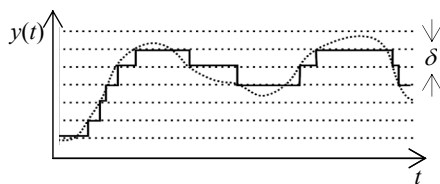


Figura 5. Muestreo por cruce de nivel

El uso de este tipo de estrategias de muestreo ha recibido diferentes nombres en la literatura: criterio de diferencia de amplitudes constante (Mitchell and

McDaniel, 1969; Dormido *et al.*, 1973, 1977), muestreo basado en magnitud (Miskowicz 2003), modulación delta asincrónica (Inose *et al.*, 1966), método de banda muerta (Otanéz *et al.*, 2002), método *send-on-delta* (Miśkowicz 2006), muestreo por cruce de nivel (Sayiner *et al.*, 1996) y muestreo de Lebesgue (Åström and Bernhardsson 2002).

A pesar de sus muchos nombres, el principio básico es el mismo: la señal se muestrea cuando el valor absoluto de la diferencia entre el valor actual  $x(t)$  y el último valor muestreado  $x_k(t)$  es mayor que un límite especificado  $\delta$  (figura 6b). Cuando el cambio en la señal es relativamente pequeño, el número de muestreos es significativamente menor que aquel del método del muestreo periódico (figura 6a).

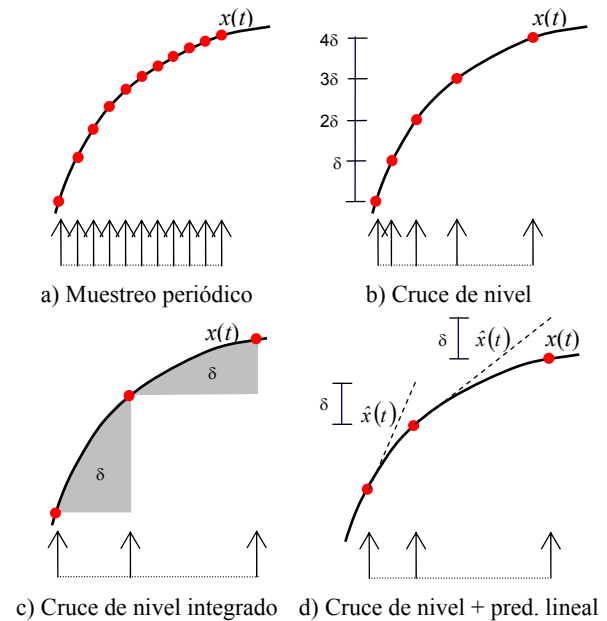


Figura 6. Diferentes estrategias de muestreo

La utilización del muestreo por cruce de nivel abarca prácticamente todas las aplicaciones del muestreo convencional: control (Ellis, 1959; Årzén, 1999; Åström and Bernhardsson, 2002; Kofman, 2003), filtrado (Nielsen and Sparso 1999; Aeschlimann *et al.*, 2004; Li *et al.*, 2006) y procesamiento de señales (Zhu *et al.* 1996; Tsividis 2004). También es una característica estándar en Lon Network (Plönnings *et al.*, 2004) que se utiliza para la interconexión de dispositivo sensores/actuadores inteligente y que proporciona objetos de programación llamados *Standard Network Variable Types* (SNVT) que especifican, por convenio, varias magnitudes físicas comunes tales como tensión y temperatura, su rango, unidad y resolución.

Así mismo, se han diseñado diversos dispositivos de conversión analógico digital que muestrean siguiendo esta técnica (Allier *et al.*, 2005a; Akopyan *et al.*, 2006). Tales conversores no están controlados por ningún reloj global, sino que se actualizan por la señal de entrada analógica. Más aún, lo que se



cuantifica son los intervalos de tiempo entre eventos consecutivos relacionados con la amplitud en lugar de la amplitud de la señal. En los resultados reportados puede verse que esta metodología de muestreo permite reducir de manera importante el consumo de energía en la conversión, lo que sugiere una gran aplicabilidad en dispositivos de sensado remoto (Allier *et al.*, 2005b) a la vez que permite alcanzar un ancho de banda considerable.

La figura 7 ilustra el esquema básico funcional de los dispositivos de muestreo por cruce de nivel. Dicho esquema consta de un comparador que provoca una señal cuando la diferencia entre la entrada analógica y la última muestra se hace mayor que un determinado valor. De acuerdo con el signo de la diferencia, la señal indicará a un contador que debe subir o bajar en una cuenta el valor de la salida digital. La comparación entre la última muestra y la entrada se realiza de manera analógica, por lo cual se debe utilizar un conversor D/A que produzca el valor analógico correspondiente a la última muestra tomada. Los conversores D/A asíncronos representan una nueva área de investigación interesante para el caso de la tecnología de sensores inalámbricos.

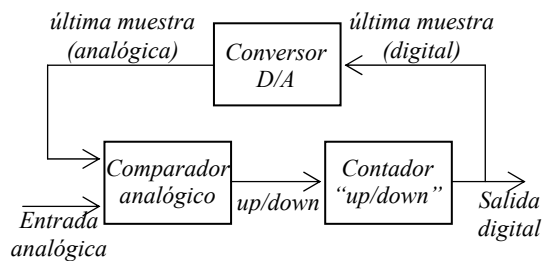


Figura 7. Muestreador por cruce de nivel

Formalmente diremos que dada una señal continua  $y(t)$ , el resultado de muestrear dicha señal por cruce de nivel produce la secuencia  $y(t_k)$ , donde los instantes de muestreo  $t_k$  están definidos según:

$$t_{k+1} = \min_{t \geq t_k} \{t\} \text{ tal que } |y(t) - y(t_k)| \geq \delta \quad (1)$$

donde  $\delta$  es el paso de cuantificación, es decir, la distancia entre niveles sucesivos. Este esquema de muestreo se corresponde con el criterio de diferencia de amplitudes constante del muestreo adaptativo.

En algunos casos, por cuestiones prácticas o de seguridad, se suele modificar levemente la definición de la ecuación.(1) de manera tal que se fuerza a que el tiempo entre dos muestras no supere cierto valor límite ó, por el contrario, que este tiempo no sea menor que algún valor mínimo (Árzen, 1999). De hecho, una técnica que pone una cota inferior al intervalo entre dos muestras es el denominado *muestreo de banda muerta* (Vasyutynsky and Kabitzsch, 2007), en el cual se utiliza una frecuencia alta de sobre-muestreo y se consideran luego sólo aquellas muestras que implican un cambio de nivel.

En lo referente a la utilización del muestreo por cruce de nivel en aplicaciones de control, la ventaja principal está relacionada con los sistemas de control en red, ya que cada muestra puede codificarse con un único bit (indicando si la señal subió o bajó respecto de la muestra anterior), lo que puede reducir mucho la tasa de información transmitida (Miśkiewicz, 2006; McCann *et al.*, 2004; Kofman and Braslavsky 2006).

Por otro lado, el muestreo por cruce de nivel constituye una operación no lineal sobre la señal muestreada ya que se puede considerar equivalente a una función de cuantificación con histéresis. Por lo tanto, el diseño y estudio analítico de las propiedades de los sistemas de control muestreado con estas técnicas es más complicado que en el caso del control convencional. En el siguiente apartado se justifica esta equivalencia y como el muestreo por cruce de nivel induce con frecuencia la aparición de ciclos límites en el sistema de control.

### 3.2 Equivalencia de un sistema muestreado por cruce de nivel con un sistema no lineal

En (Dormido and Mellado, 1975) se estudia la equivalencia de un sistema muestreado por cruce de nivel (figura 8) con un sistema no lineal. En la figura 9 se muestra la señal de error  $e(t)$  muestreada de acuerdo con el criterio de cruce por nivel y en la figura 10 la relación que existe entre la salida  $u(t)$  del muestreador retenedor y su entrada. Se observa que la no linealidad resultante se puede considerar una generalización de un relé con histéresis.

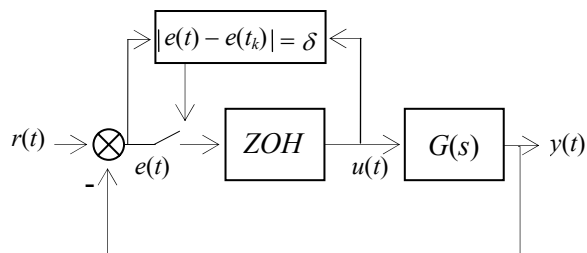


Figura 8. Sistema de control muestreado por cruce de nivel

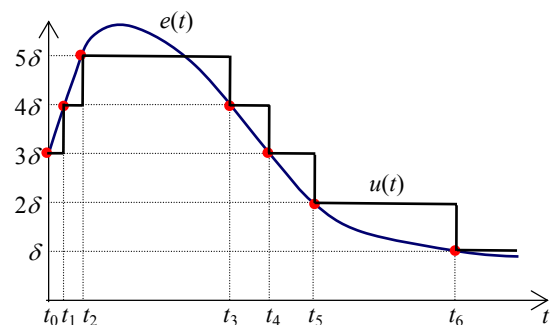


Figura 9. Señales  $e(t)$  y  $u(t)$

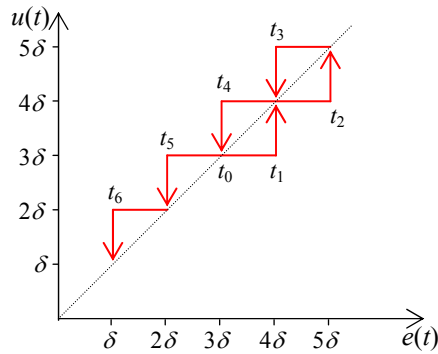


Figura 10. Relación no lineal entre  $e(t)$  y  $u(t)$

De esta forma el sistema de muestreo por cruce de nivel de la figura 8 se puede representar como el sistema no lineal de la figura 11.

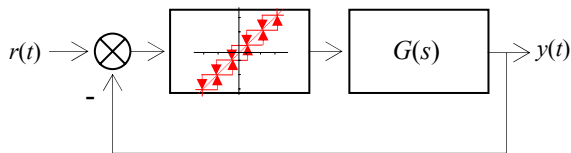


Figura 11. Sistema no lineal equivalente

Los resultados formales alcanzados en estos últimos años en el estudio de la existencia y estabilidad primero local y luego global de ciclos límites en sistemas con relés (Åström, 1995; Johansson *et al.* 1998 y 1999; Goncalves *et al.*, 2001, 2002) permiten ahora abordar el estudio de una manera análoga de este tipo de conducta que aparece de forma natural en los sistemas de muestreo basados en eventos por cruce de nivel. En la figura 12 se muestran algunos resultados de simulación cuando  $G(s) = 3/(s^2+s+1)$  y  $r(t) = 0$ . Se observa que el número de muestreos por cruce de nivel depende de las condiciones iniciales de las cuales parte la planta. La caracterización de los diferentes tipos de ciclos límites es un problema abierto.

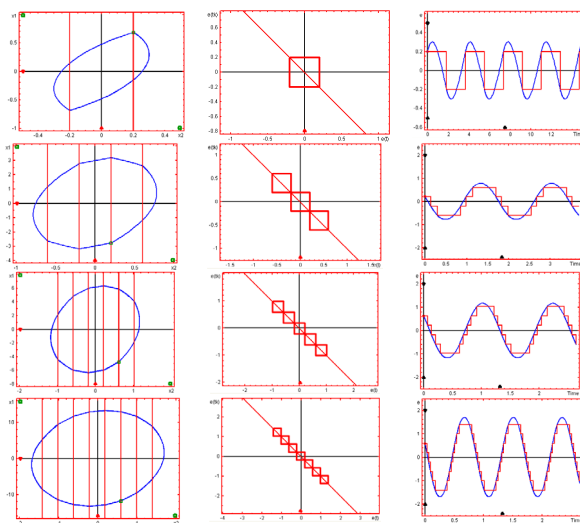


Figura 12. Simulación de un sistema de control muestreado por cruce de nivel

### 3.3 Muestreo por cruce de nivel integrado

La idea del muestreo basado en eventos por cruce de nivel se puede generalizar al muestrear en diferentes dominios de medidas de la señal. Desde el punto de vista práctico, las medidas de señales más importantes son aquellas que tienen una interpretación física clara (p. ej. la integral, la energía, etc). Utilizando estas medidas para la definición del disparo de los eventos es posible definir el muestreo integral y en el dominio de la energía. Tal como se ha comentado una estrategia conducida por eventos, donde la señal de entrada se muestrea de acuerdo con un criterio integral fue ya propuesta por Dorf *et al* en los años 60. La motivación para utilizar el muestreo integral es que con frecuencia la integral del valor absoluto del error (IAE) se toma como índice del comportamiento de un sistema de control (Åström and Hägglund, 2006). En el muestreo con el criterio integral (figura 6c) una señal  $x(t)$  se muestrea si la integral del valor absoluto de la diferencia entre el valor actual de la señal y el último valor muestreado  $x(t_k)$  alcanza un valor umbral dado  $\delta$  (Miskowicz, 2007).

$$\int_{t_k}^{t_{k+1}} |x(t) - x(t_k)| dt = \delta$$

Se pueden señalar algunas ventajas del muestreo integral en aplicaciones de control. En primer lugar en comparación con el muestreo por cruce de nivel da una medida más útil de la calidad en el seguimiento de una señal. Esto se debe a que el criterio integral es una función monótona creciente con el tiempo y como resultado de ello no compensa entre si los errores positivos y negativos. En segundo lugar el muestreo integral garantiza siempre el disparo de un evento en el futuro cosa que no se puede asegurar en el muestreo por cruce de nivel donde puede transcurrir un tiempo excesivamente grande ya que los cambios en la señal no son suficientes como para activar la adquisición de una nueva muestra. En ciertos casos el muestreo integral puede resultar útil con medidas indirectas, en particular si las medidas se llevan a cabo en el dominio de la función primitiva como sucede por ejemplo si la distancia recorrida se estima a través de la velocidad.

### 3.4 Muestreo por cruce de nivel y predicción lineal

En un trabajo reciente (Suh, 2007) ha propuesto un nuevo algoritmo que utiliza un mecanismo de predicción lineal en combinación con un esquema de muestreo por cruce de nivel que reduce aun más el número de muestras necesaria. Su principio de funcionamiento es el siguiente: la señal se transmite si la diferencia entre su valor actual y el valor predicho es mayor que el límite  $\delta$  especificado (figura 6d). El valor predicho se calcula utilizando los valores pasados. De hecho, el método por cruce de nivel se podría considerar como un caso especial del algoritmo propuesto en el sentido de que el valor

predicho es justamente el último valor transmitido. Intuitivamente cabe esperar que en el método por cruce de nivel, el número de muestreos necesarios sea pequeño cuando la señal no cambia, esto es cuando la primera derivada es casi constante. En el algoritmo propuesto, el número de transmisiones se reduce cuando la derivada de orden superior (2ª y 3ª) es casi constante. Se puede demostrar que el nuevo algoritmo reduce el número de transmisiones comparado con el método por cruce de nivel convencional. En la tabla 2 se muestra el código del algoritmo. En el nodo sensor se supone que se muestrea con período  $h$ . Una señal de tiempo discreto  $x_k$  se define por  $x_k = x(kh)$ . Aunque el nodo emisor se muestrea con período  $h$ , la velocidad de transmisión no es  $h$  ya que no se transmiten todas las muestras.

Tabla 2 Cruce de nivel con predicción lineal

Nodo emisor	Nodo receptor
$\hat{x}_0 = \dots = \hat{x}_{-M+1} = 0$ FOR $k := 1$ TO $N$ REPEAT $\hat{x}_k = f(\hat{x}_{k-1}, \dots, \hat{x}_{k-M-1})$ UNTIL $ x_k - \hat{x}_k  > \delta$ transmitir $x_k, \dots, x_{k-M}$ ENDFOR;	$\hat{x}_0 = \dots = \hat{x}_{-M+1} = 0$ FOR $k := 1$ TO $N$ WHILE dato no recibido $\hat{x}_k = f(\hat{x}_{k-1}, \dots, \hat{x}_{k-M-1})$ ENDWHILE $\hat{x}_k = x_k, \dots, \hat{x}_{k-M} = x_{k-M}$ ENDFOR;

En el nodo emisor,  $\hat{x}_k = f(\hat{x}_{k-1}, \dots, \hat{x}_{k-M-1})$  representa un predictor lineal, donde  $M$  denota la longitud de la memoria. Si  $M = 1$ ,  $\hat{x}_k$ , se calcula en función de  $\hat{x}_{k-1}$  y  $\hat{x}_{k-2}$ . Conviene observar que se utiliza  $f(\hat{x}_{k-1}, \dots, \hat{x}_{k-M-1})$  y no  $f(x_{k-1}, \dots, x_{k-M-1})$  como se podría pensar. Si se utilizase  $f(x_{k-1}, \dots, x_{k-M-1})$ , se obtendría  $\hat{x}_k$  de forma más precisa, sin embargo esto obligaría a transmitir  $x_k$  al nodo receptor cada  $h$  segundos lo que implicaría que no se podría reducir el número de transmisiones. Si la diferencia entre el valor actual  $x_k$  y el valor predicho  $\hat{x}_k$  es mayor que  $\delta$  se transmite  $x_k, \dots, x_{k-M}$  en lugar de transmitir  $x_k$  tal como se hace en el método de cruce por nivel.

#### 4. CONTROL BASADO EN EVENTOS

En la figura 13 puede verse un sistema de control muestreado en el cual la señal de salida de la planta  $y(t)$  se muestrea cada vez que la misma cruza por ciertos valores preestablecidos. Como consecuencia, las muestras  $y(t_k)$  no están equiespaciadas en el tiempo. En esta estrategia el muestreo se dispara por la ocurrencia de un evento (en este caso, el cruce de la señal por ciertos valores dados). En consecuencia, cualquiera que sea la estrategia de control utilizada, estaremos frente a un control por eventos. Un detalle importante, sobre todo a la hora de implementar de manera práctica estas estrategias, es que el sistema de muestreo debe tener algún tipo de histéresis. De otra manera, debido a la presencia de ruido o a la propia

realimentación, puede producirse un número muy grande (en principio no acotado) de muestras.

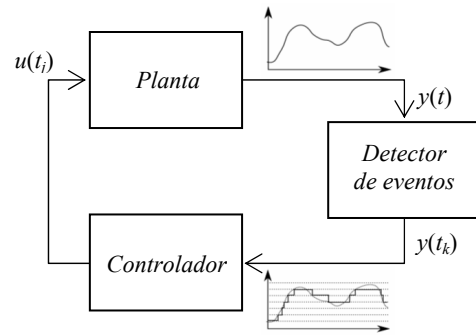


Figura 13. Sistema de control basado en eventos

Hay diferentes formas de diseñar el controlador que recibe las muestras  $y(t_k)$  y calcula las acciones de control  $u(t_j)$ . Según la estrategia de control adoptada, los instantes de muestreo  $t_k$  podrán coincidir o no con los instantes de acción de control  $t_j$ . Analicemos la estructura general de un sistema de control basado en eventos y las diferentes estrategias de diseño de control que funcionan con este tipo de muestreo.

##### 4.1 Estructura general de un sistema de control basado en eventos

En la figura 14 se muestra el diagrama de bloques de la estructura general de un sistema de control basado en eventos (Åström, 2007; Cervin and Åström, 2007). El sistema consta de los elementos siguientes:

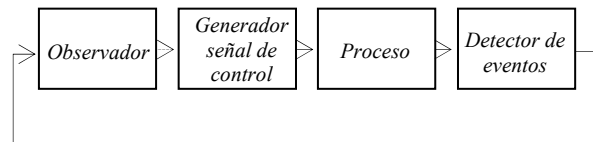


Figura 14. Estructura de un controlador basado en eventos. Las líneas continuas representan transmisión de señales y las punteadas indican transmisión de una señal basada en eventos.

1. *Proceso*
2. *Detector de eventos*. Genera una señal cuando ocurre un evento, por ejemplo cuando la salida cruza ciertos niveles. Los niveles se seleccionan como compromiso entre el número de eventos por unidad de tiempo y el comportamiento del control. Es posible generar eventos diferentes para los cruces hacia arriba y hacia abajo.
3. *Observador*. Actualiza las estimas cuando ocurre un evento y transmite esta información al generador de la señal de control.
4. *Generador de la señal de control*. Produce la señal de entrada al proceso. Se puede considerar un retenedor generalizado (Ohno *et al.*, 2005).

Si se comparan las figuras 13 y 14 se observa que el papel del controlador corresponde conjuntamente al observador y al generador de la señal de control. Estos dos subsistemas operan en lazo abierto entre eventos. Sin embargo la ausencia de eventos es también información útil que puede utilizar el observador. La estrategia de control es una combinación de realimentación + *feedforward*. La acción de realimentación ocurre solo cuando hay eventos, por lo que el actuador está operado en lazo abierto por el generador de la señal de control entre eventos. Es interesante comparar este esquema con el caso de un sistema muestreado convencional donde los eventos los genera un reloj y donde la conducta del sistema viene dictada por la ley de control.

4.2 Controles por emulación de tiempo discreto

Si se supone que la planta de la figura 13 tiene salida  $y(t)$  escalar, habrá una única secuencia  $t_k$  de instantes de muestreo. Por lo tanto, es posible pensar que se tiene un sistema de tiempo discreto en el cual el intervalo entre instantes de muestreo  $t_{k+1}-t_k$  es variable<sup>2</sup>. De esta manera, una alternativa simple es diseñar una ley de control de tiempo continuo y luego discretizarla utilizando alguna técnica convencional (por ejemplo, el método de Euler o la regla trapezoidal). Dado que los parámetros del control discretizado dependen del intervalo de muestreo, estos se irán modificando a medida que se vayan tomando nuevas muestras. Esta es la idea seguida en (Árzén, 1999), donde se estudia la utilización de una ley PID con muestreo basado en eventos. La aproximación propuesta por Árzén se basa en dividir el controlador en dos partes: 1) la lógica de detección de eventos y 2) la lógica de control.

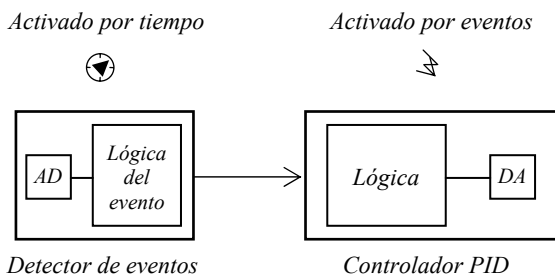


Figura 15 Control PID basado en eventos de Árzén

En la figura 15 se pueden apreciar las dos partes del controlador PID. El detector de eventos se ocupa de indicar a la lógica de control que debe producir una acción de control debido a que se dan las circunstancias adecuadas. La lógica de detección de eventos es sincrónica y está sujeta a un periodo de muestreo nominal de forma que cada  $h_{nom}$  se examina si se cumplen las condiciones necesarias para activar la lógica de control. Estas condiciones pueden ser cualesquiera y tener en consideración los parámetros

2 Si hubiera más de una salida, esto no sería posible ya que habría distintas secuencias de muestreo para las distintas salidas

o variables de estado que se consideren adecuados: valores absolutos o relativos de las variables de estado, puntos de operación, etc. La condición de activación que se utiliza en este trabajo es sencilla:

$$|e(t_k) - e(t_s)| \geq \delta \quad \text{o} \quad (t_k - t_s) \geq h_{max}$$

y expresa que el cálculo de una nueva acción de control puede producirse ante dos situaciones: por error o por tiempo. La primera de las condiciones analiza si el valor absoluto de la diferencia entre el valor actual del error,  $e(t_k)$ , y el valor que disparó el cálculo de la última acción de control,  $e(t_s)$ , supera un umbral  $\delta$ . La segunda condición es una medida de seguridad que obliga a producir una nueva acción de control cuando el tiempo transcurrido desde que se calculó la última supere un límite  $h_{max}$  (figura 16).

Esta doble condición produce también un doble efecto: el control se realizará a la frecuencia máxima,  $1/h_{nom}$ , durante transitorios (cambios de referencia, perturbaciones) y a la mínima,  $1/h_{max}$ , durante los periodos estacionarios.

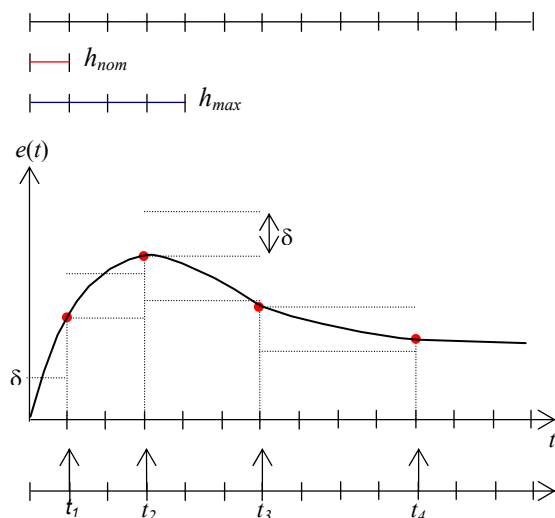


Figura 16. Muestreo en el controlador PID de Árzén

Por lo que respecta a la lógica de control, el algoritmo utilizado es un controlador PID tradicional que se ha discretizado utilizando una aproximación por diferencias hacia adelante en el término integral y hacia atrás en el término derivativo.

$$U(s) = U_P(s) + U_I(s) + U_D(s)$$

$$U_P(s) = K(\beta Y_{sp}(s) - Y(s))$$

$$U_I(s) = \frac{1}{sT_I} E(s)$$

$$U_D(s) = -\frac{KT_D(s)}{1 + sT_D/N} (\gamma Y_{sp}(s) - Y(s))$$

En la Tabla 3 se presenta el código completo del controlador PID. A diferencia del código de otros controladores PID discretizados, una peculiaridad que

presenta éste es que no se puede precalcular una gran parte del código. Ello se debe a que el periodo de muestro,  $h_{act}$ , que se utiliza para generar la acción de control, es variable ya que oscila de un mínimo  $h_{nom}$  a un máximo  $h_{max}$ . En el código del controlador se puede apreciar que un valor elevado de  $h_{act}$  produce un efecto análogo a la aplicación de un valor reducido de  $T_i$  y, recíprocamente, un valor pequeño de  $h_{act}$  produce el mismo efecto que un  $T_i$  elevado. Una de las conclusiones del trabajo es que esta estrategia de muestreo permite reducir de manera importante la utilización de CPU en el controlador digital. Por otro lado, se reconoce la dificultad para tratar analíticamente de esta manera las propiedades del sistema de control muestreado. Una idea similar (con alguna modificación relacionada con el límite superior entre muestras) se desarrolla en (Vasyutynskyy and Kabitzsch, 2006).

Tabla 3. Código del controlador PID de Árzén

```
// Precálculo
bi := K / Ti;
// Lógica de detección de eventos
ysp := ADIn(ch1);
y := ADIn(ch2);
e := ysp - y;
hact := hact + hnom;
IF (abs(e - es) >  $\delta$ ) OR (hact >= hmax) THEN
  es := e;
  ad := Td / (Td + N * hact);
  // Cálculo de la señal de control
  up := K * ( $\beta$  * ysp - y);
  ud := ad * ud - ad * K * N * (y - yold);
  u := up + ui + ud;
  // Enviar acción de control al actuador
  DAOut(u, ch3);
  // Actualización del estado
  ui := ui + bi * hact * (ysp - y);
  yold := y;
  hact := 0.0;
ENDIF;
```

En (Sandee *et al.*, 2005) se presenta otro controlador PID basado en eventos. Aunque la discretización del PID es diferente a la realizada en (Árzén, 1999), su filosofía subyacente es similar ya que consideran la utilización de lógica de detección de eventos y lógica de control. La principal diferencia radica en que utilizan una condición de activación más elaborada:

$$t_k - t_s > \frac{0.5\pi h_{nom}}{\arctan(K_e |e(t_k)|)} \quad y \quad |e(t_k)| > \delta$$

En este caso, la generación de una nueva acción de control se produce cuando se cumplen dos requisitos: el tiempo transcurrido desde el cálculo de la última acción de control y el valor absoluto del error actual,  $e(t_k)$ , superen unos umbrales. Los autores en este trabajo presentan un algoritmo heurístico para el

ajuste de los parámetros de diseño  $\delta$  y  $K_e$  con el objeto de minimizar el error y los eventos generados. En sentido estricto, el controlador PID de Árzén no se puede considerar un controlador basado en eventos puro ya que el detector de eventos se activa por tiempos (figura 15). Lo que hace en realidad es utilizar una técnica de muestreo rápido con período  $h_{nom}$  comprobando si la salida se ha desviado respecto de la muestra previa por encima del umbral  $\delta$  o ha transcurrido un tiempo igual a  $h_{max}$ . En los sistemas de adquisición de datos a estos sistemas se los denomina de *arquitectura compuesta* (de Paoli and Tisato, 1996; Savigni and Tisato., 1999) ya que los eventos asíncronos son presincronizados mediante un muestreo periódico que funciona como tarea de fondo. Los tiempos de muestreo son pues un múltiplo de  $h_{nom}$  acotados superiormente por  $h_{max}$ .

Si se aplica muestreo por cruce de nivel en el controlador PID el detector de eventos se activa en realidad por eventos y no por tiempos tal como se muestra en la figura 17. En la Tabla 4 se muestra el código del controlador PID basado en eventos y en la figura 18 como se produce la actualización de las muestras de la señal de error. Ahora el muestreo únicamente depende de la activación del detector de eventos cuando la señal de error cambia en un valor  $\delta$  respecto de la muestra previa.

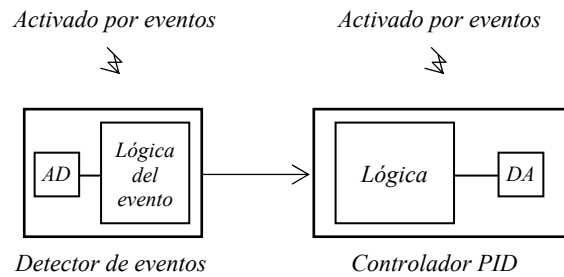


Figura 17. Controlador PID basado en eventos

Tabla 4. Código controlador PID basado en eventos

```
// Precálculo
bi := K / Ti;
// Lógica de detección de eventos
ysp := ADIn(ch1);
y := ADIn(ch2);
e := ysp - y;
IF (abs(e - es) >  $\delta$ ) THEN
  es := e;
  ad := Td / (Td + N * (tr - tr-1));
  // Cálculo de la señal de control
  up := K * ( $\beta$  * ysp - y);
  ud := ad * ud - ad * K * N * (y - yold);
  u := up + ui + ud;
  // Enviar acción de control al actuador
  DAOut(u, ch3);
  // Actualización del estado
  ui := ui + bi * (tr - tr-1) * (ysp - y);
  yold := y;
ENDIF;
```

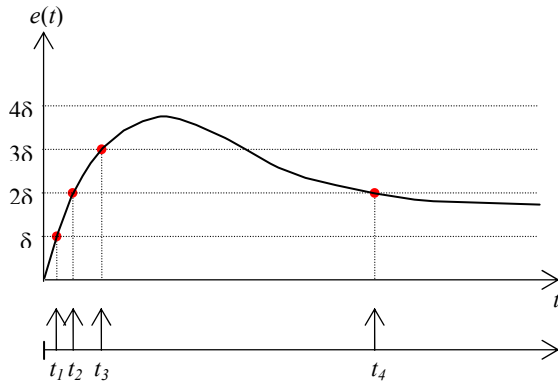


Figura 18. PID con muestreo por cruce de nivel

#### 4.3 Controles asíncronos estáticos

Otra opción para implementar un control por cruce de nivel, ahora sí aplicables a sistemas con múltiples salidas, es calcular una nueva acción de control cada vez que se obtiene una nueva muestra de alguna de las salidas. Dicha acción de control será normalmente de tipo proporcional o, con mayor generalidad, una función de la última muestra tomada en cada salida.

Un ejemplo de esta técnica se presenta en (McCann *et al.*, 2004), donde se estudia el uso de un sistema de control de movimiento en el cual se miden tanto la velocidad como la posición de un motor mediante muestreo por cruce de nivel. La acción de control se calcula como una ganancia matricial sobre ambas variables medidas. Además, se aplica también un muestreo por cruce de nivel sobre las acciones de control, de manera tal que solo se consideran cambios en el control cuando éste cruza ciertos niveles. Este ejemplo, donde el lazo de control está implementado sobre una red de comunicaciones, ilustra bien las ventajas referidas a la reducción del número de muestras transmitidas a través de la red.

Un ejemplo similar se trata en (Otanéz *et al.*, 2002), donde además se comenta la posibilidad de tratar analíticamente el efecto del muestreo por cruce de nivel como perturbaciones. Efectivamente, ignorando retardos y otras no idealidades, con esta idea de muestreo sí es posible establecer algunas propiedades analíticas de manera relativamente simple. Si se observa la figura 5, puede verse que la diferencia entre la señal muestreada y la señal continua original está siempre acotada por  $\delta$ . Por lo tanto, es válido tratar a cada muestreador como si fuera un dispositivo que introduce una perturbación acotada en el sistema de control.

Esto es además equivalente a lo que ocurre en los sistemas de control con mediciones cuantificadas. En estos casos, el método típico de análisis es el de Lyapunov (Liberzon, 2000), que permite establecer una cota final para los estados del sistema en función del ancho de la cuantificación.

#### 4.4 Emulación de eventos discretos

Otro camino para implementar un controlador que reciba las señales muestreadas por cruce de nivel es el de construir un sistema de eventos discretos que se comporte de manera aproximada a la de un controlador continuo previamente diseñado.

Una forma sistemática de aproximar un sistema continuo (en este caso el controlador original) mediante un sistema de eventos discretos (el controlador digital resultante) la da el método de cuantificación de estados (Kofman and Junco, 2001). Este método de integración numérica es también denominado método de QSS (*Quantized State Systems*). La combinación de muestreo por cruce de nivel de la salidas de la planta y la aproximación por QSS del controlador da lugar al llamado control de estados cuantificados (Kofman, 2003), o control QSC (*Quantized State Control*).

Formalmente, consideremos una planta continua (posiblemente no lineal) descrita de la forma siguiente:

$$\begin{aligned}\dot{x}_p &= f_p(x_p, u_p, t) \\ y_p &= g_p(x_p, t)\end{aligned}\quad (2)$$

y se supone que se diseñó un controlador continuo (posiblemente no lineal) descrito por:

$$\begin{aligned}\dot{x}_c &= f_c(x_c, u_c, u_{ref}(t)) \\ y_c &= g_c(x_c, u_c, u_{ref}(t))\end{aligned}\quad (3)$$

donde  $u_p = y_c$  y  $u_c = y_p$  definen la realimentación y  $u_{ref}$  es una señal de referencia. Un controlador QSC asociado al control continuo (3) está dado por una aproximación QSS del mismo:

$$\begin{aligned}\dot{x}_c &= f_c(q_c, u_c, u_{ref}(t)) \\ y_c &= g_c(q_c, u_c, u_{ref}(t))\end{aligned}\quad (4)$$

donde cada componente de  $q_c$  es una versión cuantificada de la correspondiente componente de  $x_c$ . Esto es, entre los estados del controlador (en el vector  $x_c$ ) y sus versiones cuantificadas (en el vector  $q_c$ ) hay relaciones idénticas a las mostradas en la figura 5.

Puede demostrarse que el sistema dado por (4) se comporta idéntico a un sistema de eventos discretos<sup>3</sup> bajo el formalismo DEVS (Å *et al.*, 2000). Particularmente, dicho sistema de eventos discretos generará eventos cada vez que haya un cambio en  $u_c$  (o sea, cada vez que alguna salida de la planta sea muestreada debido a un cruce de nivel), cada vez que haya un cambio en  $u_{ref}$  (cambios de referencia), y cada vez que haya un cambio en alguna variable

<sup>3</sup> El detalle de los modelos de eventos discretos equivalentes a la aproximación QSS de un sistema genérico puede encontrarse en (Kofman and Junco 2001) y en (Cellier and Kofman 2006).

cuantificada  $q_c$  (o sea, cada vez que el estado del controlador cruce un nivel de cuantificación).

Al igual que en el caso del control asíncrono estático, el control QSC puede analizarse a partir de considerar la presencia de perturbaciones acotadas: por un lado hay perturbaciones en la salida de la planta (debido al muestreo por cruce de nivel), y por otro hay perturbaciones en los estados internos del controlador (debido a la cuantificación). Estas últimas perturbaciones son también acotadas ya que la diferencia entre  $x_c$  y  $q_c$  está también acotada.

Utilizando el método directo de Lyapunov se ha demostrado que cuando el sistema continuo original de lazo cerrado formado por el conjunto de ecuaciones (2) y (3) es asintóticamente estable, al utilizar el esquema QSC (con una cuantificación no muy grande) se consigue estabilidad práctica, es decir, las trayectorias se mantienen en un entorno acotado del punto de equilibrio. Este resultado es válido para sistemas no lineales (Kofman 2003) incluidos los casos no estacionarios (Kofman 2005a).

En casos lineales, el resultado es aun más fuerte: se ha demostrado que la estabilidad práctica se conserva siempre, sin importar cuan grande sea la cuantificación. Más aun, la diferencia entre las trayectorias del sistema continuo ideal en lazo cerrado y del sistema con el control QSC se mantiene siempre acotada por una magnitud que depende linealmente de la cuantificación utilizada en los estados del controlador y en el muestreo por cruce de nivel (Kofman 2002, Kofman and Braslavsky 2006). Esta relación lineal se encuentra mediante una técnica menos conservadora que la del análisis de Lyapunov (Kofman 2005b).

El diseño de control usando QSC combina las ventajas de los enfoques vistos anteriormente: realiza una emulación de un sistema de control continuo arbitrario (lo que permite utilizar estrategias de control complejas) y, por otro lado, puede analizarse el comportamiento teórico a partir del efecto de perturbaciones acotadas. Esto es incluso una ventaja sobre los métodos tradicionales de emulación con muestreo uniforme, donde en casos no lineales es muy difícil dar resultados teóricos de estabilidad.

Desde un punto de vista práctico, QSC conserva las ventajas de los métodos de control con muestreo por cruce de nivel: reducción de la tasa de transmisión en sistemas de control en redes (Kofman and Braslavsky 2006) y reducción del uso de CPU (Kofman 2002). Un último comentario respecto de QSC es que los controles asíncronos estáticos analizados antes pueden verse como casos particulares de QSC donde la dimensión del estado del control es cero.

#### 4.5 Diseño directo por eventos discretos

En todos los casos vistos hasta aquí, el controlador por cruce de nivel se diseñó como una emulación de

un controlador continuo. Una alternativa a esto es intentar diseñar un control sabiendo de antemano que lo que se va a recibir es una secuencia de muestras de la salida equiespaciadas en los valores, y sabiendo por otro lado que se cuenta con libertad para realizar cambios en la acción de control en cualquier momento.

Usando esta idea, en (Kofman and Braslavsky 2006) se presenta una estrategia de estabilización de una planta lineal inestable basada en el muestreo por cruce de nivel. Suponiendo que la planta tiene orden  $n$ , se la deja en lazo abierto hasta que se reciben  $n+1$  muestras. Luego, utilizando la información del tiempo transcurrido entre cada par de muestras, se demuestra que es posible calcular el estado inicial de la planta. Una vez conocido el estado, se propone utilizar un control *bang-bang* de tiempo mínimo que (con un número mínimo de conmutaciones y un tiempo mínimo transcurrido) lleva el estado de la planta nuevamente al origen.

Este método minimiza el número de transmisiones a través de la red tanto desde la planta hacia el controlador (gracias al muestreo por cruce de nivel) como desde el controlador hacia la planta (por la estrategia de control *bang-bang* de tiempo mínimo). De todas maneras, es un método poco robusto ya que se basa en la medición exacta del tiempo entre muestras y asume un conocimiento exacto de la dinámica de la planta y de la ausencia de ruido. El comportamiento de este método en presencia de perturbaciones y cuantificación en la medición temporal se estudia en (Braslavsky *et al.*, 2006).

## 5. APLICACIÓN DEL MUESTREO Y CONTROL BASADOS EN EVENTOS A LOS SISTEMAS DE CONTROL EN RED

Como consecuencia directa del proceso de implantación que han tenido las redes de comunicación en todos los ámbitos de la sociedad, durante los últimos años ha surgido un creciente interés en toda la comunidad de control por la utilización de redes de comunicación de propósito general para la construcción de lazos de realimentación en los niveles más bajos de los sistemas jerárquicos de control y supervisión, es decir, y coloquialmente hablando, realizar el cierre del lazo de control a través de redes de datos carentes de las especificaciones adecuadas para operar en tiempo real (Ethernet o redes basadas en protocolos TCP/IP). Esto se ha dado en llamar sistemas de control en red, teniendo en inglés dos equivalentes: *network-based control systems*, *networked control systems* (la más habitual) o, sencillamente, NCS. La figura 19 presenta un esquema simplificado de un NCS en el que la red de comunicación se basa en la familia de protocolos TCP/IP. La principal diferencia con un sistema de control tradicional es que el canal de comunicación local entre el controlador y el proceso se sustituye por una red de comunicación de

datos basada en paquetes, lo que introduce retardos de transmisión de longitud variable y posibles pérdidas de paquetes de información.

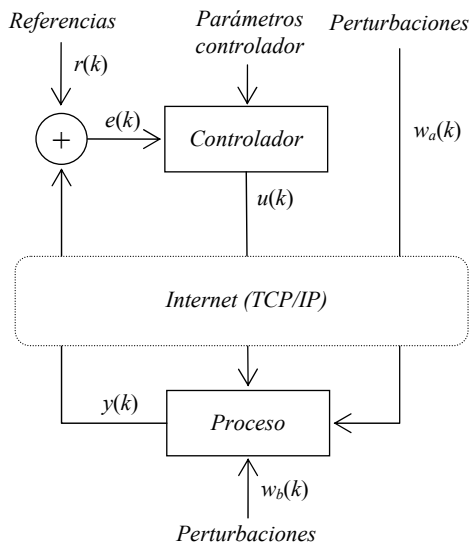


Figura 19. Esquema de un sistema de control en red.

Las motivaciones para el uso industrial de estas redes de datos basadas en paquetes son muy variadas: bajo coste, portabilidad, accesibilidad, flexibilidad, interoperabilidad y mantenimiento. Por otra parte, en el ámbito de la enseñanza de la ingeniería de control comienzan a aparecer grupos de trabajo que pretenden situar el controlador del laboratorio remoto basado en web en el computador que maneja el estudiante y no en el laboratorio junto al equipo a controlar. En este caso, la red a disposición del estudiante, por lo general, siempre es Internet.

Los problemas que conlleva la aproximación basada en redes cableadas y protocolos de propósito general vienen motivados por las propias imperfecciones de la comunicación: retardos de transmisión no acotados y pérdida de paquetes de datos (dependiendo del nivel del protocolo utilizado). Además, cuando se recurre al uso de redes inalámbricas (Bluetooth, Wi-Fi, ZigBee), los problemas anteriores se agudizan y el ancho de banda es otra restricción adicional que obliga a recurrir a técnicas de cuantificación para optimizar el tamaño de los datos a transmitir.

Dado el interés que han despertado los sistemas de control en red, la literatura tanto a nivel de congresos como de revistas es muy abundante. La mayor parte de los trabajos son extensiones de la teoría de control de sistemas discretos basados en la aproximación clásica del flujo de información de un sistema de control, es decir, en el envío de la misma información que circularía en un sistema de control tradicional (estados o salidas del proceso y acción de control a aplicar). Muchos de estos trabajos consideran las características del tipo de red utilizada como especificaciones iniciales y proceden al diseño de un controlador que garantice estabilidad asintótica o

práctica frente a retardos y pérdidas de información. Otros trabajos, se basan en el diseño del controlador considerando inicialmente una red ideal (capacidad infinita, ausencia de pérdidas de datos), para, seguidamente, proceder a manipular el tamaño e instancias de los datos para optimizar el rendimiento del sistema en lazo cerrado. Ya que no es el objetivo de este trabajo efectuar una revisión del estado del arte de los sistemas de control en red, algunas referencias útiles son (Tipsuwan and Chow, 2003; Hristu-Varsakelis and Levine, 2005; Yang, 2006; Antsaklis and Baillieul, 2007).

### 5.1 Control en red basado en paquetes

Pese a todos los inconvenientes que conlleva la utilización de redes generalistas basadas en paquetes, éstas poseen una ventaja que no se debe obviar: el hecho de que la cantidad mínima de datos útiles que un paquete puede llevar es, por lo general, muy superior a lo que es requerido en un sistema de control tradicional. Por ejemplo, la longitud del campo de datos de un paquete Ethernet puede oscilar de un mínimo de 46 a un máximo de 1500 bytes sin introducir *overhead* adicional (Lian et al., 2001). En lo que respecta a la familia TCP/IP, el estándar IPv4 establece una longitud mínima de 576 bytes para el envío de datagramas IP sin fragmentación. Como la cabecera máxima de un datagrama IP es de 60 bytes quedan libres 516 bytes para paquetes UDP o TCP en un datagrama IP; ya que la cabecera de un paquete UDP es de 8 bytes y la de un paquete TCP es de 24 bytes, quedan disponibles 508 y 492 bytes, respectivamente, para el envío de datos sin introducir sobrecarga adicional por fragmentación de los datagramas IP. En contraste con esto, un dato consume 2 ó 4 bytes, con lo que la mayor parte de la carga útil que puede llevar un paquete al enviar un único dato no se aprovecha. Además, es por esta razón, disponer de suficientes bits para la codificación y el envío de información, que el error de cuantificación no debe constituir un problema y queda enmascarado por el ruido de medida y del proceso.

El maximizar el aprovechamiento del campo de datos de los paquetes para el envío de información sin incurrir en costes extras de sobrecarga ha dado lugar a sistemas de control en red basados en el envío de paquetes con acciones de control futuras (*packed-based -PB-NCS-* o *packetized networked control systems -PNCS-*). Esta aproximación se apoya en técnicas e ideas del ámbito del control predictivo, y permite, hasta cierto punto, ignorar las características de la red en lo relativo al diseño del controlador ya que únicamente requiere de un modelo de la planta en base al cual producir las acciones de control futuras a enviar en cada paquete de control.

Gran parte del interés de los investigadores en el control basado en paquetes viene acentuado por la utilización de sistemas autónomos de comunicación inalámbrica dotados de sensores y procesadores que



permiten el despliegue de sistemas de medida y control en lugares de difícil acceso a un coste muy bajo (por ejemplo, en invernaderos o en galerías subterráneas). En estos entornos resulta fundamental optimizar la utilización de los recursos disponibles (tráfico de datos, potencia de las transmisiones, cálculos realizados) para aumentar al máximo el tiempo de vida de las baterías de que están dotados estos sistemas. Resulta obvio que el control basado en eventos tiene mucho que decir en este tipo de entornos de trabajo distribuido.

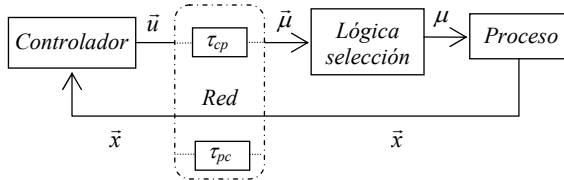


Figura 20. Esquema genérico de un sistema PB-NCS. En el controlador se engloban todos los instrumentos adicionales para la generación de los paquetes de control: observador, predictor y ley de control. Los retardos que introducen el controlador y la lógica de selección son despreciables en comparación con los retardos de la red.

Veamos en un ejemplo como funciona esta metodología de diseño de NCS. Consideremos el sistema NCS que se presenta en la figura 20 en donde  $\bar{u}$  y  $\bar{\mu}$  se denominan *paquetes de control* y  $\bar{x}$  *paquete de estado*. Los paquetes de control  $\bar{u}$  y  $\bar{\mu}$  contienen las acciones de control antes de enviarse y una vez recibidas, respectivamente, mientras que  $\bar{x}$  contiene el vector del estado con información actual de la planta (no se debe descartar que se incluyan vectores de estado anteriores si el controlador carece de esa información y la requiere). El proceso considerado es un sistema LTI discreto de la forma:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k \end{aligned}$$

donde  $x_k \in R^m$  es el estado,  $u_k \in R^n$  es la señal de control,  $w_k \in R^m$  es una perturbación desconocida pero acotada,  $v_k$  es el ruido de medida, y el controlador viene definido por:

$$u_k = Kx_k$$

donde  $K$  ha sido seleccionada de forma que  $(A+BK)$  es estable y las matrices  $A$ ,  $B$  y  $C$  son conocidas, disponiendo el controlador de un modelo de la planta para estimar su comportamiento futuro

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k$$

El actuador y el sensor funcionan de manera síncrona con periodo  $h$  y el controlador es asíncrono (espera a

recibir un paquete de datos con el estado actual de la planta para generar la acción de control)<sup>4</sup>. En lo que respecta a la red se considera que el retardo total  $\tau_{total} = \tau_{pc} + \tau_{contr} + \tau_{cp}$  es inferior a  $h$  y que pueden existir pérdidas de información en ambos canales. En la figura 21 se puede apreciar cómo se realiza el intercambio de datos del sistema de la figura 20.

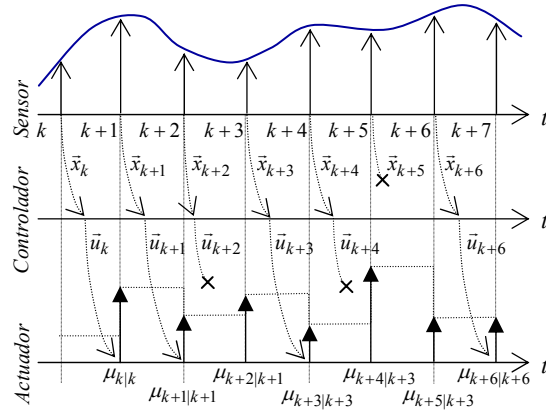


Figura 21. Esquema del intercambio de información en un sistema de control PB-NCS.

Denotamos  $u_{k+j|k}$ ;  $j = 0, 1, \dots, N$  a la acción de control calculada en el instante  $k$  que se aplicará en el instante  $k+j$ , siendo  $N$  el número total de acciones de control que se pueden enviar en un paquete.

Supongamos que en el instante  $k$ , el sensor envía el estado de la planta  $x_k$  al controlador, el cual mediante el modelo de la misma calcula las acciones de control hasta el instante  $k+N$ , es decir,  $\bar{u}_k = \langle u_{k|k}, u_{k+1|k}, \dots, u_{k+N|k} \rangle$  constituye el paquete de control a enviar a la planta. Obsérvese que toda la secuencia, con la excepción del primer valor, depende del modelo de la planta que se utilice como predictor.

Cuando la planta reciba el paquete  $\bar{u}_k$ , denominado a partir de ahora  $\bar{\mu}_k$ , procederá a eliminar del buffer de actuación todas las acciones de control previas (calculadas en los instantes  $k-1$  o anteriores) y la lógica de selección procederá a aplicar la secuencia de control en el instante de tiempo en que corresponda: en  $k$  aplicará  $\mu_{k|k}$ ; en  $k+1$  corresponderá  $\mu_{k+1|k}$  y, en general, la acción de control a aplicar en el instante  $k+M$  será:

$$\mu_{k+M} = \begin{cases} \mu_{k+M|k} & \text{si } M \leq N \\ \gamma \mu_{k+N|k} & \text{si } M > N \end{cases}$$

donde  $\gamma$  indica si la elección cuando finalizan las acciones de control recibidas en el último envío es

<sup>4</sup> En algunas ocasiones este tipo de controladores se denominan *basados en eventos* aunque su comportamiento es *semisíncrono*.

aplicar 0 ( $\gamma = 0$ ) como acción de control o mantener la última ( $\gamma = 1$ ).

En caso de producirse la pérdida de un paquete, por ejemplo, el paquete  $\bar{\mu}_{k+1}$  no llega a la planta, entonces se procede a la aplicación de las acciones de control del envío previo, en este caso,  $\bar{\mu}_k$ . En este supuesto, en  $k+1$  y los instantes sucesivos se aplicarán  $\mu_{k+1|k}$ ,  $\mu_{k+2|k}$ ,  $\mu_{k+3|k}$ , ..., que son acciones de control recibidas en el paquete  $\bar{\mu}_k$ .

En la figura 21 se plantean dos situaciones de pérdida de información. La primera corresponde a pérdidas entre el controlador y el actuador (paquetes de control  $\bar{u}_{k+2}$  y  $\bar{u}_{k+4}$ ), que provocan que se continúen aplicando las acciones de control almacenadas en el buffer que llegaron en los envíos previos (en el caso de la pérdida del paquete  $\bar{u}_{k+2}$  se aplican las acciones del paquete previo,  $\bar{u}_{k+1}$  y en el caso del  $\bar{u}_{k+4}$  se aplican las acciones del  $\bar{u}_{k+3}$ ). La segunda situación sucede cuando el paquete con el estado actual del proceso no llega al controlador (paquete  $\bar{x}_{k+5}$ ) que no realizará la emisión del paquete  $\bar{u}_{k+5}$  con lo que la situación pasa a ser análoga a la anterior, es decir, la pérdida de información del paquete de control. Además, se puede observar en este ejemplo que puede ser conveniente que el sensor envíe en el mismo paquete no sólo el vector de estado actual sino todos los vectores anteriores de que no disponga el controlador. El objetivo de ello es mantener en el controlador información completa de la evolución del proceso para que pueda utilizarse por el predictor. Para el supuesto anterior, el sensor debería haber enviado el paquete de estado  $\bar{x}_{k+6} = \langle x_{k+5}, x_{k+6} \rangle$ .

Obsérvese que se ha considerado que la planta y el controlador mantienen la sincronía de sus relojes, aunque este último trabaje sólo ante la llegada de nuevos datos. Mediante ciertas modificaciones este esquema puede adaptarse a situaciones más reales: hay pérdidas de paquetes y, además, los retardos de comunicación están presentes en ambos canales y son, por lo general, superiores a  $h$ . Una solución ya estudiada se basa en colocar marcas de tiempo (*time-stamps*) a los vectores de estado de forma que manteniendo el sincronismo del reloj entre planta y controlador es posible conocer la magnitud del  $\tau_{pc}$  en cada envío y obtener el estado actual de la planta mediante filtros de Kalman u otros métodos (Goodwin *et al.*, 2004).

El enfoque del control en red basado en paquetes que contienen información extra de control no ha sido profundamente explorado pero comienza a despertar un gran interés en la comunidad que investiga sobre sistemas NCS. La razón de ello es que aborda el problema mediante un *principio de separabilidad*: se aprovechan las características de la infraestructura de

una red existente para aplicar un esquema de control que se sabe que trabaja en un marco ideal (retardos inferiores a  $h$ , no hay pérdidas de datos y total sincronía entre planta y controlador). Sin embargo, el PB-NCS en su concepción original presenta un inconveniente: el excesivo tráfico cuando se trabaja con periodos de muestro pequeños puede incrementar el tráfico de red y provocar mayores retardos debido a la congestión que se crea.

El primer trabajo, ya clásico, en el que se recurre al concepto de *buffers* para anular la existencia de retardos en la comunicación y de un observador/predictor para calcular las acciones de control futuras es (Luck and Ray, 1990), aunque el planteamiento no considera la transmisión de paquetes sino valores individuales. Donde por primera vez ya se plantea el uso conjunto de predictores en el controlador y de buffers en el actuador como remedio contra la llegada aperiódica e incompleta de datos es en (Kim *et al.*, 2005). Hasta este momento, los trabajos en esta línea se basan en la utilización de técnicas que provienen del ámbito del control predictivo (Liu *et al.*, 2006; Yang *et al.*, 2006; Epstein *et al.*, 2007; Quevedo *et al.*, 2007) y del control óptimo (Georgiev and Tilbury, 2006; Gupta *et al.*, 2007; Kim *et al.*, 2006).

En lo que respecta a la utilización de control basado en eventos en NCS no existen muchos trabajos hasta el momento, ya sea desde la perspectiva clásica (un dato por envío) como desde la basada en paquetes. El trabajo que más se aproxima es (Epstein *et al.*, 2007) en el que presenta un algoritmo para la transmisión de paquetes de control denominado IDTS (*Input Difference Transmission Scheme*) basado en la técnica de muestreo por cruce por nivel. Supóngase que en el instante  $k+M$  el controlador produce el paquete  $\bar{u}_{k+M}$  y se define la diferencia entre los elementos de dos paquetes  $\bar{u}_{k+M}$  y  $\bar{u}_k$  como:

$$\Delta u_k^{k+M}(j) = u_{k+M}(j) - \begin{cases} u_k(M+j) & \text{si } M+j \leq N+1 \\ \gamma u_k(N+1) & \text{en caso contrario} \end{cases}$$

para  $j = 1, \dots, N$ . El criterio que utilizan para determinar si enviar el paquete de control se define como:

$$\|\Delta u_k^{k+M}(1)\| > \bar{U}$$

donde  $\bar{U}$  es un parámetro de diseño. La principal motivación para recurrir a un esquema IDTS es coincidente con el uso del muestreo y control basado en eventos: minimizar el tráfico de información a costa de maximizar el envío de información de control que sea relevante.

Es importante remarcar que todas las aproximaciones que se han citado en este apartado son sincronicas ya que se basan en el empleo de un periodo  $h$  para muestrear la planta así como para el cálculo y aplicación de las acciones de control futuras. Por otra

parte, varios de los trabajos referenciados consideran en sus planteamientos una red ideal y descartan la existencia de retardos en uno o en ambos canales y/o consideran un ancho de banda infinito.

## 6. CONCLUSIONES

Los sistemas de control periódicos o activados por tiempo han dominado casi de manera exclusiva la investigación en ingeniería de control ya que existe una teoría de sistemas madura y bien desarrollada para señales muestreadas periódicamente. Sin embargo, hay aplicaciones donde el muestreo periódico no resulta deseable. El control basado en eventos es en estos casos una alternativa muy prometedora particularmente cuando se consideran sistemas con capacidades reducidas de computación y de comunicación.

Un sistema de control basado en eventos viene caracterizado por el hecho de que es la ocurrencia de un evento, en lugar del paso del tiempo lo que decide cuando se debe realizar el muestreo. En este trabajo se ha presentado una revisión panorámica de la situación actual del campo de los sistemas de control basados en eventos que es un tema que está teniendo un interés creciente en los últimos años dentro de la comunidad de control. Sin embargo la teoría de este tipo de sistemas está como muy bien apuntan Åström y Bernhardsson todavía en su infancia (Åström and Bernhardsson, 1999) con relativamente pocos resultados de carácter general y muy escasas aplicaciones significativas.

Se han analizado los compromisos que plantean la implementación de los sistemas de control basados en eventos desde el punto de vista del software y los esquemas de muestreo basados en eventos más comúnmente utilizados. Finalmente se ha presentado la implicación que tiene su aplicación en el campo de los nuevos sistemas de control en red.

## REFERENCIAS

- Aeschlimann, F., E. Allier, L. Fresquet and M. Renaudin (2004). Asynchronous FIR filters: towards a new digital processing chain. *Proceedings of the 10th International Symposium on Asynchronous Circuits and Systems (ASYNC'04)*. 198-206.
- Akopyan, F., R. Manohar and A. B. Apsel (2006). A level-crossing flash asynchronous analog-to-digital converter. *Proceedings of the 12th IEEE International Symposium on Asynchronous Circuits and Systems*.
- L. Albertoni, A. Balluchi, A. Casavola, C. Gambelli, E. Mosca, and A. L. Sangiovanni-Vincentelli (2005). Idle speed control of port-injection engines via the polynomial approach, *IFAC World Congress*, Prague.
- Allier, E., G. Sicard, L. Fesquet and M. Renaudin (2005a). Asynchronous level crossing analog to digital converters. *Measurement* **37**, 296-309.
- Allier, E., J. Goulier, G. Sicard, A. Dezzani, E. Andre and M. Renaudin (2005b). 120nm low power asynchronous adc. *Proceedings of the 2005 International Symposium on Low Power Electronics and Design. ISLPLED'05*, 60-65.
- Antsaklis, P. and J. Baillieul (2007). Special issue on Technology of Networked Control Systems, *Proceedings of the IEEE*, **95**(1), 5-8.
- Araki, M. and K. Yamamoto (1986). Multivariable multirate sampled-data systems. *IEEE Trans on Automatic Control* **31**, 145-154.
- Årzén, K. J. (1999). A Simple event-based PID controller. *Proceedings of 14th IFAC World Congress*. vol. 18. Beijing, China. 423-428.
- Åström, K. J. (1995). Oscillations in systems with relay feedback. *The IMA Volumes in Mathematics and its Applications: Adaptive Control, Filtering, and Signal Processing*, **74**, 1-25.
- Åström, K. J. (2007) Event based control. In A. Astolfi and L. Marconi, editors, *Analysis and Design of Nonlinear Control Systems*. Springer Verlag.
- Åström, K. J. and B. Wittenmark (1997). *Computer controlled systems: Theory and design*. Third Edition. Prentice Hall. New Jersey.
- Åström, K. J. and B. M. Bernhardsson (1999): Comparison of periodic and event based sampling for first order stochastic systems. *Proceedings of the 14th IFAC World Congress*, Beijing, China. **11**, 301-306.
- Åström, K. J. and B. M. Bernhardsson (2002). Comparison of Riemann and Lebesgue sampling for first order stochastic systems. *Proceedings of the 41st IEEE Conference on Decision and Control*. Las Vegas, 2011-2016.
- Åström, K. J. and T. Häggglund (2006). *Advanced PID Control*, ISA..
- Baillieul, J. (2002). Feedback design in information-based control, *Lecture Notes in Control and Information Sciences*, **280**, 35-58
- Balluchi, A., P. Murrieri and A. L. Sangiovanni-Vincentelli (2005). Controller synthesis on non-uniform and uncertain discrete-time domains. *Proceedings of Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, **3414**, Springer Verlag, 118-133.
- Bekey, G. A. and R. Tomovic (1966). Sensitivity of discrete systems to variation of sampling interval. *IEEE TAC*, **11**, 282-284.
- Berge, M.H., B. Orlic and J.F. Broenink (2006). Co-simulation of networked embedded control systems, a CSP-like process oriented approach. *Proceedings of the IEEE International Symposium on Computer-Aided Control Systems Design*, Munich, Germany. 434-439.
- Blanchini, F. (1994). Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. *IEEE Trans on Automatic Control*, **39**(2), 428-433.

- Blanchini, F. (1999). Set invariance in control. *Automatica*, **35**, 1747-1767.
- Braslavsky, J., E. Kofman and F. Felicioni (2006). Effects of time quantization and noise in level crossing sampling stabilization. *Anales de AADECA 2006*. Buenos Aires, Argentina.
- Buttazzo, G. C. (1997). *Hard real-time computing systems: Predictable scheduling algorithms and applications*. Kluwer Academic Publishers.
- Cassandras, C. G. and S. Lafortune (1999). *Introduction to discrete event systems*. Kluwer Academic Publishers.
- Cellier, F. and E. Kofman (2006). *Continuous System Simulation*. Springer. New York.
- Cervin, A., D. Henriksson, B. Lincoln, J. Eker, and K. E. Årzén (2003). How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime. *IEEE Control Systems Magazine*, **23**(3), 16-30.
- Cervin, A. and K. J. Åström (2007). On Limit Cycles in Event-Based Control Systems, *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans.
- Ciscato, D. and L. Mariani (1967). On increasing sampling efficiency by adaptive sampling. *IEEE Trans on Automatic Control*, **12**, 318.
- Cloosterman, M., N. van der Wouw, M. Heemels and H. Nijmeijer (2006). Robust stability of networked control systems with time-varying network-induced delays. *Proceedings of the IEEE Conference on Decision and Control*, San Diego, 4980-4985.
- de Bruin, D. and P.P.J. van den Bosch. (1998). Measurement of the lateral vehicle position with permanent magnets. *IFAC Workshop on Intelligent Components for Vehicles (ICV'98)*, Seville, 9-14.
- de la Sen, M. (1996). Non-periodic and adaptive sampling: A tutorial review. *Informatica, Journal of the Lithuanian Academy of Sciences*, **7**(2), 175-228.
- Dormido, S., M. Mellado, J. Ruiz and J. M. Guillen (1973). Sistemas de muestreo adaptativo mediante un criterio de diferencia de amplitudes constante, *Revista de Automática*, **16**, 13-17.
- Dormido, S. and M. Mellado (1975). Determinación de ciclos límites en sistemas de muestreo adaptativo, *Revista de Automática*, **26**, 21-31.
- Dormido, S. and M. Mellado (1977). A study on fixed-difference sampling scheme, *Applications and Research in Information Systems and Sciences*, **2**, 496-500.
- Dormido, S., M. de la Sen, M. Mellado (1978). Criterios generales de determinación de leyes de maestro adaptativo, *Revista de Informática y Automática*, **38**, 13-29.
- de Paoli, F. and F. Tisato, (1996). On the complementary nature of event-driven and time-driven models. *Control Engineering Practice* **4** (6), 847-854.
- Dodds, S. J. (1981). Adaptive, high precision, satellite attitude control for microprocessor implementation, *Automatica*, **17**(4), 563-573.
- Dorf, R. C., M. C. Farren, C. A. Phillips (1962). Adaptive sampling for sampled-data control systems. *IEEE TAC*, **7** (1), 38-47.
- El Gamal, A., C. Nair, B. Prabhakar, E.U. Biyikoglu and S. Zahedi (2002). Energy efficient scheduling of packet transmissions over wireless networks. *Proceedings of IEEE INFOCOM*, 1773-1780.
- Ellis, P. (1959). Extension of phase plane analysis to quantized systems. *IRE Transactions on Automatic Control* **4**(2), 43-54.
- Epstein, M., L. Shi, S. Di Cairano and R. M. Murray (2007). Control over a network: Using actuation buffers to reduce transmission frequency, *Proceedings of the European Control Conference*, 597-602.
- Feency, L. M. and M. Nilsson (2001). Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. *Proceedings of IEEE Infocom*, 1548-1557.
- Franklin, G. F., J. D. Powell and A. Emami-Naeini (2005). *Feedback Control of Dynamic Systems*. Fifth edition, Prentice Hall.
- Georgiev, D. and D.M. Tilbury (2006). Packet-based control: The  $H_2$ -optimal solution, *Automatica*, **42**(1), 137-144.
- Glad, T. and L. Ljung (1984). Velocity estimation from irregular, noisy position measurements. *Proceedings of the IFAC 9th Triennial World Congress*, Budapest, **2**, 1069-1073.
- Goncalves, J. M., A. Megretski, and M. A. Dahleh (2001). Global stability of relay feedback systems", *IEEE Trans on Automatic Control*, **46**(4), 550-562.
- Goncalves, J. M. (2002). *Constructive global stability analysis of hybrid systems*, Ph. D., MIT.
- Goodwin, G.C., H. Haimovich, D.E. Quevedo and J.S. Welsh (2004). A moving horizon approach to networked control system design. *IEEE Trans on Automatic Control*, **49**(9), 1427-1445.
- Gupta, S. C. (1963). Increasing the sampling efficiency for a control system. *IEEE Trans on Automatic Control*, **8** (3), 263-264.
- Gupta, V., S. Adlakh, B. Sinopoli and A. Goldsmith (2007). Towards receding horizon networked control, *Automatica* (Submitted).
- Heemels, W.P.M.H., R.J.A. Gorter, A. van Zijl, P.P.J. van den Bosch, S. Weiland, W.H.A. Hendrix and M.R. Vonder (1999). Asynchronous measurement and control: a case study on motor synchronization, *Control Engineering Practice*, **7**, 1467-1482.
- Hsia, T. C. (1972a). On a unified approach to adaptive sampling system design. *IEEE Control and Decision Conference*, 618-622.
- Hsia, T. C. (1972b). Comparisons of adaptive sampling control laws. *IEEE Trans on Automatic Control*, **17** (6), 830-831.

- Hsia, T. C. (1974). Analytic design of adaptive sampling control laws. *IEEE Trans on Automatic Control*, **19** (1), 39-42.
- Hristu-Varsakelis, D. and W. S. Levine (2005). *Handbook of networked and embedded control systems*. Birkhäuser Boston.
- Inose, H., T. Aoki and K. Watanabe (1966). Asynchronous delta-modulation system. *Electronics Letters* **2**(3), 95-96.
- Ishii, H. and B. A. Francis (2002). *Limited data rate in control systems with networks*, Springer Verlag, **275**, Lecture Notes in Control and Information Sciences..
- Johansson M. and A. Rantzer (1998). Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Trans on Automatic Control*, **43**(4), 555-559.
- Johansson, K. H., A. Rantzer, and K. J. Åström (1999). Fast switches in relay feedback systems, *Automatica*, **35**(4), 539-552.
- Kao, C.-Y. and B. Lincoln (2004). Simple stability criteria for systems with time-varying delays. *Automatica*, **40**, 1429-1434.
- Kawka, P.A., and A.G. Alleyne (2005). Stability and feedback control of wireless network systems. *Proceedings of the American Control Conference*, Portland. 2953-2959.
- Kim, K., K. Ji and A. Ambike (2005). Networked real-time control strategies dealing with stochastic time delays and packet losses, *Proceedings of the American Control Conference*, 621-626.
- Kim, K., K. Ji and A. Ambike (2006). Real-time operating environment for networked control systems, *IEEE Trans on Automation Science and Engineering*, **3**(3), 287-296.
- Kofman, E. (2002). Quantized-state control of linear systems. *Proceedings of AADECA 2002*. Buenos Aires, Argentina.
- Kofman, E. (2003). Quantized-state control. A method for discrete event control of continuous systems.. *Latin American Applied Research* **33**(4), 399-406.
- Kofman, E. (2005a). Discrete event control of time varying plants. *Latin American Applied Research* **35**(2), 135-141.
- Kofman, E. (2005b). Non-conservative ultimate bound estimation in LTI perturbed systems. *Automatica*, **41**(10), 1835-1838.
- Kofman, E. and J. Braslavsky (2006). Level crossing sampling in feedback stabilization under data-rate constraints. *Proceedings of CDC'06, IEEE Conference on Decision and Control*. San Diego, 4423-4428.
- Kofman, E. and S. Junco (2001). Quantized state systems. A DEVS approach for continuous system simulation. *Transactions of SCS* **18**(3), 123-132.
- Krucinski, M., C. Cloet, M. Tomizuka, and R. Horowitz (1998). Asynchronous observer for a copier paper path. *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, Florida, **3**, 2611-12.
- Levis, P, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer and D. Culler (2004). The emergence of networking abstractions and techniques in TinyOS. *Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation*, 1-14.
- Lian, F.L., J.R. Moyne and D.M. Tilbury (2001). Performance evaluation of control networks, *IEEE Control Systems Magazine*, **21**(1), 66-83.
- Li, Y., K. Shepard and Y. Tsividis (2006). A continuous-time programmable digital FIR filter. *IEEE Journal of Solid-State Circuits* **41**(11), 2512-2520.
- Liberzon, D. (2000). Nonlinear stabilization by hybrid quantized feedback. In: *Proceedings of the Third International Workshop on Hybrid Systems, Computation and Control*. Pittsburgh, PA. 243-257.
- Lincoln, B. (2002). Jitter compensation in digital control systems. *Proceedings of the American Control Conference*, 2985-2990.
- Lindgärde, O. and B. Lennartson (1998). Comparing frequency analysis methods for sampled-data control. *Proceedings of the 37th Conference on Decision and Control*, Tampa, Florida USA. New York: IEEE, **1**, 829-834.
- Liu, G.P., J.X. Mu, D. Rees and S.C. Chay (2006). Design and stability analysis of networked control systems with random communication time delay using the modified MPC, *International Journal of Control*, **79**(4), 288-297.
- Luck, R. and A. Ray (1990). An observer-based compensator for distributed delays. *Automatica*, **26**(5), 903-908.
- McCann, R., A. K. Gunda and S. R. Damugatla (2004). Improved operation of networked control systems using Lebesgue sampling. *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting*, **2**, 1211-1216.
- Mead, C. A. (1989). *Analog VLSI and Neural Systems*, Addison Wesley.
- Miskowicz, M. (2003). The event-triggered sampling optimization criterion for distributed networked monitoring and control systems. In: *Proc. of the IEEE International Conference on Industrial Technology*, **2**, 1083-1088.
- Miskowicz, M. (2006). Send-on-delta concept: An event-based data reporting strategy. *Sensors* **6**(1), 29-63.
- Miskowicz, M. (2007). Asymptotic effectiveness of the event-based sampling according to the integral criterion, *Sensors*, **7** (1), 16-37.
- Mitchell J. R. and W. L. McDaniel (1969). Adaptive sampling technique. *IEEE Trans on Automatic Control*, **14** (2), 200-201
- Nielsen, L. and J. Sparso (1999). Designing asynchronous circuits for lowp: An IFIR filter bank for a digital hearing aid. *Proceedings of the IEEE*, **87**(2), 268-281.
- Nilsson, J. (1998). *Real-time control systems with delays*. PhD thesis ISRN LUTFD2/TFRT-

- 1049-SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- Normey-Rico, J. and E. F. Camacho (2006). Predicción para control: Una panorámica del control de procesos con retardo. *Revista Iberoamericana de Automática e Informática Industrial*, **5**(4), 5-25.
- Norsworthy, S. R., R. Schreier and G. Temes (1996). *Delta-Sigma Data Converters*, IEEE Press.
- Ohno, K., M. Hirata, and R. Horowitz (2005). A comparative study of the use of the generalized hold function for HDDs, *IEEE/ASME Trans on Mechatronics*, **10**(1), 26-33.
- Otanez, P. G., J. R. Moyne and D. M. Tilbury (2002). Using deadbands to reduce communication in networked control systems. *Proceedings of the American Control Conference*, **4**, 3015-3020.
- Phillips, A.M., and M. Tomizuka (1995). Multirate estimation and control under time varying data sampling with applications to information storage devices. *Proceedings of the American Control Conference*, **6**, 4151-4155.
- Plönnings, J., M. Neugebauer, K. Kabitzsch (2004). A traffic model for networked devices in the building automation. *IEEE International Workshop on Factory Communication Systems*, Vienna, Austria, 137-145.
- Ploplys, N., P. Kawka and A. Alleyne (2004). Closed-loop control over wireless networks. *IEEE Control Systems Magazine*, **24**(3), 58-71.
- Quevedo, D., E.I. Silva, y G.C. Goodwin (2007). Packetized predictive control over erasure channels, *Proceedings of the American Control Conference*, 1003-1008.
- Sandee, J.H., W.P.M.H. Heemels and P.P.J. van den Bosch (2005). Event-driven control as an opportunity in the multidisciplinary development of embedded controllers, *Proceedings of the American Control Conference*, 1776-1781.
- Sanz, R. and K. E. Årzén (2003). Trends in software and control. *IEEE Control Systems Magazine*, **23**(3), 12-15.
- Savigni, F. and F. Tisato, (1999). Kaleidoscope: A reference architecture for monitoring and control systems. *Proceedings of the First Working IFIP Conference on Software Architecture*, 369-388
- Sayiner, N., H.V. Sorensen and H.R. Viswanathan (1996). A level-crossing sampling scheme for a/d conversion. *IEEE Transactions on Circuits and Systems II* **43**(4), 335-339.
- Schinkel, M., C. Wen-Hua, A. Rantzer (2002). Optimal control for systems with varying sampling rate. *Proceedings of the American Control Conference*, 2979-2984.
- Schinkel, M., W.P.M.H. Heemels and A.Lj. Juloski (2003). State estimation for systems with varying sampling rate. *Proceedings 42nd IEEE Conference on Decision and Control*, 391-392.
- Sira-Ramirez, H. (1989). A geometric approach to pulse width modulated control in nonlinear dynamical systems. *IEEE Transactions on Automatic Control*, **35**(12), 184-187.
- Smith, M. J. (1971). An evaluation on adaptive sampling. *IEEE Trans on Automatic Control*, **16**, 282-284.
- Stark, W., H. Wang, A. Worthen, S. Lafortune and D. Teneketzis (2002). Low energy wireless communication network design. *IEEE Wireless Communication*, 60-72.
- Suh, Y. S. (2007). Send-on-delta sensor data transmission with a linear predictor. *Sensors* **7**, 537-547.
- Tait, K. E. (1966) An analysis and evaluation of signal dependent sampling in discrete continuous control systems. *International Journal of Control*, **4**, 201-239.
- Tomovic, R. and G. A. Bekey (1966) Adaptive sampling based on amplitude sensitivity, *IEEE Trans on Automatic Control*, **11**, 282-284.
- Tipsuwan, Y. and M. Y. Chow (2003). Control methodologies in networked control systems, *Control Eng. Pract.*, **11**, 1099-1111.
- Tsividis, Y. (2004). Digital signal processing in continuous time: a possibility for avoiding aliasing and reducing quantization error. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, **2**, 589-592.
- Tsyppkin, Y. Z. (1984). *Relay control systems*. Cambridge University Press, Cambridge, UK.
- Vasyutynskyy, V. and K. Kabitzsch (2006). Implementation of PID controller with send-on-delta sampling. *Proceedings of ICC'2006*. Glasgow, Scotland.
- Vasyutynskyy, V. and K. Kabitzsch (2007). Towards comparison of deadband sampling types. *IEEE International Symposium on Industrial Electronics. ISIE 2007*, 2899-2904.
- Yang, C., Z. Shan-an, K. Wan-zeng and L. Li-ming (2006). Application of generalized predictive control in networked control system, *Journal of Zhejiang University SCIENCE A*, **7**(2), 225-233.
- Yang, T.C. (2006). Networked control system: a brief survey, *IEE Proceedings - Control Theory and Applications*, **153**(4), 403-412.
- Zeigler, B., T. G. Kim and H. Praehofer (2000). *Theory of modeling and simulation. Second edition*. Academic Press. New York.
- Zhang, W., M. S. Branicky and S.M. Philips (2001). Stability of networked control systems. *IEEE Control Systems Magazine*, **21**(1), 84-99.
- Zhao, F and L. Guibas (2004). *Wireless sensor networks*. Elsevier
- Zhu, Y. S., S. W. Leung and C. M. Wongm (1996). A digital audio processing system based on nonuniform sampling delta modulation. *IEEE Trans on Consumer Electronics* **42** (1), 80-86