

Generación extrínseca de eventos para el control de sistemas continuos

Oscar Miguel-Escrig (omiguel@uji.es)

Julio-Ariel Romero-Pérez (romeroj@uji.es)

Departamento de Ingeniería de Sistemas y Diseño. Universitat Jaume I

Campus del Riu Sec. Av. Vicent Sos Baynat s/n.

CP 12071 Castelló de la Plana. España

Resumen

En este artículo se propone el diseño de un circuito electrónico para la generación de eventos por cruce de niveles de una señal analógica. Este tipo de circuito puede ser útil en la implementación de controladores basados en eventos para evitar la generación de eventos por programa dentro del propio código del controlador a partir de un muestreo periódico de la señal del sensor. Se muestra un estudio experimental que demuestra cómo la generación extrínseca de eventos por vía electrónica puede contribuir a reducir el coste computacional asociado al control, además, acerca la implementación de los controladores basados en eventos a los principios que este tipo de control persigue.

Palabras clave: control basado en eventos, PID, convertidor AD

1. INTRODUCCIÓN

Recientemente se han realizado numerosas investigaciones sobre el control basado en eventos (CBE) de sistemas continuos. El CBE persigue dos objetivos fundamentales: 1) Reducir la cantidad de información necesaria para realizar el control en bucle cerrado 2) Disminuir el coste computacional medio que requieren los algoritmos de control.

En los controladores basados en eventos el algoritmo de control se ejecuta sólo tras la ocurrencia de eventos asíncronos que indican cambios significativos en el estado del sistema, en contraposición con los controladores basados en tiempo, usados comúnmente, los cuales se ejecutan a un periodo constante. Un ejemplo de generación de eventos asíncronos son los generados cuando la señal cruza distintos niveles preestablecidos [4].

Una de las primeras contribuciones al desarrollo de los controladores PID basados en eventos fue introducida por Árzén en [1]. El objetivo era reducir el uso de la CPU de los sistemas de control basados en computador sin afectar de forma significativa el comportamiento del bucle de control. Varios trabajos posteriores estuvieron dirigidos a resolver algunos problemas desvelados por Árzén

en su propuesta, principalmente relacionados con el tiempo integral [5, 6, 10, 11].

En concreto, el algoritmo de Árzén plantea una llamada periódica al controlador no siéndolo así la ejecución del cálculo, o sea, la actualización de la acción de control. Para realizar esto último Árzén propone una lógica de detección de eventos basada, por una parte, en que la diferencia de errores entre la última ejecución y la llamada actual supere un umbral. Además impone una condición basada en el tiempo máximo que el controlador puede estar sin recalcularse su salida.

La técnica de generación de eventos usada por Árzén se conoce como *Send-on-Delta* (SOD) y ha sido usada en numerosos estudios y aplicaciones dada su simplicidad. Una variante de SOD, la cual incorpora una histéresis en el muestreo y umbrales de generación de eventos fijos, conocida como *Symmetric-Send-on-Delta* (SSOD), fue propuesta en [2]. Varios métodos de sintonía de controladores PID han sido desarrollados para esta estrategia de generación de eventos [3, 7, 8, 9].

En la gran mayoría de trabajos presentados en el ámbito del CBE, la lógica de generación de eventos es implementada mediante programación, formando parte del código del propio controlador. En este trabajo se propone un generador electrónico de eventos, que produce un evento cada vez que una señal analógica cruza un nivel preestablecido. Esto evita el uso de la detección de eventos por programa, ofreciendo así una aproximación más acorde con el principio fundamental del CBE: la ejecución del controlador responde a un evento.

El artículo se estructura como sigue: en la sección 2 se plantea la problemática general de la implementación del CBE y se presenta la alternativa propuesta con generación extrínseca de los eventos. En la sección 3 se detalla el diseño electrónico del circuito de generación de eventos y seguidamente, en la sección 4, se muestra el comportamiento del circuito obtenido. En la sección 5 se realiza una caracterización temporal del coste de los algoritmos con un ejemplo de control PID implementado en la norma IEC 61499. Finalmente, en la sección 6, se exponen las conclusiones sobre

este trabajo.

2. PROBLEMÁTICA

Como se ha comentado anteriormente, el control basado en eventos por cruce de niveles se ha desarrollado tradicionalmente con un muestreo periódico a una frecuencia suficientemente alta para detectar el cruce de los niveles y con ello actualizar la acción de control, tal y como se muestra en la Figura 1, donde se aprecia que se realiza un muestreo periódico de la señal del sensor mientras que el control esta basado en eventos según la lógica de generación de eventos.

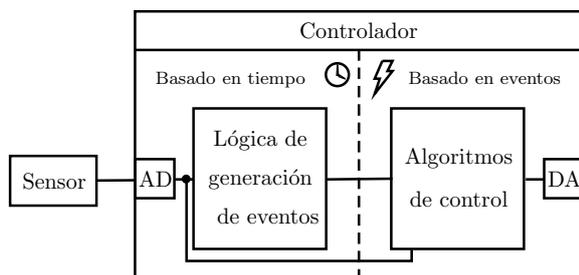


Figura 1: Control basado en eventos con lógica de generación de eventos interna.

Este hecho conlleva que no se reduzca el coste computacional en los casos donde la lógica de detección de eventos se implementa directamente en el controlador, ya que una gran parte de la carga computacional la forma el muestreo y el propio cálculo de la lógica de generación de eventos.

La alternativa que se propone en este artículo es la incorporación de un circuito electrónico el cual permite la detección de los cruces de nivel, y por ende, la generación de eventos, tal y como muestra la Figura 2. En esta figura se asume que la salida del circuito de generación de eventos estará conectada a una entrada de interrupción de un computador industrial (PLC, micro-controlador,...), que una vez reciba la señal de interrupción realizará el muestreo y los cálculos de la acción de control, ahorrando de esta manera todo el tiempo de cómputo usado en muestrear la señal analógica y aplicar la lógica de generación de eventos por programa.

2.1. Generación de eventos

En cuanto a la lógica de generación de eventos basada en cruce de nivel, en la literatura se distinguen dos variantes fundamentales: la más conocida es la SOD que consiste en asignar a la señal de salida el valor de la señal de entrada siempre y cuando esta cambie en más de un cierto valor δ . Otra estrategia es la conocida como SSOD, en

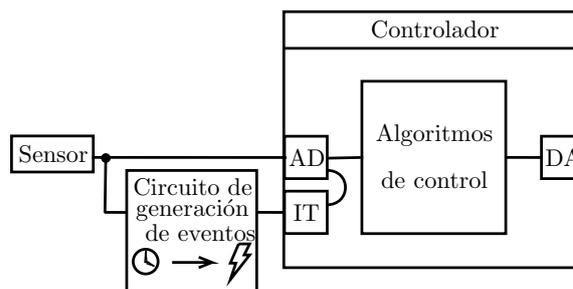


Figura 2: Control basado en eventos con lógica de generación de eventos externa.

la cual los eventos son generados cada vez que la señal de entrada cruza los valores $k * \delta$.

A pesar de su similitud, entre las estrategias SOD y SSOD existe una diferencia significativa: los umbrales de la primera dependen del valor inicial de la señal de entrada mientras que en SSOD los umbrales de generación de eventos son fijos. Este hecho se puede observar en la Figura 3, donde se ha muestreado una señal con y sin offset con estas estrategias y se puede observar claramente que mientras que en el muestreo SSOD los niveles de muestreo son fijos, en el muestreo SOD estos niveles varían en función de donde empezó a realizarse el muestreo.

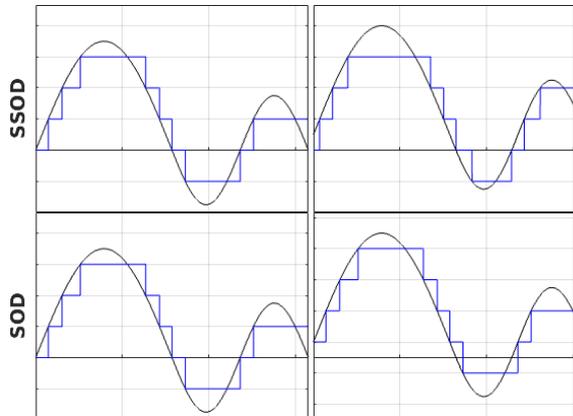


Figura 3: Señal con y sin offset muestreada con estrategias SSOD (parte superior) y SOD (parte inferior).

3. DISEÑO DEL GENERADOR DE EVENTOS

La estructura general del circuito de generación de eventos que se propone en este trabajo se muestra en la figura 4. El mismo cuenta con un convertidor A/D o ADC (*Analog to Digital Converter*) cuya entrada es la señal analógica a partir de la cual se quieren generar los eventos, y cuyos datos convertidos serán pasados a un circuito secuencial que

generará un cambio en la salida del valor lógico alto al bajo y viceversa cada vez que se genera un evento.

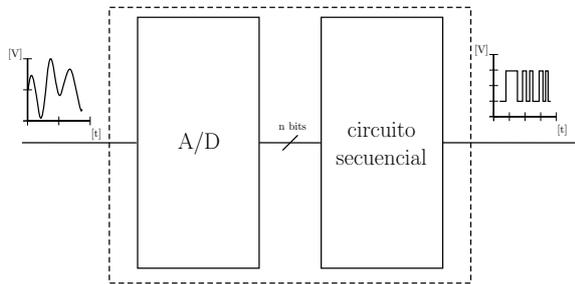


Figura 4: Esquema de la estructura del circuito a diseñar.

Esta estructura es adecuada para implementar estrategias de generación de eventos con umbrales fijos, del estilo de SSOD, ya que los cambios de las salidas del ADC, a partir de los cuales se generaran los eventos por el circuito secuencial, tienen lugar para valores fijos de la señal de entrada.

Para el diseño se ha considerado en un principio que la entrada del circuito son los 2 bits menos significativos o LSB (*Least Significant Bits*) de la conversión del ADC, por lo tanto, en primer lugar se ha caracterizado el comportamiento del muestreo SSOD según estos dos bits como se muestra en la Figura 5, donde una señal tipo rampa se ha convertido según la lógica SSOD y se ha asignado a cada tramo de valor $\delta/2$ un par de bits.

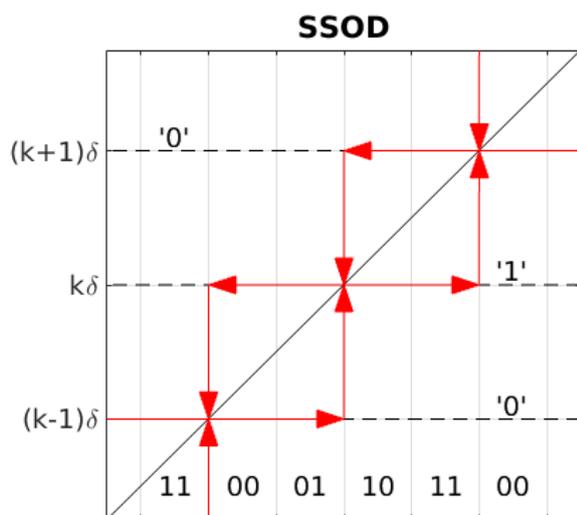


Figura 5: Lógica del circuito de detección SSOD sobre la conversión de una señal de entrada.

Sobre esta misma figura, se puede observar entrecuadrado el valor de la salida del circuito que corresponde a cada uno de los valores que toma la señal muestreada $((k-1)\delta, k\delta, (k+1)\delta)$, y a partir de estos datos se puede construir el diagrama

de transición de estados como se muestra en la Figura 6, donde en cada estado se representa el par de bits (bit1,bit0) y su salida correspondiente.

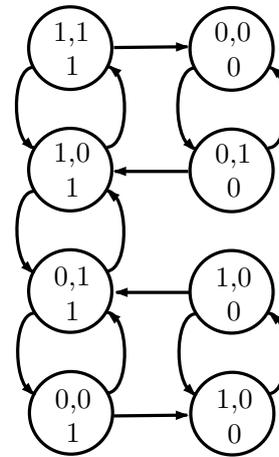


Figura 6: Diagrama de transición de estados para la lógica de generación de eventos SSOD.

Sobre este diagrama se ha aplicado un modelado usando la metodología de Huffman y las simplificaciones oportunas sobre la tabla de Karnaugh dando como resultado un circuito de generación de eventos que resultó ser demasiado complejo, por lo que esta solución fue desechada.

En su lugar, se ha planteado una alternativa al método de muestreo SSOD, la cual es bastante similar a este método y nos permite el diseño mediante la estructura propuesta para la tarjeta, ya que mantiene los umbrales de detección fijos. Esta variante del SSOD la cual nombraremos a partir de ahora como QSSOD (*Quasi Symmetric Send on Delta*) mantiene la detección por niveles δ pero se reduce el ancho de histéresis a $\delta/2$. Sobre la señal muestreada esto se traduce en un menor desfase en el muestreo de la señal y una reacción más rápida a los cambios de dirección de la señal a muestrear.

Caracterizando el comportamiento del circuito como se ha hecho anteriormente, se puede observar en la Figura 7 el comportamiento del muestreo, la asignación a cada tramo del par de bits correspondientes del ADC y la salida del circuito para cada nivel.

Como en el caso anterior, se ha representado el diagrama de transición de estados como se muestra en la Figura 8, y se ha realizado la síntesis del circuito secuencial correspondiente.

La síntesis concluyó con un circuito muy simple el cual se muestra en la Figura 9. La entrada son los 2 LSB del ADC, y su funcionamiento se corresponde con la lógica buscada: se guarda el bit 1 cuando la conversión da un número par (bit 0

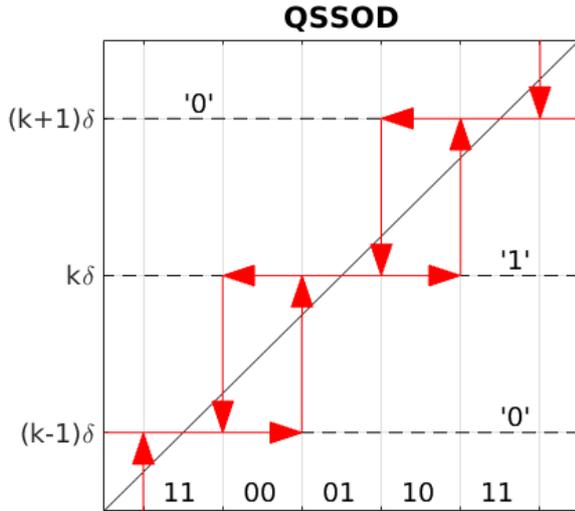


Figura 7: Lógica del circuito de detección QSSOD sobre la conversión de una señal de entrada.

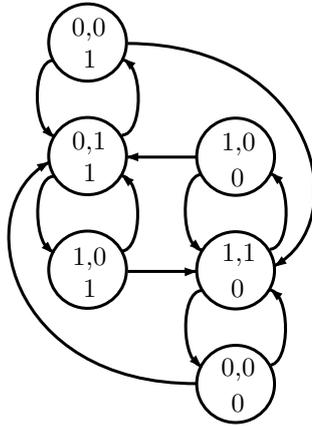


Figura 8: Diagrama de transición de estados para la lógica de generación de eventos QSSOD.

igual a 0) por medio de un biestable tipo D, y se compara este valor del bit 1 con el anterior valor guardado de este bit, si son diferentes, se envía un evento, es decir, se envía un evento cuando hay un cambio de valor par entre conversiones del ADC. Sin embargo, por las características de conversión del ADC no se sigue este mismo comportamiento cuando hay un cambio de dirección en la señal muestreada, dando como resultado la reducción del valor de la histéresis a la mitad de ese valor. El resultado final es que se envían eventos cada vez que se pasa un umbral de valor δ siempre y cuando no haya cambio de dirección, en cuyo caso, el envío se realizará cuando se pase un valor $\delta/2$.

El valor de la δ de detección se puede configurar fácilmente en función de los parámetros del ADC. En caso de cambiar los bits 0 y 1 de detección por otros bits, la explicación anterior sobre el funcio-

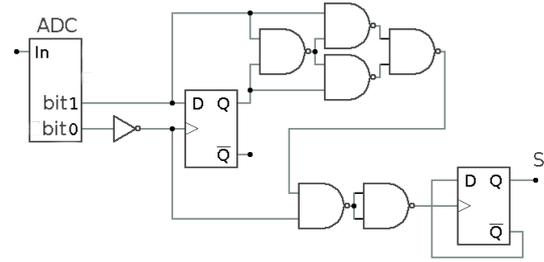


Figura 9: Circuito de generación de eventos con lógica tipo QSSOD.

namiento sigue siendo válida, lo único que cambia es que ahora no es entre conversiones pares consecutivas, es decir, se realiza cada n valores pares, por ejemplo, si se usaran el bit 2 y 1, en vez de cada 2 unidades, sería cada 4 unidades cuando se enviaría un evento. El valor de δ viene definido por la expresión (1), donde FSR es el ancho de conversión del ADC, n_{ADC} es el número de bits del ADC y n_{SEL} es el menor de los bits de conversión seleccionado.

$$\delta = \frac{FSR}{2^{n_{ADC} - (n_{SEL} + 1)}} \quad (1)$$

La señal de salida del circuito planteado es una señal cuadrada que cambia de valor entre 0 y 1 (niveles lógicos) cada vez que se detecta el cambio correspondiente de la señal, por lo tanto, el autómata deberá programarse para detectar la interrupción por flanco de subida y de bajada (*Rising Edge* y *Falling Edge*).

4. EXPERIENCIAS DE FUNCIONAMIENTO

En la Figura 10 podemos ver el resultado de la tarjeta desarrollada con dos generadores de eventos montados sobre ella. Se ha desarrollado con entrada de alimentación externa debido a los posibles picos de consumo de los componentes que pueden dañar a algunos equipos no protegidos debidamente.

Para comprobar su funcionamiento se han realizado varios experimentos. Primero se ha estudiado la respuesta ante una entrada sinusoidal con la ayuda de un generador de ondas. Como podemos ver en la Figura 11 la señal digital de salida cambia entre el valor alto y bajo lógico con más velocidad en función de la pendiente de la señal de entrada, cambiando con más frecuencia de valor cuando esta es más grande y no cambiando en absoluto cuando el seno cambia de dirección. Cada cambio de la salida corresponde con un cruce

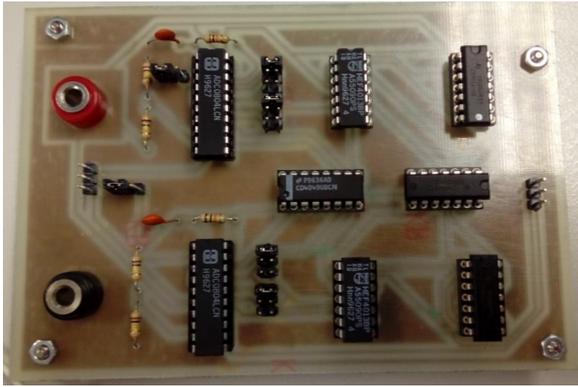


Figura 10: Circuito de generación de eventos en físico con la lógica propuesta.

de nivel de la entrada, o sea, con la generación de un evento.

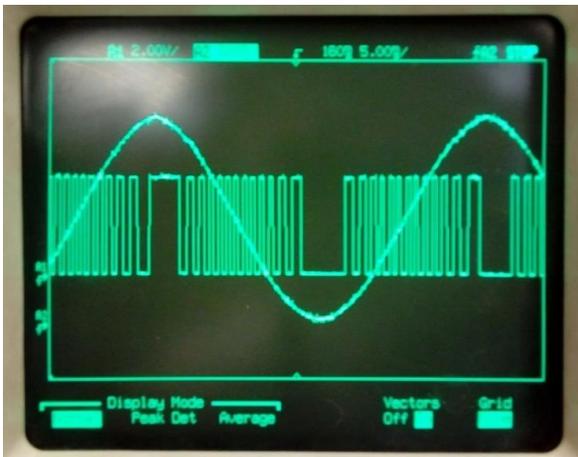


Figura 11: Muestreo de una señal sinusoidal con el circuito de generación de eventos.

Para mostrar el correcto funcionamiento del circuito generador de eventos con un ejemplo concreto, se han escogido los bits 1 y 2 del convertidor ADC, y sabiendo las características del convertidor ADC, el cual tiene una amplitud de escala de 5V y 8 bits de conversión, podemos calcular el valor de δ con la expresión (1) sustituyendo directamente ($FSR = 5V$, $n_{ADC} = 8$ y $n_{SEL} = 1$), obteniendo los valores de histéresis $\delta = 78,12mV$ y $\delta/2 = 39,06mV$.

Con estos datos se ha introducido a la entrada del circuito una señal triangular, cuya pendiente es constante, y por lo tanto, también lo es la duración de los pulsos generados porque los niveles definidos por δ se cruzan de forma regular, siempre y cuando la señal no cambie de dirección. En la Figura 12 se puede observar que la señal de salida cambia de valor, es decir, se genera un evento, cada vez que la señal muestreada cambia en un valor δ , ya sea cuando esta se incrementa o cuando disminuye (en

la Figura 12, en la parte inferior derecha se puede observar $\Delta V = 78,12mV$). Este comportamiento cuando no hay cambio de dirección es el mismo que se obtiene con el muestreo tipo SSOD.

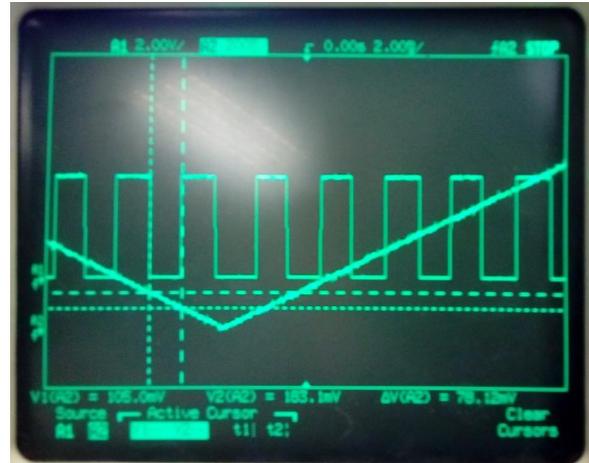


Figura 12: Muestreo de una señal triangular con el circuito de generación de eventos (I).

Cuando la señal cambia de dirección, se puede observar en la Figura 13 que el generador de eventos conmuta antes de que se llegue a un incremento de δ (en la Figura 13, en la parte inferior derecha se puede observar $\Delta V = 40,62mV$) a un valor bastante próximo a $\delta/2$ con lo cual se puede confirmar que la propuesta presentada gráficamente en la Figura 7 se puede implementar mediante un circuito.

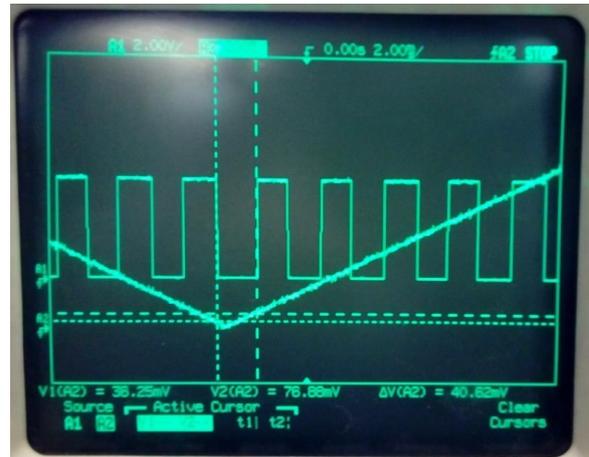


Figura 13: Muestreo de una señal triangular con el circuito de generación de eventos (II).

En este punto cabe hacer hincapié en dos aspectos fundamentales sobre el circuito y sobre la implementación. En primer lugar, para el correcto funcionamiento del circuito se requiere que las conversiones del ADC sean consecutivas respecto a los bits significativos que se hayan escogido, es decir, si al escoger los bits 1 y 0 para definir δ se

hacen dos conversiones consecutivas con los valores de los tres últimos bits 000 y 100 el circuito no detectará que ha habido un cambio de estado cuando en realidad sí que lo ha habido.

Con respecto a este punto de conversiones sucesivas, se han realizado diferentes ensayos sobre este montaje y se ha comprobado que se puede aumentar la frecuencia de una señal triangular de entrada hasta 50 Hz, valor a partir del cual se empieza a perder información en la generación de eventos. Esta velocidad de detección viene dada por la suma acumulativa de los tiempos de conversión del ADC, el tiempo de tratamiento de datos de las puertas lógicas y por el tiempo consumido por los biestables. Podemos concluir pues, que el circuito es plenamente funcional en aquellos sistemas cuya dinámica sea más lenta que la velocidad de generación de eventos.

En segundo lugar, hay que destacar que el circuito tal y como se ha planteado en un origen en la Figura 2 actúa directamente sobre la señal del sensor aunque se le puede introducir al circuito la señal de error si esta se calcula previamente. En los casos donde actúe sobre la señal del sensor, en la implementación del controlador habrá que incluir una ejecución del algoritmo de control cuando se produzca un cambio en la referencia, ya que tomando las condiciones de ejecución con el error y desarrollando:

$$\begin{aligned} |e(T_1) - e(T_2)| &> \delta \\ |(y_r(T_1) - y(T_1)) - (y_r(T_2) - y(T_2))| &> \delta \\ |(y_r(T_1) - y_r(T_2)) - (y(T_1) - y(T_2))| &> \delta \end{aligned}$$

Observando la última expresión obtenida podemos ver que el evento generado por el cambio $|(y(T_1) - y(T_2))| > \delta$ es el que produce el circuito, pero el cambio $|(y_r(T_1) - y_r(T_2))| > \delta$ hay que implementarlo en la rutina del autómata.

5. CARACTERIZACIÓN TEMPORAL

En sistemas donde se dispone de una entrada de interrupción para el control, es evidente que la tarjeta presentada supone una gran mejora en el uso medio del autómata ya que nos permite ahorrarnos todo el tiempo usado por leer una entrada analógica y comprobar la lógica de generación de eventos en el caso de no ejecución del controlador.

Sin embargo, para ejemplificar la utilidad de este circuito, se ha considerado un ejemplo en el cual no se dispone de entrada de interrupción para mostrar que aun así la mejora en cuanto a la reducción del coste computacional respecto de un controlador con generación intrínseca (por programa) de

eventos es considerable.

Se dispone de una tarjeta BeagleBone Black, en la cual se ha implementado el control de un sistema con un control basado en eventos utilizando para la implementación el estándar industrial IEC-61499 [12]. Dicho estándar utiliza como medio de implementación el concepto de bloque de funciones (FB), teniendo cada tipo de FB una funcionalidad.

Se puede caracterizar el tiempo empleado por el CBE con generación interna de eventos en Q periodos de muestreo como el porcentaje de ejecuciones (η) por la suma del tiempo empleado por los bloques encargados de la lógica y el control más la probabilidad restante por el tiempo usado únicamente por la lógica de generación de eventos, como muestra la ecuación (2), donde t_{AIN} , t_{PWM} , t_{CLK} , t_{PID_e} y $t_{PID_{ne}}$, son los tiempos de lectura analógica, modificación de señal PWM, consulta del reloj, ejecución y no ejecución del controlador respectivamente.

$$t_{CBE_i} = Q(\eta(t_{AIN} + t_{PID_e} + t_{PWM} + t_{CLK}) + (1 - \eta)(t_{AIN} + t_{PID_{ne}} + t_{CLK})) \quad (2)$$

Mientras que utilizando la tarjeta propuesta se puede caracterizar el tiempo del CBE mediante la expresión (3), donde t_{DIn} es el tiempo empleado en leer una entrada digital.

$$t_{CBE_p} = Q(\eta(t_{AIN} + t_{PID_e} + t_{PWM} + t_{CLK} + t_{DIn}) + (1 - \eta)t_{DIn}) \quad (3)$$

Con las expresiones (2) y (3) se puede obtener el porcentaje de tiempo necesario por el CBE propuesto con respecto al CBE con generación interna, en función del porcentaje de generación de eventos, definiendo $\gamma = t_{CBE_p}/t_{CBE_i}$ se obtiene:

$$\frac{1}{\gamma} = \frac{\eta(t_{PID_e} + t_{PWM} - t_{PID_{ne}})}{\eta(t_{AIN} + t_{PID_e} + t_{PWM} + t_{CLK}) + t_{DIn}} + \frac{t_{AIN} + t_{PID_{ne}} + t_{CLK}}{\eta(t_{AIN} + t_{PID_e} + t_{PWM} + t_{CLK}) + t_{DIn}} \quad (4)$$

Para caracterizar numéricamente el tiempo usado por cada bloque empleado se ha realizado una serie de experiencias de medida para cada bloque, cuyos datos vienen recogidos en la Tabla 1.

Se ha usado la desviación media absoluta porque las experiencias contienen algunos datos atípicos con lo cual se evita el falseo de la varianza. Como

Tabla 1: Recopilación estadística de los tiempos de ejecución (en μs) de los bloques implicados.

	Min	Media	Mediana	t90 %	t95 %	DMA ¹
CLK	3.625	4.116	3.875	4.375	4.833	0.083
AIn	8.875	10.095	9.5	10.75	11.917	0.25
PID_e	14.375	16.004	15.041	16.917	18.875	0.25
PID_{ne}	7.833	9.025	8.334	9.834	11.25	0.209
PWM	5.583	6.713	6.125	7.375	8.542	0.166
DIn	4.5	5.2234	4.792	5.708	6.708	0.125
Medida	2.208	2.604	2.417	2.792	3.208	0.042

¹ Desviación Media Absoluta = mediana($|x_i - \text{mediana}(X)|$)

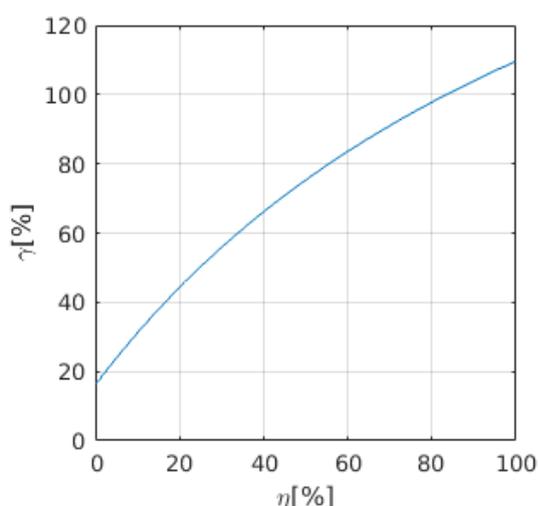


Figura 14: Tiempo usado por la alternativa propuesta con respecto al CBE con generación interna en función de η .

podemos ver en la Tabla 1, los datos están bastante concentrados en torno a la mediana, por lo tanto usaremos esta medida para caracterizar los bloques. Cabe decir que a los tiempos obtenidos hay que restarles el tiempo de Medida, por lo tanto se les restará su mediana.

A partir de la expresión obtenida se puede ver gráficamente en la Figura 14 que el uso del circuito propuesto en el CBE supone una mejora considerable, especialmente cuando la tasa de generación de eventos η es baja.

La expresión de γ tiende a aumentar a medida que el valor de η aumenta, llegando a ser el 100 %, es decir, cuando se usa el mismo tiempo el controlador con el CBE con generación interna que con la tarjeta electrónica, cuando $\eta = 83$ %.

Usando estos datos obtenidos para contextualizarlos con el controlador de Árzén, hay que decir que cuando este solo se ejecuta por tiempo máximo sin ejecución, el cual puede fijarse como cinco veces el periodo de muestreo, se corresponde con una tasa de generación de eventos $\eta = 20$ %, lo cual im-

plementado con la alternativa propuesta supone solo un 44,5 % del tiempo empleado con el CBE con generación interna, en cambio, con una generación de eventos del $\eta = 100$ % solo se empeora el tiempo consumido en un 9,5 %.

6. CONCLUSIONES

En este artículo se ha evaluado la implementación típica del control basado en eventos, el cual consta en la mayor parte de los casos y, en su controlador más importante, de una componente de muestreo periódico.

Con el fin de prescindir de este muestreo periódico, el cual contradice el principio del control basado en eventos de tratar de reducir el coste computacional medio requerido por el algoritmo de control, se ha planteado el desarrollo de una tarjeta electrónica diseñada para generar eventos según una lógica SSOD pero con el valor de histéresis reducido a la mitad.

Finalmente, se han realizado diferentes experiencias con algunas señales para observar el funcionamiento, por una parte cualitativamente con una señal sinusoidal, y con señales triangulares para observar los valores de conmutación (y por tanto de generación de eventos) del circuito diseñado.

Mediante un ejemplo de aplicación se realizó una caracterización temporal que ha permitido ilustrar el beneficio en cuanto a la reducción del coste computacional del uso de la propuesta de generación extrínseca de eventos respecto de la generación de eventos por programa (intrínseca al controlador).

La principal desventaja que tiene la propuesta presentada en este artículo es el uso de una entrada digital adicional por cada entrada analógica a la que se conecta una señal medida. En contraposición a esta desventaja, la propuesta permite implementar controladores basados en eventos con generación extrínseca en computadores industriales ampliamente usados en la industria como los PLC o micro-controladores, sin necesidad de intro-

ducir modificaciones tecnológicas significativas.

Agradecimientos

Este trabajo ha sido financiado mediante los proyectos TEC2015-69155-R del MINECO y P1-1B2015-42 de la Universitat Jaume I.

English summary

EXTRINSIC EVENT GENERATION FOR CONTINUOUS SYSTEM CONTROL

Abstract

In this paper a study about the implementation of event based controllers is presented, and the design of an electronic circuit that generates events outside of the controller is proposed. With this circuit, the periodic sampling of the error signal is avoided, contributing to reduce its associated computational cost, and, hence, approaching the implementation of event based controllers to its main principles.

Keywords: event based control, PID, A/D converter.

Referencias

- [1] Karl-Erik Arzén. A simple event-based pid controller. In *Proc. 14th IFAC World Congress*, volume 18, pages 423–428, 1999.
- [2] Manuel Beschi, Sebastián Dormido, José Sánchez, and Antonio Visioli. Characterization of symmetric send-on-delta pi controllers. *Journal of Process Control*, 22(10):1930–1945, 2012.
- [3] Manuel Beschi, Sebastián Dormido, José Sánchez, and Antonio Visioli. Tuning of symmetric send-on-delta proportional–integral controllers. *IET Control Theory & Applications*, 8(4):248–259, 2014.
- [4] Sebastián Dormido, J Sánchez, and Ernesto Kofman. Muestreo, control y comunicación basados en eventos. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5(1):5–26, 2008.
- [5] Sylvain Durand and Nicolas Marchand. An event-based pid controller with low computational cost. In *8th International Conference on Sampling Theory and Applications (SampTA'09)*, pages Special–session, 2009.
- [6] Sylvain Durand and Nicolas Marchand. Further results on event-based pid controller. In *Control Conference (ECC), 2009 European*, pages 1979–1984. IEEE, 2009.
- [7] Julio Ariel Romero Pérez and Roberto Sanchis Llopis. A new method for tuning pi controllers with symmetric send-on-delta sampling strategy. *ISA transactions*, 64:161–173, 2016.
- [8] Julio Ariel Romero and Roberto Sanchis. Analysis of a simple rule for tuning ssod based pids. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on*, pages 1–8. IEEE, 2016.
- [9] Julio Ariel Romero, Roberto Sanchis, and Ignacio Penarrocha. A simple rule for tuning event-based pid controllers with symmetric send-on-delta sampling strategy. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, pages 1–8. IEEE, 2014.
- [10] Volodymyr Vasyutynskyy and Klaus Kabitzsch. Time constraints in pid controls with send-on-delta. *IFAC Proceedings Volumes*, 42(3):48–55, 2009.
- [11] Volodymyr Vasyutynskyy and Klaus Kabitzsch. A comparative study of pid control algorithms adapted to send-on-delta sampling. In *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, pages 3373–3379. IEEE, 2010.
- [12] Alois Zoitl and Robert Lewis. *Modelling control systems using IEC 61499*, volume 95. IET, 2014.



© 2018 by the authors.
Submitted for possible
open access publication
under the terms and conditions of the Creative Commons Attribution CC-BY-NC 3.0 license (<http://creativecommons.org/licenses/by-nc/3.0/>).