

CONTROL NEUROMÓRFICO DEL BRAZO ROBÓTICO BIOROB DEL CITEC DE LA UNIVERSIDAD DE BIELEFELD

Linares Barranco, Alejandro*; Jiménez Fernández, Angel F.; Jiménez Moreno,
Gabriel, Gutiérrez Galán, Daniel, Ríos Navarro, José Antonio

Robótica y Tecnología de Computadores.

Departamento de Arquitectura y Tecnología de Computadores. Escuela
Politécnica Superior. Universidad de Sevilla.

*E-mail: alinares@atc.us.es

RESUMEN

Los sistemas neuronales biológicos responden a estímulos de una forma rápida y eficiente en el movimiento motor del cuerpo, comparado con los sistemas robóticos clásicos, los cuales requieren una capacidad de computación mucho más elevada. Una de las claves de estos sistemas es la codificación de la información en el dominio pulsante. Las neuronas se comunican por eventos con pequeños pulsos de corrientes producidas por intercambio de iones entre las dendritas y los axones de las mismas. La configuración en redes de neuronas permite no sólo el procesado de la información sensorial y su procesamiento en el dominio pulsante, sino también la propia actuación sobre los músculos en el formato pulsante. Este trabajo presenta la aplicación de un modelo de control motor basado en el procesado de pulsos, incluyendo la propia actuación sobre motores en el contexto de los pulsos. Se ha desarrollado un sistema de control en lazo cerrado por pulsos, denominado spike-based PID controller para FPGA, el cual se ha integrado en el esqueleto de un robot bioinspirado, BioRob X5 del CITEC de la Universidad de Bielefeld, para su uso en el desarrollo de modelos bioinspirados. El Robot, de más de 1m de largo, permite controlar las posiciones de las articulaciones usando control por pulsos y con un consumo menor de 1A para todos los grados de libertad funcionando al mismo tiempo.

PALABRAS CLAVE

Procesado por eventos, Modelado neuronal, Robótica, Actuación motora pulsante, FPGA.

ABSTRACT

Compared to classic robotics, biological nervous systems respond to stimulus in a fast and efficient way regarding to the body motor movement. Classic robotic systems usually require higher computational capacity. One of the main keys of biological systems respect to robotic machines is the way the information is coded and transmitted. They use spikes. A neuron is the “basic” element that form biological nervous systems. Neurons communicate in an event-driven way through small current pulses (spikes) produced when ions are interchanged between dendrites and axons of different neurons. When neurons are arranged in networks, they allow not only the sensory information processing, but they also allow the actuation over the muscles in a spiking way. This paper presents the application of a motor control model based on spike processing, including the motor actuation in the spike domain. A close-loop control system, called spike-PID controller, has been developed for FPGA. This controller has been embedded into a bioinspired robot, called BioRob X5, at CITEC of the University of Bielefeld during a “Salvador de Madariaga” grant for a research visit in the July-September 2018 term. The robot, longer than 1 meter tall, allows the joint position control through spiking signals with a power consumption below 1A for the 4 DoF working at the same time.

KEYWORDS

Event-based processing, Spiking systems, Neuromorphic engineering, Neuronal modeling, Robotic, Spiking motor actuation, FPGA.

INTRODUCCIÓN Y OBJETIVOS

Este trabajo se enmarca en el campo de la ingeniería neuromórfica, la cual busca inspiración en la biología, y concretamente en el sistema nervioso central, para resolver problemas de ingeniería imitando la forma en que se resuelven tan eficientemente en un cerebro. La implementación hardware de dispositivos neuromórficos requiere una base tecnológica que se elige para cumplir los requisitos de la aplicación deseada. Las bases tecnológicas más comunes son los diseños de chips “full custom” (ASIC) [1], las FPGAs (Field Programmable Gate Arrays) [2] o los FPAAs (Field Programmable Analog Arrays) [3]. El hardware neuromórfico desarrollado en la última década se puede clasificar en dos clases principales: los sensores y las redes neuronales. En el campo de los sensores, los ejemplos más representativos son los sensores de visión [4][5], de audición [6][7] y de olfato [8]. Los últimos procesadores neuronales analógicos son: ROLLS (procesador neuromórfico pulsante con aprendizaje reconfigurable on line) [9], Neurogrid [10], los dispositivos desarrollados en el HBP (Human Brain Project) basados en resultados de los proyectos EU BrainScaleS [11] o SpiNNaker [12], o los procesadores pulsantes digitales como Loihi de Intel [13] o Neurovision; o el chip de IBM TrueNorth que es un chip híbrido (analógico y digital) [14].

Para integrar e implementar arrays de neuronas pulsantes y arquitecturas neuromórficas complejas, una estrategia de comunicación para el envío de eventos es necesaria. Para ello se desarrolló el protocolo AER (Address-Event-Representation) [15], y hoy día es un estándar en la comunidad neuromórfica. AEs (Address Events) son eventos digitales con una etiqueta asociada (dirección). El tiempo se representa a sí mismo y las etiquetas asociadas identifican la fuente del evento (neurona que emite el impulso). El uso de ROMs permite el mapeo de eventos hacia diferentes destinos, habilitando así la construcción de topologías de redes arbitrarias.

La robótica actual no está adaptada o no ofrece soluciones específicas para los sistemas neuromórficos, ya que los productos en el mercado incluyen controladores de motores en cada articulación del robot, siendo normal encontrar dispositivos de comunicación entre los grados de libertad del robot y la controladora (por ejemplo, usando el bus CAN), lo cual no permite ni llegar, ni conocer, las señales que se usan para el control directo del motor. En este trabajo se presenta un robot que se ha adquirido sin controladora alguna. Está compuesto por 4 grados de libertad que se manejan con 4 motores de corriente continua (DC motors) y que incluye un *encoder* óptico por articulación para informar sobre el estado actual de cada articulación.

Mediante el uso de tarjetas de circuito impreso basadas en FPGAs y microcontroladores se ha implementado e implantado un sistema de control motor pulsante para los 4 DC motors del robot usando controladores PID en el dominio pulsante. Así mismo se ha desarrollado el software necesario para comandar al robot desde un portátil. El controlador se ha caracterizado. Y, por último, se ha dotado al controlador de una interfaz hacia sistemas pulsantes implementados en la plataforma Dynap-SE.

METODOLOGÍA

En esta sección se procede a describir las características del robot utilizado en este trabajo, posteriormente se expone un resumen del controlador PID basado en impulsos que se integra en el robot, y por último, se expone la arquitectura de la plataforma final del robot con la integración del controlador neuroinspirado.

El brazo robótico BioRob

El robot BioRob [16] está basado en el concepto de actuación elástica y antagonista que se inspira en el aparato elástico músculo-tendón (ver Figura 1). Cada articulación dispone de un motor DC acoplado mediante cuatro cables o cuerdas que contienen muelles y otros componentes elásticos. De esta forma cada articulación posee características de resortes progresivos y no lineales. Adicionalmente, las articulaciones están dotadas de dos sensores: un sensor de posición del motor DC de tipo *encoder* óptico, y un sensor de posición angular de la articulación. El diseño de la actuación está descrito en detalle en [17] [18].

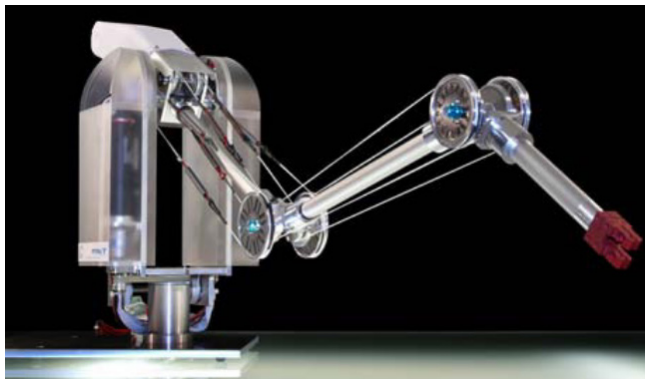


Figura 1. Brazo robótico BioRob X4.

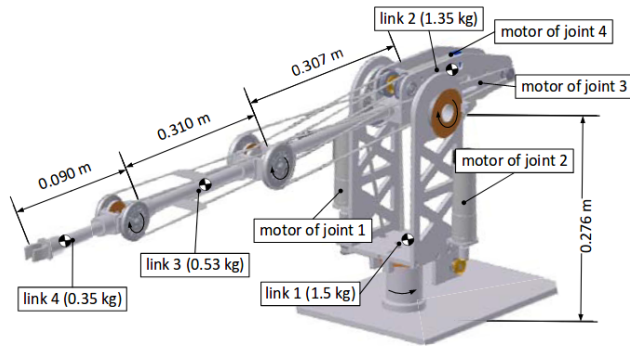


Figura 2. Diagrama de arquitectura del BioRob: articulaciones y motores.

La Figura 2 indica la ubicación de cada motor y cada sensor en el brazo robótico. Los *encoders* de los motores y los sensores de posición de las articulaciones aseguran una resolución cartesiana por debajo de 1mm. Los sensores de posición de las articulaciones permiten la medida de las fuerzas elásticas de las articulaciones mediante el uso de las curvas características de la rigidez de la articulación. Esto habilita el control de fuerzas, la detección de colisiones y la reacción ante estos efectos.

El control motor pulsante

El controlador PID pulsante de motor es un controlador en lazo cerrado PID totalmente diseñado en el dominio pulsante. Tanto su señal de referencia de entrada, la señal de salida hacia el motor, así como la señal de sensado del motor para el cálculo del error, son señales codificadas en pulsos. Se utiliza la modulación por frecuencia de pulsos (PFM) incluso para atacar a las bornas de los motores. De esta manera se permitiría que cualquier sistema neuro-inspirado pueda producir una señal que sea directamente usada para mover un motor a una velocidad determinada o a una posición concreta.

Un controlador motor PID clásico se compone de tres bloques que ajustan la proporcionalidad, la integración y la deriva del error del sistema de control en lazo cerrado. Para implementar en el dominio pulsante un controlador PID se requiere del uso de una serie de elementos de procesamiento pulsante que nos permitan realizar las mismas funciones. Para ello se han diseñado e implementado una serie de elementos de procesamiento pulsante cuya formulación se realiza en el dominio S. Los bloques básicos son:

- **Spike-Generator o Generador de impulsos:** se utiliza para convertir una señal digital en una secuencia de pulsos con una frecuencia determinada.
- **Integrate & Generate (I&G) o Integrador y Generador de pulsos:** Este bloque se compone de dos partes: la primera es un contador de pulsos de entrada, y la segunda es un generador de pulsos (descrito en el epígrafe anterior). De esta forma, y teniendo en cuenta los pulsos de polaridad positiva y negativa, la señal pulsante de salida de este bloque está representando la integral de la señal de entrada. En el dominio S, este bloque se representa con una función de transferencia como la siguiente:

$$out = \frac{count}{2^{NBITS}-1} \times \frac{f_{CLK}}{genFreqDiv}$$
$$out = \int in dt \times \frac{f_{CLK}}{2^{NBITS}-1 \cdot genFreqDiv}$$
$$F(s) = \frac{Ki}{s}$$

Donde *out* es la frecuencia en pulsos de la señal de salida, *count* es el valor de contador de pulsos de entrada, *in* es la frecuencia de la señal de entrada, N_{BITS} es la resolución en bits del generador de pulsos, *genFreqDiv* es el divisor del reloj del generador de pulsos y f_{CLK} su frecuencia de reloj.

- **Hold & Fire (H&F) o retén y dispara:** Este bloque permite la fusión de dos señales pulsantes en una nueva señal que representa la diferencia de ambas o la suma de ambas, de una forma intuitiva y sencilla. Teniendo en cuenta las polaridades de los impulsos y el origen de la señal, se usa una máquina de estados para, una vez llegado un pulso por una de las entradas, esperar un próximo pulso y, teniendo en cuenta las polaridades de ambos, así como la operación a realizar (por ejemplo, restar señales pulsantes o sumarlas), se cancelarán los dos pulsos o se mandará el primero que se recibió para retener el último en llegar.
- **Spike Expanse block o Expansor de pulsos:** Debido a que los pulsos de los sistemas neuromórficos son muy estrechos, este bloque se usa para expandir la anchura de los pulsos antes de ser enviados a las bornas de los motores. En este sentido, este bloque permite aplicar un factor multiplicativo o proporcional a la señal de salida de un controlador.
- **Spike Derivative block o derivador de pulsos:** Mediante el uso combinado de un I&G y un H&F, tal como muestra la Figura 3, se puede obtener la función de transferencia en el dominio S de un derivador para su uso en un controlador PID.

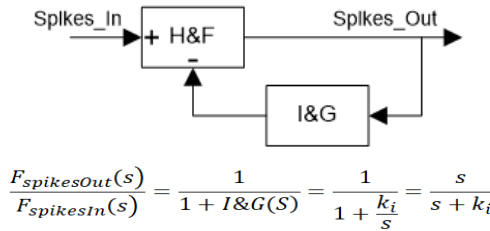


Figura 3. Diagrama del derivador y formulación en S.

- Spike PID:** La Figure 4 muestra el diagram de bloques de un controlador PID clásico, así como la solución adoptada para su implementación en el dominio pulsante. Se incluye la formulación en el dominio S. X(s) representa la señal enviada al motor, mientras que el estado actual del motor se representa por Y(s), la cual es enviada al controlador una vez medida por el sensor que cierra el lazo, y se representa como FB(s). La función de transferencia del controlador PID es TF(s) y la señal de referencia es U(s). En la solución pulsante, los bloques de integración y derivación trabajan sobre la señal de error del sistema (referencia menos realimentación), mientras que la proporcionalidad se aplica al expandir los pulsos.

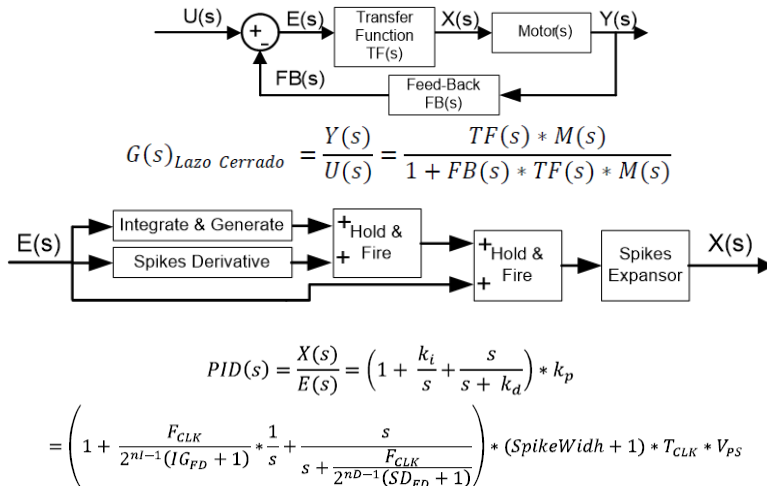


Figura 4. Arriba: controlador PID clásico y formulación. Abajo: Solución pulsante.

Este controlador PID permite controlar en velocidad a un motor. Para poder aplicarlo al robot del CITEC se debe añadir un bloque I&G al sensor que mide el estado actual del motor para poder tener control sobre la posición actual del motor, que reflejará la posición actual de la articulación del robot. En la Figura 5 se muestra un diagrama de bloques que representa la solución implementada para el control del robot de cuatro articulaciones del CITEC.

Se han utilizado dos plataformas basadas en FPGA con la intención de mantener en la Spartan-3 la expansión de los pulsos y la adaptación de los sensores al dominio pulsante, lo cual es necesario para cualquier controlador pulsante que se realice. Por otro lado, en la Spartan-6, de mayor capacidad, se ha implementado el controlador pulsante PID en un porcentaje reducido de sus recursos, con la intención de permitir que la plataforma final permita probar e implementar nuevos controladores neuro-inspirados en este robot.

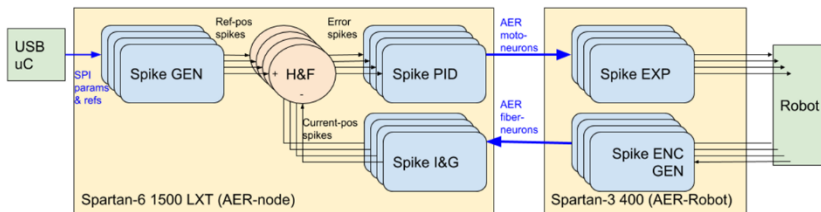


Figura 5. Diagrama de bloques del controlador pulsante PID del brazo robótico BioRob del CITEC.

La infraestructura desarrollada

La Figura 6 muestra el Sistema completo, su configuración y una fotografía. La tarjeta AER-Robot es capaz de alimentar hasta 4 motores DC con señales de potencia que se manejan desde una FPGA Spartan-3. Además, permite la lectura de los sensores. Se han usado dos buses paralelos AER para comunicar esta tarjeta con la segunda, la AER-node, que implementa los controladores pulsantes para todas las articulaciones. Se usa un bus SPI para enviar la configuración desde la placa AER-Robot hacia la AER-node. Esta configuración llega al sistema a través del bus USB que conecta el computador con la tarjeta AER-Robot.

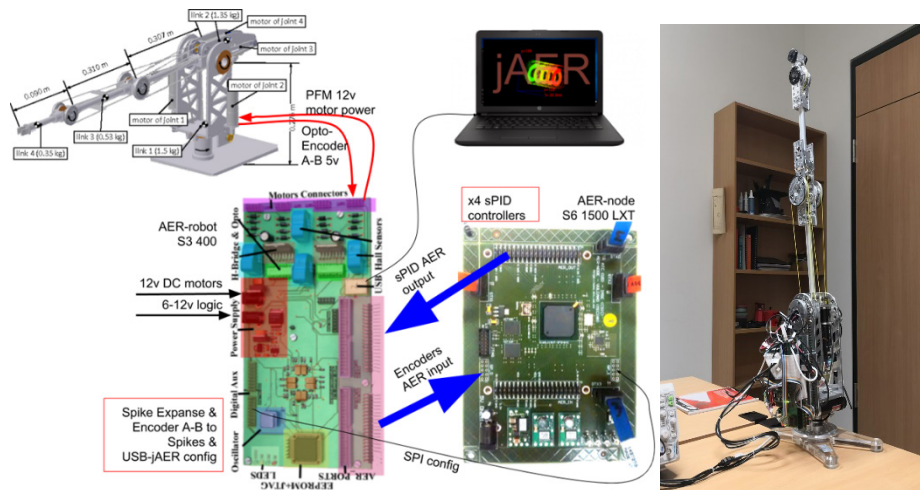


Figura 6. Configuración HW de la plataforma BioRob para control pulsante y fotografía del robot con el controlador (derecha).

- **AER-Robot:** esta tarjeta ha sido desarrollada en el grupo RTC durante el proyecto SAMANTA-2 (2008) para controlar una mano robótica antropomórfica. Se centra en una FPGA Spartan-3 y un microcontrolador 8051 con conexión USB. Dispone de puentes de potencia opto-acoplados para controlar hasta 4 motores DC. En este escenario la FPGA está recibiendo impulsos en formato AER para poder controlar los motores. Se usan en total 8 direcciones para codificar los dos posibles sentidos de cada uno de los motores. Estas direcciones son decodificadas y se generan pulsos expandidos que se envían directamente a los motores a través de puentes de potencia. Al mismo tiempo, señales de los *encoders* anexados a los motores se usan, gracias a parejas desfasadas, para detectar el sentido de giro del motor y su velocidad. Estas señales se convierten a pulsos y se envían en formato AER a la AER-node. La longitud de expansión de los impulsos es configurada a través de comandos recibidos por el microcontrolador 8051, que se conecta al software del PC mediante USB. El software del PC usado es el jAER, de libre distribución, al que se le ha añadido un filtro para controlar el robot. Estos comandos de configuración se envían a la FPGA de la placa AER-Robot y también a la FPGA de la placa AER-node a través del bus SPI.
- **AER-Node:** Esta segunda tarjeta dispone de una FPGA de mayor capacidad, la Spartan-6, para poder albergar holgadamente los cuatro controladores pulsantes PID para fijar las articulaciones en una posición concreta. Además, esta tarjeta deja abierta la posibilidad de implantar nuevos modelos de controladores motores pulsantes, como puede ser el SVITE [20] u otros por

llegar. El bus de entrada AER trae información sensorial del robot en formato pulsante, proveniente de la AER-Robot, la cual se usa para cerrar el lazo de control. Existe un bus SPI para configurar los controladores. Este bus es el mismo que se tenía en la AER-Robot. Gracias al formato de 16-bits dirección más dato, cada bloque del controlador recibe de forma distribuida sus parámetros. Por último, la salida pulsante de los controladores se envía en formato AER a la tarjeta AER-Robot para su tratamiento y envío a los motores.

RESULTADOS Y DISCUSIÓN

Una vez montado, programado y configurado tanto el robot como el controlador pulsante expuesto anteriormente se procedió a la realización de varias pruebas para la validación del sistema para su uso en los proyectos del CITEC. La validación fue doble: en primer lugar, se programó el filtro desarrollado en el jAER, para que se pudiera comandar al robot una secuencia de posiciones con unos tiempos de espera entre cada una y se realizó una medida del consumo de la lógica y los motores, no llegando a 1A al tener en funcionamiento los 4 motores. Este bajo consumo se ha conseguido gracias al diseño del robot, que está fabricado en aluminio y por tanto es de bajo peso, pero también a los controladores pulsantes, los cuales reducen considerablemente el gasto energético por adaptarse la cantidad de pulsos que envían energía a los motores al error de posición de los mismos. Actualmente se están realizando trabajos para poder comparar cuantitativamente los controladores pulsantes con los tradicionales en este robot.

Por otro lado, ya que el objetivo del trabajo es poder disponer de una actuación robótica manejable desde un sistema de control neuroinspirado que emule las funcionalidades del cerebro, se trabajó en la integración del controlador con la plataforma Dynap-SE. Esta plataforma está compuesta por cuatro chips que integran numerosas neuronas de silicio distribuidas en cuatro núcleos (cores) en cada chip. Una interfaz con Linux permite programar las conexiones entre las neuronas y sus pesos para poder implementar redes neuronales pulsantes de la complejidad que se desee hasta 1M de neuronas. La plataforma cuenta con una interfaz gráfica que permite activar conjuntos de neuronas para que produzcan una actividad en la salida regulable. En esta prueba se ha aprovechado el mismo *core* de los 4 chips para unir poblaciones de neuronas suficientes para generar actividad de salida que pueda usarse como señal de referencia para las cuatro articulaciones del robot. En la Figura 7 se muestran fotografías de la articulación 4ª en diferentes grados como resultado de la actividad de las poblaciones de neuronas del mismo *core* de cada chip, dejando el resto de *cores* inactivos para este experimento y por tanto el resto de las articulaciones

en su posición de *home*. El experimento fue repetido numerosas veces sin observar una deriva en la posición alcanzada. Se tiene como trabajo futuro la caracterización del posible error de deriva tras realizar el experimento repetidamente.

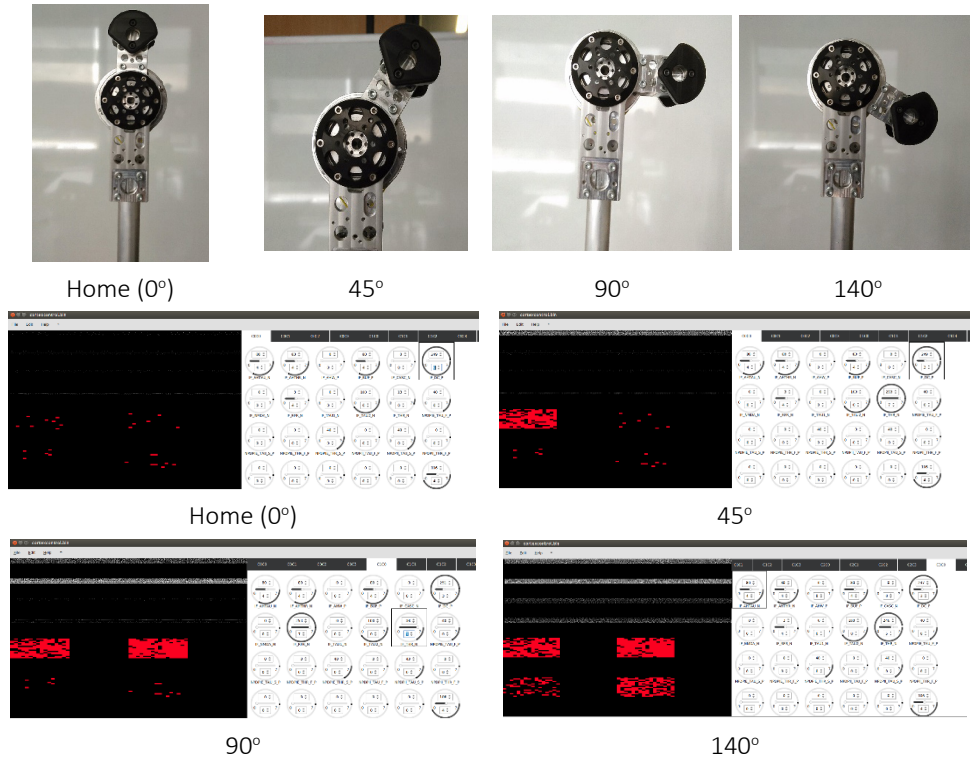


Figura 7. Activaciones de neuronas de 1 núcleo de 4 chips del Dynapse para generar impulsos de referencia al controlador pulsante para 4 posiciones de la articulación 4.

CONCLUSIONES

Este trabajo describe la integración de controladores pulsantes de motores DC para un brazo robótico de la Universidad de Bielefeld utilizando tarjetas basadas en FPGA diseñadas por el grupo RTC (TEP-108) para el procesamiento y control neuroinspirado de robots. Además, se han adaptado los circuitos de las FPGAs para recibir la señal de referencia pulsante de un sistema de redes neuronales pulsantes implementado en la plataforma Dynap-SE, diseñada en la Universidad de Zurich y comercializada por la empresa AI-Cortex.

AGRADECIMIENTOS

Este trabajo ha sido desarrollado gracias a la financiación dentro del programa de movilidad para investigadores Senior Salvador de Madariaga del Ministerio de Educación y Ciencia del Gobierno de España. Las tareas realizadas se encuadran dentro del trabajo del Proyecto COFNET (TEC2016-77785-P) del plan nacional de I+D del Ministerio de Educación y Ciencia del Gobierno de España con el soporte de los Fondos FEDER de la Unión Europea.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Chicca, E., *et al.* (2014). Neuromorphic electronic circuits for building autonomous cognitive systems. *Proceedings of the IEEE*, pp. 1–22.
- [2] Maguire, L. P., *et al.* (2007). Challenges for large-scale implementations of spiking neural networks on FPGAs. *Neurocomputing*, 71, pp. 13–29.
- [3] Rocke, P., *et al.* (2008). *Evolvable Systems: From Biology to Hardware* (Eds. Hornby, G., *et al.*) pp. 118–129.
- [4] Lichtsteiner, P., *et al.* (2008). A 128x128 120 dB 15 us latency asynchronous temporal contrast vision sensor. *Solid-State Circuits. IEEE Journal*, 43, pp. 566–576.
- [5] Serrano-Gotarredona, T. y Linares-Barranco, B. (2013). A 128x128 1.5 Contrast Sensitivity 0.9 FPN 3 us Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers. *Solid-State Circuits. IEEE Journal*, 48, pp. 827–838
- [6] Chan, V., *et al.* (2007). AER EAR: A matched silicon cochlea pair with address event representation interface. *Circuits and Systems, I: Regular Papers. IEEE Transactions*, 54, pp. 48–59.
- [7] Jiménez-Fernández, A., *et al.* (2017). A Binaural Neuromorphic Auditory Sensor for FPGA: A Spike Signal Processing Approach. *IEEE Transactions on Neural Networks and Learning System*, s28(4), pp. 804–818.
- [8] Koickal, T. J., *et al.* (2007). Analog VLSI circuit implementation of an adaptive neuromorphic olfaction chip. *Circuits and Systems, I: Regular Papers. IEEE Transactions*, 54, pp. 60–73.
- [9] Qiao, N., *et al.* (2015). A Re-configurable On-line Learning Spiking Neuromorphic Processor comprising 256 neurons and 128K synapses. *Frontiers in Neuroscience*, 9.

- [10] Benjamin, B. V., *et al.* (2014). Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations. *Proceedings of the IEEE*, 102, pp. 699–716,
- [11] Calimera, A., *et al.* (2013). The human brain project and neuromorphic computing. *Funct Neurol* 28, pp. 191– 196
- [12] Furber, S.B., *et al.* (2014). The SpiNNaker Project. *Proceedings of the IEEE* 102, pp. 652–665.
- [13] Davies, M., *et al.* (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1), pp. 82-99.
- [14] Cassidy, A.S., *et al.* (2013). Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. En *International Joint Conference on Neural Networks (IJCNN)*. *IEEE*.
- [15] Sivilotti, M.A. (1990). Wiring considerations in analog VLSI systems, with application to field programmable networks. PhD thesis (California Institute of Technology, July 1990)
- [16] Lens, T., Kunz, J., von Stryk, O., Trommer, C. y Karguth, A. (2010). BioRob-Arm: A Quickly Deployable and Intrinsically Safe, Light- Weight Robot Arm for Service Robotics Applications. *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, Munich, Germany, pp. 1-6.
- [17] Klug, S., Lens, T., von Stryk O., Möhl, B., y Karguth, A. (2008). Biologically inspired robot manipulator for new applications in automation engineering. En *Proceedings of Robotik 2008, ser. VDI-Berichte, no. 2012. Munich, Germany: VDI Wissensforum GmbH, June 11-12.*
- [18] Möhl, B. (2000). Bionic robot arm with compliant actuators. En *Proceedings of SPIE - Sensor Fusion and Decentralized Control in Robotic Systems III, vol. 4196*, pp. 82–85.
- [19] Jimenez-Fernandez, A., Jimenez-Moreno, G., Linares-Barranco, A., Dominguez-Morales, M.J., Paz-Vicente, R. y Civit-Balcells, A. (2012). A Neuro-Inspired Spike-Based PID Motor Controller for Multi-Motor Robots with Low Cost FPGAs. *Sensors*, 12, pp. 3831-3856. Recuperado de: <http://www.mdpi.com/1424-8220/12/4/3831>

Linares Barranco, A.; Jiménez Fernández, A. F.; Jiménez Moreno, G.; Gutiérrez Galán, D.; Ríos Navarro, J. A.

- [20] Perez-Peña, F., Morgado-Estevez, A., Linares-Barranco, A., Jimenez-Fernandez, A., Gomez-Rodriguez, F., Jimenez-Moreno, G., y Lopez-Coronado, J. (2013). Neuro-Inspired Spike-Based Motion: From Dynamic Vision Sensor to Robot Motor Open-Loop Control through Spike-VITE. *Sensors*, 13, pp. 15805-15832.