# Spike-based VITE control with Dynamic Vision Sensor applied to an Arm Robot

Fernando Perez-Peña[1], Arturo Morgado-Estevez[1], Teresa Serrano-Gotarredona[2]
[2] Institute of Microelectronics, Sevilla, SPAIN
[1] Applied Robotics Lab, School of Engineering
University of Cadiz, Spain
fernandoperez.pena@uca.es

F. Gómez-Rodríguez, V. Ferrer-García, A. Jimenez-Fernandez, A. Linares-Barranco
Robotic and Technology of Computers Lab
School of Computer Engineering
University of Seville, Spain

***Abstract*— Spike-based motor control is very important in the field of robotics and also for the neuromorphic engineering community to bridge the gap between sensing / processing devices and motor control without losing the spike philosophy that enhances speed response and reduces power consumption. This paper shows an accurate neuro-inspired spike-based system composed of a DVS retina, a visual processing system that detects and tracks objects, and a SVITE motor control, where everything follows the spike-based philosophy. The control system is a spike version of the neuroinspired open loop VITE control algorithm implemented in a couple of FPGA boards: the first one runs the algorithm and the second one drives the motors with spikes. The robotic platform is a low cost arm with four degrees of freedom.**

## I. INTRODUCTION

Nowadays, it is possible to integrate several thousands of artificial neurons into the same electronic device – VLSI (Very-large-scale integration) chip [1-2] or FPGA (Field-programmable gate array) [3] - with the intention of emulating brain like behavior into hardware and apply that to robotic systems. They are called neuromorphic devices. The information flow in these devices consists of spikes or graduated potentials like those used by neurons to carry information across the organism.

On the one hand VLSI chips were first developed by Caver Mead [4] in the late eighties, to mimic the behavior of neuronal systems. This was a full analog technology. The main features of these full-custom chips are noise due to thermal fluctuations, high speed/bandwidth usage and computation with continuous values as information. Many works can be shown to proof the advances in this field. We will focus our attention on those applied in robotics.

We have neuro-inspired vision sensors [5] where a matrix of analog pixels mimic part of the human retina and, recently, some applications by joining these devices with VLSI chips [6]. The field of VLSI neuron is largely studied; in [7] silicon neuron blocks such as synapse, low pass filters, spikes generators and integrate and fire neurons are shown. However, there is an important field to investigate where large-scale neuromorphic circuits and systems designs will increasingly combine full-custom analog and synthesized digital designs [7] to improve the results.

On the other hand, using digital components such as FPGAs to build up complete control architectures provides stability, fast development times, low noise and high precision properties. Although a clock is inherent to digital designs, the information is in the firing rate which can be an accurate analog approach. Regarding the field of robotics, once the first approaches to bio inspired image sensors appeared a few years ago (early 2000s), the race to make a complete system began. Until those days, there were some advances in describing neuro-inspired control algorithms: in [8] a couple of them were shown: one to generate non-planned trajectories (VITE – Vector Integration To-End Point); and the other one to follow them in muscles (FLETE – Factorization of LEngth and TEnsion). Then, many related works using them were published [9-10].

We have designed, implemented and tested on a single motor of a robotic platform a neuro-inspired controller based on a translated version of VITE algorithm, the Spike VITE [11]. With this new algorithm, we can now generate the trajectory to be followed by the robot with spike processing blocks on a multi-motor robot. This algorithm will feed the actuation layer using Address-Event-Representation (AER).

By using AER protocol, all neurons are continuously sending information about their excitation level to the central system and it could be processed in real time by a higher layer. AER is based on the concept which mimics the structure and information coding of the brain. Thus, AER let us process the information in real time, which is an important point in motor control. This is one of the reasons for using it: the provided speed. Another reason is the scalability that allows it by parallel connections.

In the next section, the vision processing system is described. Then, the third section shows the algorithm. In section four we describe the layers of the control system. Then, the methodology and results accomplished are shown. Finally, to sum up, the conclusions about the results achieved are exposed.

## II. VISION PROCESSING SYSTEM

One of the most important advantages of DVS (Dynamic Vision Sensor) is the response time from a luminosity change in the photoreceptor to its signaling on the output of the sensor. This sensor has a latency of 3µs for each of the 128x128 pixels, a dynamic range of 120dB, a 1.5% contrast sensitivity, a power consumption bellow 4mW, and a FPN of 2 (Fixed Pattern Noise down to 0.9%) [12].

The vision processing system is completed with a visual processing system consisting of several cells interconnected, like neurons that process the AER sequences [13]; the goal of this system is to detect, track and recognize moving objects in the scene. In this approach every cell tries to track one object obtaining its position and estimating its velocity. A new architecture is used: instead of using a classical fully parallel cell architecture, we use cascade architecture. Therefore, cells of one layer are distributed sequentially: the first cell receives the complete AER sequence and extracts, retains and processes several events; which and how many events are extracted depend on the application; non-used events are resent to the second cell in the layer. This procedure is repeated in the rest of the cells in a layer and in the rest of the layers Fig 1.a shows this idea.

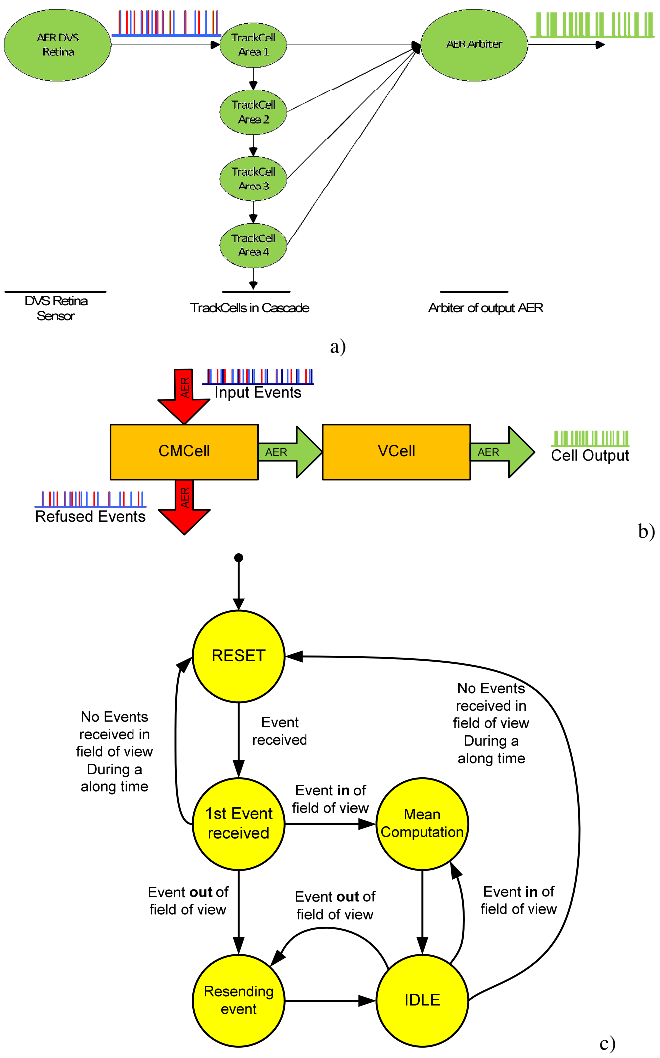


**Fig. 1.** a) Cascade Architecture, b) CMCell+VCell and c) CMCell state machine diagram

The most important advantage of this architecture is that each cell only processes the extracted events. Furthermore, this scheme presents an implicit inhibition mechanism, that is, the events processed by one cell are eliminated from the AER sequence and do not interfere with the computation. This scheme allows reducing the AER sequence complexity from one cell to the next. In this system all cells use AER for information transmission, so each cell has 3 AER ports: one AER input and two AER outputs: the first output gives the result of events computation and the second one resends the refused events (see Fig. 1.b). We propose an object tracking procedure using the cascade architecture presented above. We will only use one layer, where each cell, called *TrackCell*, consists of two sub-cells: one is devoted to object's detection and position determination, called *CMCell*, and the other, called *VCell*, is devoted to object's velocity estimation. Fig. 1.a shows these cells.

From now on, we suppose that the input is the visual information provided by the silicon retina, so events correspond to the movement in the scene. The first cell in the layer receives all the events. Just after the first event is received, the *CMCell* only extracts and computes events from a small area around this first event (area of interest), resending the rest of the events to the second cell. If during a period of time the *CMCell* does not receive enough events, typically 10 events, it will reset. On the contrary, if during this period of time the *CMCell* receives enough events from the area of interest, it computes the object position as the mean value between the last positive event location and the last negative event location. After the object position computation, the *CMCell* moves the center of the area of interest to the object position. This procedure is repeated after each event is received. If the *CMCell* receives events near the area of interest (a few pixels around) it will change the area size allowing adapting it to the object size. Fig. 1.c shows the *CMCell* state machine diagram.

The output of the *CMCell* consists of an AER sequence that encodes the object position. This information can be used by the *VCell* to estimate the object's velocity. The *VCell* takes the object position periodically and computes the velocity as the mean value of the last two object positions. Initially, the period used is 100ms, but it changes dynamically depending on the velocity computed. If the velocity is high the period will be reduced. On the contrary, if the velocity is low the period will be increased. The velocity is also transmitted using AER.

With this scheme it is possible to track as many objects as *TrackCells* can be synthesized in a FPGA in cascade. The key point of this procedure is that events are processed as soon as they are received, without frame integration. Therefore the response time of each cell is very short; in fact it is the delay time (ns).

## III. SVITE ALGORITHM

The original VITE algorithm [8] is used for calculating a non-planned trajectory. It computes the difference between the target and the present position. It models planned human arm movements. In contrast to approaches which require the stipulation of the desired individual joint positions, this trajectory generator operates with desired coordinates of the end vector and generates the individual joint driving functions in real-time employing geometric constraints which characterize the manipulator.

In Fig. 2 the block diagram of the algorithm and the translation into spikes processing blocks are shown.

The translation into spike-processing blocks is done by solving the equations using Laplace transform to build up a system under frequency domain. As we consider the firing rate as the information of our neural code, this method of using Laplace transform allow us to supposedly accept the match between both concepts: firing rate and Laplace frequency.

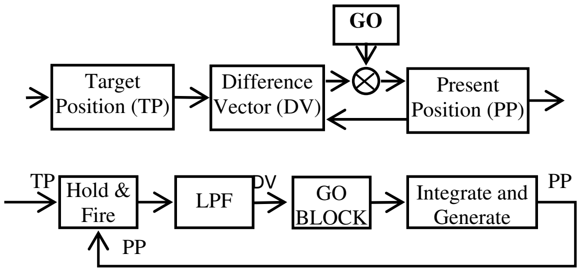


**Fig. 2.** Up. Block diagram of the VITE algorithm. Down. Block diagram of the SVITE generated from existing spikes processing blocks.

The neuro-controller running in the FPGA is composed of four different types of spikes processing blocks:

- *Hold and Fire (H&F):* this block performs the addition or subtraction of spike flows to compute the error signal. The task of this block can be matched with a neuron synapse [14]. This block has two inputs: one excitatory coming from the visual processing layer and one inhibitory from the end-block of the algorithm.
- *Spikes low pass filter (LPF):* the behavior of the block is the same as an analog classical low pass filter but it operates with the spike's input firing rate. The result of this block is a uniform distribution of the spikes input [14]. There are two filter blocks in the SVITE algorithm: one at the H&F's output and another one included in the GO Block.
- *GO Block:* the main function of this block is to control the speed of the movement and also to be the gate of it. It is done by modifying the input firing rate. We inject spikes according to a user parameter which define the speed desired. The behavior of this block can be matched with an excitatory neuron.
- *Integrate and Generate:* this block is the analogous of the Integrate-and-Fire in VLSI designs but it is made by digital components [14].

From a biological point of view, this algorithm conforms something similar to a forward model and evaluates the corollary discharge with the Integrate and Generate block. So, no sensory discrepancies are noticed within this algorithm as it was expected without feedback from the robot. The assumption is that the commanded position is reached.

From a classic control theory, this algorithm cannot be exactly matched with any of the traditional controllers such as Proportional, Derivative or Integral. If we consider the GO block as a disturbance and the Integrate and Generate as the robot, the system could be matched with a pseudo-proportional control.

## IV. SPIKE-PROCESSING MOTOR SYSTEM

The DVS AER retina delivered a continuous events flow to the system. Applying a cascade architecture processing layer to these events flow, the output of a *CMCell* meets the center of an object. Therefore, each output event of this cell plays the role of the target robot position. So, the vision processing system will deliver the reaching position to the control system.

The first layer is the processing one centered on *SVITE* algorithm. It receives the target position from the visual processing layer. The algorithm can be replicated in many units as degrees of freedom the robot has. In this paper, we have four algorithms calibrated for each movement; having different algorithms for each joint allows an individual speed control to achieve synchronized movements. The interface with the next layer is managed by the AER protocol. There are four different addresses transmitted: one for each motor of the robot.

The last layer is the actuation one where the commanded position is received and adapted to feed the motors of the robot. We propose to use PFM (Pulse Frequency Modulation) to drive the motors because it is intrinsically a spike-based solution almost identical to the solution that animals and humans use in their nervous systems for controlling the muscles. PFM modulation uses the firing rate to run the motors, so there is no transformation to do; the spikes are delivered in real time to the motors with a little expansion to avoid the spike filtering by the motor.

## V. METHODOLOGY

The setup to check the neuromophic architecture (Fig. 3) includes the DVS retina, all the layers: the AER_Node board to implement both the visual object detection and tracking and the neuro-inspired SVITE controller;

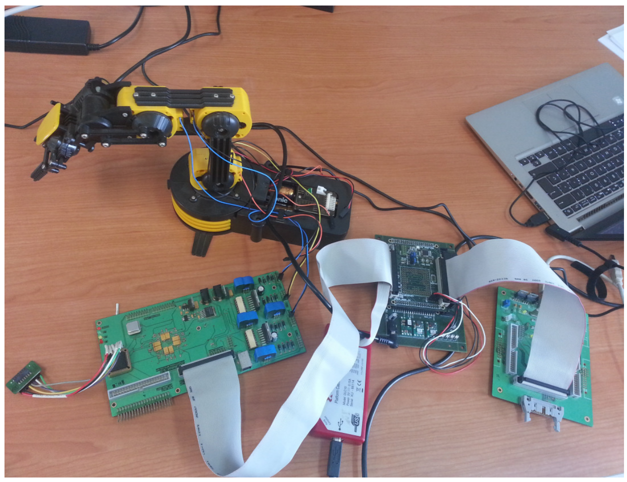

**Fig. 3.** Complete setup with the yellow robotic arm, AER_Robot board (left) AER_Node board (middle), and monitor board (right).

the AER_Robot board to drive the motors with spikes, and the USBAERmini2 for monitoring spikes and visualize them in jAER on the computer screen.

The Spartan 6 XC6S150LX FPGA located at the AER-Node board will run the SVITE (Spike Vector Integration To-End point) algorithm and the visual processing system. The four replicated algorithms will generate the trajectories that should be followed by the robotic arm's joints. The AER-Node board also includes a SPI (Serial Peripheral Interface Bus) plugin to communicate with the computer for configuration purpose; with this interface, the speed parameters can be fixed.

The plane seen by the DVS retina is divided in a grid of 128x128 elements. When the object is detected, the visual processing system delivers the proper area AER address. Then, an interface before the SVITE algorithm matches the address received with four different targets according to each joint angle to reach. In the whole process, the targets are computed as a fixed firing rate.

## VI. Results

Fig 4 shows SVITE response for a real motor under a fixed reference. For these results the target position of the robot has been generated with a VHDL spikes generator. Fig 4 top shows the target position and the evolution of the motor in time. It can be seen how the motor reaches the target position by applying a PFM signal obtained from the SVITE with a fixed GO block slope of 500%. Fig 4 bottom shows that different PFM responses can be obtained depending on the slope programmed in the GO block.

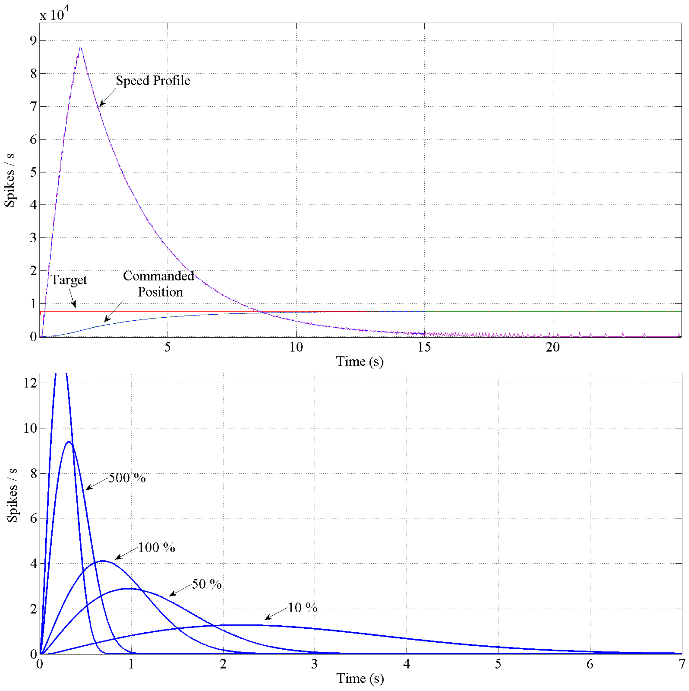


**Fig. 4.** Top: SVITE motor response (commanded position) under a PFM output (speed profile) for a reference fixed position (target). Bottom: output PFM expanded signals driven to motors for different GO slopes.

When applying this SVITE algorithm to each joint of the arm robot the result is a combination movement of all the joints reaching the target position in parallel. Robot trajectory represents the combination of the different joints speed profiles.

## VII. Conclusions

This paper presents a complete neuromorphic systems totally governed by spikes: from dynamic vision sensor, through spike-based cascade architecture of vision processing for object detection and tracking, to a neuro-inspired motor control algorithm implemented in the spike-domain (SVITE).

A Live Demonstration of the complete system will allow other researchers to see and interact with the vision sensor to make the robot move accordingly.


## Acknowlegments

This work was supported by the Spanish grant (with support from the European Regional Development Fund) VULCANO (TEC2009-10639-C04-02/01) and BIOSENSE (TEC2012-37868-C04-02/01).


## References


[1] Indiveri, G., Chicca, E., & Douglas, R. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. IEEE TNN, 2006, 17(1), 211-221.

[2] Chicca, E., Badoni, D., Dante, V., D'Andreagiovanni, M., Salina, G., Carota, L, et al. A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long-term memory. Neural Networks, IEEE Transactions on, 2003, 14(5), 1297-1307.

[3] Cassidy, A., Andreou, A.G., Georgiou, J. Design of a one million neuron single FPGA neuromorphic system for real-time multimodal scene analysis. Information Sciences and Systems (CISS), 2011 45th Annual Conference on , vol., no., pp.1,6, 23-25 March 2011.

[4] Mead, C. A. (1990). Neuromorphic electronic systems. Proc. IEEE 78, 1629–1636.

[5] Lichtsteiner, P., Posch, C., Delbruck, T. A 128 x 128 120 dB 15 us latency asynchronous temporal contrast vision sensor. IEEE J Solid State Circuits. 2008, 43, 566-576.

[6] C. Denk, F. Llobet-Blandino, F. Galluppi, L. A. Plana, S. Furber, J. Conradt. Real-Time Interface Board for Closed-Loop Robotic Tasks on the SpiNNaker Neural Computing System. International Conf. on Artificial Neural Networks (ICANN), Sofia, Bulgaria , 2013 , p. 467-74

[7] Indiveri, Giacomo, Bernabé Linares-Barranco, Tara Julia Hamilton, André Van Schaik, Ralph Etienne-Cummings, Tobi Delbruck, Shih-Chii Liu et al. Neuromorphic silicon neuron circuits. Frontiers in neuroscience 5 (2011).

[8] Bullock, D., Grossberg, S. Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. Psychol Rev. 1988, 95, 49-90.

[9] Bullock, D., Contreras-Vidal, J.L., Grossberg, S. A neural network model for spino-muscular generation of launching and braking forces by opponent muscles. In Proceedings of International Joint Conference on Neural Networks Neural networks, IJCNN 1992.Baltimore, MD, 7-11 June 1992, pp. 91-94.

[10] Gaudiano, P.;Grossberg, S. Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories. Neural Networks. 1991, 4,147-83.

[11] Perez-Peña, F., Morgado-Estevez, A., Linares-Barranco, A., Jimenez-Fernandez, A., Dominguez Morales M. SVITE: A Spike-Based VITE Neuro-Inspired Robot Controller. In Proceedings of the 12th International Conference on Neural Information Processing, ICONIP 2013, Daegu, Korea, November 2013; Volume 8226, pp. 276-383.

[12] Serrano-Gotarredona, T. ; Linares-Barranco, B. "A 128 ,\times 128 1.5% Contrast Sensitivity 0.9% FPN 3 µs Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers". Solid-State Circuits, IEEE Journal of. Vol. 48, No. 3. Pp: 827 - 838. 2013.

[13] Gómez-Rodríguez, F. ; Miró-Amarante, L. ; Diaz-del-Rio, F. ; Linares-Barranco, A. ; Jimenez, G. "Real time multiple objects tracking based on a bio-inspired processing cascade architecture". ISCAS-2010. pp 1399-1402.

[14] Jimenez-Fernandez A, Jimenez-Moreno G, Linares-Barranco A, Dominguez-Morales MJ, Paz-Vicente R, Civit-Balcells A. A Neuro-Inspired Spike-Based PID Motor Controller for Multi-Motor Robots with Low Cost FPGAs. Sensors. 2012; 12(4):3831-3856.