Real-time motor rotation frequency detection with event-based visual and spike-based auditory AER sensory integration for FPGA

A. Rios-Navarro, E. Cerezuela-Escudero, M. Dominguez-Morales, A. Jimenez-Fernandez, G. Jimenez-Moreno, A. Linares-Barranco

Department of Computer Architecture and Technology
University of Seville
Seville, Spain
Email: arios@atc.us.es

***Abstract*— Multisensory integration is commonly used in various robotic areas to collect more environmental information using different and complementary types of sensors. Neuromorphic engineers mimics biological systems behavior to improve systems performance in solving engineering problems with low power consumption. This work presents a neuromorphic sensory integration scenario for measuring the rotation frequency of a motor using an AER DVS128 retina chip (Dynamic Vision Sensor) and a stereo auditory system on a FPGA completely event-based. Both of them transmit information with Address-Event-Representation (AER). This integration system uses a new AER monitor hardware interface, based on a Spartan-6 FPGA that allows two operational modes: real-time (up to 5 Mevps through USB2.0) and data logger mode (up to 20Mevps for 33.5Mev stored in onboard DDR RAM). The sensory integration allows reducing prediction error of the rotation speed of the motor since audio processing offers a concrete range of rpm, while DVS can be much more accurate.**

Keywords*— *Address-Event-Representation; spike-based filters; neuromorphic enginnering; event-based vision; DVS; silicon retina; sensor integration.

I. Introduction

Neuromorphic Engineering tries to solve or improve engineering problems taking inspiration from biology. Central Nervous Systems (CNS) present in many species are able to solve easily many problems that have been very difficult for engineers to be solved through the history. For example, the way how artificial vision has evolved in a completely different way as any CNS solve it. The use of digital cameras and computers and how they work have provoked the design and implementation of many algorithms to process visual information around a non-efficient principle: a sequence of static and big size frames with small differences between two of them that are consecutives on time. Biological retinas do not work with frames, they sense in a continuous way the visual information, and only when one of the pixels is detecting a change in the information, it informs about that change by itself to the CNS in the simplest way: by sending a spike; or an event, that signalizes not only the change but also the kind of change (like positive or negative change).

Many artificial systems that implement bio-inspired software models use biological-like processing that outperform more conventionally engineered machines[1][2][3]. However, these systems generally run several orders of magnitude under real-time, because the models are implemented as software programs. Therefore, direct hardware implementations are required. Neuromorphic research groups around the world are implementing these principles onto real-time spiking and event-based hardware through the development and exploitation of the so-called AER (Address Event Representation) technology, proposed by the Mead lab in 1991 [4] and able to communicate spikes/events between neuromorphic chips using digital words that represent a code or address for each pixel that is transmitting a spike together with extra bits that types a spike to convert it into an event.

In this paper a sensory integration of AER DVS128 silicon retina [5] and an audio frequencies decomposer for FPGA [6][7] is presented integrating both sensors output.

This paper is structured as follows: in section II, both spike based neuromorphic sensory systems (visual and audio) architecture are explained. In section III the sensory integration method is described. Experimental results are shown in section IV and finally, in section V the conclusion are presented.

This work has been supported by BIOSENSE (TEC2012-37868-C04-02/01)

II. Hardware Components

Data acquisition of the working environment is made through the DVS silicon retina and a spike-based audio frequencies decomposer that converts a stereo audio signal into two streams of spikes [8] and divides each stream into a set of streams that correspond to different frequencies channels.

A. AER DVS128 silicon retina

The AER DVS128 (silicon retina) contains an array of autonomous pixels with real-time response to relative changes in light intensity by placing the address of that pixel in an arbitrated asynchronous bus. It is called event when an address and its polarity is transmitted. Pixels that are not stimulated by any change of lighting are not altered, so they do not produce any event. Thus scenes without motion should not generate any output. Some parasite currents at pixel levels make them to fire spikes at very low frequencies when the pixel is inactive. This particular behavior can be removed by applying some basic post-processing to the sensor output [9]. This address, called AE (Address Event), contains the (x,y) coordinates of the pixel that generated the event. The AER DVS128 sensor used considers an array of 128×128 pixels, so 7 bits are needed to encode each dimension of the array of pixels. It also generates a polarity bit indicating the sign of contrast change, whether positive (light increment) or negative (light decrement) [5]. The DVS128 sensor is placed on the PAER [10] interface that allows biases configuration through a USB microcontroller and parallel AER output through the CAVIAR connector [11].

B. Spike-based audio frequencies decomposer

This circuit processes audio signal using classical Digital Signal Processing techniques but in the spikes domain. It processes in the frequency domain the audio information directly encoded as a stream of spikes (Pulse Frequency Modulation - PFM), and it provides the output through an AER interface.

The general system is formed by two digitalized audio streams, which represent the left and right ear's audio signals. A spike transformation step converts the aural information into spikes streams with two Synthetic Spike Generators [12].

The spiking information received from each generator is filtered by two banks of 64 spike-based low-pass filters (SLPF) [6] connected in cascade. Each one of these banks is composed by a set of SLPFs connected in cascade, as many as the number of channels that are implemented for a particular application. In this case 128 SLPFs, 64 by each banks, have been used. Fig. 1 shows part of the architecture. Each stage of the bank represents a channel, and it is formed by a time domain SLPF and a concrete module capable of subtracting two spike-coded signals; the Spikes Hold & Fire (SH&F) [13].

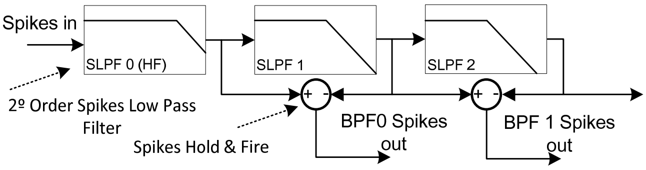

Fig. 1. Spike-based audio filters bank with cascade topology.

The SH&F block is producing a stream of spikes which frequency is the difference between the two input signals frequencies. This is done concretely by holding last input spike coming from any of the two inputs, until next spike arrives from any of the two inputs. When the second spike arrives the subtraction operation is obtained by cancelling both events if they have different polarities, or sending one spike out with the right polarity in the other case. This SH&F block and other spike-based building blocks are explained with details in [14].

Each stream of spikes coming out from SH&F blocks are codified with different addresses and arbitrated together in an AER output bus. So each address in the bus represents the activity of one frequency channel of the audio information.

Fig. 2 shows the spiking output over time of these two audio filters banks in the presence of a rotating motor. In this figure, the x-axis represents time and the y-axis represents the AER address. Every time a concrete event appears, it is represented in this figure by a dot. The bottom values of y-axis (addresses from 0 to 127) shows the left audio source activity and the top values (addresses from 128 to 255) shows the right one. Each bank filter has 128 address because positive and negative spikes can be fired by each channel. In general, both banks outputs present an incremental delay in the output of bigger addresses due to the cascade architecture [15]. This system has been implemented in a Xilinx Virtex 5 FXT FPGA (XC5VFX70T) ML507 for real-time working.

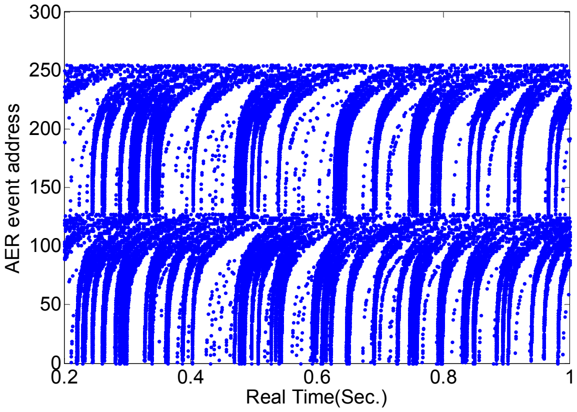

Fig. 2. Output information of two 64-channels banks for audio frequencies decomposition.

All the elements required for designing the two filters banks (i.e. SLPF, SH&F) have been simulated with Xilinx System Generator under Simulink and then implemented in VHDL and designed as small spike-based building blocks. Each of these blocks performs a specific operation on spike streams and can be combined with others in order to build complex spike processing systems. These kinds of systems have been used before, for example, in closed-loop spike-based PID controllers [16] and neuro-inspired SVITE motor controller [17].

C. *Sensors integration*

The sensors information is joined by an AER-Merge module which add a new most significant bit to the address and assigns a different value to it to signalize each of the two sources to merge. When an event is captured by a sensor, it is sent the next stage (processing, filtering or learning) without any temporal distortion, thus the inter-spike-intervals are respected in the after the two sensors signal merging in the same AER bus.

The merged data stream is the input to both, monitor or logger module, depending on the selected operation mode. Then this data stream is sent to a computer through USB 2.0 interface. Each of this two operation modes has its pros and cons. In monitor mode, the AER data stream from the AER Merge module is time-stamped and sent to a computer directly through USB 2.0 high-speed interface. But in logger mode, the AER data stream is time-stamped and stored immediately in on-board DDR2 memory. After the data logging, the on-board sequence of events stored in DDR are sent to the computer. The main difference between the modes is the AER bandwidth while capturing. In monitor mode the bandwidth is limited by bottleneck of USB 2.0 interface. The maximum signaling rate of USB 2.0, in theory, is 480Mbit/s but the effective throughput is limited to 280Mbit/s or 35 MB/s. In this case 4 bytes are used to represent an event (sensor data), thus 8Mev/s could be theoretically achieved, but up 6Mev/s are really achieved. In the other hand, in logger mode, the maximum bandwidth of information captured by the system is higher (about 20Mev/s) because the FPGA-DDR2 memory interface is faster than USB 2.0 interface; so it can be stored with a higher event rate. The on-board DDR2 memory has a maximum capacity of 33.5Mevents, therefore several seconds of typical activity could be captured.

The global architecture of presented hardware system is shown in Fig. 3, where all of descripted modules have been implemented on Spartan 6 FPGA XC6SLX150-FGG484.

Once the sensors data stream are captured and sent to a computer, a jAER [18] software filter is used to estimate rpm motor frequency. The global idea is that while an audio filter will estimate a range of rpm for the motor, the vision filter will offer a more accurate rpm estimation of the motor frequency. The visual stimulus placed on the motor must be clean enough.

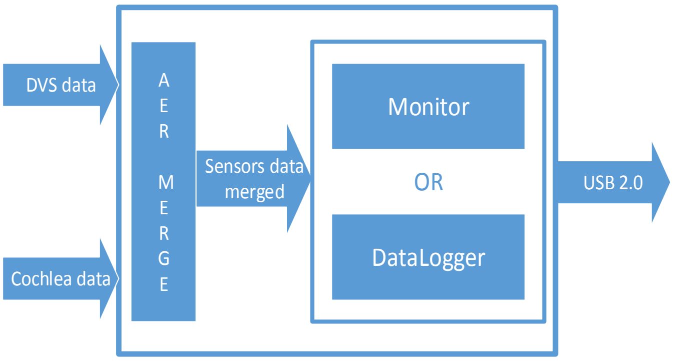

Fig. 3. Completed block diagram of sensory integration hardware system.

The used filter to estimate rpm motor frequency has an easy functioning principle. In this filter a R.O.I (region of interest) is defined in order to detect the figure painted on the surface of a metal disc which has been added to the motor shaft. Each time the figure passes through the R.O.I (one for each spin) the timestamp difference between two sets of events corresponding to two consecutive spins is calculated. Fig. 4 shows the rpm approaches from DVS sensor (green) compared to rpm calculated from an optical encoder (blue) located on the structure holding the motor. This optical encoder represents the ground truth of the disc rotating speed.

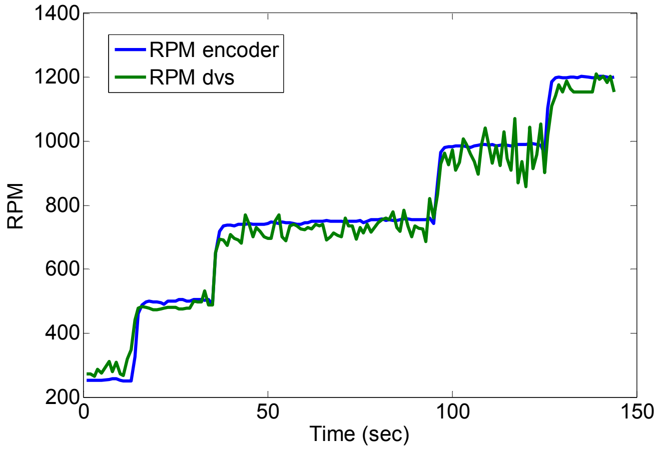

Fig. 4. Calculated rpm from DVS information compared to rpm from encoder.

Moreover, Spike-based audio frequencies decomposer filter defines some ranges of rpm which limit the calculated approaches by DVS filter. In order to calculate that ranges, a pattern recognition approach based on AER convolution is used. Each pattern corresponds with each range of rpm to be calculated, so as many neurons as desirable patterns are implemented. Every time an event is received by the filter, a one-dimensional convolution kernel is applied to that event and the state of the neurons are updated. When a neuron reaches its threshold, it generates an event and the neuron is reset. Each neuron of the one-dimensional convolution is defined mathematically by the equation (1), being *t* an instant time, *W* the kernel of convolutions, *S* the output of each channel of the audio frequencies decomposer, (input of the recognition system) and *Y* the convolved output. The equation (2) shows the output produced by generated events. In equation (1), the *M* length is determined by the number of channels. In this case, *M* length is 128, because of the two banks of 64-channels each.

$$\mathrm{Y(t+1) = Y(t) + \sum_{m=0}^{M}(W(m) * S(t))} \qquad (1)$$

$$Out = Y(t) \geq \theta \rightarrow Y(t) = 0 \qquad (2)$$

The kernel values (*W(m)*) have been obtained from the normalized frequency value of each channel output during a test playback. The values are normalized, in range [0,1], in order to get a volume-independent recognition system. Fig. 5 shows the values of the kernel obtained after the playback for five rpm ranges of the motor. The x-axis represents channels of the left Spike-based audio frequencies decomposer and the y-axis represents the channel normalized rate from each rpm range. Then, the threshold values (θ) have been calculated with the result of (1). The system has been implemented by a two-layer neural network, composed by a one-dimensional convolution layer and a Winner-Take-All (WTA) step in the second layer. Fig. 6 shows the architecture of a single neuron of the system.

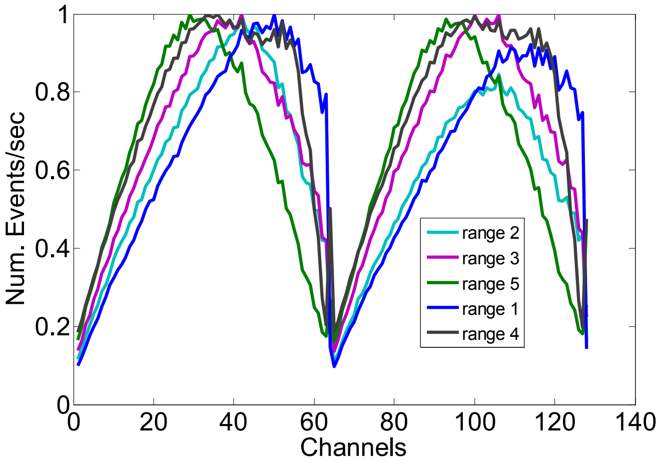

Fig. 5. Values of kernel for different rpm ranges.

When both sensors are integrated, only the values from vision part that are in-between calculated rpm range by auditory filter are taken into account.

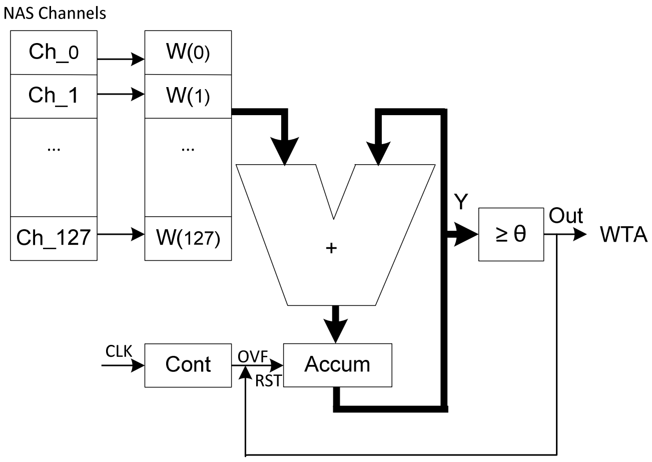

Fig. 6. Architecture of simple neuron of the CNN.

III. The Experiment

As described previously, both sensors information are used to estimate rpm motor frequency. Fig. 7 shows the completed testing scenario where a motor is rotating on a platform in front of two neuromorphic sensory systems. The motor velocity can be changed by a microcontroller and its rotation speed is measured also through an optical encoder, used as ground truth.

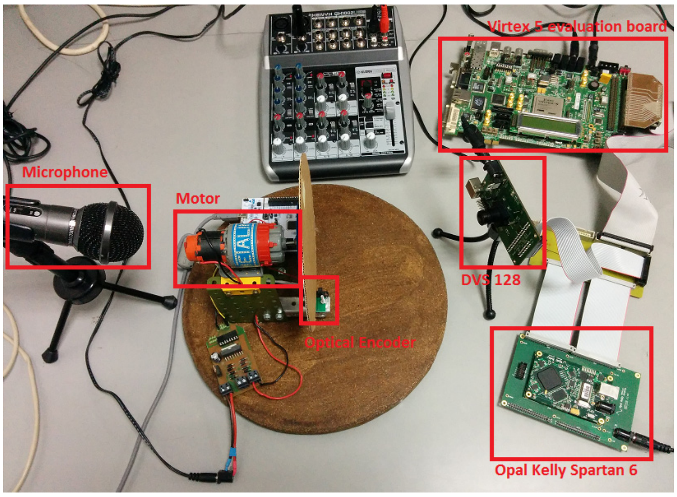

Fig. 7. Photograph of the testing scenario. Left side is the motor and a microphone that captures the motor noise. Middle side is the DVS retina focusing to a disc attached to the motor. Right side is the hardware composed by a Virtex 5 evaluation board for the spike-based audio frequencies decomposition and an Opal Kelly PCB (Spartan 6) where resides the two sensors integration and new AER monitoring circuits.

While motor is rotating, both sensor are capturing event from scene and sending them to the new AER monitor, presented in this paper. Two operation modes have been described for this system: monitor mode that takes the sensors information and sends it directly to a computer; and logger mode that stores the sensor information on a DDR2 memory in a higher bandwidth, and then it transmits this information to a computer.

Whatever the used mode, the information obtained is processed on software (jAER) through two filters which functionality has been described previously.

IV. Results

The performed experiment consists of increasing the motor speed to different values which are estimated by the proposed system. Both algorithms that calculate the rpm, have been implemented like filters under the open source project jAER.

Fig. 8 shows how the outputs of AER DVS128 retina filter are bounded by the range of the Convolution + WTA audio output. The green line is the final system output that matches with the AER DVS128 retina filter output, but taking into account the ranges that have been obtained by the audio classifier. This audio system output is represented in red and brown lines.

As can be seen in the Fig. 8, the system output is very close to the estimated speed measured by an optical encoder. That speed has been considered as the real motor speed, represented in blue line on the same figure.

It can be observed some discrepancies between the real speed values and the estimated ones. When the motor speed changes, the system output suffers sometimes a short delay. This effect is occasioned by the noise produced in the abrupt speed changes of the motor. Due to those circumstances, the AER DVS128 retina filter output gives values very far from the threshold values estimated by the audio system. Those values are not taken into account to obtain the final system output, they are discarded.

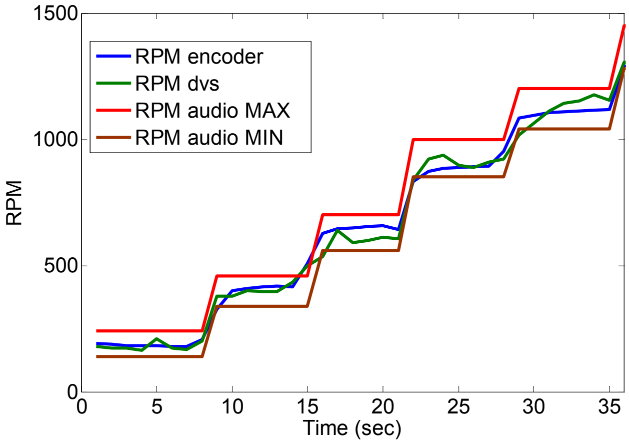

Fig. 8. System output for different motor speeds.

The optical encoder output has been defined as ground truth. Using the equation (3) it is decided whether the system estimation is considered as failure or success, being t an instant of time, Vr the speed calculated by the encoder, Ve the speed calculated by the proposed system and ΔVr the support tolerance.

$$E(t) = Vr(t) - \Delta Vr \le Ve(t) \le Vr(t) + \Delta Vr \quad (3)$$

Considering that rpm ranges obtained by the audio system have a speed difference of 40% between the center of the range and its limits, the tolerance has been set to 10%. Thus the accuracy of system has been calculated to be 94.33%.

V. Conclusions

An event-based visual and spike-based auditory sensory integration system has been presented in this paper. The whole system has been implemented by combining hardware (FPGA) and software (jAER) in a live-demonstration.

In order to validate the system, a testing scenario has been developed that consists in calculating the rotational speed of a direct-current motor using information obtained by an AER DVS128 silicon retina and a bio-inspired digital auditory system.

As future work, we propose to implement on FPGA all jAER algorithms that have been implemented on software for filtering and rpm estimation, in order to obtain a completed hardware system.

The sensory integration system proposed in this work can be used in automotive industry to test the engine under quality revision. For example, when the outputs from both stage are very different and the global output is illogical, or to signalizing a fault in the rotation of the motor as quick as it is produced to facilitate the localization of the problem in the engine.

References

[1] J. Lee: A Simple Speckle Smoothing Algorithm for Synthetic Aperture Radar Images. Man and Cybernetics. Vol. SMC-13(1981).

[2] T. Crimmins: Geometric Filter for Speckle Reduction. Applied Optics. Vol. 24, pp. 1438-1443 (1985).

[3] A. Linares-Barranco et al: AER Convolution Processors for FPGA. ISCASS (2010).

[4] M. Sivilotti: Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks. Ph.D. Thesis, Caltech (1991).

[5] P. Lichtsteiner, C. Posh, T. Delbruck: A 128×128 120dB 15 us Asynchronous Temporal Contrast Vision Sensor. IEEE Journal on Solid-State Circuits. Vol. 43, no 2, pp. 566-576(2008).

[6] M. Dominguez-Morales, A. Jimenez-Fernandez, E. Cerezuela-Escudero, R. Paz-Vicente, A. Linares-Barranco and G. Jimenez,“On the Designing of Spikes Band-Pass Filters for FPGA,” Artificial Neural Networks and Machine Learning. (ICANN 2011). LNCS 2011, 6792, pp. 389-396.

[7] Jiménez-Fernandez, A., 2010.Diseño y Evaluación de sistemas de control y procesamiento de señales basados en modelos neuronales pulsantes. University of Seville, PhD Thesis, pp. 229-245.

[8] F. Gomez-Rodriguez, R. Paz, L. Miro, A. Linares-Barranco, G. Jimenez and A. Civit,“Two hardware implementation of the exhaustive synthetic AER generation method”. LNCS 2005, 41, pp.534–540.

[9] A. Linares-Barranco et al: A USB3.0 FPGA Event-based Filering and Tracking Framework for Dynamic Vision Sensors. International Sysposium in Circuits and Systems (ISCAS 2015). (Accepted).

[10] DVS128_PAER Dynamic Vision Sensor. http://www.inilabs.com/support/dvs128paer

[11] R. Serrano-Gotarredonda, et al., “CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory-Processing-Learning-Actuating System for High-Speed Visual Object Recognition and Tracking,” Neural Networks, IEEE Transactions on, vol.20, no.9, pp.1417,1438, Sept. 2009.

[12] F. Gomez-Rodriguez, et al.,“Two hardware implementation of the exhaustive synthetic AER generation method”. LNCS 2005, 41, pp.534–540.

[13] A. Jimenez-Fernandez, et al., “Building blocks for spikes signals processing”. In Proceedings of the 2010 International Joint Conference onNeural Networks (IJCNN), 18–23 July 2010; pp. 1-8.

[14] A. Jimenez-Fernandez, G. Jimenez-Moreno, A. Linares-Barranco, M.J. Dominguez-Morales, R. Paz-Vicente and A. Civit-Balcells, “A neuro-inspired spike-based PID motor controller for multi-motor robots with low cost FPGAs” Sensors. 2012, 12, pp. 3831-3856.

[15] A. Jimenez-Fernadez, et al., “ On AER Binaural Cochlea for FPGA: design, synthesis and experimental analysis”. *Sensors2014*. (Under revision).

[16] A. Jimenez-Fernandez, et al., “A neuro-inspired spike-based PID motor controller for multi-motor robots with low cost FPGAs”. Sensors. 2012, 12, pp. 3831-3856.

[17] F. Perez-Peña, et al., “Neuro-Inspired Spike-Based Motion: From Dynamic Vision Sensor to Robot Motor Open-Loop Control through Spike-VITE”. *Sensors*2013, *13*, pp. 15805-15832.

[18] jAER Open-Source Software Project. Avaible online http://jaer.wiki.sourceforge.net/