AER Auditory Filtering and CPG for Robot Control

F. Gómez-Rodríguez, A. Linares-Barranco, L. Miró.

Arquitectura y Tecnología de Computadores.
Universidad de Sevilla.
Av. Reina Mercedes s/n, 41012-Sevilla, SPAIN
gomezroz@atc.us.es

Shih-Chii Liu[1], André van Schaik[2], R. Etienne-Cummings[3], M. A. Lewis[4].
[1] ETH Zürich. Switzerland
[2] University of Sydney, Australia.
[3] John Hopkins University, USA.
[4] Iguana Robotics, Inc., USA.

Abstract— **Address-Event-Representation (AER) is a communication protocol for transferring asynchronous events between VLSI chips, originally developed for bio-inspired processing systems (for example, image processing). The event information in an AER system is transferred using a high-speed digital parallel bus. This paper presents an experiment using AER for sensing, processing and finally actuating a Robot. The AER output of a silicon cochlea is processed by an AER filter implemented on a FPGA to produce rhythmic walking in a humanoid robot (Redbot). We have implemented both the AER rhythm detector and the Central Pattern Generator (CPG) on a Spartan II FPGA which is part of a USB-AER platform developed by some of the authors.**

I. INTRODUCTION

The Address-Event-Representation (AER) protocol was proposed in 1991 as a way of transferring the state of an array of analog time-dependent values from one chip to another chip or module [1]. These AER chips or modules typically use mixed analog and digital principles and exploit spikes for coding information. Figure 1 illustrates the principle behind the AER communication. The emitter chip contains an array of cells (for example, the pixels of an imager or an artificial retina) where each cell contains a continuously varying time-dependent state that changes with a slow time constant (in the order of *ms*). Each cell or pixel includes a local oscillator that generates digital pulses of minimum width (a few *ns*) at a rate proportional to the state of the cell (e.g. pixel intensity for a retina) when a spike rate coding is used. Each time a pixel generates a pulse (which is called an "event"), it communicates with the array periphery and a digital word representing the address for that pixel is placed on the external inter-chip digital bus (the AER bus). Additional handshaking lines (Acknowledge and Request) are used for completing the asynchronous communication. The inter-chip AER bus operates at the maximum possible speed. In the receiver chip the pulses are directed to the cells corresponding to the address on the bus. In this way, cells with the same address in the emitter and receiver chips are virtually connected with a stream of pulses. The receiver cell integrates the pulses and reconstructs the original low frequency continuous-time waveform. Cells that are more active access the bus more frequently than those that are less active.

Transmitting cell addresses allows extra operations to be performed on the events before these addresses are sent to another chip. For example in a retina, the activity of the pixels in the array represents the input image. By translating the address of the events during transmission, the image can be shifted or rotated. This translation of the addresses can be achieved for instance by inserting properly coded EEPROMs into the communications pipeline. In addition, the event-based nature of the AER protocol also allows for very efficient convolution operations within a receiver chip [2]. Furthermore, an image transmitted by one chip can be broadcast to many receiver chips in parallel through the asynchronous communication protocol. .
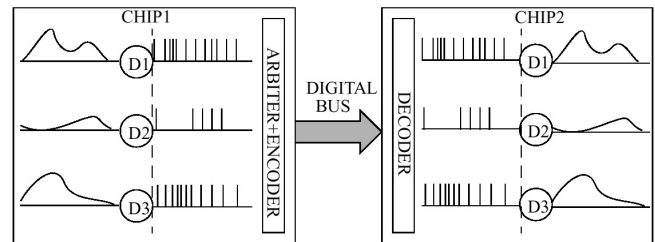

Figure 1. Rate-Coded AER inter-chip communication scheme.

There is a growing community of designers that are developing vision and audition AER systems for motor control in various bio-inspired applications, as demonstrated for instance by the recent projects in the AER workgroup at the Neuromorphic Engineering Workshop series [3]. The goal of this community is to build large multi-chip and multi-layered hierarchically-structured systems which are capable of performing complex massively-parallel processing in real time [4][5].

In this paper we use an AER silicon cochlea (AER EAR [6][9]) and an AER filter to detect rhythmic sounds. We also describe the implementation of Central Pattern Generators (CPG) responsible for rhythmic control of locomotion, to drive the servo motors of a bio-inspired robot, Redbot, so

that it walks at different speeds depending on the cadence of the sound.

We have connected the output of the AER EAR to a USB-AER platform described in [7]. This interface was programmed to filter the audio AER signal to extract the rhythm from a sound stream. The output of this USB-AER board was connected to a second USB-AER board which implemented the CPG that makes Redbot walk at different speeds. Figure 2 shows a block diagram of the experimental setup.
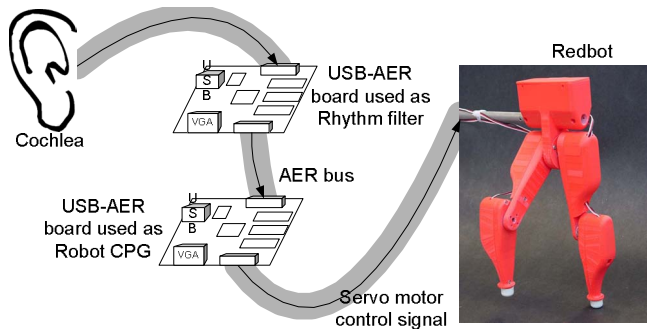

Figure 2. Block diagram of the experiment. A silicon cochlea is connected, via an AER bus, to a USB-AER board implementing a rhythm filter, which in turn is connected to another USB-AER implementing the redbot CPG that generates the servo motor control signal for the Redbot.

In the following sections we describe the different AER components that were used in this audio processing experiment: the silicon cochlea, the audio filter, the Redbot and the CPG. Finally we present the conclusions.
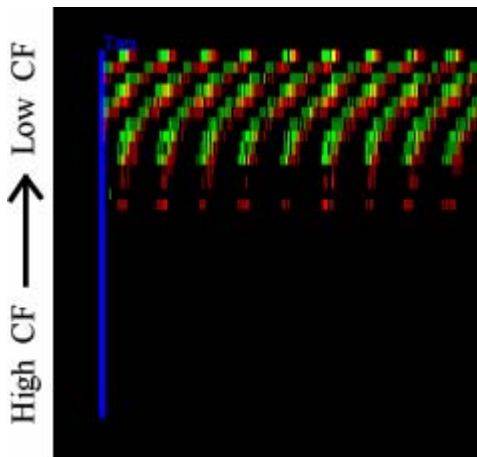

Figure 3. Output spikes from the cochlea in response to a pure-tone 100 Hz auditory input. Only the channels with low center frequencies (CF) respond to the input. The red spikes are outputs of one cochlea and the green spikes are output of the second cochlea. The spikes are phase-locked to this pure-tone signal.

II. AER SILICON COCHLEA

The cochlea chip has a pair of silicon cochleas with 32 sections each [9]. Each section is modeled by a second-order low-pass filter [6], followed by a simplified Inner Hair Cell circuit and a spiking neuron circuit. The exponentially decreasing cutoff frequencies across the 32 sections range from 50 Hz to 1 kHz. The AER spikes of the cochlea correspond to the addresses of the spiking neurons of the different frequency channels. The cochlea chip can be tuned so that the output spikes of each channel are phase-locked to the input frequencies as shown in Figure 3.

III. AER AUDIO FILTER FOR FPGA

To extract the rhythm, we look for "beats" in the sound input. If the original sound consists of a sequence of "beats", the output of the cochlea will consist of several bursts of spikes; with one burst for each "beat".

We implement the "beat" detection by looking at the inter-spike interval (ISI) of the cochlear spikes: if the ISI is small, the spikes are part of the same burst. On the other hand, if the ISI is large the spikes came from different bursts. Figure 4 illustrates this.
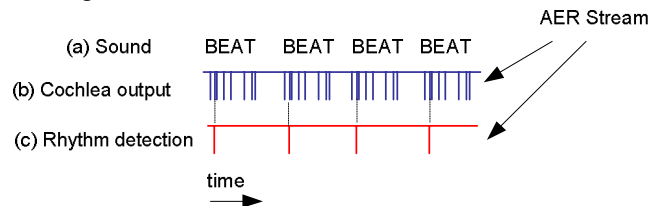

Figure 4. "Beat" detection. (a) The original sound consists of a sequence of "beats". (b) The cochlea output consists of a sequence of spike bursts, each burst correspond to a "beat". (c) Rhythm of the original sound, one spike for each beat.

When the ISI drops below a set threshold a new "beat" is detected and a signal is sent to the next stage to make the robot move. The threshold depends on the robot response time. In our case the threshold is set to 250 ms, which is the time necessary to make the Redbot to walk one step. Depending on the sound, it is possible that some "beat" is lost in the 250ms, but in our case this is not a problem because Redbot cannot move faster.

IV. USB-AER BOARD

Standalone applications require an interface which is independent from a computer. The USB-AER board has been developed under the CAVIAR EU project for interconnecting AER chips or systems and is also capable of performing its own event-based operations. These operations are implemented by a circuit that resides in a FPGA, which is the kernel of the interface. This FPGA also connects to an external RAM to allow event address mapping capabilities. The USB-AER board has to be able to load the FPGA and the mapping RAM from some type of non-volatile storage that can easily be modified by the users. MMC/SD cards used in digital cameras are a very attractive possibility. However during a system development stage, users generally prefer to load the board directly from a computer and for this purpose USB seems the most suitable solution.

Many AER researchers would like to be able to easily interface their system to a laptop computer. This requirement can also be supported with the USB-AER board as it includes a relatively large FPGA that can be loaded from MMC/SD or USB, a large SRAM bank and two AER ports. Thus the board can be used also as a sequencer or as a monitor of a sequence of AER events. Due to the bandwidth limitations of full speed USB (12Mbit/s), hardware based event-to-frame conversion is essential in this board for high, or even moderate, event rates.

The USB-AER board shown in Figure 5 is based around a Spartan-II 200 Xilinx FPGA, with a 12ns 512K by 32 SRAM memory bank. The board uses a Silicon Laboratories C8051F320 microcontroller to implement the USB and the MMC/SD interface. A simple VGA monitor interface is also provided to allow the board to act as a monitor (frame grabber).

The board will act as a different device according to the module that is loaded in the FPGA either through a MMC/SD card or from the USB bus. Currently the following modes are implemented [7]:

A. Mapper

Mapping functionality involves a change of the event addresses. Theses changes can be (a) one to one, (b) one to one replicated, (c) one to several, (d) one to several replicated, (e) one to several depending on a probability function and (f) no event.

B. Monitor

This functionality allows users to record the traffic on an AER bus and to present the data on a computer display. Monitor types are: (a) frame-grabber: capturing events during a time and then reconstructing an image; (b) sniffer: events are captured and stored with a timestamp without interfering in the normal traffic; (c) logger: same function as the *sniffer* but with traffic interference; (d) VGA: capturing the events and converting them into a Gray VGA.

Figure 5. USB-AER Board.

C. Sequencer

Here, the USB-AER board is used as an event generator. There are two types of AER traffic generation: (a) Frame-based generation: generating events from a digital image using Random or Exhaustive methods [8]; (b) Player: generating events using timestamp information.

A MATLAB interface to control this board is available. It allows the loading of modules into the FPGA, the uploading or downloading of data to or from the board, and the display of the received images when the board acts as a monitor [13].

A Linux driver for the USB-AER is currently under test. With this driver the USB-AER board can easily be integrated with several MATLAB applications developed at the Institute of Neuroinformatics, Univ of Zurich [10].

V. THE REDBOT ROBOT

Redbot, a product of Iguana Robotics, is a biped with hip and knee actuators. The actuators are servo motors that are controlled with 50Hz Pulse Width Modulation (PWM) signals. Each motor/muscle is controlled independently. The robot is constrained to walk in the sagittal plane. It can fall vertically but not laterally because a horizontal boom is used to maintain lateral stability. Redbot, which is ~20cm high, can walk at speeds up to 4Hz. Figure 2 shows a picture of Redbot.
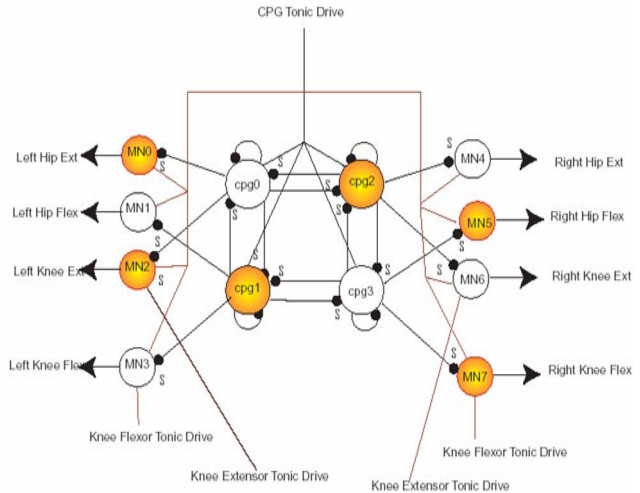

Figure 6. Typical CPG network for controlling Redbot [11]. The network is implemented with Integrated and Fire with Adaptation neurons (IFA). The neurons are connected to realize half-center oscillators for hip control. Knee muscles are 90 degrees out of phase with the hips, which is realized also using mutual inhibition.

Typically, a CPG network, composed of Integrate-and-Fire with Adaptation (IFA) neurons, is used to control Redbot [11]. Figure 6 shows a picture of the network. The hip muscles are controlled with neurons that form a half-center oscillator. The knee muscles are typically 90 degrees out of phase with the hips and are also generated with IFA neurons, where mutual inhibition is used to realize the phase

lag. In this way, we are able to realize all the control signals required to make Redbot walk. The CPG networks are usually implemented with CPG chips that contain programmable and reconfigurable, but continuously connected, spiking IFA networks [12]. In the present work, the CPG network is not implemented with continuous time IFA neurons, but with AER networks. The construction of this network is shown in Figure 6.

VI. AER CPG

For this experiment, we have designed a basic CPG in VHDL that transforms the output from the AER audio filter (explained in section III) into a sequence of commands for the four servo motors that make Redbot walk. The output of the AER rhythm detector consists of a sequence of AER spikes. The spike frequency depends on the rhythm of the sound.

The walking speed of Redbot depends on the frequency of the incoming events at the CPG. To make the Redbot walk, we have to generate the extension and flexion signals for the hips and knees (see Figure 7). These signals are used to generate the PWM signals for controlling the motor/muscle. The movements of each leg have a phase difference of 180º.
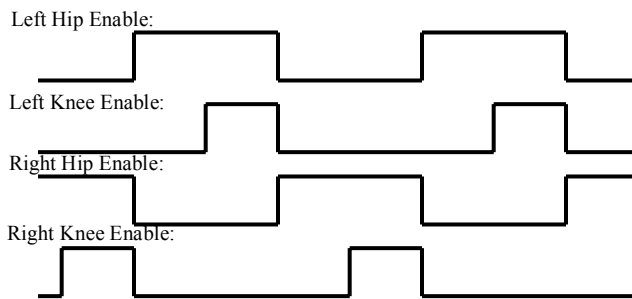

Figure 7. Servo enables for Redbot walk CPG.

The three PWM signals for controlling the motor/muscle are the: *idle signal*, *flexion signal and extension signal.* **The first signal** allows Redbot to keep the straight leg. The second signal allows the articulation to flex (hip or knee) and the last signal allows the articulation to extend.

VII. CONCLUSIONS

This paper presents an AER system composed of several components developed by different laboratories. The components are connected together to illustrate the possibilities of using an Address-Event-Representation in a multi-component system.

With this demo, we demonstrate that the AER approach can be used all the way from the sensor stage to the actuators. We present a simple process for detecting

rhythms from the spike outputs of an AER cochlea and a central pattern generation based on AER to make a robot walk at different speeds.

The FPGA-based USB-AER board is a very attractive tool for interfacing different AER chips and for performing spike processing at the same time.

REFERENCES

[1] M. Sivilotti, *Wiring Considerations in Analog VLSI Systems with Application to Field-Programmable Networks*, Ph.D. Thesis, California Institute of Technology, Pasadena CA, 1991.

[2] Teresa Serrano-Gotarredona, Andreas G. Andreou, Bernabé Linares-Barranco. "AER Image Filtering Architecture for Vision-Processing Systems". *IEEE Transactions on Circuits and Systems: Fundamental Theory and Applications*, Vol. 46, N0. 9, September 1999.

[3] A. Cohen, R. Douglas, C. Koch, T. Sejnowski, S. Shamma, T. Horiuchi, and G. Indiveri, *Report to the National Science Foundation: Workshop on Neuromorphic Engineering*, Telluride, Colorado, USA, June-July 2004. [www.ini.unizh.ch/telluride]

[4] Kwabena A. Boahen. "Communicating Neuronal Ensembles between Neuromorphic Chips". *Neuromorphic Systems*. Kluwer Academic Publishers, Boston 1998.

[5] Misha Mahowald. *VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function*. PhD. Thesis, California Institute of Technology Pasadena, California, 1992.

[6] A. van Schaik, E. Fragnière, and E. Vittoz, "Improved Silicon Cochlea using Compatible Lateral Bipolar Transistors," *Advances in Neural Information Processing Systems 8*, edited by D. Touretzky, et al., MIT Press, Cambridge MA, pp. 671-677, 1996.

[7] F. Gómez-Rodríguez, R. Paz, A. Linares-Barranco, M. Rivas, L. Miró, G. Jiménez, A. Civit. "AER tools for Communications and Debugging". *Proc. IEEE ISCAS06*. Kos, Greece, May 2006.

[8] A. Linares-Barranco, G. Jimenez-Moreno, A. Civit-Ballcels, and B. Linares-Barranco. "On Algorithmic Rate-Coded AER Generation". *IEEE Transaction on Neural Networks*. May-2006.

[9] V. Chan, A. van Schaik, and S.-C. Liu, "Spike Response Properties of an AER EAR", *Proc. IEEE ISCAS06*, pp. 859-862, 2006.

[10] M. Oster, Serverbased Software Architecture for AER systems [http://www.ini.unizh.ch/~mao/AerSoftware/SoftwareOverview.pdf]

[11] M. A. Lewis, F. Tenore and R. Etienne-Cummings, "CPG Design using Inhibitory Networks," *Proc. ICRA 2005*, Barcelona, Spain, 2005.

[12] F. Tenore, M. A. Lewis and R. Etienne-Cummings, "Entrainment of Silicon Central Pattern Generators for Legged Locomotory Control," *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul and B. Scholkopf (Eds.), MIT Press, Cambridge, MA, 2004.

[13] Address Event Representation tools. [http://www.atc.us.es/AERtools]