A Simple Model of Spike Processing [*]

Aurel A. Lazar
Department of Electrical Engineering
Columbia University, New York, NY 10027
aurel@ee.columbia.edu

December 4, 2005

Abstract

We describe a simple model of spike processing build upon a number of neural hardware primitives including integrate-and-fire neurons, passive dendritic trees, simple integrators, inhibition logic and one-to-many axonal/dendritic tree connectivity. Functionally, our model of spike processing consists of neuro-modulators, communication channels, neuro-demodulators and filters. Integrate-and-fire neurons play the role of neuro-modulators. They represent dendritic currents in the spike domain through a process of reversible computation (non-linear modulation). Communication channels model axons. Irreversible computation takes place, at least in part, in dendritic trees as time domain linear and/or non-linear operations. Assuming that the stimulus at the input of an integrate-and-fire sensory neuron is bandlimited, we demonstrate how to construct a linear operator that maps an arbitrary stimulus into a desired neuronal signal.

1 Introduction

A variety of high performance information processing devices use a signal processing chain as shown in Figure 1. Examples abound in telecommunications, biomedical, multimedia, robotics and automotive signal processing applications. An analog output signal from a sensor is converted into a digital representation by an analog-to-digital converter (ADC); it is sampled at discrete time intervals provided by a clock and quantized into a set of discrete levels. The digital signal processing block transforms the digital input signal into the desired output signal. The digital output is subsequently converted back to an analog format with a digital-to-analog converter (DAC). The latter drives an actuator with the appropriate

[*] BNET Technical Report #2-05, Department of Electrical Engineering, Columbia University, New York, NY, April 2005. Presented at the Computational Neuroscience Meeting, July 17-21, 2005, Madison, WI. Neurocomputing, to appear.

waveform. More formally, a continuous waveform $u = u(t), t \in \mathbb{R}$, bandlimited to $[-\Omega, \Omega]$, at the input of the Analog-to-Digital Converter is represented by a set of discrete values $u(kT), k \in \mathbb{R}$, with $t = \pi/\Omega$. Processing is executed on a quantized version of the discrete samples $u(kT), k \in \mathbb{Z}$.

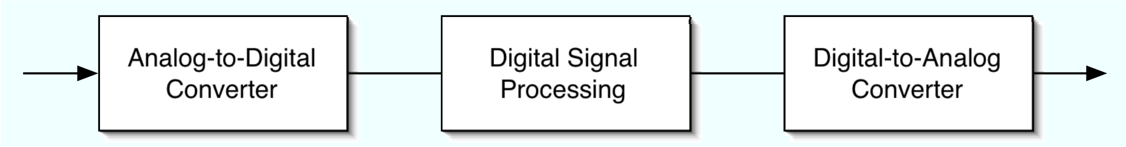
Analog-to-Digital Converter
Digital Signal Processing
Digital-to-Analog Converter

Figure 1: General Signal Processing Chain with a Digital Signal Processing Core

The model of spike processing considered in this paper is shown in Figure 2. It has the same broad scope as the signal processing chain briefly presented above. The Time Encoding Machine in Figure 2 corresponds to the Analog-to-Digital Converter in Figure 1. Similarly, the Time Domain Computing block in Figure 2 corresponds to the Digital Signal Processing block in Figure 1 and the Time Decoding Machine to the Digital-to-Analog Converter. Recall that, a TEM maps a continuous waveform $u = u(t), t \in \mathbb{R}$, into an increasing time sequence $(t_k), k \in \mathbb{Z}$. The TDM implements the inverse of the TEM map (both maps are non-linear).

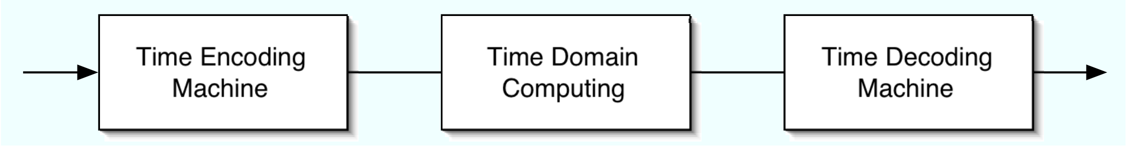

Figure 2: General Signal Processing Chain with a Time Domain Core

In this paper we examine the realizability of the spike processing model shown in Figure 2 using neural hardware primitives such as integrate-and-fire neurons, one-to-many dendritic tree connectivity, (passive) dendritic trees, inhibition logic, simple integrators, etc.. Our work builds on [4] and [5] where we have shown that integrate-and-fire models of sensory neurons act akin modulators in communications. They represent analog inputs (*i.e.*, aggregated dendritic currents) in the spike domain without loss of information. The perfect recovery algorithm, however, calls for the computation of a pseudo-inverse.

For a TEM consisting of an integrate-and-fire neuron with bias, we demonstrate that a recursive algorithm already investigated in [4] and [5] provides an approximate method of recovery that can be readily implemented in neural hardware. We also demonstrate that an arbitrary linear operator (filter) can be implemented as a parallel filter bank consisting of simple integrators with feedback. The integrators model synapses. A combination of these integrators and an appropriate choice of their parameters results in the desired transfer function of the filter. Each operational integrator contributes to the overall transfer function of the filter. The integrators are operational only if spikes are routed to their inputs. A simple mixing circuit is used to block or to allow through individual spikes. This allows for a very flexible routing of spikes to the appropriate integrators and, consequently, the construction of the desired filter by simple inhibition logic. Arbitrary time invariant filters can be realized in this manner from very simple neural hardware elements.

Functionally, our model of spike processing consists of neuro-modulators, communication channels, neuro-demodulators and filters. Integrate-and-fire neurons play the role of neuro-modulators. They represent dendritic currents in the spike domain through a process of reversible computation (non-linear modulation). Communication channels model axons. Irreversible computation takes place, at least in part, in dendritic trees. Such a model offers a platform for a calculus with spikes including learning algorithms. In [7] and [1], the authors argue based on experimental grounds that the dendritic tree decodes the received spike trains and executes linear operations on the decoded waveforms. Thus, our investigations of spike processing in a network consisting of neural hardware primitives provide a rigorous theoretical model for the above mentioned assertions and, more broadly, spike computation.

This paper is organized as follows. In section 2 the mapping of a dendritic current into a spike train by a TEM consisting of an integrate-and-fire neuron with bias is reviewed. An algorithm for the loss free recovery of the dendritic current based on reading the neural spike train is briefly discussed. The neural hardware realization of the recovery algorithm is investigated in section 3. In section 4 a methodology for the implementation of arbitrary linear operators is presented. Section 5 concludes the paper.

2 Information Representation in the Spike Domain

Consider an integrate-and-fire neuron representing a dendritic current $u = u(t), t \in \mathbb{R}$, as a sequence of trigger times $(t_k), k \in \mathbb{Z}$, where $\mathbb{R}$ and $\mathbb{Z}$ denote the set of real numbers and integers, respectively. The trigger times represent the time instances when spikes are generated. The block diagram of an integrate-and-fire neuron is shown on the left hand side of Figure 3. We assume that the dendritic current u, $|u(t)| \leq c < b$, has finite energy on $\mathbb{R}$ and is bandlimited to $[-\Omega, \Omega]$.

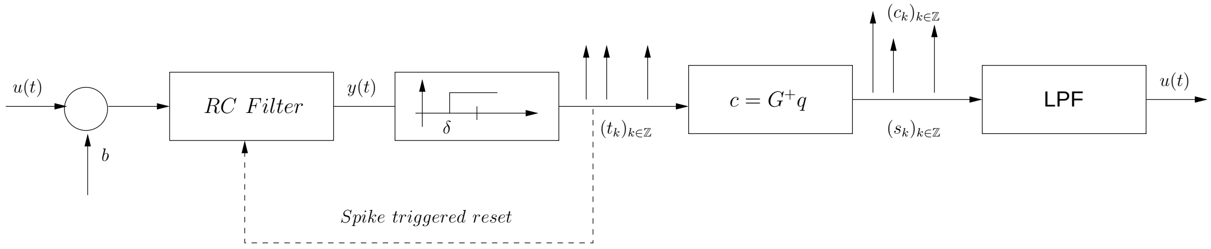

Figure 3: Information Representation with an Integrate-and-Fire Neuron and Perfect Recovery.

In [4], [5] we established that, under natural conditions (see below), an observer reading the spike train generated by an integrate-and-fire neuron can recover the dendritic current loss-free. The structure of the perfect decoder is shown on the right hand side of Figure 3. Its operation is highly intuitive. Dirac-delta pulses (spikes) are generated at times s_k

with weight c_k, $k \in \mathbb{Z}$, and then passed through an ideal low pass filter with unity gain for $\omega \in [-\Omega, \Omega]$ and zero otherwise. Thus, the output of the decoder is given by

$$u(t) = \sum_{k \in \mathbb{Z}} c_k \cdot g(t - s_k), \tag{1}$$

where $g(t) = sin(\Omega t)/\pi t$ and $s_k = (t_{k+1} + t_k)/2$ for all $t, t \in \mathbb{R}$. The c_k's, are the solution to a system of linear equations both in the case of an integrate-and-fire neuron with an absolute refractory period [4] or a leaky integrate-and-fire neuron [5]. Thus, the integrate-and-fire neuron is, under certain conditions, invertible and the mapping of the input dendritic current $u(t), t \in \mathbb{R}$, into the sequence of trigger times $(t_k), k \in \mathbb{Z}$, can be interpreted, as the result of a process of *reversible* computation. The condition for invertability for the case of an *ideal* integrate-and-fire neuron with capacity κ is particularly simple. It is given below as part of the formal recovery algorithm. Let $\mathbf{c} = [c_k]$, $\mathbf{q} = [q_k] = [\int_{t_k}^{t_{k+1}} u(s)\ ds]$ and $\mathbf{G} = [G_{lk}] = [\int_{t_l}^{t_{l+1}} g(s - s_k)\ ds]$. We have the following [4], [5],

Theorem 1 (Recovery Algorithm) *If $\kappa\delta < (b - c)\frac{\pi}{\Omega}$, the bandlimited stimulus $u = u(t), t \in \mathbb{R}$, can be perfectly recovered from $(t_k)_{k \in \mathbb{Z}}$ as*

$$u(t) = \sum_{k \in \mathbb{Z}} c_k g(t - s_k),$$

where $g(t) = \sin(\Omega t)/\pi t$ and $s_k = (t_{k+1} + t_k)/2$. Finally, $\mathbf{c} = \mathbf{G}^+\mathbf{q}$, where $\mathbf{G}^+$ denotes the pseudo-inverse of $\mathbf{G}$.

Proof: Informally, under appropriate conditions [4], [5]

$$u(t) = \sum_{k \in \mathbb{Z}} c_k g(t - s_k)$$

and by integrating both sides from t_l to t_{l+1} we obtain

$$\int_{t_l}^{t_{l+1}} u(s) ds = \sum_{k \in \mathbb{Z}} c_k \int_{t_l}^{t_{l+1}} g(s - s_k) ds,$$

i.e., $\mathbf{q} = \mathbf{G}\mathbf{c}$, and thus, $\mathbf{c} = \mathbf{G}^+\mathbf{q}$,

Note that the condition $\kappa\delta < (b - c)\frac{\pi}{\Omega}$ guarantees the convergence of the recovery algorithm (1). Theorem 1 has a very simple and intuitive interpretation. The information contained in the spike train $(t_k), k \in \mathbb{Z}$, is equivalent with the information contained in the dendritic current $u(t), t \in \mathbb{R}$, provided that, the maximum number of spikes per unit of time does not exceed the Nyquist rate.

3 Stimulus Recovery with NeuroHardware

Can the recovery algorithm, shown in block diagram form in Figure 3, be implemented by only using *neural hardware* primitives? By this we mean, integrate-and-fire neurons, simple integrators, (passive) dendritic trees, inhibition logic, one-to-many axonal/dendritic tree synaptic connectivity, etc.. An indication of the solution space to this question can be obtained by investigating the neural hardware realization of the pseudo-inverse building block and the LPF building block of Figure 3.

3.1 Formulation of a Recursive Recovery Algorithm

In order to investigate the implementation of the pseudo-inverse building block, we shall first reformulate the recovery algorithm as a recursion [4], [5]. We will show that already the zero'th order approximation of this recursion offers a good approximation of the bandlimited stimulus.

Consider the processor $\mathcal{A}$ described by:

$$\mathcal{A}u = \sum_{k\in\mathbb{Z}} \int_{t_k}^{t_{k+1}} u(s)ds\; g(t-s_k) = \sum_{k\in\mathbb{Z}} [\kappa\delta - b(t_{k+1}-t_k)]\; g(t-s_k),$$

where $g(t) = \sin(\Omega t)/\pi t$ and $s_k = (t_{k+1}+t_k)/2$. Let $u_l = u_l(t)$, $t \in \mathbb{R}$, be a sequence of bandlimited functions defined by the recursion:

$$u_{l+1} = u_l + \mathcal{A}(u - u_l),$$

for all $l,\ l \in \mathbb{N}$, with the initial condition $u_0 = \mathcal{A}u$.

Theorem 2 (Recursive Recovery Algorithm) *If $\kappa\delta < (b-c)\frac{\pi}{\Omega}$, the bandlimited stimulus $u, u(t) \leq c < b$, $t \in \mathbb{R}$, can be perfectly recovered from $(t_k)_{k\in\mathbb{Z}}$ as*

$$\lim_{l\to\infty} u_l(t) = u(t),$$

and

$$\parallel u - u_l \parallel \leq r^{l+1} \parallel u \parallel, \tag{2}$$

where $r = \frac{\kappa\delta}{b-c}\frac{\Omega}{\pi}$. Furthermore,

$$u_l(t) = \mathbf{g}^T \mathbf{P}_l \mathbf{q},$$

where $\mathbf{P}_l$ is given by $\mathbf{P}_l = \sum_{k=0}^{l} (\mathbf{I}-\mathbf{G})^k$.

Since zero'th order approximation of $u(t)$ amounts to

$$u_0(t) = \sum_{k\in\mathbb{Z}} [\kappa\delta - b(t_{k+1}-t_k)]\; g(t-s_k),$$

the associated frame coefficients are given by

$$c_k = \kappa\delta - b(t_{k+1} - t_k) = \int_{t_k}^{t_{k+1}} u(s)\ ds.$$

In other words $\mathbf{G} = \mathbf{I}$ and thus $\mathbf{c} = \mathbf{q}$. Consequently, the zero'th order approximation of the recursive recovery algorithm has an exceedingly simple implementation as there is no need to compute the pseudo-inverse $\mathbf{G}^+$. The bound on the performance of this approximate recovery algorithm can be obtained by setting $l = 0$ in the inequality (2) of Theorem 2 above.

Example 1 Referring again to Figure 3, the recovery algorithm consists of two building blocks. The first building block computes a pseudo-inverse followed by a matrix multiplication. The second building block implements a linear filter. The zeroth order approximation of the first building block is given by $\mathbf{c} = \mathbf{q}$ and is shown in Figure 5. As shown in Figure 4 the recovery algorithm based on this approximation has a stimulus recovery error below 1%. In our experiments we used arbitrary stimuli bandlimited to 40 Hz and bounded by 1.

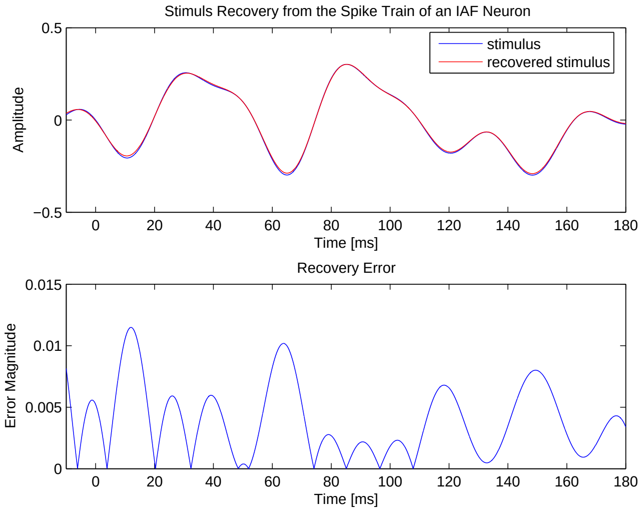

Figure 4: Zero'th Order Stimulus Recovery.

3.2 NeuroHardware Realization of the LPF

The answer to the question of neural hardware realizability of the linear low pass filter in Figure 3 turns out to be surprisingly simple. It is inspired by the graph structure of the typical axonal/dendritic tree connectivity. For example, the linear low pass filter in Figure 3 has a parallel connection realization [2] in terms of simple integrators with feedback whose parameters can be arbitrarily set or learned (or are programmable in the language of VLSI). These filters model the synapses and the dendritic tree that, in the parallel implementation,

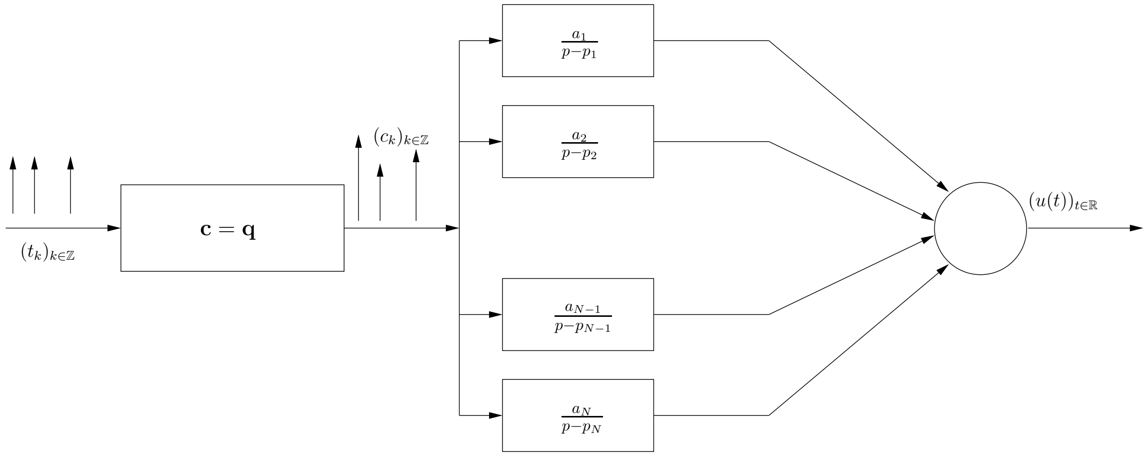

Figure 5: Modeling Stimulus Recovery with a Dendritic Tree.

receive the same input spike train through broadcast (see Figure 5). As before, the primary neuron generates a spike train that is fed into the dendritic tree of a secondary neuron.

The Laplace transform of the (single-input single-output) filter bank described in Figure 5 is given by

$$H(p) = \sum_{k=1}^{N} \frac{a_k}{p - p_k},$$

where the p_k's are the poles and the a_k's are a set of constants for all $k, 1 \leq k \leq N$. This realization is particularly amenable to neural implementation as it consists of a set of parallel integrators with feedback. Each of these filters models the synaptic junction between the axon of a primary neuron and the dendritic tree of a secondary neuron. Clearly, an arbitrarily precise approximation of an ideal low pass filter with unity gain on $[-\Omega, \Omega]$ and zero otherwise can be obtained in this way.

In what follows we shall assume that there is an abundance of integrators for constructing arbitrary filters. While this assumption used to be violated in classical realizability theory of linear time invariant filters [2], it appears to be reasonable in the context of the dense synaptic connectivity that often exists between primary and secondary cortical neurons.

4 Elements of Spike Processing

Building on these observations, the question that we investigate in this section is whether an arbitrary linear operator of the type

$$z(t) = \int_{\mathbb{R}} h(t-s)u(s)ds, \tag{3}$$

can be realized in the spike domain, that is, by directly computing with spikes. Here $h = h(t), t \in \mathbb{R}$, is the impulse response of an arbitrary causal filter. The design of such linear operators is a well established art in the linear systems literature [2]. A biologically inspired implementation in the spike domain is pursued below.

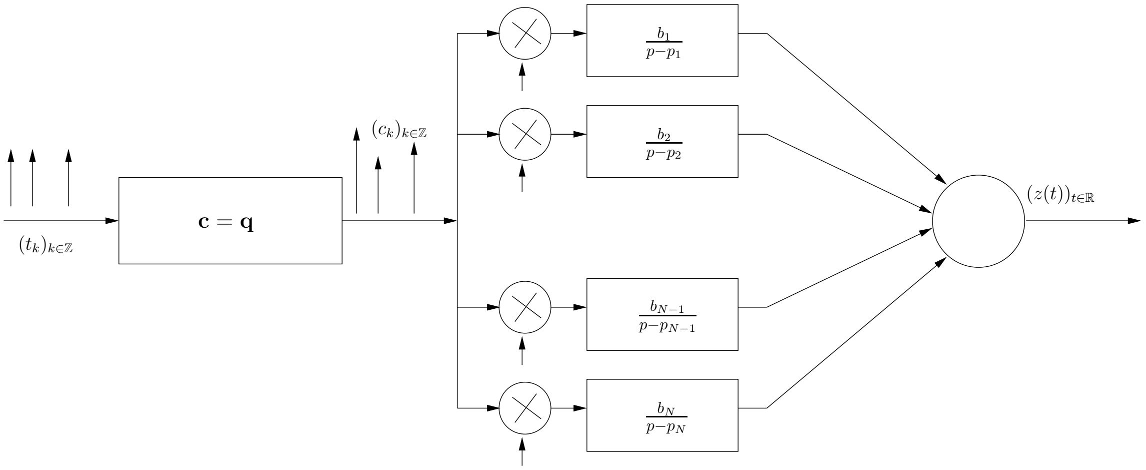

Figure 6: Parallel Connection Realization of an Arbitrary Operator.

A simple realization of such an operator can be achieved in time domain by using methods of linear algebra and linear system theory. Informally, under appropriate conditions,

$$z(t) = \int_{\mathbb{R}} h(t-s) \sum_{k\in\mathbb{Z}} c_k g(s-s_k) ds = \sum_{k\in\mathbb{Z}} c_k \int_{\mathbb{R}} h(t-s) g(s-s_k) ds = \sum_{k\in\mathbb{Z}} c_k f(t-s_k), \quad (4)$$

where f is the impulse response of a filter bandlimited to $[-\Omega, \Omega]$. The linear filter with impulse response f admits, as the low pass filter mentioned in the previous section, a parallel connection realization.

Assume that the Laplace transform of f is given by:

$$F(p) = \sum_{k\in\mathcal{I}} \frac{b_k}{p-p_k},$$

where $\mathcal{I} \subseteq \mathbb{N}$ represents the indices of a subset of integrators. Such a filter can be realized from a large number of parallel integrators with feedback as shown in Figure 6. All integrators that do not belong to the set $\mathcal{I}$ are simply rendered non-operational by inhibition, or equivalently, disconnected. This is obtained in Figure 6 by a multiplication of the input spike train with a one or zero valued signal (mixing). The integrators that belong to the set $\mathcal{I}$ simply remain connected to the upstream (primary) axon. More general transfer functions are also amenable to analysis (complex poles, higher order poles, etc.). Details on the realization methodology can be found in [2], [8]. Therefore, this scheme allows for the realization of arbitrary filters using integrators modeling synaptic connectivity. Finally, we note that the values of the b_k's and p_k's can be preset or more generally obtained through various learning algorithms. Similarly, the mixing signal can be derived from the spike train of the primary or other neurons.

5 Conclusions

By realizing an arbitrary linear operator in the spike domain we have demonstrated that *any sequence* of linear operations on stimuli can be executed in the time domain. The following picture of spike processing emerges for an arbitrary network of integrate-and-fire neurons densely interconnected through synaptic contacts at the axonal/dendritic tree interface.

Integrate-and-fire neurons act as neuro-modulators. They represent analog inputs (corresponding to aggregated dendritic currents) in the spike domain through a process of reversible computation. All irreversible processing in the network takes place in the dendritic tree as time domain linear and/or non-linear operations. The spike train generated by a primary neuron is first decoded by the dendritic tree of a secondary neuron. Linear and/or non-linear operations are then executed on the decoded waveform using simple integrators and inhibition logic.

References

[1] Cash, S. and Yuste, R., *Linear Summation of Excitatory Inputs by CA1 Pyramidal Neurons*, Neuron, Vol. 22(2), pp. 383-94, 1999.

[2] Kailath, T., *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980, pp. 45-48.

[3] Lazar, A.A. and Toth, L.T., *Perfect Recovery and Sensitivity Analysis of Time Encoded Bandlimited Signals*, IEEE Transactions on Circuits and Systems-I: Regular Papers, Vol. 51, No.10, October 2004, pp. 2060-2073.

[4] Lazar, A.A., *Time Encoding with an Integrate-and-Fire Neuron with a Refractory Period*, Neurocomputing, Vol. 58-60, pp. 53-58, June 2004.

[5] Lazar, A.A., *Multichannel Time Encoding with Integrate-and-Fire Neurons*, Neurocomputing, Vol. 65-66, pp. 401-407, 2005.

[6] Lazar, A.A., *Time Encoding Machines with Multiplicative Coupling, Feedforward and Feedback*, submitted for publication.

[7] Rose, G.J. and Call, S.J., *Evidence for the Role of Dendritic Spines in the Temporal Filtering Properties of Neurons: The Decoding Problem and Beyond*, PNAS, Vol. 89, pp. 9662-9665, October 1992.

[8] Sedra, A.S. and Smith, K.C., *Microelectronic Circuits*, Fifth Edition, Oxford University Press, New York, 2004, pp. 1083-1135.