2004 IEEE 11th Digital Signal Processing Workshop
& IEEE Signal Processing Education Workshop

A NEUROMORPHIC IC CONNECTION BETWEEN CORTICAL DENDRITIC PROCESSING AND HMM CLASSIFICATION

Paul Hasler, Paul D. Smith, Ethan Farquhar, David V. Anderson

Georgia Institute of Technology
Electrical and Computer Engineering
Atlanta, GA 30332-0250

ABSTRACT

We show connections between dendritic processing structures and Hidden-Markov Model (HMM) decoding that we arrived through siumultanious circuit design of these systems. From an integrated circuit (IC) toward biology perspective, these simple spreading networks relate well to cable theory and are similar to biological structures such as dendrites and cortical cells. Going in the other direction, from IC to classical digital signal processing (DSP), these structure hold similarities to HMM decoders. In order to implement both structures requires a compact array of variable conductance elements. Using floating-gate transistors we are able to individually vary the conductance of each diffuser element in the array, which dramatically changes the analysis of these arrays.

We believe there is a common framework underlying the computation performed by Hidden Markov Models (HMM) and the computation performed by cortical dendrites. We arrived at this conclusion through initially independant projects exploring each system and, in the process, discovered similarities, first in the common circuit techniques and then within the systems as well. This approach illustrates the design synergy of neuromorphic engineering — biological systems inspire engineering design, and engineering practice inspires biological theory. This research provides a good example of how an integrated-circuit approach can bridge signal processing techniques and neural modeling; only simultaneous construction of these two different IC models made similarities obvious.

Both techniques are based on diffusor networks, originally presented by Boahen, et. al [1]; classic diffusors have a constant conductance and a constant leak term. In particular, we use programmable diffusor networks, built from floating-gate transistors, for which we first presented experimental results elsewhere [2]. Floating-gate elements have been used in a wide variety of applications [3]. In the diffusor network, floating-gate transistors set the conductance of each element individually, thereby cancelling mismatch and allowing a desired conductance to be programmed [4]. The result is that scaling the floating-gate voltages, particularly in a linear fashion, we can choose between classic diffusive behavior [1], described by parabolic PDEs, and forward or backward wave propagation, described by hyperbolic PDEs. Charge on each floating gate determines the wave direction and wave propagation speed.

Given the depth of this subject, we will organize this paper to give an overview of our IC implementations and resulting mathmatical derivations for HMM classification (Section 1), as well as IC implementations of dendritic processing (Section 2) . Experimental and simulation results derive from ICs built in 0.5μm processes (through MOSIS).

1. HMM IC IMPLEMENTATIONS

A Hidden Markov Model (HMM) can be viewed as a state machine in which the states themselves are not observable, but an output, whose statistics are determined by the current state, is observable. For example, in using an HMM to model speech production the states are the desired utterance (phonemes and words) and the observations are features of the audio signal produced by the talker. The audio features are determined by the spoken word but they are randomly distributed since each time that same word is spoken it will sound a little different.

For recognition problems, the goal is to estimate the underlying states of the state machine based on the observed outputs. For speech recognition, the HMM decoder takes as inputs the signal statistics or features and generates a probability of occurrence on any one of a set of speech "symbols." These "symbols" can be grouped over multiple short windows to generate larger symbols, such as phonemes or words. The ongoing input train of symbols is used to map a path through a probability trellis for the larger blocks [5].

We want to revisit the mathematical modeling for HMM classification and translate the formulation to be more easily implemented in continuous-time analog hardware. For the stereotypical speech production HMM (Fig. 1), the likelihood update equation is

$$\phi_i(n) = b_i(n)\left((1-a_i)\phi_i(n-1) + a_{i-1}\phi_{i-1}(n-1)\right), \quad (1)$$

and rewriting in the continous-time case is

$$\phi_i(t) = b_i(t)\big((1 - a_i)\phi_i(t - \tau) + a_{i-1}\phi_{i-1}(t - \tau)\big), \quad (2)$$

where ϕ_i represents the current state at time (t) or time index (n), τ is the time between index (n) and (n-1), δ is the distance between node (i) and (i-1), b_i is the input to the current state and $a_i(n)$ are the transition probabilities between adjacent states. We express a differential-equation Continuous-time formulation by Taylor expand the terms,

$$\phi_i(t - \tau) \approx \phi_i(t) - \tau\frac{d(\phi_i(t))}{dt},$$

$$\phi_{i-1}(t - \tau) \approx \phi_{i-1}(t) - \tau\frac{d(\phi_i(t))}{dt}, \quad (3)$$

which results in the following equation which is consistant through 1^{st}-order terms (both in position and time):

$$\tau\frac{d\phi_i(t)}{dt} + \left(\frac{1}{b_i(t)} - 1\right)\phi_i(t) = a_i\left(\phi_{i-1}(t) - \phi_i(t)\right). \quad (4)$$

Although this equation is sufficient for IC implementation, we expand this equation in position (x) to first order:

$$\underbrace{\tau\frac{d\phi_i(t)}{dt}}_{\text{state element}} + \underbrace{\left(\frac{1}{b_i(t)} - 1\right)\phi_i(t)}_{\text{decay term}} + \underbrace{a_i\delta\frac{d\phi_i(t)}{dx}}_{\text{wave propagation}} = 0. \quad (5)$$

The resulting equation is a wave propagating PDE, where a_i modifies the velocity of the resulting wave, and the input probabilities, $b_i(t)$, set the decay rate of the $\phi_i(t)$ values. The velocity of this wave is $a_i\delta/\tau$.

Figure 1b shows our HMM branch implementation and HMM network implementation. For our implementation, we look at HMMs as propogating waves and the probabilities relate to the velocity of propogation. We can build compact, programmable wave-propagating structures using a floating-gate programmable diffusor circuit with each voltage programmed such that we get either forward or backward propagating waves with minimal diffusion components [4]. A comparison with the terms in (4) shows a direct correspondance to the circuit implementation as

- State Element → Capacitor (storage),
- Wave propagation → Propagation transistor, and
- Decay term → Leak transistor ($b_i(t)$ input).

We implement the $b_i(t)$ terms through log-compressed voltage signals modifying the floating-gate input of the i^{th} leak transistor. We represent $\phi_i(n) = I_i/I_{so}$ for I_i as the leakage current at that node for the reference level of $b(t)$, where I_{so} as a reference current for the array of elements. Although

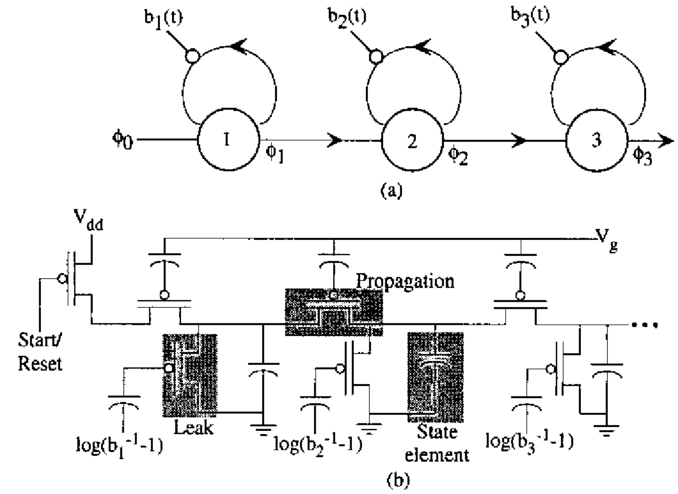

Fig. 1. Circuit design for our HMM branch element. (a) State diagram for a branch of an HMM classifier network typical in speech recognition. (b) Circuit diagram of our HMM network. Our branch element design is based upon diffusor elements to perform the classical HMM calculation. In this framework, each branch element exhibits wave propagation. We can build these branch elements into an array for classification. The shaded areas show the transistors corresponding to terms in the continuous HMM equation.

$b_i(t) = 0$ requires ∞ leakage conductance and $b_i(t) = 1$ requires zero leakage conductance, we practically do not reach these limits, although the variation between these two conductances is typically 9 orders of magnitude.

We can show that τ is a function of the current at that position ($\tau = CU_T/I_i$). At a lower current, the tau will decrease, resulting in the information persisting longer than in a typical HMM network. Unlike a digital system, which has many orders of magnitude available in floating-point representations, analog systems have a finite usable current range. As a result, a lower τ as the signal is decreasing will allow for the signal to exist longer in the system.

2. DENDRITE IC IMPLEMENTATIONS

Dendritic computation is a mixture of digital and analog computing paradigms. Dendrites are a major portion of neurons. They function as the inputs to the cell. Dendrites use complex connectivity to make a wide range of connection morphologies. They do not, however, merely transmit data from pre-synaptic neurons to the cell body to which they are connected; recent experimental measurements suggest 70 percent of power in the brain is supplied at dendrites. Instead, they are instrumental in performing sometimes complex computations. Since synapses (connections between cells) are made on the surface of these cells, dendrites increase the number of connections which can be made while still optimizing the cell to fit into a small space. [7]. As we previously stated, it is the dendrites that carry signals from pre-synaptic cells toward the cell body of the post-

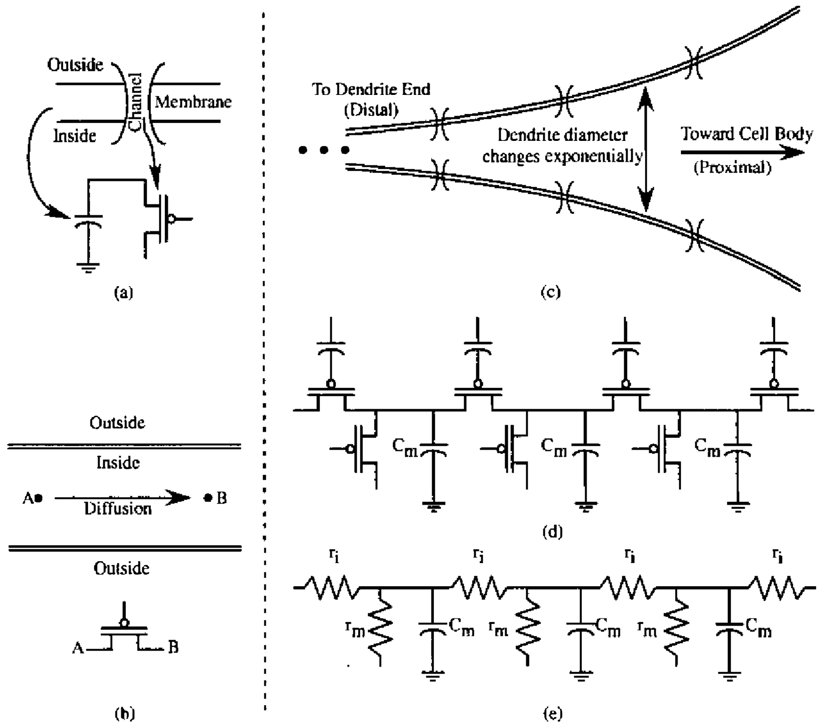

Fig. 2. (a) The membrane of a cell (including the dendrites) is comprised of a two layer lipid and channels which span both layers of the lipid. The channels selectively allow certain ions to pass across, and can be modelled by a transistor [6]. The membrane itself does a very good job of separating charge and has been classically modelled with a capacitor. (b) Inside the cell, ions spread from point A to B by means of diffusion. A subthreshold MOSFET transistor models this well as electrons spread through the channel of a subthreshold MOSFET transistor by diffusion. Putting these pieces together gives us the classic diffusor [1] which is similar to the circuit shown in (c) shows a cartoon picture of a dendrite. The diameter of a dendrite decreases exponentially as you move from the proximal to distal end (from the cell body to the end of the dendrite). (d). Here we have made the diffusive elements Floating Gate elements to greatly reduce the number of wires needed, and to facilitate programming of these voltages. (e) shows the classical view of dendrites that neurobiologists have used to model dendrites. It is similar to our view, however falls short in several areas.

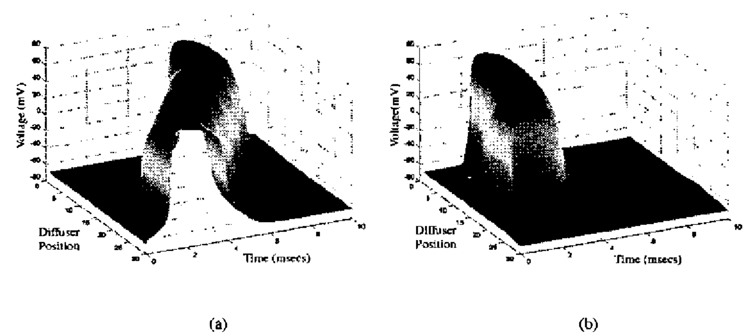

Fig. 3. Simulation data from an active diffuser network under various bias conditions. (a) Network response to a single input at the end of the array, showing propagation of a voltage spike down the entire length of the array. The diffuser voltages are equal throughout the array. (b) Network response with an input similar to "a", however, the leak conductances are linearly increasing from node 0 up to node 30. A input pulse at node 15 only propagates in one direction due to the increase in the diffuser voltages.

synaptic one (the cell that the dendrite is physically a part of). Whether or not the signal from the pre-synaptic neuron is able to generate a response in the post-synaptic cell depends on several factors: Strength of the synapse, Being "boosted" by other excitatory signals, Being suppressed by inhibitory signals, Distance the signal must be transmitted, and Morphology of the dendrite itself.

Looking at Fig. 1 we see the development of our argument. Part (e) shows the classical model of how neurobiologist emulate dendrites. However, it falls short in several areas. It does capture the fact that there is in fact a resistance down the line of the dendrite itself, and that current leaks out through the membrane. However, this model linearizes the system considerably. What really happens? If we look closely at the membrane (a), we can see that we have a membrane which separates the charge, which a capacitor models very nicely. Current is able to actually leak out through protein channels in the membrane. The channels are better modelled with a MOSFET transistor biased in the sub-threshold regime [6]. Axially, (b), ions move from point A to point B through the dendrite by diffusion. Again, this point is better modelled by a sub-threshold MOSFET transistor than a resistor due to the fact that charge transport through a sub-threshold MOSFET transistor is diffusive. Using MOSFET's we are able to overcome another limitation with the classic view. That is, dendrites do not have isometric diameters from the cell body to the distal end. In fact, the diameter changes exponentially. This yields an exponential change in axial conductance, which can be emulated using a linear change on the gate voltages of the diffusion transistors. Our model for the dendrite is found if (d). Therefore, we model a passive dendrite—a dendrite using channels with no voltage dependence—using programmable diffusors.

This dendrite circuit can incorporate active channels based upon similar modeling [2]. Our channel models are based upon transistor channel models instead of the neural modeling paradigm started by Hodgkin and Huxley [6]. In-

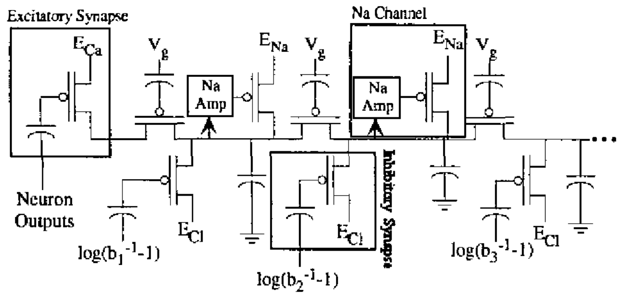

Fig. 4. Circuit design for our HMM branch element with its correlations to transistor modeling of biological channels. The floating-gate elements are programmed to support wave propagation in one direction. The leak elements are models of inhibitory synapses, and the input transistors are models of excitatory synapses. We can include positive feedback in the form of Sodium (Na) channels resulting in signal dependant normalization and pruning.

stead of only simple channels with no voltage-dependence, we add Hodgkin-Huxley type channels in various spatial locations. We can directly match the channel types and dynamics, usually involving combinations of sodium, potassium, and calcium channels, seen in biological dendrites.

Figure 1 shows results for a dendrite using these active dendrites for a single excitatory synapse input. Excitatory, inhibitory, and membrane dependant synapses are efficiently built using these transistor channel modeling approaches [8]. Theoretical descriptions of these channels and elements agree well with SPICE simulation and experimental measurements [2, 9].

3. COMPARISON BETWEEN DENDRITIC COMPUTATION AND HMM CLASSIFICATION

The remaining question is to link these two implementations. Using concepts from IC transistor modeling of dendrites and the IC implementation of an HMM network, we can reformulate a given HMM tree. A diffusive network with many possible bifurcation points biased such that it is a low-spreading wave-propagation network. This effect is achieved in dendrites by an exponentially increasing diameter from distal to proximal parts of the cable, is a direct consequence of (4) for the HMM computation, and is achieved electronically using programmable diffusor elements. The inputs ($b_i(t)$) to the HMM network set leakage conductances to decrease $\phi_i(t)$ over time, which is implemented by allowing $b_i(t)$ to be applied as an input to part of the programmable diffusor network, and is similar to a transistor channel model of an inhibitory synapse [2, 8] using a Cl reveral potential. Many of the external inputs to a layer of cortical pyramidal cells are inhibitory. The starting inputs at the beginning of the HMM branch are similar to the large number of excitatory synapses at the distal end of the dendritic tree. These synapses start of the HMM classification process, and allow for multiple starting conditions as would be necessary for an asynchronous, multiple winner classification process. Adding positive feedback through circuits that model Na channels helps renormalize the signal level.

Further, we see significant comparisions to HMM classifier networks, which will be elaborated in further discussions. In particular, the selection of a particular HMM branch, and therefore its corresponding symbol location, is similar to the Winner-Take-All (WTA) networks between the excitatory pyramidal cells and other inhibitory cells in cortex, and would be implemented by a range of WTA circuits [10]. The reseting function also has strong correlations to the resulting feedback between cortical neurons in cortex

The comparison between existing HMM algorithms and models of dendritic computation opens many opportunities in both areas, opportunities only begun in the writing of this paper. We have a resulting computational model for dendritic computation, that expands well to models of layers of cortical cells. Starting from training HMM classification networks, we have a starting point to discuss the system-level implications of synaptic learning (LTP) in dendrites. The active channels in dendrites provide a range of nonlinear functions to implement signal dependant normalization and pruning in HMM networks.

4. REFERENCES

[1] K. Boahen and A. Andreou, "A contrast-sensitive retina with reciprocal synapses," in *Advances in Neural Information Processing Systems 4*, J. E. Moody, Ed. Morgan Kaufman Publishers, San Mateo, CA, 1991.

[2] Ethan Farquhar and Paul Hasler, "A biophysically based ic model of biological na and k channels," in *Submitted to International Symposium on Circuits and Systems*, Vancouver, BC, 2004.

[3] P. Hasler and T. S. Lande, "Special issue on floating-gate devices, circuits, and systems," *IEEE Journal of Circuits and Systems*, vol. 48, no. 1, Jan. 2001.

[4] Paul Smith and Paul Hasler, "A programmable diffuser circuit based on floating-gate devices," in *Midwest Circuits and Systems*, Tulsa, OK, 2002.

[5] Steve Ranals, Nelson Morgan, Harve Bourlard, Michael Cohen, and Horacio Franco, "Connectionist probability estimators in hmm speech recognition," in *IEEE Transactions On Speech And Audio Processing*, January 1994, vol. 2, pp. 161–174.

[6] Ethan Farquhar, Ed., *A biologically inspired silicon neuron*, M.S. Thesis, Georgia Institute of Technology, jan 2003.

[7] J. C. Fiala and K. M. Harris, Eds., *Dendrite structure*, Oxford University Press, Oxford, 1999.

[8] Christal Gordon, Ethan Farquhar, and Paul Hasler, "A family of floating-gate adapting synapse based upon transistor channel models," in *Submitted to International Symposium on Circuits and Systems*, Vancouver, BC, 2004.

[9] Ethan Farquhar, David Abramson, and Paul Hasler, "A bidirectional active-dendrite model silicon," in *Submitted to International Symposium on Circuits and Systems*, Vancouver, BC, 2004.

[10] John Lazzaro, S. Ryckebusch, M.A. Mahowald, and Carver A. Mead, *Advances in Neural Information Processing Systems 1*, chapter Winner-take-all networks of O(N) complexity, pp. 703–711. Morgan Kaufman Publishers, San Mateo, CA, 1988.