31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

A Biomimetic Adaptive Algorithm and Low-Power Architecture for Implantable Neural Decoders

Benjamin I. Rapoport, *Student Member, IEEE*, Woradorn Wattanapanitch, *Student Member, IEEE*, Hector L. Penagos, Sam Musallam, Richard A. Andersen, and Rahul Sarpeshkar, *Senior Member, IEEE*

***Abstract*—Algorithmically and energetically efficient computational architectures that operate in real time are essential for clinically useful neural prosthetic devices. Such devices decode raw neural data to obtain direct control signals for external devices. They can also perform data compression and vastly reduce the bandwidth and consequently power expended in wireless transmission of raw data from implantable brain-machine interfaces. We describe a biomimetic algorithm and micropower analog circuit architecture for decoding neural cell ensemble signals. The decoding algorithm implements a continuous-time artificial neural network, using a bank of adaptive linear filters with kernels that emulate synaptic dynamics. The filters transform neural signal inputs into control-parameter outputs, and can be tuned automatically in an on-line learning process. We provide experimental validation of our system using neural data from thalamic head-direction cells in an awake behaving rat.**

***Index Terms*—Brain-machine interface, Neural decoding, Biomimetic, Adaptive algorithms, Analog, Low-power**

I. Introduction

BRAIN-MACHINE interfaces have proven capable of decoding neuronal population activity in real-time to derive instantaneous control signals for prosthetics and other devices. All of the decoding systems demonstrated to date have operated by analyzing digitized neural data [1]–[7]. Clinically viable neural prosthetics are an eagerly anticipated advance in the field of rehabilitation medicine, and development of brain-machine interfaces that wirelessly transmit neural data to external devices will represent an important step toward clinical viability. The general model for such devices has two components: a brain-implanted unit directly connected to a multielectrode array collecting raw neural data; and a unit outside the body for data processing, decoding, and control. Data transmission between the two units is wireless. A 100-channel, 12-bit-precise digitization of raw neural waveforms sampled at 30 kHz generates 36 Mbs^{-1} of data; the power costs in digitization, wireless communication, and population signal decoding all scale with this high data rate. Consequences of this scaling, as seen for example in cochlear-implant systems, include unwanted heat dissipation in the brain, decreased longevity of batteries, and increased size of the implanted unit. Recent designs for system components have addressed these issues in several ways. However, *almost no work has been done in the area of power-efficient neural decoding*.

In this work we describe an approach to neural decoding using low-power analog preprocessing methods that can handle large quantities of high-bandwidth analog data, processing neural input signals in a slow-and-parallel fashion to generate low-bandwidth control outputs.

Multiple approaches to neural signal decoding have been demonstrated by a number of research groups employing highly programmable, discrete-time, digital algorithms, implemented in software or microprocessors located outside the brain. We are unaware of any work on continuous-time analog decoders or analog circuit architectures for neural decoding. The neural signal decoder we present here is designed to complement and integrate with existing approaches. Optimized for implementation in micropower analog circuitry, it sacrifices some algorithmic programmability to reduce the power consumption and physical size of the neural decoder, facilitating use as a component of a unit implanted within the brain. Trading off the flexibility of a general-purpose digital system for the efficiency of a special-purpose analog system may be undesirable in some neural prosthetic devices. Therefore, our proposed decoder is meant to be used not as a substitute for digital signal processors but rather as an adjunct to digital hardware, in ways that combine the efficiency of embedded analog preprocessing options with the flexibility of a general-purpose external digital processor.

For clinical neural prosthetic devices, the necessity of highly sophisticated decoding algorithms remains an open question, since both animal [3], [4], [8], [9] and human [5] users of even first-generation neural prosthetic systems have proven capable of rapidly adapting to the particular rules governing the control of their brain-machine interfaces. In the present work we focus on an architecture to implement a simple, continuous-time analog linear (convolutional) decoding algorithm. The approach we present here can be generalized to implement analog-circuit architectures of general Bayesian algorithms; examples of related systems include analog probabilistic decoding circuit architectures used in speech recognition and error correcting codes [10], [11]. Such architectures can be extended through our mathematical approach to design circuit architectures for Bayesian decoding.

B. I. Rapoport, W. Wattanapanitch, and R. Sarpeshkar are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts 02139 USA; B. I. Rapoport is also with the Harvard–MIT Division of Health Sciences and Technology, Cambridge, Massachusetts 02139 and Harvard Medical School, Boston, Massachusetts 02115; E-mail: rahuls@mit.edu.

H. L. Penagos is with the MIT Department of Brain and Cognitive Sciences and the Harvard–MIT Division of Health Sciences and Technology.

S. Musallam is with the Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec, Canada.

R. A. Andersen is with the Division of Biology, California Institute of Technology, Pasadena, California 91125.

Manuscript received April 2009.

II. A Biomimetic Adaptive Algorithm for Decoding Neural Cell Ensemble Signals

In convolutional decoding of neural cell ensemble signals, the decoding operation takes the form

$$\vec{M}(t) = \mathbf{W}(t) \circ \vec{N}(t) \tag{1}$$

$$M_i(t) = \sum_{j=1}^{n} W_{ij}(t) \circ N_j(t);\ i \in \{1, \ldots, m\}, \tag{2}$$

where $\vec{N}(t)$ is an n-dimensional vector containing the neural signal (n input channels of neuronal firing rates, analog signal values, or local field potentials, for example) at time t; $\vec{M}(t)$ is a corresponding m-dimensional vector containing the decoder output signal (which in the examples presented here corresponds to motor control parameters, but could correspond as well to limb or joint kinematic parameters or to characteristics or states of nonmotor cognitive processes); $\mathbf{W}$ is a matrix of convolution kernels $W_{ij}(t)$ (formally analogous to a matrix of dynamic synaptic weights), each of which depends on a set of p modifiable parameters, $W_{ij}^k,\ k \in \{1, \ldots, p\}$; and $\circ$ indicates convolution. Accurate decoding requires first choosing an appropriate functional form for the kernels and then optimizing the kernel parameters to achieve maximal decoding accuracy. Since the optimization process is generalizable to any choice of kernels that are differentiable functions of the tuning parameters, we discuss the general process first. We then explain our biophysical motivations for selecting particular functional forms for the decoding kernels; appropriately chosen kernels enable the neural decoder to emulate the real-time encoding and decoding processes performed by biological neurons.

Our algorithm for optimizing the decoding kernels uses a gradient-descent approach to minimize decoding error in a least-squares sense during a learning phase of decoder operation. During this phase the correct output $\hat{\vec{M}}(t)$, and hence the decoder error $\vec{e}(t) = \vec{M}(t) - \hat{\vec{M}}(t)$, is available to the decoder for feedback-based learning. We design the optimization algorithm to evolve $\mathbf{W}(t)$ in a manner that reduces the squared decoder error on a timescale set by the parameter τ, where the squared error is defined as

$$E(\mathbf{W}(t), \tau) = \int_{t-\tau}^{t} |\vec{e}(u)|^2 du \tag{3}$$

$$= \sum_{i=1}^{m} \int_{t-\tau}^{t} |\vec{e}_i(u)|^2 du \equiv \sum_{i=1}^{m} E_i, \tag{4}$$

and the independence of each of the m terms in Equation 4 is due to the the independence of the m sets of np parameters $W_{ij}^k,\ j \in \{1, \ldots, n\}\ k \in \{1, \ldots, p\}$ associated with generating each component $M_i(t)$ of the output. Our strategy for optimizing the matrix of decoder kernels is to modify each of the kernel parameters W_{ij}^k continuously and in parallel, on a timescale set by τ, in proportion to the negative gradient of $E(\mathbf{W}(t), \tau)$ with respect to that parameter:

$$-\vec{\nabla}_{ij}^k E(\mathbf{W}(t), \tau) \equiv -\frac{\partial E}{\partial W_{ij}^k} \tag{5}$$

$$= -\sum_{l=1}^{m} \int_{t-\tau}^{t} du \left\{ 2 \left(M_l(u) - \sum_{j=1}^{n} W_{lj}(u) \circ N_j(u) \right) \times \left(-\frac{\partial W_{ij}(u)}{\partial W_{ij}^k} \circ N_j(u) \right) \right\} \tag{6}$$

$$= 2 \sum_{l=1}^{m} \int_{t-\tau}^{t} e_l(u) \left(\frac{\partial W_{ij}(u)}{\partial W_{ij}^k} \circ N_j(u) \right) du. \tag{7}$$

The learning algorithm refines $\mathbf{W}$ in a continuous-time fashion, using $-\vec{\nabla} E(t)$ as an error feedback signal to modify $\mathbf{W}(t)$, and incrementing each of the parameters $W_{ij}^k(t)$ in continuous time by a term proportional to $-\vec{\nabla}_{ij}^k E(\mathbf{W}(t))$ (the proportionality constant, ϵ, must be large enough to ensure quick learning but small enough to ensure learning stability). If $\mathbf{W}(t)$ is viewed as an array of linear filters operating on the neural input signal, the quantity $-\vec{\nabla}_{ij}^k E(\mathbf{W}(t), \tau)$ used to increment each filter parameter can be described as the product, averaged over a time interval of length τ, of the error in the filter output and a secondarily filtered version of the filter input. The error term is identical for the parameters of all filters contributing to a given component of the output, $M_i(t)$. The secondarily filtered version of the input is generated by a secondary convolution kernel, $\frac{\partial W_{ij}(u)}{\partial W_{ij}^k}$, which depends on the functional form of each primary filter kernel and in general differs for each filter parameter. Figure 1 shows a block diagram for an analog circuit architecture that implements our decoding and optimization algorithm.

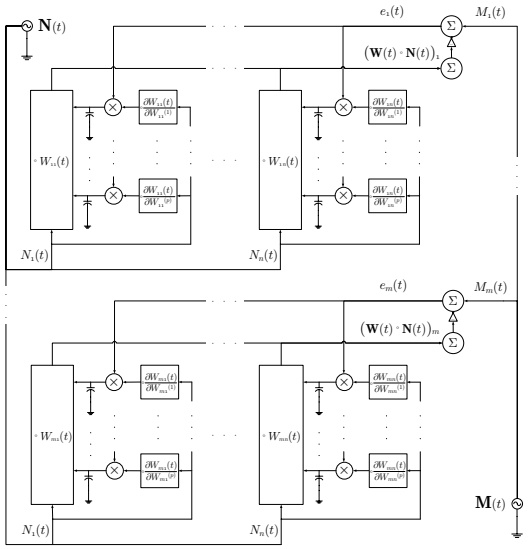

Fig. 1: Block diagram of a computational architecture for linear convolutional decoding and learning.

Many functional forms for the convolution kernels are both theoretically possible and practical to implement using low-power analog circuitry. Our approach has been to emulate

biological neural systems by choosing a biophysically inspired kernel whose impulse response approximates the postsynaptic currents biological neurons integrate when encoding and decoding neural signals *in vivo* [12]. Combining our decoding architecture with the choice of a first-order low-pass decoder kernel enables our low-power neural decoder to implement a biomimetic, continuous-time artificial neural network. Numerical experiments have also indicated that decoding using such biomimetic kernels can yield results comparable to those obtained using optimal linear decoders [13]. But in contrast with our on-line optimization scheme, optimal linear decoders are computed off-line after all training data have been collected. We have found that this simple choice of kernel offers effective performance in practice, and so we confine the present analysis to that kernel.

Two-parameter first-order low-pass filter kernels account for trajectory continuity by exponentially weighting the history of neural inputs:

$$W_{ij} = \frac{A_{ij}}{\tau_{ij}} e^{-\frac{t}{\tau_{ij}}}, \tag{8}$$

where the two tunable kernel parameters are $W_{ij}^{k=1} = A_{ij}$, the low-pass filter gain, and $W_{ij}^{k=2} = \tau_{ij}$, the decay time over which past inputs $\vec{N}(t')$, $t' < t$, influence the present output estimate $\vec{M}(t) = \mathbf{W} \circ \vec{N}(t)$. The filters used to tune the low-pass filter kernel parameters can be implemented using simple and compact analog circuitry. The gain parameters are tuned using low-pass filter kernels of the form

$$\frac{\partial W_{ij}(t)}{\partial W_{ij}^{k=1}} = \frac{1}{\tau_{ij}} e^{-\frac{t}{\tau_{ij}}}, \tag{9}$$

while the time-constant parameters are tuned using band-pass filter kernels:

$$\frac{\partial W_{ij}(t)}{\partial W_{ij}^{k=2}} = \frac{A_{ij}}{\tau_{ij}^2} e^{-\frac{t}{\tau_{ij}}} \left(\frac{t}{\tau_{ij}} - 1 \right). \tag{10}$$

When decoding discontinuous trajectories, such as sequences of discrete decisions, we can set the τ_{ij} to zero, yielding

$$W_{ij}(t) = W_{ij}^{k=1} \delta(t) = A_{ij} \delta(t). \tag{11}$$

Such a decoding system, in which each kernel is a zeroth-order filter characterized by a single tunable constant, performs instantaneous linear decoding, which has successfully been used by others to decode neuronal population signals in the context of neural prosthetics [5], [14]. With kernels of this form, $\mathbf{W}(t)$ is analogous to matrices of synaptic weights encountered in artificial neural networks, and our optimization algorithm resembles a 'delta-rule' learning procedure [15].

III. Results

Head direction was decoded from the activity of $n = 6$ isolated thalamic neurons according to the method described in [16]. The adaptive filter parameters $W_{ij}^{(p)} \in \{A_{ij}, \tau_{ij}\}$ were implemented as micropower analog circuits and simulated in SPICE; they were optimized through gradient descent over training intervals of length T during which the decoder error, $e_i(t) = M_i(t) - \hat{M}_i(t)$ (where $\vec{M}(t) = (\cos(\theta(t)), \sin(\theta(t)))$ and θ denotes the head direction angle), was made available to the adaptive filter in the feedback configuration described in Section II for $t \in [0, T]$. Following these training intervals feedback was discontinued and the performance of the decoder was assessed by comparing the decoder output $\vec{M}(t)$ with $\hat{\vec{M}}(t)$ for $t > T$.

Figure 2 compares the output of the decoder to the measured head direction over a 240 s interval. The filter parameters were trained over the interval $t \in [0, T = 120]$ s. The figure shows $\vec{M}(t)$ (gray) tracking $\hat{\vec{M}}(t)$ (black) with increasing accuracy as training progresses, illustrating that while initial predictions are poor, they improve with feedback over the course of the training interval. Feedback is discontinued at $t = 120$ s. Qualitatively, the plots on the interval $t \in [120, 240]$ s illustrate that the output of the neural decoder reproduces the shape of the correct waveform, predicting head direction on the basis of neuronal spike rates.

IV. Discussion

Simulations using basic circuit building-blocks for the modules shown in Figure 1 indicate that a single decoding module (corresponding to an adaptive kernel W_{ij} and associated optimization circuitry, as diagrammed in Figure 1) should consume approximately 54 nW from a 1 V supply in 0.18 μm CMOS technology and require less than 3000 μm^2. Low power consumption is achieved through the use of subthreshold bias currents for transistors in the analog filters and other components. Analog preprocessing of raw neural input waveforms is accomplished by dual thresholding to detect action potentials on each input channel and then smoothing the resulting spike trains to generate mean firing rate input signals. SPICE simulations indicate that each analog preprocessing module should consume approximately 241 nW from a 1 V supply in 0.18 μm CMOS technology. A full-scale system with $n = 100$ neuronal inputs comprising $\vec{N}(t)$) and $m = 3$ control parameters comprising $\vec{M}(t)$ would require $m \times n = 300$ decoding modules and consume less than 17 μW in the decoder and less than 25 μW in the preprocessing stages.

Direct and power-efficient analysis and decoding of analog neural data within the implanted unit of a brain-machine interface could also facilitate extremely high data compression ratios. For example, the 36 Mbs^{-1} required to transmit raw neural data from 100 channels could be compressed more than 100,000-fold to 300 bs^{-1} of 3-channel motor-output information updated with 10-bit precision at 10 Hz. Such dramatic compression brings concomitant reductions in the power required for communication and digitization of neural data. Ultra-low-power analog preprocessing prior to digitization of neural signals could thus be beneficial in some applications.

V. Conclusions

The algorithm and architecture presented here offer a practical approach to computationally efficient neural signal decoding, independent of the hardware used for their

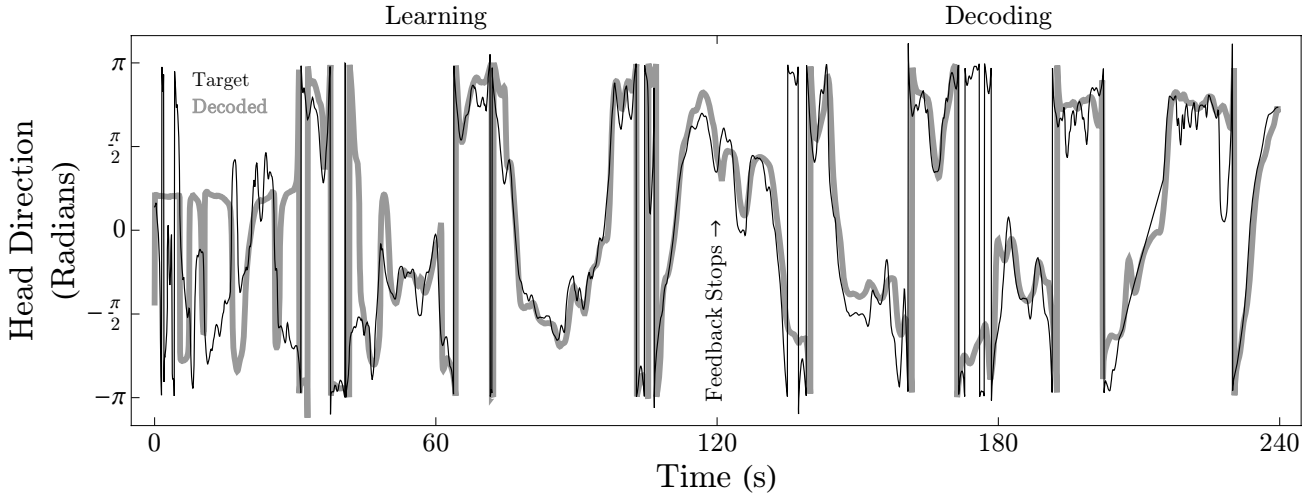

Fig. 2: Continuous decoding of head direction from neuronal spiking activity.

implementation. While the system is suitable for analog or digital implementation, we suggest that a micropower analog implementation trades some algorithmic programmability for reductions in power consumption that could facilitate implantation of a neural decoder within the brain. In particular, circuit simulations of our analog architecture indicate that a 100-channel, 3-motor-output neural decoder can be built with a total power budget of approximately $43\ \mu$W. Our work could also enable a $100,000$-fold reduction in the bandwidth needed for wireless transmission of neural data, thereby reducing to nanowatt levels the power potentially required for wireless data telemetry from a brain implant. Our work suggests that highly power-efficient and area-efficient analog neural decoders that operate in real time can be useful components of brain-implantable neural prostheses, with potential applications in neural rehabilitation and experimental neuroscience. Through front-end preprocessing to perform neural decoding and data compression, algorithms and architectures such as those presented here can complement digital signal processing and wireless data transmission systems, offering significant increases in power and area efficiency at little cost.

ACKNOWLEDGMENTS

This work was funded in part by National Institutes of Health grants R01-NS056140 and R01-EY15545, the McGovern Institute Neurotechnology Program at MIT, and National Eye Institute grant R01-EY13337. Rapoport received support from a CIMIT–MIT Medical Engineering Fellowship.

REFERENCES

[1] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. L. Nicolelis. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neuroscience*, 2:664–670, 1999.

[2] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. L. Nicolelis. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408:361–365, November 2000.

[3] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz. Direct cortical control of 3d neuroprosthetic devices. *Science*, 296:1829–1832, June 2002.

[4] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen. Cognitive control signals for neural prosthetics. *Science*, 305:258–262, July 2004.

[5] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171, July 2006.

[6] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy. A high-performance brain-computer interface. *Nature*, 442:195–198, July 2006.

[7] A. Jackson, J. Mavoori, and E. E. Fetz. Long-term motor cortex plasticity induced by an electronic neural implant. *Nature*, 444:55–60, November 2006.

[8] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. ODoherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis. Learning to control a brain-machine interface for reaching and grasping by primates. *Public Library of Science Biology*, 1(2):1–16, October 2003.

[9] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz. Cortical control of a prosthetic arm for self-feeding. *Nature*, 453(7198):1098–1101, June 2008.

[10] John Lazzaro, John Wawrzynek, and Richard P. Lippmann. A micropower analog circuit implementation of hidden markov model state decoding. *IEEE Journal of Solid-State Circuits*, 32(8):1200–1209, August 1997.

[11] Hans-Andrea Loeliger, Felix Tarköy, Felix Lustenberger, and Markus Helfenstein. Decoding in Analog VLSI. *IEEE Communications Magazine*, pages 99–101, April 1999.

[12] A. Arenz, R. A. Silver, A. T. Schaefer, and T. W. Margrie. The contribution of single synapses to sensory representation in vivo. *Science*, 321:977–980, August 2008.

[13] C. Eliasmith and C. H. Anderson. *Neural Engineering*, chapter 4, pages 112–115. MIT Press, 2003.

[14] J. Wessberg and M. A. L. Nicolelis. Optimizing a linear algorithm for real-time robotic control using chronic cortical ensemble recordings in monkeys. *Journal of Cognitive Neuroscience*, 16(6):1022–1035, 2004.

[15] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prendice Hall, Upper Saddle River, New Jersey, 1999.

[16] R. Barbieri, L. M. Frank, M. C. Quirk, M. A. Wilson, and E. N. Brown. A Time-Dependent Analysis of Spatial Information Encoding in the Rat Hippocampus. *Neurocomputing*, 32-33:629–635, 2000.