34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Applying Best Practices from Digital Control Systems to BMI Implementation

Charlie Matlack[1], Chet Moritz[2] and Howard Chizeck[1]

Abstract— **Many brain-machine interface (BMI) algorithms, such as the population vector decoder, must estimate neural spike rates before transforming this information into an external output signal. Often, rate estimation is performed via the selection of a bin width corresponding to the effective sampling rate of the decoding algorithm. Here, we implement real-time rate estimation by extending prior work on the optimization of Gaussian filters for offline rate estimation. We show that higher sampling rates result in improved spike rate estimation. We further show that the choice of sampling rate need not dictate the number of parameters which must be used in an autoregressive decoding algorithm. Multiple studies in other neural signal processing contexts suggest that BMI performance could be improved substantially via careful choice of smoothing filter, discrete-time decoder representation, and sampling rate. Together, these ensure minimal deviation from the behavior of the modeled continuous-time systems.**

I. Introduction

Brain-machine interfaces mapping activity of well-isolated neurons or multiunits in cortex have shown promise for allowing the continuous control of robotic limbs and computerized assistive devices [8]. Original single-neuron BMI experiments required the use of analog circuitry to implement low-pass filters for neural spike rate estimation [4]. This was an encumbrance that limited design flexibility. As recently as 2004, studies investigating the effect of sampling rate on digital BMI performance were limited by the available computing hardware [16]. Enabled by ever more compact and powerful computing hardware, contemporary algorithms for converting the recorded activity of large populations of individual neurons into movement commands are increasingly complex, drawing on tools from control theory and machine learning such as Wiener Filters [12], [16], Support Vector Machines [6], [15], and Kalman filters [7], [9], [10]. Sampling rates for the digitally implemented dynamics of these systems remain as low as 10 Hz [1], [2], [10], [13]. By comparison, modern action-oriented computer games typically refresh the screen at 30 Hz or above and simulate underlying game physics at even higher sampling rates in order to achieve a sense of realism.

Many BMI architectures implement spike rate estimation and decoders such that the behavior of both is driven by the choice of a shared sampling rate. This need not be the case. Consequently, there is an opportunity for improvement by increasing sample rates that has been overlooked thus far.

Our use of the term *decoder* refers broadly to the mathematical transform between spike rate estimates and BMI output signals, and is inclusive of transforms which do not attempt to literally decode limb movement.

Here we demonstrate three advances. In Section II, we show that the use of a Gaussian smoother with optimized bandwidth outperforms the time-bin (histogram) method in spike rate estimation. In Section III, we discuss the benefits of high sampling rates and illustrate with an example of spike rate estimation performance. In Section IV, we show the benefits of choosing a model with the minimum number of parameters to capture dynamics. And in Section V, we describe how down-sampling can be used to implement such a model with a high sampling rate.

II. Spike Rate Estimation with Smoothing Filters

Spiking neurons are well-modeled as inhomogenous Poisson processes, making it possible to estimate the underlying time-varying distribution parameter using a variety of methods [14]. The vast majority of BMI designs estimate instantaneous neural firing rates by counting the number of spike events in a temporal bin, with widths typically in the range of 30 to 100 ms. This histogram process is equivalent to two separate steps: filtering using a rectangular kernel, then sampling with a sample period equal to the kernel width. This is illustrated in Fig. 1.

This simple method produces an estimate because, for a time-invariant Poisson process, the expected number of

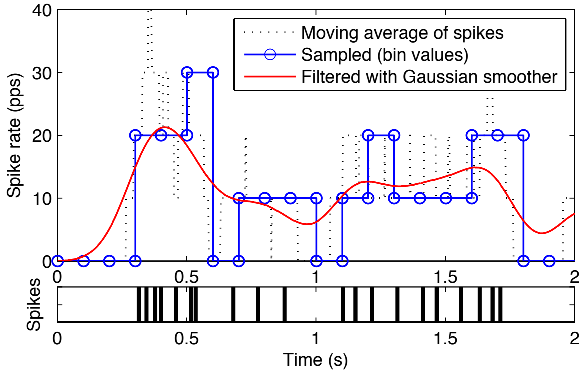

Fig. 1. **Illustration of the two underlying components of the histogram spike rate estimation method: filtering with a rectangular moving average, followed by sampling with sample period equal to window width, 100ms. For comparison, the output of a Gaussian smoothing filter with σ = 100ms is shown as well.**

This work is supported through the NSF Center for Sensorimotor Neural Engineering, Award Number EEC-1028725, and American Heart & Stroke Association Scientist Development Grant, Award Number 09SDG2230091.

[1] Electrical Engineering Department, University of Washington, Seattle, WA, USA {cmatlack,chizeck}@uw.edu

[2] Departments of Rehabilitation Medicine and Physiology & Biophysics, University of Washington, Seattle, WA, USA ctmoritz@uw.edu

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

events in a given time interval is proportional to the distribution parameter. This estimate is still rather noisy, and in some designs is fed into another moving average filter [12], [16].

The rectangular kernel can be replaced with an arbitrary kernel function, and the sampling rate need not be the inverse of the kernel width. Shimazaki and Shinomoto (2009) developed an algorithm for selecting the optimal width of a given symmetric kernel function by minimizing the mean integrated square error (MISE) between the rate estimate and unknown underlying rate [14],

$$MISE = \int E(\hat{\lambda}_t - \lambda_t)^2 dt \qquad (1)$$

Their algorithm utilizes sample data as a proxy for the statistics of unknown rate λ_t to optimize rate estimate $\hat{\lambda}_t$.

They showed that using a Gaussian kernel significantly outperformed the histogram method when both had widths optimized for offline spike rate estimation. Their performance comparison of different rate estimation methods using synthesized neural recordings is shown in Fig. 2. The Gaussian kernel is also an attractive choice for real-time applications because it has minimal rise and fall time with no overshoot. For a more in-depth discussion of methods of spike rate estimation, see [3].

The optimization algorithm generates the optimal kernel width for a specific data set. Therefore, to find the kernel width best suited for a particular BMI application, a data set which captures neural firing statistics during BMI operation is needed.

Moritz and Fetz (2011) showed that modulation depth of neural firing can vary substantially between tasks performed manually and under single-neuron BMI control by macaque monkeys [11]. This means that optimizing a kernel based on recorded data prior to BMI control may not result in good performance under BMI control. To bootstrap the system, we can use a data set recorded during manual tasks or while the animal is at rest. Once data recorded during BMI control is available, a sufficiently large duration of recordings should be used to capture the full range of firing behavior and re-select the optimal kernel bandwidth. Fig. 3 shows the resultant MISE cost functions for neural firing recorded from the same neuron under different conditions. We can see that the cost function is relatively flat for a wide range of kernel widths, regardless of recording context. In choosing a kernel width for better robustness, an experimenter can select one that may not correspond to the global minimum for the most recent data set, but will result in an acceptable MISE. As a low-pass filter, the Gaussian smoother places limits on the bandwidth of the entire BMI system and also introduces a delay, creating an incentive to choose the smallest kernel width that results in an acceptable error. The threshold near zero width at which the cost function begins to increase dramatically is fairly consistent among recording contexts, indicating that a kernel width near this threshold should continue to perform well as neural modulation changes with BMI use. Thus, firing rate estimation variance can be robustly reduced by the replacement of a histogram estimator with a Gaussian smoothing filter. Gaussian kernel smoothing is optimal in terms of both *latency* and firing rate estimation accuracy compared to other moving-window smoothing filters, and no more computationally demanding in on-line implementation than the rectangular filter. In Section III, we address the digital implementation of a Gaussian filter.

III. Digital Implementation of Dynamic Systems

Multiple rules of thumb exist for selecting a sufficient sampling rate for the discrete-time representation of a dynamic system. In order to avoid aliasing (signal artifacts that result from too low a sampling frequency), the signal must be sampled faster than the Nyquist rate. The Nyquist rate is defined as two times the bandwidth, or maximum frequency, of a signal or dynamic system response. The Nyquist rate

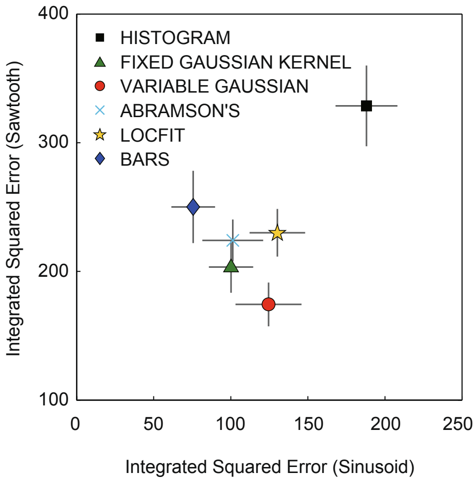

Fig. 2. **Performance comparison of six rate estimation methods based on MISE when applied to synthesized data with sinusoidal and sawtooth underlying rate functions. Modified from [14].**

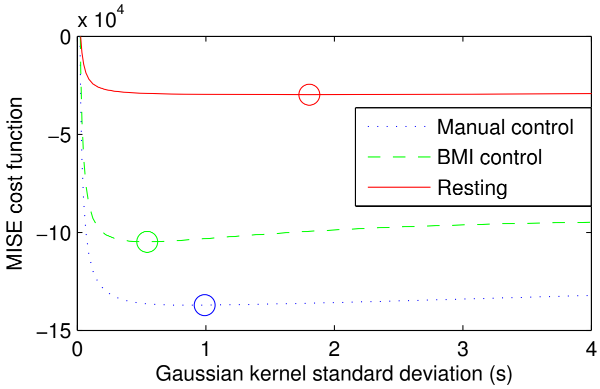

Fig. 3. **Estimated MISE as a function of Gaussian kernel bandwidth for the same neuron in different recording conditions. Optimal bandwidths, marked by open circles, are different, but in relatively flat regions of the cost function. Sample data sets are 1- to 3-minute single neuron recordings from microwire array in motor cortex of a macaque monkey. Recorded with a Cerebus Neural Signal Processor (Blackrock Microsystems, Salt Lake City, UT). See [11] for additional details of recording and task configuration.**

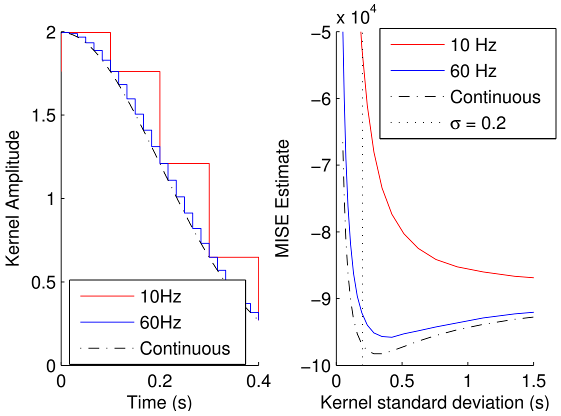

Fig. 4. **Comparison of continuous and sampled Gaussian kernels, $\sigma = 0.2$s, at left, and MISE cost as a function of standard deviation at right. Sampling rate affects both the minimum achievable MISE and the optimal bandwidth to achieve minimum MISE. Note that this analysis does not account for the latency in the firing rate estimate introduced by the use of a causal filter. Sample data recorded from motor cortex of macaque during 10-minute brain control session.**

represents an absolute *lower bound* on sampling rate for reconstruction of the original signal or system dynamics; sampling at or marginally above this rate, as is sometimes done [10], [16], does not result in *good* reconstruction up to the bandwidth frequency. In fact, it is desirable to sample 10 or more times faster than the bandwidth, because this factor represents the number of sample points used to represent one cycle of a sinusoidal signal.

For example, Figure 4 illustrates the effect of sampling rate selection on the MISE associated with a Gaussian filter sampled at different rates and truncated at $\pm 2\sigma$ in order to reduce latency. Gaussian filters have a cutoff frequency of $1/\sigma$ where σ is the standard deviation of the kernel, making the Nyquist sampling rate $2/\sigma = 10$Hz in this example. We can see that MISE is significantly improved at a sampling rate of 60Hz, which is 12 times the bandwidth.

We have chosen an appropriate sampling rate for a Gaussian filter based on an application-specific criteria, without regard to the bandwidth of the incoming spike train. In general, it is essential to select a sampling rate which captures the full bandwidth of the input signal as well as the dynamic response of the system. This will be our approach to implementing the decoder following the rate estimation filter.

By virtue of being the solution to the rate estimator optimization problem, the bandwidth of the Gaussian filter provides an upper bound on the bandwidth of the firing rate estimate signal. More information may exist in the spike activity, but we implicity ignore it as noise. We maintain the sampling rate of 60Hz for the decoder, and in the next section we address the system identification problem of literally decoding limb movement.

IV. Model Selection

Dynamic models are often used to decode limb movement from neural firing, and subsequently to map neural firing to BMI output signals. Moving-average architectures such as the Wiener filter have been widely used [1], [12], and more recently Kalman filtering methods based on state-space models have gained popularity [3], [10].

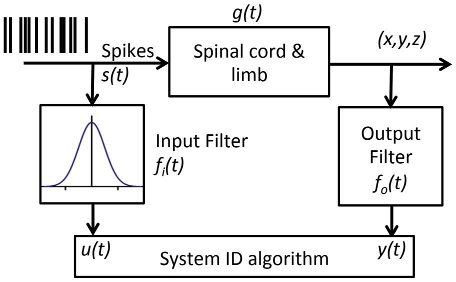

Fig. 5. **Block diagram illustrating the system identification problem for decoding limb movement. The relationship between $u(t)$ and $y(t)$ is only equivalent to the spinal cord and limb transformation if the output filter is the same as the input smoothing filter.**

The block diagram in Fig. 5 illustrates the system identification problem for decoding limb movement. Consider the frequency-domain transformations from the spike train $S(s)$ to $U(s)$ and $Y(s)$,

$$\begin{aligned} U(s) &= F_i(s)S(s) \\ Y(s) &= F_o(s)G(s)S(s) \\ \frac{Y(s)}{U(s)} &= G(s)\frac{F_o(s)}{F_i(s)} \end{aligned}$$

We can see that in order for the relationship between the recorded input $u(t)$ and output $y(t)$ to be equal to the unknown transformation from neural spiking to limb movement $g(t)$, the two filters in the recording pathways, $f_i(t)$ and $f_o(t)$, must be identical. Although this approach appears to discard information from the output signal, remember that the smoothing filter implicitly considers signal energy above its bandwidth to be noise. Note that we are sampling a tiny subset of the neural activity driving low-level control of muscle recruitment. Therefore, it is possible that the spike rate signal has higher bandwidth than observable gross limb movement. We can bound the bandwidth of the dynamics relating neural firing rate to limb movement by the bandwidth of the observed output signal, limb trajectories during motor activity.

In order to build a higher-bandwidth model of the relationship, it would be necessary to obtain a higher-bandwidth input signal, such as by incorporating the firing of additional neurons.

In selecting a model architecture either for directly transforming neural activity or for use in a Kalman filter, it is important be aware of the implications of the model structure and of the number of free parameters. Specifically, a higher number of model parameters will be required to capture higher-order dynamics, but requires substantially more data for a fitting algorithm to arrive at good parameter estimates. Therefore, the experimenter should choose a model with the minimum number of parameters required to capture the dominant dynamics thought to exist in the system. Autoregressive moving average (ARMA) models have been shown to perform well among other linear models [5]. A dynamic system of a given order can be modeled with less parameters with the ARMA structure than via a state-space or impulse-response formulation.

V. Decoder Implementation

Because a small number of model parameters will be fit to a large data set with noise, the sampling rate of the model must be chosen so that the dynamics capture the underlying system behavior while ignoring the noise. This motivates the selection of a sampling rate only a few times greater than the bandwidth of the *signal* content of the recorded data. The bandwidth of limb trajectories and the optimal kernel bandwidth both suggest choices for the model bandwidth. To convert the 60 Hz sampled signal from the smoothing filter to, for example, a 10 Hz dynamic model (Nyquist rate of 20Hz), we can down-sample, as shown in Fig. 6. In this case, even if only every third sample is used by the dynamic model at a given instant, the prior Gaussian smoothing incorporates a significantly larger temporal window into this data. Note that the 20 Hz model is still *updated* at 60 Hz, but it models dynamics below 10 Hz while ignoring higher frequencies in the input signal.

Finally, the output command from the brain will drive a visual feedback signal. As mentioned previously, 30 Hz is at the lower end of refresh rates used for computer games, and should be a reasonable rate for this application. Most computer displays now support a refresh rate of 60 Hz or above. In the absence of other hardware constraints on computation, there is no reason not to implement the system at a base sampling rate of 60 Hz. Thus, a model implementation allowing a small number of parameters to capture desired low-bandwidth dynamics can be implemented in a digital system with a high sampling rate for smooth and responsive visual feedback.

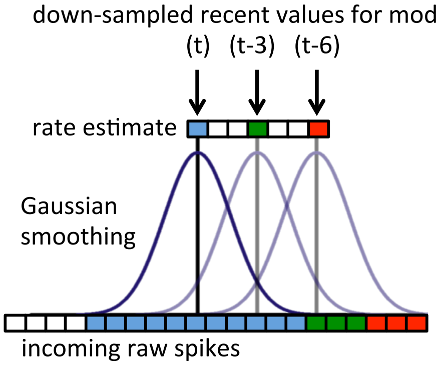

Fig. 6. **Illustration of down-sampling spike rate estimate to increase temporal window of data available to model while decreasing redundancy of sample information. The transparent Gaussian kernels and green and red samples represent delayed copies of the current filter output (blue) used by the model.**

VI. Conclusion

We have described three areas for improvement in BMI systems which can be implemented without significant additional computational requirements or changes to the underlying models used to decode brain activity: (1) Gaussian kernel smoothing using a kernel width optimized for observed cell behavior provides the best firing rate estimate with the lowest latency among other choices of smoothing filters; (2) careful choice of dynamic model structure and number of parameters can improve the performance of decoding algorithms, potentially leading to improved performance of decoder-based BMIs; and (3) implementing the full system at a sampling rate of 60 Hz will allow faithful reproduction of the underlying continuous-time dynamics and provide seamless and smooth visual feedback to the subject. Together, these improvements should facilitate substantial performance improvement of existing BMI designs.

References

[1] J. M. Carmena and K. Ganguly. Emergence of a Stable Cortical Map for Neuroprosthetic Control. *PLoS Biology*, 7(7):e1000153, 2009.

[2] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, 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. *PLoS Biology*, 1(2):E42, Nov. 2003.

[3] J. P. Cunningham, V. Gilja, S. I. Ryu, and K. V. Shenoy. Methods for estimating neural firing rates, and their application to brain-machine interfaces. *Neural Networks: The Official Journal of the International Neural Network Society*, 22(9):1235–46, Nov. 2009.

[4] E. E. Fetz. Operant Conditioning of Cortical Unit Activity. *Science*, 163(3870):955–958, 1969.

[5] J. Fisher and M. Black. Motor cortical decoding using an autoregressive moving average model. *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, 2:2130–3, Jan. 2005.

[6] G. J. Gage, K. a. Ludwig, K. J. Otto, E. L. Ionides, and D. R. Kipke. Naive coadaptive cortical control. *Journal of Neural Engineering*, 2(2):52–63, June 2005.

[7] G. J. Gage, K. J. Otto, K. a. Ludwig, and D. R. Kipke. Co-adaptive Kalman filtering in a naïve rat cortical control task. *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, 6:4367–70, Jan. 2004.

[8] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, May 2012.

[9] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[10] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Lebedev, C. S. Henriquez, and M. A. L. Nicolelis. Unscented Kalman filter for brain-machine interfaces. *PloS One*, 4(7):e6243, July 2009.

[11] C. T. Moritz and E. E. Fetz. Volitional control of single cortical neurons in a brain-machine interface. *Journal of Neural Engineering*, 8(2):025017, Apr. 2011.

[12] J. O'Doherty, M. Lebedev, T. Hanson, N. Fitzsimmons, and M. Nicolelis. A brain-machine interface instructed by direct intracortical microstimulation. *Frontiers in Integrative Neuroscience*, 3(September):1–10, 2009.

[13] L. Paninski, M. R. Fellows, N. G. Hatsopoulos, and J. P. Donoghue. Spatiotemporal tuning of motor cortical neurons for hand position and velocity. *Journal of Neurophysiology*, 91(1):515–32, 2004.

[14] H. Shimazaki and S. Shinomoto. Kernel bandwidth optimization in spike rate estimation. *Journal of Computational Neuroscience*, 29 (1-2)(2010):171–182, Aug. 2009.

[15] E. Stark and M. Abeles. Predicting movement from multiunit activity. *The Journal of Neuroscience*, 27(31):8387–94, Aug. 2007.

[16] 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):1022–35, 2004.