A Quantitative Comparison of Linear and Non-linear Models of Motor Cortical Activity for the Encoding and Decoding of Arm Motions

Yun Gao[*] **Michael J. Black**[†] **Elie Bienenstock**[*§] **Wei Wu**[*] **John P. Donoghue**[§]

[*]Division of Applied Mathematics, Brown University, Providence, RI 02912
[†]Dept. of Computer Science, Brown University, Box 1910, Providence, RI 02912
[§]Dept. of Neuroscience, Brown University, Providence, RI 02912

gao@cfm.brown.edu, *black@cs.brown.edu*, *elie@dam.brown.edu*,
weiwu@cfm.brown.edu, *john_donoghue@brown.edu*

Abstract

Many models have been proposed for the motor cortical encoding of arm motion. In particular, recent work has shown that simple linear models can be used to approximate the firing rates of a population of cells in primary motor cortex as a function of the position, velocity, and acceleration of the hand. Here we perform a systematic study of these linear models and of various non-linear generalizations. Specifically we consider linear Gaussian models, Generalized Linear Models (GLM), and Generalized Additive Models (GAM) of neural encoding. We evaluate their ability to represent the relationship between hand motion and neural activity, by looking at the likelihood of observed patterns of neural firing in a test data set and by evaluating the decoding performance of the different models (i.e. in terms of the error in reconstructing hand position from firing rates). To provide a level playing field for evaluating the decoding performance, we test all the models using a general recursive Bayesian estimator known as the particle filter*, thus isolating the effect of the encoding model on reconstruction accuracy.*

1 Introduction

We seek a model of the neural encoding of arm (or hand) movement in motor cortex. In addition to providing insight into neural coding, such a model contributes to our goal of decoding neural activity for the control of neural prosthetic devices including computer displays and robotic systems [4]. In this paper we adopt a Bayesian encoding/decoding framework within which we systematically compare a variety of encoding models. We find that non-linear, non-Gaussian, models relating hand kinematics with neural firing show promise for neural decoding applications.

A full review of encoding and decoding methods for motor cortical activity is beyond the scope of this brief report and we focus on recursive Bayesian decoding methods. In particular, the Kalman filter [6] is an effective technique for reconstructing continuous hand motion and gives more accurate results than previous methods based on population vector coding [5] or linear filtering [4]

The Kalman filter, while accurate and computationally efficient, assumes a linear relationship between hand motion and neural firing rates and, moreover, assumes Gaussian noise in the observed firing activity. The goal of this paper is to systematically and quantitatively explore and generalize these assumptions. We present a variety of models that relax the assumptions of linearity and Gaussian noise and we evaluate their encoding and decoding performance. Not surprisingly, we find that non-linear models yield better encoding and decoding than comparable linear models and that the assumption of Poisson spike counts gives better results than the Gaussian assumption. We exploit a general particle-filter decoding algorithm which allows us to compare all the models in the same computational framework, the only change being the linearity/non-linearity of the encoding and the probability distribution of spike counts.

2 Methods

Briefly, action potentials of 42 cells in the arm area of primary motor cortex in a macaque monkey were recorded using a chronically implanted Bioinc Technologies 100 microelectrode array. Subjects were trained to move a cursor viewed on a computer monitor using a two-jointed, low-friction, manipulandum moved on a 2D tablet parallel to the floor. For the experiments reported here, the subjects received a reward (e.g. juice) when they moved the cursor to "hit" a target circle displayed on the screen. After the target was hit it disappeared and then reappeared in a random new location. The subject performed this task continually, moving the manipulandum in an unconstrained fashion.

The position of the hand was recorded along with the neural activity using a Plexon acquisition system. After spike detection and sorting, the spike counts for each cell were computed in non-overlapping $70ms$ time bins. Hand velocity and acceleration were computed from the hand-position data using finite-difference approximation.

Separate training and testing data sets were recorded during the same experimental session with durations of approximately 3.5 minutes and 1 minute respectively. Each set consisted of both the hand kinematics (position, velocity, acceleration) and the associated spike counts.

3 Encoding Models

We adopt a Bayesian approach to encoding and formulate a *generative model* of neural firing,

$$\mathbf{z}_k = f_1(\mathbf{x}_k) + \mathbf{q}_k, \tag{1}$$

where $\mathbf{x}_k = [x, y, v_x, v_y, a_x, a_y]_k^T$ is a vector representing the position, velocity, and acceleration of the hand at time (i.e. bin) k, and the observations $\mathbf{z}_k \in \mathbb{R}^C$ represent a $C \times 1$ vector containing the spike counts at time k for C observed neurons within $70ms$. In our experiments, $C = 42$ cells.

We model the kinematics of the hand as

$$\mathbf{x}_k = f_2(\mathbf{x}_{k-1}) + \mathbf{w}_k = A\mathbf{x}_{k-1} + \mathbf{w}_k \tag{2}$$

where $f_2(\cdot)$ describes how the system state evolves over time and is taken to be linear. The matrix $A \in \mathbb{R}^{6\times6}$ is the coefficient matrix and the noise term $\mathbf{w}_k \sim N(0, \mathbf{W}), \mathbf{W} \in \mathbb{R}^{6\times6}$ is taken to be Gaussian with zero mean and covariance matrix $\mathbf{W}$ (the kinematic data is made zero-mean by subtracting the mean of the training kinematics).

In this paper we take equation (2) as fixed. We consider different versions of (1), which relate kinematics and spike counts. We consider both linear and non-linear forms of $f_1(\cdot)$. We also consider both Gaussian and Poisson models of spike counts.

3.1 Linear Gaussian Model (LGM)

First, consider a linear, Gaussian, formulation of (1) where

$$\boldsymbol{\mu}_k = H\mathbf{x}_k \tag{3}$$
$$\mathbf{z}_k \sim N(\boldsymbol{\mu}_k, \mathbf{Q}) \tag{4}$$

where $H \in \mathbb{R}^{C\times6}$ is a matrix that linearly relates the hand state to the neural firing. Here the observed spike counts, $\mathbf{z}_k$, are assumed to be normally distributed with mean $\boldsymbol{\mu}_k = H\mathbf{x}_k$ and covariance $\mathbf{Q} \in \mathbb{R}^{C\times C}$.

One strong reason for using the LGM is that the decoding problem (reconstructing the sequence of $\mathbf{x}_k$'s from the spike counts) can be implemented as a Kalman filter. For details of Kalman-filter decoding of motor cortical activity see [6].

For comparison, we will also consider below a simplified version of the linear, Gaussian, model, in which the noise covariance matrix is taken to be diagonal.

3.2 Generalized Linear Model (GLM)

While powerful, the LGM makes strong assumptions about neural firing. In particular, it assumes that the spike counts are normally distributed about $H\mathbf{x}_k$; that is, $\mathbf{z}_k \sim N(H\mathbf{x}_k, \mathbf{Q})$. We observe that a Poisson noise model is more consistent with the measured firing rates; the Poisson conditional density function is defined as:

$$f(\mathbf{z}_k|\mathbf{x}_k) = \prod_{c=1}^{C} \frac{\mu_{k,c}^{z_{k,c}}}{z_{k,c}!} e^{-\mu_{k,c}}, \tag{5}$$

where $\mu_{k,c} = E[z_{k,c}|\mathbf{x}_k]$, and $z_{k,c}$ is the spike count for time k and cell $c = 1, \ldots, 42$. The product over the cells, c, in (5) implies an assumption of conditional independence.

Previous Gaussian decoding methods have addressed the problem of Poisson data by preprocessing the firing-rate data in various ways to make it appear more Gaussian [6]. Instead, we formulate an explicit linear generative model for Poisson observations [3]. Rather than model the mean by $\boldsymbol{\mu}_k = H\mathbf{x}_k$ as above, we introduce a one-to-one *link* function g, we define $\boldsymbol{\eta}_k = g(\boldsymbol{\mu}_k)$, and we take $\boldsymbol{\eta}_k$ to be linearly related to the firing rates

$$\boldsymbol{\eta}_k = H\mathbf{x}_k.$$

The conditional mean of the spike counts is then given by

$$\boldsymbol{\mu}_k = g^{-1}(H\mathbf{x}_k). \tag{6}$$

For Gaussian data the link function, $g(\cdot)$, is just the identity while for Poisson data the appropriate link function is the natural logarithm [3].

3.3 Generalized Additive Model (GAM)

Both the LGM and GLM assume that the conditional mean (or its transformation through $g(\cdot)$) is a linear function of the hand kinematics, as in (3) and (6). A Generalized Additive Model (GAM) relaxes this constraint by expressing the observations $\mathbf{z}_k$ as sum of non-parametric, nonlinear, functions of the kinematic parameters [2].

Combining the GLM and GAM models, we let $\boldsymbol{\eta}_k$ be a linear combination of non-linear functions,

$$\boldsymbol{\eta}_k = g(\boldsymbol{\mu}_k) = \sum_i s_i(x_{k,i}), \tag{7}$$

where $x_{k,i}$ is the i-th component of $\mathbf{x}_k$, and the $s_i(\cdot)$'s are smoothing functions such as splines or local regression functions. Once again, we use the log link function in conjunction with a Poisson spike count.

3.4 Comparison

The various models are fit to the training data by maximum likelihood estimation. The GLM is fit using built-in Matlab function (Mathworks Inc., MA). The GAM is fit using the

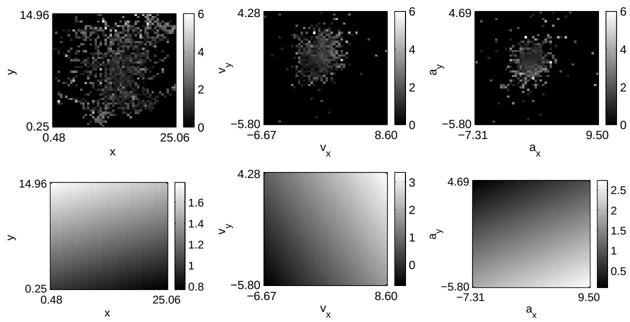

Figure 1: Top row shows the observed mean firing rate for one cell conditioned on position, velocity and acceleration (units are are cm, cm^2, and cm^3 respectively). Second row shows the best linear fit to the firing rates.

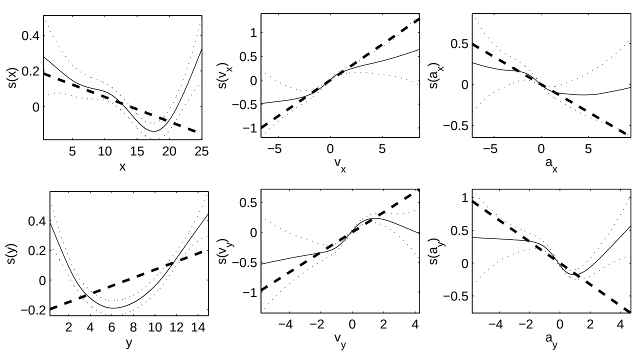

Figure 2: Generalized Additive Model versus the Linear Model for each kinematic variable. The solid line in each plot is the nonlinear smooth function $s_i(\cdot)$. The dotted line represents $\pm 2\times$ the standard error. The dashed line is the linear fit to the observed firing rates.

`gam` function in Splus (MathSoft Inc., WA) and the $s_i(\cdot)$ are taken to be splines of degree 4. Figures 1 and 2 show fitted models for a typical cell. The linear model is seen to be a reasonable first-order approximation, yet it misses all the nonlinear structure.

To compare the models we look at how well they explain the test data. Given the known hand kinematics, $\mathbf{x}_k$, and a particular generative model of the firing rate, we assume that the observations, $\mathbf{z}_k$, are realizations from an inhomogeneous Poisson process with estimated rate $\hat{\boldsymbol{\mu}}_k = f_1(\mathbf{x}_k)$

$$f(\mathbf{z}|\mathbf{x}) = \prod_{k=1}^{T} \prod_{c=1}^{C} \frac{\hat{\mu}_{k,c}^{z_{k,c}}}{z_{k,c}!} e^{-\hat{\mu}_{k,c}}, \tag{8}$$

where the product is over all time instants $k = 1, \ldots, T$ and all cells.

We compare the models against a homogeneous Poisson process with fixed mean $\boldsymbol{\mu}^*$. The log-likelihood ratios between the various inhomogeneous models and the homogeneous Poisson model are show in Table 1. Not surprisingly, all the inhomogeneous models do a good job of modeling

Method	Log Likelihood Ratio	Comparison
Linear Gaussian	2.252 e+3	1.00
GLM	2.650 e+3	1.18
GAM	2.994 e+3	1.33

Table 1: Log-likelihood ratio of various models computed on test data compared with a homogeneous Poisson model. The Comparison column shows the improvement in the log-likelihood ratio over the linear Gaussian model (LGM); that is, LGM/LGM, GLM/LGM, and GAM/LGM.

the firing data (the larger the log-likelihood ratio, the better the model explains the data). When we compare the different log-likelihood ratios we find that the GLM gives an 18% improvement over the linear Gaussian model while the GAM gives a 33% improvement.

4 Decoding

The experiments above suggest that both a Poisson spike-count model and a non-linear encoding model result in a more faithful encoding of neural data in (1). Our ultimate goal however is the *decoding* of neural data, so we consider the effects of the above models on the inference of hand kinematics. While the linear Gaussian model admits a closed-form, recursive, decoding algorithm, namely the Kalman filter, this technique cannot be applied directly to GLM or GAM encoding. To fairly compare all the methods in the same framework we use a recursive Bayesian estimator, known as the particle filter [1]. This approach can cope with both linear and non-linear observation equations and both Gaussian and Poisson models.

The reader is referred to [1] for details of the method, which is only summarized here. We assume: 1) the $\mathbf{x}_k$'s form a Markov Chain; 2) the $\mathbf{z}_k$'s are conditionally independent given all $\mathbf{x}_k$'s. Let $\mathbf{Z}_k$ be the entire history of spike counts up to time k. Then, we seek the maximum *a posteriori* (MAP) estimate, $p(\mathbf{x}_k|\mathbf{Z}_k)$, which can be written

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \kappa p(\mathbf{z}_k|x_k) p(\mathbf{x}_k|\mathbf{Z}_{k-1}), \tag{9}$$

where κ is a constant which makes the right hand side integrate to one. The prior term can be written as

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}) d\mathbf{x}_{k-1} \tag{10}$$

in which $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is determined by system equation (2) and $p(x_{k-1}|\mathbf{Z}_{k-1})$ is the posterior distribution at $k-1$.

The posterior distribution is represented by a set of N discrete random samples. Each sample represents a state of the hand (position, velocity, and acceleration) and an associated weight such that the weights of all the samples sum to one. Given sufficient samples, this approach can approximate both Gaussian and non-Gaussian probability densities.

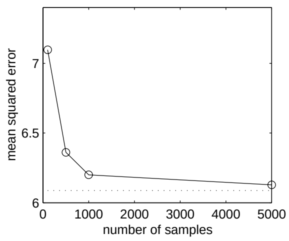

Figure 3: Convergence of the particle filter (solid line) to the Kalman filter estimate (dashed line) as the number of samples increases.

Method	MSE	ρ_x	ρ_y
(a) PF with LGM	6.1285	0.8149	0.9276
(b) PF (LGM, diag. cov.)	7.1689	0.7995	0.9224
(c) PF with GLM	6.3598	0.7925	0.8891
(d) PF with GAM	6.0447	0.8435	0.8989

Table 2: Comparison of decoding results using the particle filter (PF) with various encoding models. MSE = mean squared error in cm. ρ_x and ρ_y are the correlation coefficients for the reconstructed hand position in the x and y coordinates respectively. Number of particles is 5000.

The prior term, $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$, is evaluated using Monte Carlo sampling. We draw N samples from the posterior $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ according to the weights of the samples. For each sample we apply the system equation (2) by making a linear prediction and adding Gaussian noise. This gives a new set of N samples for with we evaluate the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ which is given by the observation model (1). Normalizing the likelihoods to sum to one gives the weights for the new particle set at time k.

Decoding using each of our models only involves changing the likelihood computation. The accuracy of the resulting estimate increases with the number of samples. In the case of the linear Gaussian model, as the number of particles increases, the result approaches that of the optimal Kalman filter as shown in Figure 3.

The reconstruction results are shown in Table 2. The LGM with full covariance matrix (a) does a good job of reconstruction. We compare this LGM with full covariance matrix to one with diagonal covariance (b) and note that the reconstruction accuracy drops significantly, indicating that the correlation between cells contains useful information.

Like the diagonal Gaussian model (b), both GLM (c) and GAM (d) assume independent firing of the cells, yet achieve more accurate decoding. The improvement in the GLM case (c) is due solely to the Poisson noise assumption, indicating that it is more appropriate than Gaussian.

The superiority of the GAM over the diagonal Gaussian and generalized linear models suggests the importance of non-linearity in our neural-coding problem. Whether this non-linearity is intrinsic to the neural code or results from our experimental paradigm is unknown. The GAM achieves the lowest mean-squared error of all the methods tested, though it is quite similar in accuracy to the full LGM.

These results suggest that modeling correlations in the Poisson case and combining this with a non-linear model should result in superior decoding results. This is the subject of current research.

5 Conclusions

We have presented and compared a variety of generative models of motor cortical activity. Both linear and non-linear mappings between hand kinematics and firing rates of a population of cells were considered, as were Gaussian and Poisson spike-count models. The non-linear, non-Gaussian, models provide a significantly better encoding of neural activity. For decoding, however, the linear Gaussian model (with full covariance) performs about as well as the non-linear generalized additive model with Poisson counts. The decoding results point to the importance of modeling the covariation of the neural firing rates of multiple cells. This suggests that extending the GAM approach to model such covariation may result in a significant improvement over the techniques explored here.

Acknowledgments. This work was supported in part by: the DARPA Brain Machine Interface Program, NINDS Neural Prosthetics Program grant NS25074, and the National Science Foundation (ITR Program award 0113679). We thank Mijail Serruya, Ammar Shaikhouni, Juliana Dushanova, Carlos Vargas, Lauren Lennox, and Matthew Fellows for their assistance.

References

[1] Y. Gao, M. Black, E. Bienenstock, S. Shoham, J. Donoghue. Probabilistic inference of hand motion from neural activity in motor cortex. *Adv. in Neural Info. Proc. Sys.*, 14, MIT Press, 2002.

[2] T. Hastie and R. Tibshirani. Generalized Additive Models. Chapman and Hall, London. 1990.

[3] P. McCullagh and J.A. Nelder. Generalized Linear Models Chapman and Hall, London. 1989.

[4] M. Serruya, N. Hatsopoulos, L. Paninski, M. Fellows, and J. Donoghue. Brain-machine interface: Instant neural control of a movement signal. *Nature*, (416):141–142, 2002.

[5] D. Taylor, S. Tillery, A. Schwartz. Direct cortical control of 3D neuroprosthetic devices. *Science*, Jun. 7;296(5574):1829-32, 2002.

[6] W. Wu, M. Black, Y. Gao, E. Bienenstock, M. Serruya, A. Shaikhouni, J. Donoghue. Neural decoding of cursor motion using a Kalman filter, *Adv. in Neural Info. Proc. Sys.*, 15, MIT Press, 2003.