Physically-based model for decoding motor-cortical activity

Gregory Shakhnarovich   Sung-Phil Kim   Michael J. Black
Department of Computer Science, Brown University, Providence, RI 02912
{gregory,spkim,black}@cs.brown.edu

A standard paradigm in decoding motor-cortical population activity, in particular in the context of neuromotor prostheses (NMP), is to infer from the recorded neural signal the kinematics of the movement: position, velocity, and/or acceleration. The movement has been traditionally associated with a moving computer cursor, with no mass and no medium. This is in contrast to natural tasks of the motor system, and to the future prosthesis devices, in which movement is subject to constraints imposed by the laws of physics.

Here we propose a model that makes a first step toward addressing the neural control of novel artificial motor systems. This provides an important proof-of-concept for human NMPs. Our approach is to decode the *dynamics* of hand movement directly from the neural activity. We do not attempt to accurately model the musculoskeletal structure of the arm. Instead, we propose a computationally effective framework to represent the dynamics of the limb moving in two dimensional plane.

Our approach is inspired by the generative model for hand-written digits in [1], sketched out in Figure 1. The endpoint of the limb (wrist) is assumed to be connected to one end of four imaginary springs, the other end of which is sliding with no friction along rails forming the boundaries of the $2L \times 2L$ "work area". Thus, according to Newton's second law, the acceleration of the hand at time t along x axis is given by

$$ma_x(t) \;=\; k_A(t)(L - x(t)) \;-\; k_B(t)(L + x(t)) - \beta v_x(t), \tag{1}$$

where $v_x(t)$ is the instantaneous velocity of the wrist at time t along the x axis, m is the point mass assumed at the wrist location, and β is the viscosity coefficient that represents the medium resistance and the elasticity of the actuator. We also introduce a constant stiffness constraint $k_A(t) + k_B(t) = \kappa$ in order to ensure non-negative coefficients. The y component is treated similarly.

Control of movement along x axis is achieved by modulating the spring coefficients k_A and k_B. When both movement and neural signal Z are observed, the functional relationship between the ks and Z is learned by the decoding algorithm. Note that the choice of the decoder is decoupled from the use of the spring model. In our experiments, we have used both linear decoders such as the linear filters and non-linear ones such as Support Vector Machines [3]. Finally, when only Z is observed, the coefficients, and thus the accelerations, can be inferred by the trained decoder. Hand position and velocity are then calculated directly by integrating the acceleration.

We evaluated the proposed model on the off-line movement reconstruction task using data sets previously collected from two behaving monkeys [2]. The animals were trained to control the cursor by moving the endpoint of a two-link manipulandum constrained to plane. The results demonstrate that the proposed model achieves state-of-the-art accuracy in predicting cursor position (in terms of MAE and correlation coefficients). Furthermore, compared to traditional approaches that directly infer hand kinematics from Z, decoding with the spring model produces trajectories that are closer to the natural ones in their power spectrum (in particular, more smooth) as illustrated by the typical power spectra in Figure 2 and trajectories in Figure 3.

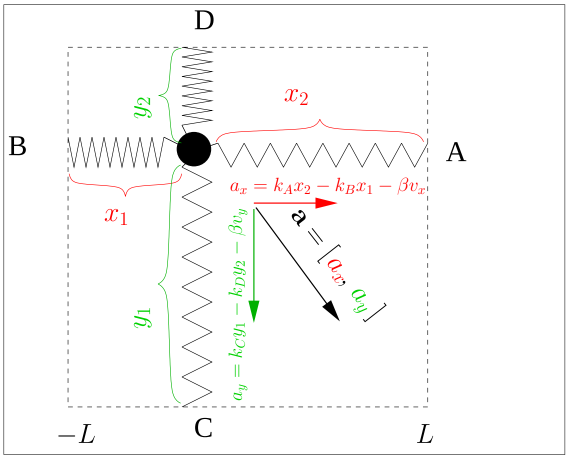

Figure 1: Sketch of the spring-based model. The outer endpoints of the springs are assumed to slide without friction, so that A and B are always orthogonal to C and D. The rest length is assumed to be zero for all springs. Movement is controlled by varying the stiffness coefficients k_A, k_B, k_C and k_D.

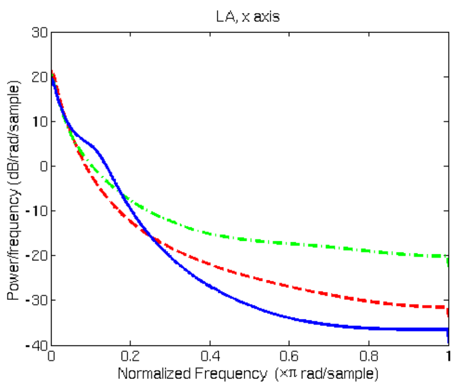

Figure 2: Power spectrum density for true kinematics (dashed red) and reconstruction: SVM directly on kinematics (dash-dot green), SVM with spring-based model (solid blue). Left: monkey LA, continuous tracking, x-axis. Right: monkey CL, sequential reaching, y-axis. Estimated using Burg's method.

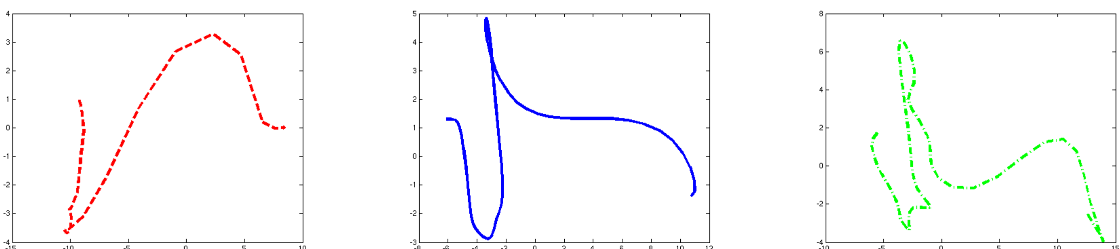

Figure 3: Typical trajectory and reconstructions, from left to right: ground truth, SVM decoding of spring model coefficients, direct SVM decoding of kinematics (both shown after post-hoc Butterworth smoothing). Note the more ragged form of the SVM-kinematics trajectory.

References

[1] G. E. Hinton and V. Nair. Inferring motor programs from images of handwritten digits. In *Advances in Neural Information Processing*, 2005.

[2] L. Paninski, M. Fellows, N. Hatsopoulos, and J. P. Donoghue. Spatiotemporal tuning of motor cortical neurons for hand position and velocity. *J. of Neurophysiology*, 91:515–532, 2004.

[3] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.