Recasting brain-machine interface design from a physical control system perspective

Yin Zhang[1] · **Steven M. Chase**[2,3]

Received: 15 January 2015 / Revised: 17 June 2015 / Accepted: 18 June 2015 / Published online: 5 July 2015

Abstract With the goal of improving the quality of life for people suffering from various motor control disorders, brain-machine interfaces provide direct neural control of prosthetic devices by translating neural signals into control signals. These systems act by reading motor intent signals directly from the brain and using them to control, for example, the movement of a cursor on a computer screen. Over the past two decades, much attention has been devoted to the decoding problem: how should recorded neural activity be translated into the movement of the cursor? Most approaches have focused on this problem from an estimation standpoint, i.e., decoders are designed to return the best estimate of motor intent possible, under various sets of assumptions about how the recorded neural signals represent motor intent. Here we recast the decoder design problem from a physical control system perspective, and investigate how various classes of decoders lead to different types of physical systems for the subject to control. This framework leads to new interpretations of why certain types of decoders have been shown to perform better than others. These results have implications for understanding how motor neurons are recruited to perform various tasks, and may lend insight into the brain's ability to conceptualize artificial systems.

Keywords Brain-machine interfaces · Decoding algorithm · Physical control system

1 Introduction

Brain-machine interfaces (BMIs) are a powerful class of assistive devices that may one day restore movement ability to paralyzed individuals (Schwartz et al. 2006). These devices act by creating a direct mapping between recorded neural activity and the movement of an external actuator, like a computer cursor or a robotic arm (Fig. 1) (Chapin 2004; Serruya et al. 2002; Taylor et al. 2002; Carmena et al. 2003; Musallam et al. 2004). Early clinical trials with intracortical BMIs, which use as their control signal the recorded activity of populations of single neurons, have recently shown that paralyzed individuals can effectively control computer cursors (Hochberg et al. 2006) and robotic arms of varying complexity (Hochberg et al. 2012; Collinger et al. 2013; Aflalo et al. 2015). However, much work yet needs to be done to give subjects control over artificial limbs that might rival control of the natural limb (Gilja et al. 2012). The design of BMI control algorithms that might enable stable, robust, and naturalistic control is an active area of research (Gilja et al. 2011).

A BMI decoding algorithm specifies how recorded signals (like recordings from intracortical multielectrode arrays) get translated into movement of the prosthesis.

Action Editor: Simon R Schultz

✉ Steven M. Chase
schase@cmu.edu

Yin Zhang
yinzhang@cs.cmu.edu

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

[2] Center for the Neural Basis of Cognition, Carnegie Mellon University, Pittsburgh, PA, USA

[3] Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

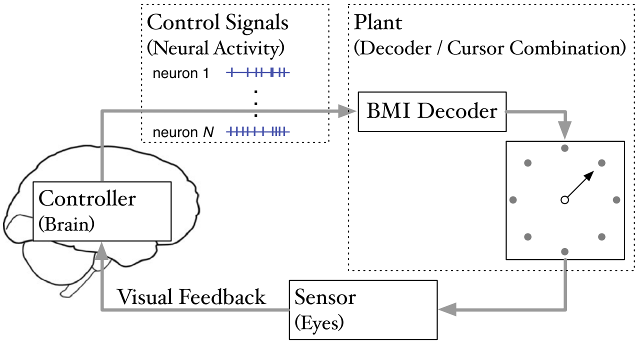

Fig. 1 Schematic of a BMI as a feedback control system. The major components of a feedback control system (namely, the controller, control signals, plant, and feedback) are laid out on top of a typical BMI cursor control schematic, where the brain is identified as the controller, the control signals are neural activity (often tapped out of primary motor cortex), the plant is the combination of the BMI decoder and the cursor, and feedback is accomplished by watching the cursor movements

Currently, nearly all BMI decoding algorithms are designed from an *estimation standpoint*: it is assumed that neurons are tuned to different intended movements, and recorded firing rates are treated as noisy observations of that underlying motor intent. Thus, differences in BMI decoding algorithms are typically interpreted to result from the different assumptions that they make about how neurons represent motor intent. Algorithms that fall into this class include linear estimators such as the population vector algorithm (PVA) (Georgopoulos et al. 1986; Georgopoulos et al. 1988; Velliste et al. 2008) and optimal linear estimator (OLE) (Kass et al. 2005; Salinas and Abbott 1994; Collinger et al. 2013), state-space models such as the Kalman filter (Wu et al. 2006; Hochberg et al. 2012), Laplace-Gaussian filter (Koyama et al. 2010a; Koyama et al. 2010a), and the unscented Kalman filter (Li et al. 2009), as well as a host of related variants (eg, particle filter (Brockwell et al. 2004), SSKF (Malik et al. 2011), VBR (Li et al. 2011), ReFIT-KF (Gilja et al. 2012), PT (Zhang Y and Chase SM 2013), SDKF (Golub et al. 2014), and others). Differences in the performance of various algorithms are typically assumed to relate to how well the algorithms' assumptions about motor intent encoding match the true underlying encoding of the neurons.

However, another way to interpret BMI system design is from a *control system* perspective. Each prosthetic effector, in conjunction with its decoding algorithm, acts as a control system that the subject needs to learn how to use through trial-and-error. Robotic arms are, by definition, physical control systems, while brain-controlled computer cursors may or may not be programmed to follow physical laws. When interpreted this way, differences in the performance of two algorithms might instead stem from differences in the control systems themselves: some control systems may be more usable than others, due to their physical characteristics or ease of conceptualization. Here we review some common BMI cursor decoding algorithms and derive the physical systems with which they correspond. We then re-interpret findings from the literature on which decoding algorithms work best in light of the physical systems that they represent. Intriguingly, when interpreted in this way, the literature suggests that BMI systems that follow physical laws are more usable than those that do not. Further, in on-line control it appears that BMI systems that reduce to equivalent physical forms tend to be equally-well controlled. These results have implications not only for BMI design, but may also shed light on the brain's ability to conceptualize motor effectors of varying forms.

2 Linear physical systems with control signals

Before proposing our view of BMI design from a control system perspective, we first look at a general control system, as shown in Fig. 1. In this system, control signals generated by the controller (the brain) are used to drive the plant (the decoder and cursor). Feedback about the system is observed by the sensors (the eyes) and used by the controller to generate new control signals. If it is a physical system, the system dynamics, which dictate the plant's evolution as driven by the control signal, should obey physical laws. Formally, the dynamics for a control system can be represented as

$$d\boldsymbol{x}/dt = f(\boldsymbol{x}, \boldsymbol{u}, t), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the plant's state in d dimensional space, $\boldsymbol{u} \in \mathbb{R}^n$ is the control signal and t is the current time. To make the control system a legitimate physical system, f should obey physical laws, i.e., velocity should be the derivative of position, acceleration should be the derivative of velocity, etc.

For computational simplicity, a linear discretized approximation of Eq. (1) is typically used in practice. To do that, continuous time is discretized into a series of steps with the step size as a small time interval Δ ($\Delta = 1/60$s for example), and the system dynamics at the t-th time step is expressed as

$$\boldsymbol{x}_{t+1} = A_t \boldsymbol{x}_t + B_t \boldsymbol{u}_t. \tag{2}$$

Here, $A_t \in \mathbb{R}^{d\times d}$ is a matrix that dictates the internal physics of the control system, and $B_t \in \mathbb{R}^{d\times n}$ is a matrix that dictates how the control system responds to its control inputs.

If we are studying a physical system driving a prosthetic effector's movement, then $\boldsymbol{x}$ is often the kinematics of the prosthetic effector, which could include the position, $\boldsymbol{p}$, the velocity, $\boldsymbol{v}$, and potentially higher-order terms as well.

For example, the 1st order linear discretized approximation, where the state is the position $\boldsymbol{x}_t = \boldsymbol{p}_t$, has the form

$$\boldsymbol{p}_{t+1} = A_t \boldsymbol{p}_t + B_t \boldsymbol{u}_t. \tag{3}$$

To ensure that position is the integral of velocity, we need to have the following relationship:

$$\begin{pmatrix} \boldsymbol{p}_{t+1} \\ \boldsymbol{v}_{t+1} \end{pmatrix} = \begin{pmatrix} I & \Delta I \\ \alpha_t & \beta_t \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_t \\ \boldsymbol{v}_t \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ B_t \end{pmatrix} \boldsymbol{u}_t. \tag{4}$$

Note that the control signal affects the next velocity, only.

To gain some insight into the meaning of these control system parameters, consider a simple 2nd order physical system (Fig. 2). This system consists of a point mass, m_p, attached to a viscous damper and an elastic spring connected in parallel. The forces operating on the point mass come from external sources, F_{ex}, as well as internal from the spring and damper. The force from the spring is $F_s = -kp$ (where k is the spring constant and p is the position of the point mass, measured relative to the equillibrium position of the spring), and the force from the damper is $F_d = -\eta v$ (where η is the damping coefficient and v is the velocity of the point mass). Because these elements act in parallel, these two forces sum at the point mass, to create a total internal force F_{in}, where

$$F_{in} = F_d + F_s = -\eta v - kp. \tag{5}$$

Together with the external force the acceleration of the point mass may be derived from Newton's laws of motion:

$$a = F_{tot}/m_p = (F_{in} + F_{ex})/m_p = (-kp - \eta v + F_{ex})/m_p. \tag{6}$$

Discretized by a small time duration Δ, the updated velocity is given as

$$v_{t+1} = v_t + \Delta a_t = v_t + \Delta(-kp_t - \eta v_t + F_{ex,t})/m_p. \tag{7}$$

and the updated position is given as

$$p_{t+1} = p_t + \Delta v_t, \tag{8}$$

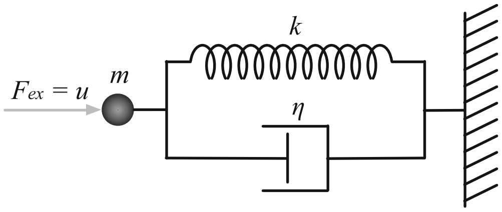

Fig. 2 A simple 2nd order physical control system. Here, a point mass m is hooked up to a parallel combination of a spring (with spring constant k) and a damper (with damping coefficient η). This configuration corresponds to the well-known Kelvin-Voigt model from material science

Putting this all together into matrix form, the discretized control system for the Kelvin-Voigt model in Fig. 2 is seen to be:

$$\begin{pmatrix} p_{t+1} \\ v_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & \Delta \\ -\Delta k/m_p & 1 - \Delta\eta/m_p \end{pmatrix} \begin{pmatrix} p_t \\ v_t \end{pmatrix} + \begin{pmatrix} 0 \\ \Delta/m_p \end{pmatrix} F_{ex,t}. \tag{9}$$

Comparing Eq. (9) with the general 2nd order physical control system presented in Eq. (4) reveals that the external force $F_{ex,t}$ plays the role of the control signal $\boldsymbol{u}_t$, the spring-like elastic effects account for α_t, and the viscous damping effects account for β_t.

3 BMI systems from a physical control systems perspective

In this section we re-interpret some fairly common decoding algorithms, including the PVA, OLE and Kalman filter, in terms of the physical control systems to which they correspond. Figure 1 casts the general BMI system into the control system framework. Here the motor cortex drives the BMI system by generating a proper set of neural activation patterns, which are the end result of a sequence of brain computations that take both visual feedback and goal information into account. Visual feedback of the prosthetic effector's movement is used to correct future movement. Therefore, BMI system design is essentially a control system design problem where one attempts to construct a BMI system that is as usable as possible. Formally, the dynamics of a BMI system can be represented as

$$d\boldsymbol{x}/dt = g(\boldsymbol{x}, \boldsymbol{y}, t). \tag{10}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the recorded firing rates from a population of n neurons. The only difference between Eqs. (10) and (1) is that the control signal $\boldsymbol{u}$ takes the form of neuronal firing rates $\boldsymbol{y}$.

When discussing the physical implementation of different BMI decoders, it will be necessary to distinguish the following three variables: motor intent, estimated motor intent, and implemented movement. Note that here and elsewhere in this document, we use the star notation to refer to intended movements (e.g., the intended velocity will be denoted as $\boldsymbol{v}^*$). Estimates of those intents will be indicated by an overhead carat (e.g., the estimated velocity will be denoted $\hat{\boldsymbol{v}}$). Finally, the movement that is implemented by the prosthesis will come without any notation (e.g., the implemented velocity will be denoted as $\boldsymbol{v}$). Please refer to Table 1 for all the notations.

Table 1 Notations

$\boldsymbol{y}$	neurons' observed firing rates
$\boldsymbol{x}^*(\boldsymbol{p}^*, \boldsymbol{v}^*)$	the intended kinematics (position, velocity)
$\hat{\boldsymbol{x}}\ (\hat{\boldsymbol{p}}, \hat{\boldsymbol{v}})$	the estimated kinematics (position, velocity)
$\boldsymbol{x}\ (\boldsymbol{p}, \boldsymbol{v})$	the implemented kinematics (position, velocity)

3.1 Linear estimators

We start with one of the earliest decoding algorithms, the population vector algorithm (PVA), first introduced by Georgopoulos and colleagues in 1986 (Georgopoulos et al. 1986). The central assumption of the PVA is that neurons are tuned to the direction of desired movement. Formally, with this assumption the firing rate of the neuron, y, can be written as

$$y = b_0 + b_x d_x^* + b_y d_y^* + \varepsilon \tag{11}$$

where b_0, b_x, and b_y are linear regression coefficients, $(d_x^*, d_y^*)^T$ is a unit vector that points in the direction of the intended movement, and the noise ε is assumed to be Gaussian. For simplicity, we have written this encoding model in two dimensions, though it has been found to generalize to three dimensions as well (Georgopoulos et al. 1986). This linear regression reduces to the well known "cosine-tuning" function (Georgopoulos et al. 1982).

If we allow m to be the magnitude of the regression coefficient vector $(b_x, b_y)^T$ (also known as the "modulation depth"), and we denote the "preferred direction" of the neuron as $\boldsymbol{d} = (b_x, b_y)^T/m$, we can write down the decoding equation for the PVA as:

$$\hat{\boldsymbol{v}} = \frac{k_s}{n}\sum_{i=1}^{n}\frac{y_i - b_{0,i}}{m_i}\boldsymbol{d}_i. \tag{12}$$

Here, i indexes each of the n recorded neurons and k_s is a constant that scales the unitless decoded direction into a velocity (Chase et al. 2012). The general interpretation of this equation is that each neuron "pushes" the cursor in its preferred direction, with the amount of the push being proportional to its firing rate. The sum of all of these pushing vectors is decoded as the population vector, which gets implemented in the BMI system. In this document, for simplicity, we assume the firing rates are alway centralized so we replace $(y_i - b_{0,i})$ with y_i in the following sections. We can rewrite Eq. (12) in matrix form by gathering the preferred directions into a single matrix D of size $2 \times n$, where each column corresponds to the preferred direction of a single neuron, and gathering the modulation depths m_i into a single diagonal matrix M:

$$\hat{\boldsymbol{v}} = (k_s/n)DM^{-1}\boldsymbol{y}. \tag{13}$$

To implement the PVA estimate of velocity into an actual device, the velocity of the prosthesis is set equal to its estimate, i.e., $\boldsymbol{v}_t = \hat{\boldsymbol{v}}_t$. The implemented position is then set equal to the integral of these velocity commands, $\boldsymbol{p}_{t+1} = \boldsymbol{p}_t + \Delta\boldsymbol{v}_t$. Therefore, to represent the decoding from the physical system perspective, we have

$$\text{PVA physical system: } \boldsymbol{p}_{t+1} = \boldsymbol{p}_t + \Delta(k_s/n)DM^{-1}y_t, \tag{14}$$

which is a special case of a 1st order linear physical control model (3), where $A_t = I$ and $B_t = \Delta(k_s/n)DM^{-1}$.

The PVA is a biologically-inspired algorithm. In practice, however, it has been shown to return biased estimates of motor intent when the preferred directions of the recorded neurons are not uniformly distributed (Salinas and Abbott 1994; Kass et al. 2005; Chase et al. 2009). To compensate for this bias, the optimal linear estimator (OLE) has been proposed. As the name implies, the OLE computes the optimal linear estimate according to square errors. The encoding model is written as

$$\boldsymbol{y} = B\boldsymbol{v}^* + \boldsymbol{\varepsilon}, \tag{15}$$

and the decoding model is

$$\hat{\boldsymbol{v}} = (B^TB)^{-1}B^T\boldsymbol{y}, \tag{16}$$

We can see that if the preferred directions of recorded neurons are uniformly distributed, then $B^TB \propto I$ and OLE is equivalent to PVA. Similar to PVA, the implemented velocity is also set equal to the estimated velocity in the decoding model of OLE and the position of the prosthesis is derived by integrating velocity. Thus, the physical system corresponding to OLE decoding is

$$\text{OLE physical system: } \boldsymbol{p}_{t+1} = \boldsymbol{p}_t + \Delta(B^TB)^{-1}B^T\boldsymbol{y}_t. \tag{17}$$

This is again another special case of a 1st order linear physical control system (Table 2 and 3).

Both the PVA and the OLE correspond to first-order physical control systems, albeit with slightly different mappings from neural firing to cursor movement. Experimental results from (Salinas and Abbott 1994; Kass et al. 2005; Chase et al. 2009) have shown that the OLE overcomes that bias. From an estimation standpoint, the OLE should be a better decoding algorithm than the PVA.

However, Chase and colleagues (Chase et al. 2009) also demonstrated that the PVA and the OLE perform equivalently on-line: subjects were just as adept at controlling the PVA as they were at controlling the OLE, despite the fact that neural activity was mapped to different cursor movements under the two algorithms. One possible interpretation of this is that subjects learn the mapping from neural activity to decoder, be it a biased decoder like the PVA or an unbiased decoder like the OLE. The difference between the mapping from neural activity to cursor movement produced

Table 2 BMI decoders under physical system perspective

1st order physical system	PVA	$\boldsymbol{p}_{t+1} = \boldsymbol{p}_t + \Delta(k_s/n)DM^{-1}\boldsymbol{y}_t$
	OLE	$\boldsymbol{p}_{t+1} = \boldsymbol{p}_t + \Delta(B^T B)^{-1}B^T\boldsymbol{y}_t$
2nd order physical system	VKF	$\begin{pmatrix}\boldsymbol{p}_t\\ \boldsymbol{v}_t\end{pmatrix} = \begin{pmatrix} I & \Delta I \\ \boldsymbol{0} & A - K_t B A\end{pmatrix}\begin{pmatrix}\boldsymbol{p}_{t-1}\\ \boldsymbol{v}_{t-1}\end{pmatrix} + \begin{pmatrix}\boldsymbol{0}\\ K_t\end{pmatrix}\boldsymbol{y}_t$
	PVKF *velocity implementation*	$\begin{pmatrix}\boldsymbol{p}_t\\ \boldsymbol{v}_t\end{pmatrix} = \begin{pmatrix} I & \Delta I \\ -K_{v,t}B_p & A_v - K_{v,t}H\end{pmatrix}\begin{pmatrix}\boldsymbol{p}_{t-1}\\ \boldsymbol{v}_{t-1}\end{pmatrix} + \begin{pmatrix}\boldsymbol{0}\\ K_{v,t}\end{pmatrix}\boldsymbol{y}_t$
Not a simple physical system	*PVKF position implementation*	$\begin{pmatrix}\boldsymbol{p}_t\\ \hat{\boldsymbol{v}}_t\end{pmatrix} = \begin{pmatrix} I - K_{p,t}B_p & \Delta I - K_{p,t}H \\ -K_{v,t}B_p & A_v - K_{v,t}H\end{pmatrix}\begin{pmatrix}\boldsymbol{p}_{t-1}\\ \hat{\boldsymbol{v}}_{t-1}\end{pmatrix} + K_t\boldsymbol{y}_t$

under these two decoders is akin to a visuomotor distortion, and visuomotor distortions are learned very quickly (Krakauer et al. 2000; Paz et al. 2005; Wu and Smith 2013). From this standpoint, the neurons are rapidly changing their activity to provide appropriate control signals to the device. Once learning is accomplished, both the OLE and the PVA give the subject a 1st order linear physical control system to control, and there appears to be no significant difference between the usability of these systems. This learning process is somewhat intuitive. Experiments demonstrate subjects can even learn the shuffled decoder with no predictive power after several days of practice (Ganguly and Carmena 2009; 2010).

3.2 Linear state-space decoders

A problem with the linear estimators described in the previous section is smoothing: when implemented on small time bins Δ, the movement estimates can be quite noisy. To compensate for this it is common to smooth the firing rate estimates (Koyama et al. 2010a). Naturally, any smoothing also affects the physical system.

Linear state space models handle smoothing in a more elegant manner by applying a smooth prior to the evolution of the intended kinematics. This prior takes the form of a linear dynamics equation:

$$\boldsymbol{x}_t^* = A\boldsymbol{x}_{t-1}^* + \boldsymbol{\omega}_t, \tag{18}$$

where $\boldsymbol{\omega}_t \sim \mathcal{N}(\boldsymbol{0}, R)$ is assumed to be zero mean Gaussian noise. The observations (recorded spike counts) are assumed to be linearly related to the kinematics:

$$\boldsymbol{y}_t = B\boldsymbol{x}_t^* + \boldsymbol{\varepsilon}_t, \tag{19}$$

where again the noise in the fit is assumed to be Gaussian, $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\boldsymbol{0}, Q)$.

In this implementation, the decoding problem is a Bayesian inference problem where the goal is to estimate the a posteriori probability of the intended kinematics when given the recorded spike counts. The solution in this case is the well-known Kalman filter (Kalman 1960), first introduced for BMI use by Wu et al. (Wu et al. 2003). The Kalman filter provides an efficient recursive algorithm to compute the posterior probability $p(\boldsymbol{x}_t^*|\boldsymbol{y}_{1,\ldots,t})$. The optimal estimate $\hat{\boldsymbol{x}}$ of the intended kinematics $\boldsymbol{x}^*$ is the mean of this distribution, which can be estimated through the following set of recursive equations:

$$K_t = (A\hat{\Sigma}_{t-1}A^T + R)B^T(B(A\hat{\Sigma}_{t-1}A^T + R)B^T + Q)^{-1}, \tag{20}$$

Table 3 BMI decoders comparison

Decoders compared	Scenario (decoder found to perform best in that scenario)	Reference
PVA vs. OLE	off-line trajectory reconstruction (OLE)	(Salinas and Abbott 1994) (Kass et al. 2005) (Chase et al. 2009)
	on-line closed loop control (equivalent)	(Chase et al. 2009) (Koyama et al. 2010a)
PVA/OLE vs. VKF	off-line trajectory reconstruction (VKF)	(Wu et al. 2006) (Koyama et al. 2010a)
	simulation (VKF)	(Koyama et al. 2010a)
PVKF-position vs. VKF	on-line closed loop control (VKF)	(Kim et al. 2008)
PVKF-velocity vs. VKF	on-line closed loop control (PVKF-velocity)	(Gilja et al. 2012)

$$\hat{\boldsymbol{x}}_t = (A - K_t BA)\hat{\boldsymbol{x}}_{t-1} + K_t \boldsymbol{y}_t, \quad (21)$$

$$\hat{\Sigma}_t = (I - K_t B)(A\hat{\Sigma}_{t-1}A^T + R). \quad (22)$$

Here K_t is the familiar Kalman gain which computes the optimal mixing between reliance on information from the prior and information from the observations according to the noise assumed to be in each. $\hat{\Sigma}_t$ is the estimate of the covariance of $p(\boldsymbol{x}_t^*|\boldsymbol{y}_{1,\ldots,t})$.

To use the Kalman filter, an initial covariance matrix $\hat{\Sigma}_0$ and kinematics $\hat{\boldsymbol{x}}_0$ must be given. In practice, it is common to center the prosthesis so all terms in $\hat{\boldsymbol{x}}_0 = \boldsymbol{0}$ with certainty $\hat{\Sigma}_0 = 0$. The Kalman gain is time dependent. However, it tends to converge to a stable value within a few timesteps, independent of the observations $\boldsymbol{y}$ (Chui and Chen 2009; Malik et al. 2011). Another common practice is to initialize the system with $\hat{\boldsymbol{x}}_0$ and $\hat{\Sigma}_0$ as the matrix that ensures that K_t is stable (Dethier et al. 2011; Sadtler et al. 2014).

The relationship between the kinematics $\boldsymbol{x}_t$ that are actually implemented and the estimated kinematics $\hat{\boldsymbol{x}}_t$ depends on exactly how the Kalman filter is implemented. Here we introduce two popular implementations, the velocity Kalman filter (VKF) (Kim et al. 2008; Hochberg et al. 2012) and the position-velocity Kalman filter (PVKF) (Wu et al. 2006; Homer et al. 2013; Gowda et al. 2014).

3.2.1 Velocity Kalman filter

The VKF assumes that neurons are tuned to the intended velocity. Thus, the state evolution equation is

$$\boldsymbol{v}_t^* = A\boldsymbol{v}_{t-1}^* + \boldsymbol{\omega}_t \quad (23)$$

and the observation equation is

$$\boldsymbol{y}_t = B\boldsymbol{v}_t^* + \boldsymbol{\varepsilon}_t. \quad (24)$$

From Eq. (21), we have the estimated intended velocity as

$$\hat{\boldsymbol{v}}_t = (A - K_t BA)\hat{\boldsymbol{v}}_{t-1} + K_t \boldsymbol{y}_t. \quad (25)$$

The implemented velocity $\boldsymbol{v}_t$ in this case is set equal to $\hat{\boldsymbol{v}}_t$ and the position is the integral of the velocity (Kim et al. 2008; Hochberg et al. 2012). In matrix form, the implemented movement can be written as

VKF physical system:

$$\begin{pmatrix} \boldsymbol{p}_t \\ \boldsymbol{v}_t \end{pmatrix} = \begin{pmatrix} I & \Delta I \\ 0 & A - K_t BA \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_{t-1} \\ \boldsymbol{v}_{t-1} \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ K_t \end{pmatrix} \boldsymbol{y}_t. \quad (26)$$

Comparing with Eq. (4), we can see that the VKF is a 2nd order linear physical control system where the system state includes position and velocity, $\boldsymbol{x}_t = (\boldsymbol{p}_t, \boldsymbol{v}_t)^T$, with an elastic term, α_t, that is equal to zero and a viscous term, $\beta_t = A - K_t BA$. It is interesting to note the parallel between force and velocity representations that emerge in this implementation of the Kalman filter. Even though the VKF makes the assumption that neurons are driven by intended velocities, they play the role in Eq. (26) of providing a force input to the system.

To our knowledge, nobody has directly compared the VKF to either the OLE or the PVA in on-line control. However, Koyama and colleagues demonstrated that a variant of the VKF called the Laplace-Gaussian filter (LGF) outperformed the PVA and OLE on-line (Koyama et al. 2010a). The LGF and the VKF are both state-space decoders, and differ in only two main respects: the LGF fits an observation model that assumes Poisson noise statistics and log-linear tuning to intended velocity, whereas the VKF assumes linear Gaussian tuning of neurons to intended velocity. Importantly, this implies that the physical control system representing the VKF and the LGF will be of the same form: second order with no elastic terms.

Koyama and colleagues performed simulations and offline trajectory reconstructions to determine that the key factor that allowed the LGF to outperform the PVA and OLE was its state-space formulation: they found no significant differences in the performance of the LGF relative to the VKF. We therefore take this as indirect evidence that the VKF would outperform the OLE and the PVA on-line. Why should this be the case? From an estimation standpoint, the interpretation would be that intended velocities really do evolve smoothly over time as implied by Eq. (23), and so incorporating the fact enables better estimates of the velocity intent. However another interpretation is that the VKF and OLE are fundamentally different control systems, and 2nd order physical control systems may simply be easier to control than 1st order physical control systems.

3.2.2 Position velocity Kalman filter

Another widely used Kalman filter model is the PVKF. In contrast to the VKF, the PVKF assumes that neurons are tuned to both the intended position and intended velocity. Thus, the state is $\boldsymbol{x}_t^* = (\boldsymbol{p}_t^*, \boldsymbol{v}_t^*)^T$ and to encourage the state evolution model to obey physical laws, it is typically set to be

$$\begin{pmatrix} \boldsymbol{p}_t^* \\ \boldsymbol{v}_t^* \end{pmatrix} = \begin{pmatrix} I & \Delta I \\ \boldsymbol{0} & A_v \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_{t-1}^* \\ \boldsymbol{v}_{t-1}^* \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{\omega}_{v,t} \end{pmatrix}. \quad (27)$$

The PVKF observation model is

$$\boldsymbol{y}_t = (B_p, B_v) \begin{pmatrix} \boldsymbol{p}_t^* \\ \boldsymbol{v}_t^* \end{pmatrix} + \boldsymbol{\varepsilon}_t. \quad (28)$$

From Eq. (20) we can compute K_t. Dissociating K_t into two parts corresponding to position and velocity as $K_t = (K_{p,t}, K_{v,t})^T$, we can write the estimated intended position and velocity as

$$\begin{pmatrix} \hat{\boldsymbol{p}}_t \\ \hat{\boldsymbol{v}}_t \end{pmatrix} = \begin{pmatrix} I - K_{p,t}B_p & \Delta I - K_{p,t}H \\ -K_{v,t}B_p & A_v - K_{v,t}H \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{p}}_{t-1} \\ \hat{\boldsymbol{v}}_{t-1} \end{pmatrix} + K_t \boldsymbol{y}_t, \quad (29)$$

where $H = \Delta B_p + B_v A_v$

In the PVKF, the estimated position is not, in general, equal to the integral of the estimated velocity. Even though the state evolution equation (27) biases estimates of position and velocity to obey this rule, it is not a hard constraint: a compromise between position and velocity will be estimated that best explains the observed spike rates. This then leaves one with a choice when trying to implement the PVKF, since both the position and velocity estimates cannot be simultaneously implemented. One common method is to use the estimated position as the implemented position, and to allow the implemented velocity to evolve as $\boldsymbol{v}_t = (\hat{\boldsymbol{p}}_{t+1} - \hat{\boldsymbol{p}}_t)/\Delta$ (Wu et al. 2006):

PVKF, position implementation:

(Not a simple physical control system)

$$\begin{pmatrix} \boldsymbol{p}_t \\ \hat{\boldsymbol{v}}_t \end{pmatrix} = \begin{pmatrix} I - K_{p,t}B_p & \Delta I - K_{p,t}H \\ -K_{v,t}B_p & A_v - K_{v,t}H \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_{t-1} \\ \hat{\boldsymbol{v}}_{t-1} \end{pmatrix} + K_t \boldsymbol{y}_t. \quad (30)$$

With this implementation, the estimated velocity, $\hat{\boldsymbol{v}}_t$, becomes a latent variable that keeps the system from having a simple physical control system correlate. Experimental results from (Kim et al. 2008) show that when the PVKF is implemented in this way, its performance is inferior to the performance of the velocity-only Kalman filter. This fact is hard to rationalize from the estimation viewpoint, since the encoding model assumed by the PVKF typically fits the firing rates better than the encoding model assumed by the VKF. From the control system viewpoint, however, a possible explanation of this result is that the PVKF with position implementation is not a simple physical control system, and is therefore difficult to use.

There are other ways to implement the PVKF. One way is to use the estimated velocity as the implemented velocity and treat the estimated position $\hat{\boldsymbol{p}}_t$ as the hidden variable. Another is to use a linear combination of estimated velocity and estimated position as the implemented velocity. This latter method was shown in (Homer et al. 2013) to work better than the position-implementation of the PVKF. However, none of these versions give the subject a simple physical system without hidden states to learn to control.

There is one implementation of the PVKF, however, that does correspond to a simple physical control system with no hidden states. To do this, the implemented velocity is made equal to the estimated velocity, and *the estimated position becomes the integral of the estimated velocity*:

PVKF, velocity implementation:

(Physical control system)

$$\begin{pmatrix} \boldsymbol{p}_t \\ \boldsymbol{v}_t \end{pmatrix} = \begin{pmatrix} I & \Delta I \\ -K_{v,t}B_p & A_v - K_{v,t}H \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_{t-1} \\ \boldsymbol{v}_{t-1} \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ K_{v,t} \end{pmatrix} \boldsymbol{y}_t. \quad (31)$$

As it turns out, this implementation of the PVKF has been used by Gilja and colleagues to achieve the best BMI control demonstrated to date (Gilja et al. 2012). Their implementation of this equation was one of two design innovations of their ReFIT-KF algorithm, the other being a new method for re-calibrating the device from on-line training data. Although they derived (31) in a different way, using a causal-intervention step that forces the variance of the position estimate to go to zero, the net effect is the same, and leads to a second order physical system with both elastic and damping terms.

The velocity-implementation of the PVKF has been shown to handily outperform the VKF (Gilja et al. 2012). One interpretation of why this is true is that the causal intervention step better captures the fact that subjects know the position of the cursor from sensory feedback, so there should be no uncertainty about it. While this clearly cannot be exactly true, due to sensory feedback delays and sensory noise (Golub et al. 2013), it may be true enough to allow for better control. Another interpretation might be that neurons are driven by a non-volitional, positional signal representing the real position of the cursor, and this equation allows that "nuisance variable" term to be removed. However, yet another interpretation might be that this implementation of the PVKF results in a simple, 2nd order physical control system that still allows for neurons to modulate as a function of position. It may further be the case that 2nd order physical control systems that include a certain elastic component (a non-zero α_t in Eq. (4)) are more usable than those systems, like the VKF, that do not have that component.

4 Discussion

The optimal way to implement a BMI decoding algorithm is an important question relevant to clinical deployment of neural prostheses. Here we recast this problem from the perspective of control system design, and derive the physical control systems corresponding to various types of decoders commonly used in BMI cursor control. This process enables new insights into BMI design, and suggests novel explanations about why some decoders have been shown to perform better than others. In particular, the literature suggests that: 1) 2nd order physical systems tend to be more usable than 1st order physical systems, 2) decoders that cannot be expressed as simple physical control systems do not appear to work as well as those that can be expressed this way, and 3) a 2nd order control system with elastic terms seems to work better than one without.

Recent work has highlighted the utility of approaching BMI design as a separate problem from inferring natural behavior (Tillery and Taylor 2004; Chase and Schwartz 2010). Marathe and Taylor (Marathe and Taylor 2011) studied the effect of mapping one control parameter (e.g., position, velocity, or goal) to the control of another. They found that the optimal mapping was not necessarily one-to-one, but rather changed as a function of different types of decoding noise. Gowda and colleagues (Gowda et al. 2014) have presented a thorough investigation of the dynamical systems properties of the PVKF. They found that certain implementations of the decoder could create workspace attractor points that might be detrimental to BMI control. These studies point out the gains that may be realized when BMI control is not constrained to reflect the neural encoding of natural arm dynamics, and emphasize the importance of the physical control system perspective when interpreting BMI performance.

4.1 Suggestions for new BMI systems

Our analyses suggest a number of new approaches to BMI decoding algorithm design that might prove fruitful. Of all the decoding algorithms we reviewed, none went beyond 2nd order. Given that 2nd order systems appear more usable than 1st order systems, it is interesting to speculate as to whether a 3rd or 4th order system would be even easier to control. These higher-order systems may actually be a closer match to the human arm: in (Liu and Todorov 2007), Liu and colleagues model the arm as a 3rd order linear physical system and are able to capture many of the emergent features of natural reaching movements. There have been instances in the literature that have included acceleration and higher order terms in their decoding algorithms. For example, Wu and colleagues compared the performance of a Kalman filter decoder with up to 6th order terms, and found that the 3rd order model consisting of position, velocity, and acceleration terms provided the best performance in their off-line trajectory reconstruction (Wu et al. 2006). However, they used the position-implementation of their decoder, which we have already demonstrated does not correspond to a simple physical system beyond 1st order. It would be interesting to test how physical implementations of higher order control systems might perform on-line.

Another fruitful approach might be the design of state-dependent control systems. The PVA and the OLE are both static systems, i.e., the physical control parameters do not vary with time. The Kalman filter technically has time varying parameters, but in practice the Kalman gain converges to a constant within a few timesteps, and some researchers even initialize it at the converged-values to keep it time-invariant (Malik et al. 2011; Sadtler et al. 2014). On the other hand, state-dependent control is also studied to deal with the situations where static system is no longer appropriate. One such situation is the instability of tuning curves between sessions (Rokni et al. 2007; Chestek et al. 2009). To compensate for such instability, different adaptive decoding algorithms are proposed where decoding parameters are updated iteratively over time (Li et al. 2011; Shpigelman et al. 2009; Zhang Y and Chase SM 2013; Suminski et al. 2013; Kowalski et al. 2013; Orsborn et al. 2014). Another kind of state-dependent control is used to assist the subject during training with the BMI system, where the assistance is adjusted according to the subject's performance (Velliste et al. 2008). Other state-dependent control algorithms include (Shanechi et al. 2013), where in order to simultaneously estimate movement trajectory or target intent, the decoding parameters are adjusted as targets are approached. Such state-dependent control was the goal behind the recently-developed speed dampening Kalman filter (Golub et al. 2014), which effectively manipulated the viscous damping forces as a function of trajectory curvature to allow for more stable transitions from moving to stopping. The physical control system viewpoint we espouse here may be a way of integrating these approaches into a simple, unified framework for robust prosthetic applications.

Finally, we should note that we have focused exclusively in this review on kinematic BMI decoders of the type that are commonly used to drive cursors on computer screens. Another class of decoders attempts to extract kinetic information (forces and torques) from recorded neural activity for direct control of force output (Fagg et al. 2009; Nazarpour et al. 2012; Nishimura et al. 2013; Chhatbar and Francis 2013; Oby et al. 2013). It is unclear at present how to best integrate these two approaches to BMI design. One intriguing idea is that the brain represents impedances (relationships between kinetics and kinematics) rather than desired forces or kinematics per se (Hogan 1985; Tin and Poon 2005; Hogan and Sternad 2012). An interesting, active direction of research is to design decoders that seemlessly transition between free movement and object interaction. It is possible that decoding impedance control signals directly from the brain would enable this transition.

4.2 Do details of the control signal mapping matter?

We have focused primarily on the overall form of the physical control system represented by different decoders: e.g., whether it is 1st order or 2nd order and whether it contains both viscous and elastic elements, etc. We have not focused on the particular values of those parameters as much, except to point out that the PVA and OLE, which have different mappings between firing rates and cursor movement, perform similarly on-line (Chase et al. 2009). Do the details of the control system parameters matter at all?

The answer is certainly yes. Sadtler and colleagues have recently demonstrated that some mappings between neural activity and cursor movement are easily learned, while others are very difficult to learn (Sadtler et al. 2014). In that study they discovered that subjects could easily learn to map an existing pattern of neural activity to an arbitrary movement; what was difficult was creating new correlation patterns in the neurons themselves. This suggests that, provided the mapping from neural activity to cursor movement preserves the natural correlations within the neural population, two physical systems of the same form may be equivalently usable.

Also, while Sadtler and colleagues noted substantial improvement in performance with short amounts of practice on novel ‘within-manifold’ (correlation preserving) mappings, the performance under those mappings was still substantially worse than performance with the initially-estimated natural tuning curves. Further, Gilja and colleagues (Gilja et al. 2012) have noted a significant improvement in control if tuning curves are re-estimated from on-line training data, which demonstrates different mappings between firing rates and cursor movement can lead to different on-line performance.

Under the estimation-framework, performance will be best with mappings from neural activity to cursor movement that best capture the natural neural tuning. Under the control framework, performance will be best with those mappings that most accurately capture the volitionally usable correlation structures within the neural population. Of course, all of this ignores long-term learning, which may allow subjects to become proficient even at non-intuitive mappings (Ganguly and Carmena 2009; 2010). Further work will need to be done to determine the precise details that determine which mappings from neural activity to movement perform better than others, and how learning may ultimately play a role in sculpting their performance (Orsborn et al. 2014).

4.3 Alternate views of motor cortical recruitment

Decades of motor control studies have established that neural activity in motor cortex correlates with various features of movement. In BMI design, it is common to interpret these correlations by thinking of neurons as being tuned to the desired outcome or intended movement of an effector. However, neural activity can be flexibly dissociated from the effector (Schieber 2011): using operant conditioning, individual neurons can be trained to correlate and decorrelate from particular muscles (Fetz 1969; Fetz and Finocchio 1971), even when spike triggered averaging of EMG traces provide evidence of monosynaptic connections between the neuron and the muscle (Davidson et al. 2007). Evidence of a flexible relationship between neural activity and behavior also comes from natural movement paradigms. Neural activity in motor cortex readily changes during motor learning (Jarosiewicz et al. 2008; Paz et al. 2003; Wise et al. 1998; Gandolfo et al. 2000; Ganguly and Carmena 2009; Sadtler et al. 2014). Neural activity has also been shown to change during associative learning: neurons in M1 will respond to the color of a target during an associative learning task, and will often maintain that tuning after color is no longer relevant (Zach et al. 2008). Even context can modulate neural tuning. Hepp-Reymond and colleagues (Hepp-Reymond et al. 1999) demonstrated that neurons in primary motor cortex are sensitive to the context of an isometric force task, and will change their firing to particular force levels according to how many force targets are presented in the task.

These studies suggest that motor cortical activity is extremely fungible, restructuring itself in the face of new task demands in a manner that is not at all well understood. It is hard to reconcile this view with a static view of tuning to intended movements, unless one assumes that these motor intent signals can themselves be dissociated from the motor outcome (Chase and Schwartz 2010). If this is the case, there may be only subtle differences among the viewpoints that neurons tune to flexible motor intent signals, that neurons act as control signals to drive an effective behavior, or that populations of neurons act as a flexible pattern generator on which movements can be built (Shenoy et al. 2013). In this review, we mainly wish to highlight the importance of these multiple viewpoints when attempting to interpret BMI performance and, ultimately, design the optimal decoding algorithm.

4.4 Relationships to embodiment, internal models, and natural motor control

Natural motor control is fraught with computational difficulties, not least of which is the necessity to compensate for noisy, delayed sensory feedback. To generate fast, dexterous movements, it is necessary to compensate for these sensory delays. It is widely believed that we do this with the aid of internal models that allow us to predict, in real time, the outcomes of our motor commands before sensory feedback becomes available (Crapse and Sommer 2008; Shadmehr et al. 2010). These internal (forward) models are thought to take as input efference copies of our motor commands, and use them to predict the sensory consequences, such as the new arm or eye position, that result from those commands (Sommer and Wurtz 2002; Wolpert et al. 1995). Essentially, these models embody our internal conception of the physics of our limbs and how they respond to our motor commands. These predicted locations can then be used as the basis for planning the next movement before real sensory feedback becomes available, allowing for faster motor sequence production.

Internal models may also be used in BMI control. Motor commands of subjects using a BMI to control a computer cursor in a center-out movement task are more appropriate to the real time position of the cursor than the last sensory feedback position, indicating that subjects compensate for sensory feedback delays while using a BMI (Golub et al. 2012). Further, differences between a subject's internal model of the decoder and the actual cursor dynamics may explain errors in on-line control (Golub et al. 2013).

It is intriguing to speculate on the utility of internal models in BMI control. Internal models are thought to be a key component in motor adaptation (Shadmehr et al. 2010; Kawato 1999), and thus subjects who build internal models of BMIs may be able to take advantage of all of the computational motor adaptation machinery that natural motor control relies upon. A reliable internal model of the BMI device may also be the key to embodiment, when the device feels like a natural extension of one's body (Schwartz et al. 2006; Giummarra et al. 2008). However, can subjects build internal models of any BMI device? While the answer to this question is well beyond the scope of this review, we posit that it might be easier to build internal models of physical systems than it is to build internal models of non-physical systems, simply because all of the internal models we build in the context of natural reaching are, by definition, models of physical systems. It could be that the reason decoding algorithms corresponding to physical systems appear to be more useable than those that do not is because they are more readily conceptualized with an internal model.

Acknowledgments This work was supported by the Pennsylvania Department of Health Research Formula Grant SAP #4100057653 under the Commonwealth Universal Research Enhancement program. We thank A. Batista, W. Bishop, M. Golub, S. Kennedy, P. Lund, E. Oby, S. Perel, K. Quick, P. Sadtler and B. Yu for insightful comments on a preliminary version of this paper.

Conflict of interests These authors declare no conflict of interests.

References

Aflalo, T., Kellis, S., Klaes, C., Lee, B., Shi, Y., Pejsa, K., Shanfield, K., Hayes-Jackson, S., Aisen, M., Heck, C., Liu, C., & Andersen, R.A. (2015). Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science*, *348*(6237), 906–910.

Brockwell, A.E., Rojas, A., & Kass, R.E. (2004). Recursive bayesian decoding of motor cortical signals by particle filtering. *Journal of Neurophysiology*, *91*(4), 1899–1907.

Carmena, J.M., Lebedev, M.A., Crist, R.E., O'Doherty, J.E., Santucci, D.M., Dimitrov, D.F., Patil, P.G., Henriquez, C.S., & Nicolelis, M.A.L. (2003). Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biology*, *1*(2), e42.

Chapin, J.K. (2004). Using multi-neuron population recordings for neural prosthetics. *Nature Neuroscience*, *7*(5), 452–455.

Chase, S.M., & Schwartz, A.B. (2010). Inference from populations: going beyond models. *Progress in Brain Research*, *192*, 103–112.

Chase, S.M., Schwartz, A.B., & Kass, R.E. (2009). Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural Networks*, *22*(9), 1203–1213.

Chase, S.M., Kass, R.E., & Schwartz, A.B. (2012). Behavioral and neural correlates of visuomotor adaptation observed through a brain-computer interface in primary motor cortex. *Journal of Neurophysiology*, *108*(2), 624–644.

Chestek, C.A., Cunningham, J.P., Gilja, V., Nuyujukian, P., Ryu, S.I., & Shenoy, K.V. (2009). Neural prosthetic systems: current problems and future directions. In *Annual international conference of the ieee engineering in medicine and biology society* (pp. 3369–3375).

Chhatbar, P.Y., & Francis, J.T. (2013). Towards a naturalistic brain-machine interface: Hybrid torque and position control allows generalization to novel dynamics. *PloS ONE*, *8*(1), e52,286.

Chui, C.K., & Chen, G. (2009). *Kalman filtering: with real-time applications*, 4th edn. Berlin: Springer.

Collinger, J.L., Wodlinger, B., Downey, J.E., Wang, W., Tyler-Kabara, E.C., Weber, D.J., McMorland, A.J., Velliste, M., Boninger, M.L., & Schwartz, A.B. (2013). High-performance neuroprosthetic control by an individual with tetraplegia. *Lancet*, *381*(9866), 557–564.

Crapse, T.B., & Sommer, M.A. (2008). Corollary discharge circuits in the primate brain. *Current Opinion in Neurobiology*, *18*(6), 552–557.

Davidson, A.G., Chan, V., O'Dell, R., & Schieber, M.H. (2007). Rapid changes in throughput from single motor cortex neurons to muscle activity. *Science*, *318*(5858), 1934–1937.

Dethier, J., Nuyujukian, P., Eliasmith, C., Stewart, T.C., Elasaad, S.A., Shenoy, K.V., & Boahen, K.A. (2011). A brain-machine interface operating with a real-time spiking neural network control algorithm. In *Advances in neural information processing systems* (pp. 2213–2221).

Fagg, A.H., Ojakangas, G.W., Miller, L.E., & Hatsopoulos, N.G. (2009). Kinetic trajectory decoding using motor cortical ensembles. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *17*(5), 487–496.

Fetz, E.E. (1969). Operant conditioning of cortical unit activity. *Science*, *163*(3870), 955–958.

Fetz, E.E., & Finocchio, D.V. (1971). Operant conditioning of specific patterns of neural and muscular activity. *Science*, *174*(4007), 431–435.

Gandolfo, F., Li, C.S., Benda, B.J., Schioppa, C.P., & Bizzi, E. (2000). Cortical correlates of learning in monkeys adapting to a new dynamical environment. *Proceedings of the National Academy of Sciences*, *97*(5), 2259–2263.

Ganguly, K., & Carmena, J.M. (2009). Emergence of a stable cortical map for neuroprosthetic control. *PLoS biology*, *7*(7), e1000,153.

Ganguly, K., & Carmena, J.M. (2010). Neural correlates of skill acquisition with a cortical brain machine interface. *Journal of Motor Behavior*, *42*(6), 355–360.

Georgopoulos, A.P., Kalaska, J.F., Caminiti, R., & Massey, J.T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, *2*(11), 1527–1537.

Georgopoulos, A.P., Schwartz, A.B., & Kettner, R.E. (1986). Neuronal population coding of movement direction. *Science*, *233*(4771), 1416–1419.

Georgopoulos, A.P., Kettner, R.E., & Schwartz, A.B. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *The Journal of Neuroscience*, *8*(8), 2928–2937.

Gilja, V., Chestek, C.A., Diester, I., Henderson, J.M., Deisseroth, K., & Shenoy, K.V. (2011). Challenges and opportunities for next-generation intracortically based neural prostheses. *IEEE Transactions on Biomedical Engineering*, *58*(7), 1891–1899.

Gilja, V., Nuyujukian, P., Chestek, C.A., Cunningham, J.P., Yu, B.M., Fan, J.M., Churchland, M.M., Kaufman, M.T., Kao, J.C., Ryu, S.I., & Shenoy, K.V. (2012). A high-performance neural prosthesis enabled by control algorithm design. *Nature Neuroscience*, *15*(12), 1752–1757.

Giummarra, M.J., Gibson, S.J., Georgiou-Karistianis, N., & Bradshaw, J.L. (2008). Mechanisms underlying embodiment, disembodiment and loss of embodiment. *Neuroscience and Biobehavioral Reviews*, *32*(1), 143–160.

Golub, M.D., Chase, S.M., & Yu, B.M. (2012). Internal models engaged by brain-computer interface control. In *Proceedings of the 35th annual international conference of the ieee engineering in medicine and biology society* (pp. 1327–1330).

Golub, M.D., Chase, S.M., & Yu, B.M. (2013). Learning an internal dynamics model from control demonstration. In *Proceedings of the 30th international conference on machine learning* (pp. 606–614).

Golub, M.D., Yu, B.M., Schwartz, A.B., & Chase, S.M. (2014). Motor cortical control of movement speed with implications for brain-machine interface control. *Journal of Neurophysiology*, *112*(2), 411–429.

Gowda, S., Orsborn, A.L., Overduin, S.A., Moorman, H.G., & Carmena, J.M. (2014). Designing dynamical properties of brain-machine interfaces to optimize task-specific performance. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *22*(5), 911–920.

Hepp-Reymond, M.C., Kirkpatrick-Tanner, M., Gabernet, L., Qi, H.X., & Weber, B. (1999). Context-dependent force coding in motor and premotor cortical areas. *Experimental brain research*, *128*(1-2), 123–133.

Hochberg, L.R., Serruya, M.D., Friehs, G.M., Mukand, J.A., Saleh, M., Caplan, A.H., Branner, A., Chen, D., Penn, R.D., & Donoghue, J.P. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, *442*(7099), 164–171.

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

Hogan, N. (1985). The mechanics of multi-joint posture and movement control. *Biological Cybernetics*, *52*(5), 315–331.

Hogan, N., & Sternad, D. (2012). Dynamic primitives of motor behavior. *Biological Cybernetics*, *106*(11–12), 727–739.

Homer, M.L., Harrison, M.T., Black, M.J., Perge, J.A., Cash, S.S., Friehs, G., & Hochberg, L.R. (2013). Mixing decoded cursor velocity and position from an offline kalman filter improves cursor control in people with tetraplegia. In *Annual international conference of the IEEE engineering in medicine and biology society* (pp. 715–718).

Jarosiewicz, B., Chase, S.M., Fraser, G.W., Velliste, M., Kass, R.E., & Schwartz, A.B. (2008). Functional network reorganization during learning in a brain-computer interface paradigm. *Proceedings of the National Academy of Sciences*, *105*(49), 19,486–19,491.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, *82*(1), 35–45.

Kass, R.E., Ventura, V., & Brown, E.N. (2005). Statistical issues in the analysis of neuronal data. *Journal of Neurophysiology*, *94*(1), 8–25.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, *9*(6), 718–727.

Kim, S.P., Simeral, J.D., Hochberg, L.R., Donoghue, J.P., & Black, M.J. (2008). Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *Journal of Neural Engineering*, *5*(4), 455–476.

Kowalski, K.C., He, B.D., & Srinivasan, L. (2013). Dynamic analysis of naive adaptive brain-machine interfaces. *Neural Computation*, *25*(9), 2373–2420.

Koyama, S., Chase, S.M., Whitford, A.S., Velliste, M., Schwartz, A.B., & Kass, R.E. (2010a). Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control. *Journal of Computational Neuroscience*, *29*(1–2), 73–87.

Koyama, S., Pérez-Bolde, L.C., Shalizi, C.R., & Kass, R.E. (2010a). Approximate methods for state-space models. *Journal of the American Statistical Association*, *105*(489), 170–180.

Krakauer, J.W., Pine, Z.M., Ghilardi, M.F., & Ghez, C. (2000). Learning of visuomotor transformations for vectorial planning of reaching trajectories. *The Journal of Neuroscience*, *20*(23), 8916–8924.

Li, Z., O'Doherty, J.E., Hanson, T.L., Lebedev, M.A., Henriquez, C.S., & Nicolelis, M.A. (2009). Unscented kalman filter for brain-machine interfaces. *PloS ONE*, *4*(7), e6243.

Li, Z., O'Doherty, J.E., Lebedev, M.A., & Nicolelis, M.A.L. (2011). Adaptive decoding for Brain-Machine interfaces through bayesian parameter updates. *Neural Computation*, *23*(12), 3162–3204.

Liu, D., & Todorov, E. (2007). Evidence for the flexible sensorimotor strategies predicted by optimal feedback control. *The Journal of Neuroscience*, *27*(35), 9354–9368.

Malik, W.Q., Truccolo, W., Brown, E.N., & Hochberg, L.R. (2011). Efficient decoding with steady-state kalman filter in neural interface systems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *19*(1), 25–34.

Marathe, A.R., & Taylor, D.M. (2011). Decoding position, velocity, or goal: Does it matter for brain–machine interfaces? *Journal of Neural Engineering*, *8*(2), 025,016.

Musallam, S., Corneil, B.D., Greger, B., Scherberger, H., & Andersen, R.A. (2004). Cognitive control signals for neural prosthetics. *Science*, *305*(5681), 258–262.

Nazarpour, K., Ethier, C., Paninski, L., Rebesco, J.M., Miall, R.C., & Miller, L.E. (2012). Emg prediction from motor cortical recordings via a nonnegative point-process filter. *IEEE Transactions on Biomedical Engineering*, *59*(7), 1829–1838.

Nishimura, Y., Perlmutter, S.I., & Fetz, E. E. (2013). Restoration of upper limb movement via artificial corticospinal and musculospinal connections in a monkey with spinal cord injury. *Frontiers in neural circuits*, 7.

Oby, E.R., Ethier, C., & Miller, L.E. (2013). Movement representation in the primary motor cortex and its contribution to generalizable emg predictions. *Journal of Neurophysiology*, *109*(3), 666–678.

Orsborn, A.L., Moorman, H.G., Overduin, S.A., Shanechi, M.M., & Carmena, D.F.D.J.M. (2014). Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron*, *82*(6), 1380–1393.

Paz, R., Boraud, T., Natan, C., Bergman, H., & Vaadia, E. (2003). Preparatory activity in motor cortex reflects learning of local visuomotor skills. *Nature Neuroscience*, *6*(8), 882–890.

Paz, R., Nathan, C., Boraud, T., Bergman, H., & Vaadia, E. (2005). Acquisition and generalization of visuomotor transformations by nonhuman primates. *Experimental Brain Research*, *161*(2), 209–219.

Rokni, U., Richardson, A.G., Bizzi, E., & Seung, H.S. (2007). Motor learning with unstable neural representations. *Neuron*, *54*(4), 653–666.

Sadtler, P.T., Quick, K.M., Golub, M.D., Chase, S.M., Ryu, S.I., Tyler-Kabara, E.C., Yu, B.M., & Batista, A.P. (2014). Neural constraints on learning. *Nature*, *512*, 423–426.

Salinas, E., & Abbott, L.F. (1994). Vector reconstruction from firing rates. *Journal of Computational Neuroscience*, *1*(1–2), 89–107.

Schieber, M.H. (2011). Dissociating motor cortex from the motor. *Journal of Physiology*, *589*(23), 5613–5624.

Schwartz, A.B., Cui, X.T., Weber, D.J., & Moran, D.W. (2006). Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron*, *52*, 205–220.

Serruya, M.D., Hatsopoulos, N.G., Paninski, L., Fellows, M.R., & Donoghue, J.P. (2002). Instant neural control of a movement signal. *Nature*, *416*(6877), 141–142.

Shadmehr, R., Smith, M.A., & Krakauer, J.W. (2010). Error correction, sensory prediction, and adaptation in motor control. *Annual Review of Neuroscience*, *33*, 89–108.

Shanechi, M.M., Williams, Z.M., Wornell, G.W., Hu, R.C., Powers, M., & Brown, E.N. (2013). A real-time brain-machine interface combining motor target and trajectory intent using an optimal feedback control design. *PloS ONE*, *8*(4), e59,049.

Shenoy, K.V., Sahani, M., & Churchland, M.M. (2013). Cortical control of arm movements: a dynamical systems perspective. *Annual review of neuroscience*, *36*, 337–359.

Shpigelman, L., Lalazar, H., & Vaadia, E. (2009). Kernel-ARMA for hand tracking and Brain-machine interfacing during 3d motor control. In *Advances in neural information processing systems* (pp. 1489–1496).

Sommer, M.A., & Wurtz, R.H. (2002). A pathway in primate brain for internal monitoring of movements. *Science*, *296*(5572), 1480–1482.

Suminski, A.J., Fagg, A.H., Willett, F.R., Bodenhamer, M., & Hatsopoulos, N.G. (2013). Online adaptive decoding of intended movements with a hybrid kinetic and kinematic brain machine interface. In *Proceedings of the 35th annual international conference of the ieee engineering in medicine and biology society* (pp. 1583–1586).

Taylor, D.M., Tillery, S.I., & Schwartz, A.B. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science*, *296*, 1829–1832.

Tillery, S.I.H., & Taylor, D.M. (2004). Signal acquisition and analysis for cortical control of neuroprosthetics. *Current Opinion in Neurobiology*, *14*(6), 758–762.

Tin, C., & Poon, C.S. (2005). Internal models in sensorimotor integration: perspectives from adaptive control theory. *Journal of Neural Engineering*, *2*(3), S147—S163.

Velliste, M., Perel, S., Spalding, M.C., Whitford, A.S., & Schwartz, A.B. (2008). Cortical control of a prosthetic arm for self-feeding. *Nature*, *453*, 1098–1101.

Wise, S.P., Moody, S.L., Blomstrom, K.J., & Mitz, A.R. (1998). Changes in motor cortical activity during visuomotor adaptation. *Experimental Brain Research*, *121*(3), 285–299.

Wolpert, D.M., Ghahramani, Z., & Jordan, M.I. (1995). An internal model for sensorimotor integration. *Science*, *269*(5232), 1880–1882.

Wu, H.G., & Smith, M.A. (2013). The generalization of visuomotor learning to untrained movements and movement sequences based on movement vector and goal location remapping. *The Journal of Neuroscience*, *33*(26), 10,772–10,789.

Wu, W., Black, M.J., Gao, Y., Bienenstock, E., Serruya, M., Shaikhouni, A., & Donoghue, J.P. (2003). Neural decoding of cursor motion using a Kalman filter. In *Advances in neural information processing systems* (pp. 133–140).

Wu, W., Gao, Y., Bienenstock, E., Donoghue, J.P., & Black, M.J. (2006). Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Computation*, *18*(1), 80–118.

Zach, N., Inbar, D., Grinvald, Y., Bergman, H., & Vaadia, E. (2008). Emergence of novel representations in primary motor cortex and premotor neurons during associative learning. *Journal of Neuroscience*, *28*(38), 9545–9556.

Zhang Y, & Chase SM (2013). A stabilized dual kalman filter for adaptive tracking of brain-computer interface decoding parameters. In *Proceedings of the 35th annual international conference of the ieee engineering in medicine and biology society* (pp. 7100–7103).