LETTER **Communicated by Joseph Makin**

Neural Decoding: A Predictive Viewpoint

Sonia Todorova*
Valérie Ventura
vventura@stat.cmu.edu
Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213, and Center for the Neural Basis of Cognition, Pittsburgh, PA 15213, U.S.A.

Decoding in the context of brain-machine interface is a prediction problem, with the aim of retrieving the most accurate kinematic predictions attainable from the available neural signals. While selecting models that reduce the prediction error is done to various degrees, decoding has not received the attention that the fields of statistics and machine learning have lavished on the prediction problem in the past two decades. Here, we take a more systematic approach to the decoding prediction problem and search for risk-optimized reverse regression, optimal linear estimation (OLE), and Kalman filter models within a large model space composed of several nonlinear transformations of neural spike counts at multiple temporal lags. The reverse regression decoding framework is a standard prediction problem, where penalized methods such as ridge regression or Lasso are routinely used to find minimum risk models. We argue that minimum risk reverse regression is always more efficient than OLE and also happens to be 44% more efficient than a standard Kalman filter in a particular application of offline reconstruction of arm reaches of a rhesus macaque monkey. Yet model selection for tuning curves–based decoding models such as OLE and Kalman filtering is not a standard statistical prediction problem, and no efficient method exists to identify minimum risk models. We apply several methods to build low-risk models and show that in our application, a Kalman filter that includes multiple carefully chosen observation equations per neural unit is 67% more efficient than a standard Kalman filter, but with the drawback that finding such a model is computationally very costly.

1 Introduction

Encoding models linking neural activity to biological and behavioral variables have long been used to understand how neural systems represent information or demonstrate that information about a signal is present and can be extracted from models of neural activity (Perel et al., 2013; Nishimoto &

*Deceased.

doi:10.1162/NECO_a_01020

Gallant, 2011; Keat, Reinagel, Reid, & Meister, 2001; Pillow, Paninski, Uzzell, Simoncelli, & Chichilnisky, 2005; Stanley, 2013; Rieke, Warland, de Ruyter van Steveninck, & Bialek, 1997). In this letter, we focus specifically on decoding kinematic variables from spike-based encoding models in the context of brain-computer interfaces (BCIs).

Building efficient decoding models is crucial for BCIs to obtain the most accurate kinematic predictions possible. We briefly describe some approaches to build and optimize decoding models. Decoding initially relied mostly on modeling the kinematics as linear or additive nonparametric functions of lagged spike counts (Rieke et al., 1997; Warland, Reinagel, & Meister, 1997), with the lag often chosen to maximize R^2 of the model. Mulliken, Musallam, & Andersen (2008) included spike counts at three consecutive lags to induce smoothness in the predictions and fitted the model by ridge regression to minimize its prediction error. Another class of decoders relies on physiological models of neurons' firing rates as functions of kinematics (the tuning curves) and includes the optimal linear decoder (Salinas & Abbott, 1994) and the Kalman filter (Brown, Frank, Tang, Quirk, & Wilson, 1998). The latter incorporates a prior kinematic model that can improve predictions substantially. In this framework, spike counts are typically assumed to be gaussian or Poisson distributed (Truccolo, Eden, Fellows, Donoghue, & Brown, 2005) with means assumed to be additive functions of the kinematics, lagged by a time chosen to maximize the model R^2 or minimize the prediction mean squared error (Wu, Gao, Bienenstock, Donoghue, & Black, 2006). Response transformations can improve the fit of such models to data; for example, square roots of spike counts may be more gaussian with more constant variance (Moran & Schwartz, 1999; Wu et al., 2006). Selecting which neurons to use, for example, by rating the amount of kinematic information they encode (Vargas-Irwin et al., 2010; Liao, Wang, Zheng, & Principe, 2012), can also improve decoding (Kahn, Sheiber, Thakor, & Sarma, 2011). Rather than select neurons, Tankus, Fried, and Shoham (2012) suggest identifying a sparse decomposition of neural features and decoding from a low-dimensional projection.

These examples illustrate the wide range of model families and model selection procedures that have been successful in improving decoding performance. But they also show that the search for good models is not done systematically or with the exclusive objective of minimizing the prediction error. In this letter, we focus specifically on this objective. We search for risk-optimized models within a large model space composed of parametric and nonparametric transformations of neural spike counts at multiple temporal lags, applying model selection techniques that let the data determine which variables should enter the models to minimize the prediction error. While the reverse regression decoding framework is a standard statistical prediction problem, where penalized methods such as ridge regression or Lasso are routinely used to find minimum risk models, model selection for tuning curves–based decoding models such as

OLE and Kalman filtering is not standard, and no efficient method exists to identify minimum risk models. We thus investigate the effectiveness of several existing and new ad hoc methods to build low-risk models.

2 Methods

Assume that we record the signal of n electrodes to decode the three-dimensional velocity $\mathbf{v}_t = (v_{1t}, v_{2t}, v_{3t})$ of an arm at time t; the methods described here apply to other kinematic variables. We use *neural units* to mean neurons or unsorted electrodes treated as putative neurons and *spikes* to mean extracellular voltage threshold crossings. For simplicity and brevity, we did not spike-sort the electrode signals in this letter; we observed comparable results with spike-sorted data. We let $\mathbf{c}_t$ denote a vector of neural covariates, which we can initially think of as $\mathbf{c}_t = \mathbf{s}_{t-\tau}$, the $n \times 1$ vector of the n neural units' spike counts in the time bins centered at $t - \tau$, where τ is an appropriate lag between neural activity and kinematics; later we include several lags and transformations of spike counts (see equation 2.7).

2.1 Linear Decoding Models. We assume that the lagged spike counts $\mathbf{c}_t$ are gaussian with linear tuning curves:

$$\mathbf{c}_t = \beta + \mathbf{B}\mathbf{v}_t + \eta_t, \tag{2.1}$$

where β is an $n \times 1$ vector and $\mathbf{B}$ an $n \times 3$ matrix of regression coefficients and η_t is a $n \times 1$ gaussian noise vector with covariance matrix $\mathbf{U}$; assume that β, $\mathbf{B}$, and $\mathbf{U}$ are fitted to training data by maximum likelihood (ML). We consider decoding $\mathbf{v}_t$ using optimal linear estimation (OLE) and Kalman filtering (KF) so we can compute the predicted kinematics in closed form. The former consists of estimating $\mathbf{v}_t$ by ML in equation 2.1:

$$\tilde{\mathbf{v}}_t^{OLE} = \mathbf{M}^{OLE}(\mathbf{c}_t - \beta), \tag{2.2}$$

where $\mathbf{M}^{OLE} = (\mathbf{B}'\mathbf{U}^{-1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{U}^{-1}$. The latter supplements equation 2.1 with a kinematic evolution model. Here we use an autoregressive process of order one:

$$\mathbf{v}_t = \mathbf{A}\,\mathbf{v}_{t-1} + \epsilon_t, \tag{2.3}$$

where $\mathbf{A}$ is a 3×3 matrix of coefficients and ϵ_t are gaussian perturbations with covariance matrix $\mathbf{W}$, with $\mathbf{A}$ and $\mathbf{W}$ fitted to training data. Equation 2.3 serves as a prior model for equation 2.1, and the decoded velocity is the mean of the posterior distribution of $\mathbf{v}_t$ given the spike counts up to time $t - \tau$, obtained using the Kalman recursive equations,

$$\tilde{\mathbf{v}}_t^{KF} = \mathbf{M}_t^{KF}(\mathbf{c}_t - \beta) + (\mathbf{I} - \mathbf{M}_t^{KF}\mathbf{B})\mathbf{A}\tilde{\mathbf{v}}_{t-1}^{KF}, \tag{2.4}$$

where $\mathbf{M}_t^{KF} = \mathbf{\Sigma}_{t|t-1}\mathbf{B}'(\mathbf{B}\mathbf{\Sigma}_{t|t-1}\mathbf{B}' + \mathbf{U})^{-1}$, $\mathbf{\Sigma}_{t|t-1}$ satisfies the Riccati recursion $\mathbf{\Sigma}_{t+1|t} = \mathbf{A}\mathbf{\Sigma}_{t|t-1}\mathbf{A}' + \mathbf{W} - \mathbf{A}\mathbf{\Sigma}_{t|t-1}\mathbf{B}'(\mathbf{B}\mathbf{\Sigma}_{t|t-1}\mathbf{B}' + \mathbf{U})^{-1}\mathbf{B}\mathbf{\Sigma}_{t|t-1}\mathbf{A}'$ with initial condition $\mathbf{\Sigma}_{\mathbf{0}|-\mathbf{1}} = \Sigma_0$, and Σ_0 is the covariance matrix of the initial velocity $\mathbf{v}_0$ (Kalman, 1960; Brown et al., 1998). In section 3, we initiate the KF decoder at the true velocity and set $\Sigma_0 = 0$. Note that in keeping with KF terminology, we will refer to tuning curves such as those in equation 2.1 as observation equations.

Nonlinear nongaussian observation equations have also been used (Truccolo et al., 2005), but we do not consider them here because they do not yield closed-form predictions and may thus be too computationally expensive for real-time decoding. However, in section 2.4, we introduce nonparametric response transformation observation equations, which can capture nonlinear relationship between velocity and spike counts yet remain linear in $\mathbf{v}_t$ and thus yield closed-form predictions.

Reversing the roles of $\mathbf{v}_t$ and $\mathbf{c}_t$ in equation 2.1 yields another class of linear decoders called forward filters or reverse regressions (RR). They model the kinematics as linear combinations of spike counts: $\mathbf{v}_t = \delta + \mathbf{M}^{RR}\,\mathbf{c}_t + \zeta_t$, yielding predictions:

$$\tilde{\mathbf{v}}_t^{RR} = \delta + \mathbf{M}^{RR}\,\mathbf{c}_t, \tag{2.5}$$

where δ and $\mathbf{M}^{RR}$ are a 3×1 vector and $3 \times n$ matrix of regression coefficients fitted to training data and ζ_t is a 3×1 gaussian noise vector.[1] We also consider combining the RR model with the kinematic model in equation 2.3 to form an autoregressive reverse regression (AR-RR) model: $\mathbf{v}_t = \alpha + \mathbf{M}^{AR}\,\mathbf{c}_t + \mathbf{A}^{AR}\,\mathbf{v}_{t-1} + \chi_t$, where α, $\mathbf{M}^{AR}$, and $\mathbf{A}^{AR}$ are regression coefficients fitted to training data and χ_t is gaussian noise, from which predictions are recursively obtained as

$$\tilde{\mathbf{v}}_t^{AR} = \alpha + \mathbf{M}^{AR}\,\mathbf{c}_t + \mathbf{A}^{AR}\,\tilde{\mathbf{v}}_{t-1}^{AR}. \tag{2.6}$$

Equations 2.2 and 2.4 to 2.6 define four classes of velocity predictions that we summarized as graphs in Figure 3 in appendix A. They are all linear combinations of the same neural covariates $\mathbf{c}_t$, so we should be able to compare their efficiencies (see section 2.5). But first we specify the model space that we will search for efficient models.

2.2 Decoding Model Space. The common practice is to use at most one velocity encoding quantity per neural unit in a decoding model—for example, $\mathbf{c}_t = \mathbf{s}_{t-\tau}$, the spike counts lagged by τ, or their square roots, if judged appropriate. Here we explore a much larger space of models, allowing $\mathbf{c}_t$

[1] ζ_t is typically assumed to have independent components. In that case, the models for each velocity component v_{kt}, $k = 1, 2, 3$, are estimated independently of one another.

to be any subset of a large set $\mathbf{C}_t$ composed of transformed and untransformed spike counts at several lags, and we apply model selection techniques that let the data determine which variables should enter the model to minimize the prediction error. The resulting model may therefore include several transformed and untransformed spike counts per neural unit, possibly at several lags, while some units may not be represented at all.

Specifically, we set

$$\mathbf{C}_t = \left\{ \mathbf{s}_{t-i}, \mathbf{s}_{t-i}^{\frac{1}{2}}, \mathbf{s}_{t-i}^{2}, \mathbf{s}_{t-i}^{3}, \mathrm{npm}(\mathbf{s}_{t-i}), i = 0, 1, \ldots, 12 \right\}, \tag{2.7}$$

where the nonparametric transformation of the spike counts $\mathrm{npm}(\mathbf{s}_{t-i})$, which we specify in sections 2.3 and 2.4, and power transformations may capture possible nonlinearities between firing rates and kinematics, and the temporal lags allow spike counts and kinematics to be contemporary ($i = 0$) or lagged by up to 12 time bins, corresponding to 192 ms with the 16 ms wide bins we use in section 3. Note that while the square root transformation is sometimes used (Moran & Schwartz, 1999; Wu et al., 2006), the squared or cubed is not; we rely on the model selection procedure to identify useful transformations. Other lags and transformations could be added to $\mathbf{C}_t$, as well as any other available quantities known to encode kinematics—for example, waveform moments when decoding from unsorted electrodes (Ventura & Todorova, 2015).

Our goal in this letter is to identify the minimum risk model across the four linear velocity prediction classes within the linear model space defined by $\mathbf{C}_t$, where we measure the risk of predicting the true velocity $\mathbf{v}$ with $\tilde{\mathbf{v}}$ by the expectation of the squared distance between $\mathbf{v}$ with $\tilde{\mathbf{v}}$, $E(||\mathbf{v}_t - \tilde{\mathbf{v}}_t||_2^2)$, and estimate it by the mean squared error (MSE) in a training set of size T:

$$\mathrm{MSE} = T^{-1} \sum_{t=1}^{T} ||\mathbf{v}_t - \tilde{\mathbf{v}}_t||_2^2 = T^{-1} \sum_{t=1}^{T} \left[\sum_{k=1}^{3} (v_{k,t} - \tilde{v}_{k,t})^2 \right]. \tag{2.8}$$

2.3 Model Selection for Reverse Regression. Reverse regression predictions that have been considered include equation 2.5 with $\mathbf{c}_t = \mathbf{s}_{t-\tau}$, the spike counts lagged by τ, and extensions that replace $\mathbf{s}_{t-\tau}$ by nonparametric transformations of $\mathbf{s}_{t-\tau}$ such as splines (Rieke et al., 1997; Warland et al., 1997; Wagenaar, Ventura, & Weber, 2009). Mulliken et al. (2008) used three consecutive lagged spike counts, that is, $\mathbf{c}_t = (\mathbf{s}_{t-\tau}, \mathbf{s}_{t-\tau-1}, \mathbf{s}_{t-\tau-2})$, and fit the model using ridge regression to minimize the prediction risk. Here we consider basic RR with $\mathbf{c}_t = \mathbf{s}_{t-\tau}$, and minimum risk RR, the prediction that minimizes the cross-validated risk estimate (see equation 2.8) within the space defined by equation 2.7, where we take $\mathrm{npm}(\mathbf{s}_{t-i})$ to be a smoothing spline of $\mathbf{s}_{t-i}$ with four degrees of freedom. The other models mentioned

above belong to our model space and are therefore necessarily less efficient than minimum-risk RR.

An exhaustive search through the model space to identify the minimum-risk model is prohibitively time consuming with the large number of predictors we allow, as is stepwise regression. Regularized regression is faster. It consists of including all $p = \text{card}(\mathbf{C_t})$ available predictors in equation 2.5, first standardized to control for differences in scale and penalizing the norm of their regression coefficients by a factor chosen to minimize the prediction risk; Lasso (Tibshirani, 1996) and ridge regression use the L_1 and L_2 norms, respectively. The ridge regression coefficients for the kth component of velocity in the kth row of $\mathbf{M}^{RR}$ (see equation 2.5) are estimated by $(\mathbf{X}^T\mathbf{X} + \alpha_k\mathbf{I})^{-1}\mathbf{X}^T(\mathbf{v}_k - \bar{\mathbf{v}}_{k\cdot})$, where $\mathbf{X}$ is the $T \times p$ matrix with rows $\mathbf{C}_t$, $t = 1, \ldots, T$; T is the size of the training data; $\mathbf{v}_k = \{v_{kt}, t = 1, \ldots, T\}$ is the vector of the kth velocity component values; $\bar{\mathbf{v}}_{k\cdot}$ is the average of $\mathbf{v}_k$ over t; $\mathbf{I}$ is the $p \times p$ identity matrix; and $\{\alpha_k, k = 1, 2, 3\}$ are chosen to minimize the cross-validated risk estimate (see equation 2.8) of the predicted velocity. The Lasso estimates do not have closed forms. In section 3, we report only the performances of ridge estimates for brevity; Lasso results were practically identical.

Finally, we also consider minimum-risk AR-RR models obtained by fitting equation 2.6 using ridge regression. We apply either (1) the same penalty on $\tilde{\mathbf{v}}_{t-1}^{AR}$ and $\mathbf{c}_t$ for simplicity, (2) two different penalties on $\tilde{\mathbf{v}}_{t-1}^{AR}$ and $\mathbf{c}_t$ since they measure different types of quantities, or (3) a penalty on $\mathbf{c}_t$ and none on $\mathbf{v}_{t-1}$ to preserve the kinematic model, since it captures a physical process. Option 3 effectively is a minimum-risk kinematic model with penalized neural control variables, and option 2 with an infinite penalty on $\tilde{\mathbf{v}}_{t-1}^{AR}$ yields the minimum-risk RR model.

2.4 Model Selection for OLE and Kalman Filtering. Decoding in the OLE and KF framework is a parameter estimation rather than a prediction problem, since we want to predict the covariate $\mathbf{v}_t$ in a regression model like equation 2.1; the typical prediction framework consists instead of predicting the response variable of a regression.

The prediction risk, estimated in equation 2.8, can be expressed as the sum of the variance and squared bias of the estimate of $\mathbf{v}_t$. Given a decoding/regression model like equation 2.1, OLE predictions are maximum likelihood estimates (MLEs), so they are unbiased if the observation equations represent true relationships and they have minimum variance; moreover, the variance decreases with the number of observations, which in our context is the number of observation equations. We could therefore reduce the MLE variance by including in the decoding model several observation equations per neural units, for example, if spike counts encode kinematics at several lags or if quantities other than spike counts also encode kinematics such as waveform moments (Ventura & Todorova, 2015). However, it is unlikely that the observation equations are true models—indeed, models

are often useful but seldom correct so they may induce bias in the MLE; they also contain noise since they are estimated from training data,[2] so using too many will contribute more noise than signal and increase the MLE variance. Model selection for OLE decoding therefore consists of choosing which observation equations to include in the model (see equation 2.1) to balance the bias and variance of the predictions.

Model selection for KF also amounts to choosing observation equations that balance the prediction bias and variance, although they will likely be different from the equations chosen for OLE. Indeed, the KF prediction combines the MLE (OLE) and prior predictions, so their biases and variances combine too. For example, KF and OLE predictions are equal when the prior model, equation 2.3, has infinite variance so the same equations would be chosen, while a small prior variance reduces the variance and increases the bias of the KF prediction compared to the OLE prediction, so different equations would be chosen.

We select observation equations from the model space $\mathbf{C}_t$ in equation 2.7,

$$g(\mathbf{s}_{t-i}) = \beta_{0gi} + \mathbf{B}_{gi}\,\mathbf{v}_t + \eta_{git}, \quad i = 0, 1, \ldots, 12, \tag{2.9}$$

with $g(s) = s^p$, $p = 1/2, 1, 2, 3$ (see equation 2.9 with $p = 1$ and $i = \tau$ is equation 2.1). We also include a nonparametric response transformation model by taking g to be the alternative conditional expectation (ACE) of Breiman and Friedman (1985), which seeks a function g that maximizes the correlation between the left- and right-hand sides of equation 2.9. We fit a specific ACE function g for each neural unit and each lag as follows after initializing $\hat{g}(s_{t-i}) = s_{t-i}$, $t = 1, \ldots, T$: (1) standardize $\hat{g}(s_{t-i})$; (2) regress $\hat{g}(s_{t-i})$ on $\mathbf{v}_t$ to obtain $(\hat{\beta}_{0gi} + \widehat{\mathbf{B}}_{gi}\,\mathbf{v}_t)$; (3) regress $(\hat{\beta}_{0gi} + \widehat{\mathbf{B}}_{gi}\,\mathbf{v}_t)$ on s_{t-i} using a smoothing spline g to obtain $\hat{g}(s_{t-i})$; and repeat steps 1 to 3 until convergence. That is, the first iteration consists of fitting a linear tuning curve as in equation 2.1, and if the tuning curve is not actually linear, the subsequent iterations seek a better fit by replacing the spike counts s_{t-i} with a more flexible function of s_{t-i}, instead of the usual approach of fitting a nonlinear tuning curve to s_{t-i}. The advantage of this approach is that equation 2.9 remains linear in $\mathbf{v}_t$, so that velocity predictions are obtained in closed form.

Exhaustive and stepwise searches through the model space defined by equation 2.9 to identify the minimum cross-validated risk OLE and KF models are prohibitively time-consuming with the many observation equations considered here, and we know of no computationally efficient alternative search method like penalized regression, so we will investigate ad hoc model building strategies to identify low-risk OLE and low-risk KF models

[2] Decoding relies on a regression model, but unlike in typical regression problems, the covariates are not observed; rather, they are estimated from training data in the encoding stage.

in section 3. We also consider the commonly used basic OLE and basic KF models, based on the observation equations in equation 2.1 with $\mathbf{c}_t = \mathbf{s}_{t-\tau}$, the spike counts lagged by τ.

2.5 Comparing the Four Classes of Predictors. The classes of predictors we consider—OLE in equation 2.2, KF in equation 2.4, RR in equation 2.5, and AR-RR in equation 2.6—are all linear in $\mathbf{c}_t \in \mathbf{C}_t$ (see equation 2.7). There are 2^p different subsets $\mathbf{c}_t$ that either include or exclude each of the p variables in $\mathbf{C}_t$. To identify the minimum-risk prediction in that model space, we should fit all 2^p models to data by maximum likelihood, evaluate their prediction risks, and choose the model with the smallest risk. If the true prediction is indeed linear in $\mathbf{c}_t \in \mathbf{C}_t$, so that it belongs to the model space defined by equation 2.7, this procedure ensures that we identify it, up to the error of fitting it to data (this error decreases as the sample size increases). If the true prediction is not linear, this procedure identifies the linear prediction that is closest to the true prediction.

The exhaustive search described is precisely how one identifies the minimum-risk RR prediction; hence, minimum-risk RR is the best linear prediction within the model space defined by equation 2.7.[3] The OLE prediction, equation 2.2, is also linear in $\mathbf{c}_t \in \mathbf{C}_t$ and therefore cannot be more efficient than minimum-risk RR. This makes sense: the OLE model coefficients in equation 2.2 are functions of the observation equations' coefficients in equation 2.1 according to $\mathbf{M}^{OLE} = (\mathbf{B}'\mathbf{U}^{-1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{U}^{-1}$, so they are not fitted to data with the specific objective of minimizing the prediction error. In contrast, $\mathbf{M}^{RR}$ in equation 2.5 is unconstrained and fitted to data to minimize the prediction risk.

For AR-RR predictions, we replace $\tilde{\mathbf{v}}_{t-1}^{AR}$ on the right-hand side of equation 2.6 by the left-hand side of equation 2.6 with t replaced by $t-1$, and so on recursively, so that AR-RR predictions can be written as linear combinations of $\mathbf{c}_t$, $\mathbf{c}_{t-1}$, $\mathbf{c}_{t-2}$, and so on. Depending on which elements of $\mathbf{C}_t$ compose $\mathbf{c}_t$, the set $\{\mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{c}_{t-2}, \ldots\}$ may contain spike counts that are older than 12 lags, in which case AR-RR predictions do not belong to the model space defined by $\mathbf{C}_t$. However, neural activity this old should not have meaningful predictive power, so AR-RR models are unlikely to outperform minimum-risk RR when the sample space includes many lags. It makes sense: if we could use the true velocity $\mathbf{v}_{t-1}$ on the right-hand side of equation 2.6, and if, as is likely the case, $\mathbf{v}_{t-1}$ provides information about $\mathbf{v}_t$ that $\mathbf{C}_t$ does not already contain, then the best AR-RR prediction would certainly outperform the best RR prediction since it uses an informative additional predictor (compare equations 2.5 and 2.6). However, we use $\tilde{\mathbf{v}}_{t-1}^{AR}$ and not $\mathbf{v}_{t-1}$ in

[3] There is a caveat: we fit the minimum risk of a convex relaxation of the model, which is not equal to the corresponding unrelaxed minimum-risk model; no result exists to evaluate the distance from one model to the other.

equation 2.6, which is a function of $\mathbf{C}_{t-1}$ that is unlikely to contain much, if any, additional information about $\mathbf{v}_t$ that is not already provided by $\mathbf{C}_t$, provided $\mathbf{C}_t$ includes many lags. We actually expect AR-RR predictions to underperform more or less depending on how we penalize $\tilde{\mathbf{v}}_{t-1}^{AR}$ and $\mathbf{c}_t$ in equation 2.6. Using no penalty on $\tilde{\mathbf{v}}_{t-1}^{AR}$ (option 3 at the end of section 2.3) forces $\tilde{\mathbf{v}}_t^{AR}$ to be close to $\tilde{\mathbf{v}}_{t-1}^{AR}$, which could accumulate the prediction error as t increases if $\tilde{\mathbf{v}}_{t-1}^{AR}$ is a poor prediction of $\mathbf{v}_{t-1}$. The other two options penalize a potentially bad prediction $\tilde{\mathbf{v}}_{t-1}^{AR}$ so they might fare better, especially option 2, since the penalty on $\tilde{\mathbf{v}}_{t-1}^{AR}$ is independent of the penalty on $\mathbf{c}_t$; setting the penalty on $\tilde{\mathbf{v}}_{t-1}^{AR}$ to infinity removes the influence of $\tilde{\mathbf{v}}_{t-1}^{AR}$ entirely and yields the minimum-risk RR prediction.

Finally, KF predictions (see equation 2.4) can also be recursively expressed as linear combinations of $\{\mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{c}_{t-2}, \ldots\}$, so they might not outperform minimum-risk RR predictions, as we concluded already for AR-RR predictions. Additionally, like OLE predictions, KF predictions are constrained: $\mathbf{M}_t^{KF}$ in equation 2.4 is a function of the coefficients of the observation equations in equation 2.1 and kinematic model in equation 2.3, so that unlike the minimum-risk RR prediction parameters $\mathbf{M}^{RR}$ in equation 2.5, $\mathbf{M}_t^{KF}$ is not fitted to data with the specific objective of minimizing the prediction error. However, KF predictions are different in two ways. First, the model is not fixed within the model space, since its coefficients $\mathbf{M}_t^{KF}$ are functions of time t; hence its prediction risk is an average over t of the risks at each t, which is not comparable to the risk of a nondynamic decoder: it could be larger or smaller than the risk of minimum-risk RR, depending on the application. Second, for each reach we decode, we initialize the decoders at the observed initial velocity; then the kinematic prior model constrains the next few KF predictions to be close to the initial velocity, which happens to match the speed profile of a typical reach. Therefore, KF models need only decode well the last portion of reaches, and because the reaches in our data set are short, KF predictions may be more efficient than minimum-risk RR predictions. They may not be as efficient to decode continuous motion.

3 Results

We illustrate the benefit of using models that minimize the prediction error to decode the arm velocity of a rhesus macaque in an experiment performed in Andrew Schwartz's MotorLab (Fraser & Schwartz), 2012; Todorova, Sadtler, Batista, Chase, & Ventura, 2014). The data were recorded during four days, with five sessions per day and one session consisting of 52 reaches to and from 27 targets arranged evenly on a virtual 3D sphere (see Figure 1A). We analyze the portion of the reaches between movement onset and target acquisition, which amounts to 8 minutes of data. The neural activity was recorded in ventral premotor cortex on the 71 active channels of a 96-electrode Utah array. We decode from threshold crossings without spike

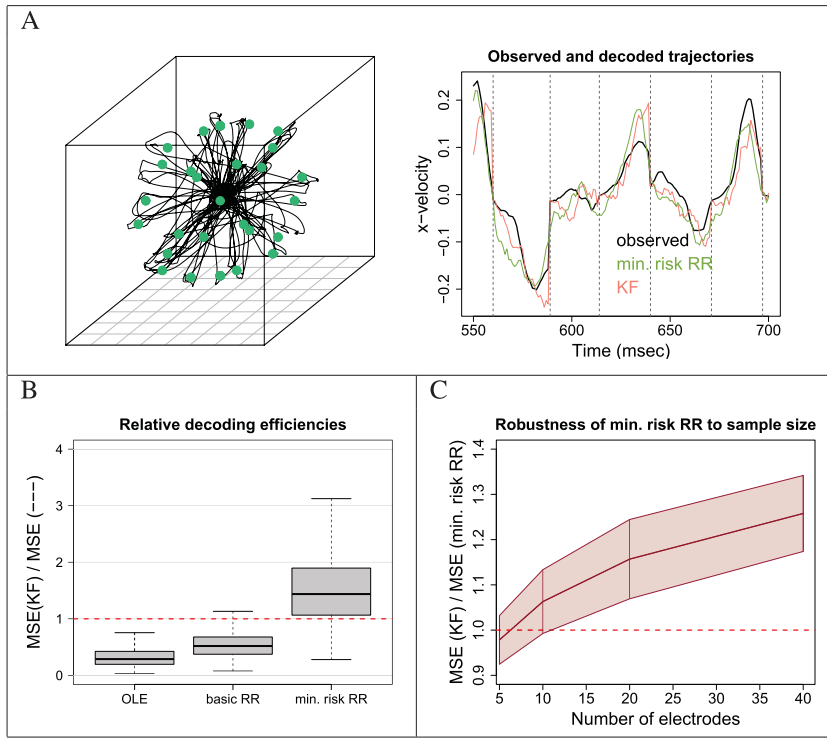

Figure 1: (A) Experimental data. Arm position over several trials of center-out and out-center reaches in 3D and sample of observed and reconstructed x-velocity trajectories. (B) Efficiency of OLE and reverse regressions (RR) relative to Kalman filtering (KF). Basic RR, OLE, and KF all predict kinematics using spike counts $\mathbf{s}_{t-\tau}$ lagged by $\tau = 5$ bins (80 ms); minimum-risk RR uses all the lagged and transformed spike counts in equation 2.7 and is fitted by ridge regression to minimize the prediction risk. The accuracy of a decoded reach is measured by its MSE (see equation 2.8) and the relative efficiency of two decoders by their MSE ratio. The box plots summarize the distributions of MSE ratios across the 1040 test reaches when comparing basic KF to the decoders listed on the x-axis: OLE and minimum-risk RR are, respectively, the least and most efficient decoders; the latter is 44% more efficient than KF, in median. (C) Robustness analysis: median $\pm$ SD efficiency of minimum risk RR relative to KF across 100 random subsamples of n electrodes, $n = 5, 10, 20, 40$: minimum-risk RR is more efficient than KF, in median, at all sample sizes greater than $n = 5$.

sorting, using 16 ms spike count bins initially lagged $\tau = 5$ bins (80 ms) compared to arm movement for all electrodes, where $\tau = 5$ is the best uniform lag, that is, the integer value in $[0, 12]$ that maximizes the average R^2

of the electrodes' tuning curve models in equation 2.1.[4] We use all combinations of four sessions recorded the same day ($4 \times 52 = 208$ trials) to train the decoding models and evaluate their performances to decode the 52 reaches of the remaining session of that day, totaling $5 \times 4 = 20$ training sets and $20 \times 52 = 1040$ test trials. We initialize the decoders at the observed initial velocity for each trial. We measure the relative efficiency of two decoders to reconstruct a test trial by the ratio of their respective mean squared errors (MSE, equation 2.8) and summarize the distribution of MSE ratios across the 1040 test trials using box plots.

We start with the decoding methods that are computationally easy to implement: the basic OLE, basic KF, minimum-risk RR, and the three minimum-risk AR-RR models. Figure 1B shows the distribution over the 1040 test trials of the decoding efficiencies of basic OLE (equation 2.2 with $\mathbf{c}_t = \mathbf{s}_{t-\tau}$), basic RR (equation 2.5 with $\mathbf{c}_t = \mathbf{s}_{t-\tau}$), and minimum-risk RR (see section 2.3), relative to basic KF (equation 2.4 with $\mathbf{c}_t = \mathbf{s}_{t-\tau}$), where relative efficiencies greater than one mean that KF is less efficient than the competing method. Basic OLE is the least efficient. Basic OLE and basic RR use the same information, the lagged spike counts $\mathbf{c}_t = \mathbf{s}_{t-\tau}$ as observation equations and predictors, respectively, yet the latter is about twice as efficient in median across the 1040 test trials. Both methods are less efficient than KF on most trials because KF uses the extra kinematic information provided by the state equation (see equation 2.3), which is akin to using information from previous spike counts (see section 2.4). For example, RR with $\mathbf{c}_t = (\mathbf{s}_{t-\tau}, \mathbf{s}_{t-\tau-1})$ improves the decoding efficiency of basic RR by over 30% in median (not shown). Finally, minimum-risk RR uses all the covariates in equation 2.7 to capture linear and nonlinear associations between velocity and spike counts at lags up to 12×16 ms $= 192$ ms and is fitted by ridge regression to minimize its prediction risk, using 10-fold cross-validation to prevent overfitting; it is five times more efficient than OLE (not shown) and 44% more efficient than KF in median across the 1040 test trials (see Figure 1B, third box plot). The spike counts and their square roots have the largest ridge regression coefficients (see appendix B, Figure 4A), but it is the inclusion of multiple lags that matters most, certainly because they help smooth the predictions, but perhaps also because the spike counts encode the kinematics across several time bins. Finally, as conjectured in section 2.5, none of the minimum-risk autoregressive RR models (see equation 2.6) outperformed minimum-risk RR; the models that do not penalize $\tilde{\mathbf{v}}_{t-1}^{AR}$ at all or not

[4] We analyzed these data in Todorova et al. (2014) and Ventura and Todarova (2015) using $\tau = 8$ (128 ms), the optimal lag for spike counts in a different bin size than we eventually used. Using $\tau = 5$ in these two papers would not change the results. In this letter, using $\tau = 8$ instead of $\tau = 5$ degrades the performances of the basic models relative to the optimized models but does not affect the latter, since their lags are chosen to minimize the prediction risk. It would be a good idea to also let the model selection procedure choose the spike count bin size to lower the risk.

independent of the neuronal covariates (options 1 and 3 at the end of section 2.3) actually underperformed by 53% in median (not shown).

To evaluate the robustness of these results to the number and quality of available electrode recordings, we compared basic KF to minimum-risk RR in 100 random subsets of n neural units, capping n at 40 to ensure sufficient variability across subsamples. For each subset of n neural units, we recorded the median relative efficiency of KF versus minimum-risk RR for the 1040 test trials and calculated the average and standard deviation (SD) of these medians over the 100 subsamples. Figure 1C shows the average median efficiency $\pm$ SD as a function of n: minimum-risk RR is superior to KF in median when $n \geq 6$, it is about 15% and 25% more efficient when n is 20 and 40, respectively, and it outperforms KF on all test trials when $n \geq 20$ (not shown). As n increases, the information provided by the spike counts outweighs the prior kinematic information (see equation 2.3) and KF loses some edge against the non-Bayesian RR decoder.

3.1 Model Selection for OLE and Kalman Filtering. Minimum-risk RR outperforms minimum-risk autoregressive RR, basic OLE, and basic KF decoders for the data analyzed here. We now investigate how better OLE and KF models within the space defined by equation 2.9 compare to minimum-risk RR. Exhaustively evaluating the prediction risks of the 65^{71} models in that space to identify the minimum-risk models would take roughly 10^{75} years per training set on a 2 GHz Inter Core i7 processor. We are not aware of computationally efficient searches in this framework, so we perform greedy searches to identify low-risk models.

3.1.1 Low-Prediction Risk Model Search. We first build an optimal model with a single observation equation per unit, then we consider adding other observation equations. Specifically, we start with the basic OLE and KF models, equations 2.2 and 2.4 with $\mathbf{c}_t = \mathbf{s}_{t-\tau}$ and $\tau = 5$, and optimize the lags in a procedure similar to the "nonuniform lag" selection of Wu et al. (2006): for each unit in turn, we determine which of the 13 lagged equations among equation 2.9 with $g(s) = s$ (untransformed spike counts) reduces the prediction risk the most and replace the current with the optimal equation. (We evaluate 71 units $\times$ 13 models rather than the full combinatorial 13^{71} models.) We repeat this process thrice to allow the lags to stabilize. Next, we hold the lags fixed and apply a similar procedure to search for the best response transformations among equation 2.9: about half the units retain untransformed spike counts, and a quarter each select the square root and ACE transforms. The resulting OLE and KF models are 4% and 11% more efficient in median than the basic OLE and KF models, respectively (see Figures 2Aa and 2Ba, first box plots).

Decoding models typically include at most one observation equation per unit; here we allow several because together, they may contribute more information about the kinematics, compensate for lack of fit of the observation

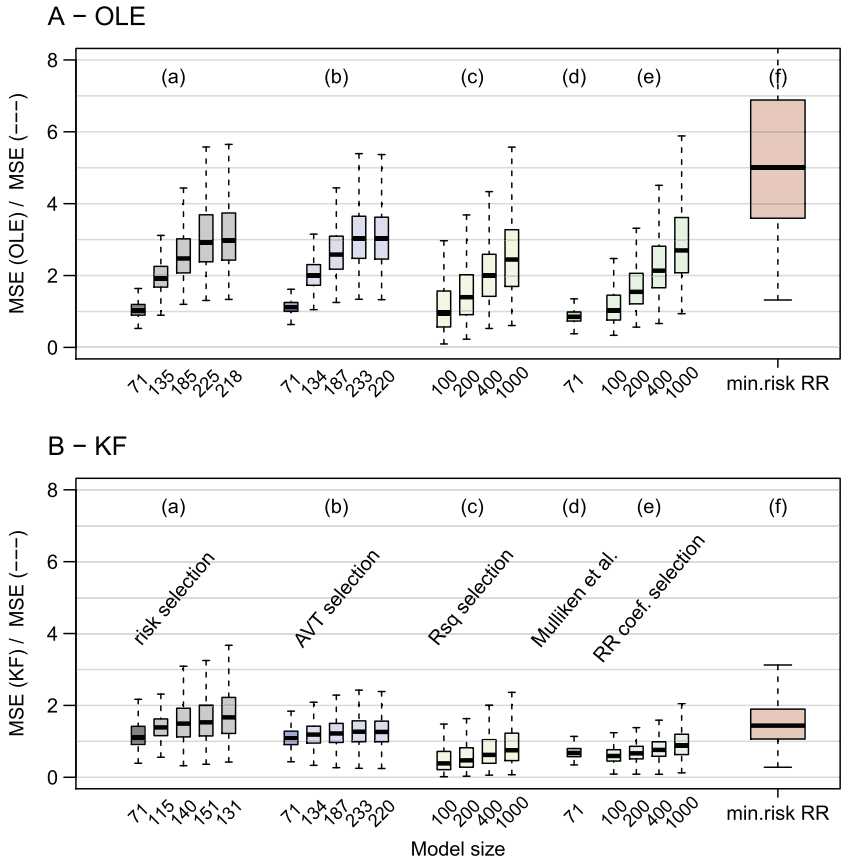

Figure 2: (A) Efficiencies of various OLE models relative to basic OLE, and (B) of various KF models relative to basic KF. (A) Box plots of the distributions of MSE ratios across the 1040 test reaches when comparing basic OLE to OLE decoders optimized using methods a to e, described below. A ratio of 2, say, means that the optimized decoder is twice as efficient as basic OLE. (f) Efficiency of basic OLE relative to minimum-risk RR. (B) Same for KF models. The x-axes record the number of observation equations in the decoding models after successive model search sweeps using the following criteria: (a) stepwise search using low-risk and (b) added variable test (AVT) criteria; noniterative searches: (c) model composed of the m observation equations with largest R^2; (d) one observation equation per unit, corresponding to the absolute largest coefficient in the minimum-risk RR model (Mulliken et al., 2008); (e) m observation equations corresponding to the m absolute largest coefficients of the minimum-risk RR model.

and state equations by balancing bias and variance (see section 2.4), and smooth the predictions. We start with the previous models, and for each unit in turn, we add the equation for that unit that reduces the prediction risk the most, if one exists. The resulting OLE and KF models include 135 and 115 equations on average over the 20 training data sets and are 90% and 35% more efficient in the testing data sets than the basic OLE and KF models (see Figures 2Aa and 2Ba, second box plots). We perform a second and third equation selection sweep across all units (see Figures 2Aa and 2Ba, third and fourth box plots) and prune the resulting models, removing sequentially the equation that reduces the risk the most, until the reduction is negligible. The final OLE and KF models contain 218 and 131 equations—that is, an average of about three and two equations per electrode—and are, respectively, 190% and 67% more efficient in median than the basic OLE and KF models (Figures 2Aa and 2Ba, fifth box plots). The final KF model is also 19% more efficient than the minimum-risk RR model (compare Figures 2Bf and 2Ba, fifth box plot), although it took 23 hours to fit the former versus 3.6 hours for the latter, using a 2 GHz Inter Core i7 processor.

3.1.2 Searches That Do Not Explicitly Lower the Prediction Risk. Risk-reducing searches are computationally demanding, so we investigate alternatives. The added variable test (AVT) search (Ventura & Todorova, 2015) starts with an initial model—two options are described below—and for each unit in turn, augments the model with the equation for that unit that has the most variability explained by the kinematics, after accounting for the spike counts that are already in the model. For example, if the model contains the observation equations for $\mathbf{s}_{t-\tau}$ (see equation 2.1) and we consider adding another equation for unit 1, we regress each transformed lagged spike count of unit 1 on $\mathbf{s}_{t-\tau}$ and record how much these regressions are improved by adding the kinematics as regressors. If the largest improvement is statistically significant at the Bonferroni-corrected level $0.05/M$, where M is the number of tests performed, we add the corresponding equation to the decoding model.

We considered two computationally simple initial models composed of one observation equation per unit: the truncated input space model of Mulliken et al. (2008) includes the equation corresponding to the predictor with the largest absolute RR ridge coefficient (see Figures 2Ad and 2Bd) and the unit-specific best R^2 model includes the equation with the largest R^2; we used the latter because it proved more efficient. Figures 2Ab and 2Bb show the efficiencies of this initial model before and after one, two, and three sweeps across the neural units of the AVT procedure and after pruning the last model: for our data set, the AVT and low-risk selection procedures are comparable when building OLE models, but the former is less efficient in the KF framework.

We considered other fast model selection options: the largest R^2 model contains the m observation equations with the largest R^2 among all

equations in equation 2.9, and the largest RR ridge coefficients model includes the equations corresponding to the predictors with the m largest coefficients, in absolute value, of the minimum-risk RR model; these models do not necessarily include equations for each unit. Figures 2Ace and 2Bce show the efficiencies of these models relative to the basic OLE and KF models for several values of m: larger models have lower risk but perform comparatively poorly even with $m = 1000$ equations, so we cannot recommend them.

4 Discussion

Building good decoding models is crucial to obtain the most accurate kinematic predictions possible. While many models have been successful in improving decoding performance, the search for optimal models is not done systematically or with the exclusive objective of minimizing the prediction error. In this letter, we focused on this objective. We considered a large model space, larger than typically considered, that we searched to identify low-risk models and showed that we can improve the accuracy of offline reconstructions of arm reaches of a rhesus macaque monkey.

With computationally efficient decoding in mind, we considered models based on linear gaussian observation and state equations, specifically OLE, Kalman filtering (KF), and reverse regression (RR), because they provide closed-form predictions. We also introduced autoregressive RR (AR-RR) models by combining RR with the KF state equation. All four classes of models yield predictions that are additive functions of covariates, which were chosen by cross-validation from a large set of parametric and nonparametric transformations of spike counts at several lags, to minimize or reduce the prediction error. For the data analyzed here, the minimum-risk RR and low-risk KF models outperformed the others and were 44% and 67% more efficient than a standard KF decoder, respectively. Any other quantity thought to encode kinematic information should be added to the set of candidate covariates—for example, spike counts binned at multiple timescales (Kim, Carmena, Nicolelis, & Principe, 2005), counts derived from different spike-sorting schemes (Todorova et al., 2014), and waveform moments of unsorted electrodes (Ventura & Todorova, 2015). For non-Poisson neural units, statistical summaries of spike trains other than spike counts may be modulated by kinematics—for example, the variance of interspike intervals or repeated temporal spike patterns such as triplets or bursts of spikes might encode kinematic information that a rate code cannot capture (Brown et al., 1998; Bansal, Truccolo, Vargas-Irwin, & Donoghue, 2012).

We emphasize that our low-risk OLE and KF models can include several observation equations per neural unit, which deviates from the standard practice of using at most one. Multiple equations may contribute more information about the kinematics, for example, if neural units encode kinematics across several lags; they also certainly compensate for the likely lack

of fit of the observation and state equations, and they smooth the predictions. Ventura and Todorova (2015) also used multiequation per unit models to include waveform moment equations in addition to spike counts equations from unsorted electrodes and were thus able to decode kinematics as efficiently as from sorted spike trains, without having to spike sort the electrode signals.

From a computational viewpoint, exhaustive searches through the model space to identify minimum-risk models is prohibitively time-consuming with the large number of candidate covariates we allowed. The Lasso and ridge regression provide faster alternatives in the RR and AR-RR frameworks, but we know of no computationally efficient method to find the minimum-risk OLE and KF models; we are currently developing such methods. Nevertheless, we learned from the unequal performances of the model selection methods we implemented that to build low-risk OLE and KF models, one could start with a single unit-specific lag and transformation observation equation per neural unit, chosen to reduce the risk the most (optimal but tedious) or maximize R^2 (suboptimal but fast); and include additional equations if they reduce the prediction risk (optimal but tedious) or explain significant kinematic variability not already explained by the current model (AVT procedure, suboptimal but fast). The decoding model could be alternatively grown and pruned to control its size when the number of equations must be limited due to computational constraints (Bansal et al., 2012).

In addition to seeking optimal models within the four model classes we considered, we also compared them across classes. We argued that minimum-risk RR is always more efficient than all other RR and OLE models, but because KF predictions are dynamic, their risk cannot be compared to the risk of nondynamic predictions; KF predictions may thus be more, or less, efficient than minimum-risk RR predictions depending on the applications. Minimum-risk RR also outperformed AR-RR predictions, likely because of the many lagged spike counts we included in the sample space. Using a smaller sample space would give KF predictions, and perhaps AR-RR predictions, an edge over minimum-risk RR. These comparisons are theoretically motivated and therefore apply generally to data collected online, offline, and in closed loop, although the benefit of the better methods may vary across experiments and data. Indeed, Koyama et al. (2010) contrasted the performance of various models in the offline and closed-loop contexts and found that improvements could be marginal. It is also possible that other classes of decoders, for example nonlinear, nongaussian, or other decoders, could outperform those we considered here; for example, Li et al. (2009) showed that an unscented KF outperformed the basic KF in both offline and closed-loop decoding, although these results are likely dependent on the number of units, task type, single units versus multiunit, and how the models were optimized. Whichever decoders are considered, the same conceptual approach should apply: we should specify a large space of

models and search through that space to retain the model with the lowest prediction risk (Bansal et al., 2012), provided methods exist or can be developed to do so efficiently. Model searching could also be extended to the state equation in equation 2.3; for example, Wagenaar et al. (2011) reduced the prediction error two-fold to decode the joint angles of a cat's limb by modeling their evolution within a lower-dimensional manifold rather than using the more common random walk or AR(1) evolution model we used in this letter.

Finally, we offer a word of caution: low-risk models are designed to decode optimally data that are similar to the training data. Hence, what an optimized model gains in efficiency could be lost to lack of robustness to changes in behavioral tasks. It is also known that neurons can adapt to a decoder in closed loop (Koyama et al., 2010). It would therefore be useful to determine which classes of decoders are more robust to task changes and which facilitate neuroplasticity.

Appendix A: Graphs of the Four Model Classes

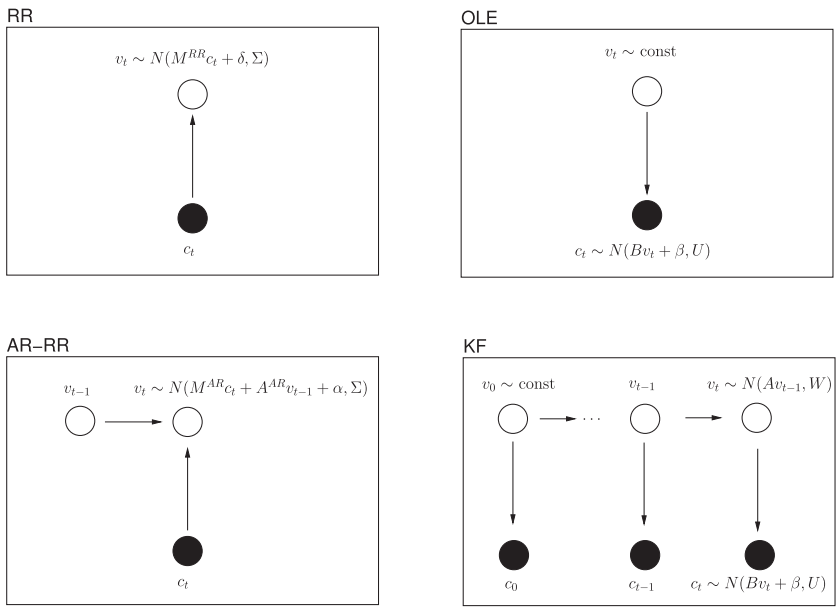

Figure 3: The four classes of decoders introduced in section 2.1, represented as graphs.

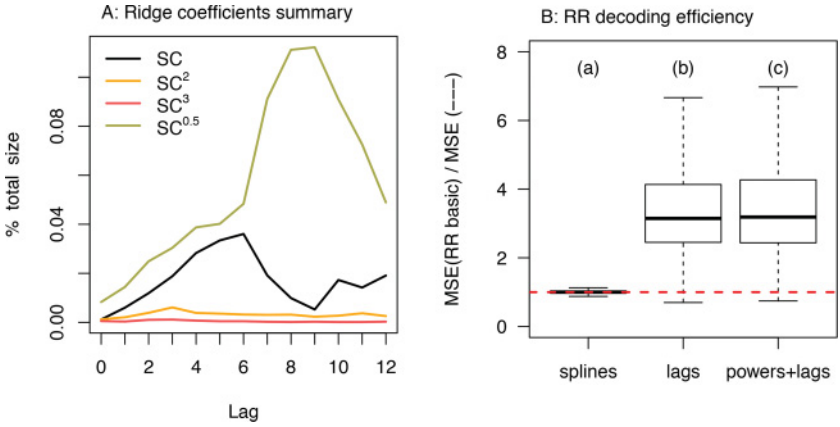

Figure 4: Predictors contributing to minimum-risk RR decoding models. (A) The relative magnitude of the absolute regression coefficients of $\mathbf{s}_{t-i}^{p}$ (denoted by SC^{p}) measure the overall contributions of each transformed spike count at each lag i to the minimum-risk RR model in the model space equation 2.5 with $\mathbf{C}_t = (\mathbf{s}_{t-i}, \mathbf{s}_{t-i}^2, \mathbf{s}_{t-i}^3, \sqrt{\mathbf{s}_{t-i}}, i = 0, 1, \ldots, 12)$ (B) Efficiencies of three minimum-risk RR models relative to basic RR. The model spaces are equation 2.5 with (a) $\mathbf{C}_t = \text{splines}(\mathbf{s}_{t-5})$; (b) $\mathbf{C}_t = (\mathbf{s}_{t-i}, i = 0, 1, \ldots, 12)$; and (c) $\mathbf{C}_t = (\mathbf{s}_{t-i}, \mathbf{s}_{t-i}^2, \mathbf{s}_{t-i}^3, \sqrt{\mathbf{s}_{t-i}}, i = 0, 1, \ldots, 12)$.

Appendix B: Postmortem on the Minimum-Risk RR Model

Here, we investigate the contributions to the minimum-risk RR model of the transformed and lagged spike counts. First, Figure 4Ba shows the distribution over the 1040 test trials of the MSE ratios of basic RR (see equation 2.5 with $\mathbf{c}_t = \mathbf{s}_{t-5}$) and minimum-risk RR in a small model space composed of nonparametric splines of $\mathbf{c}_t = \mathbf{s}_{t-5}$; no other lags or transformations are included. The latter model does not improve on basic RR, which suggests that the relationship between velocity and spike counts is close to linear in our data, at least at lag τ. Next, Figure 4Bb compares basic RR to minimum-risk RR in the model space that includes all lags but no power transformations, that is, $\mathbf{C}_t = (\mathbf{s}_{t-i}, i = 0, 1, \ldots, 12)$, and Figure 4Bc compares basic RR to minimum-risk RR within the space $\mathbf{C}_t = (\mathbf{s}_{t-i}, \mathbf{s}_{t-i}^{\frac{1}{2}}, \mathbf{s}_{t-i}^2, \mathbf{s}_{t-i}^3, i = 0, \ldots, 12)$. The two minimum-risk models are almost equivalent, which suggests that the lagged spike counts have more predictive power than the transformed spike counts.

We now look more closely at the minimum-risk RR model within the model space $\mathbf{C}_t = (\mathbf{s}_{t-i}, \mathbf{s}_{t-i}^{\frac{1}{2}}, \mathbf{s}_{t-i}^2, \mathbf{s}_{t-i}^3, i = 0, \ldots, 12)$. Models selected with the objective of minimizing the prediction risk are not interpretable: regression coefficients no longer quantify the effects of predictors on the response

variable after accounting for all other predictors. Large coefficients may nevertheless suggest that the corresponding predictors have large effects on the response. To measure the overall influence of particular predictors in the minimum-risk RR model, we summed the coefficients' absolute values across units for each combination of power p and lag i, standardized them by the overall sum, and plotted them in Figure 4A (the 4×13 plotted points sum to one). The spike counts and their square roots contribute most to the fit compared to the other power transformations. Older lags contribute substantially, most likely because they help smooth the predictions for the type of movement in our data set, that is, short and straight reaches. The optimal model makeup is data dependent and would likely be different given a different task (e.g., circular movements). An advantage of using a predictive approach is that we do not have to carefully screen which predictors matter most; they are likely to be different in different data sets. We can consider all predictors that encode the kinematics and let the data determine which contribute to lowering the prediction risk.

Acknowledgments

We thank Professor Andrew Schwartz, Professor Steven Chase, and the MotorLab for providing experimental data for this analysis. Support was provided by NIH grant 2R01MH064537 (V.V.).

References

Bansal, A. K., Truccolo, W., Vargas-Irwin, C. E., & Donoghue, J. P. (2012). Decoding 3d reach and grasp from hybrid signals in motor and premotor cortices: Spikes, multiunit activity, and local field potentials. *Journal of Neurophysiology*, *107*(5), 1337–1355.

Breiman, L., & Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, *80*(391), 580–598.

Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C., & Wilson, M. A. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J. Neurosci.*, *18*(18), 7411–7425.

Fraser, G. W., & Schwartz, A. B. (2012). Recording from the same neurons chronically in motor cortex. *J. Neurophysiology*, *107*(7), 1970–1978.

Kahn, K., Sheiber, M., Thakor, N., & Sarma, S. V. (2011). Neuron selection for decoding dexterous finger movements. In *Proceedings of the IEEE Conference in Medicine and Biology Society* (Vol. 33, pp. 4605–4608).

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

Keat, J., Reinagel, P., Reid, R. C., & Meister, M. (2001). Predicting every spike: A model for the responses of visual neurons. *Neuron*, *30*(3), 803–817.

Kim, S.-P., Carmena, J. M., Nicolelis, M. A., & Principe, J. C. (2005). Multiresolution representations and data mining of neural spikes for brain-machine interfaces. In *Conference Proceedings of the 2nd International IEEE Conference on Neural Engineering* (pp. 221–224). Piscataway, NJ: IEEE.

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

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

Liao, Y., Wang, Y., Zheng, X., & Principe, J. C. (2012). Mutual information analysis on non-stationary neuron importance for brain machine interfaces. In *Proceedings of the IEEE Conference in Medicine and Biology Society* (pp. 2012–2748). Piscataway, NJ: IEEE.

Moran, D. W., & Schwartz, A. B. (1999). Motor cortical representation of speed and direction during reaching. *J. Neurophysiology*, *82*, 2676.

Mulliken, G. H., Musallam, S., & Andersen, R. A. (2008). Decoding trajectories from posterior parietal cortex ensembles. *J. Neurosci.*, *28*(48), 12913–12926.

Nishimoto, S., & Gallant, J. L. (2011). A three-dimensional spatiotemporal receptive field model explains responses of area MT neurons to naturalistic movies. *Journal of Neuroscience*, *31*(41), 14551–14564.

Perel, S., Sadtler, P. T., Godlove, J. M., Ryu, S. I., Wang, W., Batista, A. P., & Chase, S. M. (2013). Direction and speed tuning of motor-cortex multi-unit activity and local field potentials during reaching movements. In *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 299–302). Piscataway, NJ: IEEE.

Pillow, J. W., Paninski, L., Uzzell, V. J., Simoncelli, E. P., & Chichilnisky, E. (2005). Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, *25*(47), 11003–11013.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1997). *Spikes: Exploring the neural code*. Cambridge, MA: MIT Press.

Salinas, E., & Abbott, L. F. (1994). Vector reconstruction from firing rates. *J. Comp. Neuroci.*, *1*(1), 89–104.

Stanley, G. B. (2013). Reading and writing the neural code. *Nature Neuroscience*, *16*(3), 259–263.

Tankus, A., Fried, I., & Shoham, S. (2012). Sparse decoding of multiple spike trains for brain-machine interfaces. *J. Neural Eng.*, *9*(5), 054001.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*, 267–288.

Todorova, S., Sadtler, P., Batista, A., Chase, S., & Ventura, V. (2014). To sort or not to sort: The impact of spike-sorting on neural decoding performance. *J. Neural Eng.*, *11*(5), 056005.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *J. Neurophysiology*, *93*(2), 1074–1089.

Vargas-Irwin, C. E., Shakhnarovich, G., Yadollahpour, P., Mislow, J. M. K., Black, M. J., & Donoghue, J. P. (2010). Decoding complete reach and grasp actions from local primary motor cortex populations. *J. Neurosci.*, *30*(29), 9659–9669.

Ventura, V., & Todorova, S. (2015). Decoding motor control signal directly from spike waveforms. *Neural Computation*, *27*, 1033–1050.

Wagenaar, J., Ventura, V., & Weber, D. (2009). Improved decoding of limb-state feedback from natural sensors. In *Proceedings of the IEEE Conference in Medicine and Biology Society* (pp. 4206–4209). Piscataway, NJ: IEEE.

Wagenaar, J., Ventura, V., & Weber, D. (2011). State-space decoding of primary afferent neuron firing rates. *J. Neural Eng.*, *8*(1), 016002.

Warland, D. K., Reinagel, P., & Meister, M. (1997). Decoding visual information from a population of retinal ganglion cells. *J. Neurophysiology*, *78*(5), 2336–2350.

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.

Received July 14, 2016; accepted July 11, 2017.