Parameter estimation for maximizing controllability of linear brain-machine interfaces

Suraj Gowda, Amy L. Orsborn, Jose M. Carmena, Senior Member, IEEE

Abstract—Brain-machine interfaces (BMIs) must be carefully designed for closed-loop control to ensure the best possible performance. The Kalman filter (KF) is a recursive linear BMI algorithm which has been shown to smooth cursor kinematics and improve control over non-recursive linear methods. However, recursive estimators are not without their drawbacks. Here we show that recursive decoders can decrease BMI controllability by coupling kinematic variables that the subject might expect to be unrelated. For instance, a 2D neural cursor where velocity is controlled using a KF can increase the difficulty of straight reaches by linking horizontal and vertical velocity estimates. These effects resemble force fields in arm control. Analysis of experimental data from one non-human primate controlling a position/velocity KF cursor in closed-loop shows that the presence of these force-field effects correlated with decreased performance. We designed a modified KF parameter estimation algorithm to eliminate these effects. Cursor controllability improved significantly when our modifications were used in a closed-loop BMI simulator. Thus, designing highly controllable BMIs requires parameter estimation techniques that carefully craft relationships between decoded variables.

I. INTRODUCTION

Brain-machine interfaces (BMIs) drive artificial actuators using volitional neural activity and have the potential to restore motor function for patients suffering from spinal cord injury or other neurological disorders. Experimental demonstrations in rodents, monkeys and humans have provided a proof of concept, but marked performance improvements are needed before BMIs are clinically viable. Recent work suggests that designing BMIs as closed-loop systems, where neural activity and the decoding algorithm both contribute to performance, may be critical to improving performance [1], [2]. In this view, it is critical to design a BMI decoder that is easy for the subject to learn and control. Here, we study the properties of the Kalman filter (KF), a commonly used decoder in BMI, to understand how its properties may influence a BMI’s controllability.

The standard KF does not model feedback control, so it cannot guarantee that it optimally matches the user’s feedback control strategy. In particular, the standard KF makes BMI “state” variables fully correlated. (Sec. II-D).

For instance, in a 2D cursor control task where velocity is decoded with a standard KF, cursor velocities in perpendicular directions become correlated. In closed-loop operation of the cursor, this means cursor velocity in one direction can alter velocity in the other direction (Sec. III-A). This effect, and other similar effects, resemble force fields (FFs) in arm control and are unlikely to match the BMI user’s control strategy. The unpredictability of these disturbances increases the difficulty of both open-loop and closed-loop cursor control.

The presence of these undesirable effects in the KF explains the importance of selecting decoder variables. For instance, some (but not all) FF effects are eliminated when the KF decodes velocity-only (VOKF) instead of position/velocity (PVKF). This explains experimental results in [1], [3] where the VOKF outperforms the PVKF in closed-loop control (CLC). We utilize a control-theoretic design perspective to identify and eliminate FFs to maximize BMI controllability.

In an experiment where a non-human primate controlled a PVKF-based BMI in closed-loop, FFs generated by the decoder changed unpredictably when the KF parameters were re-estimated. The unpredictable FF changes seen in experiments indicated that the emergence of FFs was likely to be a decoder training artifact. Analysis of 87 sessions/decoders showed that the emergence of FFs had a significant correlation with CLC performance, suggesting that they indeed reduced BMI controllability. We present a modified PVKF that overcomes the training limitations of the standard PVKF and outperforms the VOKF in an online prosthesis simulator (OPS) which simulates closed-loop cursor control.
II. METHODS

A. Electrophysiology

One adult male rhesus macaque (macaca mulatta) was used in this study. The subject was chronically implanted with microwire electrode arrays for neural recording. One array of 128 teflon-coated tungsten electrodes (35µm diameter, 500µm wire spacing, 8 x 16 array configuration; Innovative Neurophysiology, Durham, NC) was implanted in each brain hemisphere. Arrays were implanted targeting the arm areas of primary motor cortex (M1) and dorsal premotor cortex (PMd). Single and multi-unit activity was recorded using a 128-channel MAP system and sorted online using Sort Client (Plexon, Inc., Dallas, TX). Only neural units with well-identified waveforms were used for BMI control. All procedures were conducted in compliance with the National Institute of Health Guide for Care and Use of Laboratory Animals and were approved by the University of California, Berkeley Institutional Animal Care and Use Committee.

B. Task

The subject was trained to perform a self-paced delayed 2D center-out reaching task to 8 targets (1.7cm radius) uniformly spaced about 14cm diameter circle. After being trained to perform the task with arm movements, the subject controlled the cursor using a KF BMI (Sec. II-D) in closed-loop without overt arm movements. Fig. 1 shows an illustration of the task setup and trial timeline. Trials were initiated by moving the cursor to the center target and holding for 400ms. The subject had an unlimited amount of time to enter the center target to initiate a trial. Upon entering the center, the reach target appeared. After the center-hold period ended, the subject was cued to initiate the reach (via target flash), after which he was required to move the cursor to the peripheral target within a given 3s time-limit and hold for 400ms to receive a liquid reward. Failure to hold at the center or target, or reach the target within the time-limit, restarted the trial without reward. Targets were block-randomized to ensure the same number of reaches to each target in approximately random order.

C. Performance measures

BMI performance was assessed using four metrics:

1) Hold error rate: The occurrence rate of hold errors (at the peripheral target) on trials where the cursor entered the target within 3s.
2) Time-to-target: The time elapsed between leaving the center and entering the target.
3) Length of reach: The distance traveled between leaving the center and entering the target.
4) Movement Error: the average deviation perpendicular to the reach direction [4].

Upon successfully initiating a reach (via center-hold), there were two possible task-errors: failure to reach the peripheral target in time, and failure to hold at the peripheral target. Thus, metrics 1 and 2 adequately assessed task performance. Metrics 3 and 4 quantified the precision and accuracy of control. Because the task was self-paced, the number of initiated trials varied significantly across decoders, partly reflecting subject motivation. To alleviate motivation biases in cross-decoder performance comparisons, performance metrics were weighted by the number of initiated trials.

D. The KF in closed-loop BMI

In the PVKF, the $x_t$ represents the cursor kinematics:

$$x(t) = \begin{bmatrix} p_x(t) & p_y(t) & v_x(t) & v_y(t) & 1 \end{bmatrix}^T$$

Hereafter, we omit the last term which represents constant offsets. The KF models $x(t)$ as a Gaussian process:

$$x(t+1) = Ax(t) + w(t), w(t) \sim N(\bar{x}, W)$$

Neural firing rate observations (100ms binned spike counts) $u_t$ are modeled as correlated Gaussians where the mean depends linearly on the state $x_t$.

$$u(t) = Hx(t) + q(t), q(t) \sim N(\bar{u}, Q)$$

In our experiment, $A$ and $W$ were biomimetic models of arm movements and were fixed for all sessions. $H$ and $Q$ were estimated using closed-loop decoder adaptation (CLDA) to optimize CLC performance [2]. Spike observations are used to recursively estimate intended cursor kinematics:

$$\dot{x}(t) = \underbrace{(I_n - K_t H) A \dot{x}(t-1) + K_t u(t)}_{\hat{A}}$$

$$K_t = P_{t|t-1} H (HP_{t|t-1} H^T + Q)^{-1} \underbrace{B}_{\hat{B}}$$

(1)

(2)

where $n$ is the dimension of $x_t$, $I_n$ is the $n$-dimensional identity matrix, $K_t$ is the time-varying Kalman gain and $P_{t|t-1} = \text{cov}(x_t|u_1, \ldots, u_{t-1})$. We defer the full derivation to [5]. In the VOKF, $x_t = [v_x(t), v_y(t), 1]^T$ contains only the velocity components. Position estimates are generated by “integrating” the decoded velocity: $\tilde{p}_t = k\tilde{v}_t + \tilde{p}_{t-1}$, where $k$ is the rate of the spike count observations (100ms).

Every linear BMI algorithm can be written in the form of Eq. 1, a time-varying linear dynamical system where $\hat{A}$ is the state transition matrix. $\hat{A}$ has the same meaning for all linear BMI algorithms with the same state $x_t$, enabling comparisons between different linear BMI algorithms. Though $\hat{A}$ and $\hat{B}$ are technically time-varying in the KF, our experimental KFs converged in less than 1 minute. We analyzed only the time-invariant form of the KF, the steady-state KF, because convergence time took a small fraction of experiment time. Thus, design principles presented here are generally applicable to linear BMIs.

E. Simulating closed-loop BMI

We simulated CLC of different decoders in an OPS similar to [6]. At each time, a position was decoded and subsequently used by the simulated subject to calculate an intended cursor
velocity. The intended velocity was always exactly in the
direction of the task target, consistent with our assumptions
about subject behavior when training decoders in closed-
loop [2]. The firing rates of 25 simulated neurons were
Poisson distributed with rate dependent on the neuron’s
preferred direction and the intended movement direction. All
simulations used the same preferred directions. Decoders
were trained using CLDA [2], without prior knowledge of
neural preferred directions. The simulated subject did not
plan movements that compensated for FFs described in Sec.
III-A.

III. RESULTS
A. Recursive decoders can generate “force fields”

An arbitrary $\bar{A}$ can cause links between kinematic variables
that would not exist in normal arm control of a cursor. For
2D linear BMIs:

\[
\begin{bmatrix}
    p(t+1) \\
    v(t+1)
\end{bmatrix} =
\begin{bmatrix}
    T p(t) + K v(t) \\
    M p(t) + N v(t)
\end{bmatrix} + B u_t
\]

To reference elements of the $2 \times 2$ matrices $T$, $K$, $M$, and
$N$, we use the convention $T = [[t_{xx}, t_{xy}], [t_{xy}, t_{yy}]]$. $T$ and $M$ represent position dependent changes to the
cursor state. $K$ represents “integration” of $v_t$ to move the cursor’s
position. When $0 < \text{eigenvalues}(N) < 1$, the cursor
smoothly decelerates, modeling momentum.

Decoder-dependent distortions can induce kinematic ef-
fects like FFs in arm control, including shaking, pushing, or
curling effects. To illustrate, we show how using a decoder in
open-loop distorts a planned trajectory (Figure 2A). To plan
this trajectory, our simulated subject assumes that the state
transition matrix of the cursor is of the form

\[
\bar{A}_{\text{internal model}} =
\begin{bmatrix}
    I_2 & kI_2 \\
    0 & nI_2
\end{bmatrix}
\]

This is a reasonable internal model—the cursor moves only
if input is applied to make the velocity nonzero, and hori-
zontal/vertical kinematics can be controlled independently. In
the following examples, we generate different types of FFs
by changing at most 2 parameters of Eq. 4.

The cursor becomes intrinsically jumpy if $T \neq I_2$. An
arm control analog is a position-dependent “tremor” FF. In
Fig. 2B, we set $t_{xy} = 0.9$, which results in a leftward
shaking effect. When the simulated subject performed the
planned reach in open-loop, not accounting for the mismatch,
significant leftward push was exerted on the cursor when its
velocity was small, during the hold period.

$M \neq 0$ generates position-dependent changes to the cursor
velocity, emulating position-dependent FFs in arm control.

To illustrate, we show how using a decoder in
open-loop distorts a planned trajectory (Figure 2A). To plan
this trajectory, our simulated subject assumes that the state
transition matrix of the cursor is of the form

\[
\bar{A}_{\text{internal model}} =
\begin{bmatrix}
    I_2 & kI_2 \\
    0 & nI_2
\end{bmatrix}
\]

This is a reasonable internal model—the cursor moves only
if input is applied to make the velocity nonzero, and hori-
zontal/vertical kinematics can be controlled independently. In
the following examples, we generate different types of FFs
by changing at most 2 parameters of Eq. 4.

The cursor becomes intrinsically jumpy if $T \neq I_2$. An
arm control analog is a position-dependent “tremor” FF. In
Fig. 2B, we set $t_{xy} = 0.9$, which results in a leftward
shaking effect. When the simulated subject performed the
planned reach in open-loop, not accounting for the mismatch,
significant leftward push was exerted on the cursor when its
velocity was small, during the hold period.

$M \neq 0$ generates position-dependent changes to the cursor
velocity, emulating position-dependent FFs in arm control.

An example is shown in Fig. 2C, where $v_{yy} = 0.03$ pushed
the cursor upward with non-uniform strength dependent on
distance from the origin. The effect was again most signifi-
cant during the target hold, when the intended velocity was
small. In Sec. III-B2, we analyzed the impact of the hold
FF, the position-dependent velocity change at the center of
peripheral targets, since holds imply small intended speed.

Nonzero $k_{xy}$, $k_{yy}$, $n_{xy}$, or $n_{yx}$ cause links between hori-
zontal and vertical kinematics, emulating velocity-dependent
curl FF in arm control. In Fig. 2D, the decoder created
clockwise curl with $k_{xy} > 0$ and $n_{yx} > 0$, causing the
open-loop trajectory to overshoot horizontally. $K$-induced
curl causes direct changes to the position while $N$-induced
curl causes changes to the velocity, which is not directly
observable. Unlike the previous two FF types, this effect is
more pronounced when cursor speed is large. We analyze the
impact of decoder curl on performance in Sec. III-B3.

B. Correlations between decoder parameters and per-
formance

We found significant correlations between CLC perfor-
mance and the emergence of FFs (Sec. III-A). The hold error
rate did not improve significantly over the 2-month course of

\[
\begin{align*}
|t_{xx} - 1|, |t_{yy} - 1|, |t_{xy}|, |r_{yx}|\end{align*}
\]

Significant correlations between decoder parameters and
hold error rate.

<table>
<thead>
<tr>
<th>Decoder Parameters</th>
<th>Hold Error Rate Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>t_{xx} - 1</td>
</tr>
<tr>
<td>$|B[0 : 2, :]|_2$</td>
<td>$&lt; 10^{-4}$, $&lt; 10^{-5}$, $&lt; 10^{-3}$</td>
</tr>
<tr>
<td>$|M|_2$</td>
<td>$0.35$, $0.35$, $0.27$, $0.26$</td>
</tr>
<tr>
<td>$\Delta n = n_{xx} - n_{yy}$, $\Delta k = k_{xx} - k_{yy}$</td>
<td>$0.27$, $0.26$</td>
</tr>
</tbody>
</table>

We found significant correlations between CLC perfor-
mance and the emergence of FFs (Sec. III-A). The hold error
rate did not improve significantly over the 2-month course of
the experiment ($p > 0.59$). The number of units used in the
decoder was also not significantly correlated with the hold
error rate ($p > 0.59$). To offset learning/fatigue effects, we
analyzed only the first continuous block of use of the decoder
if it was used for at least 100 trials. Decoders that the subject
was less willing to use (fewer trials per second were initiated)
were used in blocks of longer duration to prevent selection
bias ($r = -0.204$, $p < 0.06$). 87 decoders were used in
correlations between decoder parameters and performance.

1) FF emergence correlated with increased hold error rate: Significant correlations between hold error rate and decoder
parameters are summarized in Table I. Hold error rates
improved when $T \rightarrow I_2$, $M \rightarrow 0$, and $\|B[0 : 2, :]\|_2 \rightarrow 0$, implying that position terms should not be included in the
KF state for maximum controllability. Asymmetric horizontal/
vertical velocity dynamics ($\Delta n$, $\Delta k$) also correlated with
increased hold error rate.
2) Holding FFs which pushed the cursor towards the workspace center correlated with shorter reaches: For targets 0, 2, 3, 4, and 5, mean time-to-target decreased when the hold FF (Sec. III-A) angle directed the cursor to the center target ($|r| > 0.22, p < 0.04$). The typical hold FF angle for the remaining targets pointed toward the center of the workspace for unknown reasons.

3) Curl FFs correlated with decreased accuracy: Larger $|k_{xy}|$ and/or $|k_{yz}|$ were correlated with longer mean reach length for targets 0, 1, 2, 3, 4, and 7 ($r > 0.18, p < 0.05$). A less significant effect was found for the remaining targets ($r > 0.17, p < 0.1$). Increasing $|n_{xy}|$ and/or $|n_{yz}|$ was correlated with increased hold error rate for targets 0, 4, 5, 6, and 7 ($r > 0.24, p = 0.021$). The equivalent relationship for the remaining targets was insignificant.

4) Tradeoff between speed and accuracy: In Fig. 3, we show $n_{xx}$, the horizontal “momentum”, versus the hold error rate and mean time-to-target for all decoders. Both correlations are statistically significant: as $n_{xx}$ increases, hold error rate increased ($|r| > 0.65, p < 10^{-5}$) while the mean time-to-target decreased ($|r| > 0.39, p < 2 	imes 10^{-4}$). Similar relationships exist for $n_{yy}, k_{xx}$ and $k_{xy}$, reflecting a tradeoff between accuracy and speed in linear BMIs.

C. Modifying the PVKF for optimal decoder structure

The data in Sec. III-B suggest that many elements of $\tilde{A}$ in Eq. 3 should be 0 to maximize curror controllability. $M \rightarrow 0$, $T \rightarrow I_2$, and $K \rightarrow kI_2$ correlated with improved hold performance and reaching accuracy. Imposing the restrictions $M = 0$, $T = I_2$, $K = kI_2$ onto the PVKF mathematically eliminates position-dependent FFs explained in Sec. III-A, implying that the presence of the position-dependent FFs reduced curror control. The PVKF under these restrictions becomes nearly equivalent to the VOKF, with the exception that $k$ in the PVKF need not be exactly the spike bin width. Velocity-dependent curl FFs must still be eliminated from both the VOKF and the PVKF. For the VOKF, we expand Eq. 2 with the matrix inversion lemma,

$$K_iH = \left[ P_{t|t-1} - D \left( P_{t|t-1}^{-1} + D \right)^{-1} \right] D, \quad D = H^TQ^{-1}H$$

If $H^TQ^{-1}I = sI_2$ and $A = aI_2$, then $(I - K_iH)A = nI_2$ and the KF will be curl-free. This constraint must be applied to the maximum-likelihood estimation of $H$ and $Q$ [5]. We omit the equivalent PVKF constraint due to limited space.

D. OPS performance of the modified PVKF and VOKF

In the OPS described in Sec. II-E, we found that the PVKF with modifications described in Sec. III-C had a significant 23% less movement error than the VOKF over 10 independent simulation runs (2-sample KS test, $p < 10^{-4}$). Trajectories from one simulation are shown in Fig. 4. In the PVKF, $k$ provides an extra degree of freedom in the plant design. In our experimental data where $k_{xx}$ and $k_{yy}$ are learned using CLDA, values were between 0.05 and 0.075. In the VOKF, $k$ is fixed by the spike bin width (Sec. II-D).

IV. CONCLUSION

Interpreting BMI as a feedback control problem allowed us construct simple design rules to create KF cursors free of force field effects, which should improve BMI controllability in closed loop. This control-theoretic approach explains previous studies of closed-loop cursor control where the VOKF has outperformed the PVKF. The method we employ for understanding feedback control dynamics scales easily to high degree-of-freedom BMIs.

REFERENCES