34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Compensating for Delays in Brain-Machine Interfaces by Decoding Intended Future Movement

Francis R. Willett, Aaron J. Suminski, *Member, IEEE*, Andrew H. Fagg, *Member*, *IEEE*, and Nicholas G. Hatsopoulos

Abstract—**Typically, brain-machine interfaces that enable the control of a prosthetic arm work by decoding a subjects' intended hand position or velocity and using a controller to move the arm accordingly. Researchers taking this approach often choose to decode the subjects' desired arm state in the *present* moment, which causes the prosthetic arm to lag behind the state desired by the user, as the dynamics of the arm (and other control delays) constrain how quickly the controller can change the arm's state. We tested the hypothesis that decoding the subjects' intended *future* movements would mitigate this lag and improve BMI performance. Offline results show that predictions of future movement (≤ 200ms) can be made with essentially the same accuracy as predictions of present movement. Online results from one monkey show that performance increases as a function of the future prediction time lead, reaching optimum performance at a time lead equal to the delay inherent in the controlled system.**

I. Introduction

Brain-machine interfaces (BMI) record neural signals in real time, interpret them as motor commands, and reroute them to a device (e.g., a computer cursor or prosthetic arm) in order to restore the subject's lost motor function. Typically, a BMI that enables the control of a prosthetic arm decodes an intended hand position or velocity from the subject and uses a controller to generate joint torques to drive the arm accordingly. Previous studies taking this approach have chosen to decode the subject's desired arm state in the *present* moment and use it as the command signal [1,2]. This approach, however, causes the prosthetic arm to lag behind the state desired by the user, as the dynamics of the arm usually constrain how quickly the controller can bring the arm's state in accordance with the commanded state. Furthermore, neural activity or the decoder output is sometimes smoothed in order to eliminate high frequency noise in the decoder output, which introduces additional delays in the control loop. Generally, no attempt is made to mitigate these time delays which may lead to degraded performance.

In this study, we tested the hypothesis that predicting the subject's intended *future* movement at a time lead equal to the control delay, and using this as the command signal for our controller, would compensate for delay and improve performance. Furthermore, we expected that compensating for the delay in our BMI control loop would improve reach performance by allowing the subject to make quicker corrections, and by allowing the subject to react more quickly to a new target.

II. Methods

A. Behavioral Task

One adult male rhesus macaque (Macaca mulatta) was trained to control a cursor in a two-dimensional workspace with our BMI that decoded the monkey's intended hand position based on neural activity in primary motor cortex (MI) from the recent past. The monkey sat in a primate chair holding his arm still while it was abducted 90 degrees and supported by the KINARM, a two-link robotic exoskeleton (BKIN Technologies, Kingston, ON). Direct vision of the monkey's arm was precluded by a horizontal projection screen, on which the cursor and the targets were projected during the random target pursuit (RTP) task. This task required the monkey to continuously move a cursor (6 mm diameter) to a square target (2.25cm^2). Each time the monkey hit a target, a new target appeared immediately in a randomly chosen position of one of 9 locations defined by a 3x3 grid within the workspace (14.2 by 13 cm).

Each experimental session consisted of two conditions: visual observation followed by brain control. During visual observation, the monkey held his arm still while he observed movements of the cursor hitting targets. These movements were recorded previously while the monkey performed the task with his own arm. We trained the neural decoders in our BMI by associating recent spiking activity with the position of the cursor during visual observation. During brain control, the monkey used these neural decoders to control a simulated arm (Figure 1), whose hand position was displayed to the monkey as a cursor in the 2D workspace. In order to complete a successful trial and receive a juice reward, the monkey was required to sequentially acquire two to three targets (brain control) or five to nine targets (visual observation). In either condition, if the monkey moved his arm outside of a 2 cm circle in the center of the KINARM's workspace, the screen was shut off and the robotic exoskeleton controlled his arm back to the center of the workspace. A trial was aborted if any movement between

Manuscript received March 15, 2012. This work was supported by NIH NINDS R01 N545853-01 to NGH and AHF.

F. R. Willett and A. J. Suminski are with the Department of Organismal Biology and Anatomy at the University of Chicago, Chicago, IL 60637 USA (fwillett@uchicago.edu and asuminski@uchicago.edu).

A. H. Fagg is with the School of Computer Science at the University of Oklahoma, Norman, OK 73019 USA (fagg@cs.ou.edu).

N. G. Hatsopoulos is with the Department of Organismal Biology and Anatomy and the Committee on Computational Neuroscience at the University of Chicago, Chicago, IL 60637 USA (nicho@uchicago.edu).

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

targets took longer than 5 seconds.

B. *Real-Time BMI*

Our BMI (Figure 1) converts neural activity into elbow and shoulder joint torques in order to drive the movement of a two-link simulated arm. This simulated arm captures the dynamics of both the KINARM and the monkey's arm [3]. X_c represents the position of the hand and determines the location of the cursor on the screen. The torques, T, which drive the arm, are generated by a Proportional-Derivative (PD) controller that moves the simulated arm towards X_F, the position decoder's prediction of the monkey's intended hand position after it has been low-pass filtered. The PD controller causes the simulated arm to lag behind the filtered, decoded position by approximately 100ms.

The position decoder, implemented as a linear, finite impulse response (FIR) filter, predicts the monkey's intended hand position, X_D, from the neural data. In our approach, the neural activity is represented as a series of binned spike counts, and the hand position is reconstructed from a linear combination of these spike counts plus a constant offset representing the mean hand position [3]. We employ a history of $B = 20$ bins (filter taps) of $\Delta t = 50$ ms each for every neuron, giving the filters access to a total of one second of neural spiking history. Specifically, signal k (X or Y hand position in the present, or at a time lead into the future) at discrete time bin t, is reconstructed as follows:

$$S(t,k) = \sum_{j=0}^{B-1}\sum_{i=0}^{C-1} A(k,j,i) \times N(i,t-j) + \bar{k} \ ,$$

where i indexes over the C neurons, j indexes over time bins, $N(i, t)$ is the spike count of neuron i over time bin t, A are the coefficients of the filter, and $\bar{k}$ is the mean value of signal k. The coefficients are solved for analytically with ridge regression [3], using 250 seconds of spiking activity and cursor movement data collected during visual observation.

To obtain a decoder that predicts intended future movement, we solve for the coefficients that best predict the cursor position at some time lead into the future (relative to the one second of spiking history). This requires only a simple alteration of the training data set to pair the neural data with the state of the cursor some at a given time lead, τ, into the future. The decoded position X_D, representing the monkey's intended hand position in the present or in the future, is fed through a low pass Butterworth filter (cutoff frequency = 3.0 Hz), adding an additional delay to the decoded position signal of approximately 100ms.

C. *Electrophysiology*

The monkey was chronically implanted with a 100-electrode microelectrode array (Blackrock Microsystems, Inc., Salt Lake City, UT) in MI contralateral to the arm used for the task. During the recording session, signals from up to 96 electrodes were recorded using a Cerebus acquisition system (Blackrock Microsystems). Single and multiunit spiking events were sorted online and used to train and drive the BMI during the experiments. On average, 68.6 ± 4.7 (mean ± standard deviation) neural channels were sampled during each BMI session. All of the surgical and behavioral procedures were approved by the University of Chicago Institutional Animal Care and Use Committee and conform to the principles outlined in the *Guide for the Care and Use of Laboratory Animals*.

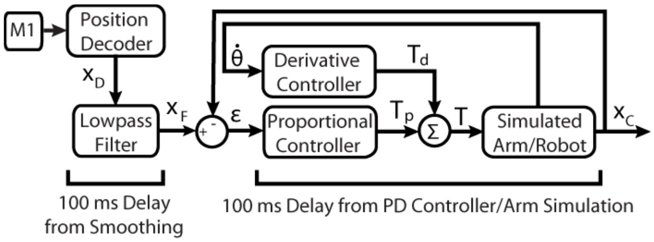

Figure 1: Our BMI uses neural activity recorded from the primary motor cortex to generate an intended hand position in Cartesian space which the monkey then uses to hit targets. X_D, X_F and X_C are two-element column vectors containing the X and Y components of a hand position in Cartesian space, ε is a two-element column vector containing the X and Y components of an error signal, $\dot{\theta}$ is a two-element column vector containing elbow and shoulder joint velocities, and T_p, T_v, and T are two-element column vectors containing joint torque terms. Two sources of delay are illustrated: a low-pass Butterworth filter that smooths the output of the position decoder (X_D), and a PD controller that drives the arm towards X_F.

D. *Offline Decoding*

We used visual observation data for an offline analysis of decoder performance and decoder coefficients. A cross-validation approach was used to examine the performance of the decoder when predicting different time leads into the future. Binning spike counts and cursor positions into 50ms bins, we slid a 5000 bin training set window across the dataset in 1000 bin increments to generate a series of training sets. For each training set, we then tested the decoder's ability to reconstruct the observed cursor position on the remaining bins in the dataset. To measure performance, we computed the fraction of variance accounted for in the cursor position by the decoder.

In addition to decoding performance, we also examined the value of the coefficients for decoders of differing future prediction time leads. Since the decoder makes use of 20, 50ms bins (taps) of spiking history for each neuron, we can examine the relative weights of the coefficients for these 20 taps to assess the time period from which the decoder is extracting the most information. To compare the magnitude of the decoder coefficients, we normalized the coefficients for the X and Y position decoders independently by taking the absolute value of the z-score of the coefficients. We then pooled coefficient observations across decoders, days, and neurons, to yield 20 distributions of coefficients corresponding to each of the 20 taps. Finally, we computed the median normalized coefficient value for each tap and for each future prediction time lead to examine the shape of the decoder.

E. *Online Experimental Procedure*

We varied the future prediction time lead (τ) while we held the delay in the BMI control loop constant at 200ms (i.e. a low pass Butterworth filter which delayed the position signal by 100ms and a PD controller with a lag of 100ms). We expected that performance would be optimal at τ = 200 and would degrade proportionally away from this point. The experiment included multiple decoders, predicting intended movement at time leads of 0ms, 50ms, 100ms, 200ms, 300ms, 400ms, 450ms, 500ms, and 600ms. Approximately 110 reaches were collected for each condition on each day, though not all conditions were tested on all 9 days.

F. *Kinematic Analyses*

We used three kinematic measures to quantify the performance differences between BMI conditions for a given movement: 1) normalized time-to-target, 2) normalized path length, and 3) normalized orthogonal direction changes. The normalized time-to-target metric is defined as the time difference between two consecutive target hits divided by the distance between the targets and has units of s/cm. The normalized path length metric is defined as the path length of the reach trajectory between two consecutive targets divided by the distance between the targets and is a unitless ratio of distance measures. The normalized orthogonal direction changes metric counts the number of times the reach trajectory reverses direction along the axis defined by the line connecting two consecutive targets, and is divided by the distance between the targets.

We further normalized these metrics by taking out the day to day variation in performance in order to isolate differences in performance between conditions across days. For each day, we subtracted a constant *c*, which represented the average level of performance on that day, from the mean performance in each condition. We called these metrics, which represent the difference between the performance in a given condition and the average performance on that day, as Δ Time to Target, Δ Path Length, and Δ ODC (Orthogonal Direction Changes). For a given day, *c* was computed by averaging the mean performances of a subset of conditions that were tested every day.

III. Results

A. *Offline Experiment*

To determine the feasibility of predicting the monkey's intended hand position some time lead into the future, we examined the ability of our decoder to use neural activity to reconstruct observed movements of a cursor (Figure 2A). Performance, measured by the fraction of variance of the cursor movements accounted for by the decoder, remains relatively steady when predicting 0ms to 200ms into the future, but rolls off quickly for τ > 200ms.

We also examined the coefficients of the decoders used to generate the performance results in Figure 2A. By pooling together normalized coefficients from the filters across days, neurons, and decoders, we were able to compute which taps were weighted most heavily for the different future prediction time leads (τ) tested. As τ increased, the coefficients from earlier taps became more heavily weighted (larger coefficient values), suggesting that the decoder uses more information from recent neural activity as it is asked to predict movements farther into the future (Figure 2B).

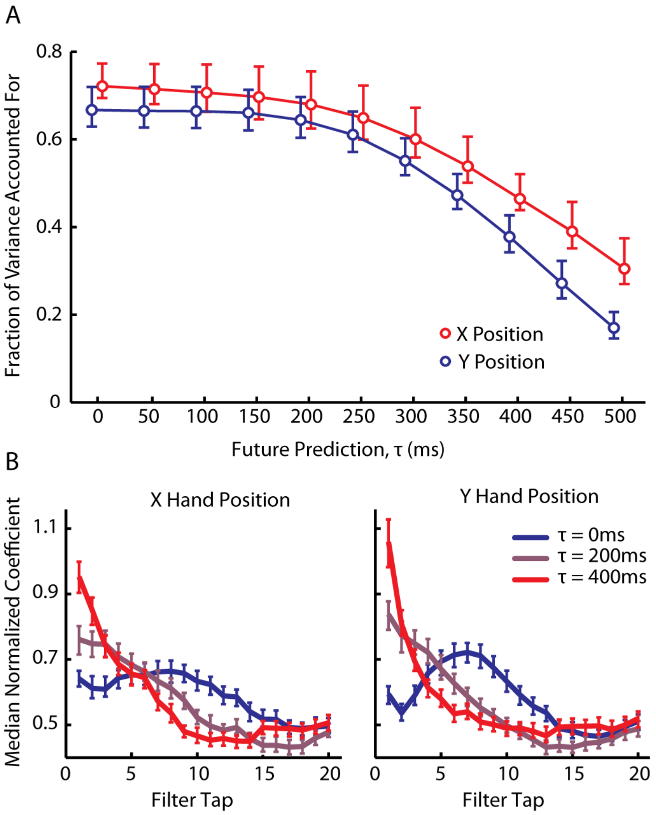

Figure 2: Characteristics of decoders as a function of τ during offline decoding. (A) The median fraction of variance accounted for (FVAF) when using our decoder to reconstruct observed cursor movements at various time leads (τ). For each time lead, median FVAF for X (red) and Y (blue) hand position is shown with a 95% CI about the median. Curves are offset in the x-direction to aid visualization. (B) The relative weight of decoder coefficients when our decoder is trained to predict various time leads into the future (τ, denoted by lines of different color). Error bars denote 95% CI about the median. Filter tap 1 is the coefficient relating the most recent spiking activity to the hand position.

B. *Online Experiment*

In this experiment (Figure 3), we examined how performance varied as we varied the future prediction time lead (τ), while holding the level of delay in the BMI controller constant at 200ms. To examine how performance varied with τ, we compared distributions of the differences between the mean performance for a given condition and the average level of performance on a given day (see Methods: Kinematic Analysis). Mean differences greater than zero indicate above average performance for a given condition on a given day. A Kruskal-Wallis test was first performed for each of the three performance metrics, showing a main effect of τ in all cases (alpha = 0.05). Post hoc comparisons between different conditions were made with a Tukey-Kramer multiple comparison procedure after rank-transforming the values (family wise error rate = 0.05).

We found that the time taken to complete a reach decreased as τ increased to 200, and then flattened for later conditions (significant decrease in time-to-target between

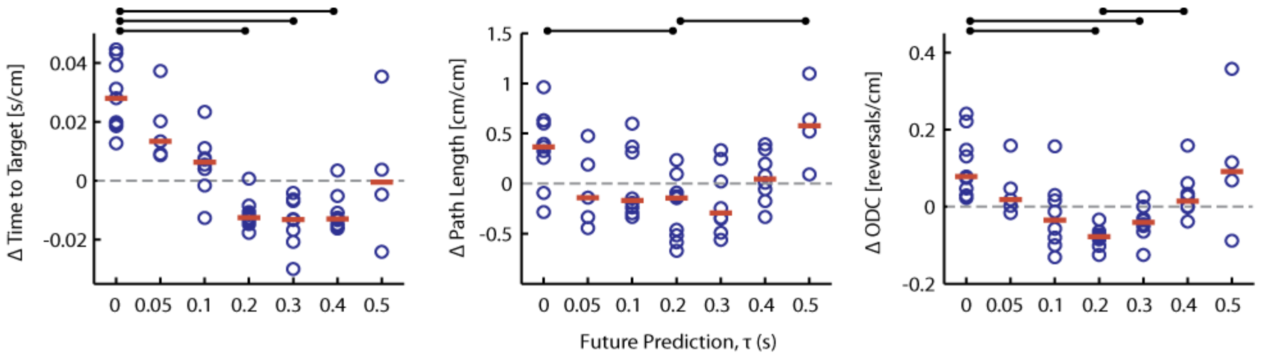

Figure 3: Online performance results from 9 datasets where multiple future prediction time leads (τ) were tested on each day while we held the BMI controller's level of delay constant at 200ms. The differences between the mean performance and the average level of performance on a given day are plotted (blue circles), and the median of these means is plotted for each condition (red bars). Optimal performance is reached around τ=200. The black bars indicate significant differences between conditions.

τ=0, and τ = 200, 300, and 400). The path length and ODC metrics show that reaches became straighter and smoother as τ increased to 200, then became less so as τ increased further (significant decrease in path length and ODC from τ=0 to τ=200, significant increase in path length and ODC from τ = 200 to τ = 500 (path length) and τ = 400 (ODC)).

IV. Discussion

Our offline results demonstrate that future movements can be accurately reconstructed without much loss in fidelity up to a time lead of 200ms, verifying the feasibility of predicting future intended movements. Testing this technique online, we also demonstrated that decoding intended future movements and using them as a command signal for a BMI can significantly improve performance, making reaches faster, straighter, and smoother. These results represent a significant advance in the BMI field, as they show that it is possible to compensate for the negative effects of delay in BMI control loops through a simple solution. Decoders incorporating future prediction may yield performance benefits in all BMI systems that interpret recent neural activity as an intended motion in the present.

It is worth noting that the activity of MI neurons typically leads movement by approximately 120ms or 40ms in intact individuals making or observing movements, respectively [4]. This physiological delay is attributed to the conduction time necessary for cortical activity to reach the periphery and to the muscle activation dynamics that transform action potentials into muscle activity. We believe that our future prediction decoders are able to accurately predict future intent by making use of this neural activity which naturally leads movement.

A limitation of using future prediction to mitigate control delays is that delay is never fully removed from the system. That is, delay in the control loop places an absolute floor on the reaction time, because even a command signal representing future intentions is delayed in execution. In the case of a pure delay (as opposed to a delay from low-pass filtering), the cursor would be unable to move in response to a new movement command for at least the amount of time delay. However, when the command signal is executed, if it represents a future intent, then the cursor will "catch up" to the subject's intention. This will never happen if the decoder predicts present intent. We believe that a command signal representing future intent is therefore the most appropriate in a BMI system with delays.

Other issues remain to be addressed in future work. First, the offline results show that the decoder coefficients corresponding to recent neural activity become more heavily weighted when the decoder is asked to predict future movements. This could have a side effect of making the decoder more responsive to newer movement commands, possibly accounting for some or all of the performance improvements.

Second, our performance metrics fail to disambiguate the effects of future prediction on the start time of the reach and on the quality of the reach itself. We defined a reach as the entire cursor trajectory from the appearance of a new target to the next target hit. Therefore, we include any movement of the cursor made *after* a new target appears, but *before* a movement command to the new target is generated and executed. Since predicting future movement may decrease this "reaction time" interval by compensating for delays, reaches made with future prediction could appear to be improved only because the time taken, the distance traveled, and the direction changes made during this reaction time period have been decreased. From the data shown, it is unclear if the performance benefits result from a decreased reaction time, an improved reach, or both.

References

[1] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates," *PLoS Biology*, vol. 1, no. 2, p. e2, 2003.

[2] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," Nature, vol. 453, no. 7198, pp. 1098-1101, May 2008.

[3] A. H. Fagg, G. W. Ojakangas, L. E. Miller, and N. G. Hatsopoulos, "Kinetic Trajectory Decoding Using Motor Cortical Ensembles," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 17, no. 5, pp. 487-496, Oct. 2009.

[4] A. J. Suminski, D. C. Tkach, and N. G. Hatsopoulos, "Exploiting multiple sensory modalities in brain-machine interfaces," Neural Networks, vol. 22, pp. 1224–1234, 2009.