The impact of command signal power distribution, processing delays, and speed scaling on neurally-controlled devices

A R Marathe\textsuperscript{1,2,3,4} and D M Taylor\textsuperscript{1,2,3,5}

\textsuperscript{1}Department of Neurosciences, The Cleveland Clinic, Cleveland, OH 44195, USA
\textsuperscript{2}Department of Biomedical Engineering, Case Western Reserve University, Cleveland, OH 44106, USA
\textsuperscript{3}Cleveland Functional Electrical Stimulation (FES) Center of Excellence, Louis Stokes VA Medical Center, Cleveland, OH 44106, USA
\textsuperscript{4}Human Research and Engineering Directorate, US Army Research Laboratory, Aberdeen Proving Ground, MD 21005, USA

E-mail: dawn.taylor@case.edu and taylord8@ccf.org

Received 26 January 2015, revised 4 June 2015
Accepted for publication 10 June 2015
Published 14 July 2015

Abstract

Objective. Decoding algorithms for brain-machine interfacing (BMI) are typically only optimized to reduce the magnitude of decoding errors. Our goal was to systematically quantify how four characteristics of BMI command signals impact closed-loop performance: (1) error magnitude, (2) distribution of different frequency components in the decoding errors, (3) processing delays, and (4) command gain. Approach. To systematically evaluate these different command features and their interactions, we used a closed-loop BMI simulator where human subjects used their own wrist movements to command the motion of a cursor to targets on a computer screen. Random noise with three different power distributions and four different relative magnitudes was added to the ongoing cursor motion in real time to simulate imperfect decoding. These error characteristics were tested with four different visual feedback delays and two velocity gains. Main results. Participants had significantly more trouble correcting for errors with a larger proportion of low-frequency, slow-time-varying components than they did with jittery, higher-frequency errors, even when the error magnitudes were equivalent. When errors were present, a movement delay often increased the time needed to complete the movement by an order of magnitude more than the delay itself. Scaling down the overall speed of the velocity command can actually speed up target acquisition time when low-frequency errors and delays are present. Significance. This study is the first to systematically evaluate how the combination of these four key command signal features (including the relatively-unexplored error power distribution) and their interactions impact closed-loop performance independent of any specific decoding method. The equations we derive relating closed-loop movement performance to these command characteristics can provide guidance on how best to balance these different factors when designing BMI systems. The equations reported here also provide an efficient way to compare a diverse range of decoding options offline.

\textsuperscript{5} Author to whom any correspondence should be addressed.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Keywords: brain-machine interface, brain-computer interface, closed-loop control, neuroprosthesis, visually-guided reaching

(Some figures may appear in colour only in the online journal)

1. Introduction

Advances in assistive technology now provide disabled individuals the opportunity to control a wide variety of devices using movement commands decoded directly from neural signals or from retained motor activity measured via electromyograms. Translating neuromotor signals into one’s intended movement is an imperfect process that inevitably results in some errors in the decoded commands. The spatiotemporal characteristics of the command errors will depend on the quality and type of neuromotor signals used and the choice of signal processing and decoding methods applied. The assistive device being controlled may contribute additional errors and delays to the intended movement thus creating further discrepancies between the user’s intent and the resulting device motion. While the human visuomotor system is well equipped to correct for the type of errors that typically occur during visually-guided reaching, our natural error correction response may be suboptimal or even counterproductive for these human-machine systems with unnatural spatiotemporal error characteristics. A better understanding of how these command error features can positively or negatively impact one’s ability to correct for the resulting movement errors will facilitate optimization of command systems for closed-loop control.

In the brain-machine interfacing (BMI) literature, researchers often report their decoding errors in terms of the coefficient of determination ($R^2$) or variance accounted for when comparing the decoded movement to the assumed desired movement at each timestep. These numbers reflect the relative magnitude of the decoding errors and are often used to compare one command option or decoding method against the next. However, in this study we show that the usefulness of any command is highly dependent on much more than error magnitude. The spatiotemporal characteristics of the command error itself also affects one’s response to the visually-perceived movement errors and can greatly impact one’s ability to make corrections to these movement errors in real time.

In this study, we quantify the combined impact of four factors on how well users can generate target-directed cursor movements: (1) command error magnitude, (2) the distribution of power across different frequency components in the command error, (3) visual feedback delay, and (4) command signal gain. A handful of studies have looked at some of these factors in the context of specific decoder parameters that alter subsets of these features in tandem [1–3]. This study took a broader decoder-independent approach and characterized how people respond to all possible combinations of these four factors during closed-loop control. Human subjects were tested in a closed-loop BMI simulator where each of these command characteristics was systematically controlled. The resulting data enabled us to generate equations relating these different command properties and their interactions to metrics of closed-loop performance. These generalized equations can now serve as a guide for setting design criteria for assistive device command systems as well as facilitate efficient comparisons between a diverse range of decoding algorithms offline. We also provide examples illustrating how some common neural decoding techniques for reducing error magnitude may not be optimal as they can also exacerbate additional harmful characteristics of the command.

2. Methods

2.1. Experimental design overview

Seven able-bodied participants (4 male and 3 female) performed a two dimensional center-out target-acquisition task under conditions designed to simulate BMI use. Controlled levels of error and delay were added to the person’s actual wrist movements as they used their wrist to control a cursor on a screen in a center-out target acquisition task. Their wrist conveyed their actual intended movement, while the added errors and visual feedback delays represented different types of BMI decoding errors and system processing delays. Movements were tested in blocks of 16 radial targets. These 16 targets included two target sizes (1.2 or 1.8 cm), two radial distances from the center (10 or 16 cm), and four ‘corner’ positions (up and left, up and right, down and right, down and left). Participants had 30 s to get the cursor from the center to the target and hold the cursor within the target continuously for two seconds.

The person’s actual wrist position was tracked at 10 Hz via an Optotrak Certus system (Northern Digital). The cursor position was updated on the screen every ~100 ms. Instead of using a one-to-one mapping where wrist position controlled cursor position, subjects used their arm like a joystick where their wrist position controlled cursor velocity much like using the position of a joystick to control the velocity of an object in a video game. This mapping of wrist position (with noise and delays added) to cursor velocity allowed the person to easily convey how they wanted the cursor to move. However, this mapping served to disassociate normal proprioceptive feedback from the cursor movements, thus forcing participants to rely primarily on visual feedback to correct for errors. This disassociation more realistically mimics BMI use where sensory feedback is usually limited to vision, especially for control of external devices such as a computer cursor. This position-to-velocity mapping also ensured that all test conditions had a similar disassociation between visual and proprioceptive feedback including the no-noise-added control blocks representing perfect error-free decoding.

We have previously studied this wrist-position-to-cursor-velocity transformation and have shown that people learn to
Before data collection started, and all participants felt confident in starting data collection after practicing several blocks of these movements.

2.2. Simulating movement errors

Movement errors can come from inaccuracies in decoding the intended movement command from the neural signals as well as from inaccuracies produced by the device being controlled. The characteristics of device movement errors will vary substantially depending on the type of device being controlled. Therefore, for this study, we focused on movement errors with power distributions or temporal ‘smoothness’ characteristics that are common in BMI-controlled devices. Specifically, we spanned a range of error magnitudes and power distributions seen in our experimental data when different types of decoders were applied as well as additional smoothing and delays that can come from the low-pass filtering effects of various neuroprosthetic devices and multi-stage data transmission systems.

BMI decoding functions such as linear filters, Kalman filters, and some types of neural network decoders use both current neural data plus neural or predicted movement data from the recent past to produce smoother, more-gradually-changing movement predictions. This strategy works to generate command trajectories that resemble natural arm movements because the movement parameters being decoded (e.g. position, velocity, joint angles, etc) generally change on a slow enough timescale that recent past data tells us something useful about the movement in the current timestep. While the reliance on substantial amounts of past data for decoding can improve prediction accuracy and reduce the magnitude of errors, the inclusion of past data in the decoding tends to have a low-pass filtering effect on the predicted movement and therefore also on any remaining decoding errors. Decoders that use up to a second or more of past data are common in the BMI field [6–18].

Figures 2(a)–(d) gives a typical example of how the inclusion of more past neural data improved the accuracy of the decoded arm movements in a monkey model by smoothing out the high-frequency components in the predicted movement. The gray lines indicate the actual hand position of a Rhesus macaque performing a center-out arm movement task. The black lines indicate the hand positions predicted by four different decoding functions applied to the firing rates of 70 neurons and multi-neuron clusters simultaneously recorded via two 64-channel intracortical electrode arrays (Blackrock Microsystems) implanted in the arm area of the primary motor cortex. Predicted hand positions were generated using linear filters fitted via least-squares regression (a–c) or with a Kalman filter (d). In all cases, firing rates were calculated in 100 ms time bins. In (a–c) linear filters used 1, 7, or 13 100 ms time bins of firing rate data to predict current hand position—equivalent to using (a) 100, (b) 700, or (c) 1300 ms of neural data in the prediction. The addition of extra neural data made the predicted movement more closely match that of the animal’s actual arm movement by smoothing out much of the high frequency error components present in figure 2(a). While the smoother movements more closely match the actual movements, this smoothing effect also shifted the relative distribution of power in the decoding errors toward more low-frequency components and less high-frequency components.

During this study, random noise was added to the wrist-derived command signal in real-time to simulate decoding errors in the closed-loop cursor control task. Additionally, different velocity scaling factors and visual feedback delays were imposed between the movements of the person’s wrist and the movement of the cursor on the screen to simulate various BMI system configurations.
Simulated decoding errors consisted of Gaussian random noise that was either left alone (no smoothing) or convolved with a decaying exponential of two possible lengths—seven timesteps (moderate smoothing) or 13 timesteps (high smoothing). Convolving the Gaussian noise with the exponential decay functions had a low-pass filtering effect on the noise thus lowering the frequency content of the artificially-generated errors that were added to the cursor movements during the task. Figure 3 shows the normalized power distribution of the added Gaussian noise when the different smoothing options were applied. Specifically, the distribution of power shifted from the original uniform distribution associated with Gaussian random noise to one where the proportion of the power distribution in 0–1.345 Hz range increased and the proportion in the 1.345–5 Hz range decreased relative to the values seen in the uniform distribution.

Decoding functions that use more past data to predict current movement typically have the dual effect of reducing error magnitude as well as smoothing the errors by shifting the distribution of power in the errors toward the lower frequencies. However, in this study, we adjusted error power distribution (i.e. smoothness) and error magnitude independently to individually assess their impact on movement control as well as assess their interactions. Therefore, the three different types of smoothed Gaussian noise were each rescaled to generate four different magnitudes of error. Noise scaling factors were generated that resulted in $R^2$ values of 1.0 (i.e. zero noise), 0.94, 0.8, and 0.64 when calculating the coefficient of determination, $R^2$, between arm movements with and without artificially added errors. These noise scaling

**Figure 2.** Examples illustrating how different neural decoding methods change the spatiotemporal characteristics of the predicted movement. Black lines indicate the hand position predicted by four different decoding functions applied to intracortical unit activity recorded from the motor cortex of a Rhesus macaque. Gray lines indicate the monkey’s actual hand position. Predicted hand positions were generated using (a)–(c) linear filters or (d) with a Kalman filter. In all cases, firing rates were calculated in 100 ms time bins. In (a)–(c) linear filters used 1, 7, or 13 100 ms time bins of firing rate data to predict current hand position—equivalent to using the latest (a) 100, (b) 700, or (c) 1300 ms of neural data in the prediction.

**Figure 3.** Normalized power distribution of the Gaussian noise added to the cursor movements. As expected, the unsmoothed Gaussian noise (black line) had a uniform power distribution. When the Gaussian noise was convolved with decaying exponential functions of either 7 (medium gray) or 13 timesteps (light gray), the distribution of power shifted toward more low-frequency components. The dotted line at $\sim 1.345$ Hz shows a common transition point across all three distributions. Ratios of mean power values below/above this 1.345 Hz transition point were 3.2 and 4.2 for moderately-smoothed and highly-smoothed noise respectively.

Decaying exponentials were chosen because these functions approximated how contributions from current and past neural data are often weighted when optimizing arm movement decoding functions via least squares regression. Figure 4(a) shows examples of the least squares regression coefficients used to predict hand position from neural firing rates in the same animal whose data is shown in figure 2. Each line indicates the normalized magnitude of the regression coefficients for one neural unit where current and past firing rates were used to predict current hand position. Figures 4(b)–(d) qualitatively illustrates the spatiotemporal characteristics of the movement errors generated for this study. The gray line shows the actual wrist position of a participant in this study during the center-out task. The black lines show that same limb position data with the three types of simulated errors added (b = no smoothing, c = moderate smoothing, d = high smoothing).

Decoding functions that use more past data to predict current movement typically have the dual effect of reducing error magnitude as well as smoothing the errors by shifting the distribution of power in the errors toward the lower frequencies. However, in this study, we adjusted error power distribution (i.e. smoothness) and error magnitude independently to individually assess their impact on movement control as well as assess their interactions. Therefore, the three different types of smoothed Gaussian noise were each rescaled to generate four different magnitudes of error. Noise scaling factors were generated that resulted in $R^2$ values of 1.0 (i.e. zero noise), 0.94, 0.8, and 0.64 when calculating the coefficient of determination, $R^2$, between arm movements with and without artificially added errors. These noise scaling
n = 0, 1, 2, or 3 timesteps of approximately 100 ms each. The magnitude of these transmission delays can be very short in efficient, self-contained systems or substantial if the raw signals need to be transmitted to a separate receiver for processing and then the decoded command transmitted again to the device being controlled. In this study, we limited our range of delays to four different command combinations tested (a block of movements consisted of one movement to each of the 16 targets presented in random order). To minimize possible effects of arm fatigue, participants were required to rest for at least ten seconds half way through each block and for at least 20 s between blocks. Participants could also rest as much as they wanted between blocks.

Figure 4. (a) Examples linear regression coefficients generated when motor cortical firing rates recorded in a Rhesus macaque were used to predict the animal’s hand position during a center out task. Both current (t–0) and past (t–n) timesteps of neural data were used to predict the animal’s current hand position. Notice the gradual decrease in coefficient magnitude that somewhat resembles exponential decay. (b)–(d) An example human hand movement from this study before (gray) and after (black) simulated errors of different smoothness levels were added. The added errors consisted of plain Gaussian noise (b) or Gaussian noise convolved with exponential decay functions of either 7 (c) or 13 (d) timesteps in length to simulate unsmoothed, moderately smoothing, and highly smoothed errors respectively. Note that all three types of errors were rescaled after smoothing to have the same relative magnitudes.

Figure 4. (a) Examples linear regression coefficients generated when motor cortical firing rates recorded in a Rhesus macaque were used to predict the animal’s hand position during a center out task. Both current (t–0) and past (t–n) timesteps of neural data were used to predict the animal’s current hand position. Notice the gradual decrease in coefficient magnitude that somewhat resembles exponential decay. (b)–(d) An example human hand movement from this study before (gray) and after (black) simulated errors of different smoothness levels were added. The added errors consisted of plain Gaussian noise (b) or Gaussian noise convolved with exponential decay functions of either 7 (c) or 13 (d) timesteps in length to simulate unsmoothed, moderately smoothing, and highly smoothed errors respectively. Note that all three types of errors were rescaled after smoothing to have the same relative magnitudes.

Factors were calculated from wrist position data collected during practice trials to far targets before any errors or delays were added.

When controlling external devices, delays can occur due to the time needed for neural signal acquisition and decoding as well as the time needed to transmit the command to/from the device being controlled. The magnitude of these transmission delays can be very short in efficient, self-contained systems or substantial if the raw signals need to be transmitted to a separate receiver for processing and then the decoded command transmitted again to the device being controlled. In this study, we limited our range of delays to four different options. Cursor velocity at timestep t was set to the wrist position value (with errors added) from timestep t–n where n = 0, 1, 2, or 3 timesteps of approximately 100 ms each.

In addition to the noise and delay characteristics described above, we also considered the effect of velocity gains on performance. In 1954, Fitts first showed that people naturally scale up or down the speed of their movements depending on the accuracy requirements of a task [19]. With BMIs, people can still volitionally speed up or slow down the device movement by altering their neural output just as they do when controlling their own limbs. However, the decoded velocity is often additionally scaled by the device control system for two reasons: (1) to convert the motor command to units appropriate for the device (e.g. pixels per second for a brain-controlled computer cursor versus meters per second for a brain-controlled wheelchair), and (2) to ensure movement decoding errors at each timestep do not exceed an acceptable magnitude for the accuracy requirements of the task.

In this study, we considered two different velocity gain factors were tested. These factors were chosen empirically during practice trials to emphasize either movement accuracy (low gain factor) or movement speed (high gain factor). Recall participant’s wrist position was mapped to cursor velocity. Therefore, to move the cursor to the target, a participant would move his or her wrist out and then back to the center to generate cursor trajectories that approximately followed a bell-shaped velocity profile. For the high gain factor, 0.75 s−1 was chosen because that value was just high enough that a full extension of the arm out and immediately back to the center would typically result in the cursor stopping at the correct radial distance when no noise was added. Higher gain factors were not used because the person would not be able to use the full extension of their arm without overshooting the target. Any smaller gain factor would not optimize speed because the person would have to hold their arm fully extended and wait for some duration before bringing their wrist back to center in order to stop the cursor in the target. For the low gain factor, 0.5 s−1 was chosen because that was slow enough to enable participants to acquire most targets in the high-error-magnitude conditions in the absence of delays. Note that the units of ‘per second’ serve to convert position to velocity.

Table 1 summarizes the different error magnitudes, error smoothness levels, visual feedback delays, and gain setting used in this study. All combinations of error magnitude, error smoothness, and visual feedback delays were tested using each gain factor for a total of 80 combinations tested per person; the number of combinations tested was less than the full 96 possible combinations because the tests where the error magnitude is zero did not need to be repeated with different amounts of error smoothing.

To ensure that arm fatigue did not become a problem, the testing was broken up into at least four and sometimes five different testing sessions. The order in which the different parameter combinations were tested was randomized for each person. Participants performed one complete block of movements for each combination tested (a block of movements consisted of one movement to each of the 16 targets presented in random order). To minimize possible effects of arm fatigue, participants were required to rest for at least ten seconds half way through each block and for at least 20 s between blocks. Participants could also rest as much as they wanted between blocks.

Table 1 summarizes the different error magnitudes, error smoothness levels, visual feedback delays, and gain setting used in this study. All combinations of error magnitude, error smoothness, and visual feedback delays were tested using each gain factor for a total of 80 combinations tested per person; the number of combinations tested was less than the full 96 possible combinations because the tests where the error magnitude is zero did not need to be repeated with different amounts of error smoothing.

To ensure that arm fatigue did not become a problem, the testing was broken up into at least four and sometimes five different testing sessions. The order in which the different parameter combinations were tested was randomized for each person. Participants performed one complete block of movements for each combination tested (a block of movements consisted of one movement to each of the 16 targets presented in random order). To minimize possible effects of arm fatigue, participants were required to rest for at least ten seconds half way through each block and for at least 20 s between blocks. Participants could also rest as much as they wanted between blocks.
Table 1. Parameters ranges evaluated in this study.

| Characteristic | Values tested |
|----------------|---------------|
| Relative error magnitude (coefficient of determination, $R^2$) | 1 (no errors added), 0.94, 0.8, and 0.64 |
| Error power distribution or ‘smoothness’ determined by the length of the exponential decay function (in 100 ms timesteps) convolved with the Gaussian noise | Unsmoothed (1, current timestep only), moderately smoothed (7 timesteps), and highly smoothed (13 timesteps) |
| Added visual feedback delay in 100 ms timesteps | 0, 1, 2, and 3 100 ms timesteps |
| Velocity gain factor | Lower (0.5 s$^{-1}$), and higher (0.75 s$^{-1}$) |

wanted between any two movements and were encouraged to do so any time they felt tired.

All human testing was approved by the Institutional Review Board of the Louis Stokes Cleveland Veterans Affairs Medical Center. All animal data discussed or shown were collected during testing approved by the Institutional Animal Care and Use Committee of the Cleveland Clinic.

### 2.3. Data analysis

Four performance metrics were calculated from the movement data for each participant and each test condition: (1) movement time, (2) percentage of targets successfully acquired, (3) the number of times the cursor re-entered the target after first leaving the target (overshoots), and (4) path efficiency. Path efficiency was defined as the straight line distance to the target divided by the actual distance traveled before the target was acquired or the trial timed out. Path efficiency values of one indicate optimal efficiency. Lower values indicate less efficient movement paths.

To quantify the relationship between the decoding/device parameters (magnitude, smoothness, delay, and gain) and each performance metric, least squares regression models were generated for each performance metric. Equation (1) shows the form of the linear model for predicting movement time from the four decoding parameters and their six interaction terms.

$$
\text{Movement time} = b_0 + b_1 R^2 + b_2 S + b_3 D + b_4 G + b_5 R^2 S + b_6 R^2 G + b_7 R^2 D + b_8 S G + b_9 D G + b_{10} D S
$$

In the above model, letters refer to the following: $R^2 =$ error magnitude, $S =$ smoothness, $D =$ delay, $G =$ velocity gain. Similar regression models were also calculated for the three other performance metrics (number of overshoots, percentage of targets acquired, and path efficiency). Models with higher order interaction terms were also evaluated but they provided little additional insight. Therefore, we limited our reporting here to the four primary factors and six pair-wise interaction terms.

Fitting our experimental data to these linear models enabled us to compare the relative scales of the model coefficients as a way of quantifying the relative importance of these different factors and interaction terms on movement performance. However, for this type of comparison to be valid, the different movement parameters and performance measures had to first be normalized to the same relative scales. Therefore, the actual $R^2$, $S$, $D$ and $G$ values listed in table 2 were normalized to between −1 and 1 and the resulting performance measures were each z-scored before building the models.

Matlab’s *stepwisefit* function was used to confirm that only terms that made a unique and significant contribution to each model were identified. This stepwise process added and/or removed terms to the linear regression model one at a time based on if the addition/removal made a significant improvement in the fit of the data to the model. For each movement performance measure, this stepwise fitting process was repeated 1000 times with the order in which terms were added into the model randomized differently each time. Since each order randomization resulted in the same terms always being included in the final resulting models, this process confirmed that there were no redundant terms in the final models and the terms deemed significant by the stepwise fitting process each contributed unique information to the models. However, the sets of terms determined as significant differed depending upon which movement performance metric was being fit.

### 3. Results

Figure 5 shows an example of how movement performance changed as the error characteristics were modified. Trajectories from all participants were qualitatively similar to the example shown here. Moving from left to right along the top row of plots, the upper left trajectory shows movements under ‘ideal’ conditions (no error, no added delay, low velocity). As expected, adding error resulted in increased variability or ‘wiggle’ in the trajectories. However, as the added error also became smoother, the person had a much harder time getting to the targets even though the error magnitude had not changed. Adding a delay on top of the smoothed errors further exacerbated the control problem resulting in large trajectory deviations and problems maintaining the cursor in the targets. All of the plots in the top row were generated under the low-velocity-gain condition. The plots in the bottom row were generated under equivalent error and delay conditions except that the high velocity gain was used. The higher gain further exacerbated the problems with control in all conditions except the left most plots where no errors or delays were added. Note the magnitude of the added error was constant for...
all six plots to the right of the first column. However, the movement control problems were quite different depending on the combination of error smoothness, the applied gain factor, and the amount of visual feedback delay in the system.

Figure 6 shows how the range of different parameter settings impacted the movement time or the time required to successfully acquire each target. Plots were generated by averaging movement times across all participants for each combination of parameters. Linear interpolation was used to fill in between the 80 parameter combinations tested.

Table 2 shows details of the relationships between each movement performance metric and the four parameters varied in this study ($R^2 = \text{error magnitude}$, $S = \text{error smoothness}$, $D = \text{delay}$, $G = \text{velocity gain}$). Each row contains the coefficients resulting from the least squares linear models of how predicting movement time as well as analogous equations for path efficiency. Coefficients listed in the table correspond to terms that were significant in the final model that best fit the data for each of the four performance metrics. Ninety-five percent confidence intervals for each individual coefficient are included in parentheses underneath.

Movement time and number of overshoots (first two rows in the table) are metrics where lower values indicate better performance. Number of targets acquired and path efficiency (last two rows in the table) are metrics where higher values indicate better performance. Note the signs of the coefficients are the same between the first two rows and opposite those in the last two rows. These consistent sign trends indicate that the helpful or harmful effect of each individual parameter or interaction term was consistent across all movement performance metrics.

4. Discussion

In this study, seven human subjects performed a target acquisition task on a computer screen using a closed-loop BMI simulator where different types of noise, delays, and gains were added to the cursor motion to simulate the combined effect of imperfect decoding and the response properties of various BMI systems. The BMI simulator was used here because, unlike actual BMI systems, the closed-loop simulator allowed us to systematically vary four key command/device features over a range of values in all subjects. This systematic testing enabled us to fit linear models of how these command parameters and their interactions impact BMI performance. By normalizing the different parameters to the same scale before fitting the linear models, the relative magnitudes of the model coefficients can be directly compared revealing their relative importance in BMI performance.

4.1. Effect of error characteristics

As expected, error magnitude had the largest impact on all movement performance measures as indicated by the $R^2$ terms having the largest coefficients in all performance models. However, the effects of the other parameters were less intuitive. The progression of the plots from left to right in figure 6 illustrate how simply lowering the frequency content of the simulated decoding errors can substantially increase the time required to complete the movement even when the error magnitude remained the same.

The motor control literature, as well as comments from our subjects, can provide some insights into this unexpected result. Prior studies on visually-guided reaching have shown that, if a hand-controlled cursor undergoes an unexpected visual displacement during a target-directed reach, a fast visually-driven corrective response is made to redirect the trajectory back toward the original target even without the subjects being aware they are making the correction [20–22]. Although this rapid visually-driven reflexive response is faster than the fastest responses subjects could consciously make to visual perturbations, these reflexive corrections tend to be suppressed and the deviation ignored if the person learns through experience that an equal and opposite displacement would subsequently occur making corrections to the initial deviation unnecessary [22]. Here we add to that body of knowledge by showing that the tendency to suppress this automatic error correction response is highly dependent on the time window over which sequential visual perturbations are likely to cancel each other out.

In our BMI simulator, subjects experienced a continuous stream of random visually-perceived displacements from the intended cursor path to simulate decoding/device errors. As in real BMIs, some sequential displacements will cancel each other out over time making correcting for each deviation unnecessary and even counterproductive. Participants in our study reported generally ignoring the higher-frequency jitter and focusing their efforts on getting the underlying mean cursor path to the target. However, when maximally-smoothed noise was added, participants repeatedly made corrective movements throughout the task as they perceived the lower-frequency cursor deviations as errors they generated themselves that required correcting. Figure 5 illustrates how these attempts to correct for low-frequency errors were counterproductive (column 3) as they resulted in larger deviations from a straight-to-target path compared to the unsmoothed noise case (column 2). This error correction response was most problematic in the high-gain condition.

Although it seems counter intuitive that correcting for the added errors in the smoothed-noise case caused larger path deviations than ignoring the added errors in the unsmoothed case, the zero-mean nature of the added errors made ignoring these errors an effective strategy for placing the mean cursor position over the target. In real systems, not all decoding errors will have a zero mean over time. However, previous research has shown that subjects can learn to compensate for consistent components of decoding errors in BMI systems by making compensatory feedforward alterations to their movement paths, modifying their ensemble tuning properties, and even incorporate internal models of the BMI system [2, 23–29]. Our study suggests that people will have the most trouble correcting for random errors with similar spatiotemporal characteristics as the movement one is trying to control. Those errors may be too slow to be ignored and averaged out
Table 2. Linear models predicting movement performance.

|        | $b_0$ | $b_1 R^2$ | $b_2 S$ | $b_3 D$ | $b_4 G$ | $b_5 R^2 S$ | $b_6 R^2 G$ | $b_7 R^2 D$ | $b_8 S G$ | $b_9 D G$ | $b_{10} D S$ |
|--------|-------|------------|---------|---------|---------|-------------|-------------|-------------|-----------|-----------|------------|
| Movement time |       |            |         |         |         |             |             |             |           |           |             |
| —      | -0.92 | 0.26       | 0.30    | 0.33    | -0.23   | -0.51       | -0.38       | 0.097       | 0.092     | 0.13      | 0.13       |
|        | (-0.96, -0.87) | (0.22, 0.30) | (0.27, 0.34) | (0.29, 0.37) | (-0.28, -0.17) | (-0.56, -0.46) | (-0.42, -0.33) | (0.045, 0.15) | (0.050, 0.13) | (0.093, 0.17) |
| Number of overshoots |       |            |         |         |         |             |             |             |           |           |             |
| —      | -0.96 | 0.21       | 0.37    | 0.22    | -0.16   | -0.31       | -0.40       | —           | 0.059     | 0.047     | 0.047      |
|        | (-1.0, -0.91) | (0.17, 0.26) | (0.34, 0.41) | (0.17, 0.26) | (-0.22, -0.094) | (-0.37, -0.26) | (-0.45, -0.35) | (0.013, 0.11) | (0.005, 0.89) |
| Number of targets acquired |       |            |         |         |         |             |             |             |           |           |             |
| —      | 0.73  | -0.21      | -0.33   | -0.28   | 0.26    | 0.54        | 0.45        | -0.14       | -0.12     | -0.17     | -0.17      |
|        | (0.67, 0.79) | (-0.27, -0.16) | (-0.37, -0.28) | (-0.33, -0.23) | (0.18, 0.34) | (0.47, 0.61) | (0.40, 0.51) | (-0.21, -0.075) | (-0.18, -0.066) | (-0.23, -0.12) |
| Path efficiency |       |            |         |         |         |             |             |             |           |           |             |
| —      | 1.1   | -0.28      | -0.36   | -0.19   | —       | 0.18        | 0.15        | —           | -0.041    | -0.033    | -0.033     |
|        | (1.1, 1.2) | (-0.31, -0.24) | (-0.38, -0.34) | (-0.22, -0.17) | (0.14, 0.21) | (0.12, 0.18) | (0.068, 0.014) | (-0.058, -0.0085) |
over time. However, they vary enough in time that they cannot be learned and compensated for with feedforward corrections. Most researchers tend to choose decoding algorithms and settings that produce predicted movements with spatiotemporal characteristics that best mimic natural movements. Our study suggests that the decoding errors resulting from those choices will have spatiotemporal characteristics that are particularly difficult for BMI users to appropriately compensate for during closed-loop control.

4.2. Effect of visual feedback delay
By its very nature, delay should inherently increase movement time by the amount of the delay itself. Therefore, delay

Figure 5. Trajectories from one example subjects illustrating the impact of changing error magnitude, error smoothness, visual feedback delay, and velocity gain. Each set of center-out trajectories show the cursor paths taken to all 16 targets (eight locations and two sizes) under each condition. Black dots show when the cursor was within the target. The numbers below each plot indicate the parameters from table 1 that were used for each set of trajectories. Numbers in parentheses from left to right: error magnitude in terms of $R^2$, error smoothness in terms of the length of the exponential decay function convolved with the Gaussian noise, visual feedback delay in timesteps, and velocity gain.

| Unsmoothed Errors | Moderately Smoothed Errors | Highly Smoothed Errors |
|-------------------|----------------------------|------------------------|
| Low Velocity Gain |                           |                        |
| High Velocity Gain|                           |                        |
should be a significant factor in the movement time model. However, all movement performance metrics were significantly affected by delay. In the path efficiency model, delay was the second largest factor even though path efficiency is a metric that is independent of any aspect of timing. Figure 6 shows that delaying cursor movements by only 300 ms increased average movement time by over five seconds in some error conditions—an order of magnitude higher than the delay itself. The extensive ‘wandering’ or ‘fishing’ around the targets in the right-most column of trajectories in figure 5 suggests subjects made repeated unsuccessful attempts at correcting errors when delays were present.

Delays may be unavoidable in many BMI systems. The large interaction terms between error magnitude and delay also suggests that delays specifically interfere with the normal error correction processes. A corrective movement that is delayed from the time it was designed to be implemented will likely be ineffective as the cursor will have traveled during the delay causing an unintended offset in the starting point for the corrective adjustment. This offset in the starting point would, in turn, cause the ending location to also be offset from the desired target. A vicious cycle then results from anticipating an error, and implementing a correction process that inadvertently adds new errors due to its delayed implementation.

Given that decoding errors and some processing delays are an unavoidable aspect of most BMI systems, the interaction between delays and error magnitude makes minimizing delays of high importance in BMI controlled devices, especially when decoding accuracy is poor. Note however, the interaction term between delay and error smoothness was also significant for all four performance metrics. This interaction effect may be due to more error correction attempts being made in the smoothed-noise condition compared to the unsmoothed condition.

4.3. Smoothing and inclusion of past data in BMIs

Figure 2 gives an example of how high-frequency jitter is reduced by the inclusion of more past data to predict current movements thus making the frequency content of the predicted movement closer to that of the actual limb. Optimizing smoothing parameters to the specific characteristics of neural activity and intended movement is important for maximally reducing decoding errors [2]. However, by independently varying command smoothness and error magnitude in this study, we illustrate the complex nature of this smoothing/error-reduction trade off and how control problems can arise when decoding errors have a spectral profile similar to the underlying movement being controlled, especially if there are also delays in the system.

The inclusion of more past data to predict current movement not only smooths out the noise, but it can also add delays thus compounding the control problems. Note the slight temporal shift between the actual (gray) and predicted (black) movements in figure 2, especially in (d). As part of another ongoing study, we have quantified the prediction delays that resulted from applying different arm movement decoding functions to various types of neuromotor signals collected during point-to-point reaching movements (i.e. unit activity and local field potentials from intracortical electrodes and field potentials from high-density epidural electroencephalography grids). Kalman filter decoders tended to generate the largest prediction delays (typically 100–200 msec) compared to linear filters, regardless of the type of neural signals used. Although Kalman filters do not use past neural data directly, they use past movement predictions plus current neural data to generate the current movement prediction. That past movement prediction is based on past neural data and an even older past prediction, which is based on even older neural data, etc.

Linear filters generally resulted in smaller prediction delays than Kalman filters but the size of the delay was still significantly correlated to the amount of past neural data used to predict current movement (i.e. more past data resulted in longer prediction delays). One trick that can potentially reduce the inherent lag in the decoded signal is to generate decoding functions using current/past neural data to predict future desired movement. Willett et al [16] demonstrated how predicting future desired movement could be used to counteract the lag in the BMI system thus improving closed-loop control. However, this technique has its limits as decoding accuracy typically declines the farther into the future movement is predicted.

Including more past neural data to predict current movement will have both a beneficial effect of reducing the size of decoding errors and a double harmful effect of lowering the relative frequency content of the errors and potentially adding prediction delays. Simply basing the goodness of a decoding method on an offline metric of decoding error is likely to be inadequate for predicting the best option for online use given the importance of these other factors on the ability to correct for errors in real time. Identifying the optimal amount of past data to predict current movement requires balancing the helpful and harmful effects of including past data.

Cunningham et al [3] provided a good demonstration of how offline measures of decoding accuracy are not adequate for identifying the best decoding parameters for online BMI use. In that study, Kalman filter firing rate bin sizes of 100–200 ms were optimal in offline decoding, but smaller bins of 25–50 ms resulted in better performance during closed-loop control. Consistent with our findings, these smaller bin sizes likely reduced delays in the system and may have shifted the distribution of power in the decoding errors to a higher frequency range.

The performance equations in table 1 developed in this study now provide a way to estimate and compare the relative online performances of different signal processing and decoding options based on the error reduction, smoothing, and delay effects of the different decoding algorithms under consideration. Therefore, these equations can help efficiently identify the best decoding algorithm candidates without performing time-consuming online comparative testing on all options under consideration. Note, however, when using these equations it is important to include any errors and delays.
added by the transmission process and the device itself and not just those resulting from the decoding algorithm alone.

4.4. Using velocity gain to our advantage

While BMI control can be improved by striking the best balance between reducing error magnitude at the expense of increased error smoothness and increasing prediction delays, the quality of even an optimized system will ultimately be limited by the quality of the movement-related information in the recorded neural signals. Fits et al [19] identified an inherent speed-accuracy trade off showing that people naturally adjust their movement speed to accommodate the accuracy requirements of the task. In BMI systems, speed adjustment can come from two sources—(1) the users continuously adjusting their own attempted movement speed during use, and (2) the overall speed or gain of the command system which can be adjusted directly by arbitrarily scaling up or down the decoded command, or adjusted indirectly by changing parameters in the decoding function itself [1].

On the surface, our results for velocity gain were somewhat counterintuitive. One would expect that increasing the velocity gain would shorten movement time. However, just the opposite was true. Under the high-noise conditions, a 50% increase in velocity increased the time needed to acquire the target by five to ten seconds in some cases.

While variability and errors tend to scale with speed during normal arm movement, decoding errors do not necessarily scale with the decoded speed in BMIs. This point can be illustrated by the extreme example where the speed is reduced to zero. Most healthy individuals can hold their hand relatively stationary even though most will have significant path variations during ballistic movements. However, many BMI decoding functions will have prediction errors at all speeds including at zero velocity. The inability to maintain a zero-velocity during closed-loop control is exacerbated by the fact that speed information is more poorly represented in the motor cortical firing rates compared to directional information [30]. Therefore, the user cannot simply reduce the size of the decoding errors by trying to make slower movements.

On the other hand, the velocity gain factor imposed by the BMI system does scale the decoding errors in tandem with the decoded movement. The relatively large coefficient for the interaction term between error magnitude and velocity gain suggests the same Fitts-law principles apply to how velocity scaling should be used in BMI systems. To put this in terms of mutual information, as the information transfer rate of the command signal decreases (i.e. as decoding $R^2$ goes down), the movement speed also needs to be reduced to allow more time during a movement to accumulate the amount of information needed to reach the correct goal location. To address velocity gain in more functional terms, slowing the movement down prevents the errors from throwing the movement too far off course before path corrections can be made.

This study highlighted the challenges with error correction in BMIs stemming from the delays added by BMI systems and the novel finding that the relative power distribution of the command errors alters our natural response to these visually-perceived perturbations in our trajectories. A very low gain factor could potentially enable people with very noisy, poor-quality device command signals to still make functionally-useful movements by strongly limiting the extent of any path deviation before corrective responses can be implemented. The equations we provide here in table 2 can provide guidance for optimizing this gain factor as well as for designing BMI decoders and systems that appropriately take into account the combined impact that command error magnitude, error power distribution, and visual feedback delays have on closed-loop performance.

Acknowledgments

This project was supported by the Department of Veterans Affairs merit review #B4195R, the National Institutes of Health NINDS R01NS058871, the Cleveland Clinic, and Case Western Reserve University.

References

[1] Gowda S et al 2014 Designing dynamical properties of brain-machine interfaces to optimize task-specific performance IEEE Trans. Neural Syst. Rehabil. Eng. 22 911–20
[2] Koyama S et al 2010 Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control J. Comput. Neurosci. 29 73–87
[3] Cunningham J P et al 2011 A closed-loop human simulator for investigating the role of feedback control in brain–machine interfaces J. Neurophysiol. 105 1932–49
[4] Marathe A R and Taylor D M 2011 Decoding position, velocity, or goal: does it matter for brain–machine interfaces? J. Neural Eng. 8 025016
[5] Marathe A R and Taylor D M 2013 Decoding continuous limb movements from high-density epidural electrode arrays using custom spatial filters J. Neural Eng. 10 036015
[6] Serruya M D et al 2002 Instant neural control of a movement signal Nature 416 141–2
[7] Serruya M et al 2003 Robustness of neuroprosthetic decoding algorithms Biol. Cybern. 88 219–28
[8] Carmena J M et al 2003 Learning to control a brain–machine interface for reaching and grasping by primates PLoS Biol. 1 E42
[9] Paninski L et al 2004 Spatiotemporal tuning of motor cortical neurons for hand position and velocity J. Neurophysiol. 91 515–32
[10] Wessberg J and Nicolelis M A 2004 Optimizing a linear algorithm for real-time robotic control using chronic cortical ensemble recordings in monkeys J. Cogn. Neurosci. 16 1022–35
[11] Carmena J M et al 2005 Stable ensemble performance with single-neuron variability during reaching movements in primates J. Neurosci. 25 10712–6
[12] Lebedev M A et al 2005 Cortical ensemble adaptation to represent velocity of an artificial actuator controlled by a brain–machine interface J. Neurosci. 25 4681–93
[13] Wu W and Hatsopoulos N G 2008 Real-time decoding of nonstationary neural activity in motor cortex IEEE Trans. Neural Syst. Rehabil. Eng. 16 213–22
[14] Chao Z C, Nagasaka Y and Fujii N 2010 Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkeys Front. Neuroeng. 3 3
[15] Liang N and Bougrain L 2012 Decoding finger flexion from band-specific ECoG signals in humans Front. Neurosci. 6 91

[16] Willett F R et al 2013 Improving brain–machine interface performance by decoding intended future movements J. Neural Eng. 10 026011

[17] Chhatbar P Y and Francis J T 2013 Towards a naturalistic brain-machine interface: hybrid torque and position control allows generalization to novel dynamics PLoS One 8 e52286

[18] Nakanishi Y et al 2014 Decoding fingertip trajectory from electrocorticographic signals in humans Neurosci. Res. 85 20–7

[19] Fitts P M 1954 The information capacity of the human motor system in controlling the amplitude of movement J. Exp. Psychol. 47 381–91

[20] Saunders J A and Knill D C 2005 Humans use continuous visual feedback from the hand to control both the direction and distance of pointing movements Exp. Brain Res. 162 458–73

[21] Saunders J A and Knill D C 2003 Humans use continuous visual feedback from the hand to control fast reaching movements Exp. Brain Res. 152 341–52

[22] Franklin D W and Wolpert D M 2008 Specificity of reflex adaptation for task-relevant variability J. Neurosci. 28 14165–75

[23] Taylor D M 2007 The importance of online error correction and feed-forward adjustments in brain-machine interfaces for restoration of movement Towards Brain–Computer Interfacing ed G Dornhege et al (Cambridge, MA: MIT Press) p 162–73

[24] Jarosiewicz B et al 2008 Functional network reorganization during learning in a brain–computer interface paradigm Proc. Natl Acad. Sci. USA 105 19486–91

[25] Chase S M, Schwartz A B and Kass R E 2009 Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain–computer interface algorithms Neural Netw. 22 1203–13

[26] Ganguly K and Carmena J M 2009 Emergence of a stable cortical map for neuroprosthetic control PLoS Biol. 7 e1000153

[27] Ganguly K et al 2011 Reversible large-scale modification of cortical networks during neuroprosthetic control Nat. Neurosci. 14 662–7

[28] Chase S M, Kass R E and Schwartz A B 2012 Behavioral and neural correlates of visuomotor adaptation observed through a brain-computer interface in primary motor cortex J. Neurophysiol. 108 624–44

[29] Golub M D, Yu B M and Chase S M 2012 Internal models engaged by brain-computer interface control Proc. Conf. IEEE Engineering in Medicine and Biology Society pp 1327–30

[30] Golub M D et al 2014 Motor cortical control of movement speed with implications for brain-machine interface control J. Neurophysiol. 112 411–29