A Nonlinear Latching Filter to Remove Jitter from Movement Estimates For Prostheses

Jacob Nieveen, Mark Brinton, David J. Warren *Senior Member, IEEE*, V John Mathews *Fellow, IEEE*

***Abstract*—Continuous movement intent decoders are critical for precise control of hand and wrist prostheses. Noise in biological signals (e.g., myoelectric or neural signals) can lead to undesirable jitter in the output of these types of decoders. A low-pass filter (LPF) at the output of the decoder effectively reduces jitter, but also substantially slows intended movements. This paper introduces an alternative, the latching filter (LF), a recursive, nonlinear filter that provides smoothing of small-amplitude jitter but allows quick changes to its output in response to large input changes. The performance of a Kalman filter (KF) decoder smoothed with an LF is compared with that of both an KF decoder without an additional smoother and a KF decoder smoothed with a LPF. These three algorithms were tested in real-time on target holding and target reaching tasks using surface electromyographic signals recorded from 5 non-amputee subjects, and intramuscular electromyographic and peripheral neural signals recorded from an amputee subject. When compared with the LPF, the LF provided a statistically significant improvement in amputee and non-amputee subjects' ability to hold the hand steady at requested positions and achieve movement goals faster. The KF decoder with LF provided a statistically significant improvement in all subjects' ability to hold the prosthetic hand steady, with only slightly lower speeds, when compared to the unsmoothed KF.**

I. Introduction

Loss of a hand can be devastating to a person's ability to interact with objects and perform many daily tasks. Many individuals with amputation use prosthetic devices to recapture some of the abilities lost as a result of amputation.

Highly-enabled prostheses require accurate movement intent decoders in order to provide a level of dexterity that approaches the native hand. Such decoders typically process biological control signals such as neural signals (recorded, for example, from the motor cortices or peripheral nerves) [1] or electromyograms (EMG) recorded from muscles in the forearm of transradial amputees [2].

Noise in these signals adversely affects movement intent decoding methods. Sources of noise include recording devices, electrodes in partial contact with biological sources, cable movement, and AC power sources [3]. There are also biological sources of noise, such as involuntary neural or muscle activity [4], incomplete sampling of all motor encoding signals, or recording of signals extraneous to a particular decode. It has also been demonstrated that noise in the neural signals themselves is signal dependent, *e.g.*, the variance of neural signals increases as the mean amplitude of the signals themselves increase [5]. Because the biological control signals used to inform prosthetic movement often originate with action potentials containing sharp impulses of electrical activity, features derived from these signals can also have substantial noise.

Many methods have been proposed to reduce jitter from movement intent estimates. One approach is to reduce noise in features. Wavelet denoising [6], [7], [8] and Wiener filtering [4], [9] have been used to reduce noise in raw biological signals, but these approaches may not improve the inherent variability of the extracted features. Spectral whitening has been reported effective in decreasing the variance of EMG signals [10], and in improving classification accuracy for a classifier-based movement intent decoder [11]. Bayesian filtering is a valuable tool in providing reduced-noise EMG amplitude estimates [12]. When used with a regression-based decoder, Bayesian filtering based features outperformed mean absolute value features in target reaching tasks [13], [14].

An alternative to using noise-removal techniques directly on the biological signals is to use a decoder that inherently provides smoothing. Examples include the Kalman filter (KF) [1], [15], [16], [17], [18], [19], Gaussian process autoregression [20], and recurrent neural networks [21], [22], [23]. The KF is effective at reducing jitter compared to linear regression, but, as this paper shows, this reduction can be improved further. Gaussian process autoregressions and recurrent neural networks require substantial computational resources during training, and may require specialized hardware such as a graphics processing unit.

A third option is to use a smoother on the movement decoder output. For example, Zhang, *et al.* [24], used a system that locks onto a movement class when a constant muscle contraction is detected, preventing rapid fluctuation between decoded classes. A traditional LPF can be used with continuous decoding systems, where features are mapped to a continuous range of possible outputs [25]. However, this will introduce a delay in the decoder output. To reduce the effect of these delays, Downey, *et al.* [26] used an LPF only when a grasp is detected by a camera system. Ma [27] created a jitter removal post-processing step that moves the current position by a discrete step, only when the amplitude of the difference between its current state and current input is large. Because the step size does not change from one time sample to the next, this method is an approximation to a continuous decoder, discretized at the level of the step size. The fastest velocity of this decoder is the step size times the sampling frequency.

J. Nieveen was with the Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT, 84112 USA. He is now with Alpha Hive, Wavetronix LLC, Springville, UT 84663 USA.

M. Brinton is with the School of Engineering, Math, and Computer Science, Elizabethtown College, Elizabethtown, PA, 17022, USA.

D. J. Warren is with the Department of Biomedical Engineering, University of Utah, Salt Lake City, UT, 84112 USA.

V J. Mathews is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, Oregon, 97331 USA.

TABLE I
FREQUENTLY USED MATHEMATICAL SYMBOLS

M	Number of features
N	Number of samples
P	Number of Degrees of Freedom (DoFs)
$\mathbf{X}$	Commanded movement Data
$\mathbf{Z}$	EMG feature data
$x_{p,n}$	Position of DoF p at time n
$\hat{x}_{p,n}$	Estimated position of DoF p at time n
$\hat{x}_{p,n}^{smooth}$	Smoothed estimated position of DoF p at time n
$\hat{x}_{p,n}^{thresh}$	Thresholded $\hat{x}_{p,n}^{smooth}$
$z_{m,n}$	Feature amplitude of feature m at time n
$\mathbf{x}_{p,*}$	Vector of DoF p at all times
$\mathbf{z}_{m,*}$	Vector of feature m at all times
$\mathbf{x}_{*,n}$	Vector of all DoFs at time n
$\mathbf{z}_{*,n}$	Vector of all features at time n
$\mathbf{X}_{*,[1,n]}$	Matrix of all DoFs at times 1 to n
$\mathbf{Z}_{*,[1,n]}$	Matrix of all features at times 1 to n

The step size selection is a trade-off between precision and speed. If the step size is large, the velocity will be fast, but the decoder may lack the ability to decode precise positions. If the step size is small, the velocity will be slower, and the decoder will have more precise positional control.

This paper presents the latching filter (LF), a recursive, nonlinear and signal-dependent smoothing algorithm that can be used with any continuous decoder as a post-processing step. The LF uses the difference between its current state and current input to determine, on a continuous scale, how much to move in the direction of the current input. We demonstrate that the LF provides significant smoothing compared with an unsmoothed KF, without any computational cost in training and little additional cost during decoding. The LF also introduces a much smaller delay in movement than an LPF.

The rest of this paper is organized into five sections. Methods and algorithms for decoding and analysis are described in Section II. Data collection and experimental designs are described in Section III. Experimental results are detailed in Section IV. This is followed by a discussion in Section V and concluding remarks in Section VI. Table I provides a list of mathematical symbols used throughout the paper.

II. METHODS

The movement intent decoding system used in this paper consists of several processing stages. First, features are created from input signals and used to create movement estimates. Next, estimates are smoothed, processed with a thresholding procedure that removes small amplitude position estimates, (as estimates are frequently inaccurate near rest positions,) and limited to each DoF range. The limited position estimate serves as the control input to the prosthesis.

A. Smoothing Methods

1) Single-Pole Low-Pass Filter: We use single-pole LPFs as a baseline for comparison. They are described by

$$\hat{x}_{p,n}^{smooth} = a\hat{x}_{p,n-1}^{smooth} + (1-a)\hat{x}_{p,n}, \tag{1}$$

where $a \in [0,1]$ is a parameter that controls the level of smoothing. The larger the value of a, the higher the level of smoothing, with $a = 1$ preventing any change in $\hat{x}_{p,n}^{smooth}$, and $a = 0$ resulting in no smoothing. We refer to a single-pole low-pass filter with $a = c$ as LPF_c throughout this paper.

2) Latching Filter: Low-pass filters are effective smoothers, but they repress intended quick changes in positional estimates as much as they do jitter. To increase the speed of the decoder while still reducing jitter, a time-varying and nonlinear modification can be made to (1) as follows:

$$\hat{x}_{p,n}^{smooth} = \gamma_{p,n}\hat{x}_{p,n-1}^{smooth} + (1-\gamma_{p,n})\hat{x}_{p,n}. \tag{2}$$

The function $\gamma_{p,n} \in [0,1]$ controls the amount the output of the latching filter (LF), $\hat{x}_{p,n}^{smooth}$, is able to vary from one sample to the next. Because each degree of freedom (DoF) of the hand is controlled independently, a different $\gamma_{p,n}$ is used for each DoF. The value of $\gamma_{p,n}$ is designed to vary in such a way that jitter is smoothed while intended movement is not smoothed. We make the simplifying assumption that small differences between $\hat{x}_{p,n-1}^{smooth}$ and $\hat{x}_{p,n}$ are more likely to be due to jitter than large differences, which are more likely due to intended movement. This is a reasonable assumption because jitter amplitude is usually small relative to the movement range and occurs more frequently than intended small movements. Using this assumption, $\gamma_{p,n}$ should be close to 1 if $|\hat{x}_{p,n-1}^{smooth} - \hat{x}_{p,n}|$ is small and close to 0 if $|\hat{x}_{p,n-1}^{smooth} - \hat{x}_{p,n}|$ is large. A thresholded quadratic function given by

$$\gamma_{p,n} = \max(1 - C[\hat{x}_{p,n-1}^{smooth} - \hat{x}_{p,n}]^2, 0), \tag{3}$$

where $C \in [0,\infty)$ is a user-defined tuning parameter, meets these requirements. Small values of C provide high levels of smoothness, but make quick movements more difficult, while large values of C provide less smoothing but allow for quicker movements. Similar nonlinear filters have been used in image processing [28]. Akin to our notation for lowpass filters, we will refer to a latching filter with $C = c$ as LF_c.

Theorem 1. *The LF is bounded-input, bounded-output stable.*

Proof. First we note that $|\hat{x}_{p,n}^{smooth}| \leq \max(|\hat{x}_{p,n-1}^{smooth}|, |\hat{x}_{p,n}|)$. If $|\hat{x}_{p,n-1}^{smooth}|$ is finite and greater than $|\hat{x}_{p,n}|$, $|\hat{x}_{p,n}^{smooth}| \leq |\hat{x}_{p,n-1}^{smooth}|$. That is, $\hat{x}_{p,n}^{smooth}$ can increase in magnitude only when $|\hat{x}_{p,n}|$ is larger than $|\hat{x}_{p,n-1}^{smooth}|$. It follows immediately that if $|\hat{x}_{p,0}^{smooth}| > |\hat{x}_{p,0}|$, $|\hat{x}_{p,n}^{smooth}| \leq \max\left\{|\hat{x}_{p,0}^{smooth}|, |\hat{x}_{p,k}|; k = 1,2,\cdots,n\right\}$. If $|\hat{x}_{p,0}^{smooth}| \leq |\hat{x}_{p,0}|$, $|\hat{x}_{p,n}^{smooth}| \leq \max\left\{|\hat{x}_{p,k}|; k = 1,2,\cdots,n\right\}$. Together, these statements imply that the LF is bounded-input, bounded-output stable as long as it is initialized using finite values. □

As the output of the LF approaches its input, $\gamma_{p,n}$ approaches 1, leading to a slowing of the convergence behavior. Because of this, the user of a prosthetic device needs to "push" the input to the LF past a target position to achieve that target position quickly. Users tend to use this strategy intuitively when given visual feedback of the prosthetic position.

TABLE II
KALMAN FILTER UPDATE EQUATIONS

$\mathbf{P}_n^- = \mathbf{A}\mathbf{P}_{(n-1)}^+\mathbf{A}_n^T + \mathbf{Q}_n$
$\mathbf{S}_n = \mathbf{R} + \mathbf{H}\mathbf{P}_n^-\mathbf{H}^T$
$\mathcal{K}_n = \mathbf{P}_n^-\mathbf{H}^T\mathbf{S}_n^{-1}$
$\mathbf{P}_n^+ = (\mathbf{I} - \mathcal{K}_n\mathbf{H})\mathbf{P}_n^-(\mathbf{I} - \mathcal{K}_n\mathbf{H})^T + \mathcal{K}_n\mathbf{R}\mathcal{K}_n^T$

TABLE III
KALMAN FILTER TRAINING EQUATIONS

$\mathbf{H} = \mathbf{Z}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$
$\mathbf{R} = \frac{1}{N}(\mathbf{Z} - \mathbf{H}\mathbf{X})(\mathbf{Z} - \mathbf{H}\mathbf{X})^T$
$\mathbf{A} = \mathbf{X}_{*,[2,N]}\mathbf{X}_{*,[1,N-1]}^T(\mathbf{X}_{*,[1,N-1]}\mathbf{X}_{*,[1,N-1]}^T)^{-1}$
$\mathbf{Q} = \frac{1}{N-1}(\mathbf{X}_{*,[2,N]} - \mathbf{A})(\mathbf{X}_{*,[2,N]} - \mathbf{A})^T$

B. Kalman Filter

The Kalman Filter is a recursive estimation technique [29] that updates the current position estimates based on state and observation models. Each new state, $\hat{\mathbf{x}}_n$, is related to the previous state $\hat{\mathbf{x}}_{n-1}$ as

$$\hat{\mathbf{x}}_n = \mathbf{A}\hat{\mathbf{x}}_{n-1} + \mathcal{K}_n(\mathbf{z}_n - \mathbf{H}\mathbf{A}\hat{\mathbf{x}}_{n-1}). \quad (4)$$

The various matrices in (4) are defined in Tables II and III. As a simplifying feature, each state is normalized to the range of motion for the DoF.

The computational complexity of the KF can be reduced by using the steady-state Kalman gain, $\mathcal{K}$, in place of $\mathcal{K}_n$ [30]. The steady-state Kalman gain is found during training by iterating the KF update equations in Table II, until the Kalman gain converges. Convergence occurs when the absolute value of all elements of $\mathcal{K}_n - \mathcal{K}_{n-1}$ are smaller than some tolerance (10^{-6} in this paper). In our experimental work, the standard KF converged to a steady-state KF after a few iterations.

The amplitude of $\hat{x}_{p,n}$ can be limited to prevent large excursions outside the range usable by the prosthesis. As explained in Section II-A2, we found it valuable to allow the KF outputs to exceed the prostheses' range of motion by a small amount.

C. Temporal Alignment of Movement to Features

When performing training trials, most subjects naturally introduce a delay between the movement command and their movement. Alternatively, subjects may anticipate a new command and begin moving before the movement command starts. These offsets can be corrected for in training by aligning the movements with the commanded movements. For all time shifts from $-\frac{2}{3}$s to $\frac{2}{3}$s, a least-squares regression was performed for each DoF over the entire window of training data. The shift that provided the least root-mean-square error (RMSE) across all DoFs was chosen as the optimal shift.

D. Feature Selection

Forward Selection (FS) with a Gram-Schmidt orthogonalization step was used to select features (adapted to continuous decoders from [31]). Table IV contains the equations and algorithm used for FS. To perform FS, a set of P movement residual vectors are initialized to the commanded movement, after removing the per-movement mean, and a set of M feature residual vectors are initialized to recorded feature vectors, after removing the per-feature mean. The residual vectors are centered as the FS algorithm assumes that the feature and movement vectors have zero-mean values. During each iteration of FS, correlation coefficients are found between movement residuals from each DoF and feature residuals of each feature. The feature residual that is most correlated with any single DoF is found. The feature corresponding to this feature residual is added to the set of selected features. This feature residual is used to estimate the movement residual of each DoF, and the estimate is subtracted from the movement residual. Features are then orthogonalized in a similar fashion: each feature residual is estimated using the best feature residual, and that estimate is subtracted from each residual. The process is iterated until a prescribed number of features is selected. Specific information about the number of features used in this study can be found in Section III.

TABLE IV
FS-ALL ALGORITHM

$\mathbf{r}_p^0 = \mathbf{x}_{p,*},\ \forall p \in [1, P]$	Initialize DoF residuals
$\mathbf{s}_m^0 = \mathbf{z}_{m,*},\ \forall m \in [1, M]$	Initialize feature residuals
$\rho_{m,p} = \frac{\mathbf{z}_{m,*}\mathbf{r}_{p,*}^T}{\sqrt{\mathbf{z}_{m,*}\mathbf{z}_{m,*}^T\mathbf{r}_{p,*}\mathbf{r}_{p,*}^T}}$	Find correlation coefficient
$\zeta^i = \text{argmax}_m\, \rho$	Find index of "best" feature
$\mathbf{r}_{p,*}^{i+1} = \mathbf{r}_{p,*}^i - \frac{\mathbf{r}_{p,*}^i\mathbf{s}_{\zeta^i}^{iT}}{\mathbf{s}_{\zeta^i}^i\mathbf{s}_{\zeta^i}^{iT}}\mathbf{r}_{p,*}^i$	Update movement residuals
$\mathbf{s}_{m,*}^{i+1} = \mathbf{s}_{m,*}^i - \frac{\mathbf{s}_{m,*}^i\mathbf{s}_{\zeta^i}^{iT}}{\mathbf{s}_{\zeta^i}^i\mathbf{s}_{\zeta^i}^{iT}}\mathbf{s}_{\zeta^i}^i$	Orthogonalize features

E. Thresholding

It is common for movement decoders to induce cross-movement while estimating intended movement of one or more DoFs from their rest positions, *i. e.*, to produce non-zero movement estimates in DoFs that were meant to be stationary. Further, noise can cause movement when no movement was intended. To reduce these effects, position estimates are modified by a dead zone algorithm that set all estimates with an amplitude less than T to 0 (rest). The threshold T takes a value between 0 and 1, and has the same normalized units as position estimates. Estimates are then shifted (using the numerator in the equation below) and scaled (using the denominator in the equation below) to fill the entire movement range as

$$\hat{x}_{p,n}^{thresh} = \text{sign}(\hat{x}_{p,n}^{smooth})\frac{\max(0, |\hat{x}_{p,n}^{smooth}| - T)}{1 - T} \quad (5)$$

where $\hat{x}_n^{thresh}$ is the output sent to the prosthetic hand, and $\text{sign}(\hat{x}_{p,n}^{smooth})$ is the sign of $\hat{x}_{p,n}^{smooth}$. If $\hat{x}_n^{thresh}$ exceeds 1 in magnitude, it should be limited to the usable range of [-1,1] before being sent to the prosthetic. See II-A2 for a discussion of why the magnitude may exceed the usable range. The value used for T in our experiments is described in Section III.

III. Experimental Background

A. Data Collection

The collection of data for this study from consenting human volunteers was approved by the University of Utah Institutional Review Board, the Salt Lake City Veterans Affairs Hospital Research and Development Service Center, and the Department of the Navy Human Research Protection Program as IRB 55621 that expires 27-April-2021.

A transradial amputee, subject HS3 was implanted with a 32-channel intramuscular EMG array (iEMGs; Ripple LLC, Salt Lake City, UT, USA), arranged as eight data leads with four electrodes per lead and a ninth lead with ground and reference electrodes. HS3 was also implanted with three Utah slanted electrode arrays (USEAs, Blackrock Microsystems, Salt Lake City, UT, USA) [32], two in the ulnar and one in the median nerve. Each USEA has 100 electrodes arranged as a 10×10 grid, 96 of which provided recordings. One of the ulnar arrays was not used because hardware limitations prevented recording from more than 2 arrays simultaneously. Each array (iEMG or USEA) were wired to an external Gator Connector Board (Ripple LLC) via a percutaneous incision. All possible combinations of the 32 iEMG electrodes were used to create differential pairs, yielding 496 pairs. These pairs, in conjunction with the 32 original single-ended electrodes formed 528 raw EMG data channels. Each raw data channel was recorded (single-ended channels) or calculated (differential pairs) at 1 kHz, filtered with a 4th-order Butterworth bandpass filter with passband in the range 15 to 350 Hz, rectified, and the mean of the preceeding 300 ms of rectified data was calculated every 33 ms, yielding the Mean Absolute Value (MAV) feature. Data from two of the USEA devices implanted in HS3 was recorded at 30 kHz. These data were bandpass filtered from 250 to 7500 Hz, using a Butterworth high-pass filter (4th-order) and a lowpass filter (3rd-order). Spike events were detected in these filtered data when voltage exceeded a channel-specific threshold. To calculate the threshold, the standard deviation of every 33 ms of data was computed. A running average of 33 s of these standard deviations, multiplied by 5, was used as the threshold. Spikes were detected in each 33 ms of data by finding negative voltage excursions from the mean that exceeded the spike detection threshold in amplitude. Every 33 ms, a 300 ms running average of these spike rates was computed for each neural channel and used as a feature in conjunction with the EMG features. This resulted in 192 neural features, 96 from each USEA.

HS3 is a male, and was 48 years old at the time of the study. He elected for amputation due to long-term chronic regional pain syndrome (CRPS), which inhibited use of his hand. He was implanted with the EMG electrodes and USEAs during the amputation surgery. The implant duration for HS3 was 17 months, and he was explanted at his discretion.

We also collected surface EMG data from five male non-amputee (NA) subjects, NA1-NA4 and NA6, aged 25-41 years. Data was collected using a 32 surface electrode sleeve made of neoprene with 1.5 cm-diameter circular metal electrodes intended to lie on the surface of the forearm. These electrodes were distributed across the various muscle groups of the forearm. Processing of the signals was the same for surface and intramuscular EMG signals.

In the experiments, the volunteers were shown an animated virtual hand [33]. During the training period, the virtual hand performed movements that the volunteers were asked to follow with their phantom limb (HS3) or native hand and wrist (NAs). Each movement trial consisted of moving one or more DoFs from rest to a maximum (e.g. fully flexed) or minimum (e.g. fully extended) position and back to rest. Data corresponding to these movements and target positions (with 0 as the rest position, and 1 as the position of maximal deviation) were recorded. For each of the NA subjects, 4 or 5 trials of anatomically defined movements in the flexion and extension plane were performed individually on each of 6 DoFs (thumb, index finger, middle finger, ring finger, little finger, and wrist). Similar training sets were acquired for HS3, but only the DoFs corresponding to the 5 digits were used.

Forward selection, described in Section II-D, was used to select 48 features (48 was determined an effective number of feature in previous experiments). A decoder was trained using the training data collected per the description above and calculated by the equations in Section II-B. Each subject was given control of a virtual hand, and asked to perform two different tasks. In each of these tasks, a transparent virtual hand was used to give the subject a target, while a solid hand indicated the subject's decoded position. The steady-state Kalman filter outputs were limited to a maximum amplitude of 1.1 (as described in Section II-B). The threshold, T (described in Section II-E) was set to 0.2. These methods and settings are consistent with our practice of many years [19].

In the "hold task," subjects were asked to move from rest position and hold a target position on one DoF, while holding other DoFs at rest. The target was placed midway between a rest position and the maximum deviation from rest. Each trial was 10 seconds long. Trials were performed for each of the movement types used in training. The subjects were asked to maintain all DoFs in target zones for as long as possible during each trial. The purpose of the hold task was to provide insight into the ability of a smoothing algorithm to maintain a steady estimate of a position with low jitter.

In the "speed task," the targets were placed in the same locations as the hold task, with a single DoF away from rest at a time. The subjects were requested to reach the targets as fast as possible, disregarding any overshoot of the targets that occurred. Subjects were also informed that movement of DoFs other than the single DoF being tested would not be penalized. The purpose of this test was to determine how quickly a decoder could transition from one position to another. This test was intended to approximate a practical use case in which precise control of DoFs is not required, but speed is beneficial. An example of such a use is quickly grabbing sturdy objects.

While movement positions were recorded in a range from -1 to 1, this information was transformed into degrees before being sent to the virtual hand. The normalized range from -1 to 1 varies from DoF to DoF in the range of degrees it represented, because different joints have different ranges of motion in degrees. Furthermore, within a single DoF, the range of motion of the flexion movement and extension movement

TABLE V
RANGE, TARGET WIDTHS, AND THE RATIO OF TARGET WIDTH TO RANGE OF THE VARIOUS DOFS IN DEGREES

DoF	Range (°)	Target Width (°)	Width / Range
Thumb	70	2.8	0.040
Index	90	3.2	0.035
Middle	90	2.8	0.031
Ring	90	2.8	0.031
Pinky	90	2.0	0.022
Wrist	110	1.5	0.014

can differ in terms of degrees. The normalized range for a movement type was always the same, 0 to ± 1.

The transparent hand was slightly larger than the solid hand, so a certain range of values for each DoF could be considered "in the target." Target widths were estimated for each DoF by visual inspection. The offsets between the transparent and solid hands for which the solid hand visibly extruded from the transparent hand varied from DoF to DoF in terms of normalized range and degrees. In order to account for subject difficulty in seeing the solid hand emerging near the edge of the target zone, an extra 0.2 degrees was padded on each side of each target zone when calculating metrics based on the target zone. Exact values for the width, in degrees and normalized range, of each target on each DoF are shown in Table V. The range of motion for each DoF and the ratio of the target width to this range of motion are also shown.

Three types of smoothing algorithms were used to smooth the Kalman decoder output: the LPF, LF, and no smoothing. For NA subjects, the filters used were $\text{LPF}_{0.93}$, $\text{LPF}_{0.97}$, $\text{LF}_{0.5}$, LF_1, $\text{LF}_{6.25}$ and no smoothing. For HS3, the filters used were $\text{LPF}_{0.97}$, an LF_1 and no smoothing. Weakened muscles from CRPS and arm disuse as well as residual pain deterred HS3's performance in prosthetic control. For this reason, fewer filters were tested with HS3 than the NA subjects. The C values for the LF were chosen to allow different levels of smoothing and speed by online experimentation in a pilot study. The values of a for the LPF were chosen experimentally as well, but with some theoretical considerations. A single-pole low-pass filter is equivalent to a moving average filter with an exponentially decaying window. The time constant of this exponentially decaying window is $1/(F_s(1-a))$, where F_s is the sampling rate. For the F_s=30 Hz MAV or spike rate signal (see Section III for signal processing details) used in our experiments, the time constant of the $\text{LPF}_{0.93}$ is approximately 1/2 s and that of the $\text{LPF}_{0.97}$ is approximately 1 s.

During each of three data recording sessions for the NA subjects, a series of six trials was performed sequentially for each movement type for each of the two tasks, one trial for each smoothing type, *i.e.*, 72 trials per session. For each of HS3's two data recording sessions, six trials of each movement type were conducted for each of the tasks, two trials using each smoother, equivalent to 60 trials per data set. Two trials were conducted for each movement type and smoothing method for HS3 because his performance was more inconsistent than that of the NA subjects, and acquiring more data improved the likelihood of the results accurately representing his overall performance. The order in which the smoothing filters were used in the data recording sessions was randomized, and the subject was blind to which method was used at each trial.

B. Performance Metrics

Two metrics were calculated for each trial of the hold task. The first, the hold time, is the maximum amount of time a subject was able to continuously hold all DoFs within the target range. The maximum hold time was the length of the trial, 10 s. The second metric measures the smoothness of the moving DoF by calculating the log mean absolute jerk (LMAJ) [34]. defined as follows: Let

$$\Delta \mathbf{y}^T = [y_1 - y_0, y_2 - y_1, ..., y_N - y_{N-1}]^T. \quad (6)$$

where N is the number of time samples. The jerk vector $\mathbf{j}$ of a smoothed, thresholded movement estimate vector $\hat{\mathbf{x}}_{p,*}^{thresh}$ is defined as

$$\mathbf{j}_{p,*} = F_s^3 \Delta\Delta\Delta \hat{\mathbf{x}}_{p,*}^{thresh}. \quad (7)$$

where F_s is the sampling rate. The LMAJ is calculated as

$$LMAJ = 3\log(F_s) + \log(\frac{1}{N-3}\sum_{n=1}^{N-3}|j_{p,n}|), \quad (8)$$

where $j_{p,n}$ is the nth element of the jerk vector for the pth DoF. The LMAJ was calculated over the window beginning with the first incidence where all DoFs were inside target ranges and ending with the end of the trial. The LMAJ was only calculated for the DoF with a target position not at rest. This avoided penalizing a subject attempting to move other DoFs to a rest position should cross-movement occur.

A single metric was calculated for each trial of the speed task, the time to target (TTT). This is the amount of time required to move the requested DoF from rest into the target region. This metric is used to measure the relative delays introduced by the smoothing filters in the speed test.

During each experimental session, HS3 performed two trials of each movement with each smoothing filter. The mean value for each movement type and smoothing filter is reported in place of trial-wise metrics, due to our conservative view that trials from the same movement type and experimental session are not independent. The NA subjects performed one trial of each movement type using each smoother for each dataset, so trial-wise metrics are reported for NA subjects.

C. Statistical Analyses

The relative performance of the smoothing algorithms was compared by performing statistical analyses on the three metrics just introduced. A separate test was performed for each metric. Individual, univariate statistical tests were performed on NA subjects' and HS3 subject's data separately.

For each of the three metrics, the data were partitioned into six groups for the NA subjects and into three groups for HS3, with each group corresponded to a single smoothing filter tested. Each group of NA data included 12 movement types from 15 experimental sessions, or 180 observations in total. Each group of the HS3 data involved 10 movement types and 2 data collection sessions, or 20 observations in total.

One-way analysis of variance (ANOVA) was performed to determine if differences existed between groups. If ANOVA resulted in a p-value of 0.05 or smaller, a post-hoc test using Tukey's honest significant difference was performed on all possible pairings to determine which groups differed. Here, we consider a p-value of 0.05 to be significant. These tests were performed using `anova1` and `multcompare` in MATLAB.

Occasionally, subjects were unable to perform a task in the alloted time. In the case of the hold time metric, we simply considered the hold time for that trial to be 0 seconds. In the case of the other metrics, the value was considered missing. Missing values were represented as "NaN" in MATLAB.

IV. Results

The hold task and the speed task were designed to test the smoothing ability and responsiveness of the various smoothing filters. Hold time and LMAJ were used to measure smoothness of trials of the hold task, whereas TTT was used to determine the ability to move quickly during the speed task. Representative data from one subject (NA1) demonstrates many of the good and bad aspects of the smoothing algorithms (Fig. 1).

The hold task trials illustrate the function of the smoothing filters (Fig. 1a and 2). The LF smooths by sharply reducing changes to decoded position when in the target range, effectively "latching" onto a position (and a value of γ). The $\text{LF}_{0.5}$ appears to latch most strongly, followed by the LF_1. The $\text{LF}_{6.25}$ does not appear to latch as well as the other LFs, although brief periods of latching are seen in the sample trial. The LF appears to need a larger change in input to break away from a latched value when using smaller values of C (Fig. 2). The LPFs do not latch but instead slow changes to their output. The $\text{LPF}_{0.97}$ appears to slow and smooth more than the $\text{LPF}_{0.93}$. The unsmoothed KF appears more choppy than any LF or LPF.

The speed task trials reveals the delay associated with the smoothing filters (Fig. 1b). As expected, the "no smoothing" filter was faster than all other filters. In general, the amount of delay within a filter type appears to scale with the level of smoothing. Across all the filters, the LF_1 provided a high level of smoothing with modest increase in delay.

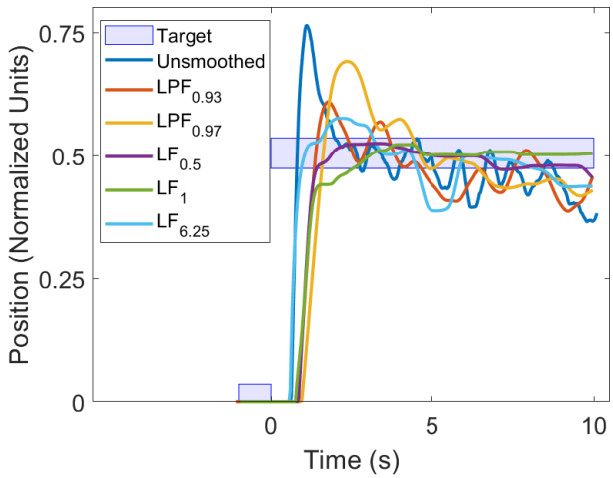
To validate these initial observations, all metrics were extracted for all trials within a class of subjects.

The summary data for hold time show clear differences in performance between the types of filters for the NA subjects (Fig. 3), and this result was found significant (ANOVA, $p < 0.001$). The $\text{LF}_{0.5}$ and LF_1 yielded the best hold times. Post-hoc tests found no significant differences between these two filters ($p = 0.74$), but significant differences were found between either of these two and all other filters ($p < 0.001$). The $\text{LPF}_{0.97}$, had hold times similar to those of the $\text{LF}_{6.25}$, (as indicated by a non-significant difference, $p = 0.25$). In other words, of the smoothing filters tested, the LPF with strongest smoothing abilities had similar performance as the LF with weakest smoothing abilities. The hold time for the unsmoothed KF was significantly less than that by any of the smoothed filters ($p < 0.001$, except in comparison to $\text{LPF}_{0.93}$ where $p = 0.03$). That is, even the worst performing smoother, $\text{LPF}_{0.93}$, yielded longer hold times than the unsmoothed KF. These results indicate that the LF methods, especially those with small C values, give users improved ability to latch onto position ranges within the target zone. In contrast, the LPF methods give users lesser latching abilities, although they still provide smoothing which increases hold times compared to an unsmoothed KF. This type of behavior was expected, since the LFs are designed to strongly resist small changes in estimated position, but allow larger changes.

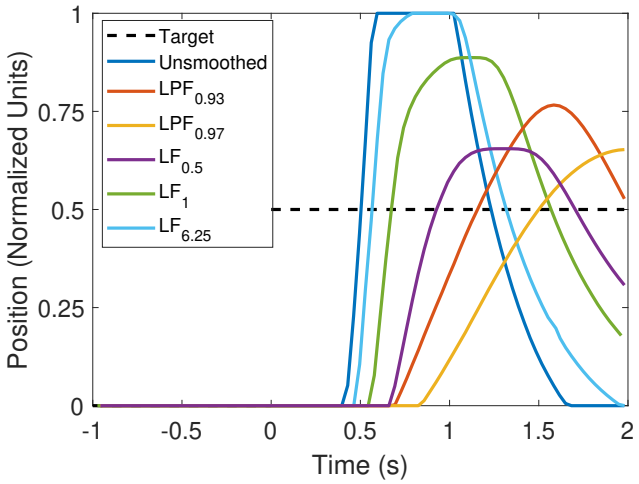
Similarly, the hold time summary data show differences in performance between the types of filters for HS3 (Fig. 4), and this result was found significant (ANOVA, $p = 0.012$). For the LF_1, a significant difference was found in comparison to the unsmoothed KF ($p = 0.01$), while for the $\text{LPF}_{0.97}$ the difference was not significant ($p = 0.48$). There appears to be a trend for slightly longer hold times with the LF than the LPF, but this difference was not significant ($p = 0.15$). Perhaps, given more data, this difference might become significant, however, collection of additional data is not possible as HS3 selected to be explanted.

The LMAJ metric directly quantifies the "smoothness" of a filter, and differences in performance with this metric were observed between the filter types for both the NA subjects (Fig. 5) and HS3 (Fig. 6). For NA subjects, the differences were found to be significant (ANOVA, ($p < 0.001$). Post-hoc tests found significant differences between all possible pairs of smoothing filters ($p < 0.001$), with the exception of the comparison between $\text{LPF}_{0.93}$ and the $\text{LF}_{6.25}$ ($p = 0.98$). The $\text{LF}_{0.5}$ had the lowest LMAJ, due to its strong position latching capabilities. The LF_1 also produced a low LMAJ, but its LMAJ was significantly higher than $\text{LF}_{0.5}$. This contrasts with the hold time metrics for these two smoothers, which were not found to be significantly different. This suggests that the $\text{LF}_{0.5}$ reduces movement jitter compared to the LF_1, but doesn't necessarily improve the ability to maintain positions inside target ranges. This result might not hold with smaller target sizes, as higher jitter in LF_1 implies more movement, albeit inside the target range. Another example where a lower LMAJ doesn't necessarily imply a higher hold time is the comparison between $\text{LPF}_{0.97}$ and $\text{LF}_{6.25}$. The $\text{LPF}_{0.97}$ has a significantly lower mean LMAJ than the $\text{LF}_{6.25}$, but the hold times were not found to be significantly different. On the other hand, the $\text{LFP}_{0.93}$ was not found to be significantly different from $\text{LF}_{6.25}$ in terms of LMAJ, but the $\text{LF}_{6.25}$ had significantly longer hold times. These inconsistencies between hold times and LMAJ are due to the ability of the LF to latch to a certain position, causing long hold times, but create fast movements when not in a latched position, creating higher LMAJ. The unsmoothed KF exhibited larger LMAJ than the other methods.

For HS3 LMAJ data, significant differences among the smoothing methods were found (ANOVA, $p < 0.001$). Post-hoc tests found that the $\text{LPF}_{0.97}$ and LF_1 were both significantly different from the unsmoothed KF ($p < 0.001$), but not from each other ($p = 0.59$). These results contrast with those of the NA subjects, as the $\text{LPF}_{0.97}$ and LF_1 were found to be significantly different with NA data. Due to challenges associated with his recent amputation, the HS3 subject had more difficulty maintaining low jitter estimates than any of the NA subjects. This led to temporally shorter latches for HS3 than



(a) Representative trials of the Hold Task



(b) Representative trials of the Speed Task

Fig. 1. Sample thumb flexion testing trials for (a) the hold test and (b) the speed test. The example trials were taken from NA3. All of the smoothing algorithms appear to reduce "chatter" in the unsmoothed estimate but result in a delay in rapidly reaching a target. For each test, the thumb started at a neutral position and the subject attempted to move towards and hold it in the target range (hold test) or move into the range as quickly as possible (speed test).

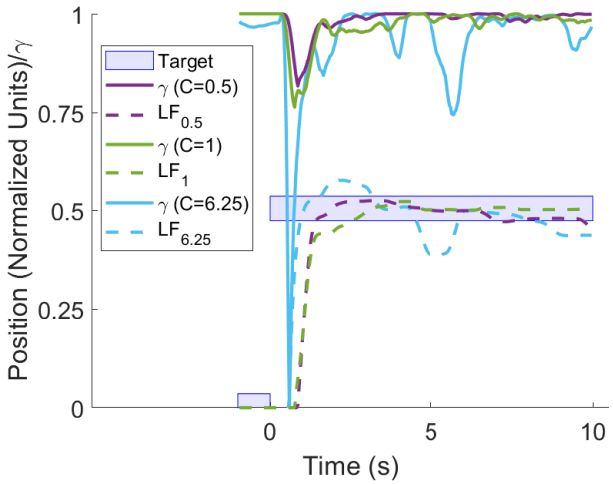


Fig. 2. Sample γ values for the LF traces shown in Fig. 1a. The solid lines show the values for γ and the dashed lines show the output of the LFs. The target range is shown as a shaded box.

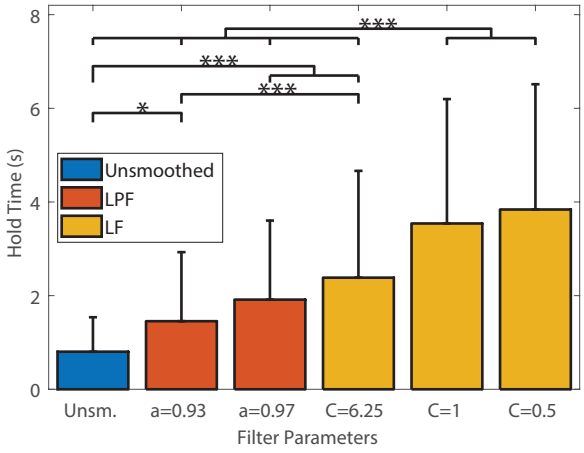


Fig. 3. Mean and standard deviation of hold times for NA subjects. Both smoothing approaches (LPF and LF) improved (increased) hold time, and modifying parameters to increase smoothing (larger a and smaller C) increased hold time. In this figure and all others, significant differences between groups are shown using * for $p \leq 0.05$, ** for $p \leq 0.01$, and *** for $p \leq 0.001$.

the NA subjects. There is also less HS3 data from which to draw statistical conclusions. Because the LF supplies different amounts of smoothing when latched than when moving, the decrease in latching drove the mean LMAJ for LF_1 higher for HS3, resulting in LMAJ values more similar to those of $LPF_{0.97}$. Unlike with NA data, the LMAJ and hold time results were not contradictory. Differences between $LPF_{0.97}$ and LF_1 were not found to be significant, but differences between either method and the unsmoothed KF were.

The TTT metric indicated that all degrees of smoothing tested led to slower movements (Fig. 7 and Fig. 8). For NA subjects, significant differences were found among the various smoothing filters (ANOVA, $p < 0.001$). Post-hoc tests, however, found no significant differences between the fastest method, the unsmoothed KF, and the $LF_{6.25}$ ($p = 0.1$).

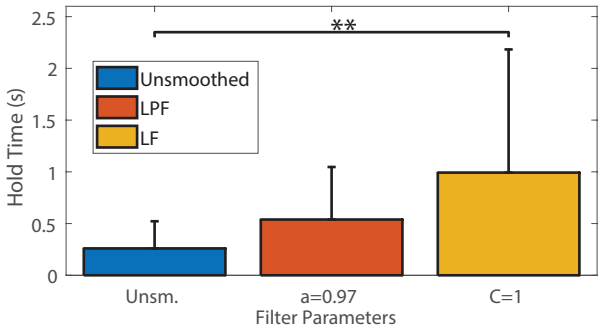


Fig. 4. Mean and standard deviation of hold times for HS3. As with NA subjects, the smoothing filters increased hold time, but only the difference between the unsmoothed KF and the LF_1 was found significant here.

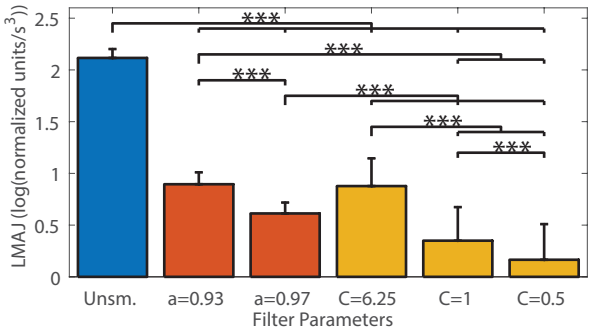


Fig. 5. Mean and standard deviation of LMAJ for non-amputee subjects. Both smoothing approaches (LPF and LF) improved (reduced) LMAJ, and modifying parameters to increase smoothing (larger a and smaller C) decreased LMAJ. Across all comparisons, only the comparison between $LPF_{0.93}$ and $LF_{6.25}$ was not significantly different. See Fig. 3 for legend.

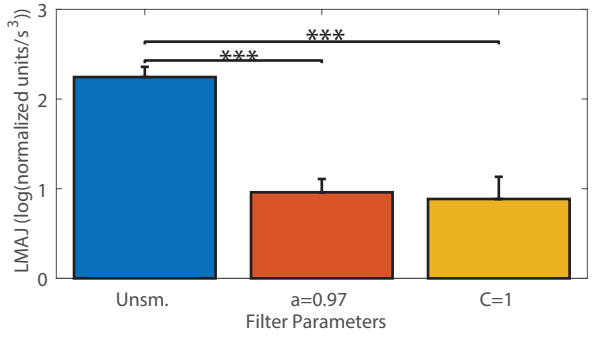


Fig. 6. Mean and standard deviation of LMAJ for amputee subject. Similar to NA subjects, smoothing filters reduced LMAJ. See Fig. 3 for legend.

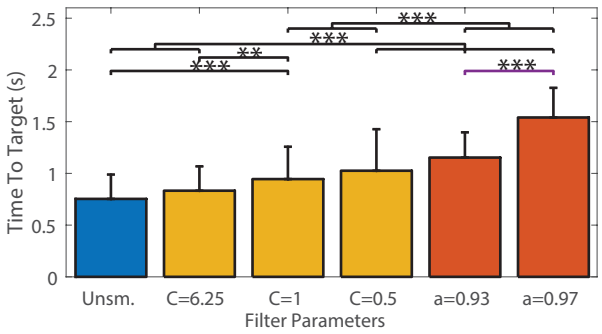


Fig. 7. Mean and standard deviation of time to target values for NA subjects. As expected, application of a smoothing filter degraded (increased) time to target, and modifying parameters to increase smoothing (larger a and smaller C) degraded performance. See Fig. 3 for legend.

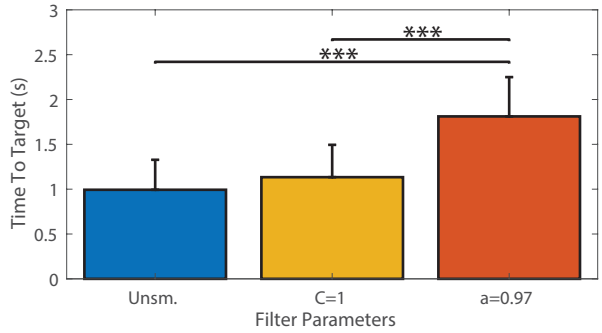


Fig. 8. Mean and standard deviation of time to target values for HS3. TTT increased with any smoothing filter but the change was not significant between the unsmoothed KF and LF_1. See Fig. 3 for legend.

Noting this and the previously reported hold times, it can be seen that the $LF_{6.25}$ provides the same target "holding" ability as $LPF_{0.97}$, the smoothest LPF tested, without a significant increase in TTT compared to the unsmoothed KF. When speed is very important, $LF_{6.25}$ is a good option to provide some smoothing without greatly reducing movement speed. Both the unsmoothed KF and $LF_{6.25}$ were found to be significantly different than all other methods ($p \leq 0.0036$).

The other LFs, $LF_{0.5}$ and LF_1, added a delay compared to the unsmoothed KF, but were not found to be significantly different from each other ($p = 0.09$). These filters produced significantly smaller TTT values than either LPF ($p < 0.01$). When considering this in conjunction with target hold times, either $LF_{0.5}$ or LF_1 provided high hold times for relatively small increases in TTT. When hold times are a primary consideration and movement speed is a secondary consideration $LF_{0.5}$ or LF_1 are both good options.

For HS3, significant differences were found among the TTT metrics of the smoothing filters (ANOVA, $p < 0.001$). The unsmoothed KF and LF_1 were the fastest methods, but were not significantly different ($p = 0.48$). Both were found significantly different from $LPF_{0.97}$ ($p < 0.001$). For HS3, LF_1 provided the same level of smoothing as $LPF_{0.97}$, both in terms of LMAJ and hold times, but without a significant impact on TTT metrics compared to the unsmoothed KF.

V. Discussion

The latching filter is an important step in allowing quick movements, while reducing jitter in movement estimates. The hold and speed tests together demonstrated that the LF could simultaneously provide improved levels of smoothness and quicker movement transitions, in comparison to a LPF. Compared to the unsmoothed case, the LF significantly improved hold times with minimal and, and for some parameterizations, statistically insignificant impact on decoder speed. The latching filter with $C = 1$ provided a high-level of smoothing, as measured by both hold times and LMAJ, with a minor increase in time to target (around 0.2 seconds) compared to the unsmoothed KF, and is our recommended smoothing method.

The delay in performing the speed task was appreciable for all smoothing algorithms, including the unsmoothed KF (Fig. 7 and 8). This delay comes from three sources: detection response task (DRT) delay, movement initiation, and movement performance delay. In daily activities, the DRT delay is negligible as the user knows the task. In this study, subjects had substantive DRT delay because they had to recognize the movement commands. DRT studies have shown that delays around 0.5 s are common [35]. On average, this delay and the movement initiation delay are the same for all smoothers examined herein. Only the movement performance delay will differ when smoothing is applied, and this increase maybe statistically significant. The amount of delay which is

considered acceptable likely depends on the subject and the task at hand, and users of prosthetic devices may be given the option of choosing what level of smoothing they prefer.

We speculate that the LF provides relative ease in maintaining a desired position compared to the unsmoothed KF. For example, once a desired position is reached, a user can relax somewhat and that position will be maintained due to its "latching" ability. In contrast, when using an unsmoothed decoder or an LPF, continuous effort is required to maintain a position, which may lead to user fatigue. The LF requires a user to "push" past a target to reach that target in a reasonable amount of time, but in our experience, users quickly learn and adapt to that need. To reach its target position quickly, the LF relies on its user having feedback, to know when to correct slow transitions in real-time by "pushing" past a target position and when the user can relax, while the LF output maintains the DoF in the desired range. In addition, the outputs used to drive the LF must allow estimation of positions outside the constraints of a prosthetic device's range of motion, enabling a user to expediently reach full range of motion. Interestingly, the latching filter's characteristics actually make it easier for a user to maintain maximal positions. Without the latching filter, users trying to reach a extreme value tend to aggressively push the decoder past its "stops" to avoid the occasional transition from the extremum. This leads to unnecessary fatigue. With the latching filter, the user can actually relax a bit at the extremes and not be worried about maintaining that position.

The nature of the LF, where it pushes to a position and then slightly relaxes, is similar to the characteristics of the biological data recorded during training sessions. When moving a DoF from a neutral position and then holding it in place at another location, the maximum amplitude of the EMG signals during the transition is frequently larger than the maximum amplitude during the hold period (Fig. 9). Here, NA1 was asked to move his index finger from a neutral position to a position of maximal deviation, hold that position for three seconds, and then relax. The EMG amplitude reached a maximum around the end of the transition period, then rapidly dropped to a lower value and slowly decreased during the hold. When using a KF or regression-based decoders, this behavior may cause a movement estimate to overshoot a target, and has lead to our investigating performance improvements by increasing the hold time during training [36].

The LF employing a quadratic nonlinearity showed significant gains over LPFs in our work. Given constraints of subject availability, we were not able to test other nonlinear functions. Although our selection of the quadratic nonlinearity is based upon previous practice [28], a different nonlinear function may provide additional performance improvement.

Due to time and experimental limitations, we were only able to test three parameterizations of the LF in intact subjects and only one parameterization in an amputee subject. It is possible that other parameterizations could provide better performance. It is also likely that the "optimal" value may differ between subjects and that different DoFs may have different "optimal" values of C. Given that examination of alternative values of the hyper-parameter (or use of different hyper-parameters per DoF) can only be done online, subject access limited our ability to fully investigate the space. Nevertheless, we anticipate providing the user some ability to adjust the latching filter's hyper-parameter in future studies with amputee subjects, and let the user investigate their own parameter space.

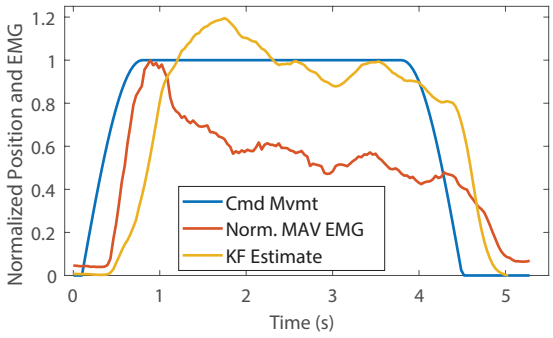


Fig. 9. The normalized commanded position (Cmd Mvmt) of a training trial, the corresponding EMG response and the unsmoothed KF estimate for subject NA1. The maximum-normalized mean MAV across all EMG channels (Norm. MAV EMG) and the KF estimate show rapid rise associated with the commanded movement and slow decrease when the command becomes constant. An LF accommodates this biological characteristic by letting rapid movements occur and reducing the effect of the slow drop off.

The tasks and metrics used to characterize jitter and speed gave valuable insights on the LF. However, additional experiments that measure performance in activities of daily living such as the Southampton hand assessment procedure (SHAP) [37] or the activities measure for upper limb amputees (AMULA) test [38] may provide additional insight into the usability of the LF, including effect of delays. Such functionality testing will be part of future studies with amputee subjects, while they use an arm prosthesis in their home environment.

The LF can be used in conjunction with any continuous decoder and perhaps with a biophysics-based feed forward decoder. Given it only acts upon the decoded estimates as they are sent to the arm prosthesis, it does not interact with any recursion that occurs with many continuous decoders (e.g., the Kalman filter and recurrent neural networks). It even allows one to switch between multiple decoders, as long at the meaning of the output signal remains unchanged. Its use with a low cross-movement decoder has the potential to bring functionality of prosthetic hands closer that of native hands.

VI. Conclusions

This paper presented the latching filter, a computationally simple method for smoothing intended movement estimates for use with prosthetic devices. The LF is so-named because it gives users the ability to "latch" onto a position and achieve a high level of smoothness, while also allowing for quick movements away from that position. In contrast, an LPF provides the same level of smoothing regardless of user intent, forcing a trade-off between unsatisfactory jitter and unsatisfactory movement speed. The LF provides high levels of smoothing, comparable to that provided by an LPF, while substantially increasing movement speed. The LF can be used to improve the reliability of prostheses, and may be able to help amputees perform difficult tasks more easily.

ACKNOWLEDGMENT

This work was supported by National Science Foundation Grants 1533649, 1901492 and 1901236 and by DARPA's HAPTIX program (Contract No. N66001-15-C-4017). We thank NVIDIA Corporation for the donation of a Tesla K40 GPU used in this research, and our volunteer subjects who provided the experimental data described in this paper.

REFERENCES

[1] D. J. Warren, S. Kellis, J. G. Nieveen, S. M. Wendelken, H. Dantas, T. S. Davis, D. T. Hutchinson, R. A. Normann, G. A. Clark, and V. J. Mathews, "Recording and decoding for neural prostheses," *Proceedings of the IEEE*, vol. 104, no. 2, pp. 374–391, Feb. 2016.

[2] G. Purushothaman, "Myoelectric control of prosthetic hands: State-of-the-art review," *Medical Devices: Evidence and Research*, vol. 9, pp. 247–255, Jul. 2016.

[3] R. L. Ortolan, R. N. Mori, R. R. Pereira, C. M. N. Cabral, J. C. Pereira, and A. Cliquet, "Evaluation of adaptive/nonadaptive filtering and wavelet transform techniques for noise reduction in emg mobile acquisition equipment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 1, pp. 60–69, Mar. 2003.

[4] J. Liu, D. Ying, and P. Zhou, "Wiener filtering of surface EMG with a priori SNR estimation toward myoelectric control for neurological injury patients," *Medical Engineering and Physics*, vol. 36, no. 12, pp. 1711–1715, Dec. 2014.

[5] C. M. Harris and D. M. Wolpert, "Signal-dependent noise determines motor planning," *Nature*, vol. 394, no. 6695, pp. 780–784, Aug. 1998.

[6] L. Citi, J. Carpaneto, K. Yoshida, K.-P. Hoffmann, K. P. Koch, P. Dario, and S. Micera, "On the use of wavelet denoising and spike sorting techniques to process electroneurographic signals recorded using intraneural electrodes," *Journal of Neuroscience Methods*, vol. 172, no. 2, pp. 294–302, Jul. 2008.

[7] M. Amirmazlaghani and H. Amindavar, "EMG signal denoising via Bayesian wavelet shrinkage based on GARCH modeling," in *IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Taipei, Taiwan, Apr. 2009, pp. 469–472.

[8] J. Maier, A. Naber, and M. Ortiz-Catalan, "Improved prosthetic control based on myoelectric pattern recognition via wavelet-based de-noising," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 506–514, Feb. 2018.

[9] G. Aschero and P. Gizdulich, "Denoising of surface EMG with a modified Wiener filtering approach," *Journal of Electromyography and Kinesiology*, vol. 20, no. 2, pp. 366–373, Apr. 2010.

[10] E. A. Clancy and K. A. Farry, "Adaptive whitening of the electromyogram to improve amplitude estimation," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 6, pp. 709–719, Jun. 2000.

[11] L. Liu, P. Liu, E. A. Clancy, E. Scheme, and K. B. Englehart, "Electromyogram whitening for improved classification accuracy in upper limb prosthesis control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 5, pp. 767–774, Sep. 2013.

[12] T. D. Sanger, "Bayesian filtering of myoelectric signals," *Journal of Neurophysiology*, vol. 97, no. 2, pp. 1839–1845, Feb. 2007.

[13] T. Pistohl, C. Cipriani, A. Jackson, and K. Nazarpour, "Abstract and proportional myoelectric control for multi-fingered hand prostheses," *Annals Biomed. Engineering*, vol. 41, no. 12, pp. 2687–2698, Dec. 2013.

[14] D. Hofmann, N. Jiang, I. Vujaklija, and D. Farina, "Bayesian filtering of surface EMG for accurate simultaneous and proportional prosthetic control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 12, pp. 1333–1341, Dec. 2016.

[15] J. Nieveen, Y. Zhang, S. Wendelken, T. Davis, D. Kluger, J. A. George, D. Warren, D. Hutchinson, C. Duncan, G. A. Clark, and V. J. Mathews, "Polynomial Kalman filter for myoelectric prosthetics using efficient kernel ridge regression," in *Int. IEEE/EMBS Conf. Neural Engineering*, Shanghai, China, May 2017, pp. 432–435.

[16] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a Kalman filter," *Neural Computation*, vol. 18, no. 1, pp. 80–118, Jan. 2006.

[17] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "A high-performance neural prosthesis enabled by control algorithm design," *Nature Neuroscience*, vol. 15, no. 12, pp. 1752–1757, Nov. 2013.

[18] J. Han, Q. Ding, A. Xiong, and X. Zhao, "A state-space EMG model for the estimation of continuous joint movements," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 7, pp. 4267–4275, Jul. 2015.

[19] J. A. George, T. S. Davis, M. R. Brinton, and G. A. Clark, "Intuitive neuromyoelectric control of a dexterous bionic arm using a modified Kalman filter," *J. Neuroscience Methods*, vol. 330, p. 108462, Jan. 2020.

[20] M. Xiloyannis, C. Gavriel, A. A. C. Thomik, and A. A. Faisal, "Gaussian process autoregression for simultaneous proportional multi-modal prosthetic control with natural hand kinematics," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1785–1801, Oct. 2017.

[21] T. Teban, R. Precup, E. Voisan, T. E. A. de Oliveira, and E. M. Petriu, "Recurrent dynamic neural network model for myoelectric-based control of a prosthetic hand," in *2016 Annual IEEE Systems Conference (SysCon)*, Orlando, Florida, USA, Apr. 2016, pp. 1–6.

[22] J. Liu, S. H. Kang, D. Xu, Y. Ren, S. J. Lee, and L.-Q. Zhang, "EMG-based continuous and simultaneous estimation of arm kinematics in able-bodied individuals and stroke survivors," *Frontiers in Neuroscience*, vol. 11, p. 480, Aug. 2017.

[23] P. Xia, J. Hu, and Y. Peng, "EMG-based estimation of limb movement using deep learning with recurrent convolutional neural networks," *Artificial Organs*, vol. 42, no. 5, pp. E67–E77, May 2018.

[24] X. Zhang, X. Li, O. W. Samuel, Z. Huang, P. Fang, and G. Li, "Improving the robustness of electromyogram-pattern recognition for prosthetic control by a postprocessing strategy," *Frontiers in Neurorobotics*, vol. 11, p. 51, Sep. 2017.

[25] D. Young, F. Willett, W. D. Memberg, B. Murphy, P. Rezaii, B. Walter, J. Sweet, J. Miller, K. V. Shenoy, L. R. Hochberg, R. F. Kirsch, and A. B. Ajiboye, "Closed-loop cortical control of virtual reach and posture using cartesian and joint velocity commands," *J Neural Eng*, vol. 16, no. 2, p. 026011, Apr. 2019.

[26] J. E. Downey, J. M. Weiss, K. Muelling, A. Venkatraman, J.-S. Valois, M. Hebert, J. A. Bagnell, A. B. Schwartz, and J. L. Collinger, "Blending of brain-machine interface and vision-guided autonomous robotics improves neuroprosthetic arm performance during grasping," *Journal of NeuroEngineering and Rehabilitation*, vol. 13, no. 1, p. 28, Mar. 2016.

[27] J. Ma, N. V. Thakor, and F. Matsuno, "Hand and wrist movement control of myoelectric prosthesis based on synergy," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 1, pp. 74–83, Dec. 2015.

[28] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*. Wiley, 2000.

[29] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, Mar. 1960.

[30] W. Q. Malik, W. Truccolo, E. N. Brown, and L. R. Hochberg, "Efficient decoding with steady-state Kalman filter in neural interface systems," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 1, pp. 25–34, Feb 2011.

[31] K. Z. Mao, "Orthogonal forward selection and backward elimination algorithms for feature subset selection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 1, pp. 629–634, Feb. 2004.

[32] A. Branner, R. B. Stein, and R. A. Normann, "Selective stimulation of cat sciatic nerve using an array of varying-length microelectrodes," *Journal of Neurophysiology*, vol. 85, no. 4, pp. 1585–1594, Apr. 2001.

[33] V. Kumar and E. Todorov, "MuJoCo HAPTIX: A virtual reality system for hand manipulation," in *IEEE Int. Conf. Humanoid Robots*, Toronto, Canada, Nov. 2015, pp. 657–663.

[34] N. Hogan and D. Sternad, "Sensitivity of smoothness measures to movement duration, amplitude, and arrests," *Journal of Motor Behavior*, vol. 41, no. 6, pp. 529–534, Nov. 2009.

[35] D. L. Strayer, J. Turrill, J. M. Cooper, J. R. Coleman, N. Medeiros-Ward, and F. Biondi, "Assessing cognitive distraction in the automobile," *Hum Factors*, vol. 57, no. 8, pp. 1300–24, Nov. 2015.

[36] J. A. George, M. R. Brinton, C. C. Duncan, D. T. Hutchinson, and G. A. Clark, "Improved training paradigms and motor-decode algorithms: Results from intact individuals and a recent transradial amputee with prior complex regional pain syndrome," in *Int. Conf. IEEE Engineering in Medicine and Biology Soc.*, July 2018, pp. 3782–3787.

[37] C. M. Light, P. H. Chappell, and P. J. Kyberd, "Establishing a standardized clinical assessment tool of pathologic and prosthetic hand function: Normative data, reliability, and validity," *Archives of Physical Medicine and Rehabilitation*, vol. 83, no. 6, pp. 776–783, Jun. 2002.

[38] L. Resnik, L. Adams, M. Borgia, J. Delikat, R. Disla, C. Ebner, and L. S. Walters, "Development and evaluation of the activities measure for upper limb amputees," *Archives of Physical Medicine and Rehabilitation*, vol. 94, no. 3, pp. 488–494.e4, Oct. 2013.