Shared Prosthetic Control Based on Multiple Movement Intent Decoders

Henrique Dantas*, *Member IEEE*, Taylor C. Hansen*, *Student Member IEEE*, David J. Warren, *Senior Member IEEE*, and V John Mathews, *Fellow IEEE*

***Abstract*—*Significance*: A number of movement intent decoders exist in the literature that typically differ in the algorithms used and the nature of the outputs generated. Each approach comes with its own advantages and disadvantages. Combining the estimates of multiple algorithms may have better performance than any of the individual methods.**

***Objective*: This paper presents and evaluates a shared controller framework for prosthetic limbs based on multiple decoders of volitional movement intent.**

***Methods*: An algorithm to combine multiple estimates to control the prosthesis is developed in this paper. The capabilities of the approach are validated using a system that combines a Kalman filter-based decoder with a multilayer perceptron classifier-based decoder. The shared controller's performance is validated in online experiments where a virtual limb is controlled in real-time by amputee and intact-arm subjects. During the testing phase subjects controlled a virtual hand in real time to move digits to instructed positions using either a Kalman filter decoder, a multilayer perceptron decoder, or a linear combination of the two.**

***Results*: The shared controller results in statistically significant improvements over the component decoders. Specifically, certain degrees of shared control result in increases in the time-in-target metric and decreases in unintended movements.**

***Conclusion*: The shared controller of this paper combines the good qualities of component decoders tested in this paper. Herein, combining a Kalman filter decoder with a classifier-based decoder inherits the flexibility of the Kalman filter decoder and the limited unwanted movements from the classifier-based decoder, resulting in a system that may be able to perform the tasks of everyday life more naturally and reliably.**

***Index Terms*—Kalman Filter, Movement Intent Decoders, Shared controllers, Neural Networks, Multilayer Perceptron**

I. Introduction

Interpreting movement intent from biological signals is a key component of limb neuroprostheses. Such prostheses are controlled by interpreting motor intent using biological signals including electromyograms (EMG), electroencephalograms (EEG), neuronal signals acquired via implanted electrode arrays in the cerebral cortex, and neuronal signals obtained via electrode arrays implanted in peripheral nerves.

* These authors contributed equally to this paper.

H. Dantas was with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA. He is now with Microsoft, Redmond, WA 98052 USA. (e-mail:hedantas@microsoft.com).

T. C. Hansen and D. J. Warren are with the Department of Biomedical Engineering, University of Utah, Salt Lake City, UT 84112 USA (e-mail: taylor.c.hansen@utah.edu; david.warren@utah.edu).

V J. Mathews is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA (e-mail:mathews@oregonstate.edu).

A number of different movement intent decoders have been presented in the literature. These include Wiener filters [1], [2], probabilistic methods [3], [4], and recursive Bayesian decoders such as Kalman filters (KF) [5]–[15] to estimate kinematic intent. Many KF-based decoders try to incorporate nonlinear aspects of the neuromuscular system in the decoders [6], [9], [11], [12], [16]. Modern machine learning algorithms, such as extreme machine learning [17], radial basis networks [18], recurrent neural networks [19], and deep and reinforcement learning algorithms [20]–[25] have also been used to infer the relationship between kinematic movements and biological signals. All the decoders presented above are capable of estimating movements simultaneously and continuously in all the degrees of freedom (DoFs) of the limb, making them members of the class of continuous decoders. As the number of DoFs increases, these methods have a tendency to present unwanted movements in some DoFs, which we refer to as cross-movement artifacts. In addition, we may also see jitter in the movements.

A second approach to interpreting movement intent from biological signals is to use classifier-based decoders. Classifier-based algorithms first identify the movement type as one of a finite number of predetermined movement types and then generate a controller trajectory that moves the prosthetic along the path prescribed by the movement type. These methods allow only a limited set of prosthetic movements, but tend to produce less crosstalk and jitter than the class of continuous decoders [26]. Classifier-based algorithms that employ linear classifiers [27], support vector machines [28], Gaussian mixture models [29], naïve Bayes decoders [30] and multilayer perceptron (MLP) neural networks [31] have been described in the literature. Recently, deep learning neural network-based decoders have shown improvements over traditional machine learning approaches to classify hand positions [32]–[35].

A third class of decoders estimates the movement goal from biological and/or other types of sensor signals. Mulliken *et al.* [8] incorporated a goal estimate into the KF framework. Hotson *et al.* [10] proposed a KF-based framework that incorporated a camera-based goal estimate in the decoding system. These decoders exhibited a performance improvement over continuous decoding methods, but they may also require large training datasets with a diverse set of movements in different scenarios.

Each decoder algorithm possesses its own unique advantages. In this paper we present an approach that combines multiple decoders to create a single movement in-

tent decoder and consequently a prosthetic controller that shares the positive characteristics of each of the component decoders employed by the system. We validated the algorithm's capabilities in amputee and non-amputee human subjects performing online movement tasks in a virtual environment, where the subjects controlled a virtual hand in real time. The shared controller approach used in the experimental validation decoded EMG signals with a linear combination of an MLP classifier-based decoder and a KF-based decoder.

A preliminary version of this work was presented at a conference, but the content of this paper differs substantively from the conference proceeding paper [36]. First, this paper generalizes the approach in [36] to combine arbitrary decoders to produce a shared controller. Unlike our preliminary work in [36] that presented a shared controller combining a simulated goal estimator with a Kalman filter-based decoder, this paper presents a complete shared controller implementation involving a Kalman filter and a classifier-based controller. Finally, this paper includes the results of an extensive set of experiments involving amputee and intact-arm subjects.

The rest of this paper is organized as follows: Section II develops the shared controller framework for decoding signals and also how to train the system. Section III describes the experiments and the decoding algorithms employed in this work. Experimental results are provided in Section IV. A discussion of the experimental results and the evaluated algorithms are presented in Section V. Finally, the concluding remarks are made in Section VI.

II. Methods

Broadly, the function of any prosthesis controller is to interpret the desired action from biological and other auxiliary signals related to movements. The objective of a shared controller is to combine the estimates of B different decoders based on their input signals and the current state of the system to produce the desired limb trajectory. For example, a shared controller could inherit the flexibility of a continuous decoder that can replicate the broad range of movements possible with a native hand and a classification-based decoder that generates small amounts of cross movement artifacts and jitter.

We define the state, s_k, of the system at the time sample k as the union of the most recent H_1 instances of the measured biological signals $Z_k = [z_{1,k},...,z_{N,k}]^T$ and the most recent H_2 instances of the position of the prosthetic limb $X_k = [x_{1,k},...,x_{M,k}]^T$. Here $z_{i,k}$ is the feature datum from the ith measurement channel at time k, N is the number of measurement channels, $x_{j,k}$ is the limb position along the jth DoF at time k, and M is the number of DoFs of the limb. That is,

$$s_k = [Z_k,...,Z_{k-H_1+1},X_k,...,X_{k-H_2+1}] \tag{1}$$

and the control signal of the bth decoder is

$$u_k^b = \hat{X}_{k+1}^b = f_b(s_k,\theta_b) \tag{2}$$

where $f_b(\cdot)$ is the bth decoder fully described by the parameters θ_b and its output is interpreted as the bth prediction of the next position of the system. The kinematic position of the prosthetic limb, X_{k+1}, is a combination of the output of all the component decoders. Herein, we use the notation $\widehat{(\cdot)}$ to refer to an estimation of $(\cdot)$; for example, in (2), $\hat{X}_{k+1}^b$ is the estimate of X_{k+1}^b predicted as $f_b(s_k,\theta_b)$.

We assume that the system evolves according to the Markov assumption, where the next state of the limb, s_{k+1}, only depends on the current state, s_k. That is, $p(s_{k+1}|s_k,...,s_1) = p(s_{k+1}|s_k)$. Let τ_a given by

$$\tau_a = \bigcup_{i=1}^{L_a-1}\left(s_i^a, F_c(u_i^{a,1},...,u_i^{a,B})\right)\bigcup s_{L_a}^a \tag{3}$$

be a trajectory in the training set acquired in response to a commanded movement X_k^a. Here, $F_c(\cdot)$ is a mixing function responsible for combining the different decoder outputs into the estimate of the next kinematic state $\widehat{X}_{k+1}$, L_a is the number of samples in the desired trajectory, and s_i^a and $u_i^{a,b}$ represent the state and bth decoder output in the ith time step, respectively, in the a-th trajectory. Let $\tau = \bigcup_{a=1}^{A}\tau_a$ represent the training set consisting of A independent trajectories.

During the training phase, the prosthetic positions are determined by the commanded movement, that is $s_k^a = [Z_k^a,...,Z_{k-H_1+1}^a,X_k^a,...,X_{k-H_2+1}^a]$. In this stage, the objective is to learn the set of parameters θ_b associated with the bth decoder for $1 \le b \le B$. As the number of possible decoders increases, the number of ways in which they can be combined also increases. Consequently, it is not practical to train all decoders at once, especially when training involves data collected from a single human user of the prosthetic. Instead, we follow ideas similar to ensemble learning [37], where the decoders are trained separately and their outputs are combined during the testing and operational phase. The decoder parameters θ_b are fitted to minimize the following objective function for each choice of b:

$$J(\theta_b) = \frac{1}{A}\sum_{a=1}^{A}\frac{1}{L_a-1}\sum_{i=1}^{L_a-1}||X_{i+1}^a - f_b(s_i^a,\theta_b)||^2 \tag{4}$$

where $||\cdot||$ represents the L_2-norm.

We used a gradient descent approach given by

$$\theta_b = \theta_b - \alpha_b\nabla_{\theta_b}J(\theta_b) \tag{5}$$

to minimize the above cost function. Here, α_b is a positive non-zero number interpreted as the learning rate and $\nabla_{\theta_b}J$ is the gradient of J with respect to θ_b. The gradient of the objective function is given by

$$\nabla_{\theta_b}J(\theta_b) = -\frac{1}{A}\sum_{a=1}^{A}\frac{1}{L_a-1}\sum_{i=1}^{L_a-1}\left[[X_{i+1}^a - f_b(s_i^a,\theta_b)][\nabla_{\theta_b}f_b(s_i^a,\theta_b)]^T\right]^T \tag{6}$$

The parameters are updated until a maximum number of L iterations are reached. In our implementation, we employed

a dropout strategy to mitigate problems with over-fitting [38]. We may also terminate the updating process before reaching L iterations if the decoding error is less than a predetermined threshold T corresponding to an acceptable level of performance of the decoder; however, we did not employ an early termination criterion in our analysis.

Once the systems are trained, the parameters θ_b are kept static while operating the prosthetic system. During the operational phase, the prosthetic positions are a collection of past estimates, that is $s_k = [Z_k, ..., Z_{k-H_1+1}, \widehat{X}_k, ..., \widehat{X}_{k-H_2+1}]$.

The derivation provided above is very general. This framework is capable of mixing multiple decoders with different advantages into a single and potentially more effective decoder. There are no restrictions placed on component decoders in this framework, and therefore they can belong to any of the types discussed in the introduction including neural networks and Kalman filters. The shared controller framework derived above is summarized in Algorithm 1.

Training phase
Collect training data
while *Stop Criteria Not Match* **do**
  Train all B decoders using (5) and (6)
end
Online Phase
foreach *time instant, k* **do**
  foreach *Decoder, b* **do**
    Estimate the decoder output, u_k^b, using (2)
  end
  Calculate $\widehat{X}_{k+1}$ as $\widehat{X}_{k+1} = F_c(u_k^1, ..., u_k^B)$
  Update the system state, s_{k+1}, based on the estimated hand position $\widehat{X}_{k+1}$
end

Algorithm 1: The shared prosthetic control algorithm.

III. Experimental Performance Evaluation

A. Shared Controller Architecture

We evaluated the performance of the shared-controller of Section II using an implementation that estimated the next kinematic state of the prosthesis, $\hat{X}_{k+1}$ as a linear combination of the estimates of the individual decoders, i.e.,

$$\widehat{X}_{k+1} = F_c(u_k^1, ..., u_k^B) = \sum_{b=1}^{B} \beta_b u_k^b \tag{7}$$

where β_b's are the mixing parameters for the different decoders with $\sum_{b=1}^{B} \beta_b = 1$ and all β_b are non-negative. We selected the mixing function, $F_c(\cdot)$, as a weighted linear combination for simplicity, but the methodology does not require this.

In this study, two component decoders were employed, i.e., $B = 2$. Given that only two decoders were combined, $\beta_2 = 1 - \beta_1$. Herein, we refer to β_2 as the mixing parameter.

The shared control architecture was a linear combination of two EMG-based decoders: a KF-based continuous decoder and an MLP classifier-based decoder. As discussed earlier, continuous decoders like KFs can control multiple DoFs simultaneously but may suffer from jitter and cross-movement errors. Combining such a decoder with another decoder that has low jitter and cross-movement errors, such as an MLP-based classifier, may result in a prosthetic controller that inherits the good qualities of the component decoders.

To validate this conjecture, we used the KF decoder reported previously in [39]. This KF employed a state vector containing M elements, and the observation vector consisted of N observations. This implies that the KF predicted the kinematic state based on the current state only, and no memory was added to the model. The MLP-based classifier consisted of two hidden layers with 128 hidden nodes in each layer and a rectifier linear unit (ReLU) as the activation function. The input of the classifier is a vector of dimension $N \times H_1$. We used $H_1 = 6$ in the experiments. Our parameter choices in the experiments presented the best performance in pilot experiments not included here.

The classifier assumed 13 possible classes: the 12 flexion/extension positions described in Section III-B and one class for all DoFs in rest position. During online testing, the MLP classifier's output was the current DoF positions plus some δT in the direction of the intended movement as classified by the MLP. In the experiments described here, we used $\delta T = 0.1$. This value presented the best performance in pilot experiments not included here. In this work, the maximum number of training iterations was set to 100 ($L = 100$). The dropout rate was set to 0.5 as commonly done in practice.

To explore the effectiveness of sharing control, a range of mixing parameter values was tested. The values used were $\beta_2 = [0,\ 0.15,\ 0.25,\ 0.4,\ 0.5,\ 0.6,\ 0.75,\ 1.0]$ with a value of 0 representing full KF control, a value of 1.0 representing full MLP control, and intermediate values representing shared control. This range of values was chosen as a result of pilot data not included in this paper.

B. Experimental Setup

The results presented here are from three transradial amputee subjects and six intact-arm subjects (Table 1). All human studies were approved by the University of Utah's Institutional Review Board under IRB Protocol 98851 on Feb. 28, 2017. Informed written consent was obtained from all subjects prior to the studies. Three of the intact-arm participants had previous experience with myoelectric prostheses, and one of these is a co-author. All subjects used their dominant arm; in the case of the amputee subjects, this was the arm they used most often with a prosthesis.

Surface EMG (sEMG) signals from the forearm of the intact-arm subjects and the residual limb of the three amputee subjects were acquired using 32 surface electrodes spaced 2 - 4 cm from each other across the full circumference and length of the forearm to record from extrinsic flexors and extensors [40]. Feature acquisition is similar to prior reports [41]. The 32 single-ended sEMG signals were acquired at 1 kHz sampling rate by a Grapevine

TABLE I
SUBJECT DEMOGRAPHICS

Subject	Gender	Age (yr)	Time Since Amputation (yr)	Amputation Description	Arm used for Experiments	Prior Myoelectric Prosthesis Experience?
AS-1	F	50	4	Left transhumeral, right transradial due to trauma	Right	No
AS-2	F	69	4	Left transradial and all but two digits on right due to trauma	Right	No
AS-3	M	64	2	Bilateral transradial due to trauma	Right	No
IS-1	M	23	N/A	N/A	Right	Some
IS-2	M	27	N/A	N/A	Right	No
IS-3	M	24	N/A	N/A	Right	No
IS-4	M	24	N/A	N/A	Left	No
IS-5	M	27	N/A	N/A	Left	Extensive
IS-6	M	25	N/A	N/A	Right	Extensive

Neural Interface Processor system (Ripple, Salt Lake City, UT) using a proprietary front-end amplifier. These signals were filtered with a 6th-order Butterworth high-pass filter with corner frequency at 15 Hz, a 2nd-order Butterworth low pass filter with corner frequency at 375 Hz, and 60, 120, and 180 Hz notch filters. Differential EMG signals for all 496 possible combinations of the 32 single-ended channels were calculated in software. For each of the 528 single-ended and differential EMG channels, the mean absolute value (MAV) was calculated over a 33.3 ms window of time and subsequently smoothed with a 300 ms rectangular window. In our previous work [22], different features described by Hudgins *et al.* [42] were considered. However, decoders trained with only the MAV features provided the best performance for sEMG decoding. Consequently, the results described here are based on MAV of EMG signals. To reduce the dimensionality and the computational complexity of the decoders, a forward selection operation, including Gram-Schmidt orthogonalization after every iteration of selecting a channel, was performed on the features in the training data [16]. The best 48 features were used as input features for the decoders in the experiments, following Nieveen *et al.* [16].

The sEMG data were recorded while the subject followed a set of predetermined movements of a virtual hand (Musculoskeletal Modeling Software [43]) with their phantom or real hand (Fig. 1). The virtual environment modeled a hand with the following 12 movements: flexion and extension of each of the five digits in isolation and simultaneous flexion and extension of all five digits. Each DoF's range of motion was normalized to the [−1, 1] range, where the relaxed position corresponded to zero value, maximum extension corresponded to -1, and maximum flexion corresponded to 1. If the participant had not previously worked with the virtual hand, they were given a brief freeform practice run of training and testing that was not included in the data analysis. This was to minimize errors associated with unfamiliarity with the testing environment.

After any initial practice, data from a period of time with no activity of the limb was recorded for calculating the baseline noise of the incoming data channels. The per-feature MAV value from this baseline period was subtracted from each feature prior to use in training and testing of algorithms.

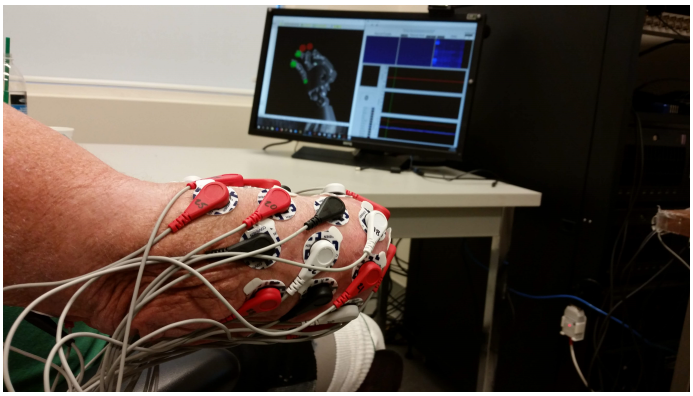

Fig. 1. Experimental setup used in this work. The volunteer had sEMG electrodes attached on his skin and was asked to follow the movements shown on the screen using his phantom hand. The screen was used to show the movements during the training phase. During the testing phase, the screen showed the targets for each digit (as circles) and the decoded position of the hand. The target was colored green if the decoded position placed its digit in the target region, and red if it did not.

During training, the subjects were instructed to mimic movements of a hand displayed in the virtual reality environment with their phantom or intact hand simultaneously with recording of the sEMG signals. Each of the 12 movement types described above was repeated six times, with small variations in the movement extent, velocity, and hold time for each repetition. The instructed movements followed a semi-sinusoidal path at velocities deemed comfortable by the subject and included target positions at ±0.8 or ±1.0 of the normalized range for each DoF. Using these data, both the KF and MLP were trained, and the subject was then given real-time control of the virtual hand.

Testing consisted of a target-touching task with the virtual prosthetic hand. Participants were asked to move all DoFs

of the hand to indicated target regions, representing one of the 12 movement types, and hold all DoFs in the target region for as long as possible during each 7 s long trial. The target regions were newly-indicated positions for each DoF with a tolerance of ±10% of the full range of the DoF. All stationary DoFs were instructed to be at the rest position. Participants had color-coded visual feedback to know if the virtual hand was in the indicated target region. As in training, testing trials were sequential in nature, with all trials of one movement type being completed before moving to the next. Although the 12 test movements were nominally the same as those during training, the targets were at 50% of full flexion/extension.

During the testing period, participants completed three sessions with several minutes of resting between sessions. A session consisted of completing all eight trials (one for each value in β_2 described in Section III-A) for all 12 movement types (96 trials per session). Each of the eight trials within a movement type had a different mixing value applied in a shuffled order such that no values were repeated. The experiment was double-blinded, preventing even participants familiar with the experimental setup (and the experimenter) from knowing the mixing parameters for a given trial. One of the intact-arm subjects (IS-3) completed just two sessions due to complications with electrode placement before his third session (for a total of 192 trials).

C. Performance Analyses

In order to perform comprehensive data analysis for the experiments, we used three performance metrics: time in the target region, root-mean-squared error (RMSE) between the target region and DoFs performing intended movements, and RMSE between the target region and DoFs instructed to be stationary. In order to define these metrics, we first define the target region associated with each commanded movement. If x_j^{DM} is a single number that represents the desired goal for the jth DoF, the target region is defined by the interval $[x_j^{DM}-\Delta R, x_j^{DM}+\Delta R]$, with $\Delta R = 0.1$ for all analyses presented in this paper. Then, the RMSE for intended movements is calculated as

$$D_M = \sqrt{\frac{1}{HM_m}\sum_{k=1}^{H}\sum_{j=1}^{M_m}(\max(|x_{j,k}^{M}-x_j^{DM}|-\Delta R,0))^2} \quad (8)$$

where $x_{j,k}^{M}$ is the decoded position of the jth DoF that was instructed to move at the kth time bin, and M_m DoFs were instructed to move. This metric covers the complete duration of each commanded movement. The RMSE for unintended movements (also called cross-movements herein) is calculated as

$$D_S = \sqrt{\frac{1}{HM_s}\sum_{k=1}^{H}\sum_{j=1}^{M_s}(x_{j,k}^{S})^2} \quad (9)$$

where $x_{j,k}^{S}$ is the position of the j-th stationary DoF for the k-th time bin and M_s represents the number of stationary DoFs. Finally, the time in the target region is the total amount of time all DoFs, both stationary and moving, spent in the target region in a trial.

D. Statistical Analyses

The data for each metric were examined for outliers; this was done separately for each subject and mixing parameter value using the upper quartile (UQ), lower quartile (LQ), and interquartile range (IQR). If a data point for a metric was outside the interval [LQ-1.5∗IQR, UQ+1.5∗IQR], it was considered an outlier and removed from the analysis. Removal of a data point for one metric and subject does not imply removal of the same data point for all metrics and subjects. To accommodate for one subject having fewer sessions than the other subjects, the mean value across all sessions for each subject was then taken for each performance metric at each mixing parameter value. For each subject type (amputee and intact-arm), the mean performance metric values derived above were grouped by mixing parameter value across all datasets. Normality was verified with a Shapiro-Wilk test for data within each group. In all cases, there was no evidence that normality should not be assumed. This was followed by a one-way ANOVA for each of the three performance metrics, where the mixing parameter (β_2) was the factor. If the ANOVA indicated that the mean value of at least one level of the mixing parameter differed from the others, pairwise comparisons were made using a Dunn-Šidák correction for multiple comparisons. Whenever data are presented in the text as $XX \pm YY$, XX is the arithmetic mean and YY is the standard error of the mean (SEM). In all graphics, error bars represent the SEM of the particular configuration, and the symbol or the horizontal line represent the arithmetic mean.

IV. Results

The enhanced performance afforded by sharing control can be visualized by representative time traces of the kinematic output of each DoF for a representative set of mixing parameters, β_2 = [0, 0.15, 1.0] (Fig. 2). It is notable that the cases of KF-only ($\beta_2 = 0$) and shared control ($\beta_2 = 0.15$) exhibited similar performance for the moving DoF, while using the classifier-only decoder ($\beta_2 = 1$) resulted in large oscillations around the target region of the moving DoF. Consequently, the classifier-only decoder ($\beta_2 = 1$) would result in smaller values of time in target and larger values of the RMSE of intended movements than the other two cases shown. The shared controller employing $\beta_2 = 0.15$ exhibited less unintended movements in the static DoFs than the KF-only decoder, as indicated by the predictions of the stationary digits remaining nearer the origin. When the classifier-based decoder was employed, no unintended movement in the stationary digits was observed (indicated by all stationary digit predictions being at the origin) in this example because the classifier made the correct decision.

The performance of eight different controller conditions corresponding to β_2 = [0, 0.15, 0.25, 0.4, 0.5, 0.6, 0.75, 1.0] were statistically analyzed according to the three performance metrics defined in Section III-C. For each metric

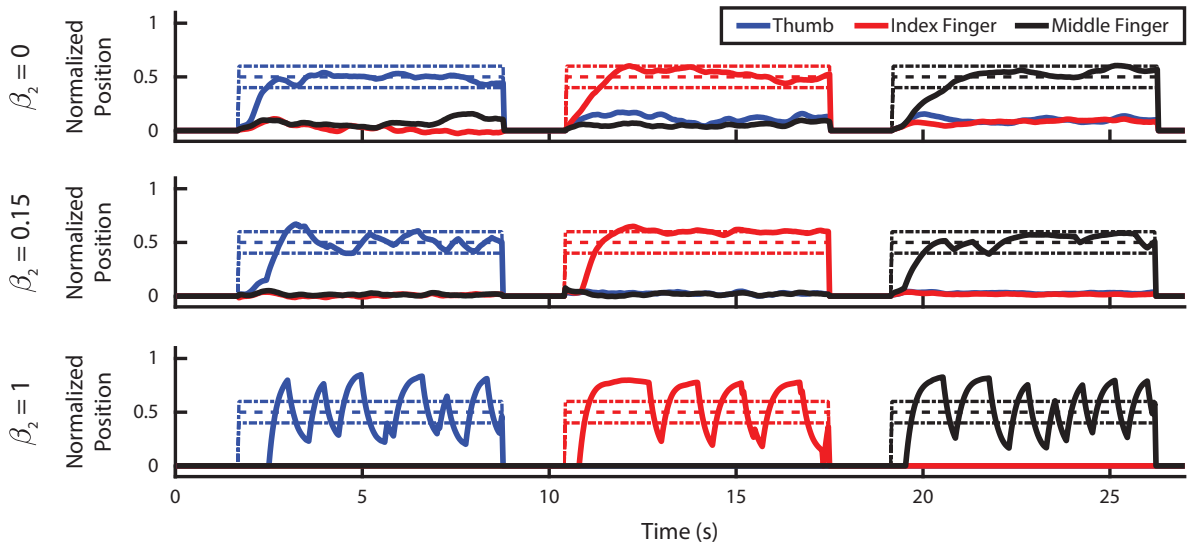

Fig. 2. Representative examples of the kinematic output of three different mixing parameters with an intact-arm subject. All cases were reproduced here from the testing phase, where IS-5's sEMG signals were used to derive the movement action via a previously trained KF and the MLP classifier-based decoder. For each condition, the dashed line represents the target position, the dot-dashed line represents the target region, and the solid line represents the controller's real-time prediction of the subject's decoded movement. The color of each line represents a particular DoF.

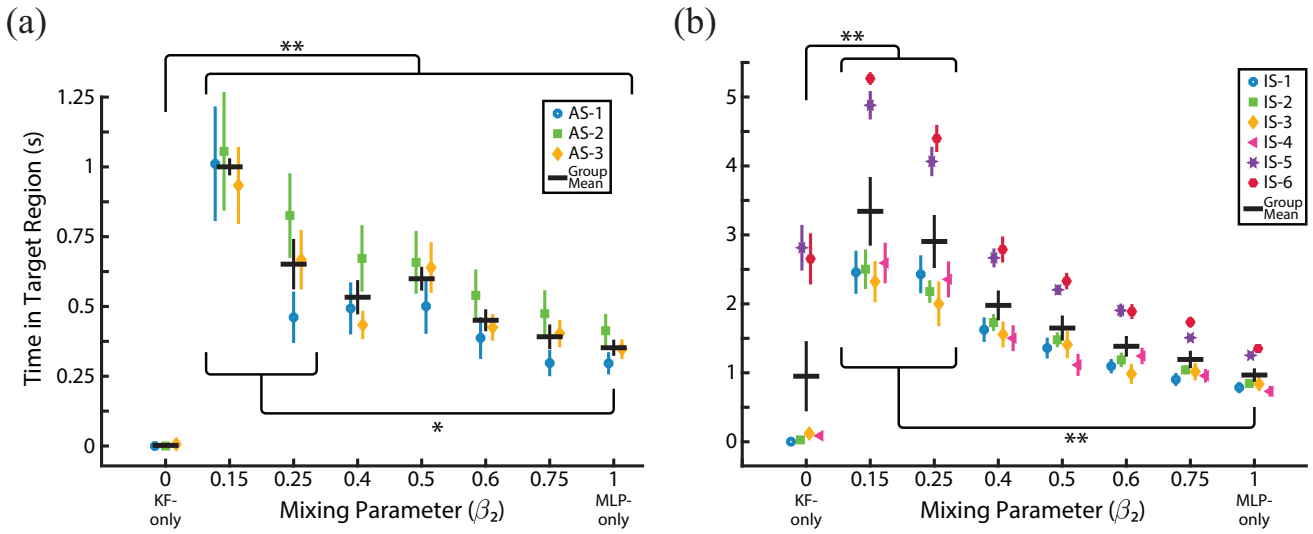

Fig. 3. Shared control improved the time in the target region for both amputee (a) and intact-arm (b) subjects compared to the KF-only and MLP-only cases. In this and similar figures, the colored symbols represent the mean individual performance of each participant over the three sessions, while the black bars represent the aggregated mean of the performances across all participants' datasets. Vertical bars indicate SEM. Statistical tests were performed on aggregated means. * p-value < 0.05; ** p-value < 0.01; *** p-value < 0.001 (after Dunn-Šidák correction for multiple comparisons). Not all significant comparisons are shown.

and each subject type, one-way ANOVA revealed that the performance of at least one β_2 value differed from the rest ($p < 0.05$ for all six cases), allowing for subsequent *post-hoc* pairwise comparisons.

For both the amputee subjects (Fig. 3a) and the intact-arm subjects (Fig. 3b), the amount of time in the target region was best when sharing control, particularly those with strong representation of the KF decoder, and poor for both the KF-only and MLP-only controllers. The KF-only controller maintained the virtual hand in the target region for 0.00 ± 0.00 s and 0.95 ± 0.52 s, for amputee and intact-arm subjects, respectively. On the other end of the mixing parameter range, the MLP classifier-based controller alone maintained the virtual hand in the target region for 0.35 ± 0.03 s and 0.97 ± 0.10 s, for amputee and intact-arm subjects, respectively. The best performance was for the shared controller with $\beta_2 = 0.15$ and yielded 1.00 ± 0.03 s and 3.34 ± 0.50 s in the target region, for amputee and intact-arm subjects, respectively. Subsequent pairwise com-

parisons revealed that the time in the target region of the shared controller with $\beta_2 = 0.15$ was significantly higher than either the KF-only or MLP-only cases, for both subject types ($p < 0.001$ for all comparisons).

The RMSE for intended movements was best (lowest values) for the KF-only controller and tended to become worse as more of the MLP classifier-based decoder's output was added to the shared controller for both amputee subjects (Fig. 4a) and intact-arm subjects (Fig. 4b). The KF-only controller had RMSE values of 0.22 ± 0.01 and 0.14 ± 0.01, for amputee and intact-arm subjects, respectively. When considering the MLP classifier-based decoder alone, the RMSE values were much higher at 0.34 ± 0.003 and 0.24 ± 0.02, for amputee and intact-arm subjects, respectively. For both subject types, when control was shared, the lowest mean RMSE was for $\beta_2 = 0.15$. For this case, RMSE values for intended movements were 0.25 ± 0.01 and 0.15 ± 0.01, for amputee and intact-arm subjects, respectively. Subsequent pairwise comparisons revealed that the RMSE of the shared controller with $\beta_2 = 0.15$ was not significantly different from the KF-only case, for both amputee and intact-arm subjects ($p = 0.8$ and $p = 1$, respectively). It was also found that this same shared controller had significantly lower RMSE than the MLP-only case, for both amputee and intact-arm subjects ($p < 0.001$ and $p < 0.01$, respectively).

As compared to a KF-only controller, the inclusion of the MLP classifier-based decoder into the controller drastically improved performance (reduced the RMSE) of unintended movements for both amputee subjects (Fig. 5a) and intact-arm subjects (Fig. 5b). The KF-only controller had RMSE values of 0.09 ± 0.01 and 0.03 ± 0.01, for amputee and intact-arm subjects, respectively. The MLP-only controller had RMSE values of 0.02 ± 0.01 and 0.006 ± 0.004, for amputee and intact-arm subjects, respectively. For both subject types, when control was shared, the lowest mean RMSE was for $\beta_2 = 0.25$, which resulted in RMSE values of 0.01 ± 0.006 and 0.002 ± 0.001, for amputee and intact-arm subjects, respectively. Subsequent pairwise comparisons revealed that no shared controller differed significantly from the MLP-only case ($p \geq 0.05$). For the amputee subjects, the shared controllers with $\beta_2 = [0.25,\ 0.4]$ had RMSE values significantly lower than the KF-only case ($p < 0.05$). For the intact-arm subjects, all shared controller configurations had significantly lower RMSE than the KF-only case ($p < 0.001$ for all comparisons).

V. Discussion

Broadly, linearly sharing control between a Kalman filter decoder and a multilayer perceptron network-based classifier performed better than the constituent parts, particularly when the controller was more heavily weighted towards the KF predictions. The performance improvement afforded by sharing control between both decoders was perhaps most intuitively exhibited by the amount of time in the target region. For both amputee and intact-arm subjects alike, the best performance was attained when the shared controller consisted of an 85% KF-based and 15% MLP classifier-based decoder (i.e., $\beta_2 = 0.15$). With amputee subjects, the mean performance at this level of sharing, 1 second, was significantly better than with a KF-only approach (effective zero mean value) or classifier-only approach (0.35 seconds). Equally, with intact-arm subjects, the mean performance (3.3 seconds) was significantly better than with a KF-only approach (1 second) or classifier-only approach (1 second). Observationally, the larger intact-arm subject performance was influenced by two subjects, IS-5 and IS-6, who had substantial prior experience with the experimental setup. Nevertheless, these two experienced subjects also demonstrated that sharing control increased their performance (e.g., mean performance of 5 seconds with $\beta_2 = 0.15$ over a KF-only controller (2.7 seconds) or classifier-only controller (1.3 seconds). To assure ourselves that the statistical comparisons were not unduly influenced by these two subjects, we repeated the intact-arm subject analysis without these subjects for all metrics and we found similar, and in some cases, stronger results.

For this metric (and the other metrics), the amputee subjects generally had lower performance than intact-arm subjects, which is a finding we often observe in studies with both amputee and intact-arm subjects performing similar movement trials. At best, we can only speculate as to the reasons why, but the differences in ages and in recency of usage of the pertinent muscles between the two types of subjects may be possible factors.

Other benefits of combining control are seen when considering the RMSE of both intended and unintended movements. The classifier strongly reduced the RMSE of unintended movements, and this effect was also observed when sharing control. Compared to the KF-only decoder, any amount of sharing in intact-arm subjects significantly reduced the unintended movement RMSE whereas this was only true for a subset of β_b values with amputee subjects. In contrast, when compared to the KF-only decoder, increasing participation of the classifier appeared to increase the intended movement RMSE, significantly for amputee subjects but not so for intact-arm subjects. For the degree of sharing that benefited the time in target metric ($\beta_2 = 0.15$) the increase in intended movement RMSE was modest and not significant, and the decrease in unintended movement was substantive and significant (only for the intact-arm subjects). From the results presented herein, we conclude that sharing control meaningfully boosted the time in the target region while reducing cross-movements without introducing significant errors in intended movements, as compared to either of its constituent parts.

Although the results we present clearly demonstrate the value of sharing control, the information presented herein provides only a limited view of what is possible with sharing control. The two decoding methods used are among the most common encountered moment-by-moment decoding methods utilizing sensorimotor-derived features, albeit with the two methods often described as two competing approaches to performing decoding. Jointly, the sum of the two was found to be better than the parts. However, there is nothing in its design that restricts sharing to be limited to only these two methods or to only two methods. The

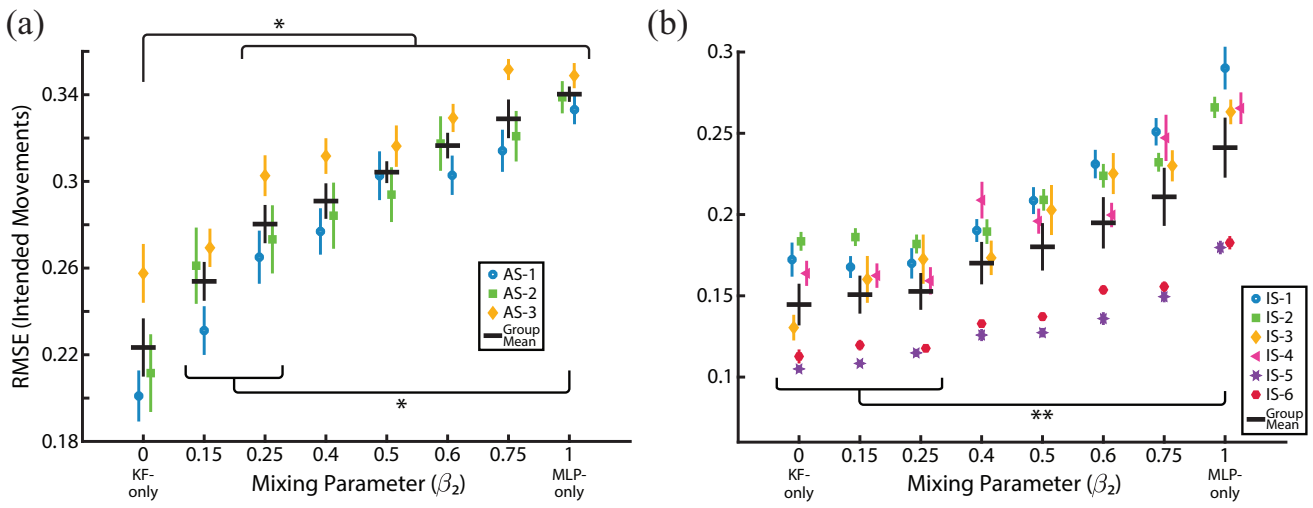

Fig. 4. Increasing amounts of the MLP classifier-based decoder's output in the shared controller produced higher RMSE for intended movements for both amputee (a) and intact-arm (b) subjects. Although the KF-only decoder had the lowest RMSE for both subject types, the difference between the KF-only decoder and the $\beta_2 = 0.15$ case was not statistically significant. In both subject types, the MLP-only case had significantly-higher RMSE than the KF-only decoder as well as the shared controllers with $\beta_2 = [0.15, 0.25]$. Not all significant comparisons are shown.

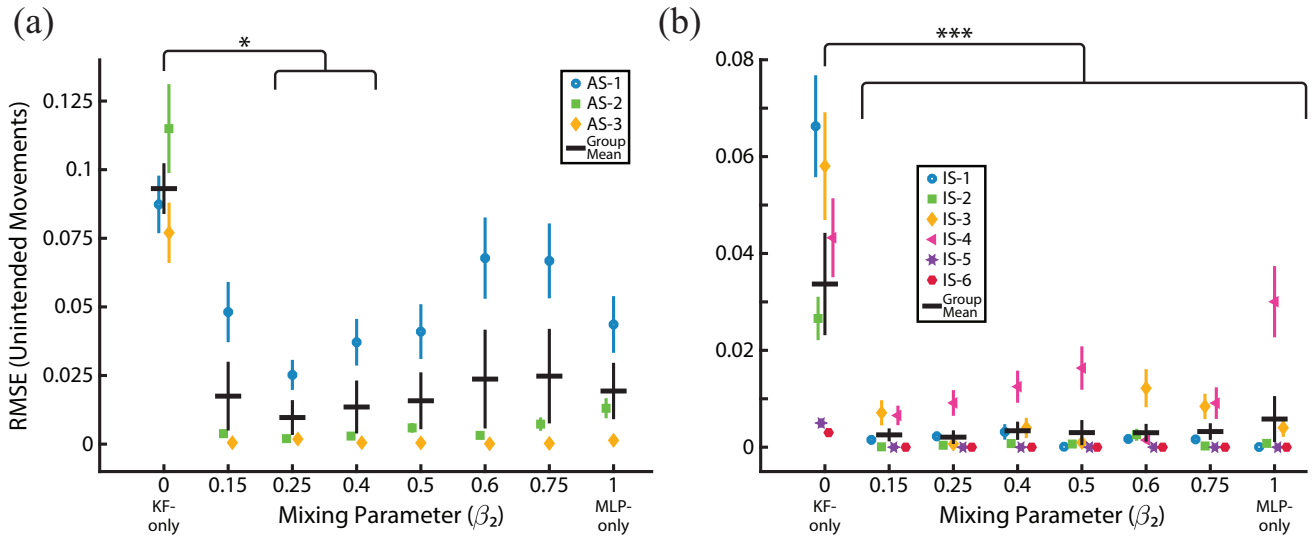

Fig. 5. Any inclusion of the output from the MLP classifier-based decoder drastically reduced cross-movements for both amputee (a) and intact-arm (b) subjects. For the amputee subjects, the shared controllers with $\beta_2 = [0.25, 0.4]$ had significantly-lower RMSE than the KF-only decoder. For the intact-arm subjects, all shared controllers had lower RMSE than the KF-only decoder. The shared controllers did not have statistically different performance from the MLP-only case for both subject types. Not all significant comparisons are shown.

design could easily handle the addition of, for example, a convolutional neural network moment-by-moment position estimator that uses the same features, resulting in a combination of three approaches. Of course, with the addition of a third component, finding the best degree of sharing would be more difficult than the evaluation done herein. A possible approach for designing such shared controllers is to choose a metric that represents user satisfaction and select the parameters that results in the best metric using online learning to find the best decoder combination. In the present study, the time-in-target is the metric that reflected the user satisfaction the most.

Further, there is no requirement in the design that restricts the combination of the parts to be linear. For example, an alternative approach with the two decoders would be to use the classifier output to gate the KF's estimates. That is, if the classifier indicated that a finger was not moving from rest, the KF's estimate for that digit would be zeroed out. A possible approach to designing such a system is to parameterize the combining function $F_c(\cdot)$ and estimate parameters of this function using cross-validation. The combining function could also be updated in real time using an online learning or a reinforcement learning algorithm [44].

Similarly, there is no requirement that only moment-by-moment estimators can be used. In the work that initiated this research [36], we found the combination of a KF and a simulated end-point goal estimator significantly improved

performance over the KF alone. In that work, we did not define the nature of the end-point estimator but one could easily imagine it being a classifier of a finite set of grip states based on EMG features. Alternatively, it could have been a system that uses non-biologically sourced features, such as an image detection system with a camera or a prosthesis with built-in proximity sensors [45]–[47]. In these applications, the same shared control algorithm developed in the present work would control the prosthesis, with control shared between a movement-intent decoder (e.g. KF) under human control, and autonomous control from a non-biological sensor(s). One could ultimately imagine a shared control system that combines its constituent parts in a non-linear, time-varying manner that uses an end-point estimator to move a prosthetic hand close to a target (e.g., a coffee cup), proximity sensors in the prosthesis to shape the fingers around the cup, and the user's biological signals to control whether the cup is firmly grasped in order to lift it to the user's mouth or lightly gripped to allow sliding the cup across the table.

We do not claim optimality for the shared controller based on the results and methodology shown in this paper. It may be possible to model the combination process as a diffusion process [48], [49]. It has been shown, using diffusion process models in other fields such as network optimization, that combining multiple system outputs as a weighted sum can achieve optimal results [49]. Similar analyses may be possible in our application also, but such an analysis will require analytical models of human performance of the task.

The concept of sharing control between multiple movement intent decoders for upper-limb prostheses has been previously explored to some degree for individuals with amputations [50]–[54]. There have been multiple efforts to endow a prosthetic hand with automatic slip detection in order to stabilize gripping of grasped objects [50], [51]. One of these has even enjoyed commercial success [52]. A shared control system was recently implemented to toggle control between an EMG-based MLP decoder (under human control) and a computer controller [54]. Once the human-operated prosthesis made contact with a desired object, control switched to the computer controller, which automatically increased contact area between the object and the prosthesis. It remains to be seen if such an approach will scale to a real-world prosthesis given the size of the sensors implemented in that work. Importantly, the term "shared control" in that work is loosely used and perhaps is better termed "traded control" given the toggling nature between control modes, as has been proposed elsewhere [55].

Shared control has also been implemented in various forms for the spinal cord injury population. Early work with non-human primates demonstrated the advantage of sharing control between a computer reflex controller and brain-derived commands [55]. Subsequent clinical work has similarly suggested advantages for hybridizing traditional brain-computer interface (BCI) systems with other movement intent decoders [56]. Utilizing computer vision predictions in tandem with BCI systems has also been explored [57], [58]. Recently, there has been much interest in sharing control between EEG decoders and neuronal information provided by BCI systems [58], [59]. Finally, a non-invasive system has been introduced to share control between EEG- and EMG-based decoders, although it has yet to be tested in amputee or spinal cord injured subjects [60]. Certainly, the shared control system described in our present work would adapt well to the spinal cord injury population as well.

VI. Conclusion

This paper presented a framework capable of combining multiple methods for interpreting volitional movement intent from biological and non-biological sensor signals. The performance enhancement possible by a shared controller architecture to control a high-degree-of-freedom prosthetic hand using sEMG signals obtained from amputee and intact-arm subjects was demonstrated through experimental results presented in the paper. The shared controller that combined a KF-based decoder and a classifier-based decoder significantly outperformed the KF-only and MLP-only decoders in the time-in-the-target-region metric. The classifier-based decoder was able to estimate which DoF or combination of DoFs the user intends to move with high precision but did not estimate the precise amount of movement accurately. On the other hand, the KF-based decoder was able to estimate the amount of movement in a given DoF more accurately but was prone to more cross-movement errors. The combination of these two decoders shared the better qualities of each component decoder.

The framework of this paper can easily incorporate decoders based on input signals acquired via external sensors such as cameras and pressure sensors. External sensor signals may be used to interpret the user's environment and movement goals. In our preliminary work [36], we presented the concept of such a shared controller that combined a controller based on a simulated goal estimator and a KF-based decoder. Additional work that employs real-time goal estimators and the shared control framework is currently underway in our labs.

In the configuration reported herein, the parameters of the decoding algorithms were set by training prior to regular use and kept frozen during testing and normal operation. The authors are currently working on extending the decoder capabilities by updating their parameters online. It is also desirable to update the mixing parameters of the shared controller online rather than use pre-selected values throughout their use.

The results presented in the paper suggest that the shared controller strategy has the potential to help individuals with upper-limb amputation achieve more precise prosthesis control than currently possible with any single movement intent decoder. As a result, the presented approach has the potential to substantially improve the quality of life of people with upper-limb loss.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation (NSF) Grants 1533649, 1901492 and 1901236 and in part by the Hand Proprioception and Touch Interfaces (HAPTIX) program administered by the Biological Technologies Office (BTO) of the Defense Advanced Research Projects Agency (DARPA), through the Space and Naval Warfare Systems Center, Contract No. N66001-15-C-4017. We gratefully acknowledge the support of NVIDIA Corporation for the donation of the Tesla K40 GPU used in this research. We also thank Jacob George, Tyler Davis, and Gregory Clark for their help with experimental work and many stimulating discussions. Finally, we thank our volunteer subjects who provided the data that made this study possible.

REFERENCES

[1] J. Wessberg *et al.*, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, no. 1, pp. 361–365, Nov. 2000.

[2] L. R. Hochberg *et al.*, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, Jul. 2006.

[3] W. Wu *et al.*, "Modeling and decoding motor cortical activity using a switching Kalman filter," *IEEE Trans. on Bio. Eng.*, vol. 51, no. 6, pp. 933–942, Jun. 2004.

[4] Y. Gao *et al.*, "Probabilistic inference of hand motion from neural activity in motor cortex," in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, ser. NIPS'01. Cambridge, MA, USA: MIT Press, 2001, pp. 213–220.

[5] S.-P. Kim *et al.*, "Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia," *J. Neural Engineering*, vol. 5, no. 4, pp. 455–476, Jul. 2008.

[6] W. Malik *et al.*, "Efficient decoding with steady-state Kalman filter in neural interface systems," *IEEE Trans. Neural Syst. Rehabil. Eng*, vol. 19, no. 1, pp. 25–34, Feb. 2011.

[7] V. Gilja *et al.*, "A brain machine interface control algorithm designed from a feedback control perspective," in *Ann. Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, San Diego, CA, Aug. 2012, pp. 1318–1322.

[8] G. H. Mulliken *et al.*, "Decoding trajectories from posterior parietal cortex ensembles," *J. Neuroscience*, vol. 28, no. 48, pp. 12 913–12 926, Nov. 2008.

[9] Z. Li *et al.*, "Unscented Kalman filter for brain-machine interfaces," *PloS One*, vol. 4, no. 7, pp. 1–18, Jul. 2009.

[10] G. Hotson *et al.*, "High Precision Neural Decoding of Complex Movement Trajectories Using Recursive Bayesian Estimation With Dynamic Movement Primitives," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 676–683, Jul. 2016.

[11] H. Dantas *et al.*, "Neural decoding using a nonlinear generative model for brain-computer interface," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 2014, pp. 4683–4687.

[12] D. J. Warren *et al.*, "Recording and decoding for neural prostheses," *Proceedings of the IEEE*, vol. 104, no. 2, pp. 374–391, Feb. 2016.

[13] G. A. Clark *et al.*, "Using multiple high-count electrode arrays in human median and ulnar nerves to restore sensorimotor function after previous transradial amputation of the hand," in *2014 36th Ann. Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, Chicago, IL, USA, Aug 2014, pp. 1977–1980.

[14] T. S. Davis *et al.*, "Restoring motor control and sensory feedback in people with upper extremity amputations using arrays of 96 microelectrodes implanted in the median and ulnar nerves," *J. Neural Engineering*, vol. 13, no. 3, p. 036001, Jun. 2016.

[15] S. Wendelken *et al.*, "Restoration of motor control and proprioceptive and cutaneous sensation in humans with prior upper-limb amputation via multiple utah slanted electrode arrays (USEAs) implanted in residual peripheral arm nerves," *J. Neuroeng. Rehabil*, vol. 14, no. 1, p. 121, Nov. 2017.

[16] J. Nieveen *et al.*, "Polynomial Kalman filter for myoelectric prosthetics using efficient kernel ridge regression," in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, Shanghai, China, May 2017, pp. 432–435.

[17] Y. Chen *et al.*, "A 128-channel extreme learning machine-based neural decoder for brain machine interfaces," *IEEE Trans. on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 679–692, Jun. 2016.

[18] M. S. Islam *et al.*, "Decoding movements from human deep brain local field potentials using radial basis function neural network," in *2014 IEEE 27th Int. Symposium on Computer-Based Medical Systems*, Washington, DC, USA, May 2014, pp. 105–108.

[19] D. Sussillo *et al.*, "A recurrent neural network for closed-loop intracortical brain-machine interface decoders," *J. Neural Engineering*, vol. 9, no. 2, p. 026027, Mar. 2012.

[20] M. Mahmud *et al.*, "Applications of deep learning and reinforcement learning to biological data," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2063–2079, Jun. 2018.

[21] E. Nurse *et al.*, "Decoding EEG and LFP signals using deep learning: Heading truenorth," in *Proceedings of the ACM Int. Conf. on Computing Frontiers*, ser. CF '16. New York, NY, USA: ACM, 2016, pp. 259–266.

[22] H. Dantas *et al.*, "Deep learning movement intent decoders trained with dataset aggregation for prosthetic limb control," *IEEE Trans. on Biomedical Engineering*, vol. 66, no. 11, pp. 3192–3203, Nov. 2019.

[23] H. Dantas *et al.*, "Neural decoding systems using Markov decision processes," in *2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, Mar. Mar. 2017, pp. 974–978.

[24] J. Bae *et al.*, "Kernel temporal differences for neural decoding," *Intell. Neuroscience*, vol. 2015, no. 17, pp. 1–17, Jan. 2015.

[25] Y. Wang *et al.*, "Neural control of a tracking task via attention-gated reinforcement learning for brain-machine interfaces," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 3, pp. 458–467, May 2015.

[26] J. V. Haxby *et al.*, "Decoding neural representational spaces using multivariate pattern analysis," *Annual Review of Neuroscience*, vol. 37, no. 1, pp. 435–456, Jun. 2014.

[27] G. Schalk *et al.*, "Decoding two-dimensional movement trajectories using electrocorticographic signals in humans," *J. Neural Engineering*, vol. 4, no. 3, pp. 264–275, Jun. 2007.

[28] K. H. Kim *et al.*, "Advantage of support vector machine for neural spike train decoding under spike sorting errors," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, China, Jan. 2005, pp. 5280–5283.

[29] J. G. Cruz-Garza *et al.*, "Neural decoding of expressive human movement from scalp electroencephalography (EEG)," *Frontiers in Human Neuroscience*, vol. 8, no. 1, p. 188, Apr. 2014.

[30] C. A. Chestek *et al.*, "Hand posture classification using electrocorticography signals in the gamma band over human sensorimotor brain areas," *J. Neural Engineering*, vol. 10, no. 2, p. 026002, Jan. 2013.

[31] G. W. Favieiro *et al.*, "Decoding arm movements by myoeletric signals and artificial neural networks," in *ISSNIP Biosignals and Biorobotics Conference 2011*, Vitoria, Brazil, Jan. 2011, pp. 1–6.

[32] M. Atzori *et al.*, "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands," *Frontiers in Neurorobotics*, vol. 10, no. 9, p. 9, Sep. 2016.

[33] J. He, *et al.*, "User adaptation in long-term, open-loop myoelectric training: implications for EMG pattern recognition in prosthesis control," *J. Neural Engineering*, vol. 12, no. 4, p. 046005, Jun. 2015.

[34] D. Farina *et al.*, "The extraction of neural information from the surface emg for the control of upper-limb prostheses: Emerging avenues and challenges," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 797–809, Jul. 2014.

[35] H. Choi *et al.*, "Movement state classification for bimanual bci from non-human primate's epidural ecog using three-dimensional convolutional neural network," in *2018 6th Int. Conf. on Brain-Computer Interfaces (BCI)*, GangWon, South Korea, Jan. 2018, pp. 1–3.

[36] H. Dantas *et al.*, "Shared human-machine control for self-aware prostheses," in *2018 IEEE Inter. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, Apr. 2018, pp. 6593–6597.

[37] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. New York, NY: Springer Publishing Company, Incorporated, 2012.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from

overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, Jan. 2014.
[39] W. Wu *et al.*, "Bayesian population decoding of motor cortical activity using a Kalman filter," *Neural Computation*, vol. 18, no. 1, pp. 80–118, Mar. 2006.
[40] J. A. George *et al.*, "Inexpensive surface electromyography sleeve with consistent electrode placement enables dexterous and stable prosthetic control through deep learning," *arXiv:2003.00070*, 2020.
[41] D. M. Page *et al.*, "Motor control and sensory feedback enhance prosthesis embodiment and reduce phantom pain after long-term hand amputation," *Frontiers in Human Neuroscience*, vol. 12, no. 1, p. 352, Oct. 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnhum.2018.00352
[42] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Trans. on Biomedical Engineering*, vol. 50, no. 7, pp. 848–854, Jul. 2003.
[43] R. Davoodi *et al.*, "Model-based development of neural prostheses for movement," *IEEE Trans. on Biomedical Engineering*, vol. 54, no. 11, pp. 1909–1918, Nov. 2007.
[44] S. Levine *et al.*, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, Jan. 2016.
[45] T. C. Hansen *et al.*, "Shared controllers improve control and performance of upper-limb prostheses," in *49th Annual Society for Neuroscience National Conference*, Chicago, IL, USA, Oct. 2019.
[46] J. Segil *et al.*, "Multi-modal prosthetic fingertip sensor with proximity, contact, and force localization capabilities," *Advances in Mechanical Engineering*, vol. 11, no. 4, Apr. 2019.
[47] A. Maldonado *et al.*, "Improving robot manipulation through fingertip perception," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 2947–2954.
[48] S. Xie and L. Guo, "Analysis of distributed adaptive filters based on diffusion strategies over sensor networks," *IEEE Trans. on Automatic Control*, vol. 63, no. 11, pp. 3643–3658, Nov. 2018.
[49] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.
[50] F. Veiga *et al.*, "Grip stabilization of novel objects using slip prediction," *IEEE Trans. on Haptics*, vol. 11, no. 4, pp. 531–542, 2018.
[51] E. D. Engeberg and S. G. Meek, "Adaptive sliding mode control for prosthetic hands to simultaneously prevent slip and minimize deformation of grasped objects," *IEEE/ASME Trans. on Mechatronics*, vol. 18, no. 1, pp. 376–385, 2013.
[52] A. L. Ciancio *et al.*, "Control of prosthetic hands via the peripheral nervous system," *Frontiers In Neuroscience*, vol. 10, 2016.
[53] C. Cipriani *et al.*, "On the shared control of an EMG-controlled prosthetic hand: Analysis of user-prosthesis interaction," *IEEE Trans. on Robotics*, vol. 24, no. 1, pp. 170–184, 2008.
[54] K. Z. Zhuang *et al.*, "Shared human–robot proportional control of a dexterous myoelectric prosthesis," *Nature Machine Intelligence*, vol. 1, no. 9, pp. 400–411, Sep. 2019.
[55] S.-P. Kim *et al.*, "A comparison of optimal MIMO linear and nonlinear models for brain-machine interfaces," *J. Neural Engineering*, vol. 3, no. 2, pp. 145–161, May 2006.
[56] B. Z. Allison *et al.*, "Toward smarter BCIs: extending BCIs through hybridization and intelligent control," *J. Neural Engineering*, vol. 9, no. 1, p. 013001, 2012.
[57] J. E. Downey *et al.*, "Blending of brain-machine interface and vision-guided autonomous robotics improves neuroprosthetic arm performance during grasping.(report)," *J. Neuroeng. Rehab.*, vol. 13, no. 30, 2016.
[58] D. P. Mcmullen *et al.*, "Demonstration of a semi-autonomous hybrid brain-machine interface using human intracranial EEG, eye tracking, and computer vision to control a robotic upper limb prosthetic," *IEEE Trans. on Neural Systems and Rehab. Engineering*, vol. 22, no. 4, pp. 784–796, 2014.
[59] T. Li *et al.*, "Brain-machine interface control of a manipulator using small-world neural network and shared control strategy," *J. Neuroscience Methods*, vol. 224, p. 26, 2014.
[60] I. Ruhunage *et al.*, "Hybrid EEG-EMG signals based approach for control of hand motions of a transhumeral prosthesis," in *2019 IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech)*, 2019, pp. 50–53.