Semi-Supervised Adaptive Learning for Decoding Movement Intent from Electromyograms

Henrique Dantas
School of Electrical Engineering
Oregon State University
Corvallis, OR, USA
dantash@oregonstate.edu

V John Mathews
School of Electrical Engineering
Oregon State University
Corvallis, OR, USA
mathews@oregonstate.edu

David Warren
Department of Biomedical Engineering
University of Utah
Salt Lake City, UT, USA
david.warren@utah.edu

***Abstract*—This paper presents an adaptive learning algorithm for predicting movement intent using electromyogram (EMG) signals and controlling a prosthetic arm. The adaptive decoder enables use of the prosthetic systems for long periods of time without the necessity to retrain them. The method of this paper employs a neural network-based decoder and we present a method to update its parameters during the operation phase. Initially, the decoder parameters are estimated during a training phase. During the normal operation, the parameters of the algorithm are updated in a semi-supervised manner based on a movement model. The results presented here, obtained from a single amputee subject, suggest that the approach of this paper improves long-term performance of the decoders over the current state-of-the-art with statistical significance.**

***Index Terms*—Kalman Filter, Neural Networks, Markov Decision Processes, Movement Intent Decoder, Semi-supervised Learning, Online Learning**

I. Introduction

In past the few years, prosthetic devices have progressed from cable-driven systems controlled by shoulder movements to systems that can interpret biological signals such as electromyograms (EMG) to determine the intended movement. Further, highly-enabled prosthetic arms have been sensorized to provide the amputee sensory stimulation, such as those involving tactile sensation, for closed-loop control of the artificial limb [1]. Advanced prostheses rely on human movement intent decoders to interpret the biological signals and estimate the desired movement. In current practice [2]–[5], the decoder parameters are estimated during a training phase and kept frozen during the operational phase. The human body is a time-varying system, and consequently the movement decoders must be retrained frequently. This paper presents a new approach to update the decoder parameters by adapting them during the operational phase.

Motor intent decoders based on Kalman filters (KF) [2], [4], [6]–[12] have become very popular due to its simplicity and relative low computational costs. In recent years, machine learning-based approaches [3], [13]–[17] have shown substantially improved performance over Kalman decoders. Dantas *et. al.* [3] have shown that multi-layer perceptron networks (MLP) trained using dataset aggregation (DAgger) [18] produce smooth, natural, and precise output trajectories. Sussillo et al. [13] proposed the use of a recurrent neural network as a neural decoder. Radial basis networks have also been used for this purpose [14]. Chen et al. [19] presented a practical implementation of neural decoders based on extreme machine learning. Further, some studies have shown that the incorporation of a higher-level goal to the decoder can yield a more natural trajectory and assist the people with limb loss with movements of the prosthetic hands [5], [20], [21].

Although these methods have shown performance improvement immediately after training, they all suffer from performance degradation over the course of time due to the dynamic nature of the human body. Consequently, frequent retraining is needed to overcome this degradation. Retraining such systems, in an online manner, is challenging, mostly because of the absence of knowledge about the movement intent in real time. If movement intent were known, online learning algorithms could be used to adapt the system parameters. A possible solution this is to check most recent decoded movements against known movement patterns. If a movement pattern is found, such information may be used to retrain the decoder. Tadipatri *et. al.* [22] used similar idea to retrain relevance vector machines (RVM) assuming that the decoder output follows a straight line on 2-D center-out tasks. This approach might not be realistic for highly enabled prostheses, which can have complex movements and a high number of degrees of freedom (DoF) to control.

In this work, we use a Markov decision process (MDP)-based framework similar to the one used in Dantas *et. al.* [3], [23] to train an MLP-based decoder using the DAgger algorithm to produce precise decoders. During the operational phase, we assume a movement model for each degree of freedom of the limb. We use such movement models to recreate a desired trajectory, which is then used to update the decoders parameters based on the difference between the actual trajectory and the estimated desired trajectory in a semi-supervised manner. We present a gradient approach that can be used to update the parameters of any neural network-based decoder including deep networks. We use EMG signals recorded from one amputee over 11 months to show that our algorithm reduces the long-term performance degradation of the decoder. These results demonstrate the ability of the method to accurately decode EMG signals and keep acceptable performance levels at all times.

II. Prosthetic Controller Design

Broadly, the function of any decoder is to interpret the biological signals and decide the best movement for the prosthetic limb. We wish to estimate a probabilistic description $\pi_\theta(u_k|s_k)$ of the control signal u_k at the kth time step, given s_k, the system state and the biological signals at time step k. We follow steps similar to those described by Dantas *et. al.* [3], [23] to derive this recursive prediction problem. We present the framework to estimate the decoder parameters first in an offline manner and then describe the algorithm used to update these parameters online during the operational phase.

A. Offline Training

The state s_k is defined as the union of the most recent H_1 instances of the measured EMG signals $Z_k = [z_{1,k}, ..., z_{N,k}]^T$ and the most recent H_2 instances of the position of the prosthetic hand $X_k = [x_{1,k}, ..., x_{M,k}]^T$. Here $z_{i,k}$ is the kth measurement from the ith measurement channel, N is the number of EMG channels, $x_{j,k}$ is the hand position at time k corresponding to the

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

jth DoF, and M is the number of DoFs of the hand. That is, $s_k = [Z_k, ..., Z_{k-H_1-1} \bigcup X_k, ..., X_{k-H_2-1}]$. The control signal is defined by $u_k = [X_{k+1}]$.

We assume that the system evolves according to the Markov assumption, where the next state of the hand, s_{k+1}, only depends on the current state s_k, *i.e.*, $p(s_{k+1}|s_k, ..., s_1) = p(s_{k+1}|s_k)$. For a desired trajectory $\tau = \bigcup_{i=1}^{H-1}(s_i, u_i) \bigcup s_H$, where H is the number of samples in the trajectory and $H > 1$, it is possible to write $p(\tau)$, the probability of the system following the desired trajectory, in a parameterized form $p_\theta(\tau)$ in the following manner $p_\theta(\tau) = p(s_1) \prod_{i=1}^{H-1} p(s_{i+1}|s_i, u_i)\pi_\theta(u_i|s_i)$.

As described in [3], if we assume $\pi_\theta(u_i|s_i)$ to be a Gaussian distribution, we can write the cost function as

$$\nabla_\theta J(\theta) = \frac{2}{H-1} \sum_{i=1}^{H-1} \left[[u_i - \phi_\theta(s_i)][\nabla_\theta \phi_\theta(s_i)]^T \right]^T \quad (1)$$

where ϕ_θ represents a possibly-nonlinear model of the control signal (*i.e.*, the decoder output) and is completely specified by the vector of parameters θ. The parameter vector θ that maximizes the probability of the system following a trajectory τ can be estimated using a gradient ascent framework for the jth iteration as

$$\theta_{j+1} = \theta_j + \alpha_1 \nabla_{\theta_j} J(\theta_j) \quad (2)$$

where α_1 is a positive constant and controls the learning rate during the training phase and ∇_θ represents the gradient with respect θ. During the training phase, for a given trajectory τ, the decoder ϕ_θ can be trained using (1) and (2) using the DAgger algorithm described in [3]. During the training phase, u_k corresponds the next kinematic position in the training data. During operation phase, this data is replaced by estimates of hand kinematics produced by the decoder, $\hat{u}_i$.

In the experiments and analyses described in Section III, we assumed that the decoder is parameterized as a MLP, however, the derivation is equally applicable to convolutional neural networks (CNN) and long short-term memory (LSTM)-based decoders.

B. Online Adaptation

A block diagram of the adaptive learning algorithm is shown in Figure 1. In order to update the parameters of the decoder model during normal operation, we assume a specific movement model for each DoF of the hand. The system also assumes a memory buffer for the last H_3 kinematic samples and, another buffer containing the last H_3 EMG data samples. At each time sample, the system first looks for specific movement patterns in the kinematic memory buffer. If a movement pattern is found for a DoF, the kinematic for that DoF are estimated and used as the desired movements by the parameter update algorithm. Using similar steps as described earlier, we can derive the gradient to update the parameter vector θ as follows:

$$\nabla_\theta J_{online}(\theta) = \frac{2}{H_3} \sum_{i=1}^{H_3} \left[[u'_k - \hat{u}_i][\nabla_\theta \phi_\theta(s_i)]^T \right]^T \quad (3)$$

In this equation u'_k is the estimated kinematic of the hand computed using the movement model. Finally we update the decoder parameters in the jth iteration using the following equation:

$$\theta_{j+1} = \theta_j + \alpha_2 \nabla_{\theta_j} J_{online}(\theta_j) \quad (4)$$

Here α_2 is a positive constant and controls the adaptive learning rate.

Algorithm 1 describes the operation of the adaptive decoder, assuming that the system was initially trained. This approach assumes a movement model that can be completely specified by a finite number of parameters, the parameters θ of the decoders are adapted according to the movement model. It is important to notice that the above derivation can be used for adapt any decoder using a variety of movement models. In Section III-B, we detail the movement model used in this work.

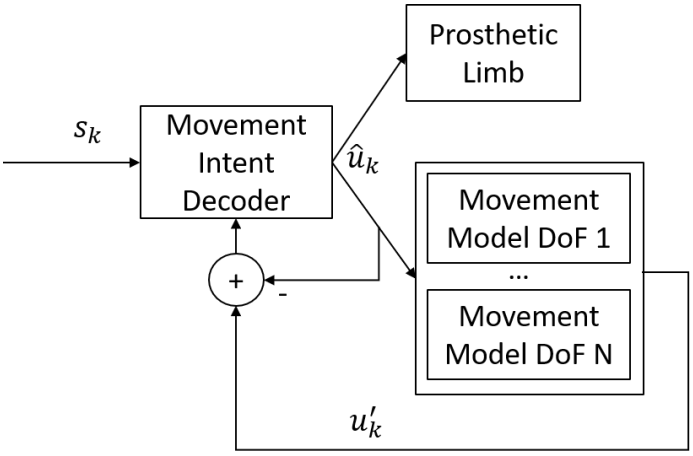

Fig. 1. Block diagram of the semi-supervised adaptive controller.

Initialize X_1 at the desired position
foreach *time sample* i **do**
  Create $\hat{s}_i = [Z_i, ..., Z_{i-H_1} \bigcup \hat{X}_i, ..., \hat{X}_{i-H_2}]$
  Run the decoder $\phi_\theta(\cdot)$ to estimate control signal: $\hat{u}_i = \phi_\theta(\hat{s}_i)$
  Update the state of the arm $\hat{X}_{i+1}$ based on the estimated control signal $\hat{u}_i$. In this case, $\hat{X}_{i+1} = \hat{u}_i$
  Update the kinematic and EMG memory buffers with $\hat{X}_i$ and Z_i
  Check if a known movement pattern is present in the kinematic buffer
  if *Movement Pattern Found,* **then**
    Update the parameters θ using equations (3) and (4)
end

Algorithm 1: Pseudo-code for adapting the parameters of the movement intent decoder

III. Experiments

A. Experiment Setup

The results presented here are from a single amputee subject, referred here as HS2. After approvals from the University of Utah and Department of the Navy Human Research Protection Program Institutional Review Boards, and receiving informed consent from the subject, he was implanted with 32 EMG electrodes to acquire intramuscular EMG data. The subject was also implanted with two 96-electrode Utah Slanted Electrode Arrays [24] in the ulnar and median nerves of their residual arm, but these devices were not used in this analysis. The thirty two single-ended EMG signals were acquired at 1-KHz sampling rate by a Grapevine NIP system (Ripple, Salt Lake City, UT) using proprietary front-end hardware. These signals were filtered with a 6th-order Butterworth high-pass filter with 3 dB cut-off frequency at 15 Hz, a 2nd-order Butterworth low pass filter with 3 dB cut-off frequency at 375 Hz, and 60, 120, and 180 Hz notch filters. More information about the implants can be found in [25]. Differential EMG signals for all 496 possible combinations of the 32 single-ended channels were calculated in software. For

each of the single ended and differential EMG channels, the mean absolute value was calculated over a 33.3-ms window of time and subsequently smoothed with a 300-ms rectangular window. To reduce the dimensionality and the computational complexity of the decoder, principal component analysis was performed on the EMG data in the training data set [2]. The first sixteen principal components (PCs) were used as decoding features to the MLP-based decoder with and without adaptation. The first thirty two PCs were used as input features to the KF-based decoder. These configurations yielded the best decoding output for each method analyzed in this work.

The subject was instructed to track the movement of a simulated hand with his phantom limb while the EMG signals were recorded. The instructed movement followed a semi-sinusoidal path at a velocity deemed comfortable by the subject. Only movements of a single DoF was instructed during the experimental sessions. For each DoF, ten trials of each movement were performed, including flexion and extension. The range of the DoFs movements is limited to between $[-1, 1]$, where -1 represents full extension and 1 represents full flexion of a DoF. The resting position is represented by the zero value. To train multi-DoF decoding methods, movement trials for different DoFs were concatenated. Data from the five digits were used in these analyses. Fifty-seven datasets from same subject HS2 were used here. We measured the decoder performance variations over L days by evaluating the decoder performance on day n, when the decoder was originally trained on day $n - L$. The maximum separation between training and testing sessions used in this paper was 45 days.

In this work, we used three distinct decoders: KF-based decoder, MLP network-based decoder without adaptation and an MLP network-based decoder with adaptation. We selected the KF-based decoder due to its popularity in the literature. The MLP-networks without adaptation performs significantly better than the KF-based decoder, and acts as a good baseline to demonstrate the effectiveness of the adaptive decoder. For the MLP-based decoder with adaptation, the parameters update were performed every time sample when a movement pattern was detected. For the other decoders, no adaptation was employed.

B. Movement Model

The movement model is an important part of the algorithm presented in this paper. In the model used here, we assume that a DoF movement contains five parts: (1) A initial resting phase where the DoF is in the resting region defined by the interval $[-0.1, 0.1]$; (2) A rising or falling phase, depending on if the movement is a flexion or an extension, where the DoF is moving to its final position. This phase starts at time T_1 and ends at time T_2; (3) A holding phase, where the DoF is near the final target A. There may be some jitter during the holding phase. This phase starts at time T_2 and finishes at time T_3; (4) The falling or rising phase, depending if the movement is a flexion or extension, where the DoF is moving back to the resting zone. This phase starts at time T_3 and end at time T_4; and (5) The final resting phase, when the DoF is back to the resting position. These five phases in the movement along a DoF are depicted in Figure 2.

In order to use this movement model, the system must estimate the transition points T_1, T_2, T_3, and T_4, and estimate the movement amplitude A. We assumed that the desired movement follows a piecewise-linear model as in Figure 2, and desired trajectory, in green, was estimated using linear regression. The algorithm to estimate the model parameters followed the steps: (1) The starting time T_1 was estimated as the time at which the decoded output took the closest value to zero in the five time samples prior to when the decoded output was grater than 0.1 for a flexion movement or less than 0.1 for an extension movement; (2) The transition time T_2 was estimated as the time when the second derivative of the decoder output was zero or negative for 2 consecutive time steps for a flexion movement or the second derivative was equal to zero or positive for 2 consecutive time steps for a extension movement; (3) Check if the newest sample in the buffer is in the resting zone, if the DoF is in the resting zone we estimate time T_4 as the time when the kinematic signal took the closest value to zero in the five time samples after the decoded output had values between $[-0.1, 0.1]$; (4) If T_4 was successfully estimated, we estimated the transition time T_3 using the second derivative test, by traversing from the newest samples to the oldest samples. The transition time T_3 was estimated as the time when the second derivative of the decoder output was zero or negative for 2 consecutive time steps for a flexion movement or the second derivative was equal to zero or positive for 2 consecutive time steps for a extension movement; (5) The movement amplitude, A, was estimated as the average of the output decoder values between T_2 and T_3. (6) We checked if $T_1 < T_2 < T_3 < T_4$ to ensure the sanity of the estimated parameters, and also assumed that no full movement was performed at a faster rate than 2 Hz, implying that $T_4 - T_1 > 0.5$ seconds. If such conditions were met the decoder was updated with the estimated movement.

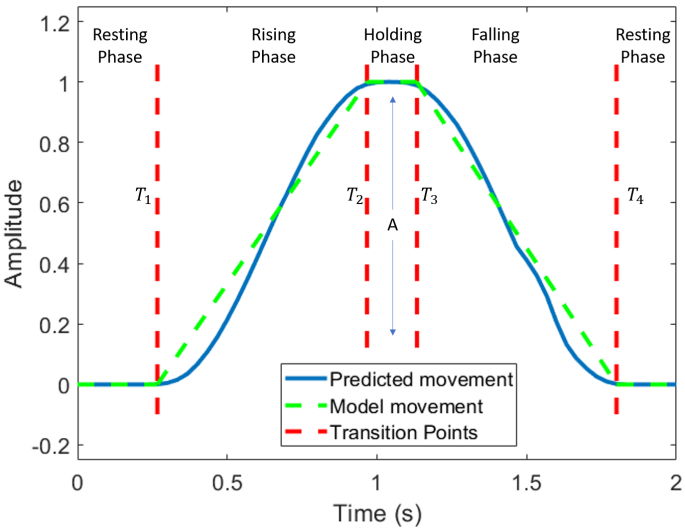

Fig. 2. Sample movement of a full flexion. Desired movement based on the movement model, in green, over the decoded position of a DoF, in blue.

IV. Results

The adaptive decoder was implemented in an offline manner to obtain the results presented here. The performance metric used in the analyses was the normalized mean-square error (NMSE) defined as $MSE_{normalized} = \frac{\sum_{k=1}^{H} ||X_k - \hat{X}_k||^2}{\sum_{k=1}^{H} ||X_k||^2}$, where $||\cdot||^2$ denotes the Euclidean norm of $(\cdot)$, X_k is the desired kinematic state, $\hat{X}_k$ is the decoded kinematic state, and H is the number of samples in the test dataset.

To investigate the robustness of the three competing decoding methods over long periods of time, we analyzed the decoding performance up to 45 days separation between the training and testing session. Figure 3 displays the normalized mean-square decoding error for the three methods along with their standard deviations as a

function of the number of days between the acquisition of the training and testing data.

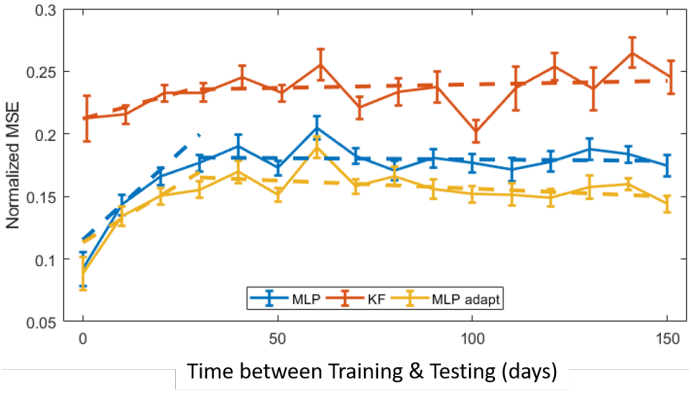

Fig. 3. Normalized MSE for KF, MLP, and MLP with adaptation-based decoders as a function of the days between training and testing. The solid lines present the mean of the NMSE in a five-day block, and the dashed lines represents the linear fit performed to detect the trend.

In order to investigate the statistical significance of the relative performance of each decoder along time, we employed a two-factor, repeated measures analysis of variance test (RANOVA). The two factors were the decoding method and the time between the dates of training and testing sessions. In addition, a multiple comparison post hoc test using Tukey's honest significant difference correction was performed if the RANOVA test indicated with statistical significance that at least one among the three systems compared were different from the others.

Initially, we investigated if the three decoders had performances that differ in a statistically significant manner. The three competing decoding methods differed via a RANOVA within subject test ($p < 0.001$), The system with the best performance was the MLP network-based decoder with adaptation, which significantly outperformed the other methods ($p < 0.001$). The second best method was the MLP network-based decoder without adaptation, which significantly outperformed the KF-based decoder ($p < 0.001$). Using RANOVA between subject test, the time categories significantly differed ($p = 0.013$) but no trend in performance was seen, when averaging across all decode methods

We selected not to study the interaction term between decode method and time category but instead studied the slopes of each decode method across time to investigate how the performance changed over time. We performed a linear fit in the data to analyze the trend of the NMSE over time for each method. The MLP network-based decoder without adaptation resulted in a slope in the NMSE of 0.0014 normalized units per day, and this slope was found to be statistically significantly from the zero slope ($p < 0.03$). The MLP network-based decoder with adaptation had a slope in the NMSE of 0.0006 normalized units per day, this slope was found to be statistically significant different from zero slope ($p < 0.03$). Finally, there was no statistical evidence that the slope associated with the KF-based decoder was different from zero ($p > 0.09$).

The linear fitting analyses suggested that the performance of the MLP-based decoder with adaptation and the MLP-based decoder without adaptation changed in a different manner. The KF-based decoder demonstrated no evidence of performance change across the 45 days between training and testing. The MLP network without adaptation showed the largest performance degradation. Finally, the results suggested that the adaptive decoder was able to reduce the slope of the performance degradation by 57% over the MLP-based decoders without adaptation. Despite the observed degradation, both MLP networks with and without adaptation were able to outperform the KF-based decoder throughout the time frame analyzed.

V. Conclusion

In this work, we investigated the feasibility of a semi-supervised adaptation framework based on a piece-wise linear model that can be used by neural networks-based decoder. We used a model of the DoFs movements to provide feedback to the neural network during the operational phase and based on this feedback we were able to adapt the MLP network. The presented results showed that the adaptation algorithm improved the decoder stability over long periods of time for the MLP network-based decoder. Our results suggest that the decoder was able to adapt to the changes in the EMG signals over time to maintain an acceptable decoding performance over time. In addition, our algorithm may not work for systems that present abrupt changes, for example, changes due to broken wires or different sensor placement for surface EMG signals. Finally, we anticipate that the performance of the adaptation framework may degrade if the initial decoder estimates an elevated number of undesired cross-movements, because these cross-movements will be used to retraing the decoder and this would encode wrong information in the decoder. A topic for future work is to study cross-movement detection.

Experiments to validate the decoder performance on additional humans subjects, for longer periods of time between the training and testing sessions, and different movement models are currently underway. Further studies using deep architecture models will be also explored.

Acknowledgment

This work was supported in part by National Science Foundation (NSF) Grant No. 1533649 and in part 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.

References

[1] L. Resnik, S. L. Klinger, and K. Etter, "The DEKA arm: Its features, functionality, and evolution during the Veterans Affairs Study to optimize the DEKA arm," *Prosthetics and Orthotics International*, vol. 38, no. 6, pp. 492–504, 2014, pMID: 24150930.

[2] 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.

[3] H. Dantas, V. J. Mathews, S. M. Wendelken, T. S. Davis, G. A. Clark, and D. J. Warren, "Neural decoding systems using markov decision processes," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, March 2017, pp. 974–978.

[4] H. Dantas, S. Kellis, V. Mathews, and B. Greger, "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.

[5] H. Dantas, J. Nieven, T. S. Davis, X. Fu, G. A. Clark, D. J. Warren, and V. J. Mathews, "Shared human-machine control for self-aware prostheses," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, April 2018, pp. 6593–6597.

[6] 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, Mar. 2006.

[7] S.-P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, and M. J. Black, "Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia," *Journal of Neural Engineering*, vol. 5, no. 4, p. 455, Jul. 2008.

[8] 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 *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, May 2017, pp. 432–435.

[9] Z. Li, J. O'Doherty, and T. Hanson, "Unscented Kalman filter for brain-machine interfaces," *PloS One*, vol. 4, no. 7, pp. 1–18, Jul. 2009.

[10] T. S. Davis, H. A. C. Wark, D. T. Hutchinson, D. J. Warren, K. O'eill, T. Scheinblum, G. A. Clark, R. A. Normann, and B. Greger, "Restoring motor control and sensory feedback in people with upper extremity amputations using arrays of 96 microelectrodes implanted in the median and ulnar nerves," *Journal of Neural Engineering*, vol. 13, no. 3, p. 036001, 2016.

[11] S. Wendelken, D. M. Page, T. Davis, H. A. C. Wark, D. T. Kluger, C. Duncan, D. J. Warren, D. T. Hutchinson, and G. A. Clark, "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.

[12] J. George, M. Brinton, C. Duncan, D. Hutchinson, and G. Clark, "Improved training paradigms and motor-decode algorithms: results from intact individuals and a recent transradial amputee with prior complex regional pain syndrome," in *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Honululu,HI,USA, April 2018.

[13] D. Sussillo, P. Nuyujukian, J. M. Fan, J. C. Kao, S. D. Stavisky, S. Ryu, and K. Shenoy, "A recurrent neural network for closed-loop intracortical brain-machine interface decoders," *Journal of Neural Engineering*, vol. 9, no. 2, p. 026027, 2012.

[14] M. S. Islam, M. S. Khan, H. Deng, and K. A. Mamun, "Decoding movements from human deep brain local field potentials using radial basis function neural network," in *27th IEEE International Symposium on Computer-Based Medical Systems*, May 2014, pp. 105–108.

[15] M. Chen and P. Zhou, "A novel framework based on fastica for high density surface emg decomposition," *IEEE Trans. Neural Syst. Rehabil. Eng*, vol. 24, no. 1, pp. 117–127, Jan 2016.

[16] M. Atzori, M. Cognolato, and H. Muller, "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, p. 9, 2016. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnbot.2016.00009

[17] J. He, D. Zhang, N. Jiang, X. Sheng, D. Farina, and X. Zhu, "User adaptation in long-term, open-loop myoelectric training: implications for emg pattern recognition in prosthesis control," *Journal of Neural Engineering*, vol. 12, no. 4, p. 046005, 2015. [Online]. Available: http://stacks.iop.org/1741-2552/12/i=4/a=046005

[18] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, vol. 15. Journal of Machine Learning Research - Workshop and Conference Proceedings, 2011, pp. 627–635.

[19] Y. Chen, E. Yao, and A. Basu, "A 128-channel extreme learning machine-based neural decoder for brain machine interfaces," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 679–692, June 2016.

[20] G. H. Mulliken and A. R. A. Musallam, Sam, "Decoding trajectories from posterior parietal cortex ensembles," *The Journal of Neuroscience*, vol. 28, pp. 12 913–12 926, Nov 2008.

[21] G. Hotson, R. J. Smith, A. G. Rouse, M. H. Schieber, N. V. Thakor, and B. A. Wester, "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.

[22] V. A. Tadipatri, A. H. Tewfik, G. Pellizzer, and J. Ashe, "Overcoming long-term variability in local field potentials using an adaptive decoder," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 2, pp. 319–328, Feb 2017.

[23] H. Dantas, D. J. Warren, S. Wendelken, T. Davis, G. A. Clark, and V. J. Mathews, "Deep learning movement intent decoders trained with dataset aggregation for prosthetic limb control," *IEEE Transactions on Biomedical Engineering*, pp. 1–1, 2019.

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

[25] D. M. Page, J. A. George, D. T. Kluger, C. Duncan, S. Wendelken, T. Davis, D. T. Hutchinson, and G. A. Clark, "Motor control and sensory feedback enhance prosthesis embodiment and reduce phantom pain after long-term hand amputation," *Frontiers in Human Neuroscience*, vol. 12, p. 352, 2018.