NEURAL DECODING SYSTEMS USING MARKOV DECISION PROCESSES

Henrique Dantas[1], *V John Mathews*[1], *Suzanne M. Wendelken* [2], *Tyler S. Davis*[3]
Gregory A. Clark [2] *and David J. Warren*[2]

[1] School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA
[2] Department of Bioengineering, University of Utah, Salt Lake City, UT 84112-9458
[3] Department of Neurosurgery, University of Utah, Salt Lake City, UT 84132 USA

ABSTRACT

This paper presents a framework for modeling neural decoding using electromyogram (EMG) and electrocorticogram (ECoG) signals to interpret human intent and control prosthetic arms. Specifically, the method of this paper employs Markov Decision Processes (MDP) for neural decoding, parameterizing the policy using an artificial neural network. The system is trained using a modification of the Dataset Aggregation (DAgger) algorithm. The results presented here suggest that the approach of the paper performs better than the state-of-the-art.

Index Terms— Neural Decoding, Markov Decision Processes, Neural Network, Dataset Aggregation, Reinforcement Learning

1. INTRODUCTION

Loss of limbs results in profound changes in one's life. However the underlying neural circuitry and much of the ability to sense and control movements of their missing limb is retained even after limb loss. This means that amputee has the ability to control artificial limbs in a similar way to how the limb was controlled before the loss. This paper is concerned with decoding neural information for controlling prosthetic devices.

For a brief overview of motor system physiology in the context of neural prostheses, the reader is referred to [1]. Electrocorticogram (ECoG) signals, recorded at the cortical surface, strikes a balance between invasive measurement of spike events recorded from implants in the cortex and electroencephalogram (EEG), signals measured from the scalp, in terms of invasiveness and spatial resolution. Technically EMG is not a neural signal, but they are used in commercially available, state-of-the-art myoelectric prostheses. We limit the discussion in this paper to ECoG and EMG signals only. However, the concepts are easily applied to other neural signals also.

A number of methods are available in the literature for neural decoding of human intent. Some of these methods are nonlinear [1–6], but most are linear. Decoding algorithms employing linear generative models relating neural signals to the actual intended movements fall into broad categories of Wiener filters [7, 8], population vectors [9, 10], probabilistic methods [11, 12], and recursive Bayesian decoders such as Kalman filters [13, 14].

Several works available in the literature use of Markov Decision Processes (MDP) to apply reinforcement learning based on policy gradient methods [15–17] to learn motor tasks for robots. Such approaches have achieved state-of-the-art results in a number of applications. In the neural decoding field, a solution applying reinforcement learning based on Q-learning to solve Bellman's Equation in kernels domain was presented in [18]. In general, such methods require considerable tuning to achieve good performance, and this is not easy depending on the problem.

This paper presents a framework for neural decoding based on MDP. The initial training is performed using a supervised algorithm. Then, the data set is expanded using the outputs of the trained model and the system is trained further using Dataset Aggregation (DAgger) [19] approach. Several iterations of dagger may be performed. The framework presented here does not use rewards as the methods of [15–18]. In addition, the approach is capable of incorporating complex environmental dynamics to the model. The framework proposed here is tested for decoding electromyogram (EMG) and electrocorticogram (ECoG) signals to interpret human intent and control prosthetic arms. These results demonstrate the ability of the method to accurately decode neural signals, and suggest that our approach may be a better alternative to many decoders currently in the literature.

The rest of this paper is organized as follows: The next section describes the MDP framework for neural decoding. Section 3 describes the system implementation and also how DAgger was used to train the system. In Section 4, experimental work on a human subject to validate the decoder algorithm and assess its capabilities is described . Finally, the concluding remarks are provided in Section 5.

2. MARKOV DECISION PROCESS FORMULATION

It is common to model motor tasks using a Markov decision process [15, 16], where goal is to learn a policy $\pi_\theta(u_k|s_k)$, where s_k is the k^{th} state and u_k is an action for the k^{th} state, the policy is responsible to generate the next action. Furthermore, $\pi_\theta(u_k|s_k)$ can be interpreted as the probability of the system taking an action u_k give that the system state is s_k, and it is desirable that the action taken maximizes $\pi_\theta(u_k|s_k)$. The goal in this framework is to maximize the probability of the system reproducing a desired trajectory.

In the prostheses control problems, signals such as EMG contain incremental information about the movements, consequently the state of the system should contain motor electrical signals and arm kinematic state. We define the state (s_k) as the union of the most recent H_1 instances of the measured neural signals $Z_k = [z_{1,k}, ..., z_{i,k}]^T$, where $Z_{N,k}$ is the k-th measurement from the i-th neural channel, N is the number of neural channels, and the most recent H_2 instances of the state of the hand $X_k = [x_{1,k}, ..., x_{M,k}]^T$, where $x_{j,k}$ is the hand state value at time k corresponding to the j-th degree of freedom, and M is the number of degree of freedom of the hand. Thus, the state s_k and the action u_k are defined as

$$s_k = [Z_k, ..., Z_{k-H_1} \bigcup X_k, ..., X_{k-H_2}] \quad (1)$$

and

$$u_k = [X_{k+1}] \quad (2)$$

respectively. Results show that such formulations provide good results for EMG and ECoG signals.

We assume that the state s_k evolves according to a Markov property such that

$$p(s_{k+1}|s_k, ..., s_1) = p(s_{k+1}|s_k) \quad (3)$$

For a given trajectory $\tau = \bigcup_{i=1}^{H}(s_i, u_i)$, where H is the number of samples in the desired trajectory, it is possible to write $p(\tau)$ parameterized as $p_\theta(\tau)$ in the following manner:

$$p_\theta(\tau) = p(s_1) \prod_{i=1}^{H-1} p(s_{i+1}|s_i, u_i)\pi_\theta(u_i|s_i) \quad (4)$$

Unlikely many prior works this formulation allows the prosthetic system to be treated as a stochastic system.

For a given model, with parameter θ, the model is learned by optimizing the cost function:

$$J(\theta) = \frac{1}{H-1}\log(p_\theta(\tau)) \quad (5)$$

Since the log function is a monotonically increasing function, by maximizing the $\log(p_\theta(\tau))$, $p_\theta(\tau)$ is also maximized. The gradient of $J(\theta)$ is given by

$$\nabla_\theta J(\theta) = \frac{1}{H-1}\sum_{i=1}^{H-1} \nabla_\theta[\log \pi_\theta(u_i|s_i)] \quad (6)$$

where $\nabla_\theta(.)$ is the gradient with respect to θ. This results means that, no knowledge of the system dynamic $p(s_{i+1}|s_i, u_i)$ is needed to maximize the $p(\tau)$. In particular, we only need the policy $\pi_\theta(u_i|s_i)$ [16].

In the rest of this paper, we assume $\pi_\theta(u_i|s_i)$ is Gaussian function as give by Equation 7.

$$\pi_\theta(u_i|s_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{[u_i - \phi_\theta(s_i)][u_i - \phi_\theta(s_i)]^T}{2\sigma^2}} \quad (7)$$

$\nabla_\theta J(\theta)$ can then be written as

$$\nabla_\theta J(\theta) = \frac{1}{H-1}\sum_{i=1}^{H-1}\{[u_i - \phi_\theta(s_i)][\nabla_\theta \phi_\theta(s_i)]^T\}^T \quad (8)$$

The system parameters can be updated using Equation 9, where α is a positive constant in the interval $]0, 1]$ and controls the learning rate.

$$\theta = \theta + \alpha \nabla_\theta J(\theta) \quad (9)$$

The above derivation is quite general, and can be applied to a varieity of system models. Consequently, architectures such as multilayer perceptron (MLP) networks, convolutional neural networks and many more can be trained to learn the parameters of the system model $\phi_\theta(\cdot)$. In this work, we deal primarily with MLPs.

3. SYSTEM

The system deployed in this work is a simple multilayer perceptron (MLP) network. The MLP for the decoder employs four layers: the first layer transforms s_i into a flat vector in $\mathbb{R}^{1,(H_1N+H_2M)}$. The second and third layers have 16 nodes and employs a rectifier linear unit (Relu) activation function. The final layer is the output layer belonging to $\mathbb{R}^{1,M}$.

During training, the kinematic data represents the desired hand movements. In the experiments done on an amputee subject, the kinematic data was obtained from a virtual hand during training. During testing, this data was replaced by estimates of hand kinematics by the system model.

The neural network often has a large number of parameters, and learning a general model without over-fitting requires very large training sets. This is especially so for regret-based reinforcement learning [15,20], [21] and to a lesser extent, for non-regret-based reinforcement learning, in special imitation learning [22,23].

In this work, we use a modified version of the Dataset Aggregation (DAgger) approach [19] to train the neural network. DAgger provides a compromise between the complexity of training and the overall performance. In the first iteration the system is trained using back propagation using the neural data and the intended movement prior to training. In subsequent iterations, the training data is augmented based on the decisions made by the system and the known intended movement.

The modified DAgger approach is described using a pseudocode in Algorithm 1. During each iteration, a small amount of pseudo-random noise is added to the neural signal to mitigate problems with over-fitting the model. DAgger is known for its ability to correct training errors during early iterations.

```
Initialize $D \leftarrow (S, U)$;
Train policy ($\pi_\theta$) with $D$
for $i = 1$ to $L$ do
    Get $\hat{x_k}$ from performing sequence with $Z_k$
    Make $\hat{S} = [Z_k + noise, ..., Z_{k-H_1} + noise \bigcup \hat{X}_k, ..., \hat{X}_{k-H_2}]$
    Make $\hat{U} = U$
    Agregate datasets = $D \leftarrow D \bigcup (\hat{S}, \hat{U})$
    Train the Policy ($\pi_\theta$) with $D$
end
```

Algorithm 1: Modified DAgger Algorithm

4. EXPERIMENTAL RESULTS

In this section the experiments and the corresponding results are presented for both EMG and ECoG signals.

4.1. Experiment Setup

4.1.1. EMG Signals

The results presented here are for a single amputee subject. After receiving approvals from the Institutional Review Board and informed consent, the subject was implanted with 32 EMG electrodes to acquire intra-muscular EMG data. Originally the signals were recorded at sampling rate of 1 kHz. The tasks were performed in a virtual environment (Musculoskeletal Modeling Software [24]). The program modeled a virtual hand with 12 degrees of freedom (DoF) divided as follows: flexion and extension of each digit (5 DoFs), adduction and abduction of all except for the third digit (4 DoF), and wrist roll, pitch and yaw (3 DoF).

During training, 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 each training trial and multiple ($\geq$6 and $\leq$11) training trials of each movement were performed. To train multi-DoF decoding methods, movement trials for different DoF were concatenated.

Ten segments from the same patient were used here. Each segment contained ten trials for each DoF. In each segment, 70% of the available data was used for training and 30% for testing. The results presented here are the averages of the performance over the test data in the 10 segments.

4.1.2. ECoG Signals

After receiving approval from the institutional review board and informed consent from the patient, two 16-channels non-penetrating arrays (PMT Corporation, Chanhassen, MN) were implanted in a patient undergoing surgery to treat epilepsy. The arrays were implanted underneath a standard clinical electrocorticographic (ECoG) grid. The electrodes were placed in the arm and hand areas of the motor cortex. The patient was instructed to perform simple tasks such as reaching using a mouse on a tablet (20 cm x 20 cm). The patient moved the mouse from the bottom center of the tablet to the upper left or upper right corner of the table. All neural channels and the 2 position data (x and y coordinates) were recorded at a sampling rate of 30 kHz.

In this experiment, the data was divided into 7 segments. The system was trained on the first segment and tested on the remaining 6 segments. The results shown for ECoG are the averages of the decoder performance across the 6 testing segments.

4.2. Results and Discussion

The measure of performance used in this work is the normalized mean-square error. The normalized error is defined as:

$$MSE_{normalized} = \frac{\sum_{k=1}^{H}(X_k - \hat{X}_k)^2}{\sum_{k=1}^{H} X_k^2} \tag{10}$$

where X_k is the desired kinematic state, $\hat{X}_k$ is the estimated kinematic state, and H is the number of samples in the test data set.

The first set of results presented here aims to show the improvement of the decoder performance when DAgger algorithm is applied. These results also show the convergence behaviour of normalized decoding error when DAgger is applied. Finally, they also explore the decoder performance for different sizes of the MLP network. Figure 1 shows the effect of applying the DAgger algorithm for different sizes of multilayer perceptron network for EMG signals. We observe that, for all sizes of MLP the normalized MSE converges in a few number of iterations of DAgger. Furthermore, DAgger improves the MLP performance substantially over a single iteration of backpropagation. Augmenting the training set each iteration of DAgger allows the learning process to correct mistakes made in previous iterations. If the system deviates from the optimiumal trajectory τ it will attempt to get back to the trajactory by considering the errors that have accumulated in the augmented training set. In this particular example, the optimum size of MLP employed 16 nodes in the second and third layers. As the size of MLP increased, the system tends to overfit, causing larger errors with the test data set. Once the system performance converged, states visited in earlier DAgger iterations were discarded as new ones were aggregated. Similar results as described for EMG signals were observed for the ECoG signals.

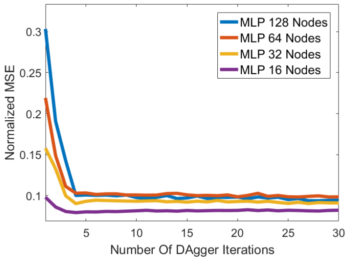

Fig. 1. Error evolution as a function of the number of iterations in DAgger for EMG signals.

Finally, we compared the performance of the system presented here with a Kalman decoder. This system was presented in [6], makes use of a polynomial nonlinear transformation of the observation vector used in the linear model originally reported in [13]. Figures 2 and 3 display the output of the MLP decoder trained with the method of this paper and the Kalman Filter with quadratic nonlinearities as proposed in [6] for EMG and ECoG signals, respectively. For the EMG signals, the normalized MSE between the desired signal and estimated movements was 0.09 for the system of this paper and 0.17 for the system in [6]. For the ECoG signals, system of this paper resulted in a MSE of 0.11 while the system in [6] with a normalized MSE of 0.18. This suggests that training the system with DAgger can enhance the system's ability to learn models under conditions of uncertainty.

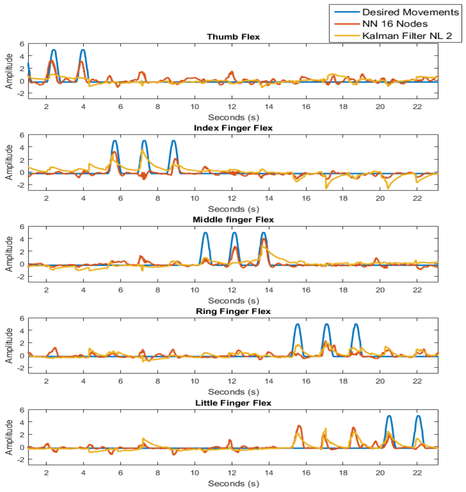

Fig. 2. Comparison of decoder performance for 5 DoF for EMG signals.

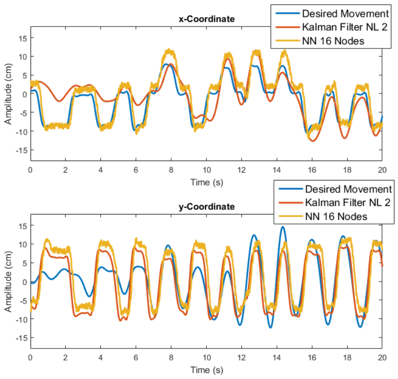

Fig. 3. Comparison of decoder performance for x and y coordinates for ECoG signals.

In general, for the EMG decoders, the results are better for movements of the thumb, the index finger and the middle finger, at least in part because of the better access to the muscles controlling these fingers in the amputee subject's arm, and partly because it very hard to control the ring and little fingers separately from each other. For the ECoG data, the system of this paper appear to decoded outputs with a small jitter. This can be addressed by processing the decoder output with a low pass filter and using the output of this filter for prosthetic control.

5. CONCLUSION

The preliminary results presented in this paper suggest that taking into account system dynamics and using DAgger to train a neural network improves the neural decoding performance. Experiments to validate the decoder performance on additional humans subjects are currently underway. Further studies on using deep architecture models and how to update the systems parameters online could enhance the decoder performance.

6. ACKNOWLEDGMENT

This work was supported in part by National Science Foundation (NSF) Grant No. 1533649 and in part by Defense Advanced Research Projects Agency (DARPA) BTO under the auspices of Dr. Doug Weber through the [Space and Naval Warfare Systems Center, Pacific OR DARPA Contracts Management Office] Grant/Contract No. N66001-15-C-4017. We gratefully acknowledge the support of NVIDIA Corporation for of the donation of the Tesla K40 GPU used in this research.

7. 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] Y. Gao, M. Black, E. Bienenstock, W. Wu, and J. Donoghue, "A quantitative comparison of linear and non-linear models of motor cortical activity for the encoding and decoding of arm motions," in *Proc. 1st Int. IEEE/EMBS Conf. Neural Eng*, March 2003, pp. 189–192.

[3] S.-P. Kim, J. C. Sanchez, Y. N. Rao, D. Erdogmus, J. M. Carmena, M. A. Lebedev, M. A. L. Nicolelis, and J. C. Principe, "A comparison of optimal MIMO linear and nonlinear models for brain-machine interfaces," *Journal of Neural Engineering*, vol. 3, no. 2, pp. 145–161, 2006.

[4] B. M. Yu and J. Cunningham, "Neural decoding of movements: From linear to nonlinear trajectory models," *Neural Information Processing*, pp. 586–595, 2008.

[5] 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.

[6] 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, May 2014, pp. 4683–4687.

[7] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361–365, Nov.

[8] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, Jul. 2006.

[9] A. P. Georgopoulos, R. E. Kettner, and A. B. Schwartz, "Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population," *The Journal of Neuroscience*, vol. 8, no. 8, pp. 2928–2937, August 1988.

[10] D. Taylor, S. Tillery, and A. Schwartz, "Information conveyed through brain-control: cursor versus robot," *IEEE Trans. Neural Syst. Rehabil*, vol. 11, no. 2, pp. 195–199, June 2003.

[11] W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. P. Donoghue, "Modeling and decoding motor cortical activity using a switching Kalman filter," *IEEE Trans. Biomed. Eng*, vol. 51, pp. 933–942, 2004.

[12] Y. Gao, M. J. Black, E. Bienenstock, S. Shoham, and J. P. Donoghue, "Probabilistic inference of hand motion from neural activity in motor cortex," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2002, pp. 213–220.

[13] 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.

[14] 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, pp. 455–476, Jul. 2008.

[15] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 156–163.

[16] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682–697, Sep. 2008.

[17] S. M. Khansari-Zadeh and A. Billard, "Bm: An iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2010, pp. 2381–2388.

[18] J. Bae, L. G. S. Giraldo, E. A. Pohlmeyer, J. T. Francis, J. C. Sanchez, and J. C. Príncipe, "Kernel temporal differences for neural decoding," *Intell. Neuroscience*, vol. 2015, pp. 1–17, Jan. 2015.

[19] 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)*, G. J. Gordon and D. B. Dunson, Eds., vol. 15. Journal of Machine Learning Research - Workshop and Conference Proceedings, 2011, pp. 627–635.

[20] J. Peters and S. Schaal, "Applying the episodic natural actor-critic architecture to motor primitive learning," Max-Planck-Gesellschaft. Evere, Belgium: D-Side, Apr. 2007, pp. 295–300.

[21] A. Y. Ng, H. J. Kim, M. I. Jordan, and S. Sastry, "Inverted autonomous helicopter flight via reinforcement learning," in *In International Symposium on Experimental Robotics*. MIT Press, 2004.

[22] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM, 2004, pp. 1–8.

[23] B. Schlkopf, J. Platt, and T. Hofmann, *Boosting Structured Prediction for Imitation Learning*. MIT Press, 2007, pp. 1153–1160.

[24] R. Davoodi, C. Urata, M. Hauschild, M. Khachani, and G. E. Loeb, "Model-based development of neural prostheses for movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1909–1918, Nov 2007.