

# Brain-Machine Interface Control via Reinforcement Learning

Jack DiGiovanna, *Student Member*, Babak Mahmoudi, *Student Member*, Jeremiah Mitzelfelt, *Student Member*, Justin C. Sanchez, *Member, IEEE*, and Jose C. Principe, *Fellow, IEEE*

**Abstract**— We investigate the capabilities of reinforcement learning (RL) to create a brain-machine interface (BMI) that uses  $Q(\lambda)$  learning to find the functional mapping between neural activity and intended behavior. This paradigm shift is intended to address the issue of paralyzed and amputee patients whom are physically unable to move, which is necessary to train traditional supervised learning BMIs. We created a RLBMI architecture incorporating a rat behavioral paradigm for prosthetic arm control. The performance results show ‘proof of concept’ that RLBMI can learn the temporal structure of neural signals to control a prosthetic arm.

## I. INTRODUCTION

BRain-machine interface (BMI) research has sought to develop technologies that restore sensorimotor function to patients suffering from stroke, paralysis, and other motor neuropathies. Traditionally, much BMI research has attempted to find functional relationships between neuronal activity and goal directed movements [1-3] using supervised learning (SL) techniques in an input-output modeling framework. This approach fundamentally requires knowledge of the inputs (neural activity) and desired signal (behavioral kinematics) to construct a functional mapping by minimizing the error between the model output and known behavior. While many groups have shown impressive results in both animal and human models, there is a fundamental limitation of this approach when designing BMIs applied in the clinical setting. A paralyzed patient is unable to generate a movement trajectory to serve as the desired signal.

To overcome the implementation limitations of SL, we propose here to use a semi-supervised technique, reinforcement learning (RL), to find the mapping between neural activity and behavior by maximizing the reward of completing a goal directed task. For the development of BMIs, the RL framework provides a mechanism of learning that is very similar to operant conditioning of biological

organisms [4] because the learner is not told what actions to take but must discover which actions yield the most reward by trying them [5]. Conceptually RL enables an agent to interact with their environment in a fashion that will maximize reward over time [5].

This paper presents our formulation of a RL architecture for BMI applications (RLBMI). We develop a semi-supervised technique for controlling the endpoint position of robot operating in a three-dimensional workspace to reach a set of targets. This experiment simulated the neural control of a prosthetic limb. Preliminary results of this approach using off-line multielectrode recordings from a behaving rat [6] will be presented. Finally limitations and implications of this work are discussed.

## II. METHODS

### A. Behavioral Paradigm

Multichannel neural firing rates from a rat were recorded during a go no-go behavioral task as described in [6]. The task was modified such that the animal initiates all trials with a nose poke breaking an IR beam. The animal can press a lever in the behavioral box when a LED cue is present. Additionally a robotic arm moves to target levers within the rat’s field of vision [7] but beyond the rat’s reach. To earn a reward the rat and robotic arm must press and hold their respective levers for 250 ms simultaneously. Performance of this task must exceed 80% before electrode implantation or the animal was excluded from the study. This criterion ensures that the animal is discriminating between cues and attending to both the levers and robot rather than guessing. Video analysis confirmed (> 92% trials) that the animal presses with the paw contralateral to the lever; ipsilateral neural firings reflect this difference in behavior.

We record single-unit firing times, reward times, and lever positions during the experiment. However, we do not record the kinematic variables of the animal’s arms. Therefore, this paradigm does not have sufficient information to train the control a robotic arm in 3-D space using SL techniques. Only the lever position (on/off) can be estimated via SL. The experimental paradigm is designed to simulate real clinical environments where the patient’s neural signal and rewards are known but kinematic variables are unavailable.

### B. Reinforcement Learning Architecture

In the classical embodiment of RL, the ‘agent’ is a model

This work was supported in part by the National Science Foundation under Grant #CNS-0540304, Children’s Miracle Network, UF Alumni Association Fellowship, Tarr Charitable Family Foundation, and the Tillie, Jennie & Harold Schwartz Foundation.

J. DiGiovanna and B. Mahmoudi are with the Department of Biomedical Engineering, University of Florida, 106 BME Building, Gainesville, FL 32611 USA (e-mails: [jfd134, babakm@ufl.edu])

J. D. Mitzelfelt is with Department of Neuroscience, University of Florida, Gainesville, FL 32611 USA (e-mail: jdmitez@ufl.edu)

J. C. Sanchez is with the Department of Pediatrics, Division of Neurology, University of Florida, P.O. Box 100296, JHMHC, Gainesville, FL 32610 USA (e-mail: jcs77@ufl.edu)

J. C. Principe is with the Department of Electrical and Computer and Biomedical Engineering, NEB 451, University of Florida, Gainesville, FL 32611 USA (e-mail: principe@cnel.ufl.edu)

of the animal conducting the task (through RL or other methods) to achieve a goal. The states of the environment and rewards gained teach the agent an optimal action policy.

In a BMI experimental paradigm, one has access to the environment, the actions, the rewards and also the real animal brain signals, i.e. one can observe the spatio-temporal activation of brain states (indirectly related to the environment) as the animal seeks a goal or reward. Relative to the “agent” (e.g. the BMI algorithm), neural activity is external to and can not be directly modified by the agent; hence it must be considered part of the environment [5]. The BMIRL agent uses information in the neural signal to create movement commands (actions) for the robot, it strives to learn the optimal neural state to action mapping.

We devise a novel architecture where the animal’s neural signal is part of the environment and defines the state, actions occur in a discrete physical space (a separate portion of the environment), and the RLBMI algorithm serves as the agent (see fig. 1). In closed-loop testing, the animal can see the actions of the robot; however, actions do not directly influence the state of the rat brain. This is an important shift in RL architecture because states are decoupled from actions. In this paradigm, we do not include robot position in the state variable because this information reduces the problem to a basic ‘grid-world’ [5] where neural signal would be ignored.

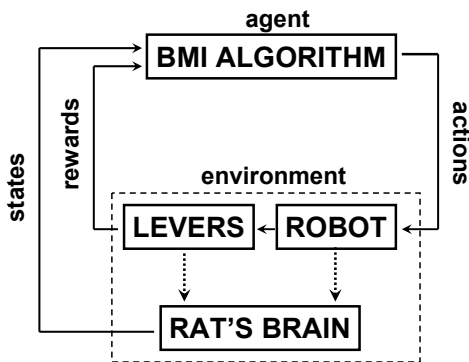


Fig. 1. Novel BMI RL Architecture

### C. Multielectrode Recordings

The rat was chronically implanted with two 16-microelectrode arrays (8 x 2 with 250  $\mu$ m spacing between the rows and 500  $\mu$ m spacing between the columns) in the forelimb regions of each hemisphere [8]. Single-unit firing times were recorded using threshold and template-based online spike sorting techniques [9]. We could discriminate 58 neurons (25 left, 33 right hemisphere) in this animal. Each neuron’s firing rate was estimated within non-overlapping 100ms windows [9], this defines the neural data.

As a control, the neural data was tested against a standard supervised learning I-O model (Wiener filter and a nonlinear threshold) to verify that there is a functional mapping to the behavioral task (lever pressing). This neural data could reconstruct the lever position with > 96% accuracy. This

suggests that they are related to the motor control task.

We emphasize that the following analysis is performed offline; the animal is unable to modulate its neural activity to influence the acquisition of rewards.

The null hypothesis is that RL can learn from random state presentations and solve the task. The spatial and temporal relationships in the neural data are randomized to create a surrogate data set; this set is used to test the null hypothesis.

### D. Neural Data Preprocessing

The evolution of dynamic patterns of activity in the cortex has been linked to temporally intermittent population synchronization/depolarization [10]. The transitions to excited states of neuronal activity, analogous to a change of phase, in the thermodynamic sense is indicative of changes in attention [11], motor intent, visual stimulation [12], and olfaction [13]. To define the ‘neural state’ of an animal we use the binned estimates of neural firing rates [9] from the forelimb area of primary motor cortex. While there are other methodologies for quantifying neural activity, there is experimental and neurophysiological support for the theory that the brain utilizes rate coding [1-3, 14]. Additionally, there is evidence that the motor cortex provides a representation of the ‘state’ of the motor control environment [15]; this supports our approach.

To define similar temporal sequences in neural states, the animal’s behavior was first segmented. For this animal, the average trial-start to lever press time was approximately 1200 ms. Segments (1200 ms) of neural data were extracted from reward-earning trials. Each segment was defined relative to the animal’s movement stop (pre-reward), the final 200 ms of the segments were excluded to account for nervous system to muscle conduction time. We excluded any trials shorter than 1200 ms. We also required equal left and right trials in the final dataset; the longest trials were excluded to balance the trial distribution.

### E. Value Function Estimation

RL is known to suffer from the ‘curse of dimensionality’ [5] and the number of possible firing rate combinations is intractable in neural data. RL also assumes that the state variable is a Markov representation [5], therefore instantaneous neural data must be emdedded in time to satisfy the Markov assumption. The gamma structure ( $K = 3$ ,  $\mu = 0.3$ ) was used to preserve 1s of firing rate history [16]. After experimenting unsuccessfully with clustering, we use the combination of the gamma memory with a multi layer perceptron (MLP) with architecture defined in Fig. 2 which contained hyperbolic tangent nonlinearities and linear output processing elements (PE).

The networks are trained on-line with temporal-difference error [5] and eligibility traces via back-propagation. The initial weights are set to small, random values. The estimated neural firing rate history is the input (state) to the network; each output PE represents the value of one action. 46 segments (trials) of the neural data are used to train the

neural networks, 16 segments are reserved for testing.

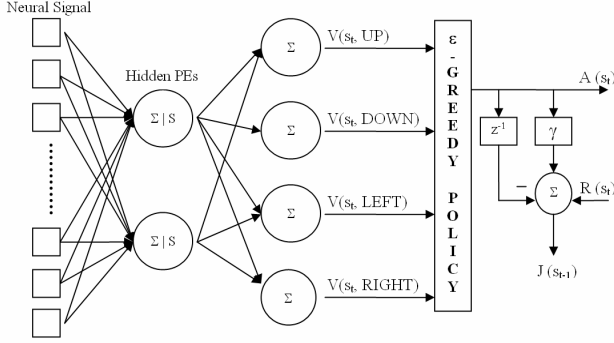


Fig. 2. Value Function Estimation.  $S_t$  is the state at time  $t$ .  $V$  is the state-action value.  $A$  is the action,  $R$  is the reward, and  $J$  is the temporal difference error. To save space only 4 actions are shown and PE bias terms are not shown

### F. Robotic Workspace Environment

The robotic environment is divided into discrete grid nodes. The two targets to reach are equidistant from the robot's starting position. There are 26 possible actions in this space: 1 unit in any single direction, 0.7 units in any direction pairs, and 0.6 units in any direction triples. The scales for multi-dimension moves are necessary to restrict all action vectors to the same length. In two (three) dimensions, a one-unit long vector at  $45^\circ$  to the axes, projects 0.7071 (0.5774) units on each axis.

When the robot reaches a target location a positive reward is generated, the trial ends, and the robot is reset to the initial position. Each action prior to reaching a lever generates a negative reward to encourage minimizing trial length.

To benchmark performance in this environment, the probability of randomly reaching the target in 10 steps is calculated by approximating the action selection as a Bernoulli random variable. The probability for the two and three dimension tasks are  $4.7e^{-7}$  and  $8.4e^{-8}$  respectively.

### G. Watkin's $Q(\lambda)$

$Q(\lambda)$  learning is an off-policy RL method that learns from sequences of state-action pairs (see eqn [1]). This algorithm is a proven technique that is appropriate for our architecture.

$$dQ(s_{t-1}, a, a) = \alpha[r_t + \gamma Q(s_t, a_t^*) - Q(s_{t-1}, a_{t-1})]e(s_{t-1}, a) \quad [1]$$

$Q$  is the estimated state-action value,  $dQ$  is the weight change in the network approximating  $Q$ ,  $s$  is the state,  $a$  is the action,  $\alpha$  is the learning rate,  $\gamma$  is the discounting factor,  $e$  is the eligibility trace, and  $\lambda$  is the trace-decay parameter (for details, see [5]). The algorithm follows an  $\epsilon$ -greedy policy. The value function is updated online; if RL parameters are appropriate,  $Q(\lambda)$  converges to the optimal policy [5].

### H. RL Training

We searched the RLBMI parameter space based on related literature and observations of learning performance. If  $\alpha$ ,  $H$ , and  $\lambda$  are selected appropriately, the RLBMI converges. Training performance was robust to parameter

selection; however, test set generalization was only possible with certain parameter combinations. The task is episodic so  $\gamma$  is set at 1. A range of  $\lambda$  centered around  $(1-1/\text{trial\_length})$  were tested to find the test set generalization.  $H$  was selected based on the number of principal components of the robotic trajectory as has been shown in other BMI literature [17].

Due to the randomness inherent in both  $Q(\lambda)$  and the initialization of value function estimation network weights, multiple training simulations are performed. The performance criterion ( $PR$ ) is the percentage of trials which reach the target. The average and best runs (when the RLBMI converges) are saved. The surrogate data is tested with the best neural data parameter set.

## III. RESULTS

Analysis of the test set performance of multiple runs showed that it was better to follow a greedy policy than to use exploration. The greedy policies'  $PR$  required more training epochs, but also had higher maximum  $PR$ . Using  $\epsilon$  (0.001 – 0.1) decreased the training epochs; however,  $\epsilon$  also decreases the maximum  $PR$ . These results suggest that there are many local minima in the performance surface; using  $\epsilon$  prevents finding better, but narrow, local minima. Test set performance of the RLBMI for two and three dimension grid environments is shown in Table 1.

TABLE 1: RLBMI TEST SET PERFORMANCE ( $PR$ ) - 10 RUNS

Data	Dimension	Max.	Mean	Standard Dev.
Neural	2	81.3 %	61.9 %	10 %
Neural	3	81.3 %	68.1 %	10.8%
Neural	3*	75 %	50 %	15.6 %
Surrogate	2	43.8 %	23.1 %	11 %
Surrogate	3	43.8 %	34.3 %	7.3 %
Surrogate	3*	43.8 %	31.8 %	8.56 %

\* 5% of trials were exploratory ( $\epsilon = 0.005$ ).  $\lambda = 0.8$ ,  $H = 3$  for 3D tests.  $\lambda = 0.8571$ ,  $H = 2$  for 2D tests

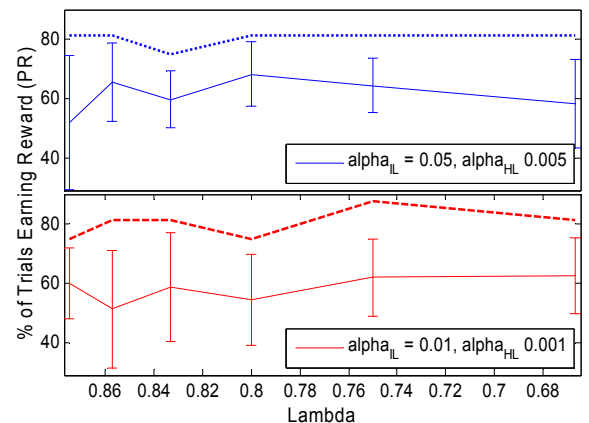


Fig. 3. RLBMI performance, 10 runs: average (solid) best (dashed).

Obviously we cannot claim to have exhausted all of the possible RLBMI parameter set combinations; however, fig. 3 shows that  $PR$  is fairly robust to  $\lambda$  and  $\alpha$  selection. We have found parameter sets that illustrate the potential of RLBMI, although they may not be ‘optimal’ parameters.

#### IV. DISCUSSION

This paper presents our preliminary RLBMI paradigm and tests its performance with rat neural data. Table 1 illustrates promising results that the RLBMI can achieve goal directed movements of prosthetic limbs 80% of the time. The RLBMI used off-line data; therefore, this implementation is only a ‘proof of concept’. However, this novel approach may be a significant step towards developing BMIs which do not require patient limb movements. RLBMI may provide an online mechanism through which a patient can modulate their neural activity to update the functional BMI mapping

To fully implement RLBMI; however, it is necessary to allow the animal interact with and get feedback from the system during this task. The RLBMI actions will indirectly influence future neural states. We hypothesize that visual feedback will improve RLBMI performance assuming that the animal learns the causal relation between its neural modulation and prosthetic limb motion (actions). However, this assumption must be tested with future research.

A potential advantage of RLBMI is that it can continue to update the state-action value estimates during testing while remaining causal. RLBMI does require ~250 training epochs to converge to a solution that generalizes. However, using an appropriate learning rate, the state-action value estimates can adapt to changes in patient’s neural modulation pattern. This adaptation potentially can also improve RLBMI performance in closed-loop testing.

The null hypothesis was included because in our prior experiments (unpublished), RL could solve some tasks using information from the model of the robot environment. This paradigm is specifically designed such that RL is only provided neural state information. By implementing no boundaries in the model of the robot environment, theoretically it should not influence RL.

The null hypothesis is disproved by the performance of the surrogate data. RLBMI can memorize the surrogate training data, but does not generalize to novel data. This suggests that RLBMI may exploit movement-related information present in the spatio-temporal activation of the neural signal.

In section II.D, the lever press-time was used to segment the neural data. The press-time is an unavailable signal from a paralyzed patient, however, the reward time in a more strictly controlled behavioral paradigm could be used. The press-time was used to exclude behavioral variability and focus on potentially task-related neural modulation.

To implement this algorithm online, it will be necessary to include a non-movement ‘action’ and create appropriate

reward signals for times when the prosthetic limb should not be moving. This state-action value would probably need to be trained with non-movement related neural-modulations and then incorporated into the full action set.

RLBMI appears to exploit information in the neural signal about the sequence of states in a limb trajectory. RLBMI does generate a sequence of actions, creating a trajectory in the prosthetic limb space. Although it may be desirable, there is no requirement that this artificial trajectory should match the kinematics of a natural limb. The path of the prosthetic limb is arbitrarily designated based on the reward distribution and must only match the timing of the natural limb trajectory. RLBMI potentially can learn a new trajectory for each neural modulation pattern.

#### REFERENCES

- [1] 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. L. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361-365, 2000.
- [2] 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, pp. 164-171, 2006.
- [3] E. C. Leuthardt, G. Schalk, D. Moran, and J. G. Ojemann, "The emerging world of motor neuroprosthetics: A neurosurgical perspective," *Neurosurgery*, vol. 59, pp. 1-13, Jul 2006.
- [4] G. H. Bower, *Theories of Learning*, 5th ed. Englewood Cliffs: Prentice-Hall, Inc., 1981.
- [5] A. G. B. Richard S. Sutton, *Reinforcement learning: an introduction*. Cambridge: The MIT Press, 1998.
- [6] J. DiGiovanna, J. C. Sanchez, and J. C. Principe, "Improved Linear BMI Systems via Population Averaging," in *IEEE International Conference of the Engineering in Medicine and Biology Society*, New York, 2006, pp. 1608-1611.
- [7] I. Q. Whishaw, *The behavior of the laboratory rat*. New York: Oxford University Press, Inc., 2005.
- [8] E. L. R. Hall, "Organization of the motor and somatosensory cortex in the albino rat," *Brain Research*, vol. 66, pp. 24-38, 1973.
- [9] M. A. L. Nicolelis, *Methods for Neural Ensemble Recordings*. Boca Raton: CRC Press, 1999.
- [10] W. J. Freeman, *Mass Action in the Nervous System: Examination of the Neurophysiological Basis of Adaptive Behavior Through EEG*. New York: Academic Press, 1975.
- [11] A. Rougeul, J. J. Bouyer, L. Dedet, and O. Debray, "Fast Somato-Parietal Rhythms During Combined Focal Attention And Immobility In Baboon And Squirrel-Monkey," *Electroencephalography And Clinical Neurophysiology*, vol. 46, pp. 310-319, 1979.
- [12] E. Niebur and C. Koch, "A Model for the Neuronal Implementation of Selective Visual Attention Based on Temporal Correlation among Neurons," *Journal of Computational Neuroscience*, vol. 1, pp. 141-158, 1994.
- [13] W. J. Freeman and B. Baird, "Relation of olfactory EEG to behavior: Spatial analysis," *Behav. Neurosci.*, vol. 101, pp. 393-408, 1987.
- [14] T. P. Trappenberg, *Fundamentals of Computational Neuroscience*. New York: Oxford University Press, 2002.
- [15] K. Doya, "What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?," *Neural Networks*, vol. 12, pp. 961-974, 1999.
- [16] J. C. Principe, B. De Vries, and P. G. Oliveira, "The gamma filter - a new class of adaptive IIR filters with restricted feedback," *IEEE Trans. Signal Processing*, vol. 41, pp. 649-656, 1993.
- [17] J. C. Sanchez, "From Cortical Neural Spike Trains to Behavior: Modeling and Analysis," in *Department of Biomedical Engineering Gainesville: University of Florida*, 2004.