Coadaptive Brain–Machine Interface via Reinforcement Learning

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

***Abstract*—This paper introduces and demonstrates a novel brain–machine interface (BMI) architecture based on the concepts of reinforcement learning (RL), coadaptation, and shaping. RL allows the BMI control algorithm to learn to complete tasks from interactions with the environment, rather than an explicit training signal. Coadaption enables continuous, synergistic adaptation between the BMI control algorithm and BMI user working in changing environments. Shaping is designed to reduce the learning curve for BMI users attempting to control a prosthetic. Here, we present the theory and *in vivo* experimental paradigm to illustrate how this BMI learns to complete a reaching task using a prosthetic arm in a 3-D workspace based on the user's neuronal activity. This semisupervised learning framework does not require user movements. We quantify BMI performance in closed-loop brain control over six to ten days for three rats as a function of increasing task difficulty. All three subjects coadapted with their BMI control algorithms to control the prosthetic significantly above chance at each level of difficulty.**

***Index Terms*—Brain–machine interface (BMI), coadaptation, neuroprosthetic, reinforcement learning (RL).**

I. Introduction

BIOLOGICAL organisms have the remarkable ability to interact with their environment and learn from experience. Insight into this ability has been advanced by analysis of *in vivo* neural ensemble recordings that have contributed to the development of computational theories of motor [1]–[4] and sensory system [5]–[8] function. Brain–machine interfaces (BMIs) provide a different perspective of functional mechanisms of motor intent because BMIs directly couple the central nervous system with engineered interfaces [9]–[12] in closed-loop motor control. The centerpiece of BMI experimental paradigms is the interpretation of brain processes involved in communication and control tasks for able bodied [13] or disabled individuals [14]. Often described as "decoding," [15] the process of discovering the functional mapping between neuronal activity and behavior has generally been implemented through two classes of learning: supervised [16] and unsupervised [17]. An unsupervised learning (UL) approach finds structural relationships in the data [18] without requiring an external teaching signal. A supervised learning (SL) approach uses kinematic variables as desired signals to train a (functional) regression model [19] or more sophisticated methods [20]. Both approaches seek spatiotemporal correlation and structure in the neuronal activity and fix model parameters after training. Fixing parameters provides a memory of the past experiences for future use, but suffers from the problem of generalization to new situations.

Neural interfaces that can also adapt to novel environments require experimental paradigms that go beyond translators of neural signals to kinematic variables. Here, we present a new BMI architecture that involves two coupled systems with the ability to model the environment: the *BMI user* and an artificial, intelligent *BMI control agent* that work in synergy. Unlike many previous BMI paradigms [21]–[23], both the user and the BMI control agent must coadapt [24], [25] and continuously learn from interactions with the environment.

The framework is based upon reinforcement learning (RL) which is a stochastic control methodology [26]. RL is a machine learning method inspired by operant conditioning of biological systems where the learner must discover which actions yield the most reward (are most beneficial) through trial and error [27]. RL originated from optimal control theory in Markov decision processes [26]; one of its strengths is the ability to learn which control actions will maximize reward given the environment's state [28]. It has been successfully applied to multiple fields including artificial intelligence in video games [29], robotic control [30], and dynamic channel allocation in telecommunications [31].

From a learning point of view, RL is considered a semisupervised technique [16], [26] because only a scalar training signal (reward) is provided after tasks, which is markedly different from SL. But perhaps more importantly, RL divides the task of learning into actions and the assessment of their values and this allows for modeling of the interaction with the environment. The appeal of RL for BMI design is centered on the facts that: 1) there is an implicit modeling of the interaction with the user; 2) an explicit training signal is not required, and 3) performance can continuously improve with usage. In fact, in many rehabilitation scenarios with paralyzed patients, the only available signals are the internal patient's intent to complete a movement task and external feedback if the task was accomplished. Hence,

Manuscript received November 7, 2007. First published June 13, 2008; current version published February 13, 2009. This work was supported in part by the U.S. National Science Foundation under Grant CNS-0540304, in part by the Children's Miracle Network, and in part by the University of Florida (UF) Alumni Association Fellowship. *Asterisk indicates the corresponding author.*

*J. DiGiovanna is with the Department of Biomedical Engineering, University of Florida, Gainesville, FL 32608 USA (e-mail: digiovaj@cnel.ufl.edu).

B. Mahmoudi is with the Department of Biomedical Engineering, University of Florida, Gainesville, FL 32608 USA (e-mail: babakm@ufl.edu).

J. Fortes and J. C. Principe are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: fortes@ufl.edu; principe@ufl.edu).

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

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBME.2008.926699

the RL-based BMI developed here attempts to learn a control strategy based on the BMI user's neuronal state and prosthetic actions in goal-directed tasks (i.e., reach targets in 3-D space) without guidance of which specific prosthetic actions are most appropriate [32]. The BMI control agent and BMI user both receive feedback after each movement is completed and only use this feedback to adapt the control strategy in *future* tasks [26].

This paper focuses on the design, theory, and testing of a novel RL-based BMI system (RLBMI). We develop a computational architecture and *in vivo* BMI experimental paradigm to show the performance of an RLBMI in goal-directed reaching tasks that parallel a paralyzed patient's goal of controlling a prosthetic. Performance is quantified by task completion accuracy and speed. Additionally, the ability to use past experience and adapt to novel situations is shown in a dynamically changing environment.

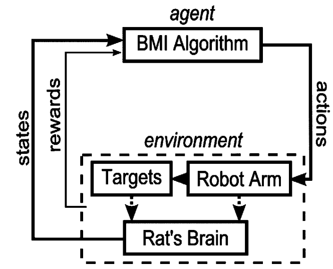


Fig. 1. RLBMI architecture with RL components labeled.

TABLE I
RL TASK FROM USER AND CA PERSPECTIVES

	User Perspective	CA Perspective
Agent	User	Control algorithm
Environment	Prosthetic & targets	User's brain
State	Prosthetic's position	User neural activity
Actions	Neural modulation	Prosthetic movement
Rewards	Task complete ($H_2$0)	Task complete (r_t)

II. METHODS

A. Computational Architecture

The conventional RL paradigm involves two entities: the *agent* and the *environment* [26]. The *agent* represents an intelligent being attempting to achieve a goal. The *environment* represents anything the *agent* cannot directly modify but can interact with. The interaction is defined by the *agent's* actions that influence the *environment* and the states and rewards observed from the *environment*. The agent's actions a_t are defined by the existing interface with the environment. The environment's state s_t is defined as a Markov descriptor vector [26]. After the agent completes an action, the environment provides a reward r_{t+1}. The agent attempts to maximize these rewards for the entire task—which is expressed as return R_t where r_n is the reward earned at time n and γ is a discounting factor ($\leq$1) that controls the horizon of future r_n that will be considered for the task.

The agent has no information about whether the selected actions leading to a reward were optimal at the time they were executed. Instead, the agent learns to estimate a value Q for the states and actions based on observed rewards. The optimal Q^* given by (2) is the expected return (sum of rewards) earned after time t given s_t and a_t. This estimation problem can be solved with techniques including dynamic programming (DP) and Monte Carlo (MC) estimation. RL provides an efficient approximation to either of these techniques because of its online learning [26]. Additionally, RL can be used without a model of the environment where DP cannot [26]:

$$R_t = \sum_{n=t+1}^{\infty} \gamma^{n-t+1} r_n \tag{1}$$

$$Q(s_t, a_t)^* = E\{R_t | s_t, a_t\}. \tag{2}$$

Our contribution is to model as a cooperative RL task the interaction of a paralyzed patient with an intelligent BMI prosthetic controller performing tasks in the environment both from the user's and the BMI's perspective. Users consider themselves the *agent* and act through the BMI to accomplish tasks (e.g., reach a glass of water) in the *environment* (e.g., the prosthetic, a glass of water). The user considers the positions of the prosthetic and the glass to be the environment's *state*. Since users cannot move, their *actions* are a high-level dialogue (neural modulations) with the BMI and the user may define *reward* as reaching the glass of water. The user seeks to learn a value for each action (neural modulation) given the relative position of the prosthetic (state) and the goal in order to achieve rewards.

The BMI controller defines the learning task differently. It considers itself the *agent* and acts through the prosthetic to accomplish tasks (e.g., reach the glass of water) in the *environment* (e.g., the user, the prosthetic). The BMI controller considers the environment's *state* to be the user's neuromodulation, where we assume the user's spatiotemporal neuronal activations reflect his or her intentions based on perception of the prosthetic. The BMI controller must develop a model of its environment (through observation of neuromodulation) to successfully interpret user *intent*. The BMI control agent's *actions* are movements of the prosthetic and *rewards* are defined in the environment based on the user's goals. In the ultimate implementation of a neuroprosthetic the goal states could be translated from the subject intent. However, it is necessary first to demonstrate the architecture's feasibility by providing the computational agent (CA) rewards based on the prosthetic position in the 3-D environment. These rewards should coincide with the user's goal (i.e., assign rewards for reaching the glass). The BMI controller seeks to learn values for each action (prosthetic movement) given the user's neural modulations (state) in order to achieve rewards.

The RLBMI architecture creates an interesting scenario where there are two "intelligent systems" in the loop. Both systems are learning to achieve rewards based on their own interpretations of the *environment*. The RLBMI must both facilitate prosthetic control for the user and adapt to the learning of both systems such that they act symbiotically. Fig. 1 shows this RL framework for BMI [32] and Table I summarizes the learning components from each perspective. We acknowledge that the user is also learning but focus on the design and testing

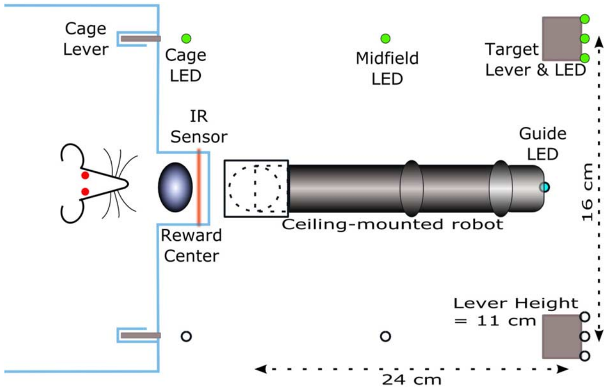


Fig. 2. RLBMI operating environment with dimensions.

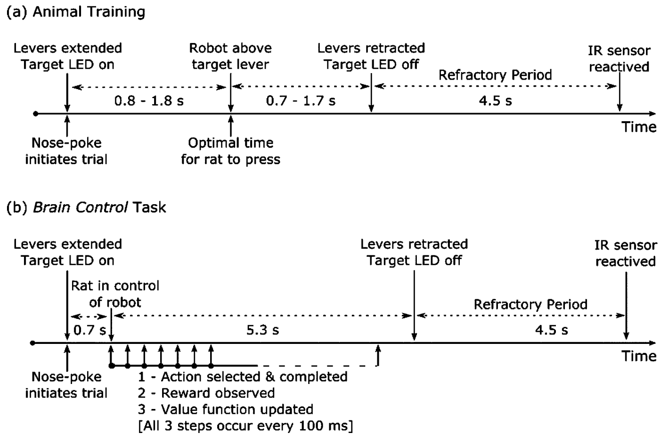

(a) Animal Training
Levers extended Target LED on
Robot above target lever
Levers retracted Target LED off
IR sensor reactived
0.8 - 1.8 s
0.7 - 1.7 s
Refractory Period
4.5 s
Time
Nose-poke initiates trial
Optimal time for rat to press
(b) Brain Control Task
Levers extended Target LED on
Levers retracted Target LED off
IR sensor reactived
Rat in control of robot
0.7 s
5.3 s
Refractory Period
4.5 s
Time
Nose-poke initiates trial
1 - Action selected & completed
2 - Reward observed
3 - Value function updated
[All 3 steps occur every 100 ms]

Fig. 3. (a) Rat training. (b) Brain-controlled two-target robot reaching task.

of the BMI controller; therefore, any future use of the term CA refers to the BMI control agent.

B. Experimental Paradigm and Rat Operant Conditioning

The experimental paradigm will be used to support the operant conditioning of the rat and closed-loop brain control of a robot arm using RLBMI. We designed a two-target choice task (shown from a top view in Fig. 2) as a rat model of a paralyzed patient that is seeking to control a prosthetic. The rat must maneuver a 5 DOF robotic arm (Dynaservo, Markham, ON, Canada) based on visual feedback to reach a set of targets and earn a water reward. The paradigm fits the RLBMI framework because both the rat and CA can earn rewards through interaction with their environments. Both "intelligent systems" are initially naïve in the closed-loop control task and must coadapt over multiple[1] trials to learn the tasks over multiple days (sessions) of training.

Male Sprague–Dawley rats were trained[2] in a two-lever choice task via operant conditioning to associate robot control with lever pressing[3] [27]. As shown in Fig. 2, the rat is enclosed in a behavioral cage with plexiglass walls. There are two sets of retractable levers (Med Associates, St. Albans, VT): the set within the behavioral cage is referred to as cage levers; the set in the robotic workspace is referred to as target levers. A solenoid controller (Med Associates) dispenses 0.04 mL of water into the reward center on successful trials. An IR beam (Med Associates) passes through the most distal portion of the reward center. There are three sets of green LEDs: the set immediately behind the cage levers are cage LEDs, the set in the robot workspace are midfield LEDs, and the set on the target levers are target LEDs. The positioning of the three sets of LEDs and levers offers a technique to guide attention from inside the cage to the robot environment outside. There is one additional blue LED mounted to the robot endpoint; it is referred to as the guide LED and it is used to assist the rat in tracking the position of the robot. Because the behavioral cage walls are constructed from plexiglass, the robotic workspace is within the rat's field of vision [33]. The workspace uses low-level lighting and is designed to maximize the rat's visual abilities. The target LEDs and guide LED provide contrast and targets are positioned to maximize the angle subtended to the rat's eye.

Initially, the robotic arm tip (guide LED) is positioned directly in front of the water reward center. The rat initiates a trial [see Fig. 3(a)] with a nose poke through the IR beam in the reward center. The target side and robot speed are randomly selected. All levers are extended synchronously and LEDs on the target side are illuminated to cue the rat. The robot follows a predetermined trajectory to reach the target lever within 0.8–1.8 s and the robot will only press the target levers while the rat is pressing the correct cage lever. If the correct cage and target levers are pressed concurrently for 500 ms, then the task is successfully completed; a water reward positively reinforces the rat's association of the robot lever pressing with reward and the trial is ended. If the rat presses the incorrect cage lever at any time, the trial is aborted, a brief tone indicates the choice was wrong, and there is a time-out (4–8 s) before the next trial can begin. Additionally, if the task is not completed within 2.5 s, the trial is ended. Whenever a trial ends: all levers are retracted, the LEDs are turned off, and the robot is reset to the initial position. A 4 s refractory period prevents a new trial while the rat may be drinking.

The rat initially seems aware of the cage levers only, and learns to press the correct lever to produce the water reward when all LEDs for a given side light up. The rat is then shaped to attend to the robot workspace by gradually moving the center of attention from within the cage to the robot workspace outside. This is achieved through turning off cage and midfield LED cues in sequence during training. The variable robot speed also encourages attention to the robot—the rat can minimize task energy by synchronizingly pressing with the robot. Eventually, the rat cues are reduced to the proximity of the guide LED to the target LED for completing the task and obtaining water. The rats learn to perform stereotypical motions for the environmental cues [33]. Barriers restrict access to cage levers such that rat only

[1] The number of trials depended on rat motivation and performance in each session; the range of trials per session was 86–236.

[2] Rats were motivated using a 21 h water withholding protocol approved by the University of Florida Institutional Animal Care and Use Committee (IACUC).

[3] The percentage of trials earning a reward is used to judge the rat's ability to discriminate between targets using visual cues and complete the task. The rat's accuracy must exceed an inclusion criterion of 80%. Rats incapable of this inclusion criterion within 25 days were excluded.

presses with the contralateral arm in a stereotypical fashion. The time-out and time limit both encourage correct behavior—rats can maximize water rewards earned by avoiding time-outs and unsuccessful trials. These measures to enforce attention to the robot workspace and stereotypical behavior are crucial to the rat RLBMI model—they couple the robot and target positions to the rat's neuronal modulations. This coupling is in accordance with the assumptions proposed in the state definition of the CA.

C. Microelectrode Array Implantation and Signal Acquisition

Rats that reach the operant conditioning inclusion criterion are chronically implanted bilaterally with two microelectrode arrays in layer V of the caudal forelimb area in the primary motor cortex (MI) [34], [35]. Neuronal signals are recorded from the caudal forelimb area of MI because this area has been shown to be predictive of limb motion in a rat model; additionally, similar modulations occur when operating a BMI without physical movements [36]. Each array is 8×2 electrodes with 250 μm row and 500 μm column spacing (Tucker Davis Technologies (TDT), Alachua, FL). The arrays are positioned stereotaxically and lowered independently with a hydraulic micropositioner to an approximate depth of 1.6 mm. Spatiotemporal characteristics of neuronal signal during insertion provide additional information about the array location relative to layer V. More details of the surgical technique are given in [37]. The rat is given up to two weeks to recover from surgery before resuming the experiment.

Electrophysiological recordings are performed with commercial neural recording hardware (TDT). A TDT system (one RX5 and two RP2 modules) operates synchronously at 24414.06 Hz to record neuronal potentials from both microelectrode arrays. The neuronal potentials are bandpass filtered (0.5–6 kHz). Next, online spike sorting [38] is performed to isolate single neurons in the vicinity of each electrode. Prior to the first closed-loop experiment, the experimenter reviews each sorted unit over multiple days to refine the spike sorting thresholds and templates. The number of sorted single units varied between rats: rat01 had 16 units, rat02 had 17 units (including one multiunit), and rat03 had 29 units. The isolation of these units was repeatable over sessions with high confidence from the recordings. Once the neurons were isolated, the TDT system records unit firing times and a firing rate estimate is obtained by summing firing within nonoverlapping 100 ms bins. Additionally, all behavioral signals (e.g., water rewards, LED activation) are recorded synchronously using the shared time clock.

D. Brain-Controlled Robot Reaching Task

Once the rats have been implanted with microelectrodes, they enter into *brain-control* mode to test the RLBMI architecture [see Fig. 3(b)]. In *brain control,* the trial initiation (nose poke) is the same; however, the robot movements are no longer automatic; instead they are generated every 100 ms by the CA based on a value function Q translated from the rat's neuronal modulations (states) and possible robot movements (actions). *After* each robot movement, the CA receives feedback about the reward earned (r_{t+1}) from the *prior* action (a_t). (The CA's use of rewards to update Q is addressed in the next section.) If the CA has selected a temporal action sequences to maneuver the robot proximal $(r_t \geq 1)$ to the target, then the trial is a success. In successful trials, the robot completes the motion to press the *target lever* and the rat earns a water reward. The trial time limit is extended to 4.3 s in *brain control* to allow the rat and agent to achieve robot control and make corrections based on visual feedback.

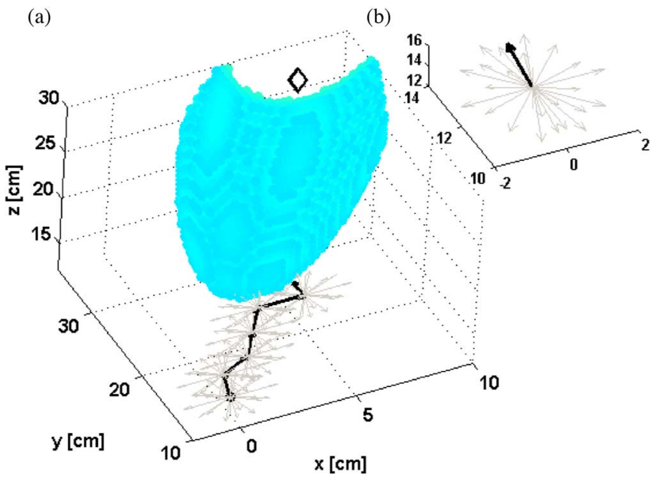


Fig. 4. (a) Example BMI agent actions and reward threshold locations (the target lever is marked by the diamond). The robot position at each time step is unknown to the BMI agent but visible to the rat (BMI user). Both the possible actions at each step (light gray) and the selected action (black) are shown. Once the robot position crosses the *dg* threshold (the gray Gaussian), the trial is considered a success [more details are given in Fig. 3(b)]. (b) Detail of the possible and selected actions from Fig. 4(a).

The action set available to the CA includes 26 movements defined in Cartesian[4] space: 6 unidirectional (e.g., up, forward, and right), 12 bidirectional (e.g., left–forward), 8 tridirectional (e.g., left–forward–down), and the "not move" option, yielding 27 possible actions. The robot is maneuvered in a 3-D workspace based on these actions (see Fig. 4); however, the diversity of actions creates an intractable amount of possible positions; thus, it is not a typical *grid world* [26].

The CA's rewards are assigned in the robot workspace based on the robot completing the task that the rat was trained to achieve. Both the CA and rat will be reinforced ($r_{t+1} = 1$ and water reward) after the robot is maneuvered proximal to the target. Similarly, both the CA and rat will be penalized ($r_t = -0.01$ and no water reward) after the robot has been moved but has not completed the task (this encourage minimization of task time). Because the experimenter controls the target locations in this rat model, it is also possible to partially reinforce the CA as the robot moves toward the target; this reward function is given

[4] To achieve robot actions in Cartesian space, inverse kinematics optimization (IKO) is required to calculate the necessary changes in each DOF. The agent uses neural networks to model the IKO such that it can be rapidly evaluated online. To maintain the same vector length, the uni-, bi-, and tridirectional action subsets have different component (*x*–*y*–*z*) lengths.

in (3). However, we do not partially reinforce the rat:

$$r_t = -0.01 + \exp\left(-r_s\left(d_{\text{thres}} - dg\right)\right) \tag{3}$$

$$dg = \exp\left(-\frac{1}{2}\left(\frac{d(x')^2}{0.001} + \frac{d(y')^2}{0.003} + \frac{d(z')^2}{0.0177}\right)\right). \tag{4}$$

The reward function in (3) includes the negative reinforcement (-0.01), two distance functions (dg and d_{thres}), and scaling factor r_s. Equation (4) describes the dg distance function and includes $d(n)$ that is the Euclidean distance (along the n-axis) between the target position (static) and robot endpoint at time t. Additionally, the axes in (4) are rotated such that the z$'$-axis originates at the target and ends at the robot initial position. The covariance terms in (4) are selected such that reward can be earned from multiple action sequences, but dg is maximal along a path directly to the target (e.g., in Fig. 4). We designed dg to maximize reward and encourage minimal control time. The target proximity threshold d_{thres} sets the necessary value of dg to complete a task ($r_t \geq 1$) and can be adjusted from close to the robot starting position to far away as a mechanism to shape complex behaviors. Finally, r_s controls the distribution of partial reinforcements that can be given to further develop the rat's control. This set of parameters for rewards and thresholds formalizes the goals of the task.

The complete brain control paradigm provides a mechanism to directly control task difficulty with d_{thres} in (2). Increasing task difficulty between sessions demonstrates the RLBMI's ability to adapt to changing environmental dynamics. In brain control, d_{thres} is initially set low to increase the probability of trial success; this keeps the rat engaged and facilitates the RLBMI coadaption to the early portion of the task. After a rat demonstrates greater than 60% accuracy (*brain control* inclusion criterion) for both targets in a session, task complexity was increased in the next session. We expect that the rat and agent will coadapt to achieve more difficult tasks, where other BMI would require retraining for new tasks.

As with rat operant conditioning, the rat and the CA must coadapt to learn the task over multiple days. The rat is not told explicitly that it is in brain control since all four levers are extended for each trial. The rats tended to remain stationary in the center of the cage directly in front of the water center, eyes facing the robot workspace. However, the rat continued to generate different neuronal modulations for each target. An illustration of the partial (due to space constraints) state signal for the two targets is given in Fig. 5. Essential to the success of this task is the coupling of the motivation and actions (neuronal modulations) of the rat with the CA's action selection (movements of the robot). While the rat is learning which neuronal modulations result in water rewards, the CA must adapt to more effectively respond to the rat's brain.

E. Value Function Estimation (VFE)

In this RLBMI architecture, the value function estimation (VFE) is a nontrivial task. The value function Q [see (2)] can be stored in a *lookup* table [26] if the number of states and actions are reasonably small. Although the RLBMI architecture contains only 27 actions, the number of possible states is intractable because they are composed of high-dimensional neural data. Therefore, it is not feasible to store Q in a *lookup* table for this application.

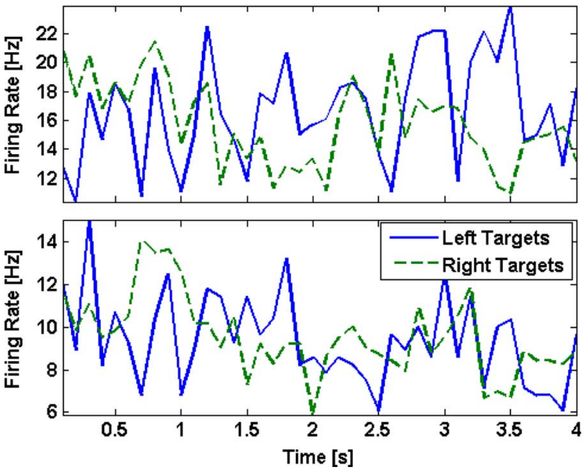


Fig. 5. Examples of the CA's state for two neurons from rat02. Nose poke is at 0 s and average trial time is at 3.7 and 4.6 s (left and right targets).

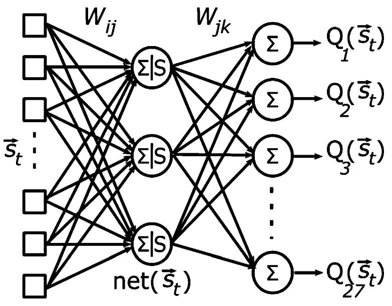


Fig. 6. VFE network. PEs also have bias inputs.

Theoretically, many function approximators can estimate Q (e.g., linear regressors, decision-tree methods [26], Gaussian process models [39]). Many of these networks require state–space segmentation, such as tiling, clustering, or hashing [26]. However, these techniques can scale poorly to high-dimensional spaces; the state spanned 48–77 dimensions in these experiments. Instead of preprocessing segmentation, a neural network is used to project the state to a space where segmentation is better performed [32].

Both single-layer perceptrons (SLPs) and multilayer perceptrons (MLPs) were investigated in [32] for this architecture but MLPs exhibited superior performance. The RLBMI uses a gamma delay line [40] ($K = 3$, $\mu = 0.3333$) to embed 600 ms of neuronal modulation history into the state. Then, an MLP (the VFE network) both segments the state and estimates Q as

$$Q_k(s_t) = \sum_j \tanh\left(\sum_i s_{i,t} w_{ij}\right) w_{jk} = \sum_j \text{net}_j\left(s_t\right) w_{jk}. \tag{5}$$

Each output processing element (PE) represents the value of the kth action given the state vector. The MLP architecture is shown in Fig. 6: there are three (set based on [32]) hyperbolic tangent hidden layer PEs and 27 linear output PEs.

The CA must adapt Q toward Q^* [see (2)] based on rewards it observes after taking actions. Temporal difference (TD) error is a known RL error metric for this adaptation [26] that learns from actual rewards and the network's own predictions. The TD error in (6) includes the actual reward r_{t+1}, the future rewards that the agent expects to earn from the next state $Q(s_{t+1}, a_{t+1})$, and the expected reward of the $s_t - a_t$ pair $Q(s_t, a_t)$. Additionally, there is a discount factor γ as in (1) to determine how far into the future rewards are considered. This metric allows CA to update Q *after* completing action a_t using the *next* available reward r_{t+1}.

Similar to the TD error, TD(λ) error uses actual rewards and self-predictions to adapt Q [26]. However, this metric includes a λ term to also consider actual rewards farther in the future. To understand the TD(λ) error, it is helpful to express it in (7) in terms of TD(0) errors, as shown in [26]. In (7), γ and λ are the same parameters defined in (1) and (6), respectively. An advantage of TD(λ) error in a BMI environment is that error can be partially computed as each r_{t+n} is observed. This allows the agent to partially update Q using currently available error (δ_{t+n-1}) information and refine Q as more rewards become known [26], [41].

The MLP is trained online using TD(λ) error via backpropagation; this training is an implementation of Watkin's Q(λ) learning [26]. The VFE network cost function is defined as squared TD(λ) error in (8):

$$\delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \tag{6}$$

$$\delta_t^{\lambda} = \delta_t + \sum_{n=1}^{T-1} (\gamma\lambda)^n \, \delta_{t+n} \tag{7}$$

$$J(t) = \frac{1}{2}\left(\delta_t^{\lambda}\right)^2. \tag{8}$$

An eligibility trace is a mechanism to gate value function learning based on the sequence of actions selected. Additionally, it provides "memory" such that reward can be distributed to prior state–action pairs that contributed to the current reward earning situation [26]. Eligibility traces facilitate partial updates by accounting for future terms in (7). The eligibility trace is given in (9) with the update in (10) where γ and λ are the same parameters defined in (7).

The eligibility trace for any unselected actions is zero because the observed rewards are not relevant for those actions. Additionally, anytime the agent takes an exploratory action, all prior eligibility traces are reset to zero. Action selection is determined by an ε-greedy policy [26] given by (11) where ε is the probability of selecting the action that maximizes Q given s_t. An eligibility trace is computed for each state (e.g., if prior states are $[s_1, s_2, s_3]$, then eligibility traces are maintained $[e(s_1), e(s_2), e(s_3)]$) and updated throughout the trial. The eligibility trace is substituted into the error gradient of (8) to yield (12). The VFE network is partially updated as δ_{t+n} becomes available using (12) with standard backpropagation equations [16] for the rest of the network. Full expansion of these update equations shows agreement with Sutton's original TD(λ) backpropagation formulation [42]:

$$e_t(s_t)_k = \begin{cases} 1, & a_t = k \\ 0, & \text{else} \end{cases} \tag{9}$$

$$e_{t+n}(s_t)_k = \begin{cases} (\gamma\lambda)^n e(s_t)_k, & \\ \qquad a_{t-n-1} = \arg\max_k Q_k(s_{t-n-1}) & \\ 0, \quad \text{else} & \end{cases} \tag{10}$$

$$a_t = \begin{cases} \arg\max_k \{Q_k(s_t)\}, & p(1-\varepsilon) \\ rand \neq \arg\max, & p(\varepsilon) \end{cases} \tag{11}$$

$$\frac{\partial J(t)}{\partial Q_k(s_t)} = -\sum_{n=0}^{T} e_{t+n}(s_t)_k \delta_{t+n}. \tag{12}$$

TABLE II
RLBMI AVERAGE PARAMETERS

RLBMI Parameter	Chance = 24.9%	Chance = 19.4%	Chance = 9.5%	Chance = 4.4%
α_{IL}	0.0016	0.0025	0.0010	0.0005
α_{OL}	0.006	0.0069	0.0023	0.0027
λ	0.8222	0.8333	0.8524	0.8468
γ	0.9	0.9	0.9	0.9
ε	0.01	0.01	0.01	0.01
Negative updates	32	33	19	13
r_s*	1000	1000	436	249

*Note: Most sessions r_s =1000, effectively making r_n binary; however, the functionality is available for future use and was used for rat03 in the last 2 difficulties (r_s = 65, 81.6). Other rats didn't use r_s so averages are skewed.

F. RLBMI Parameter Selection and VFE Training

In general, learning rates must be fast enough to estimate Q online but not destabilize the VFE network (tracking). Additionally, RL parameters must be appropriate for the task [26]. We selected the initial parameter set based on prior work [32] and adjusted the parameters heuristically to understand their effect on RLBMI performance. We continuously analyzed the weight tracks for all rats and ensured that the updates were smooth (not tracking a solution) within and between sessions. In Table II, we present the average system parameters we implemented for each difficulty. The implications of each parameter are addressed in Section IV.

Adapting a VFE network with TD(λ) error typically requires either online training with a sufficiently large dataset or off-line, batch training (repeatedly processing a smaller dataset) [26]. Online training started with random MLP weights and the rat began to control the robot immediately. The initial robot trajectories were jerky due to the random Q, but over multiple trials, the agent learned to reach at least one of the targets. Typically, there was target selection bias due to incomplete VFE adaptation (low α) or tracking (high α). However, off-line batch training between the first and second sessions resolved these issues. The VFE network was trained using the initial session's data with some crucial differences from [32]. The state data are no longer segmented based on the rat's physical behavior—instead it includes all neuronal modulations within a trial. Also, rewards are defined by (2) and the data were collected in brain control. All trials (successes and failures) were used for training. A training set was created from approximately 70% of the trials (30% reserved for a test set). From the training data, the normalization coefficients were recorded for each neuronal unit [43]; these coefficients remained static for all future sessions. Multiple training simulations were performed with each VFE network's initial weights generated by a different random seed and the networks were trained over 400–1000 epochs (depending on the rat). After training, the test dataset was presented to several VFE networks; the network with the best generalization was saved and used in the next session (no further off-line training was done).

In all sessions, the CA updates Q online using (12) based on reward observed *after* completing actions, as shown in Fig. 3(b). However, learning is constrained such that the number of unsuccessful updates was limited to 1.5–3 times the minimum number of updates in a successful trial. This prevented Q from degrading toward zero as the CA learned new control strategies for more complex tasks. If the rat exceeded the brain control inclusion criterion and/or the VFE network was stable, the session was considered a success and the final VFE weights were used as initial weights for the next session. All results are for continuous coadaptation over multiple sessions.

III. Results

A. RLBMI Task Completion Performance and Speed

The performance and usefulness of the RLBMI was evaluated only during brain-control tasks. *During brain control, all rats typically remained motionless near the reward center, faced the robot workspace, and relied on using neural activation to interact with the CA*. For goal-based BMI applications, the speed and accuracy of completing the task are two primary metrics that demonstrate the functionality of the interface. In this experimental paradigm, we quantify the percentage of trials in which the rat successfully navigated the 3-D workspace with the robotic arm to achieve a reward (PR) and compare with random walks of the robot. In addition to quantifying the successful trials, we measure the time that it takes to reach a target (TT). We expect that coordinated control will yield PR several times greater than chance level and use more direct paths; hence faster TT.

For each rat involved in the study, coadaptation of a single RLBMI model occured over multiple sessions (one session per day, 2.1±1.2 sessions per d_{thres}, and 141.6±41.3 trials per session). After each rat met the performance inclusion criterion (PR $= 60\%$), the reaching task complexity was increased (i.e., the number of successive actions necessary to earn reward) between sessions to shape the rat toward the most complex task. The PR and TT metrics were calculated in brain control for each d_{thres} and compared to *chance* performance[5] estimated from simulated RLBMI trials using a random Q. The *chance* PR provides a metric of task difficulty in all analysis.

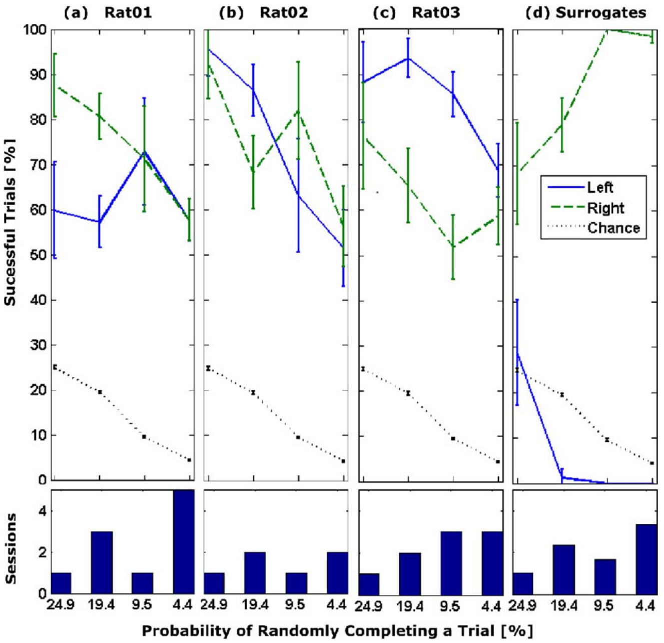


Fig. 7. PR versus chance over task difficulties (top) and the number of sessions performed at each difficulty (bottom) (a)–(c) for rat01, rat02, and rat03, respectively. (d) PR for the surrogate neural data. Error bars represent the 95% confidence interval in all plots.

The RLBMI accuracy is presented in Fig. 7(a)–(c), which shows each rat's left and right target PR averaged over trials for each difficulty. While coadapting with the CA, each rat achieved control that was significantly better (two-sample K–S test, $\alpha = 0.05$) than chance for all task complexities. RLBMI average (over difficulties and targets) PR was 68%, 74%, and 73% for rats 1, 2, and 3, respectively (average chance PR is 14.5%). Additionally, the individual PR curves indicate that the coadaptation is enabling the RLBMI to retain or improve performance in increasingly complex environments. Although classic psychometric curves [27] predict a steady performance *decrease* with increased difficulty, each rat exhibits at least one instance of *increased* PR with task difficulty [see Fig. 7(a)–(c), top]. This may reflect the role of coadaptation in the RLBMI.

We also present the 95% confidence intervals as error bars (also shown on chance curves but are difficult to see given the *y*-axis scale). The confidence intervals changed between the second step to the final step by -16%, $+26\%$, and -1% for the three rats as task difficulty increased. However, the number of trials in later sessions masks increases in standard deviation of 124%, 389%, and 364%. The PR variance with increasing task difficulty is partially due to lower PR sessions necessary for the rat and CA to coadapt to the new environment. At the second difficulty level, rats were within 9% of the inclusion criteria for all sessions. However, all rats had at least one session 20–35% below the inclusion criteria as the rat and CA learned to solve the final difficulty level.

To be thorough, we repeated the surrogate neural data tests from our prior work [32] to determine if the CA could learn a

[5] *Chance* PR is calculated using five sets of 10 000 simulated *brain control* trials using random action selection. The *PR* from each set of trials is then used to calculate the average and standard deviation. *Chance* TT is calculated from the concatenation of the five sets of random trials. The data used to calculate chance PR and TT are also used in two-sample Kolmogorov–Smirnov (K–S) (95% significance) tests for statistical comparisons.

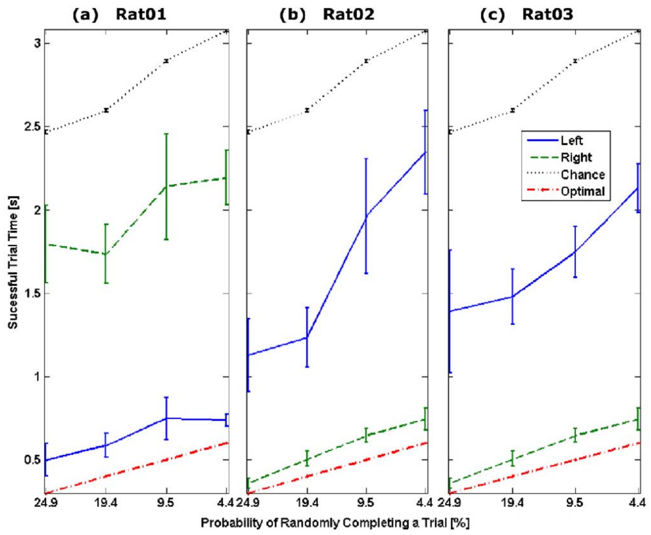


Fig. 8. (a)–(c) TT over task difficulties for rat01, rat02, and rat03, respectively. Error bars represent the 95% confidence interval in all plots.

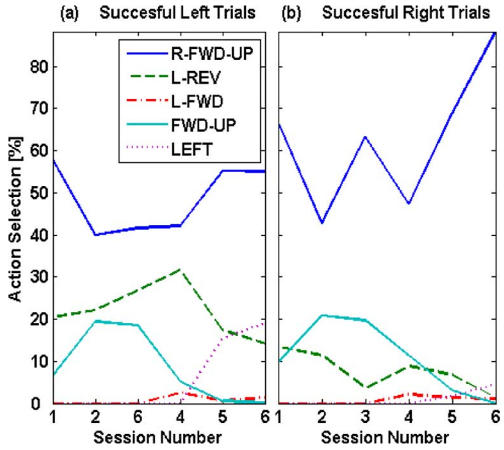


Fig. 9. RLBMI action selection for rat02. (a) Left trials. (b) Right trials.

solution regardless of the state. Rat neuronal firing rates were randomized temporally and spatially to create a surrogate *state*. A surrogate network was created using the average RLBMI parameters from all rats (see Table II). The network is trained for the same average number of trials and sessions at each difficulty. In Fig. 7(d), both the rat and surrogate PR are shown with error bars for the 95% confidence intervals. The surrogate network learned to guess one target (right side) for all trials with an average PR of 47%. This suggests that without causal neuromodulation (states) from the rat, only one solution was being memorized by the network and not generalizing to the overall task.

Fig. 8(a)–(c) shows each rat's left and right TT averaged over trials of the same difficulty versus task difficulty (the number of sessions is identical to Fig. 5). The *chance* TT is also plotted for reference (the surrogate TT was not be used because that network was unable to complete both tasks). All three rats achieved significantly faster (two-sample K–S test, $\alpha = 0.05$) trial completion than chance for all task difficulties. RLBMI average (over difficulties and targets) TT was 1.3, 1.1, and 1.1 s for rats 1, 2, and 3, respectively (average chance TT is 2.7 s). The optimal TT was computed by the time needed to move directly to each target along the shortest path. Each increase in task difficulty increased the theoretical minimum TT because the targets are farther away. Instances where the TT curve has a less negative slope than the optimal TT suggest that coadaptation of the RLBMI can improve prosthetic control.

The actions used by the RLBMI also affect both PR and TT for each target. The rats exhibited different left and right trial PR despite the trial difficulty being the same by design (all actions are the same vector length and targets are equidistant from the initial robot position). However, each CA coadapted over time with the user to only use a subset of the possible actions and users may have different strategies to reach each target. This has the net effect of unbalancing the task difficulty for left and right targets. The set of actions most commonly used by the RLBMI also affect TT for each target. For example, rat02 and rat03's left TT were longer than the right TT indicating they used less direct paths to the left target.

B. RLBMI Action Selection

The distribution of actions selected for each session illustrates the RLBMI action selection strategy. The agent seeks to maximize R_t and could accomplish this by minimizing TT using only two tridirectional *direct* actions to move the robot directly to the target. Fig. 9(a) shows the distribution of the most used actions in rat02's successful left trials (representative of all rats). The RLBMI selected robot actions directly toward (R–FWD–UP) the right target 50% of the time. The RLBMI selected corrective actions toward the left (correct) target for 40% of the time. However, Fig. 9(b) shows that a single, direct action is selected in 90% of successful right trials. Additionally, the RLBMI initially used a larger subset of five actions but over time, the subset is reduced to three. This shows that training may still be improved—the rat strategy may be suboptimal due to experimental conditions (visual feedback).

All three RLBMI systems adapted to an action subset that facilitates visual feedback to correct robot trajectories. If the action set only included two *direct* actions, the rat could minimize TT moving directly to both targets. In the event the RLBMI initially used the incorrect *direct* action, the rat would receive visual feedback of a control error; however, even if the rat modulated neuronal firing to select the other *direct* action, it would be unable to successfully maneuver the robot to the correct target. Instead the robot would move toward the lever but reach a workspace boundary (wall) condition and stop short of reaching the correct target—failing to earn a reward. Changing the safety constraints of the robot workspace may allow both optimal action sets and the use of visual feedback.

The differences in action selection illustrated in Fig. 9 explain the TT difference in Fig. 8(b): right trials are almost three times faster because the robot moves directly to the target in successful right trials. Also, the change in the left TT in Fig. 6(b) after the third session can be explained by the changing strategy observed in Fig. 8(a). After the third session, the RLBMI becomes less likely to use a combination of actions that maneuver the robot toward the left target; instead, the combination of actions includes actions away from the target and corrective actions. This creates less direct paths and it follows that TT increases.

IV. Discussion

A novel BMI architecture based on RL, coadaptation, and shaping was developed and demonstrated in a series of rat behavioral experiments. In this RLBMI, a CA observes a user interacting with an environment and develops strategies that maximize the combined reward acquisition. Rewards are a powerful learning mechanism that exist simultaneously for the BMI user and CA; hence, they coordinate and facilitate learning for both "intelligent systems" in the RLBMI architecture. Coadaptation allows *users* to modulate their neural activity and the CA to adapt the functional BMI mapping—synergistically improving prosthetic control. Finally, the concept of using shaping to achieve brain control of a prosthetic in this RLBMI framework enables the development of complex tasks while possibly reducing the "learning curve" for patients using a BMI.

The RLBMI exploited spatiotemporal structure in the firing of 16–29 MI neurons; this formed the *state* that reflected the rat's goals and perception of the workspace. The CA learned to select sequences of prosthetic *actions* to complete the tasks (earn *reward*), which suggests that sequences of *states* were distinct for different tasks. This agrees with our prior work showing that the RLBMI does not function with surrogate neural data [32]. The *actions* were experimentally designed to provide *maximal* control DOF; however, the RLBMI adapted to find a *necessary* subset of control DOF to complete tasks. The composition of the limited action set suggests that the rats did not fully use all the actions that were available in the *brain-control* task. This may indicate that the rats ignored inefficient actions, selected an action set to enable visual feedback, or that that there was not sufficient neuromodulation to trigger all actions. However, this question will be addressed in a future article using additional neural ensemble analysis. Based on the rat training and composition of the action sets, we hypothesize that the rats used visual feedback to achieve control.

The RLBMI learning parameters provide flexibility to achieve prosthetic control despite different users and environmental conditions. Throughout the course of these experiments, we discovered an effective combination of parameters to improve system performance and increase VFE stability by observing performance trends (see Table II). The MLP learning rates were very effective parameters for controlling adaptation of the CA. The input layer learning rate a_{IL} affected changes in the neuronal data projection and state segmentation. It was important to preserve the state; increasing a_{IL} could entirely destabilized the RLBMI. However, a_{IL} did allow the state to adapt to changing neuronal signal (e.g., neuron loss) over multiple sessions. The output layer learning rate a_{OL} had more effect on the actual value Q_k of each possible action. It was most effective for the output layer to learn at least five times faster than the input layer and to reduce both learning rates by 20% between each session. This suggests that the RLBMI system is more capable of adjusting values for existing state–action pairs than rapidly resegmenting the state–space and evaluating new state–action pairs; this agrees with intuition. Limiting the number of weight updates in unsuccessful trials was also an effective mechanism for preserving the VFE network while the rats adjusted to a new control task. It allowed the CA to learn rapidly after successful trials but still preserve some prior knowledge after unsuccessful trials.

The RL specific parameters had more influence on CA learning within a session. The λ parameter [see (5) and (7)] controls the history in the weight update; it was initially set based on the minimum trial length and adjusted based on performance. The discount factor γ controls the reward horizon but was kept constant throughout sessions to preserve prior VFE mappings as task difficulty increased. Exploration of ε was useful for a naïve CA and rat to earn reward. However, as shown by (10), ε slows value function adaptation; hence, ε is kept under 1% in developed VFE networks. The r_s term was helpful in one rat; however, it is a sensitive parameter that needs future investigation.

The RLBMI is an implementation of Q(λ) learning that allows online (incremental within each trial) or batch (after each trial) value function updates based on the TD(λ) error. As long as the errors are only applied to *prior* states and do not bias *current* action selections, real-time implementation of the RLBMI algorithm can be developed with *either* online or batch updates because the net online update is approximately the same as batch update [26]. We respect this design requirement in the research presented here. For control tasks where incremental behaviors are important, online updates are advantageous as discussed in [26]. Additionally, the computation complexity for online updates is on the order of an MLP, so it was possible to meet a real-time BMI deadline that keeps RLBMI on par with other decoding algorithms. However, RLBMI has the distinct benefit of a coadaptive learning rule based on rewards.

Continuous coadaptation and reward learning are two unique features of the RLBMI architecture. Conventional BMI retraining with a desired response requires the patient to physically or mentally (in the case of the paralyzed) generate a training set that imposes a delay before the interface can be used. In addition, retraining may create learning confounds because it generates a different control mapping (network weights) for the patient each day. RLBMI instead used *continuous* coadaption over six to ten days with all training (and results) using a purely *brain-controlled* prosthetic. Continuous coadaptation incorporates prior knowledge that the CA has gained; this allows the patient to learn a control strategy over multiple days (network weights are preserved; hence, prior knowledge is preserved). Therefore, RL enables a training philosophy unlike the conventional BMIs because: 1) it does not need an explicit desired signal; 2) it improves performance with usage and may allow for more difficult tasks due to the feedback between the user and the CA; and 3) it may be possible to switch between different task sets by changing the reward locations in the workspace, although this aspect was not explored here.

The RLBMI currently uses a model-free RL technique because environmental dynamics are unknown. The agent can only learn from experience and cannot predict future states. To overcome the known limitation of relatively (compared to SL) slow learning speed, the available data were reused in multiple-epoch, off-line VFE training. We are exploring new and more effective methods for training the RLBMI using multiple models [44] for rapidly learning VFE as the patient acquires prosthetic control in the initial session. Additionally, future RLBMI implementation may benefit from model-based RL that includes an environmental model to estimate future states and rewards [26]. This modification would allow the CA to learn from both experience and model prediction of possible environmental interactions, thus facilitating faster learning. In this paper, the rewards were programmed by the BMI designer, but in the future, they should also be translated from the user's brain activity. When this is achieved, the brain control of prosthetics could be made more general with the production of new goals and reduction of old goals. We believe that this paper shows feasibility of CA and user coadaptation for a set of tasks, without requiring explicit desired responses for each step of the trajectory.

Acknowledgment

The authors would like to thank M. Zhao for providing an initial C++ robot control system and TDT polling architecture, P. Rattanatamrong for assistance in translating portions of the RLBMI algorithm from Matlab to VC++, J. Mitzelfelt for assistance with rat training and surgeries, and B. J. Fregly for suggestions on IKO techniques and surrogate models.

References

[1] E. E. Fetz and D. V. Finocchio, "Correlations between activity of motor cortex cells and arm muscles during operantly conditioned response patterns," *Exp. Brain Res.*, vol. 23, pp. 217–240, 1975.

[2] E. M. Schmidt, "Single neuron recording from motor cortex as a possible source of signals for control of external devices," *Ann. Biomed. Eng.*, vol. 8, pp. 339–349, 1980.

[3] R. Shadmehr and S. P. Wise, *The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning*. Cambridge, MA: MIT Press, 2005.

[4] E. V. Evarts, "Representation of movements and muscles by pyramidal tract neurons of the precentral motor cortex," in *Neurophysiological Basis of Normal and Abnormal Motor Activities*, M. D. Yahr and D. P. Purpura, Eds. New York: Raven, 1967, pp. 215–253.

[5] A. K. Engel and W. Singer, "Temporal binding and the neural correlates of sensory awareness," *Trends Cogn. Sci.*, vol. 5, pp. 16–25, 2001.

[6] A. A. Ghazanfar, C. R. Stambaugh, and M. A. Nicolelis, "Encoding of tactile stimulus location by somatosensory thalamocortical ensembles," *J. Neurosci.*, vol. 20, pp. 3761–3775, 2000.

[7] C. Koch and J. L. Davis, *Large-scale Neuronal Theories of the Brain*. Cambridge, MA: MIT Press, 1995.

[8] J. K. Chapin and C.-S. Lin, "The somatosensory cortex of the rat," in *The Cerebral Cortex of the Rat*, B. Kolb and C. R. Tees, Eds. Cambridge, MA: MIT Press, 1990, pp. 341–380.

[9] J. C. Sanchez and J. C. Principe, *Brain Machine Interface Engineering*. New York: Morgan and Claypool, 2007.

[10] J. K. Chapin and K. A. Moxon, "Neural prostheses for restoration of sensory and motor function," in *Methods and New Frontiers in Neuroscience*. Boca Raton, FL: CRC Press, 2001.

[11] M. Nicolelis, "Brain–machine interfaces to restore motor function and probe neural circuits," *Nat. Rev. Neurosci.*, vol. 4, pp. 417–422, 2003.

[12] J. P. Donoghue, "From mind to movement: Developing neuro-technologies to restore lost function," presented at the Neural Information Processing Systems (NIPS), Whistler, BC, Canada, 2004.

[13] S. H. Scott, "Neuroscience: Converting thoughts into action," *Nature*, vol. 442, pp. 141–142, 2006.

[14] L. R. Hochberg *et al.*, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, pp. 164–171, 2006.

[15] F. Rieke, D. Warland, R. de Ruyter vanStevenick, and W. Bialek, *Spikes: Exploring the Neural Code*. Cambridge, MA: MIT Press, 1999.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York/Toronto, ON, Canada: Macmillan/Maxwell Macmillan, 1994.

[17] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.

[18] G. Buzsáki, *Rhythms of the Brain*. New York: Oxford Univ. Press, 2006.

[19] S. P. Kim *et al.*, "A comparison of optimal MIMO linear and nonlinear models for brain–machine interfaces," *J. Neural Eng.*, vol. 3, pp. 145–161, 2006.

[20] E. N. Brown, R. E. Kass, and P. P. Mitra, "Multiple neural spike train data analysis: State-of-the-art and future challenges," *Nat. Neurosci.*, vol. 7, pp. 456–461, 2004.

[21] A. B. Schwartz, D. M. Taylor, and S. I. H. Tillery, "Extraction algorithms for cortical control of arm prosthetics," *Curr. Opin. Neurobiol.*, vol. 11, pp. 701–708, 2001.

[22] M. D. Serruya *et al.*, "Brain–machine interface: Instant neural control of a movement signal," *Nature*, vol. 416, pp. 141–142, 2002.

[23] J. Wessberg *et al.*, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361–365, 2000.

[24] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Information conveyed through brain-control: Cursor versus robot," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 195–199, Jun. 2003.

[25] S. Tillery, D. Taylor, and A. Schwartz, "Training in cortical control of neuroprosthetic devices improves signal extraction from small neuronal ensembles," *Rev. Neurosci.*, vol. 14, pp. 107–119, 2003.

[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[27] G. H. Bower, *Theories of Learning*, 5th ed. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[28] F. Worgotter and B. Porr, "Temporal sequence learning, prediction, and control: A review of different models and their relation to biological mechanisms," *Neural Comput.*, vol. 17, pp. 245–319, 2005.

[29] R. Miikkulainen *et al.*, "Computational intelligence in games," in *Computational Intelligence: Principles and Practice*, G. Y. Yen and D. B. Fogel, Eds. Piscataway, NJ: IEEE Computational Intelligence Society, 2006, pp. 155–191.

[30] C. Touzet and J. F. Santos, "Q-Learning and robotics," presented at the IJCNN—Eur. Simul. Symp., Marseille, France, 2001.

[31] J. Nie and S. Haykin, "A dynamic channel assignment policy through Q-learning," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1443–1455, Nov. 1999.

[32] J. DiGiovanna, B. Mahmoudi, J. Mitzelfelt, J. C. Sanchez, and J. C. Principe, "Brain–machine interface control via reinforcement learning," in *Proc. IEEE EMBS Conf. Neural Eng.*, May, 2007, pp. 530–533.

[33] I. Q. Whishaw, *The Behavior of the Laboratory Rat*. New York: Oxford Univ. Press, 2005.

[34] J. A. Kleim, S. Barbay, and R. J. Nudo, "Functional reorganization of the rat motor cortex following motor skill learning," *J. Neurophysiol.*, vol. 80, pp. 3321–3325, 1998.

[35] J. Donoghue and S. Wise, "The motor cortex of the rat: Cytoarchitecture and microstimulation mapping," *J. Comp. Neurol.*, vol. 212, pp. 76–88, 1982.

[36] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nat. Neurosci.*, vol. 2, pp. 664–670, 1999.

[37] J. C. Sanchez, N. Alba, T. Nishida, C. Batich, and P. R. Carney, "Structural modifications in chronic microwire electrodes for cortical neuroprosthetics: A case study," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 217–221, Jun. 2006.

[38] M. S. Lewicki, "A review of methods for spike sorting: The detection and classification of neural action potentials," *Netw.: Comput. Neural Syst.*, vol. 9, pp. R53–R78, 1998.

[39] M. P. Deisenroth, J. Peters, and C. E. Rasmussen, "Approximate dynamic programming with Gaussian processes," presented at the Amer. Control Conf., Seattle, WA, 2008.

[40] 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 Process.*, vol. 41, no. 2, pp. 649–656, Feb. 1993.

[41] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learning*, vol. 8, pp. 229–256, 1992.

[42] R. S. Sutton, "Implementation details of the TD(l) procedure for the case of vector predictions and backpropagation," GTE Lab., Waltham, MA, Tech. Rep. TN87-509.1, 1989.

[43] W. C. Lefebvre *et al.*, *NeuroSolutions*, 4.20 ed. Gainesville, FL: NeuroDimension, Inc., 1994.

[44] J. DiGiovanna *et al.*, "Towards real-time distributed signal modeling for brain machine interfaces," presented at the Int. Conf. Comput. Sci., Beijing, China, 2007.

Jack DiGiovanna (S'04) received the B.S. degree in electrical engineering (minor in bioengineering) in 2002 from Pennsylvania State University, University Park, and the M.E. and Ph.D. degrees in biomedical engineering from the University of Florida, Gainesville, in 2007 and 2008, respectively.

He was with the Sensory Motor Control Group, Cambridge University, and the Advanced Robotics Technologies and Systems Laboratory, Scuola Superiore Sant'Anna. In 2004, he joined the Computational NeuroEngineering Laboratory (CNEL), University of Florida, where in 2006, he joined the Neuroprosthetics Research Group (NRG) Laboratory. In 2007, he received a National Science Foundation (NSF) International Research in Engineering and Education grant. He holds one patent in neuroprosthetic design and is the author or coauthor of over ten peer-reviewed papers. His current research interests include reinforcement-learning-based brain–machine interface (BMI) and motor control systems.

Mr. DiGiovanna was a Founding Officer in the Gainesville IEEE Engineering in Medicine and Biology Society (EMBS) Chapter.

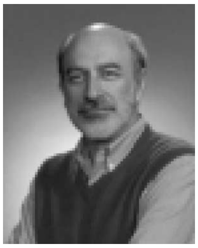
Babak Mahmoudi (S'06) received the B.S. degree in electrical engineering from the University of Tehran, Tehran, Iran, and the M.S. degree in biomedical engineering from Iran University of Science and Technology, Tehran, in 1998 and 2003, respectively. He is currently working toward the Ph.D. degree in biomedical engineering at the University of Florida, Gainesville.

In 2004, he was invited to RIKEN Brain Science Institute, Japan, as a Visiting Researcher. In 2006, he joined the Neuroprosthetics Research Group, University of Florida, where he worked on the Dynamically Data-Driven Brain–Machine Interface Project. His research has been focused on neural interfaces and brain–machine interfaces for the past six years. He has industrial experience on several signal processing and telecommunication projects.

Jose Fortes (S'80–M'83–SM'92–F'99) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1984.

From 1984 to 2001, he was on the faculty of the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. In 2001, he joined both the Department of Electrical and Computer Engineering and the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, as a Professor and BellSouth Eminent Scholar, and founded and currently directs the Advanced Computing and Information Systems (ACIS) Laboratory and the National Science Foundation (NSF) Industry—University Cooperative Center on Autonomic Computing. His current research interests include distributed computing and autonomic computing.

Prof. Fortes was a Distinguished Visitor of the IEEE Computer Society from 1991 to 1995.



Jose C. Principe (M'83–SM'90–F'00) received the Ph.D. degree in electrical engineering from the University of Florida, Gainesville, in 1979.

He is a Distinguished Professor of electrical and biomedical engineering at the University of Florida, Gainesville, where he teaches advanced signal processing and artificial neural networks (ANNs) modeling. He is also the BellSouth Professor and the Founder and Director of the University of Florida Computational NeuroEngineering Laboratory (CNEL). He is involved in biomedical signal processing, in particular the electroencephalogram (EEG) and the modeling and applications of adaptive systems. He is the author or coauthor of more than 400 refereed publications.

Prof. Principe is a former Editor-in-Chief of the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, and is currently an Editor-in-Chief of the IEEE REVIEWS IN BIOMEDICAL ENGINEERING. He was the President of the International Neural Network Society and a formal Secretary of the Technical Committee on Neural Networks of the IEEE Signal Processing Society. He is a Fellow of the American Institute of Medical and Biological Engineering (AIMBE). He was the recipient of the IEEE Engineering in Medicine and Biology Society Career Service Award. He was also a member of the Scientific Board of the Food and Drug Administration.

Justin C. Sanchez (M'02) received the B.S. degree in engineering science (with highest honors) with a minor in biomechanics in 2000 and the M.E. and Ph.D. degrees in biomedical engineering from the University of Florida, Gainesville, in 2004.

He is an Assistant Professor of pediatrics, neuroscience, and biomedical engineering in the College of Medicine, College of Engineering, and McKnight Brain Institute, University of Florida, where he is also a founding member of the Neuroprosthetics Research Group (NRG). He is involved in the development of state-of-the-art novel medical treatments by operating at the interface between basic neural engineering research and clinical care. His Neural Engineering Electrophysiology Laboratory is currently developing direct neural interfaces for use in the research and clinical settings. He has authored or coauthored over 35 peer-reviewed papers and holds three patents in neuroprosthetic design. His current research interests include neural engineering and neural assistive technologies, which include the analysis of neural ensemble recordings, adaptive signal processing, brain–machine interfaces, motor system electrophysiology, treatment of movement disorders, and the neurophysiology of epilepsy.

Dr. Sanchez received two prestigious awards for his work including the Excellence in Neuroengineering, and more recently, the American Epilepsy Society Young Investigator Award in 2005. In 2006, he founded the Gainesville Engineering in Medicine and Biology/Communications Joint Societies Chapter and is currently the Chapter Chair.