31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

An Actor-Critic Architecture and Simulator for Goal-directed Brain-Machine Interfaces

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

Abstract— **The Perception-Action Cycle (PAC) is a central component of goal-directed behavior because it links internal percepts with external outcomes in the environment. Using inspiration from the PAC, we are developing a Brain-Machine Interface control architecture that utilizes both motor commands and goal information directly from the brain to navigate to novel targets in an environment. An Actor-Critic algorithm was selected for decoding the neural motor commands because it is a PAC-based computational framework where the perception component is implemented in the critic structure and the actor is responsible for taking actions. We develop in this work a biologically realistic simulator to analyze the performance of the decoder in terms of convergence and target acquisition. Experience from the simulator will guide parameter selection and assist in understanding the architecture before animal experiments. By varying the signal to noise ratio of the neural input and error signal, we were able to demonstrate how the learning rate and initial conditions affect a motor control target selection task. In this framework, the naïve decoder was able to reach targets in the presence of noise in the error signal and neural motor command with 98% accuracy.**

I. Introduction

Brain-Machine Interfaces (BMI) have emerged as a new technology to enable patients suffering from severe motor impairment to interact with their environment. Integration of sensorimotor information in a cyclic manner is in the core of every interactive behavior of biological organisms. In patients with motor deficiency, the cyclic aspect of sensorimotor integration, called Perception-Action Cycle (PAC) [1], is disrupted and to restore it the BMI should serve as a bidirectional channel between the brain and environment. Through this channel, the brain sends motor commands to external devices and perceives the outcome in the form of sensory feedback.

The bulk of BMI research during the last decade has been focused on restoring the reaching and grasping functionality of the upper extremities [2]. Two main components of a reaching movement are trajectory and target location. Focusing on each of these two aspects of a reaching movement, BMI designers have either developed interfaces that build trajectories by direct translation of cortical neural activity to movement kinematics (*trajectory-based* BMIs) [3-5], or predicted the reaching target by extracting high-level goal information from the brain (*goal-based* BMIs) [6]. From the PAC stand point, both the trajectory-based and goal-based paradigms for BMI design are concerned with action restoration by translating neural motor commands for a reaching task. However, during goal-directed behavior, goal perception provides a valuation basis for action selection and it is instrumental for restoration of full PAC. In other words, in a bidirectional BMI, the interface establishes a dialogue with the brain rather than translating neural commands [7]. We have developed a framework in which goal information from the user is employed in action selection.

Reinforcement Learning (RL) as an interactive machine learning paradigm provides the computational basis of the PAC-based BMI. Actor-Critic (AC) is an implementation of RL which has separate structures for perception (critic) and action (actor) [8]. Given a specific state, the actor decides what action to take and the critic evaluates the outcome of the action in terms of future reward (goal). The link between action and perception in the AC architecture is an evaluative feedback, called Temporal Difference (TD) error, that the critic provides to actor. There is evidence that neurons in the Striatum represent reward expectation in the form of TD error [9, 10]. These neurons modulate their firing rate depending upon the probability of earning reward [11]. In other words, these neurons may provide a continuous measure of goal perception in the form of TD error ,which an actor (BMI decoder) could use for action selection and constructing trajectories for goal-directed reaching tasks. Throughout this paper the error refers to reinforcement that critic provides to actor in the form of evaluative feedback.

In this paper, we have designed an AC control architecture for BMI that enables the user reach any point in the continuum of its workspace through the PAC. Complexity of real neural data impedes the investigation of behavior of this control system and optimal parameter setting during in vivo experiments therefore we have developed a simulation platform to characterize the behavior of the BMI under neurobiological constraints. Using this simulator, we have studied the effect of noise in the input and TD error on the performance of the BMI during reaching multiple targets.

This work was supported in part by the U.S. National Science Foundation under Grant #CNS-0540304.

B. Mahmoudi is with the Department of Biomedical Engineering, University of Florida, 130 BME Building, Gainesville, FL 32611 USA (phone: 352-846-2171; fax: 352-392-9791; e-mail: babakm@ufl.edu).

J. C. Principe is with the Departments of Electrical Engineering and Biomedical Engineering, P.O. Box 116130 NEB 486, Bldg #33, University of Florida, Gainesville, FL 32611 USA (e-mail: principe@cnel.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)

II. METHODS

A. BMI Architecture

In this section, we formulate the control architecture of the BMI based on AC implementation of reinforcement learning. AC is a class of TD learning algorithms that intrinsically work based on PAC. By establishing a probabilistic mapping between states and actions, the actor contains an action selection policy and the critic is a value function that maps states to expected future rewards. For BMI applications, the actor plays the rule of action decoder and the critic provides an evaluative feedback in terms of the *goal* to the actor in the form of TD error. It has been shown in the literature that as animals move toward their expected goal the firing rate of neurons in the ventral striatum (the Nucleus Accumbens (NAc)), will modulate depending on if they are moving towards or away from the goal [12]; therefore we use the NAc response for evaluating the actor. In contrast to the conventional AC architecture where the critic should learn a value function for mapping states to expected cumulative future reward [13], in our BMI, the critic is biologically embedded in the user's brain and evaluates actions of the adaptive agent based on the user's reward expectation. If the agent action is favorable to the user's goal, reward expectation increases and that action would be reinforced. Otherwise, the reward expectation decreases and the action should be penalized.

Fig 1A shows the general architecture of our BMI. The actor corresponds to an adaptive agent that implements an action selection policy by mapping brain's motor states to actions of a robotic arm in a probabilistic manner. In this architecture, the role of the agent is to form *trajectories* by associating motor commands in the primary motor cortex (MI) at each time step (S_t) to robot actions in such a way that the probability of earning reward from the user's perspective is maximized. Figure 1B shows the architecture of the adaptive agent which is composed of a Multilayer Perceptron (MLP) neural network with gamma memory structure [14] at the input. Each Processing Element (PE) at the output of the network corresponds to a discrete action that actor can take.

In principle, training the actor at time step t requires that the user generates motor command S_t and expects a reward from execution of that command (V_t). The actor selects an action that is associated with the PE with highest value. If the robot moves towards the goal, reward expectation of the user (V_{t+1}) increases; otherwise it decreases. (1) translates the reward expectation into TD error for adaptation of the actor.

$$\varepsilon_t = r_t + \gamma\, V(s_{t+1}) - V(s_t),\ 0 \le \gamma < 1 \qquad (1)$$

Here r_t is the actual reward earned at time t and γ is a discount factor that affects the contribution of future expectation. The error is back propagated into the network and is used to update the parameters of the selected action in the output layer and all of the parameters in the hidden layer.

B. Experiment Setup

Since it is difficult to understand the intricacies of this BMI architecture during in vivo experiments, we have developed a simulation environment to study the effect of different conditions e.g. changing the tuning depth of neurons, on the performance of BMI. For BMI control, we seek to navigate a 2-D workspace to acquire one of 4 targets (Fig 1C). The environment consists of a 2x2 grid world with 0.1 spacing between each node. The start point for the agent was at the center of the grid. Three experiments were defined by placing one target at the right-up corner, two targets at right-up and left-up corners and four targets at each corner of the workspace.

An ensemble of 12 cortical neurons was generated based on the model in [15]. The firing rate of the neurons was computed over 100ms bins. The ensemble was composed of four subsets where neurons in each subset were tuned to a principal direction (up, down, right, and left) in the 2-D workspace. Each output PE of the agent corresponded to one of these directions. At each time step, the user's motor command was encoded into MI neural activity by exciting the corresponding subsets of neurons. For example, if the user decided to navigate the robot in the up-right direction, those neurons in the ensemble which were tuned to the right direction and up direction were stimulated. At each time step, given a particular neural state, the actor computed the action values and picked the one with highest value for navigating the robot. Based on the robot movement with respect to the target, movement and target vectors were computed at each time step (Fig 2C).

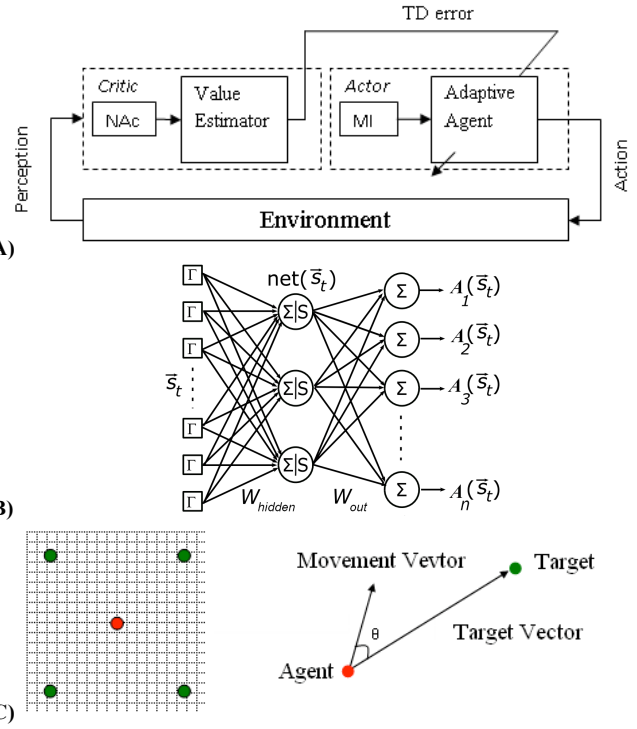


Fig. 1. A) Structure of the BMI controller, B) Architecture of the adaptive agent, C) Error (reinforcement) is defined based on the projection of the movement vector on the target vector in 2D grid world.

Displacement of the robot end-effector with respect to the target modulated the firing rate of the NAc neurons. The value estimator in Fig 1A is responsible for estimating TD error from neural activity in NAc. Estimation of the error is beyond the scope of this paper; therefore, here we used the error signal from the output of the value estimator. Considering the target vector as the desired direction, a scalar error was defined by computing the cosine of the angle between the movement and target vectors. If the movement direction was towards the target, a positive error was generated otherwise the error was negative. This error signal resembled the TD error in AC algorithm in the sense that if action was desirable the agent received a positive reinforcement otherwise the reinforcement was negative. In these simulations, we decreased the learning rate of the network after each successful trial. In other words, once the agent converged to an optimal policy it no longer needed the evaluative feedback and could reach the targets using MI.

Each experiment was composed of 100 trials. In each trial, if the agent could not reach the target in 50 steps the trial was considered unsuccessful. The agent was not confined to the borders of workspace. In the second and third experiments one target was selected randomly in each trial. For each experiment we ran 100 Monte Carlo (MC) simulations and the average performance in the top 10% MCs was computed as the performance of the agent.

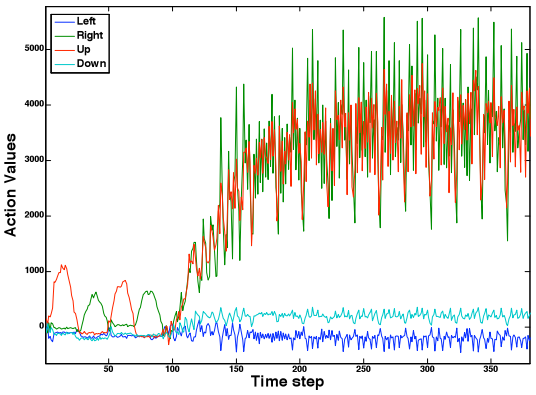
III. Results

We present the simulation results of the three different experiments in terms of action values, convergence, and the effect of noise in the input and error signal on the overall performance of the BMI. We used action values over time to quantify the speed of convergence and if the correct solution was obtained. Figures 2A-C show the values of output PEs during each experiment. Figure 2A plots the action values for the first experiment, we can see actions *right* and *up* have been selected by the agent. Based on the location of the target these two actions are required for completing the task. Since the agent can not pick a direct movement to the target, it learned how to use a sequence of these two actions in order to accomplish the task. Notice also that convergence time can be determined by the time needed to reach the steady state values which is at time step 100 in Fig 2A. During the second experiment where a new target was introduced, the agent incorporated a new action to reach the target. In Fig 2B we can see that in some instances the agent has picked action *left* in order to reach the left target. By increasing the number of targets to four, Fig 2C shows the agent has incorporated all the four actions in order to reach all the targets. In Fig. 2 we can see that by increasing the number of targets, convergence time also has increased.

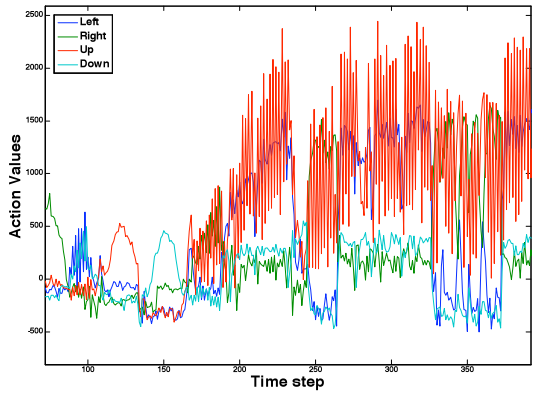
An important characteristic of any BMI decoder is its performance in the presence of noise. To investigate the effect of noise, we tested the system under three different conditions. First we reduced the tuning depth of MI neurons from 1 to 0.2 where the tuning depth was computed by (2).

$$Tuning\ Depth = 0.5 \times \left(\frac{Mean\ Firing\ Rate\big|_{Action}}{Mean\ Firing\ Rate\big|_{Baseline}} - 1\right) \qquad (2)$$

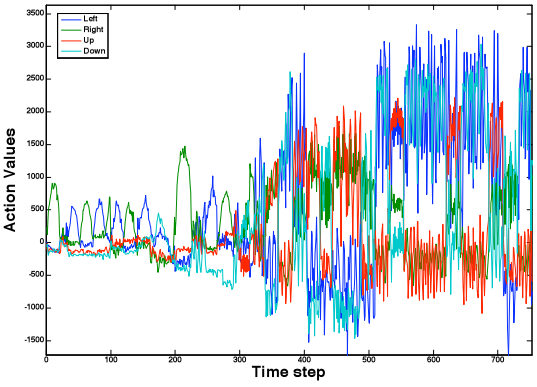
In the next step, Gaussian noise was added to the TD error to generate a noisy error signal with 5dB signal to noise ratio. Finally we put the agent under a noisy-input, noisy error condition. Table I summarizes the performance of the agent under these three conditions for different target configurations. In this table, NF, In-NF, and Err corresponds to Noise Free, Input Noise Free, Error. N-In, N-Err corresponds to Noisy Input, Noisy Error signal.



(A)



(B)



(C)

Fig. 2. Action values for A) one target on the right-up corner of the workspace, B) two targets on the left-up and right-up corners of the workspace and C) four targets on each corner of the workspace.

TABLE I
BMI PERFORMANCE WITH SYNTHETIC AND SURROGATE DATA

	1 Target	2 Target	4 Target
NF	99.8%	98.3%	95.6%
Noisy Input	96.0%	80.6%	65.3%
Noisy Error	100%	99.2%	96.2%
N-In, N-Err	98.2%	78.0%	53.3%
Surrogate	14.1%	14.6%	14.2%

In order to test against the null hypothesis that the agent can reach the target without any structure in the input data we ran a test with surrogate dataset. Surrogate data was generated by reducing the tuning depth of all the neurons to zero creating a random time-series of firing rates. In this test, the error signal was computed the same way as in the experiments with tuned neurons. Performance of the BMI with surrogate data is presented along with the results with synthetic neural data in Table I.

The results in Table I demonstrate that performance of the system is more sensitive to the noise in the input rather than noise in the error. Noise in the error signal slightly improved the performance of the system because it helped the agent escape from local minima. Since the agent is most sensitive to the sign of the error, fluctuations in the amplitude did not degrade the performance. In general, increasing the number of targets decreases the performance however this decrease in the performance is more prominent with noisy motor commands. However, even with noisy input states and noisy error, the agent performed well in all target tasks.

We performed a random walk test to specify the probability of reaching a target by chance. Since the probability of reaching all the targets were the same, we computed the chance level for one target. For the random walk test, the actions of the agent were selected randomly at each step. The same limit on the number of steps per trials was applied to the random walk test. The probability of reaching a target by chance was 0.1%

IV. CONCLUSION

In this paper, we introduced a control architecture and adaptation procedure for decoding motor commands in MI based on an evaluative feedback from NAc which indicates user's goal. Performance of the BMI controller was studied in presence of noise in the error and input for three different tasks. Our results demonstrated the feasibility of this architecture in simulated biological constraints. The adaptive agent was able to navigate the robot to the targets in the continuum of space. Here we made no assumption about the location of the target in the space. In other words, the environment for the naïve agent was novel and no training was involved. The agent learned how to decode MI neural activity on-the-fly just based on an evaluative feedback from user. Since the reinforcement signal is innately extracted from the brain, the system is self-contained therefore it doesn't require an external source of information for reorganizing itself. Here, we focused on designing a decoder for MI; however, before adapting the decoder we need to characterize the user's neural response with respect to known targets in order to estimate the evaluative feedback in NAc.

The BMI design approach in this paper faces two main challenges. First, the performance of this system relies on extracting motor commands and evaluative feedback from the brain. Provided reliable signals are extracted from the brain this system is able to reach any point in the continuum of its workspace. The BMI demonstrated robust performance in presence of noise both in the motor commands and TD error signal. However the BMI was more sensitive to the noise in the neural commands than TD error.

The second challenge is adaptability of the actor using a winner-take-all approach in training the decoder. In this approach, only the parameters of the winning action are updated; therefore, over time some actions become less competitive compared to the winner and it is difficult for the BMI to reorganize itself for accomplishing the task. We are working to revise the update rule to provide the system with enough flexibility to converge to a new control policy.

However, from the results in Table I we can see in spite of decreasing the tuning depth of MI neurons and decreasing the SNR in the TD error, the agent had a good performance in reaching to multiple targets but the convergence time increased. This observation implies that using multiple agents, each specialized for a particular task, might be a better approach than having one agent to learn multiple tasks.

REFERENCES

[1] G. Montagne, et. al. "The learning of goal-directed locomotion: A perception-action perspective," *The Quarterly Journal of Experimental Psychology Section A,* vol. 56, pp. 551 - 567, 2003.

[2] M. A. Lebedev and M. A. L. Nicolelis, "Brain-machine interfaces: past, present and future," *Trends in Neurosciences,* vol. 29, 2006.

[3] J. M. Carmena, et. al. "Learning to control a brain-machine interface for reaching and grasping by primates," *Plos Biology,* vol. 1, 2003.

[4] L. R. Hochberg, et al "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature,* vol. 442, Jul 2006.

[5] M. Velliste, et. al. "Cortical control of a prosthetic arm for self-feeding," *Nature,* vol. 453, pp. 1098-1101, Jun 2008.

[6] S. Musallam, et. al. "Cognitive control signals for neural prosthetics," *Science,* vol. 305, pp. 258-262, Jul 2004.

[7] J. C. Sanchez, B. Mahmoudi, J. DiGiovanna, and J. C. Principe, "Exploiting co-adaptation for the design of symbiotic neuroprosthetic assistants," *Neural Networks,* vol. In Press, Accepted Manuscript.

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

[9] E. T. Rolls, C. McCabe, and J. Redoute, "Expected value, reward outcome, and temporal difference error representations in a probabilistic decision task," *Cereb Cortex,* vol. 18, Mar 2008.

[10] T. A. Hare, et. al, "Dissociating the role of the orbitofrontal cortex and the striatum in the computation of goal values and prediction errors," *Journal of Neuroscience,* vol. 28, pp. 5623-5630, May 2008.

[11] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science,* vol. 275, pp. 1593-1599, Mar 1997.

[12] P. W. German and H. L. Fields, "Rat nucleus accumbens neurons persistently encode locations associated with morphine reward," *Journal of Neurophysiology,* vol. 97, pp. 2094-2106, Mar 2007.

[13] R. K. Vijay and John N. Tsitsiklis, "On Actor-Critic Algorithms," *SIAM J. Control Optim.,* vol. 42, pp. 1143-1166, 2003.

[14] J. C. Principe, B. de Vries, and P. G. de Oliveira, "The gamma-filter-a new class of adaptive IIR filters with restricted feedback," *Signal Processing, IEEE Transactions on,* vol. 41, pp. 649-656, 1993.

[15] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans Neural Network,* vol. 14, pp. 1569-72, 2003.