35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Operant conditioning of a multiple degree-of-freedom brain-machine interface in a primate model of amputation

Karthikeyan Balasubramanian[1], Joshua Southerland[2], Mukta Vaidya[1], Kai Qian[6], Ahmed Eleryan[3], Andrew H Fagg[2], Marc Sluzky[5], Karim Oweiss[3,4], Nicholas Hatsopoulos[1]

Abstract— **Operant conditioning with biofeedback has been shown to be an effective method to modify neural activity to generate goal-directed actions in a brain-machine interface. It is particularly useful when neural activity cannot be mathematically mapped to motor actions of the actual body such as in the case of amputation. Here, we implement an operant conditioning approach with visual feedback in which an amputated monkey is trained to control a multiple degree-of-freedom robot to perform a reach-to-grasp behavior. A key innovation is that each controlled dimension represents a behaviorally relevant synergy among a set of joint degrees-of-freedom. We present a number of behavioral metrics by which to assess improvements in BMI control with exposure to the system. The use of non-human primates with chronic amputation is arguably the most clinically-relevant model of human amputation that could have direct implications for developing a neural prosthesis to treat humans with missing upper limbs.**

I. INTRODUCTION

Cortically controlled brain-machine interfaces (BMIs) utilize voluntary modulation of cortical neurons to control an external device [1][2][3]. To accomplish this, a biomimetic decoder is often built which maps a set of neural inputs to a set of kinematic or kinetic motor outputs. However, this approach in untenable in cases where concurrent neural modulation and motor behavior cannot occur as with motor paralysis and amputation.

Building on the pioneering work of Fetz and colleagues [4][5], we have adopted an approach using operant conditioning with biofeedback (OCB) to train a non-human, amputated monkey to control a multiple degree-of-freedom (DOF) robotic arm and hand. Given an initialized, crude decoder, the animal learns to control the device by modifying neural modulation in order to increase the likelihood of a desired motor output via positive reinforcement, namely a juice reward. This approach relies on associating the reward with sensory (i.e. visual) feedback of the desired motor output. We discuss the synergistic control of joint DOFs, defined as a control dimension, using neurons sampled from M1 contralateral to the amputation. We also present two methods for initializing the decoder. We discuss the system architecture of the BMI, the operant conditioning approach, and behavioral metrics that demonstrate improved performance of robotic control using two control dimensions associated with reach and grasp. We suggest that this approach provides a unique opportunity to study and examine motor skill development.

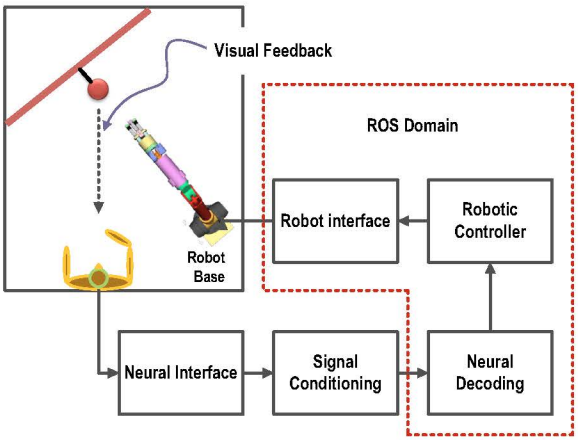

Fig. 1. Schematic of the experimental setup. Neural spike signals sampled from the M1 contralateral to the amputation were conditioned and decoded to generate control signals for the robot. Blocks within the dotted region are implemented as software nodes in the Robotic Operating System (ROS) platform.

II. SYSTEM ARCHITECTURE

A. Closed Loop BMI

The top-level diagram of the experimental setup is shown in Figure-1. Multi-electrode arrays (from Blackrock Microsystems,Inc., Salt Lake City, UT) were chronically implanted on the primary motor cortex (M1) region, bilaterally. Sorted action potentials acquired from the hemisphere contralateral to the amputation were used for the BMI discussed here. The surgical and behavioral procedures involved in this study were approved by the University of Chicago Institutional Animal Care and Use Committee and conform to the principles outlined in the Guide for the Care and Use of Laboratory Animals.

In this study, the task space was limited to two control dimensions. Each control dimension represented a synergy of joint DOFs which corresponded to either 1) reaching of

This work was supported by DARPA Grant No. N66 001-12-1-4023

[1] is with the Committee of Computational Neuroscience, University of Chicago, IL USA karthikeyanb at uchicago.edu

[2] is with the Department of Computer Science and Bioengineering, University of Oklahoma, OK USA

[3] is with the Department of Electrical and Computer Engineering, Michigan State University, MI USA

[4] is with the Neuroscience Program, Michigan State University, MI USA

[5] is with the Departments of Neurology, Physiology and Physical Medicine & Rehabilitation, Northwestern University, IL USA

[6] is with Department of Biomedical Engineering, Illinois Institute of Technology, IL USA

the hand along the X-axis in Cartesian coordinates away from and toward the base of the robot (See Figure-1) or 2) grasping by opening or closing all three digits of the robotic hand. Each control dimension was controlled by an independent cluster of neurons from M1. Selection of these neuronal clusters was based on their stability across consecutive days and on their functional connectivity [6]. Action potential spikes, binned at 50 milliseconds, from these neuronal population were used in decoding velocity signals to control the robot.

B. Decoder Initialization

Considerable data indicate that the firing rates of single neurons in M1 encode joint and Cartesian velocities more effectively than position [7][8]. Neural firing rates among a population of M1 neurons contralateral to the amputation were mapped to two "desired" velocity profile templates inspired by the minimum-jerk velocity profiles that characterize biological reaching [9][10]. Two approaches were used for decoder initialization,

1) Ipsilateral Arm Movement: Neural activities were sampled with the monkey performing an ipsilateral arm movement, and mapped to the velocity template of the reach control dimension.

2) Observation-based: For the grasp control, neural firings were sampled as the monkey was observing pre-programmed grasp trials performed by the robot. Observation-based motor learning is reported in [11][12].
The decoder was a 20-tap delay linear filter estimating the instantaneous velocity at 20 Hz. The system of equations,

$$\hat{y} = X\beta \tag{1}$$

was solved for β using a Ridge regression estimator, given by,

$$\beta = (X^T X + \lambda I)^{-1} X^T y \tag{2}$$

where, $\hat{y}$ is the decoded velocity for a given neural activity matrix X. The ridge parameter λ is to ensure that the inverted matrix has a condition number no larger than 10^3. Estimators based on ridge regression are known to reduce the variance of the estimate while introducing bias. The estimated (decoded) velocity was then fed to the robot.

C. Robotics

The robot comprised a 7 DOF redundant arm (i.e., the WAM) with a 4 DOF hand (i.e., the BarrettHand) (Barrett Technology, Inc.). The hand was slightly modified to reflect thumb abduction in a more biologically plausible way. In order to limit the movement of the robotic arm in 1D, a fixed-length minimum-jerk trajectory of $\sim$ 18 centimeters was defined in the Cartesian position along the X-axis of the robot movement and the corresponding joint space was sampled. The derivative of this trajectory defined the desired velocity trajectory onto which the neural activity matrix was mapped. The robotic arm was operated at 500 Hz, internally, via CAN (controller area network) bus communication, in real-time, enabled by a Xenomai kernel. The hand was operated at a lower frequency of 30 Hz. The decoded neural output was received every 50 milliseconds (i.e., 20 Hz), which was then linearly interpolated accordingly to match the operating frequency of the robot and the hand.

1) Reach control dimension: Decoded velocity could move the endpoint of the WAM arm either away from or toward the base of the robot. A small deadband was introduced so that tiny fluctuations in the velocity signals are filtered.

2) Grasp control dimension: Manipulation in the human hand involves up to 20 different joints of the hand. However, grasping of everyday objects typically involves the coordination of multiple joints which work together synergistically. Psychophysical data have shown that only a small number of independent joint synergies or principal components can account for a large proportion of the kinematic variance [13][14]. In particular, the first principal component represents an opening/closing of the entire hand [15]. Therefore, we chose to define a grasping control dimension that involved opening and closing of all three BarrettHand digits concurrently.

D. Robotic Controller

The low level control of the robot was accomplished using a PID controller. The velocity decoder provided non-smooth inputs at a low control rate (20 Hz). In order to improve the stability and smoothness of the system, the following changes were made to the PID controller:

- use a low-pass filter to smoothen the low-amplitude, high-frequency velocity oscillations
- interpolate the position error in joint space at 500 Hz
- impose a limit on the magnitude of the position error in joint space

The position error is the difference between the current state of the robot in Cartesian space and the commanded or desired position.

E. Robotic Operating System (ROS)

The control system was implemented in the ROS platform, an opensource library collection for robotic controls [16]. The platform can operate either as a publisher-subscriber architecture or as a service module. The former architecture enables concurrent data communication between multiple nodes, and was adapted for the present implementation. Essentially, ROS served as a middleware in the communication between the different processes and the machine/robot. Dynamic addition of processes is possible with the ROS platform, enabling simultaneously testing of several decoders during runtime. The present system was comprised of five process nodes, i.e., neural interface, decoder, controller, robot interface and reward system nodes. An additional data logger node was also used to capture all key variables for subsequent analysis, including cell activity, decoded signals, robot control signals and state, and protocol events. The schematic in Figure-2 shows the implemented system. The entire system was implemented as a finite state machine (FSM) that received state signals from the WAM and the

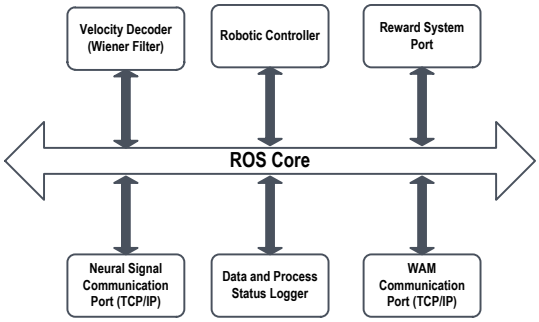

Fig. 2. The ROS platform comprises a set of core libraries that serve as middleware in robotic communications. The nodes are shown as blocks connected to the core. Each node can subscribe messages (data, state, etc.) from multiple nodes and can generate appropriate publishings. The current implementation used TCP/IP to communicate with the neural interface as well as with the WAM

hand, and generated control signals to activate a reward or update the state of the robot.

III. Reinforcement Learning: The Art of Conditioning

Reinforcement learning is a form of operant conditioning by which the likelihood of a behavior is increased based on a consequence that is reinforcing, such as a reward. However, the reinforced behavior is often quite complex and, therefore, difficult to learn. Shaping is a procedure by which a complex task is broken down into a sequence of simpler components or modules each of which can be conditioned. By gradually increasing the number of components that need to be sequenced for a reward to be given, the complete, complex task can be learned. The task of reaching to grasp an object occurs as a coordinated composite behavior in volitional motor control. We compartmentalized the task into two phases. Within a given task component, shaping was implemented by defining a threshold proportion, Th, of the entire component movement. For the reach component, we started with a threshold of 0.5 such that the monkey had to generate an integrated velocity command to move the WAM end-point 50% of the way to the object after which the robot software completed the reach and grasp. Gradually, the threshold was moved further towards the target location until the entire reach was controlled by the monkey. Similarly, for the grasp component, a threshold was set at 60% of the aperture closing and opening during the initial phases of training. It is worth mentioning that as the monkey was learning the grasp phase, she still had to perform the entire reach component for a successful trial.

A. Performance Evaluation

Three key metrics were used to quantify improvements in performance due to learning: number of successful trials per unit time, mean time-to-perform a successful trial and normalized path length. The number of successful trials were calculated in 10 minute blocks. Normalized path length is the total path length travelled by the robot in a given successful trial, normalized by the Euclidean distance between the starting and the target location.

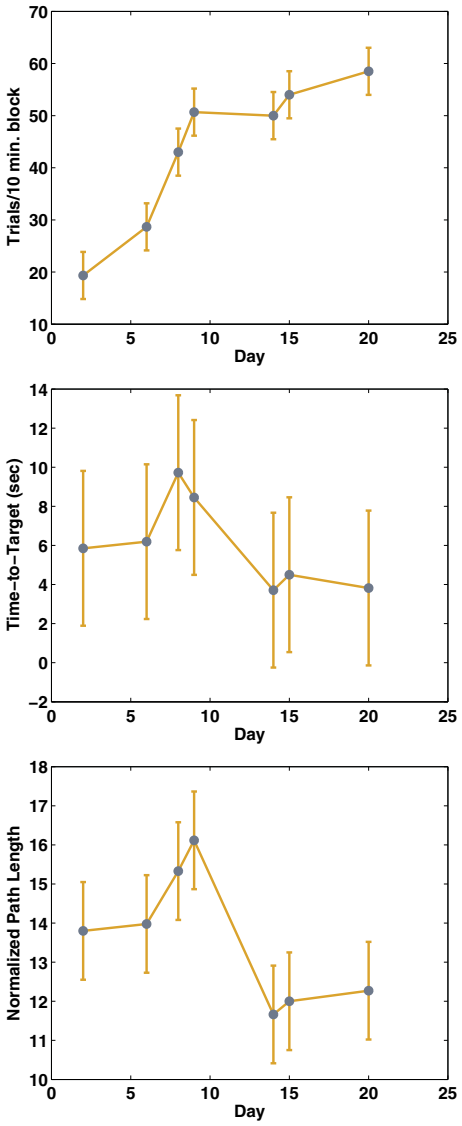

Fig. 3. Performance Metrics for the reach control dimension. (Top) shows the number of successful trials per unit time (10 minutes). (Middle) shows the time needed to reach the target. (Bottom) shows the normalized path length. Error bars show the *standard error*. These metrics do not explicitly reflect the task complexity variations associated with *shaping*

Quantifications based on these metrics are useful when the task complexity remains the same over the period of comparison. They, however, do not account for a varying task difficulty, as in the case of the *shaping* approach. We present these standard metrics of performance for the reach control dimension in Figure-3.

The training remained with $Th = 50\%$ on day-2, and between days 6 and 10 it was increased gradually to 80%. This manifests itself as increased time-to-target and total path length in the plot (see Figure-3). Nevertheless, the animal remained motivated towards maximizing the reward, resulting in consistent increase in the number of successful trials. The other two metrics also showed improvement with subsequent training. Furthermore, once both components

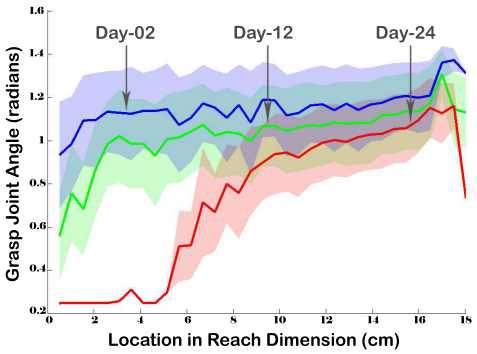

Fig. 4. Improved coordination of reach and grasp. The plot shows the joint angle of the fingers along the trajectory of *reaching*. A 0-radians is completely open hand and 1.5-radians is completely closed. During the initial days of the training, the monkey had uncoordinated aperture control while reaching for the object. Later, the monkey was able to maintain a relatively open hand posture at the start of the trajectory and closed the hand as the object was reached (see Day-24 in the plot)

were under the monkey's control we observed improved coordination between the reach and grasp components of the movement. Figure-4 shows the grasp aperture as a function of reach. As the learning occurs, the monkey was able to maintain an improved posture in terms of the aperture control while navigating through the reach control dimension.

IV. Discussion

A key application of BMIs is restoration of motor functions lost due to physical impairment or neural dysfunction. BMIs often rely on neural plasticity to achieve the required motor skill. Experiments conducted on subjects with intact limbs have provided insights into how the neural substrates tune themselves towards a given task/skill. In this work, we have recruited neural substrates from the M1 that were deafferented for several years. With consistent exposure and reinforcement, the monkey was able to modify neural modulation in order to accomplish a reach-to-grasp task.

Two different decoder initialization paradigms were tested: 1) mapping the neural activity from ipsilateral arm movements, used for the reach control dimension and 2) sampling the neural firing rates of observation-based trials, to use with the grasp control dimension. These neural signals were then mapped against the synthetic velocity profile and solved for the filter. Scaling the observation-based decoder initialization technique to other control dimensions is relatively straightforward. The framework itself is quite generic, modular and scalable. However, only limited assertions can be made about the possibility of strong modulations occurring during observations of fine motor tasks, as in cases of thumb abduction/adduction, etc.

The motor learning was also not without challenges. Since we were sampling a very sparse population of the neurons at the M1 to control an external device, maintaining an internal state of the brain was hard to achieve and irrelevant for the monkey. Assuming that a forward moving signal needs to be generated, it was not possible for the monkey without making any undesirable reversals. This is evident from the overall path length taken by the monkey to complete a successful reach (see Figure-3 (bottom)). Nevertheless, the monkey was able to generate a cumulative positive or negative velocity. Delays in the mechanical system of the robot, such as inertial and torque-related delays are not discussed here, and the compensatory learning mechanism employed by the monkey for these delays are not studied, though it could be a subject of interest.

V. Conclusions

We have presented a BMI framework that employs operant conditioning with biofeedback to generate a useful interface in an amputated subject. We used a multiple DOF robot and demonstrated a 2 DOF cortical control with a rhesus monkey. Performance evaluation showed occurrence of motor-task learning. The experimental setup and the robotic control systems discussed here are generic and can be scaled easily for additional DOFs. Examining the amputated animal model could provide us with unprecedented insights towards developing a robust and reliable neural prosthesis.

References

[1] J. P. Donoghue, Connecting cortex to machines: recent advances in brain interfaces, Nat Neurosci, vol. 5, Oct. 2002, pp. 1085-1088

[2] M. A. L. Nicolelis, Actions from thoughts, Nature, vol. 409, Jan. 2001, pp. 403-407

[3] D. M. Taylor, S. I. H. Tillery, A. B. Schwartz, Direct cortical control of 3D neuroprosthetic devices, Science, vol. 296, no. 5574, Jun. 2002, pp. 1829-1832

[4] E. E. Fetz, Operant Conditioning of cortical unit activity, Science, vol. 163, no. 3870, Feb 1969, pp. 955-958

[5] E. E. Fetz and M. A. Baker, Operantly conditioned patterns on precentral unit activity and correlated responses in adjacent cells and contralateral muscles, vol 36, no. 2, Mar. 1973, pp. 179-204

[6] S. Eldawlatly, R. Jin, K. G. Oweiss, Identifying functional fonnectivity in large-scale neural ensemble recordings: A multiscale data mining approach, vol. 21, no. 2, Feb. 2009, pp. 450-477

[7] L. Paninski, M. R. Fellows, N. G. Hatsopoulos and J. P. Donoghue, Spatiotemporal Tuning of motor cortical neurons for hand position and velocity, J Neurophysiol, vol. 91, no. 1, Jan. 2004, pp. 515-532

[8] M. Saleh, K. Takahashi and N. G. Hatsopoulos, Encoding of coordinated reach and grasp trajectories in primary motor cortex, J Neurosci., vol. 32, no. 4, Jan. 2012, pp. 1220-1232

[9] T. Flash and N. Hogan, The Coordination of arm movements: an experimentally confirmed mathematical model, J Neurosci, vol. 5, no.7, Jul. 1985, pp. 1688-1703

[10] R. Shadmehr and S. P. Wise, Computational neurobiology of reaching and pointing A Foundation for Motor Learning, MA: MIT Press, 2005, ch. 18

[11] D. Tkach, J. Reimer, and N. G. Hatsopoulos, Congruent activity during action and action observation in motor cortex, J Neurosci, vol. 27 no. 48, Nov. 2007 pp. 13241-13250

[12] D. Tkach, J. Reimer, and N. G. Hatsopoulos, Observation-based learning for brain-machine interfaces, Curr Opin Neurobiol, vol. 18 no. 6, Dec. 2008 pp. 589-594

[13] C. R. Mason, J. E. Gomez and T. J. Ebner, Hand synergies during reach-to-grasp, J Neurophysiol, vol. 86, no. 6, Dec. 2001, pp. 2896-2910

[14] M. Santello, M. Flanders and J. F. Soechting, Patterns of hand motion during grasping and the influence of sensory guidance, J Neurosci, vol. 22, no. 4, Feb. 2002, pp. 1426-1435

[15] M. Ciocarlie, C. Goldfeder and P. Allen, Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem, Proceedings of the Robotics: Science and Systems - Robot Manipulation: Sensing and Adapting to the Real World, Jun. 2007

[16] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler and A. Ng, ROS: An open-source robot operating system, Proc. Open-Source Software Workshop Int. Conf. Robotics and Automation, May 2009