Repairing Lesions via Kernel Adaptive Inverse Control in a Biomimetic Model of Sensorimotor Cortex

Kan Li\textsuperscript{1}, Salvador Dura-Bernal\textsuperscript{2}, Joseph T. Francis\textsuperscript{2}, William W. Lytton\textsuperscript{2}, and José C. Príncipe\textsuperscript{1}

Abstract—In this paper we propose a kernel adaptive filtering (KAF) approach to repairing lesions via microstimulation in a biomimetic spiking neural network of sensorimotor cortex. The fundamental challenge of designing neuroprosthetics and brain machine interfaces (BMIs) is the decoding of electrical activity of neurons and behavior. For injured or damaged brain, intracranial stimulation has the potential to modulate neural activity to match meaningful and natural response or behavior. In order to optimize the microstimulation sequences, we construct an inverse model of the target system. However, to obtain sufficient learning data, the neural system must be stimulated or probed extensively. For real brains, this is especially challenging and often unfeasible. Here, we demonstrate that by applying KAF to a biomimetic brain and realistic virtual musculoskeletal model, we can repair simulated lesion and drive a virtual arm to perform the correct motor task.

I. INTRODUCTION

Brain machine interfaces (BMIs) connect the brain with external devices by establishing communication directly between the central nervous system and artificially engineered neural prosthetics. Cognitive, motor, and sensory BMIs for direct neural control have far reaching impact in rehabilitation and the understanding of brain functions [1]–[3]. At the core of the BMI framework is the decoding of brain processes involved in communication and control tasks, by learning the functional mapping between the electrical activity of neurons and behavior.

Intracranial stimulation can deliver information directly to the brain and drive neural response to match meaningful activity. Applying optimally designed low-power electrical signals (microstimulation) to intracranial microelectrodes may elicit motor behaviors mimicking responses to natural sensory stimuli [4], [5]. This approach can be used to repair brain lesions by compensating for missing activity or by inducing plasticity which can lead to recovery [6]–[10].

In [11], the kernel least mean square (KLMS) algorithm [12], specifically the quantized KLMS (QKLMS) [13], is used to estimate the dynamic nonlinear mapping from neural responses to the stimuli. This approach exploits the fact that linear signal processing in a reproducing kernel Hilbert space (RKHS) corresponds to nonlinear processing in the input space and can be used in an adaptive inverse scheme designed for controlling neural responses.

However, in order to obtain the optimal microstimulation sequences, it is necessary to construct an inverse model of the target system. For real brains, this is especially challenging, and often unfeasible, as it requires stimulating the neural system repeatedly to obtain sufficient probing data for reconstruction and depends on an unwarranted assumption of stationarity. In contrast, a biomimetic model (BMM) of the brain provides an attractive alternative testbed for understanding the interactions between ongoing neural activities and artificial stimulations. Unlike real brains, in silico brains can be probed extensively and precisely, providing access to detailed information of all the neurons and synapses in the network. Furthermore, different types of lesions and repair methods can be simulated and evaluated with ease.

In this paper we repair a simulated lesion by optimizing a set of microstimulation patterns that compensate for missing activity in a spiking network model of sensorimotor cortex, which controls a realistic virtual musculoskeletal arm to perform reaching tasks. Unlike [11] which continuously outputs a set of microstimulation, we wish to derive a short burst of repairing pulse pattern in the early stage of each reaching task which corrects the lesioned trajectory.

The BMM of sensory and motor cortex consists of several hundred spiking model neurons, as shown in Fig. 1 [14], [15]. It is trained using spike-timing dependent reinforcement learning (RL) to drive a realistic virtual musculoskeletal arm in a motor task requiring convergence on a single target. The virtual musculoskeletal arm receives input from the BMM signaling neural excitation for each muscle. It then feeds back realistic proprioceptive information, including muscle fiber length and joint angles, which are employed in the RL process. Previous studies have shown that this BMM can be interfaced in real-time with neurophysiological data from real brains [16], as well as with a robotic arm [17], potentially allowing for a full closed-loop neural prosthetic system.

Next, the trained network is perturbed by silencing 20 (10.42\%) of its excitatory sensory (ES) neurons or cells in order to simulate a lesion. After perturbation, the virtual-arm reach trajectory is severely impacted and the network is no longer capable of completing the original reaching task. The remaining ES cells in the BMM are probed using microstimulation pulse patterns. The network activity resulting from the probing sequences are then used to construct an inverse model of the BMM’s motor layer, using kernel adaptive inverse control of neural spatiotemporal spike patterns. The inverse model is constructed using kernel adaptive filtering.

This work was supported by DARPA Contract N66001-10-C-2008.

\textsuperscript{1}K. Li and J. C. Príncipe are with the Department of Electrical and Computer Engineering (Computational NeuroEngineering Laboratory), University of Florida, Gainesville, FL 32611 USA (email: {likan, principe}@ufl.edu).

\textsuperscript{2}S. Dura-Bernal, J. T. Francis, and W. W. Lytton are with the Department of Physiology and Pharmacology, State University of New York Downstate Medical School, Brooklyn, NY 11203 USA (email: {salvadordura, joey199us}@gmail.com, billl@neurosim.downstate.edu).
where $\mathcal{K}(\mathbf{u}, \mathbf{u}')$ is a Mercer kernel, corresponding to the inner product $(\varphi(\mathbf{u}), \varphi(\mathbf{u}'))$, and $\alpha_i$ are the weight coefficients. The most commonly used kernel is the Gaussian kernel

$$\mathcal{K}_\alpha(\mathbf{u}, \mathbf{u}') = \exp \left( -\alpha \| \mathbf{u} - \mathbf{u}' \|^2 \right)$$  \hspace{1cm} (3)$$

where $\alpha > 0$ is the kernel parameter. To effectively address the growth of the radial basis function structure in KAF, the QKLMS algorithm is used [13].

A. Reproducing kernel Hilbert space (RKHS) for spike trains

A spike train or sequence of $M$ ordered spike times, i.e., $\mathbf{s} = \{ t_m \in \mathcal{T} : m = 1, \ldots, M \}$ in the interval $\mathcal{T} = [0, T]$, can be viewed as a realization of an underlying stochastic point process with conditional intensity function $\lambda(t|H_i^t)$, where $t \in \mathcal{T} = [0, T]$ denotes the time coordinate and $H_i^t$ is the history of the process up to $t$. Spike trains can be mapped into an RKHS by defining a strictly positive definite kernel, the Schoenberg kernel, between the conditional intensity functions of two point processes [21] as

$$\mathcal{K}(\lambda(t|H_i^t), \lambda(t'|H_j^t)) = \exp \left( -\frac{\int (\lambda(H_i^t) - \lambda(H_j^t))^2 dt}{\Delta t} \right).$$

The intensity function can be estimated by convolving the spike times with a smoothing function $g(t)$, yielding

$$\hat{\lambda}(t) = \sum_{m=1}^M g(t - t_m), \{ t_m \in \mathcal{T} : m = 1, \ldots, M \}.$$  \hspace{1cm} (5)$$

For simplicity, we use the rectangular function $g(t) = \frac{1}{\Delta t} (U(t) - U(t - T))$, where $T \gg$ the inter-spike interval, and $U(t)$ is a Heaviside function. Let $s_i^n(t)$ denote the spike train for the $i$-th sample of the $n$-th spiking unit, the multi-unit spike kernel is taken as the unweighted sum over the kernels on the individual units

$$\mathcal{K}(\mathbf{s}_i(t), \mathbf{s}_j(t)) = \sum_n \mathcal{K}(s_i^n(t), s_j^n(t)).$$

As shown in Fig. 2, the goal is to learn an inverse model of the plant $\mathbf{P}$, which is the lesioned BMM motor layer, and then apply the pre-lesioned motor response to the trained multiple-input-multiple-output (MIMO) decoding model for a set of optimized repair microstimulation patterns.

III. RESULTS

A BMM of sensorimotor cortex controlling a realistic virtual musculoskeletal arm (Fig. 1) is trained to perform a one-second target reaching task. After training, it is perturbed by silencing the first 20 (out of 192) excitatory somatosensory neurons in its sensory layer, in order to simulate a lesion. After perturbation, the virtual-arm reach trajectory is severely impacted and the BMM is no longer capable of completing the original reaching task.

To construct the inverse model of the damaged neural system, each of the remaining 172 ES cells were probed individually and in small, random groups of 1-20 neurons, with fixed-duration (200 ms) microstimulation sequences. Pulse frequency of 250 Hz or 500 Hz is used for single-neuron stimulation, and a random frequency in the range of 100-500 Hz is used for multiple-neuron stimulation. The BMM is stimulated starting at either the 200 ms or 400
Lesioned BMM Motor Layer

Inverse Model of Lesioned BMM Motor Layer

Error

N-channel Probing Motor Response

Inverse Model Output

M-channel Probing Microstimulation Firing Rates

Time-delayed response

N-channel Probing Motor Response

Fig. 2. System diagram of the adaptive inverse model of lesioned BMM.

Fig. 3. A. Raster plot of a multiple-neuron microstimulation probing sequence. B. Virtual arm with superimposed hand trajectories for single-neuron (light red) vs. multiple-neuron (dark red) microstimulation probing sequences (original trajectory in blue and target in green).

ms mark of a 1 s trial. The combination of start times, stimulation sites (individual neuron or group), and pulse frequencies corresponds to a training set of 1376 unique microstimulation probing patterns. For each pattern, the output motor-neuron population activities and the virtual arm trajectories were recorded, as shown in Fig. 3.

The 688 single-site stimulation sequences and their corresponding excitatory motor (EM) neuron population responses (which drive the virtual arm) are used to build an inverse model of the lesioned BMM using the QKLMS algorithm. After training, the 96-channel-input-172-channel-output decoding model takes shifting windows (in increments of 1 ms) of spike trains of duration 400 ms from the BMM’s motor-neuron population and returns the estimated optimal microstimulation pattern: rates and starting times in a 400 ms window (from 200 ms to 600 ms). Since the system is time-variant, the past 200 ms of the motor layer response is used in a feedback, along with the current response (200 ms), at the input of the inverse model.

Fig. 4 shows the estimated optimal microstimulation by feeding the desired motor-neuron activity from the pre-lesioned BMM into the inverse model. We see that the estimated optimal microstimulation from the kernel adaptive inverse model can be decomposed into distinct single-neuron stimulation sites and pulse frequencies. Next, we applied the predicted microstimulation from the inverse model to the lesioned BMM. As shown in Fig. 5, we are able to repair the damage and drive the virtual arm to the correct target. By judiciously selecting a subpopulation (3 neurons out of 172) of the excitatory sensory cells and stimulating them for only a short period of time (100 ms) in the early stage of a motor task, we are able to restore the correct reaching behavior in the lesioned biomimetic model of sensorimotor cortex.

IV. CONCLUSION

Using kernel adaptive filtering technique on spike trains, we are able to map a desired neural response into a set of repair microstimulation patterns for a lesioned biomimetic spiking network. This work demonstrates the potential of cortical prostheses to dynamically repair damaged brain regions and the corresponding motor behaviors using biomimetic brain and musculoskeletal models. In the future, we will extend this research to more complex motor tasks involving multiple targets and different types of simulated lesions.
Fig. 5. A. Hand trajectory of the virtual arm and raster plot of the original (pre-lesioned) BMM trained using reinforcement learning (target in green). B. Hand trajectory of the virtual arm and raster plot of the lesioned BMM. C. Hand trajectory of the virtual arm and raster plot of the repaired BMM.

REFERENCES