9th International IEEE EMBS Conference on Neural Engineering
San Francisco, CA, USA, March 20 - 23 , 2019

Decoding Kinematics from Human Parietal Cortex using Neural Networks

Sahil Shah[1], Benyamin Haghi[1], Spencer Kellis[2,3], Luke Bashford[2], Daniel Kramer[3], Brian Lee[3], Charles Liu[3], Richard Andersen[2] and Azita Emami[1]

Abstract— **Brain-machine interfaces have shown promising results in providing control over assistive devices for paralyzed patients. In this work we describe a BMI system using electrodes implanted in the parietal lobe of a tetraplegic subject. Neural data used for the decoding was recorded in five 3-minute blocks during the same session. Within each block, the subject uses motor imagery to control a cursor in a 2D center-out task. We compare performance for four different algorithms: Kalman filter, a two-layer Deep Neural Network (DNN), a Recurrent Neural Network (RNN) with SimpleRNN unit cell (SimpleRNN), and a RNN with Long-Short-Term Memory (LSTM) unit cell. The decoders achieved Pearson Correlation Coefficients (ρ) of 0.48, 0.39, 0.77 and 0.75, respectively, in the Y-coordinate, and 0.24, 0.20, 0.46 and 0.47, respectively, in the X-coordinate.**

I. INTRODUCTION

In the United States there are about 17,700 new cases per year of Spinal Cord Injury (SCI) [1]. SCI results in a partial or total loss of motor function. Brain-Machine Interfaces (BMI) have the potential to increase independence and improve quality of life in SCI patients by reading out neural signals and mapping them onto control signals for assistive devices [2]. There have also been efforts to use BMI to directly control paralyzed muscles [3], [4] and to decode speech signals from neural data [5], [6].

Data used in this work were recorded from the posterior parietal cortex of a tetraplegic human research participant [7]. Cells in this region have been shown to encode the goal of movements [8] and are also involved in sensorimotor integration and high-level motor planning [9]. These findings suggest that neural signals recorded from the parietal lobe could be useful for a variety of BMI tasks.

Figure 1 shows a general setup for a BMI system. Because BMI systems serve as an interface between the cortex and peripheral devices, they need to be robust over time in the face of different sources of variability. For example, electric potentials in the cortex have small amplitudes and are susceptible to noise, and electrical and mechanical properties of implanted microelectrodes change over time. Neuronal populations may also change over time. Decoders should be able to generalize across sources of variability to accurately infer movement commands from changing neural signals.

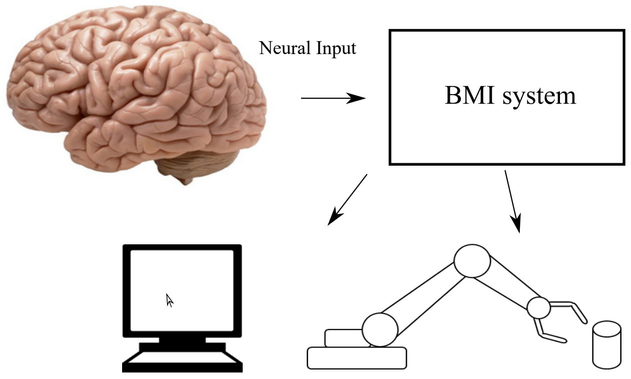

Fig. 1. General setup of a Brain-Machine Interface (BMI) system. BMIs enable direct control of computers, prosthetics and other peripheral devices by reading out and decoding brain activity. Advanced machine learning paradigms such as neural networks may be capable of learning the potentially complex relationship between recorded neural activity and control signals for these peripheral devices.

Conventionally, the algorithms used for such a BMI system have assumed a linear relation between inputs and outputs (e.g., Kalman filters or Wiener filters) [10]. In recent years, due to progress made in machine learning and neural networks there has been an increased interest in adopting these novel techniques for BMI applications [11], [12]. In [11] Sussillo et al. implement a multiplicative RNN for decoding movements from motor cortex of non-human primates. Schwemmer et al. in [12] use neural networks to perform multiple-class classification for several different actions performed by a human subject. With sufficient training data, these powerful machine learning algorithms could generalize over large variations in the recorded data.

In this work we compare the accuracy of both linear and nonlinear decoders, including the Kalman filter, a two-layer Deep Neural Network (DNN), a Recurrent Neural Network (RNN) with SimpleRNN unit cell (SimpleRNN) [13], and a RNN with Long-Short-Term Memory unit cell (LSTM) [14]. We use Pearson Correlation Coeffiecient (ρ) as an accuracy metric. The data used for training was recorded from the parietal lobe of a tetraplegic subject while the subject performed a 2D center-out task using motor imagery. We report the accuracy of these decoders in open loop configuration, i.e. where the subject uses motor imagery while observing the task, but is not in the control loop.

*This IRB approved research was supported by Chen Institute for Neuroscience at the California Institute of technology (Caltech), Pasadena, CA USA 91125

[1] Dept. of Elec. Eng. at Caltech azita@caltech.edu
[2] Division of Biol. and Biol. Eng. at Caltech and the T&C Chen BMI Center at Caltech. andersen@vis.caltech.edu
[3] Dept. of Neurol. Surg. at USC and the USC Neurorestoration Center.

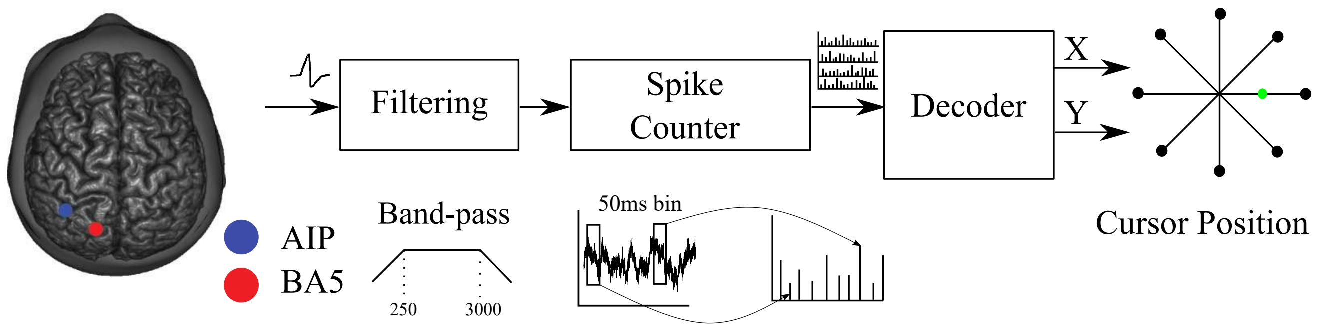

Fig. 2. System architecture for decoding neural signals into relevant kinematics. Broadband recorded data were bandpass filtered (250 Hz - 5 KHz) and thresholded at -4 times the noise RMS. Threshold crossing timestamps were binned in nonoverlapping 50ms intervals and smoothed to estimate the instantaneous threshold crossing rate. Decoding algorithms map these input features to corresponding X and Y coordinates of the cursor on screen.

II. Architecture for the BMI System and Methods

A. Subject, Implanted Electrodes and Recording

As part of an FDA- and IRB-approved study, two 96-channel Utah microelectrode arrays (Blackrock Microsystems, Inc., Salt Lake City, UT, USA) were implanted in the posterior parietal cortex of a 32-year-old tetraplegic subject with spinal cord lesions at C5-C6: one on the medial bank of the anterior intraparietal sulcus (AIP), and a second in Brodmann's area 5 (BA5) [7] (Figure 2). Data were recorded at 30,000 samples/sec.

B. Preprocessing the Neural Data

Figure 2 shows a top level block diagram of a BMI system. Broadband data were filtered (Butterworth filter, 250 Hz - 5 KHz) and thresholded at -4 times the noise RMS of each channel to identify neuronal action potentials. These spiking events were binned at 50 ms intervals and smoothed to create spike train features for the decoding algorithms. To match the online case as closely as possible, action potential waveforms were not sorted, and spike trains were computed from the raw threshold crossings.The spikes recorded from both the electrodes were processed as described above and used as features for the decoder.

C. Center-Out Reaching Task

In this work we use neural and behavioral data collected during the open-loop phase of a 2D center-out brain-control task. In this phase of the task, a cursor moves under computer control, with a minimum-jerk velocity profile, from the center of a computer screen to one of eight different target locations arranged uniformly around a unit circle, while the subject uses motor imagery to imagine controlling the cursor. Data is collected in three-minute blocks, each block consisting of 53 trials, with a pseudorandom uniform distribution of targets across trials. The dataset underlying this work consists of five such blocks recorded on the same day.

III. Algorithms and Results

We used this data to compare decoding performance between a Kalman filter, DNN, SimpleRNN, LSTM. LSTM and SimpleRNN algorithms are used for this work since the prediction task and the input neural data are sequential. The data were divided into training (80%), validation (10%) and test sets (10%). Training data was normalized to have zero mean and standard deviation of one to improve training algorithm convergence but test and validation data were normalized using scales learned from the training data. Time bins in which the cursor did not move (zero velocity) were excluded from analysis. In the case of the neural networks, separate decoders were trained for predicting X and Y coordinates (Figure 3(a)).

The standard Kalman filter uses a model of the kinematic system, and a linear model of the relationship between the kinematics and the neural data, to form new estimates of the kinematics from noisy measurements of neural data [10]. Variants of the Kalman filter support nonlinear dynamics, but in general, Kalman filters require the researcher to establish a model of the dynamical system. In contrast, neural networks learn the model from training data.

We used two different neural network paradigms: DNN and RNN. A DNN is a feedforward network with multiple layers and several nodes at each layer. The output of each node has a nonlinear activation function. DNNs with two layers have been shown to be a universal approximator [15]. A RNN is composed of feedforward network as well as a feedback network, meaning that all previous outputs are integrated to predict the next time-step (Figure 3(b)). RNNs also use previous time steps' input data when computing a new prediction. We tested two variants of RNN: one with LSTM unit cell [14] and one with the SimpleRNN unit cell [13].

A. Training and Accuracy of the Decoders

The neural networks were trained using Keras with tensorflow backend, and incorporate L1 regularization and 35% dropout for both the kernel and biases to reduce overfitting. An rmsprop optimizer was used for training the network [13]. All three neural networks use the hyperbolic tangent

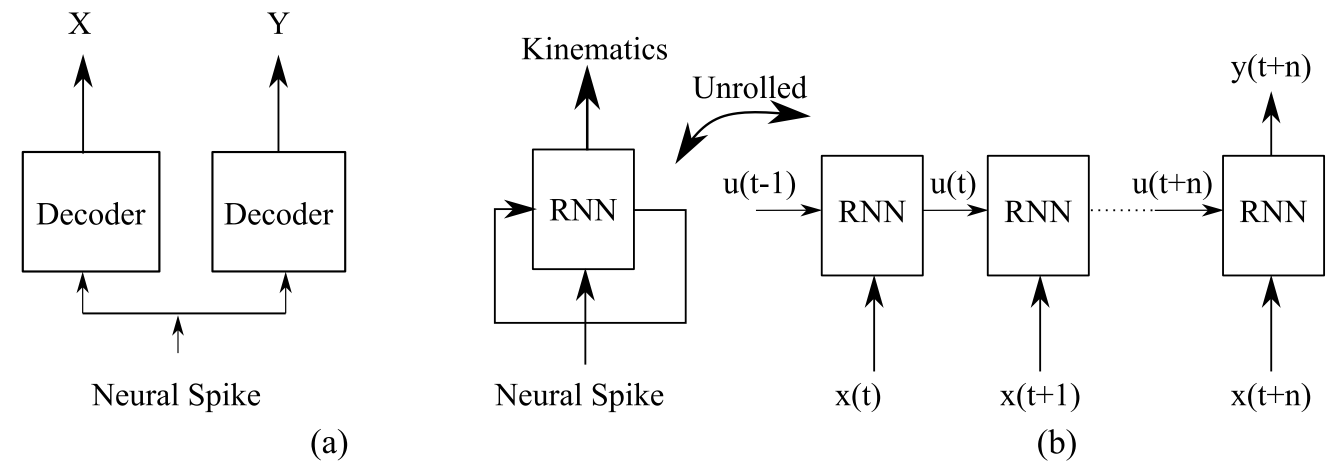

Fig. 3. Output of the decoding algorithm. (a) For the neural network algorithms, two separate decoders are used to predict X and Y position of the cursor. (b) A block diagram of RNN [13] with a single dense layer for regression. Also, an unrolled block diagram of RNN with multiple time-steps. The RNN unit can be either a fully connected SimpleRNN cell or an LSTM unit cell.

TABLE I
PARAMETERS FOR THE NEURAL NETWORKS

	Nodes	Layers	Previous Neural bins	Activation function
LSTM	10(X), 50(Y)	LSTM+NN	40	tanh
RNN	25(X), 25(Y)	SimpleRNN+NN	20	tanh
DNN	25(X), 25(Y)	NN+NN	1	tanh

TABLE II
PEARSON CORRELATION COEFFICIENT ρ FOR EACH DECODER

	Kalman Filter	DNN	SimpleRNN	LSTM
X	0.24	0.20	0.46	0.47
Y	0.48	0.39	0.77	0.75

as an activation function, and incorporate a dense layer with one node and a linear activation function at the output to perform regression. Network parameters were heuristically tuned; future studies will explore optimization techniques to tune these parameters for higher accuracy. In general, optimization techniques such as Bayesian optimization, grid search, random search etc. are used to choose optimal network parameters. The number of layers and nodes used for decoding were nominal to avoid overfitting, but with a larger dataset one could increase the size of the network to predict with consistent accuracy.

Table I summarizes the parameters used for training these neural networks. The DNN had two layers with the first layer of the DNN composed of 25 nodes. The LSTM network for X position was set to 10 nodes with 40 time-steps of prior neural data, and the Y position was set to 50 nodes with 40 time-steps. The SimpleRNN network used 25 nodes and 20 time-steps of previous neural data for both X and Y coordinates.

Table II shows the accuracy of the four different decoders. The RNN algorithms, with the ability to incorporate historical data to compute new predictions, achieved the highest performance. The DNN exhibited the lowest performance, likely because it uses only a single time step of neural data to predict the current kinematics. The Kalman filter performed better than the DNN, perhaps also because its iterative nature inherently captures prior state information to predict new states. Figure 4(a) and Figure 4(b) show the predicted X and Y coordinates of the cursor for the LSTM unit cell with a ρ of 0.47 and 0.75 respectively, and figure 4(c) and figure 4(d) show the predicted X and Y coordinates of the cursor with a ρ of 0.46 and 0.77.

IV. DISCUSSION

In this work we evaluate the performance of several different neural networks, and compare their performance to a standard Kalman filter. Algorithms with the ability to incorporate historical data and network state demonstrated the highest performance (LSTM and SimpleRNN with the highest accuracies, and the Kalman filter with the next highest performance). LSTM also has the ability to recognize long-term dependencies in the data. Network paradigms with interconnected nodes and integration of historical data and states, such as the RNN variants tested in this work, may prove critical to first capturing the complexities of the relationship between neural activity and kinematic output, and second providing stable performance for BMI users.

Our results showed a large difference in performance between X- and Y-dimension kinematics. These differences are most likely attributable to the specific neuronal population recorded for the data used in this work, which may comprise different proportions of neurons modulated by movement in either axis. It is also possible that the research participant's cognitive strategy led to these differences. Further data must be collected to understand the source of these differences.

Future work will test RNN decoders in closed loop to evaluate how well a human subject can use them for cursor control. Stability of the decoder over multiple days will also be evaluated. Also, this will determine whether the capability

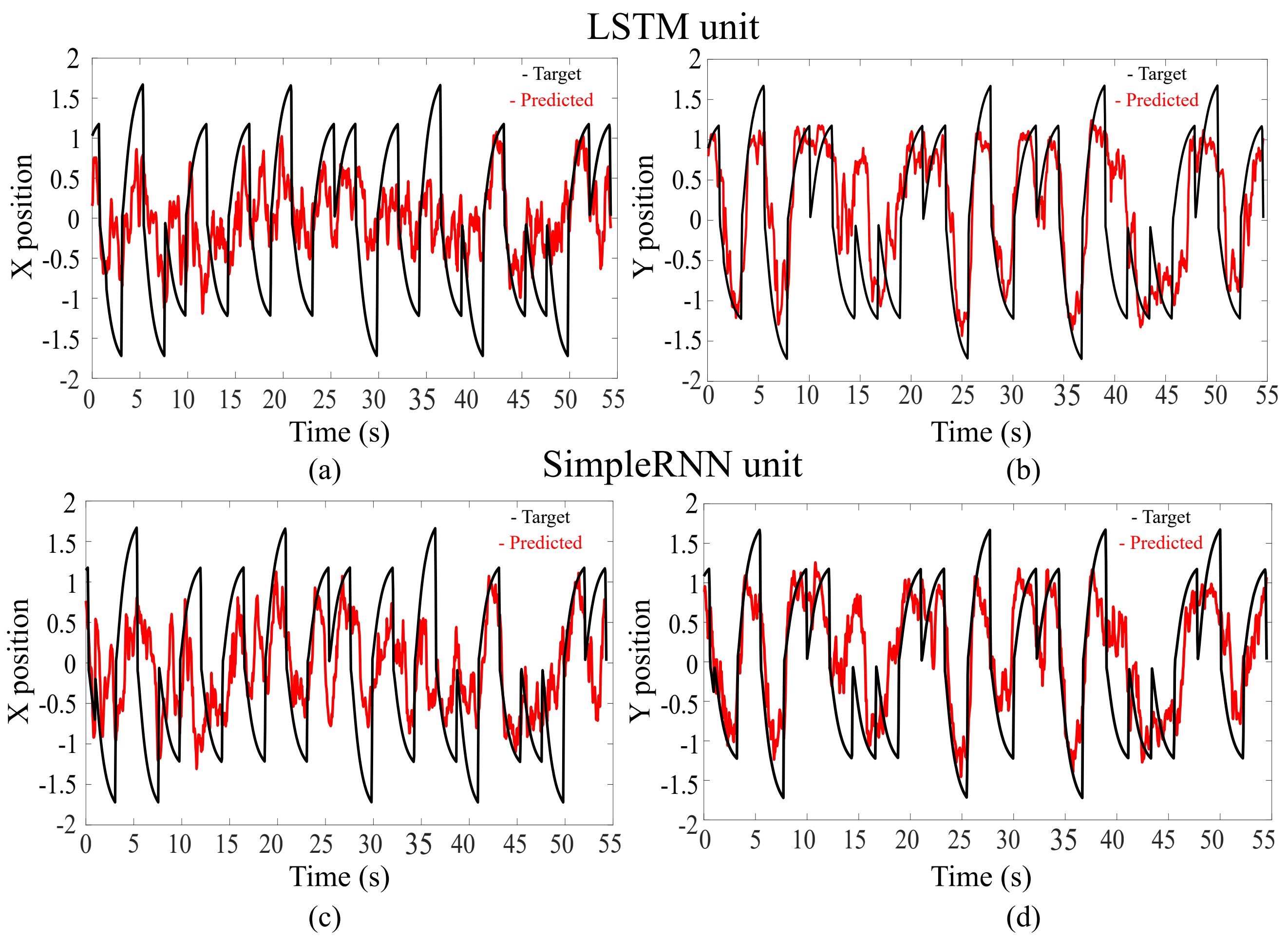

Fig. 4. (a) Output of a RNN with LSTM unit cell predicting the X coordinates of the cursor ($\rho = 0.47$). (b) Output of a RNN with LSTM unit cell predicting the Y coordinates of the cursor ($\rho = 0.75$). (c) Output of the decoder with SimpleRNN unit cell predicting X-coordinates of the cursor ($\rho = 0.46$). (d) Output of a RNN with SimpleRNN unit cell predicting the Y coordinates of the cursor ($\rho = 0.77$).

of the LSTM to capture long-term dependency leads to better performance over time.

While these algorithms are powerful in their capacity to capture complex relationships, they currently require power-hungry computational resources to operate. Part of making BMI systems clinically relevant is to design and develop size- and power-efficient hardware for decoding kinematics such that these systems can be implanted or worn on the body. Future directions would involve exploring such novel algorithms and energy-efficient hardware.

References

[1] Center, "Facts and figures at a glance," University of Alabama at Birmingham, Tech. Rep., 2018.

[2] Musallam *et al.*, "Cognitive control signals for neural prosthetics," *Science*, vol. 305, no. 5681, pp. 258–262, 2004.

[3] Moritz, Perlmutter, and Fetz, "Direct control of paralysed muscles by cortical neurons," *Nature*, vol. 456, no. 7222, pp. 639–642, Dec 2008.

[4] Jackson *et al.*, "The Neurochip BCI: towards a neural prosthesis for upper limb function," *IEEE Trans Neural Syst Rehabil Eng*, vol. 14, no. 2, pp. 187–190, Jun 2006.

[5] Kellis *et al.*, "Decoding spoken words using local field potentials recorded from the cortical surface," *J Neural Eng*, vol. 7, no. 5, p. 056007, Oct 2010.

[6] Pandarinath *et al.*, "High performance communication by people with paralysis using an intracortical brain-computer interface," *Elife*, vol. 6, 02 2017.

[7] Aflalo *et al.*, "Decoding motor imagery from the posterior parietal cortex of a tetraplegic human," *Science*, vol. 348, no. 6237, pp. 906–910, 2015.

[8] Andersen *et al.*, "Toward more versatile and intuitive cortical brainmachine interfaces," *Current Biology*, vol. 24, no. 18, pp. R885 – R897, 2014.

[9] Andersen and Cui, "Intention, action planning, and decision making in parietal-frontal circuits," *Neuron*, vol. 63, no. 5, pp. 568–583, Sep 2009.

[10] Wu *et al.*, "Bayesian population decoding of motor cortical activity using a kalman filter," *Neural Computation*, vol. 18, no. 1, pp. 80–118, 2006.

[11] Sussillo *et al.*, "Making brain-machine interfaces robust to future neural variability," *Nature Communications*, 2017.

[12] Schwemmer *et al.*, "Meeting brain-computer interface user performance expectations using a deep neural network decoding framework," *Nat. Med.*, Sep 2018.

[13] Chollet *et al.*, "Keras," https://keras.io, 2015.

[14] Gers, Schmidhuber, and Cummins, "Learning to forget: Continual prediction with lstm," IDSIA, Tech. Rep., 1999.

[15] LeCun, Bengio, and Hinton, "Deep learning," *Nature*, vol. 521, pp. 436 EP –, May 2015.