VIRTUAL NEURAL SIGNAL SIMULATOR FOR ANALYZING THE PERFORMANCE OF DECODING ALGORITHMS IN BRAIN-MACHINE INTERFACES

Taylor Chung, Angela Zhang, Jonathan Kao

Department of Electrical and Computer Engineering, University of California — Los Angeles

Introduction

Brain-Machine Interface (BMI)

- Brain-machine interfaces (BMIs) decode neural activity into control signals for prosthetics and communication devices.
- Online-prosthesis simulators (OPS) generate artificial neural data based on user movement, instead of collecting data through implants.

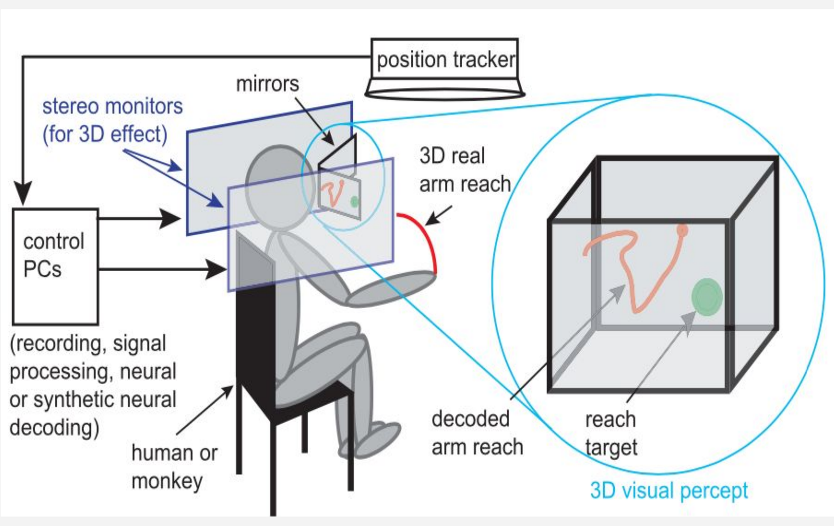

Figure 1. BMI Experiments. Schematic of a BMI experiment using real data.

Motivation

- BMI research using real neural data is difficult to facilitate due to its costs and risks.
- A well-developed neural data simulator can allow for more tests of algorithms and parameters.

Materials and Methods

Testing Interface : Getting Cursor Data

- User moves the cursor on the screen (Fig 6).
- Cursor velocity data is collected every 25 milliseconds.

Figure 2. Experimental Setup. Cursor movement data is collected then decoded into movements.

Generating Neural Signals.

- Each neuron has a minimum firing rate (λ_{min}), maximum firing rate (λ_{max}), and preferred direction (θ_{max}).
- Firing rate of a neuron at an instance is determined by the angle of cursor movement (θ).
- Spike trains for n neurons are generated by the Poisson process.

$$\lambda_t = \lambda_{min} + (\lambda_{max} - \lambda_{min}) \cdot \cos(\theta - \theta_{max})$$

$$y_t | \lambda_t \sim \text{Poisson}[\, h(\lambda_t)]$$

Figure 3. Equations used to generate spikes. Firing rates are calculated from the cosine tuning curve using mouse data, then turned into spike trains by a Poisson process.

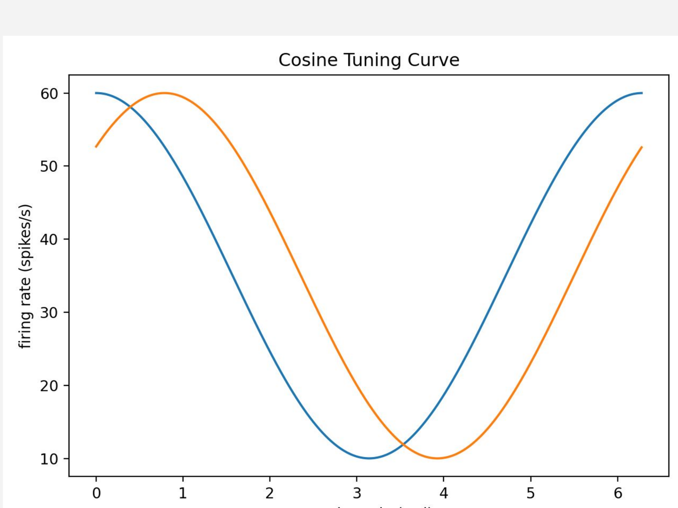

Figure 4. The cosine tuning curve

Figure 5. Logos of Python Libraries Used

Programming Tools Used

- Numpy and Matplotlib are used to compute and plot data.
- Pygame modules are used to get cursor velocity data.

Results

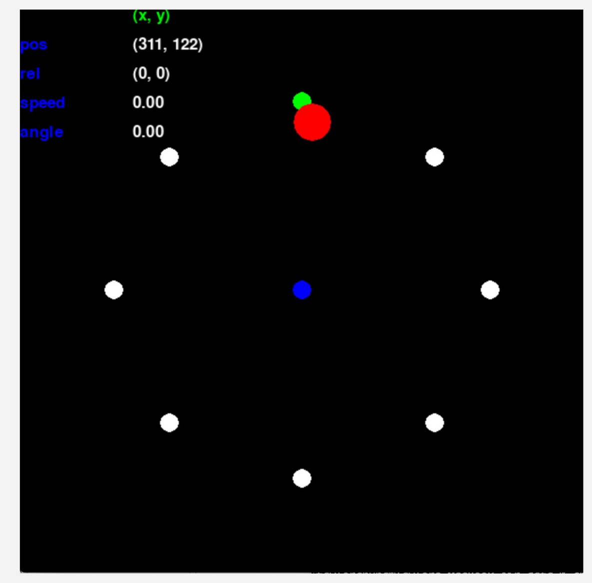

Figure 6. Testing interface built using Pygame. White dots around the center represent targets. Red dot follows the cursor as controlled by the user. Targets turn green when in contact with the cursor (red). Cursor position, speed, angle are shown in upper left corner and recorded into a file.

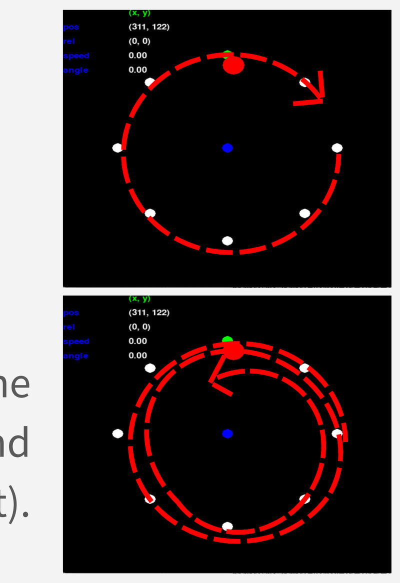

Figure 7. Mouse movements in testing corresponding to the plots (Fig 8). Single clockwise circular movement (up, left) and multiple counterclockwise circular movement (down, right).

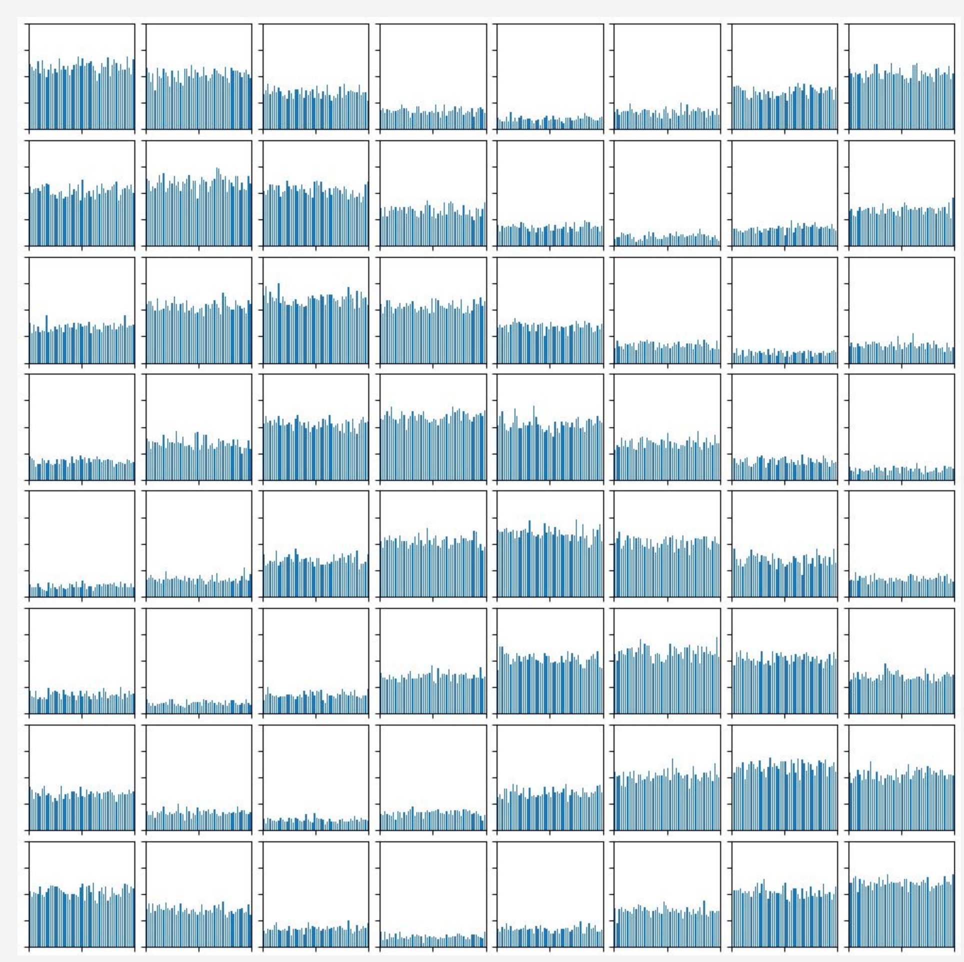

Figure 8. Simulator Generates Spike Trains by Poisson Process. Histograms of 8 neurons with preferred directions ranging from 0 to 2π at π/4 interval, firing at each direction. Each column of 8 histograms represents a single neuron firing at 0, π/4, π/2, 3π/4, π, 5π/4, 3π/2, 7π/4 radians. Spikes generated by the Poisson process using the firing rate calculated from the tuning curve (Fig 4). Spikes are averaged over 100 trials for each time bin in 1000ms period.

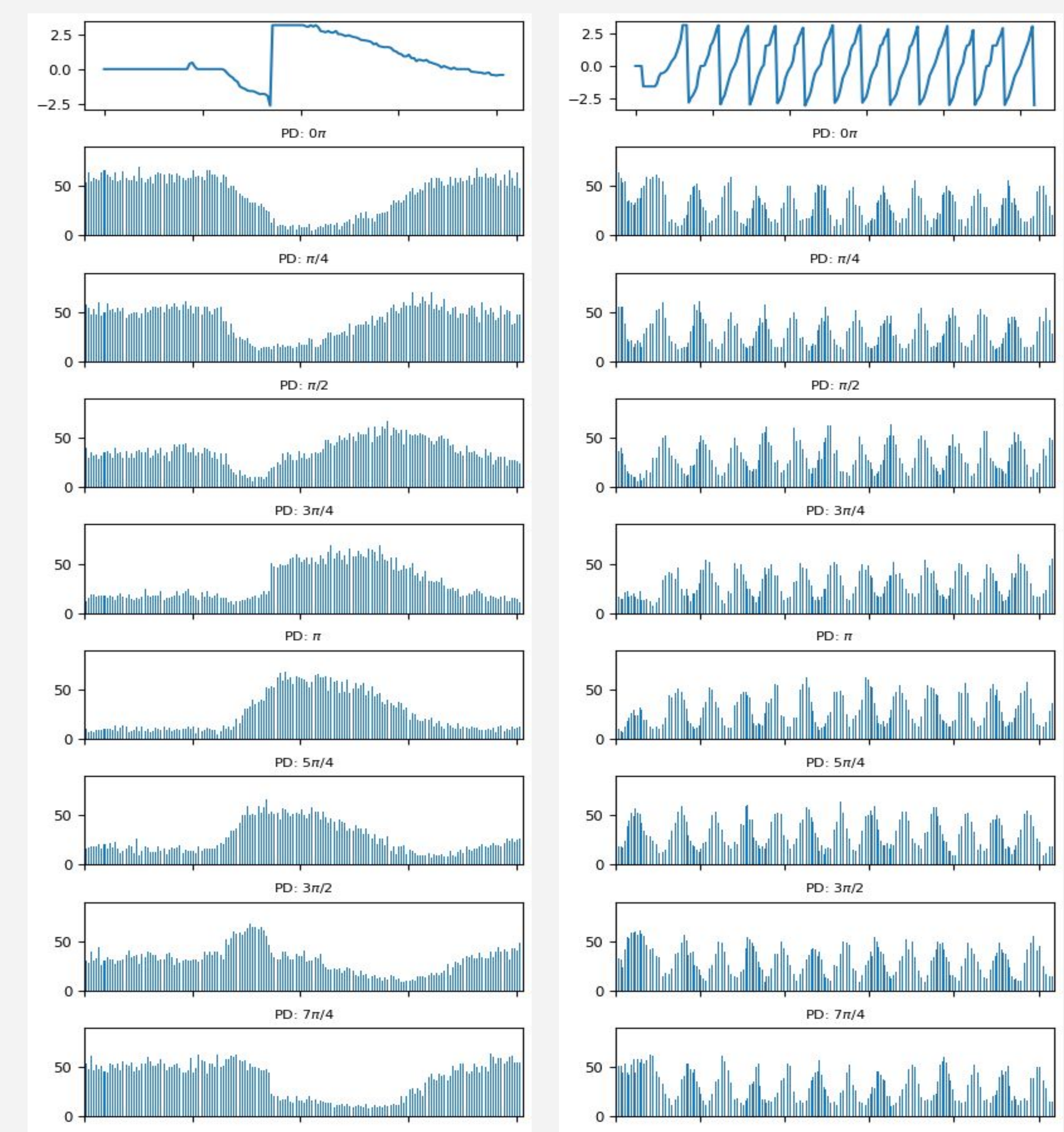

Figure 9. Simulator Generates Spikes According to the Cursor Movement. Histograms of 8 neurons with linearly spaced preferred directions firing according to cursor movement, when cursor is moved in a circle counterclockwise (left), and when cursor is moved in a circle repeatedly (right).

Conclusion

- Our findings support Cunningham's conclusion that a virtual neural signal simulator can be a viable testing option in developing BMI algorithms.
- Our fully-virtual online prosthesis simulator can significantly increase availability of BMI research, especially under current laboratory restrictions.

Future Directions

- Implement decoder algorithms to translate generated signals into cursor movements
- Test simulator with tasks performed by participants in previous studies with real neural data
- Compare simulated results to real neural data

References

Cunningham, John P., et al. "A Closed-Loop Human Simulator for Investigating the Role of Feedback Control in Brain-Machine Interfaces." Journal of Neurophysiology, vol. 105, no. 4, 2011, pp. 1932–1949., doi:10.1152/jn.00503.2010.

Acknowledgements

This research was funded by UCLA ECE's Fast Track Program Fellowship through UCLA's Summer Undergraduate Research Program and the Electrical and Computer Engineering Department. We thank Will Herrera for his guidance throughout this program.