



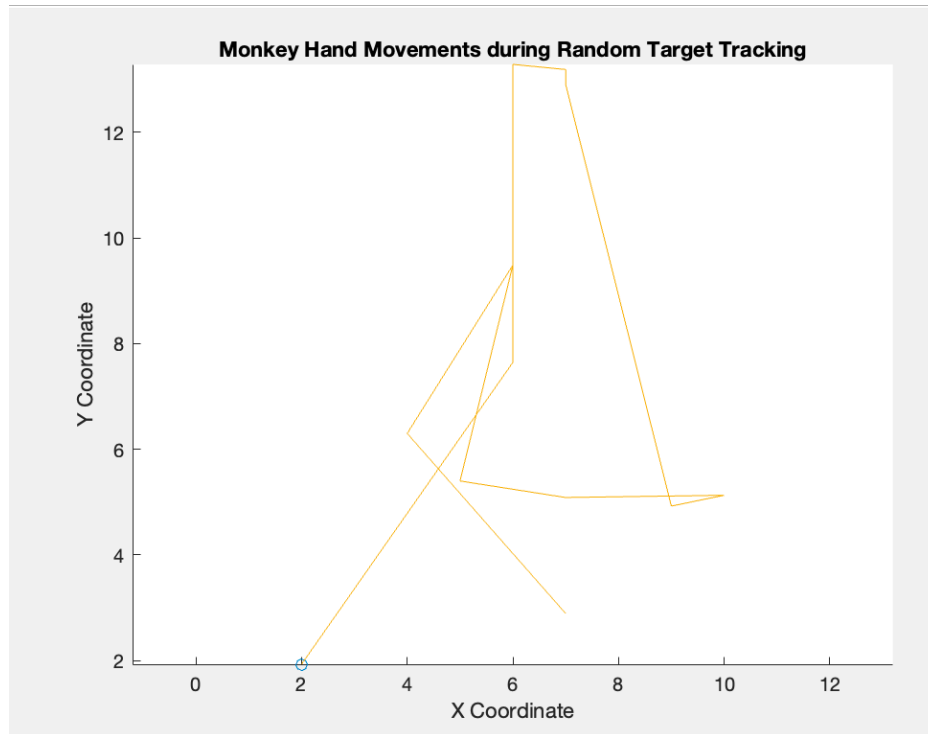
# *Brain Computer Interface*

*Neural Data Analysis in Matlab  
of BCI movement data using a  
linear-regression model fit for a  
decoder.*

*Singh, Jaspreet*

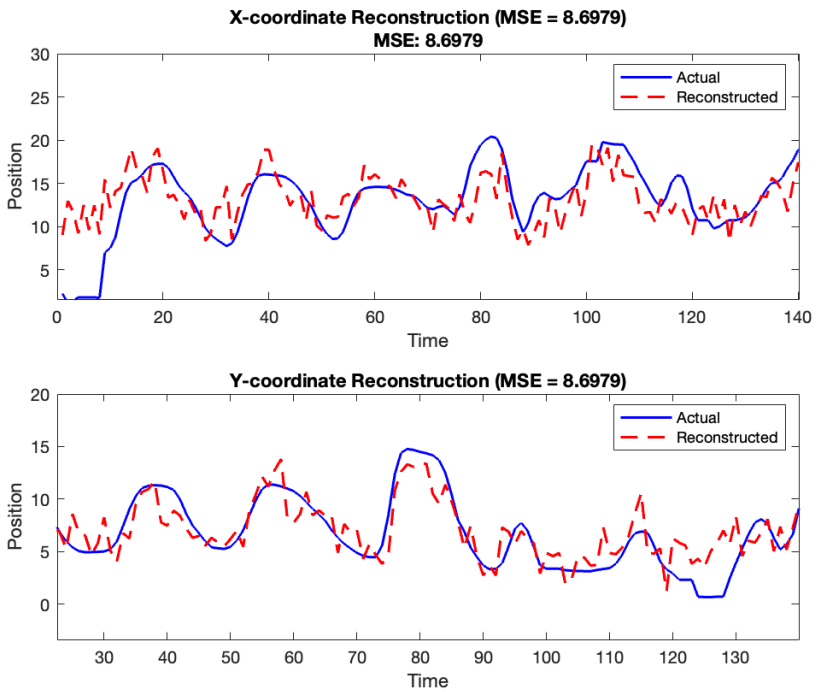
---

Step 1



**Figure 1.** This figure illustrates the trajectory of monkey hand movements during random target tracking. The comet plot provides a dynamic visualization of the hand movements over time. The x-coordinate of the hand movements is plotted on the horizontal axis, while the y-coordinate is plotted on the vertical axis. The trajectory of the hand movements is traced by a comet-like effect, allowing for easy visualization of the movement path. This helps us visualize the direct and path that the monkey moves its hands towards various targets at. The data was down sampled for better visualization. Though the comet function is dynamic, this static image shows how the monkey's hand started towards the center of the graph and then moved in various directions, tracing various paths towards the different targets.

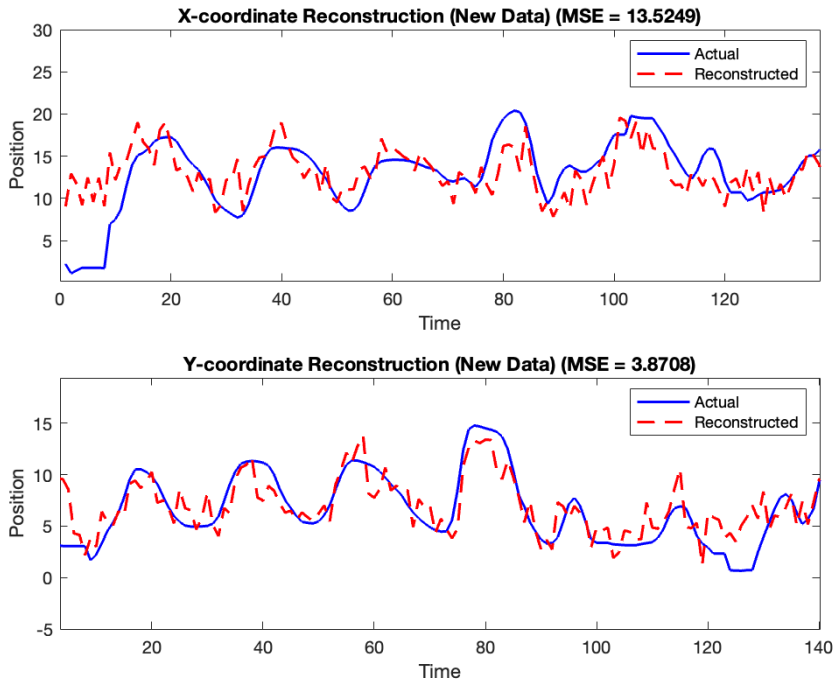
The decoder reconstructs the overall movement patterns or the "big picture" of the hand movements as it fits a model based on near regression to show the relationship between the neural activity (input) and the kinematic data (output) so while linear regression can capture general trends and movement patterns it does not capture minutia or intricate patterns in the data.



**Figure 2.** Figure 2 demonstrates the masque monkey's reconstructed hand movements (red dashed line) which were compared to compared to the actual hand movements (solid blue line). There is a figure demonstrating x-coordinate (top) and y-coordinate (bottom). The x-axis represents time, and the y-axis represents the position of the hand. The legend indicates the actual and reconstructed hand movements, with the mean squared error (MSE) providing a measure of the decoder's performance. Similarity of reconstructed data indicates that the decoder is accurately representing the data.

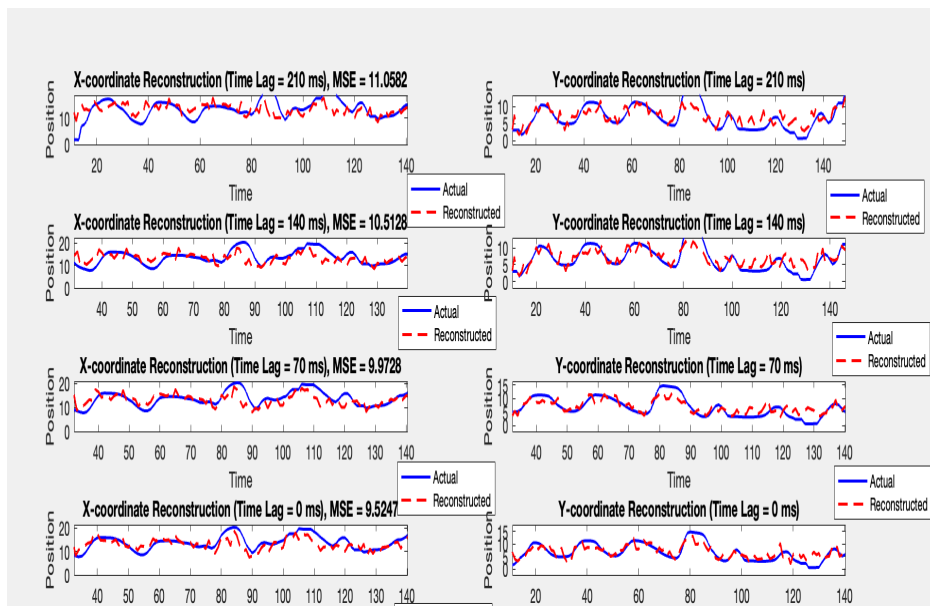
Commented [MOU]: Commented on similarity





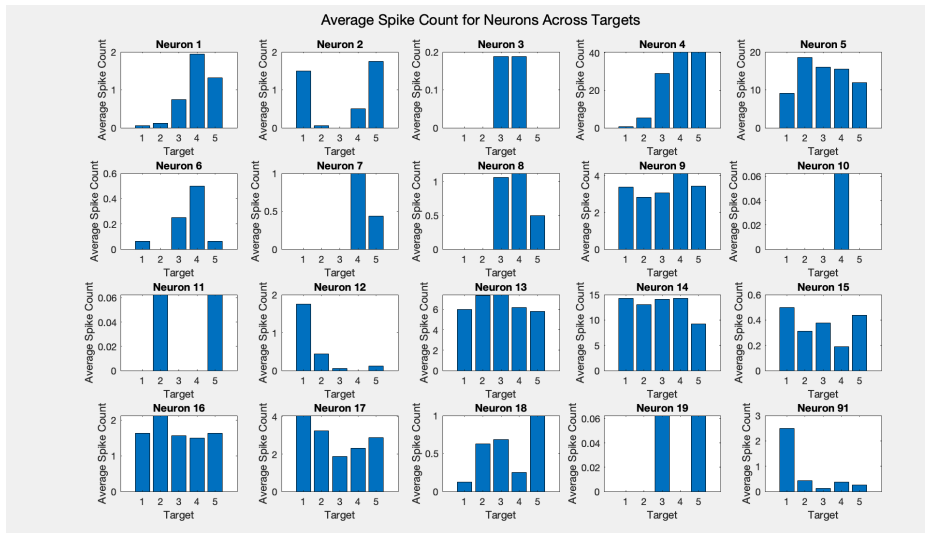
**Figure 3.** Figure 3 demonstrates the masque monkey's reconstructed hand movements (red dashed line) which were compared to the actual hand movements (solid blue line). There is a figure demonstrating x-coordinate (top) and y-coordinate (bottom). The x-axis represents time, which was measured over a 140 ms time period, and the y-axis represents the position of the hand. The legend indicates the actual and reconstructed hand movements, with the mean squared error (MSE) providing a measure of the decoder's performance. This data shows that the reconstructed data is similar to the actual data but not exactly the same. Similarity of reconstructed data indicates that the decoder is accurately representing the data.

Step 5



**Figure 4.** The figure above demonstrates shows a representation of hand movements with various time lags applied during decoding. A time lag of 0, 70, 140, and 210 milliseconds was applied and is shown for x and y coordinates (left and right sides). The actual hand movements (blue) and the reconstructed hand movements (red dashed lines) is shown in the figure above at each of the time lag values. Each subplot title indicates the time lag value and the corresponding mean squared error (MSE). MSE increases with increased lag, however, these visualizations allow for an exploration of how different time lags affect the accuracy of hand movement reconstruction. As the actual and reconstructed waveforms have a low MSE and resembled the actual hand movement, this time lag is suited for this dataset and can help overcome delays with neural processing in this computational stimulation.

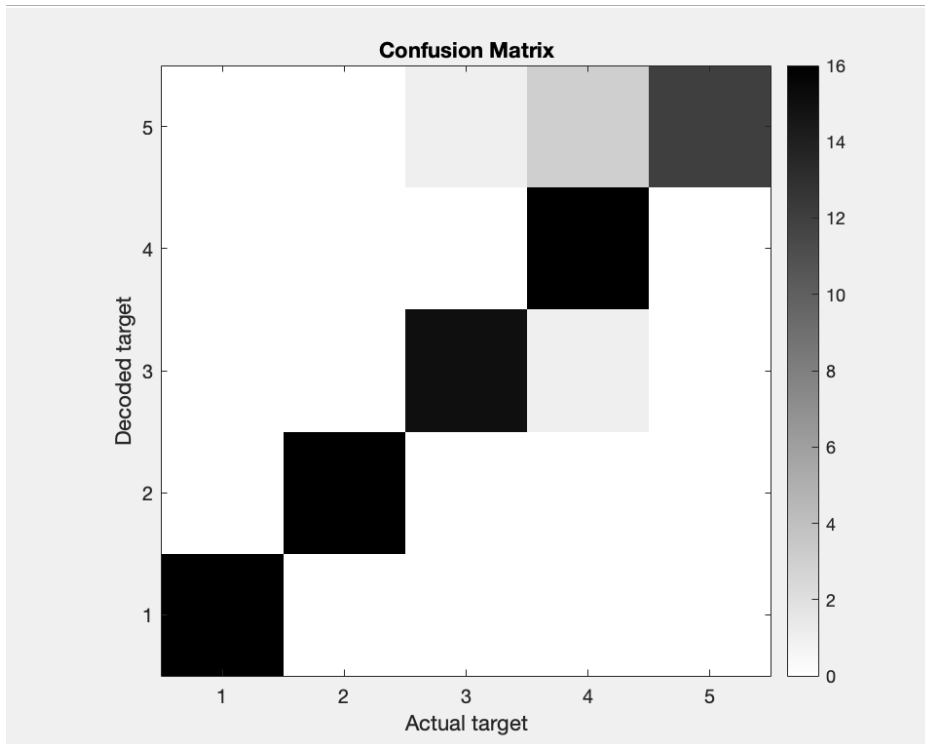
Part 2 – step 1



**Figure 6.** In the figure shown above, 20 different, randomly sampled neurons can be seen. The average spike count for neurons is shown to display tuning of the ells to understand the activity of individual neurons during different target trial. The 5 targets (x-axis) and the average spike count (y-axis) for each target can be seen. The average spike count axis changes for each neuron to reflect the increase in spike counts during certain trials for certain neurons. For instance, the average spike count for Neuron 2 is between 0-.2 but Neuron 4 has a range of 0-40 spike counts. From the randomly sampled neurons, it appears that there is activity at target 4 in 17 out of 20 neurons. This information is interesting at it may suggest neurons are in a certain area corresponding to activity reaching in target 4.



Part 2.- step 4



The figure above shows a confusion matrix which plots actual target (x-axis) versus decoded target (y axis). The diagonal of black squares represents where there is the most correlation between the two. Essentially, the confusion matrix shows what the actual target that the monkey moved its hand toward was plotted against where the decoder predicted that the monkey would move next. The plot shows the decoder performed reasonably well for target 1, 2, and 3. However, near target 4 we see that there is light gray above and below the diagonal. This means the decoder predicted target 5 and 3 at certain times, instead of the true target. The plot above shows that there appears to be a lot of activity (spike counting) at target 4. Perhaps this contributed to the inaccuracy of the confusion matrix. While further investigation is needed, the decoder works from data fed into the algorithm. If there is an abundance of data on target 4, reflected in increase spike counts, then it is possible that the decoder is more sensitive to predicting target 4.

Additionally, calculations were made to evaluate the performance of the decoder:

Performance Metrics:

Accuracy: 0.9375

Precision: 1      1   0.93682   0.78789      1

Recall (Sensitivity): 0.9375

F1-Score: 0.96498   0.96498   0.93484   0.85476   0.96498

The overall accuracy of the decoder is around 94%, while this is performing well, there are a number of ways that might assist in making the decoder more accurate. Considering the method of implementing the decoder to achieve this result, there are some alternations that could be made to increase accuracy such as: cross-validation to assess the performance and alter many hyperparameters, additionally we could use more data to fit the model which could provide better accuracy. With techniques available to us today, there are many possibilities like training a neural network rather than linear regression, you can use a cross-validation to optimize hyperparameters and use a neural network decoder on both the training and testing data. You can optimize this model to improve performance.