NEU 377 Research Summary

Sai Koukuntla

1 Project Goals

Last semester, I implemented an adaptive point process filter in Python and validated it using simulated hippocampus place cell spiking data. I also began to apply the filter to prefrontal neural spiking data collected during a center-out saccade task. My work this semester had two main goals:

1. Create an independent python package that implements the point process filter (PPF) equations
2. Analyzing saccade neuronal data using the filter to determine whether PPFs can decode ballistic movements.

I also planned to apply the filter to BMI cursor data, but I was unable to complete this analysis to due time constraints and issues with parsing and aligning the data.

2 Filter and Python Package

This section (briefly) summarizes the adaptive point process filter theory and equations, and describes the available filter functions. The Python package and documentation referenced in this section are available here.

2.1 Filter Theory

2.1.1 Point-Process Filter (PPF)

Adaptive point process filters have two main advantages over traditional Kalman filters: **1)** they are able to track the temporal evolution of neurons' receptive fields, and **2)** they operate on smaller timescales than traditional filters, so they are able to decode neuronal data at finer temporal resolutions. The system equation for the filter is a set of parameter vectors with linear evolution processes and gaussian error,

$$\boldsymbol{\theta}_{t+1} = \mathbf{F}_t \boldsymbol{\theta}_t + \eta_t \tag{1}$$

where $\mathbf{F}_t$ is a system evolution matrix and η_t is a zero-mean white noise process with covariance $\mathbf{Q}$. At each time step, the parameter vector is used to approximate the probability of observing a spike,

$$\Pr(\Delta N_t \text{ spikes in } (t-1, t\,] \mid \boldsymbol{\theta_t}) = \exp(\Delta N_t \log(\lambda(t \mid \boldsymbol{\theta_t})\Delta t) - \lambda(t \mid \boldsymbol{\theta_t})\Delta t) \tag{2}$$

where $\lambda(t \mid \boldsymbol{\theta_t})$ is a neuron's firing rate equation (e.g. a cosine tuning curve) and ΔN_t is the number of spikes that occur in each time bin. We choose an appropriate resolution such that each bin contains at most one spike. As shown in [1], the adaptive point process filter equations follow from Equations 1 and 2. The first two equations predict the parameter mean ($\boldsymbol{\theta}_{t|t-1}$) and variance ($\mathbf{W}_{t|t-1}$) according to Equation 1,

$$\boldsymbol{\theta}_{t|t-1} = \mathbf{F}_t \boldsymbol{\theta}_{t-1|t-1} \tag{3}$$

$$\mathbf{W}_{t|t-1} = \mathbf{F}_t \mathbf{W}_{t-1|t-1} \mathbf{F}_t' \tag{4}$$

and the second two equations estimate the posterior mean and variance using spiking observations to estimate the posterior variance and mean,

$$\left(\mathbf{W}_{t|t}\right)^{-1} = \left(\mathbf{W}_{t|t-1}\right)^{-1} + \left[\left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta_t}}\right)' [\lambda \Delta t] \left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta_t}}\right) - (\Delta N_t - \lambda \Delta t) \frac{\partial^2 \log \lambda}{\partial \boldsymbol{\theta_t} \partial \boldsymbol{\theta_t'}}\right]_{\boldsymbol{\theta}_{k|k-1}} \tag{5}$$

$$\boldsymbol{\theta}_{k|k} = \boldsymbol{\theta}_{k|k-1} + \mathbf{W}_{t|t} \left[\left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta_t}}\right)' (\Delta N_t - \lambda \Delta t)\right]_{\boldsymbol{\theta}_{k|k-1}.} \tag{6}$$

If we want to quantities that the neurons are tuned to (i.e. kinematics), we can simply track them as part of the parameter vector. In this case, however, the filter may diverge from the ground truth if our initial estimates ($\boldsymbol{\theta}_0$) of the receptive field parameters are not accurate.

2.1.2 Optimal Feedback Control PPF (OFC-PPF)

To solve the divergence issue described above, Shanechi et. al. [2] propose decoupling the estimation of kinematics and receptive field parameters using an optimal feedback control (OFC) model. In this framework, we assume that the monkey issues control commands based on the target location and visual feedback of the current position. We use these two quantities to estimate the monkey's intended kinematics (using one PPF), and then use this estimate to update the receptive field parameters of each neuron (using a second PPF). The kinematic PPF is similar to the PPF discussed in the previous section. It is characterized by a linear dynamical model,

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} + \mathbf{w}_{t-1} \tag{7}$$

where $\mathbf{x}_t$ is the kinematic state (position and velocity), $\mathbf{u}_t$ is the control command, and $\mathbf{w}$ is a zero mean white noise with covariance $\mathbf{W}$. The matrices $\mathbf{A}$ and $\mathbf{B}$ are selected by fitting them to the subject's movements. To estimate the brain's control commands given a current position and a target position, we define a cost function, which the brain tries to minimize:

$$J = \sum_{t=1}^{\infty} w_x \|\mathbf{x}_t - \mathbf{x}^*\|^2 + w_r \|\mathbf{u}_t\|^2, \tag{8}$$

where $\mathbf{x}^*$ denotes the target position, and the weights w_r and w_v are approximated such that the model produces trajectories similar to the subject's natural movements. From the cost equation, OFC-PPF solves for the intended control command as

$$\mathbf{u}_t = -\mathbf{L}(\mathbf{x}_{t|t} - \mathbf{x}^*), \tag{9}$$

where $\mathbf{L}$ is the solution to the discrete form of the algebraic Riccati equation. From this, we obtain equations for the OFC-PPF which are analogous in form to Equations 3 - 7.

2.1.3 OFC Parameter Fitting

As mentioned above, the OFC parameters are fit such that the model's predictions produce similar trajectories to the monkey's natural movements. The parameters $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{W}$ are chosen as follows,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{bmatrix} \tag{10}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{11}$$

$$\mathbf{W} = \text{diag}\begin{bmatrix} 0 & 0 & w & w \end{bmatrix} \tag{12}$$

where a and w are fit to the monkey's movement kinematics using maximum-likelihood estimation [3]. The structure of $\mathbf{A}$ encodes a velocity decoder; the position is obtained by integrating the decoded velocity in each direction. The cost-function constants w_x and w_r are chosen as $w_x = 1.5\tau^2$, $w_r = \tau^4$, where τ is a time constant that describes the speed of the subject's movements.

2.2 Implementation and Testing

To implement the PPF and OFC-PPF models, I created a pip-installable Python package that houses all the necessary functions. For each of the filters (PPF and OFC-PPF), the package contains a function that applies the filter to input data. Both filter functions require a firing rate equation in the form of a SymPy (symbolic algebra package) expression, which allows the decoder to be applied to neurons with a variety of tuning curves. The package also includes functions to fit the OFC parameters given input data. More detailed documentation and usage examples can be found in the GitHub repo referenced at the beginning of this section.

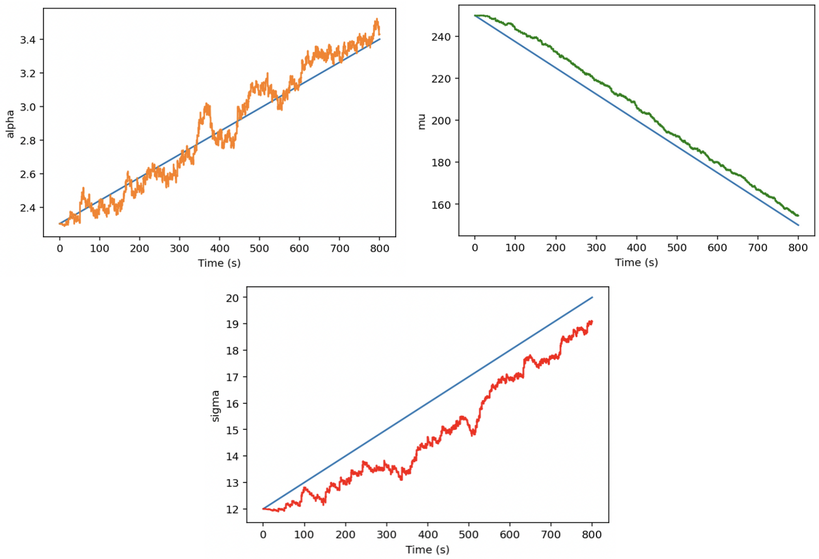

Figure 1: Sample output from symbolic PPF validation. Firing rate equation was $\lambda(t) = \exp\left(\alpha - \frac{(x-\mu)^2}{2\sigma^2}\right)$. Top left: α, top right: μ, bottom: σ

To validate the regular PPF, I used simulated hippocampal place cell data and verified that the output matched my prewritten (non-symbolic) analysis. Figure 1 above shows sample output of this testing. To validate the OFC parameter fitting and OFC-PPF decoder, I applied the OFC-PPF function to prefrontal neuronal data recorded during a saccade task [4] and verified that the results matched a non-symbolic version of the analysis. The results of this testing are shown in the next section (Saccade Task Analysis). Since the hippocampal simulations and saccade data use different firing rate expressions, these tests were sufficient to confirm that the symbolic version of the filter was operating correctly.

3 Saccade Task Analysis

3.1 Methods

Last semester, I performed a preliminary analysis of the saccade neuronal data using the regular PPF with a Gaussian firing rate equation. However, since I did not have a good estimate for the initial values of the receptive field parameters, the decoder diverged from the ground truth very quickly, and the output had essentially no correlation with the ground truth. Once I had finished implementing the OFC-PPF this semester, I decided to see if it could improve the decoding of the saccade data on a session of the saccade task that contain spike trains from 29 prefrontal neurons. Since cosine tuning curves are more common for motor neurons, I chose the following as the firing rate equation for each neuron c:

$$\lambda_c\left(t|\boldsymbol{\theta}_t\right) = \exp(\beta_t^c + \boldsymbol{\alpha}_t^{c\prime}\mathbf{v}_t) \tag{13}$$

where $\mathbf{v}_t$ is the 2D velocity and $\boldsymbol{\alpha}_t^c$ is a 2D parameter vector applied to the velocity. I initialized the $\boldsymbol{\alpha}_t^c$ for all neurons to 0, so as to not bias their direction preferences. To initialize β_t^c, I calculated the background firing rate for each neuron by summing together the non-saccade portions of the signal (i.e. before the cue was presented or after the saccade was complete). To select the value of τ for the command cost-function parameters w_x and w_r, I visualized several saccade trials to estimate the typical duration of movement (example shown in Figure 2 below). The saccade appeared to take place within a period of 20 ms for the vast majority of trials, so I chose this value for τ.

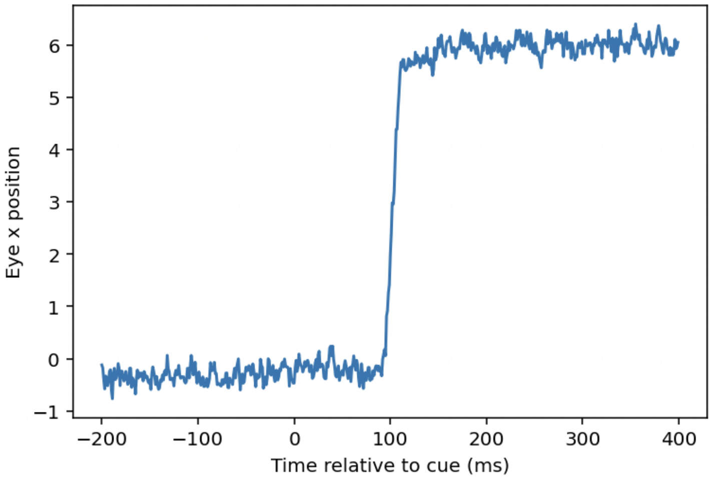

Figure 2: Example of saccade used to select τ. The saccade lasts for about 20 ms.

I then used the functions described above to fit and validate the OFC parameters **A**, **B**, and **W**, and used the symbolic OFC-PPF function to obtain decoding results for each trial in the session of data.

3.2 Results

Figure 3 below shows a comparison between natural saccade kinematics (top) and the trajectory produced by the OFC model that was fit to the saccade data (bottom). The two trajectories look similar in shape and amplitude; the OFC trajectory resembles a noiseless version of the actual trajectory.

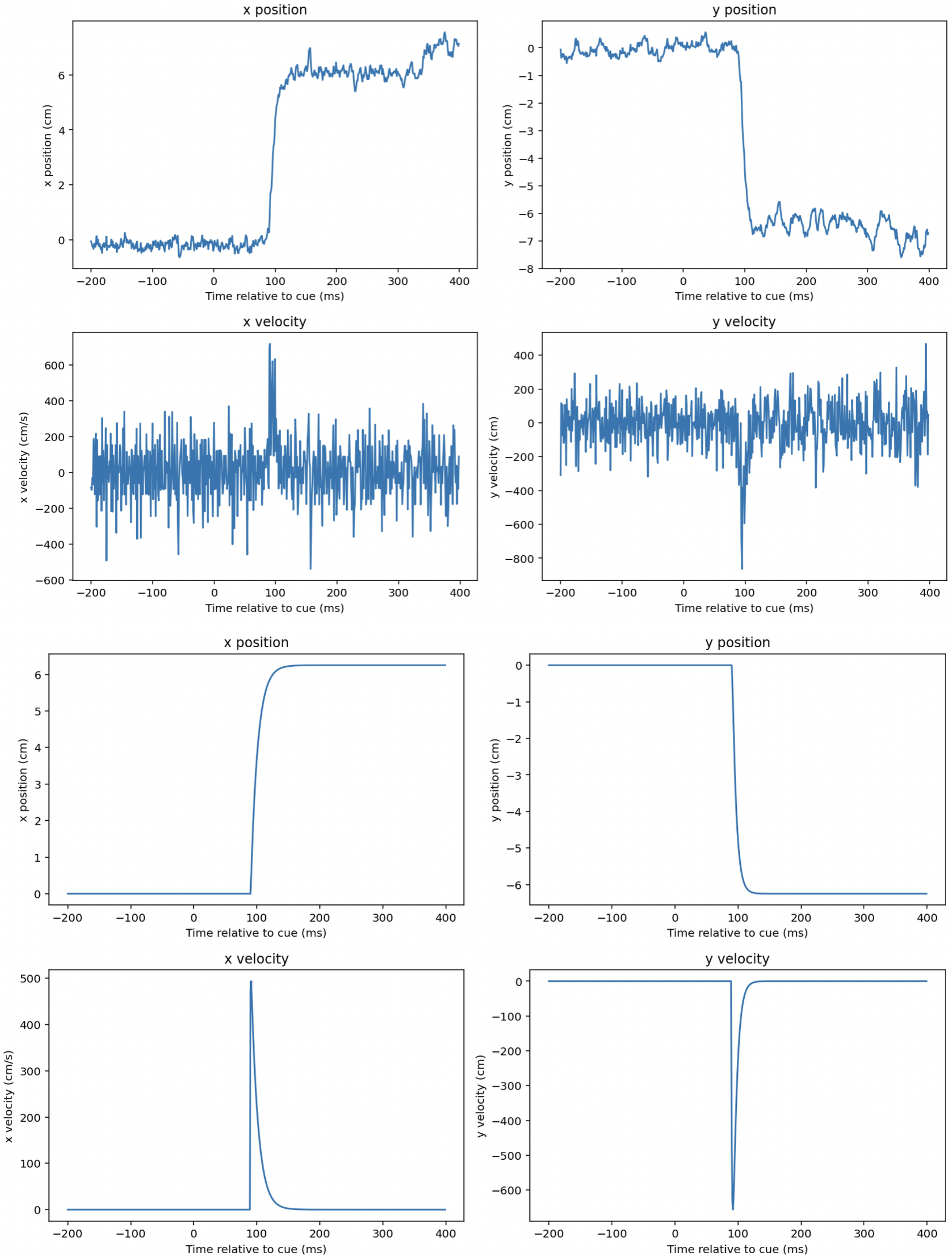

Figure 3: **Top**: Natural saccade kinematics (2D velocity and position). **Bottom:** Saccade trajectory generated by OFC model.

Figure 4 below shows the decoder output for a single saccade trial to a bottom right target. The position trajectory is in the correct direction for both dimensions, but the amplitude and slope of the trajectory are much weaker than the actual saccade. The decoded velocity looks like a series of spikes rather than a single, large magnitude spike.

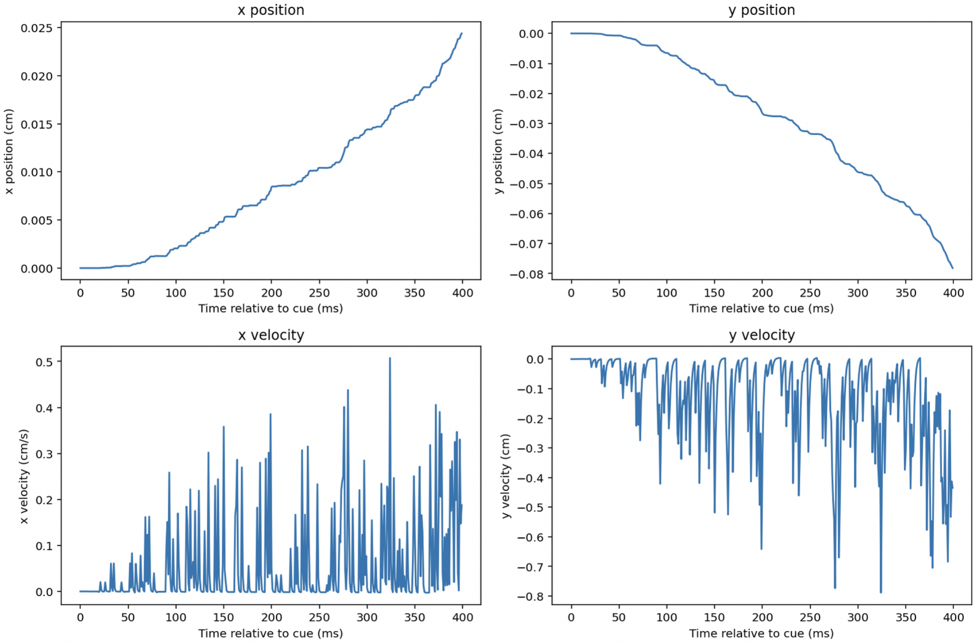

Figure 4: Decoder output for a single saccade trial (target was bottom-right of the origin).

Figure 5 below shows the average decoded position trajectories for saccades to each of the four targets. The average trajectories are in the correct direction and follow the same general shape as the single trial. However, the amplitudes of the average trajectories are even smaller than the single-trial trajectory in Figure 4, which suggests that some of the trials may have been decoded with smaller amplitudes.

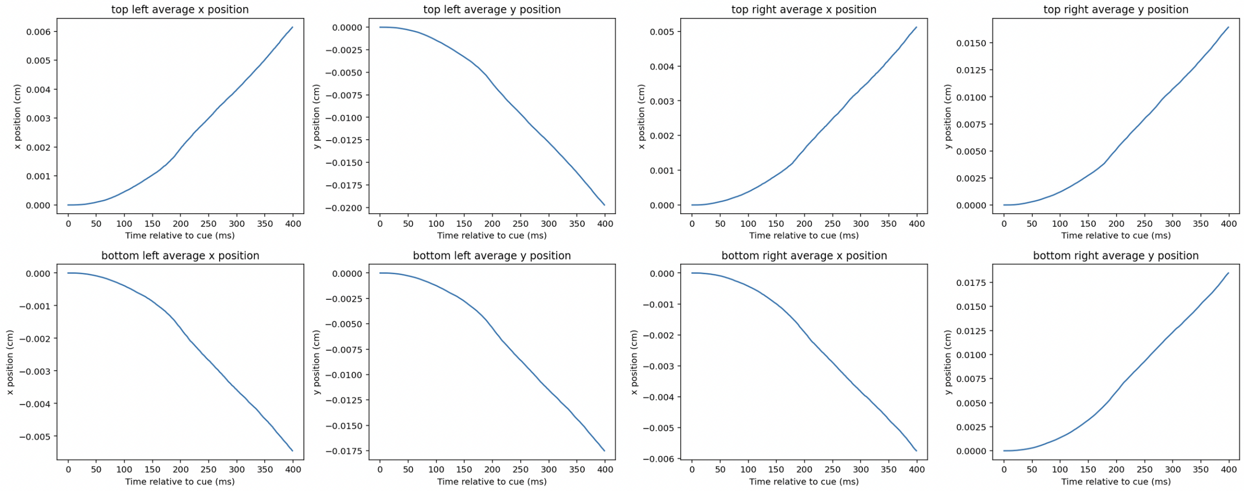

Figure 5: Average decoded trajectories in each of the four saccade directions.

3.3 Conclusions

The degree of similarity between the OFC trajectory and the actual trajectory suggests that the OFC model is capable of modeling saccade kinematics. However, the filter does not reliably decode saccade kinematics, despite correctly determining the direction of the saccade in most cases. Additionally, since the decoder trajectory amplitudes are so close to zero (1000 times smaller than the actual amplitude), the direction predictions may simply be an artifact of using the OFC model to guide the evolution of the receptive field parameters. The OFC-PPF also has a majo drawback: although the OFC-PPF can be successful at decoding movements with a defined target within the structure of a controlled experiment, it would not be suitable for effectively decoding a subject's free movement, since it would be more difficult to obtain information about the intended control commands.

If the PPF is actually correctly decoding the direction of the saccade, this suggests that the prefrontal neurons that were recorded from may encode some task or saccade-direction related information. However, the mismatch of shape between the actual trajectory and the decoded trajectory suggests that any directional tuning of neurons is very weak. In any case, the prefrontal neurons do not encode saccade kinematics in a way that is accurately represented using a cosine tuning curve model. Since the prefrontal regions are often involved in planning, it is possible that these neurons encode high-level representations of the saccade (i.e. direction and amplitude) and the representations are transformed downstream into signals that control ocular muscles.

Ultimately, we were unable to use a PPF reliably decode saccade kinematics from prefrontal spiking data. However, this does not mean that PPFs are not suitable for decoding ballistic movements; further work using data from neurons that are more strongly tuned to ocular movements could help determine whether PPFs are able to decode ballistic movements that occur on timescales too short for traditional decoder to detect.

References

1. Eden, U. T., Frank, L. M., Barbieri, R., Solo, V. & Brown, E. N. Dynamic Analysis of Neural Encoding by Point Process Adaptive Filtering. en. *Neural Computation* **16,** 971–998. ISSN: 0899-7667, 1530-888X. `https://direct.mit.edu/neco/article/16/5/971-998/6831` (2022) (May 2004).

2. Shanechi, M. M., Orsborn, A. L. & Carmena, J. M. Robust Brain-Machine Interface Design Using Optimal Feedback Control Modeling and Adaptive Point Process Filtering. en. *PLOS Computational Biology* **12** (ed Sporns, O.) e1004730. ISSN: 1553-7358. `https://dx.plos.org/10.1371/journal.pcbi.1004730` (2022) (Apr. 2016).

3. Orsborn, A. L., Dangi, S., Moorman, H. G. & Carmena, J. M. Closed-Loop Decoder Adaptation on Intermediate Time-Scales Facilitates Rapid BMI Performance Improvements Independent of Decoder Initialization Conditions. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **20,** 468–477. ISSN: 1534-4320, 1558-0210. `https://ieeexplore.ieee.org/document/6144747/` (2022) (July 2012).

4. Hunt, L. T. *et al.* Triple dissociation of attention and decision computations across prefrontal cortex. en. *Nature Neuroscience* **21,** 1471–1481. ISSN: 1097-6256, 1546-1726. `https://www.nature.com/articles/s41593-018-0239-5` (2022) (Oct. 2018).