Modeling the Oculomotor Integrator: Deterministic, Stochastic, and Connectome-Based Approaches

Yuval Rubin

Supervisor: Prof. Yoram Burak

September 2025

1 Introduction

Visual input is one of the most important sensory channels through which living creatures perceive the world. Information enters through the eyes and is then transferred to the brain for processing. Acting as the consumer of visual information, the brain determines where to focus, and the eye muscles adjust accordingly to provide the necessary input.

From a biological perspective, eye movement is controlled by two main sets of muscles: one governing horizontal position and the other vertical position. In this project, we focus on the horizontal system.

The hindbrain sends commands to a biological neural network that governs the horizontal muscles set. These commands are expressed as changes – for example, 5° rightward movement or 7° shift to the left. Such rapid movements are called **saccades**. However, to perceive an image, the eyes must rest on the object observed for a short while. This period of rest is called **fixation**. Since commands only arrive during saccades, the neural network that controls the muscles must "remember" the current eye position during fixations, as the muscles need continuous input that indicates how much to contract. This memory function is provided by the **oculomotor integrator** (OI), which is the focus of this project.

Here, we review the oculomotor integrator of the goldfish by examining three models developed over the last 20 years. We first consider Goldman's [1] approach, then expand it with a stochastic formulation proposed by NBS [2], and finally interpret recent findings acquired by state-of-the-art biological methods [3]. The goal of this project is to introduce a stochastic perspective to the different models, simulate them, and analyze the results.

1.1 Biological Neural Networks

To provide background for mathematical modeling of the oculomotor integrator, we first outline the basics of biological neural networks.

1.1.1 The Neuron

A **neuron** is a nerve cell that is the fundamental unit of the nervous system. Each neuron collects inputs, computes an internal result, and transmits outputs to other neurons. Communication occurs at **synapses**, which are biological junctions where signals are passed. Input reaches the neuron through the dendrite and can be received from multiple neurons simultaneously. The neuron executes the computation within its body and then it transmits the output along the axon and to the synapse. These signals are carried by chemical compounds called neurotransmitters that have different effects on different neurons.

Neurotransmitters can be either excitatory – increasing the likelihood of the receiving neuron to send signals, or inhibitory – reducing that likelihood. In addition, sensitivity to different neurotransmitters varies between neurons, as some are more sensitive to specific neurotransmitters, making them more responsive to those signals.

When receiving input, a neuron integrates signals across space (multiple synapses) and time (multiple arrivals). This integration is mostly done by electrical currents. If the integration crossed some threshold, the neuron produces an **action potential** – a rapid electrical spike that triggers neurotransmitter release as output. This event is also referred to as spiking, or firing.

Since this event is discrete, we often describe neurons by their **firing rate.** That is, the number of action potentials it generates per time unit. For example, the neuron's output is considered "large" if the frequency of action potentials is high. We measure firing rate in units of $\frac{\text{number of spikes}}{\text{second}}$.

When modeling mathematically, we refer to the neurotransmitters and their effect on the post-synaptic neuron (the one that receives the input) as **weights**. The neuron's integration is then modeled as a weighted sum of its inputs. Those weights incorporate various structural aspects of the connection between the two neurons, such as the effect of the neurotransmitters (excitatory or inhibitory – reflected by sign), how many synapses the neurons form between each other, how many neurotransmitters are released when a presynaptic spike occurs and other aspects.

1.1.2 A Neural Feedforward Network

If we consider this network on discrete time, then the most elementary model of a feedforward biological neural network is given by:

$$r(t + \Delta t) = \sum_{i=1}^{n} W_i r_i(t)$$

where $r(t)$ is the firing rate of a neuron, $r_i(t)$ is the firing rate of the presynaptic i'th neuron at time t and W_i is the weight of the input of neuron i on the receiving neuron.

Once modeling the integration process done by a single neuron, we can account for the action potential decision. As mentioned earlier, a neuron would fire an action potential if the integration crossed a threshold. Therefore, denoting:

$$\phi(x) = \max\{x, 0\} = [x]_+$$

for the threshold 0, we can consider the following formula:

$$r(t + \Delta t) = \phi\left(\sum_{i=1}^{n} W_i r_i(t)\right)$$

However, a biological neural network does not, in fact, operate in discrete time – as the input, integration and output are continuous. Hence, we model the firing rate as governed by the following differential equation:

$$\tau \frac{dr(t)}{dt} = -r(t) + \phi\left(\sum_{i=1}^{n} W_i r_i(t)\right)$$

where τ is a characteristic time constant, which determines the persistency of the dynamic – meaning, how quickly the neuron's firing rate changes. A small τ means the rate reacts quickly to inputs, and large τ makes the response slower and smoother, so the neuron integrates its inputs over a longer period. In the absence of input to the neuron, the equation is solved by a decaying exponential, as $r(t) = r(0)e^{-\frac{t}{\tau}}$, showing how τ plays a time scale role.

1.1.3 Recurrent Neural Network

We can now extend this model to a recurrent neural network, with bidirectional connections between neuron n_i and neuron n_j. We would describe such network using the following dynamic:

$$\tau \frac{dr_i(t)}{dt} = -r_i(t) + \phi\left(\sum_{j=1}^{n} W_{i,j} r_j(t)\right)$$

where the matrix W encodes the connection strength between neurons: the entry $W_{i,j}$ denotes the weight of the output sent by the presynaptic neuron n_j received as input by postsynaptic neuron n_i.

1.2 Attractor Networks

In the path to model the OI, we also visit attractor networks. Since the OI maintains fixations, it functions as a **line attractor**, a specific type of dynamical system. An **attractor** is a set of states towards which a dynamical system

evolves. A line attractor is a special case in which the attractor forms a continuous one-dimensional manifold of steady states. Since horizontal eye position is a one-dimensional variable and fixation supports a continuum of eye positions, the OI exhibits properties of a line attractor.

We analyze the recurrent rate model to identify line attractors. Although a non-linear model is more biologically accurate, it cannot be solved analytically (given the W matrix is known). Hence, we begin with the linearized system:

$$\tau \frac{dr_i(t)}{dt} = -r_i(t) + \sum_{j=1}^{n} W_{i,j} r_j(t)$$

we can also represent the same equation in matrix form:

$$\tau \frac{d\mathrm{r}(t)}{dt} = -\mathrm{r}(t) + W\mathrm{r}(t) = (W - I_n)\,\mathrm{r}(t)$$

Before analyzing, we assume symmetry in the mutual connection weights between neurons, meaning neuron n_i and neuron n_j have the same effect on each other. From this assumption, it follows that $W_{i,j} = W_{j,i} \rightarrow W = W^T$, and thus W is diagonalizable by the spectral theorem.

Let $P, D \in \mathbb{R}^{n\times n}$ be such that $W = P^{-1}DP$ and $D = \mathrm{diag}\,(\lambda_1, \dots \lambda_n)$. Now, the analytical solution is given by:

$$\begin{aligned}
\mathrm{r}(t) &= \exp\left(\tfrac{t}{\tau}\,(W - I_n)\right)\mathrm{r}(0) \\
&= P^{-1}\exp\left(\tfrac{t}{\tau}\,(D - I_n)\right)P\mathrm{r}(0) \\
&= P^{-1}\,\mathrm{diag}\Big(e^{t(\lambda_1-1)}, \dots, e^{t(\lambda_n-1)}\Big)\,P\mathrm{r}(0).
\end{aligned}$$

If $\Re\,(\lambda_i - 1) < 0$ for all $i \in [n]$, then the dynamic decays to 0 for all $\mathrm{r}(0)$. If there exists $i \in [n]$ such that $\lambda_i - 1 = 0$, then denoting v a corresponding eigenvector of λ_i, we get for $\mathrm{r}(0) = v$:

$$\mathrm{r}(t) = P^{-1}\mathrm{diag}\left(e^{t(\lambda_1-1)}, \dots,\ e^{t(\lambda_n-1)}\right)Pv = e^{t(\lambda_i-1)}v = e^0 v = v = \mathrm{r}(0)$$

Hence, we get $\mathrm{r}(t)$ is fixed. Since $\mathrm{span}(v)$ is a 1-dimensional vector space, we indeed get a continuous line of steady states. If, in addition, all other eigenvalues hold that $\Re\,(\lambda_j - 1) < 0$, then the dynamic is decaying in all orthogonal directions. Thus, we expect the OI to have one eigenvalue λ of W such that $\lambda = 1$ and all other eigenvalues hold $\Re\,(\lambda_i - 1) < 0$.

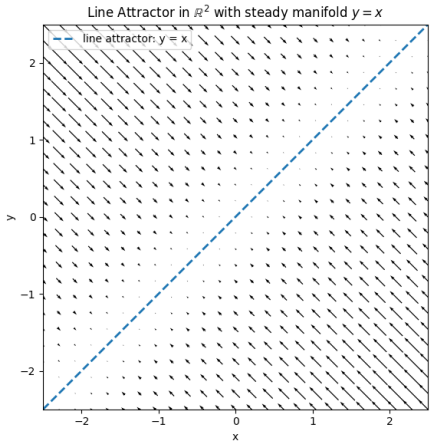

Figure 1: Line Attractor Phase Diagram – Example

Further explanation: The system corresponds to $W = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, with eigenvectors $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ (steady, $\lambda = 0$) and $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ (decaying in orthogonal direction, $\lambda = -2$). This structure defines a one-dimensional manifold of fixed points, which is the span of $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

2 The Oculomotor Integrator

2.1 The 2007 Model

We can now present the model offered by Goldman [1] for the OI. In this model, the OI is constructed by two symmetrical populations of neurons (right and left). Within-population connections are excitatory and inter-population connections are inhibitory.

2.1.1 Rate Models

Therefore, the firing rate of a right population neuron is governed by:

$$\tau_r \frac{dr_i^R(t)}{dt} = -r_i^R(t) + \sum_{j=1}^{n} W_{i,j}^R s_j^R(t) - \sum_{j=1}^{n} W_{i,j}^L s_j^L(t) + T_i(t) + B_i(t)$$

$$\tau_s \frac{ds_j^R}{dt} = -s_j^R + s_\infty \left(r_j^R\right)$$

where $B_i(t)$ is saccade-related input to the OI, $T_i(t)$ is a steady background signal coming from the balance organs of the inner ear, s_j is the synaptic output

of a neuron and $s_\infty(\cdot)$ is an activation function. Also, τ_r and τ_s are time constants. The same structure holds for the left population.

Because the OI output is one-dimensional, the authors [1] assumed that the connectivity matrix W holds $\text{Rank}(W) = 1$. Therefore, $W = \xi\eta^T$ for some $\xi, \eta \in \mathbb{R}^n$. The authors [1] assumed ξ to represent the sensitivity of each neuron to its inputs (both recurrent and saccadic), and η to denote the contribution of each neuron to synaptic output. Hence, the recurrent output is (for the right population):

$$S_R \equiv \sum_{i=1}^{n} \eta_j s_j^R$$

Thus, the former equations can be written as:

$$\begin{aligned}
\tau_r \frac{dr_i^R(t)}{dt} &= -r_i^R(t) + \sum_{j=1}^{n} W_{i,j}^R s_j^R(t) - \sum_{j=1}^{n} W_{i,j}^L s_j^L(t) + T_i(t) + B_i(t) && (1)\\
&= -r_i^R(t) + \sum_{j=1}^{n} \xi_i \eta_j s_j^R(t) - \sum_{j=1}^{n} \xi_i \eta_j s_j^L(t) + T_i(t) + B_i(t) && (2)\\
&= -r_i^R(t) + \xi_i \sum_{j=1}^{n} \eta_j s_j^R(t) - \xi_i \sum_{j=1}^{n} \eta_j s_j^L(t) + T_i(t) + B_i(t) && (3)\\
&= -r_i^R(t) + \xi_i S_R - \xi_i S_L + T_i(t) + B_i(t) && (4)\\
&= -r_i^R(t) + \xi_i \big(S_R - S_L\big) + T_i(t) + B_i(t). && (5)
\end{aligned}$$

and:

$$\tau_s \frac{dS_R(t)}{dt} = -S_R + \sum_{i=1}^{n} \eta_i s_\infty \left(r_i^R\right)$$

and by symmetry, the same equations hold for the left population. Goldman [1] further explored alternative activation mechanisms to account for the observation that silencing one population (i.e. left) affects only fixations on the corresponding side (i.e. eye position fixated to the left). The authors [1] presented the following mechanisms:

- Traditional model:
$$s_\infty(r) = \frac{r}{r + 20\,\text{Hz}}$$
- High-threshold model:
$$s_\infty(r) = \frac{[\,r - r_{\text{th}}\,]_+}{[\,r - r_{\text{th}}\,]_+ + 20\,\text{Hz}}$$

- Bistable-dendrites model:

$$s_\infty(r) = \begin{cases} 0, & \text{if } r < r_{\text{off}}, \\ \text{history dependent}, & \text{if } r_{\text{off}} < r < r_{\text{on}}, \\ 1, & \text{if } r_{\text{on}} < r. \end{cases}$$

The new models offered more stability, as it required more drastic shifts in recurrent output to cause a difference in neuronal activation.

2.1.2 Tuning Curves and Model Parameters

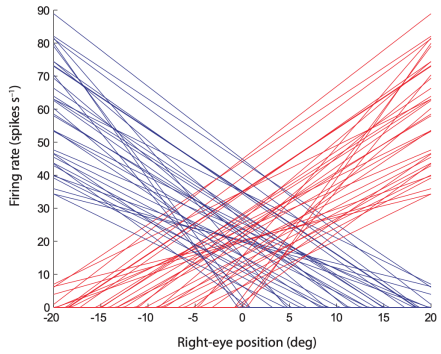

Figure 2: Tuning Curves, derived from Goldman [1]

The experimental data acquired in the study [1] also included **tuning curves**. Tuning curves are the relation between the eye position and the firing rate. Each neuron fires at a certain rate given a certain eye position, as displayed at Figure 2. Each eye position is a steady state, meaning the firing rates should remain stationary. From the experiment, we get:

$$r_i = \left[k_i E + r_i^0\right]_+$$

where E is the eye position, k_i is the slope of the tuning curve, which denotes the eye position sensitivity of that neuron and r_i^0 is the firing rate at $E = 0$. We can identify k_i with ξ_i and r_i^0 with T_i, and this suggests that the recurrent output, $S_R - S_L$, can be interpreted as an inner representation of the eye position held by the OI. With these assumptions, we can recover the eye position representation from the OI model by:

$$E = S_R - S_L \tag{6}$$

$$= \sum_{i=1}^{n} \eta_i \, s_\infty\left(r_i^R\right) - \sum_{i=1}^{n} \eta_i \, s_\infty\left(r_i^L\right) \tag{7}$$

$$= \sum_{i=1}^{n} \eta_i \, s_\infty(\xi_i E + T_i) - \sum_{i=1}^{n} \eta_i \, s_\infty(-\xi_i E + T_i)\,. \tag{8}$$

Although η values were published [1], no raw tuning curve data was included. We used tuning curves data extracted from the plot by a previous work in the lab. However, the published η values were now not consistent with ξ and r_0. Since there's a linear relation between the eye position and between $s_\infty(r_i^R)$ and $s_\infty(r_i^L)$, we could find η using linear regression with labels created by our tuning curves – by a choice of s_∞, we can create a set of samples (x, y) where $x \in \mathbb{R}^n$ is a vector of firing rates after activation in the y eye position. The fit was done using slightly different parameters (using classical linear regression instead of lsqlin, using the traditional activation and without constraints on η).

2.1.3 Simulating the Network

Since this dynamic is not linear, and as mentioned before, has no analytic solution, we use numeric methods to simulate it. The main method used is **Euler Approximation**. From Peano's existence theorem, we know that given $\frac{dx}{dt} = F(x)$ where F is locally Lipschitz, a solution that starts at $x(0) = x_0$ exists and is unique. The proof uses piecewise linearization with small Δt time steps. Therefore, when F is indeed locally Lipschitz, x could be approximated using linearization by:

$$x\left(t_n\right) = x\left(t_{n-1}\right) + \Delta t x'\left(t_{n-1}\right)$$

and for small Δt values we would get a close approximation of x.

We then extracted the eye position representation held by the OI from $S_R - S_L$ and ran various simulations, including some saccadic bursts input. This step ensured the model was implemented correctly in code. Such simulation appears in Figure 4.

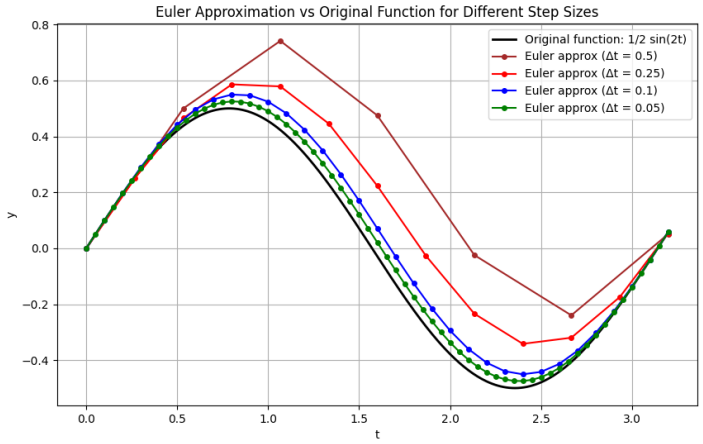

Figure 3: Example of Euler Approximation

Further explanation: This figure illustrates a simple Euler approximation of the continuous function $\frac{1}{2}\sin(2t)$. It shows how decreasing Δt improves accuracy by more compute.

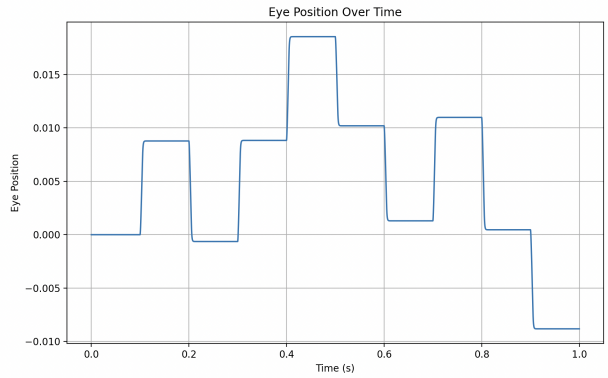

Figure 4: Intermediate result — Deterministic Eye Position Simulation with Saccades.

Further explanation: The simulation shows deterministic eye position dynamics, including brief saccadic displacements followed by stable fixations, simulated by Euler approximation (eye position given in degrees, extracted by $S_R - S_L$). Each saccade shifts the eye position, followed by exponential relaxation back toward equilibrium. Also, the fixations demonstrate the stable states of the dynamic, as the network remains stable between saccades although it doesn't receive input, confirming the attractor dynamics.

2.2 Stochasticity In The OI

2.2.1 Motivation

While the deterministic model [1] successfully reproduces saccade and fixation dynamics, real neural networks are inherently noisy. As mentioned, neural output is naturally binary, as a neuron either spikes or doesn't spike – and spiking is a discrete and stochastic event. We used firing rate as a measure of the frequency of spikes, but as the output is binary, the frequency is derived from the number of spikes per second.

However, if a neuron fires r times per second, there is uncertainty as to when exactly it will fire. Each Neuron's firing is modeled by the **spike train** – that is, the sequence of spikes – which is a **Poisson process**. Given an instantaneous firing rate $r(t)$, the neuron produces spikes with probability proportional to $r(t)$.

2.2.2 Spike Train Based Model

The study conducted by NBS [2], based on Goldman's [1] framework, proposed equations that incorporates such stochasticity. The offered OI model is governed by the following equations:

$$r_i^R(t) = \left[\xi_i E(t) + r_i^0 + \xi_i F(t)\right]_+$$

$$\tau_s \frac{d}{dt} s_i^R(t) = -s_i^R(t) + X_i^R$$

$$E(t) = \sum_{i=1}^{n} \eta_i \left(s_i^R(t) - s_i^L(t)\right)$$

where ξ_i is the sensitivity of neuron n_i to input, $E(t)$ is the eye position, $F(t)$ is a feedback term (which was not considered in this project) and X_i^R holds:

$$X_i^R(t) = \frac{\zeta_i^R}{r_i^R(t) + \lambda_0} \ (*)$$

as λ_0 is a base firing rate and ζ_i^R is the spike train. To illuminate, the spike train itself is expressed as a sum of delta functions at spike times, produced as an inhomogeneous Poisson process with rate r:

$$\zeta_i^R = \sum_j \delta\left(t - t_j\right)$$

Dividing by $r_i^R(t)$ normalizes contributions so that the average effect matches Goldman's [1] deterministic term. Replacing ζ_i^R with $r_i^R(t)$ recovers the original model.

The firing rate can also be reconstructed from the spike train by convolving ζ_i^R with an exponential decaying kernel. This makes explicit the dual representation: spikes as discrete events and rates as their smoothed average.

Since the dynamic is not linear, we wish to simulate it to explore its statistics. However, ζ_i^R is not continuous, and thus this simulation requires more delicate care, because Euler approximation fails in simulating the dynamic. If simulating s_i^R would've originally been done by:

$$s_i^R(t_n) = s_i^R(t_{n-1}) + \Delta t \cdot \dot{s_i^R}(t_{n-1}) \tag{9}$$

$$= s_i^R(t_{n-1}) + \Delta t \cdot \tfrac{1}{\tau}\left(-s_i^R(t_{n-1}) + X_i^R(t_{n-1})\right). \tag{10}$$

then to simulate the spike train correctly, we treat X_i^R as a discrete event indicator. Therefore, if X_i^R is multiplied by Δt, its impact would be underestimated. Thus, because we would like a spike occurring within that Δt time frame to count in full and not be scaled by the Δt width, we no longer multiply X_i^R by Δt.

This event-driven interpretation also respects units and dimensions. $\dot{s_i^R}$ represents synaptic output change per unit of time, and therefore $\tau_s \dot{s_i^R}$ is timeless; then, X_i^R must not be scaled by time, as it should only represent synaptic output contribution. Therefore, to simulate the OI with a spike train, we simulate by:

$$s_i^R\left(t_n\right) = s_i^R\left(t_{n-1}\right) + \Delta t \cdot \left(-s_i^R\left(t_{n-1}\right)\right) + X_i^R$$

Empirical evidence shows the OI is sub-Poisson, with coefficient of variation (CV) of inter-spike intervals ≈ 0.22, whereas a standard Poisson process has $CV = 1$ (where $CV = \frac{\sigma(\text{ISI})}{\mathbb{E}[\text{ISI}]}$ and ISI stands for inter-spike intervals). Therefore, to achieve biological accuracy, we need a procedure to generate spike trains with reduced variability.

2.2.3 The Thinning Technique

One technique to reduce the variability is sub sampling the spikes.

Claim 1. *Let $\lambda \in \mathbb{R}_+$ and $M \in \mathbb{N}$. If we sample every M'th spike from a Poisson process with rate λM, the resulting process has $CV = \frac{1}{\sqrt{M}}$.*

Proof. Denote time between consecutive in the parent process as $T_i \sim \text{Exp}(\lambda M)$. Since we sample every M'th spike, the ISI is then:

$$\widetilde{T} = \sum_{i=1}^{M} T_i$$

for $T_1, \ldots, T_n$ independent variables. From linearity of expectation:

$$\mathbb{E}\left[\widetilde{T}\right] = \sum_{i=1}^{M} \mathbb{E}\left[T_i\right] = \sum_{i=1}^{M} \frac{1}{\lambda M} = \frac{1}{\lambda}$$

and from linearity of variance of independent variables:

$$\mathrm{Var}\left(\widetilde{T}\right) = \sum_{i=1}^{M} \mathrm{Var}\left(T_i\right) = \sum_{i=1}^{M} \frac{1}{M^2\lambda^2} = \frac{1}{M\lambda^2}$$

Hence $\sigma\left(\widetilde{T}\right) = \frac{1}{\sqrt{M}\lambda}$. Therefore, by the definition of CV:

$$CV = \frac{\sigma\left(\mathrm{ISI}\right)}{\mathbb{E}\left[\mathrm{ISI}\right]} = \frac{\frac{1}{\sqrt{M}\lambda}}{\frac{1}{\lambda}} = \frac{1}{\sqrt{M}}$$

□

Claim 2. *Let $\lambda \in \mathbb{R}_+$ and $M \in \mathbb{N}$. If we sample every M'th spike from a Poisson process with rate λM and denote the number of spikes at time T by a random variable $N_{sub}(T)$, then:*

$$\lim_{T\to\infty} \frac{N_{sub}(T)}{T} = \lambda$$

Proof. Denote $N(T)$ the spike count of the original Poisson process. Taking every M'th spike is equivalent to:

$$N_{sub}(T) = \left\lfloor \frac{N(T)}{M} \right\rfloor$$

for all $T \in (0, \infty)$,

$$\frac{N(T)}{M} - 1 < \left\lfloor \frac{N(T)}{M} \right\rfloor \leq \frac{N(T)}{M}$$

thus:

$$\frac{N(T)}{M} - 1 < N_{sub}(T) \leq \frac{N(T)}{M}$$

dividing by T gives

$$\frac{N(T)}{MT} - \frac{1}{T} < \frac{N_{sub}(T)}{T} \leq \frac{N(T)}{MT}$$

from the sandwich theorem, as $T \to \infty$, we get:

$$\lim_{T\to\infty} \frac{N_{sub}(T)}{T} =^{a.s} \lim_{T\to\infty} \frac{N(T)}{MT}$$

Now, from properties of Poisson process, $\mathbb{E}\left[N(T)\right] = \lambda M$, hence from the law of large numbers,

$$\lim_{T\to\infty} \frac{N(T)}{T} = \mathbb{E}\left[N(T)\right] = \lambda M$$

and thus, we obtain:

$$\lim_{T\to\infty}\frac{N_{sub}(T)}{T} =^{a.s} \frac{1}{M}\lim_{T\to\infty}\frac{N(T)}{T} = \frac{\lambda M}{M} = \lambda$$

□

To conclude, given the claims, by using the thinning technique with $M = 25$, we can artificially reduce CV to 0.2 without altering the average firing rate.

2.2.4 Variance Reduced by Increasing Neuron Count

NBS's [2] study reviewed monkeys, which are primates, and their OI consists of significantly more neurons than the model presented by Goldman [1], which assumes 36 neurons per population. As a result, directly simulating Goldman's [1] network stochastically leads to excessively noisy activity compared with biological data. To overcome this noise, we wish to reduce the variance of the number of spikes per second, while preserving the mean, by artificially increasing the number of neurons.

Our strategy was to simulate each original neuron as N independent identical copies. Each copy inherits the same parameters (such as r_0 and ξ) but is simulated on its own by generating an independent spike train. Averaging across these copies yields a more stable signal, and instead of using equation $(*)$ we define:

$$\widetilde{X_i^R} = \frac{\frac{\sum_{j=1}^{N}\zeta_{i,j}}{N}}{r_i^R(t)+\lambda_0}$$

where $\zeta_{i,j}$ is a spike train generated randomly based on the fire rate of neuron n_i^R and for all $k \neq j$, $\zeta_{i,k}$ and $\zeta_{i,j}$ are independent.

Claim 3.

$$\mathbb{E}\left[\widetilde{X_i^R}\right] = \mathbb{E}\left[X_i^R\right],\ \ Var\left(\widetilde{X_i^R}\right) = \frac{1}{N}Var\left(X_i^R\right)$$

Proof. By linearity of expectation,

$$\mathbb{E}\Big[\widetilde{X_i^R}\Big] = \frac{1}{r_i^R(t)+\lambda_0}\cdot\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}[\zeta_{i,j}] = \frac{1}{r_i^R(t)+\lambda_0}\cdot\frac{1}{N}\sum_{j=1}^{N}\mathbb{E}[\zeta_i] \tag{11}$$

$$= \frac{1}{r_i^R(t)+\lambda_0}\cdot\frac{1}{N}\cdot N\,\mathbb{E}[\zeta_i] = \frac{\mathbb{E}[\zeta_i]}{r_i^R(t)+\lambda_0} = \mathbb{E}\big[X_i^R\big] \tag{12}$$

For the variance, using the linearity regarding independent variables:

$$\mathrm{Var}\left(\widetilde{X_i^R}\right) = \left(r_i^R(t) + \lambda_0\right)^{-2} N^{-2} \sum_{j=1}^{N} \mathrm{Var}(\zeta_{i,j}) \tag{13}$$

$$= \left(r_i^R(t) + \lambda_0\right)^{-2} N^{-2} \sum_{j=1}^{N} \mathrm{Var}(\zeta_i) \tag{14}$$

$$= \left(r_i^R(t) + \lambda_0\right)^{-2} N^{-2} \cdot N \,\mathrm{Var}(\zeta_i) \tag{15}$$

$$= \frac{1}{N} \mathrm{Var}\left(X_i^R\right) \tag{16}$$

which completes the proof. □

Thus, by increasing N, we achieve a variance reduction by a factor of N while keeping the expected value unchanged. As a matter of fact, the model can be scaled to any desired population size, allowing us to approximate the reduced variability of larger biological networks.

To conclude, we used both techniques (thinning and artificially increasing the number of neurons) to achieve reduced variability. From previous work [4], it is known that the diffusion coefficient characterizing neural noise in continuous attractor networks – and, in our case, the stochastic drift of eye position during fixation – scales approximately:

- inversely with population size ($\sim \frac{1}{N}$)
- proportionally with the squared coefficient of variation ($\sim CV^2$)
- inversely with the square of the synaptic time constant ($\sim \frac{1}{\tau_s^2}$).

Therefore, to reduce the noise in our simulations, we chose to simulate 1000 neurons for each original neuron (yielding $2 \times 36 \times 1000 = 72,000$ neurons simulated across both populations). We combined this with a thinning factor of $M = 25$ and increased τ_s to 100 milliseconds from 10. All of which contributed to more biologically realistic levels of variability.

2.2.5 Sampling Spikes in Code

We first assumed there could only be one spike in each Δt time frame. This assumption is reasonable since when taking $\Delta t \to 0$, we get:

$$\mathbb{P}\left(N(\Delta t) > 1\right) = 1 - \mathbb{P}\left(N(\Delta t) = 0\right) - \mathbb{P}\left(N(\Delta t) = 1\right) = 1 - e^{-\lambda \Delta t}(1 + \lambda \Delta t) \to 0$$

so small values of Δt result in the probability of more than 1 spike per time interval to become negligible.

With empirical firing rates below $100Hz$ and M (thinning factor) chosen to be at most 25, choosing $\Delta t = 10^{-5}$ seconds results in the probability of 2 spikes being below $\frac{1}{100}$.

This assumption allows us to treat each simulation step as an independent Bernoulli trial with probability $\Delta t\lambda$.

Claim 4. *The expectation of the sum of independent Bernoulli trials is the same as of a Poisson process, and the variance goes to the Poisson process' variance as* $\Delta t \to 0$.

Proof. Denote the random variable of number of spikes in time T:

$$N(T) = \sum_{i=1}^{\frac{T}{\Delta t}} X_i$$

where $X_i \sim \text{Ber}(\lambda\Delta t)$, we get by linearity of expectation:

$$\mathbb{E}\left[N(T)\right] = \sum_{i=1}^{\frac{T}{\Delta t}} \mathbb{E}\left[X_i\right] = \sum_{i=1}^{\frac{T}{\Delta t}} \lambda\Delta t = \lambda T$$

and:

$$\text{Var}\left(N(T)\right) = \sum_{i=1}^{\frac{T}{\Delta t}} \text{Var}\left(X_i\right) = \sum_{i=1}^{\frac{T}{\Delta t}} \lambda\Delta t(1-\lambda\Delta t) = \lambda T(1-\lambda\Delta t)$$

and as $\Delta t \to 0$ we get $\text{Var}\left(N(T)\right) \to \lambda T$, which completes the proof.

One can also show that as $N \to \infty$, the sum of N independent Bernoulli trials converges to a Poisson-distributed variable (given $\Delta t \to 0$).

□

Therefore, sampling with Bernoulli independent trials per time step reproduces the correct statistical properties, while being straightforward to implement computationally.

2.2.6 Results

Our goal was to observe the MSD of the stochastic OI model and learn about its diffusion coefficient. For a process $x(t)$, we define its MSD (mean squared displacement) by:

$$MSD(\tau) = \mathbb{E}\left[\left(x(t+\tau) - x(t)\right)^2\right]$$

In our case, we take the expectation as the average of multiple time windows (τ) and simulations. We ran 1000 independent simulations of the stochastic OI model, each lasting 25 seconds. Then, we calculated an average of each simulation's MSD. The results revealed a two-phase behavior: diffusion dominated up to roughly 10^{-2} seconds, followed by sub diffusive dynamics thereafter.

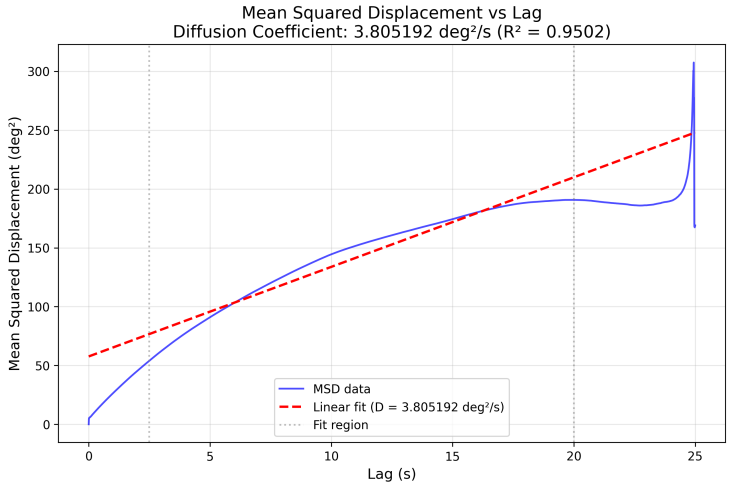

Figure 5: MSD plot for a 25-second simulation.

Further explanation: This figure shows the MSD of eye position over time, averaged from 1000 simulations, 25 seconds each. The curve initially grows approximately linearly, consistent with diffusive dynamics, before approaching a plateau indicating bounded behavior.

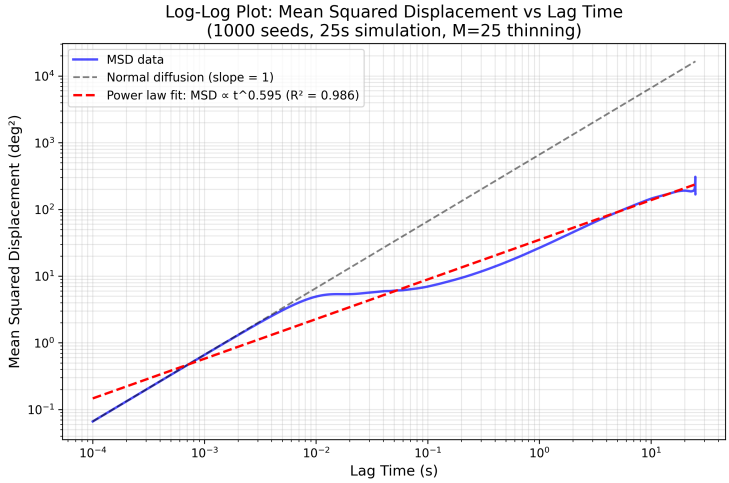

Figure 6: MSD log-log Plot for a 25-second simulation.

Further explanation: This figure shows the same results as in Fig. 5, in a log-log manner. It demonstrates normal diffusion until 10^{-2} seconds, then changes the slope of the linear growth, ending with plateau.

This finding was unexpected, as it did not align with the results reported by NBS [2], that reported more consistent diffusive behavior across time scales. The cause of the inconsistency between the results remains unresolved, as the deterministic simulation exhibits stability, whereas the non-deterministic version displays an excessive level of absolute noise, as can be observed in the simulation (Figure 7).

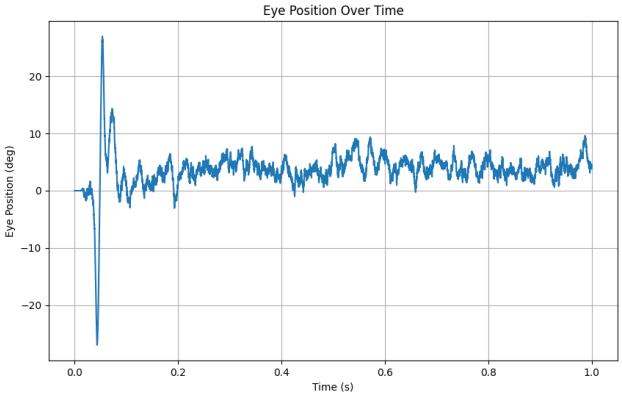

Figure 7: Eye Position Extracted From Simulation.

Observing Figure 7, we ignore the sharp initial change as it arises from the thinning procedure, because in the first iterations of the simulation, all neurons are prevented from spiking; nevertheless, the noise in later iterations remained substantially higher compared to the noise observed in simulations conducted by NBS (unpublished work). We were unable to explore the origin of these differences in the scope of this project.

2.3 The Connectome Based Model

2.3.1 Motivation

Recent technological advances in experimental neuroscience enable measurement of the connectivity in full neural circuits with synapse-level resolution, based on analysis of electron microscopy images of tissue slices. There are several such large scale projects in the world, each involving dozens or hundreds of researchers, and one of these projects has mapped the connectivity in parts of the zebrafish oculomotor integrator. We now review this connectome-based [3] study.

2.3.2 Rate Model and Parameters

The connectome-based model [3] assumes is that the strength of connection between neurons is proportional to the number of synapses they share. The

researchers were able to count the number of shared synapses, and thus an entry of the connectivity matrix holds:

$$W_{i,j} = \pm\beta \frac{N_{i,j}}{\sum_{k=1}^{n} N_{i,k}}$$

where $N_{i,j}$ is the number of synapses neuron n_i and neuron n_j share. That is, $W_{i,j}$ is the number of shared synapses between n_i and n_j out of all the synapses of neuron n_i. β is a scaling coefficient described later, and the sign encodes excitation/inhibition. Using this connection martix, the authors presented a linear rate model of the OI:

$$\tau \frac{dr_i(t)}{dt} = -r_i(t) + \sum_{j=1}^{n} W_{i,j} r_j(t)$$

As this is a linear model, an analytic solution exists. The authors [3] left two free parameters for tuning: τ and β.

- **The time constant τ:** τ represents the intrinsic cellular time constant. It determines the persistence of the network's dynamics. As shown earlier, the solution to
$$\tau \frac{dx}{dt} = -x$$
is
$$x(t) = x(0)e^{-\frac{t}{\tau}}.$$
Thus, larger values of τ lead to slower exponential decay and longer memory retention within the network, whereas smaller τ values cause faster decay and shorter persistence (similar to the discussion on the end of Section 1.1.2).

- **The scaling factor β:** β is a unitless coefficient that scales the connectivity matrix W and therefore controls the eigenvalues of the system dynamics. If λ is an eigenvalue of W, then the corresponding eigenvalue of the dynamic system is
$$\frac{\beta\lambda - 1}{\tau}.$$
This follows from the matrix equation:
$$\frac{d\mathbf{r}}{dt} = \frac{1}{\tau}(-I + \beta W)\mathbf{r}.$$
Hence, if v is an eigenvector corresponding to λ, we have:
$$\frac{1}{\tau}(-I + \beta W)v = \frac{1}{\tau}(-v + \beta\lambda v) = \frac{\beta\lambda - 1}{\tau}v,$$
proving that $\frac{\beta\lambda-1}{\tau}$ is indeed an eigenvalue of the dynamics.

From the data obtained by the authors [3], the intrinsic cellular time constant was $\tau = 1\,\mathrm{s}$ and β was chosen such that the leading eigenvalue of βW (λ such that $\Re(\lambda)$ is maximal) would be 0.9, meaning:

$$\beta = \frac{0.9}{\max\left\{\Re\left(\lambda_i\right) | \lambda_i \; is \; an \; eigenvalue \; of \; W\right\}}$$

By choosing τ and β as described above, the dynamic's real part of the leading eigenvalue is $\frac{0.9-1}{\tau} = -0.1$. Hence, the slowest direction of the dynamic decays with time constant of $\frac{1}{0.1} = 10$ seconds.

In Figure 8, we can see a simulated saccade after 1 second, followed by a decay of ≈ 10 seconds, which is consistent with the leading eigenvalue.

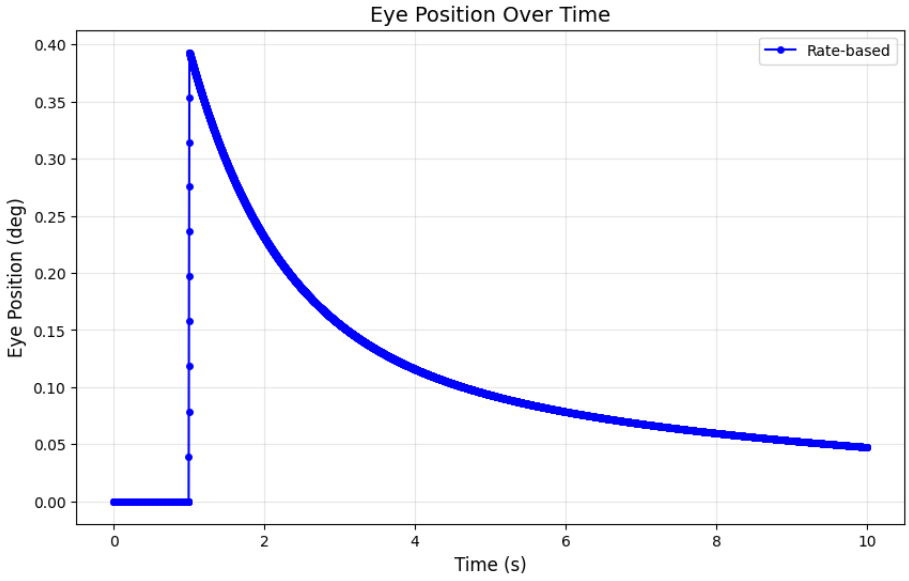

Figure 8: Eye Position Simulation With Saccade

Further explanation: This figure shows the deterministic eye position dynamics produced by the simulated connectome-based model. A single saccadic input is inserted after 1 second, after which the activity decays exponentially.

2.3.3 Modeling with a Stochastic Approach

Moving forward, we wish to apply the stochastic approach offered by NBS [2] on the connectome-based model [3]. However, the connectome model is linear and unrectified. Consequently, there could be negative firing rates, from which a Poisson process can't be generated. Also, rectifying would violate the stability of the dynamic. Therefore, we decided to simulate by the following update rule:

$$\tau r_i\left(t_n\right) = -r_i\left(t_{n-1}\right) + \sum_{j=1}^{n} W_{i,j}\zeta_j\left(t_{n-1}\right)$$

$$\zeta_j(t) = \begin{cases} \sum_k \delta\Big(t - t_k^{(j)}\Big), & \text{if } r_j(t) \geq 0, \\ \Delta t \cdot r_j(t), & \text{if } r_j(t) < 0. \end{cases}$$

meaning when $r_j(t) > 0$ we generate the Poisson spike train with rate $r_j(t)$, and when $r_j(t) < 0$ we substitute with $\Delta t \cdot r_j(t)$, normalizing $r_j(t)$ to the time step.

2.3.4 Offsetting Eye Position to Evoke Stochastic Activity

Since the network is linear, at eye position 0 all firing rates are also 0, so stochastic spiking does not arise spontaneously. To simulate a noisy fixation, we would require to artificially move the eye to a non-zero position and then enable our stochastic approach. Our strategy is to add a constant external drive to the network that is the scaled leading eigenvector, that corresponds with the leading eigenvalue.

Claim 5. *The linear model converges to a non-zero constant as $t \to \infty$ given a constant non-zero outside input.*

Proof. Let $A \in \mathbb{R}^{n\times n}$ be $A = \frac{(-I+\beta W)}{\tau}$ and $v \neq 0 \in \mathbb{R}^n$. From the Duhamel's principle, we get that a solution for $\frac{d\mathrm{r}}{dt} = A\mathrm{r} + v$ is given by:

$$\mathrm{r}(t) = e^{tA}\mathrm{r}(0) + \int_0^t e^{A(t-s)}v ds$$

Since the largest real part of A's eigenvalues is $-\frac{0.1}{\tau}$, we deduce that A is invertible. Therefore:

$$\int_0^t e^{A(t-s)}v\, ds = \left(\int_0^t e^{A(t-s)}ds\right)v = e^{At}\int_0^t e^{-As}ds\, v \tag{17}$$

$$= -e^{At}A^{-1}\int_0^t (-A)e^{-As}ds\, v = -e^{At}A^{-1}\left[e^{-As}\right]_0^t v \tag{18}$$

$$= -e^{At}A^{-1}\big(e^{-At} - I\big)v = -A^{-1}v + e^{tA}v. \tag{19}$$

Now since $\max\left\{\Re\left(\lambda_i\right)|\lambda_i \textit{ is an eigenvalue of } A\right\} < 0$ and $\mathrm{r}(0)$ and v are constants:

$$\lim_{t\to\infty} e^{tA}\mathrm{r}(0) = 0,\ \lim_{t\to\infty} e^{tA}v = 0$$

and eventually:

$$\lim_{t\to\infty} \mathrm{r}(t) = -A^{-1}v$$

as $v \neq 0$ and A is invertible, we conclude that $\lim_{t\to\infty}\mathrm{r}(t) \neq 0$.

□

Given the claim, we can reach a steady state by simulating the network with external drive for sufficiently long time.

In Figure 9, we can see how after ≈ 50 seconds, the eye position reaches a plateau of $\approx 3^\circ$ given a constant external drive. This result is consistent with the fact that linear systems typically converge after about 5 time constants, and the time constant of this network is 10 seconds. Elaboration on extracting the eye position from firing rate, as we've done in this simulation, follows in Section 2.3.5.

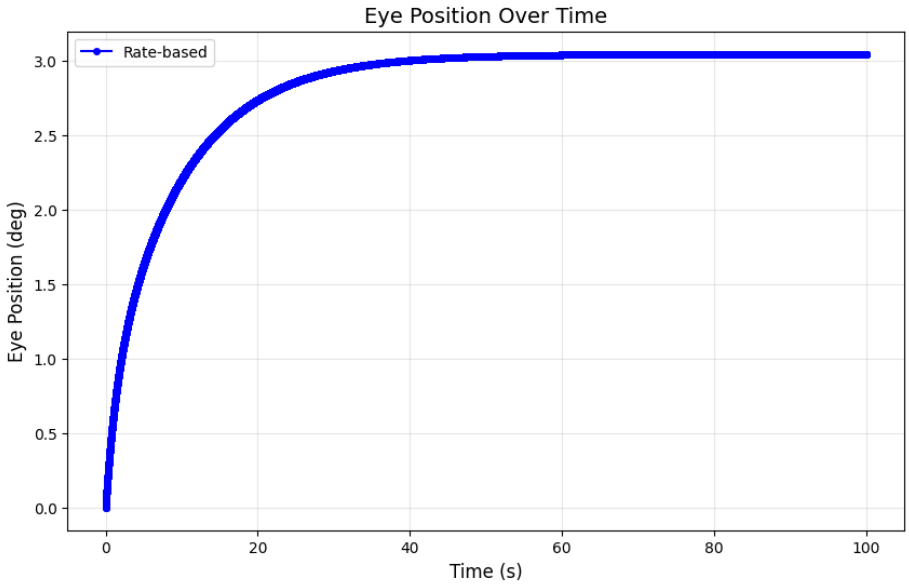

Figure 9: Eye Position Converging Given Constant External Input

Further explanation: This figure shows how the simulated eye position evolves under a constant external input. The steady input projects along the leading eigenvector of the connectome-based model, shifting the network activity toward a new equilibrium. Over time, the system converges exponentially to a stable nonzero eye position.

2.3.5 Extracting Eye Position from Firing Rates

To compute the MSD of a stochastic simulation of the connectome-based network, we first needed to derive tuning curves, since the simulation only simulates firing rates (from which we aim to extract the eye position). Since the model approximates a persistent line attractor, there exists an eigenvector whose span is a continuum of steady (or almost steady, in the connectome based dynamic) states. The entries of this eigenvector specify how each neuron's firing rate scales with eye position. Given the equation:

$$r = k \cdot E$$

for the firing rate r, slope of the tuning curve k and eye position E, we perform the calculation $E = \frac{r}{k}$ to extract the eye position from the firing rate, where k is the i'th entry of the leading eigenvector. We took E to be:

$$E = \text{avg}\left(\left\{ \frac{r_i}{k_i} \;\middle|\; |r_i| > 0.01 \right\}\right).$$

This excludes neurons with negligible or irrelevant correlation to eye position.

2.3.6 Understanding Units by Cross Model Comparison

In contrast to the previous model [1], which reported eye position in degrees, the connectome-based study [3] normalized positions for comparability across animals: subtracting the median and dividing by the 95th percentile. As a result, the slopes we obtain by the eigenvalue decomposition are unitless.

To interpret the units of the eye position, we compared the slopes that we found using the leading eigenvector with those Goldman reported [1]. Because the animals are similar, we rescaled the connectome-based slopes to match Goldman's [1] mean and standard deviation.

Then, we observed that the firing rates in the connectome-based [3] simulation were about two orders of magnitude smaller than Goldman's [1]. To correct for this and place the models on comparable scales, we multiplied the firing rates by 20 to normalize the firing rates to a similar value before extracting eye positions.

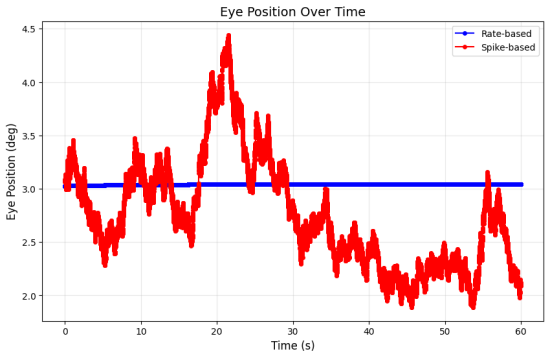

Figure 10: Simulating Both Deterministic and Stochastic Models

Further explanation: The figure displays the eye position extracted from the last 50 seconds of a 100 seconds simulation that started by stabilizing the network to a non-zero constant. Both deterministic and stochastic (spike train based) simulations included a constant external input.

2.3.7 Results

Examination of the MSD curve revealed that its form closely resembles that predicted by an Ornstein–Uhlenbeck process, meaning:

$$MSD(\tau) = \frac{2D}{\kappa} \cdot \left(1 - e^{-\kappa\tau}\right)$$

and thus we can derive that the process is approximately linear for small values of τ and saturates to a plateau for larger τ.

Curve fitting with `scipy.optimze.curve_fit` yielded $\kappa = 0.142\,\mathrm{s}^{-1}$ and diffusion coefficient of $D = 0.0176\frac{\mathrm{deg}^2}{\mathrm{s}}$. The corresponding characteristic time constant is $\frac{1}{\kappa} \approx 7.042$ seconds.

This estimate is in good agreement with the persistence predicted by the leading eigenvalue of the dynamics (≈ 10s). The diffusion coefficient is also close to empirical values reported in unpublished laboratory data ($0.15 \pm 0.09\frac{\mathrm{deg}^2}{\mathrm{s}}$). This suggests that with biophysically reasonable parameter choices (such as τ), the connectome-based model yields realistic diffusion levels.

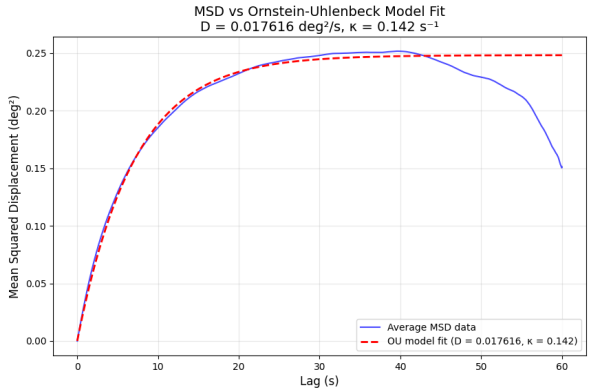

Figure 11: MSD Plot – 60 Seconds Connectome-Based Simulation

Further explanation: The figure displays MSD of eye position, averaged across 1000 simulations of 60 seconds (after stabilizing with external drive). We can see linear growth at start, followed by saturation to a plateau for larger time lags, which is consistent with an Ornstein–Uhlenbeck process.

3 Code Availability

The code is available to view in the Github repository, with supplementary installation instructions, at `https://github.com/yuval-rubin1/Amirim-Project`.

References

[1] Aksay, E., Olasagasti, I., Mensh, B. *et al.* Functional dissection of circuitry in a neural integrator. *Nat Neurosci* **10**, 494–504 (2007). https://doi.org/10.1038/nn1877

[2] Ben-Shushan, N., Shaham, N., Joshua, M. *et al.* Fixational drift is driven by diffusive dynamics in central neural circuitry. *Nat Commun* **13**, 1697 (2022). https://doi.org/10.1038/s41467-022-29201-y

[3] Vishwanathan, A., Sood, A., Wu, J. *et al.* Predicting modular functions and neural coding of behavior from a synaptic wiring diagram. *Nat Neurosci* **27**, 2443–2454 (2024). https://doi.org/10.1038/s41593-024-01784-3

[4] Burak, Y., Fiete, I.R. Fundamental limits on persistent activity in networks of noisy neurons. *Proc. Natl. Acad. Sci. U.S.A.* **109**, 17645–17650 (2012). https://doi.org/10.1073/pnas.1117386109