Inferring activity statistics in Stabilized Supralinear Networks

Undergraduate Thesis

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Vanshika Kapoor
ID No. 2017B5A70624G

Under the supervision of:

Dr. Julijana Gjorgjieva, Dr. Dylan Festa
&
Dr. Gaurav Dar

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS
December 2021

Declaration of Authorship

I, Vanshika Kapoor, declare that this Undergraduate Thesis titled, 'Inferring activity statistics in Stabilized Supralinear Networks' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Certificate

This is to certify that the thesis entitled, "*Inferring activity statistics in Stabilized Supralinear Networks*" and submitted by Vanshika Kapoor ID No. 2017B5A70624G in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by her under my supervision.

Supervisor	*Co-Supervisor*
Dr. Julijana Gjorgjieva, Dr. Dylan Festa	Dr. Gaurav Dar
Professor/Group Leader,	Asst. Professor,
Max Planck Institute for Brain Research	BITS-Pilani Goa Campus
Date:	Date:

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS

Abstract

Masters in Physics

Inferring activity statistics in Stabilized Supralinear Networks

by Vanshika Kapoor

Computational models are a valuable tool to understand the mechanisms and functions of the brain. In particular, network models of recurrently connected excitatory and inhibitiory (E/I) neural populations can be used to interpret neural response in cortical areas and to model brain functions and pathological states [Yus15].

Stabilized supralinear networks (SSN) [ARM13] are a category of E/I, rate-based models of particular interest. It has been shown that SSNs capture several biologically relevant aspects of neuronal dynamics, such as nonlinear integration of visual inputs [RVM15], variability modulation [Hen+18], sampling-based probabilistic computations [Ech+20], and positive population correlations [AM19].

Theoretical and computational studies on SSNs are based on mean-field approaches, where the network is reduced to two dimensions [e.g. ARM13], or parametrizations with a highly regular connectivity structure [e.g. Ech+20]. This limitation can be overcome by theoretical tools based on Hawkes processes [Ock+17a; Haw71], making it possible to express the statistics of neural activity for arbitrarily connected systems of firing neurons. Hawkes processes are stochastic point processes. They capture probabilistic occurrences of interdependent events, repeated in time.

In this thesis, we explore the possibility of using Hawkes processes as an alternative to SSNs by mapping the defining variables of an SSN network to the defining variables of a Hawkes process, and comparing important properties of the two models, namely : rate and variability. We begin by considering a smaller problem of comparing Linear Rate Model with Linear Hawkes Processes. This analysis is then extended to the nonlinear regime in two ways. First, SSN network is linearized around its stable point, and the resultant linearly approximated network is compared with linear Hawkes. Second, making a direct comparison of the SSN network with nonlinear Hawkes Processes.

Acknowledgements

I sincerely thank my supervisor Dr. Julijana Gjorgjieva for providing me with the opportunity to work under her. My deepest gratitude to Dr. Dylan Festa. He has guided me through each step of the project with immense support and encouragement. A huge thanks to my co-supervisor Dr. Gaurav Dar, who introduced me to the world of theoretical neuroscience and computational physics. It has been an absolute delight to work with and learn from him throughout my undergraduate studies.

Contents

List of Figures

Chapter 1

Introduction

In an attempt to understand how the brain works, computational neuroscience simulates the functions and behaviour of individual regions of the brain on computers. This is done by creating models of neurons and networks of neurons, encompassing varying depths of detail based on what the model is going to be used for. Activities in these neuronal networks depend on neural connectivity, i.e. how are different neurons in the network linked to each other, and also on the intrinsic dynamics of individual neurons. The intrinsic dynamics describes the neural response to inputs.

Stabilized Supralinear Networks (SSN) [ARM13] are a widely used category of rate-based models which have been shown to capture several biologically relevant aspects of neuronal dynamics. They model the activities of small neuronal populations, on top of which larger models capturing the behaviour of different brain regions can be built. SSNs include recurrently connected excitatory and inhibitory (E/I) neural populations, and can be used to interpret neural response in cortical areas and to model brain functions and pathological states [Yus15].

Theoretical and computational studies on SSNs are based on mean-field approaches, where the network is reduced to two units [e.g. ARM13], excitatory and inhibitory. These units can be interpreted as a population of individual E or I neurons respectively, with the SSN providing information about the behaviour of the population as a whole. The described behaviour of whole population unit approximately captures each individual neuron's behaviour. Hence, SSNs provide a good overview of neuronal activity, but they also have limitations. For example, two important properties (which are going to be considered throughout this report) which cannot be extracted from an SSN are the spike trains of neurons and their variability. Additional tools need to be added to the vanilla SSN differential equation to get these properties.

To overcome this limitation, I will consider theoretical tools based on Hawkes processes [Haw71; Ock+17a].

Hawkes processes are stochastic point processes. They have already been applied in diverse areas, from earthquake modelling to financial analysis. [LTP15] They capture probabilistic occurrences of interdependent events, repeated in time. An important feature of Hawkes Processes is that they are self-exciting, which means that each arrival increases the rate of future arrivals for some period of time. This behaviour is similar to spiking in neurons, and hints at a connection between a network of neurons and Hawkes processes.

This project analyses the possibility of replacing SSNs with Hawkes processes. Hawkes processes are of particular interest as they have extensively studied properties, and well-defined techniques to find relevant information about them, such as analytic expressions for second and higher order statistics. We start with an E/I SSN, parametrized and motivated by studies on neural dynamics, and explore the possibility of using Hawkes processes as an alternative to the SSN in the same neuroscience problems by mapping the defining variables of an SSN to the defining variables of a Hawkes process, and comparing important properties of the two models.

We begin by introducing Stabilized Supralinear Networks and analysing their behaviour around stable points. Next, we discuss their limitations, the inability to provide spike-trains and to produce intrinsic variability. We move on to studying ways to overcome this shortcoming, i.e. by adding noise and generating spikes with a probability directly proportional to the instantaneous rate of a neuron. Another important alternative discussed is Spiking SSNs, where there spike trains are fed back into the SSN differential equation so that the past spikes have a direct role in present spiking, based on how much in the past did they occur.

We then provide a list of motivations for working with Hawkes processes, and move on to defining and explaining the behaviour of different formulations of Hawkes processes, i.e. linear, self-exciting, mutually exciting and nonlinear Hawkes processes.

With these tools at our disposal, we start working on the problem of comparing SSNs with Hawkes processes. We begin by considering the smaller problem of comparing linear rate models with linear Hawkes processes. This analysis is then extended to the nonlinear regime in two ways. First, the SSN is linearized around its stable point, and the resulting linearly approximated network is compared with linear Hawkes. Second, making a direct comparison of the SSN with nonlinear Hawkes processes. All these comparisons are made first for SSN models (or linear rate models) with a fixed external input and then for different external inputs, which results in different stable points for the network.

Chapter 2

Theory/Background Information

2.1 Stabilized Supralinear Network

A Stabilized Supralinear Network (SSN) is a network model with excitatory and inhibitory units where the activity of the excitatory units is "stabilized" by the inhibitory units. Each individual unit has a "supralinear", power-law input-output function. The network can be described by the following equation :

$$\tau_i \frac{\mathrm{d}V_i}{\mathrm{d}t} = -V_i + V_{rest} + h_i + \eta_i(t) + \sum_j W_{ij} r(V_j) \tag{2.1}$$

$$r(V_j) = \alpha([V_j - V_{rest}]_+)^n \tag{2.2}$$

where V_i, the voltage of neuron i, is a function of time, V_{rest} is the resting voltage, h_i is the external input current received by neuron i and $\eta_i(t)$ is the noise in the system at time t affecting neuron i. τ_i is a time constant which scales the dV_i value at every time instant when simulating the SSN iteratively. Hence, τ_i's only role is to determine the speed with which neuron i reaches a steady state. Smaller τ_i values lead to faster stabilization. Similarly, larger τ_i values result in slower stabilization.
$r(V)$ is the supralinear, power-law output of the internal voltage V. It represents the rate of spiking of neuron i at V_i voltage. Fig 2.1 shows the rectified nonlinear input-output function, for $n = 2$, $\alpha = 0.3$ and $V_{rest} = -70$ mV. We will be working with this rectified quadratic function ($r(V) = 0.3([V + 70]_+)^2$) for all further simulations and examples of SSNs.

Fig 2.2 shows the behaviour of an SSN with two neurons, one excitatory and one inhibitory, for a fixed external input ($h = 5$mV). The excitatory/inhibitory behaviour is incorporated into the equation with the help of the weight matrix W, where W_{ij} is the weight of the connection from

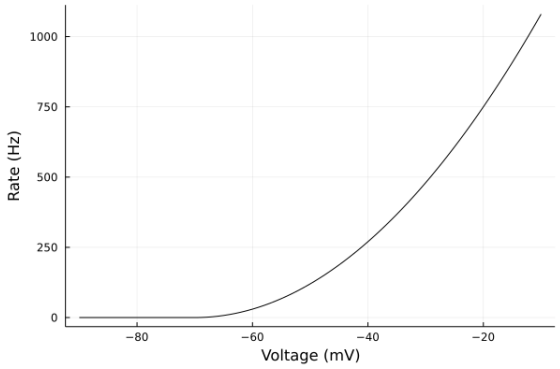


FIGURE 2.1: SSN I/O function, $r = 0.3(v - (v_{rest}))^2$

neuron j to neuron i. The SSN equation in matrix form helps to visualize this better :

$$\begin{bmatrix} \tau_E & 0 \\ 0 & \tau_I \end{bmatrix} \begin{bmatrix} \frac{dV_E}{dt} \\ \frac{dV_I}{dt} \end{bmatrix} = \begin{bmatrix} V_E \\ V_I \end{bmatrix} + \begin{bmatrix} V_{rest} \\ V_{rest} \end{bmatrix} + \begin{bmatrix} h_E \\ h_I \end{bmatrix} + \begin{bmatrix} W_{EE} & W_{EI} \\ W_{IE} & W_{II} \end{bmatrix} \begin{bmatrix} r(V_E) \\ r(V_I) \end{bmatrix} \tag{2.3}$$

The values of the variables used is as follows: $\tau_E = 20.0$ ms, $\tau_I = 10.0$ ms, $V_{rest} = -70$ mV, $h_E = h_I = 5$ mV, $W_{EE} = 1.25$, $W_{EI} = -0.65$, $W_{IE} = 1.2$, $W_{II} = -0.5$, $dt = 1$ ms. We don't take noise into consideration for now.

In the beginning of the simulation, both neurons are at resting potential. As the simulation progresses, the voltages reach their respective stable points : $V_{Estable} = -64.69$ mV, $V_{Istable} = -62.75$ mV. Since rate is a function of voltage, the rates also reach their respective stable points : $r_{Estable} = 8.45$ Hz, $r_{Istable} = 15.78$ Hz.

Fig 2.3 shows the behaviour of the same network for different external inputs. The graph shows that SSNs behave differently for smaller and larger h values. For smaller h values, the E curve has a supralinear trend which becomes sublinear for larger h values. This is a result of the inhibition stabilization in an SSN, caused by the inhibitory neuron which counters the effect of increasing h on the excitatory neuron. (Explained in greater detail in [ARM13])

The above analysis shows that Equations 2.1 and 2.2 take the network to a stable point. Once it reaches the stable point, there is no further change in the variables defining the network. We cannot extract further information about the other useful properties of a neuronal network, like the spike trains, input-dependent variability and the Fano Factor, from this limited study. We need to add more tools into our toolbox.

One way to generate spikes is by directly using the steady-state rate found from the above analysis. The spiking probability at a time instant t, will be directly proportional to the instantaneous rate at t. For this simulation, we first add an external noise $\eta_i(t)$ into the system to have control over the variability of the rates and the spike trains. [Hen+18]

(A) Voltage vs Time (B) Rate vs Time

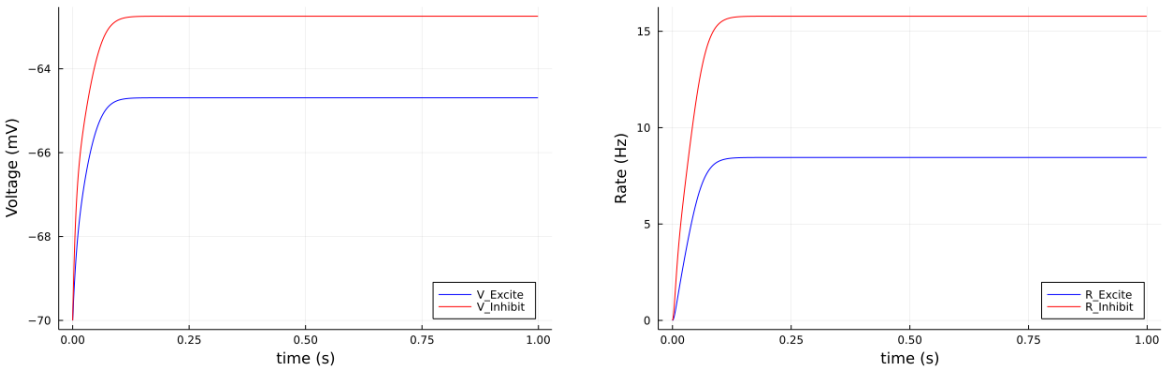


FIGURE 2.2: SSN Voltage and Rate for a fixed external input
The figure shows the behaviour of an SSN with two neurons, one excitatory and one inhibitory at h = 5 mV. In the beginning of the simulation, both neurons are at resting potential (V_{rest}). As the simulation progresses, the voltages reach their respective stable points : $V_{Estable} = -64.69$ mV, $V_{Istable} = -62.75$ mV. Since rate is a function of voltage, the rates also reach their respective stable points : $r_{Estable} = 8.45$ Hz, $r_{Istable} = 15.78$ Hz. This analysis shows that the dynamics described by Equations 2.1 and 2.2 take the network to a stable point.

(A) Voltage vs external input (B) Rate vs external input

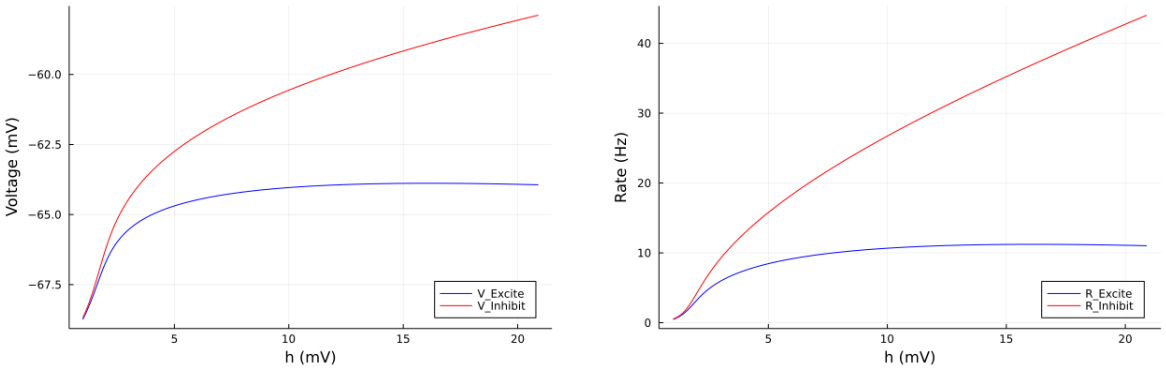


FIGURE 2.3: SSN Voltage and Rate across different external inputs
The graphs plot the stable points of a 2D SSN for different h values. It show that SSNs behave differently for smaller and larger h. For smaller h values, the E curve has a supralinear trend which becomes sublinear for larger h values. This is a result of the inhibition stabilization in an SSN, caused by the inhibitory neuron which counters the effect of increasing h on the excitatory neuron. (Explained in greater detail in [ARM13])

Noise is modelled as a multivariate Ornstein-Uhlenbeck process :

$$\tau_{noise} d\eta = -\eta dt + \sqrt{2\tau_{noise}\Sigma^{noise}} d\xi \tag{2.4}$$

$$\langle \eta_i(t)\eta_j(t+\tau)\rangle_t = \Sigma^{noise}_{ij} e^{-|\tau|/\tau_{noise}} \tag{2.5}$$

where $d\xi$ is a collection of N independent Wiener processes, and Σ^{noise} is an NxN input covariance matrix. A standard Wiener process (often called Brownian Motion) on the interval $[0, T]$ is a

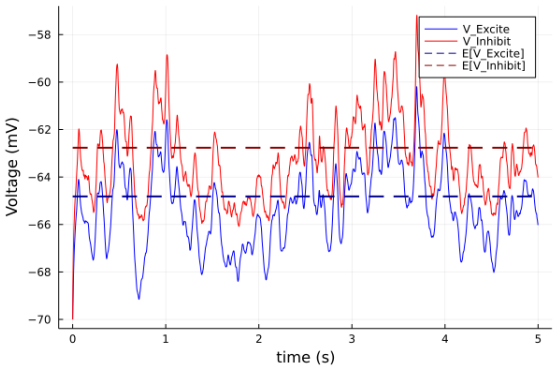
random variable $W(t)$ that depends continuously on $t \in [0, T]$ and satisfies the following :

$$W(0) = 0; \qquad dW \sim \sqrt{dt}N(0, 1) \tag{2.6}$$

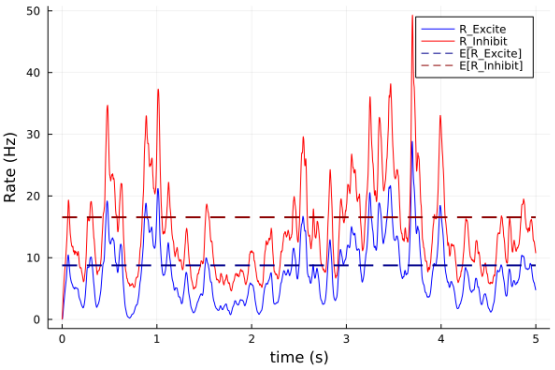
where N(0,1) is a normal distribution with mean = 0 and variance = 1.

Fig 2.4 generates spikes from an SSN with $spiking\_probability_i(t) = rate_i(t) * dt$. The SSN used is the same network from Fig 2.2 at $h = 5$ mV, with an additional noise term modelled with variable values : $mean_{wiener} = 0$, $variance_{wiener} = 0.5$, $\tau_{noise} = 50.0$ ms. The input covariance was chosen to be a diagonal matrix with values : $\Sigma^{noise} = [\alpha h, 0.0; 0.0, \alpha h]$, implying that both the neurons receive uncorrelated noise.
We observe that the mean values for voltage and rate converge to steady-state values obtained from the model without noise, thereby providing a sanity check. ($\mathbb{E}[v\_excite] = -64.818$ mV, $\mathbb{E}[v\_inhibit] = -62.766$ mV, $\mathbb{E}[r\_excite] = 8.658$ Hz, $\mathbb{E}[r\_excite] = 16.517$ Hz)

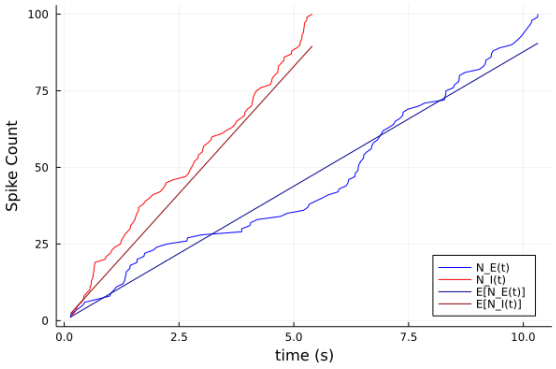
(A) Voltage vs Time



(B) Rate vs Time



(C) SSN Spike Count



(D) Visualizing spiking in SSN using Raster Plot

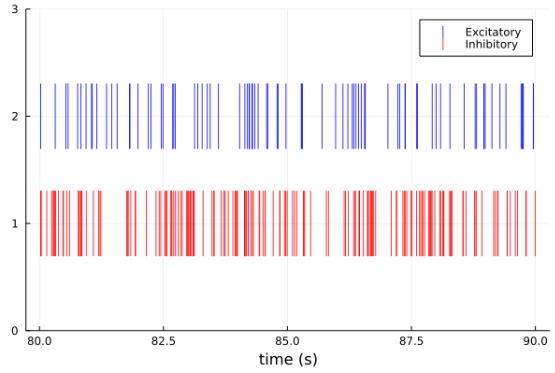


FIGURE 2.4: Generating spikes from an SSN with external noise
The SSN used is the same network from Fig 2.2 at $h = 5$ mV, with an additional uncorrelated noise term. Mean values for voltage and rate converge to steady-state values obtained from the model without noise, thereby providing a sanity check. ($\mathbb{E}[v\_excite] = -64.818$ mV, $\mathbb{E}[v\_inhibit] = -62.766$ mV, $\mathbb{E}[r\_excite] = 8.658$ Hz, $\mathbb{E}[r\_excite] = 16.517$ Hz)
(C) plots the total number of spikes generated so far at each time instant t. (D) shows a raster plot to help visualize spiking. A spike at time instant t is represented by a vertical line at the x value t.

2.2 Spiking Stabilized Supralinear Network

Another possible upgrade to our vanilla SSN equation is feeding the spike trains back into the equation [Hen+18]. A neuron i emits spikes stochastically with an instantaneous probability equal to $rate_i(t) * dt$. Presynaptic spikes are filtered by synaptic dynamics into exponentially decaying postsynaptic currents (E or I):

$$\frac{\mathrm{d}a_j}{\mathrm{d}t} = \frac{-a_j}{\tau_{syn}} + \sum_{t_j} \delta(t - t_j) \tag{2.7}$$

where the t_j's are the firing times of neuron j, $\tau_{syn} = 2$ ms is the synaptic time constant. Synaptic currents then contribute to membrane potential dynamics according to :

$$\tau_i \frac{\mathrm{d}V_i}{\mathrm{d}t} = -V_i + V_{rest} + h_i(t) + \eta_i(t) + \sum_j J_{ij} a_j(t) \tag{2.8}$$

$$r(V_j) = \alpha([V_j - V_{rest}]_+)^n \tag{2.9}$$

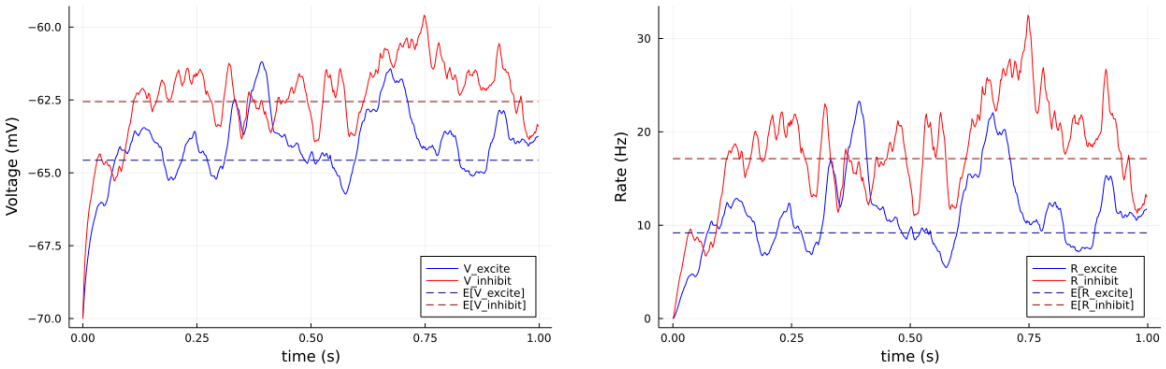
where the synaptic efficacies J_{ij} are described below, and the noise term $\eta_i(t)$ is modelled as a multivariate Ornstein-Uhlenbeck process.
We simulate a spiking SSN with $N_E = 4000$ excitatory neurons and $N_I = 1000$ inhibitory neurons. Each neuron i has $p_E N_E$ excitatory and $p_I N_I$ inhibitory presynaptic partners, assigned uniformly at random. Connection probabilities were set to $p_E = 0.1$ and $p_I = 0.4$ respectively. The corresponding synaptic weights took on values $J_{ij} = W_{\alpha\beta}/\tau_{syn} p_\beta N_\beta$ where $\{\alpha, \beta\} \in \{E, I\}$ denote the populations to which neuron i and j belong respectively, and $W_{\alpha\beta}$ are the connections in the reduced two dimensional model from Fig 2.2 : $W_{EE} = 1.25$, $W_{EI} = -0.65$, $W_{IE} = 1.2$, $W_{II} = -0.5$. This choice is such that, for a given set of mean firing rates in the E and I populations, average E and I synaptic inputs to E and I cells matched the corresponding recurrent inputs in the rate-based model (Fig 2.2). Synapses that were not drawn were set to $J_{ij} = 0$.

Voltages of each individual neuron fluctuate around the population means $\mathbb{E}[V\_excite] = -64.568$ mV and $\mathbb{E}[V\_inhibit] = -62.558$ mV. Similarly, rates of each individual neuron fluctuate around the population means $\mathbb{E}[r\_excite] = 9.184$ Hz and $\mathbb{E}[r\_inhibit] = 17.129$ Hz. Fig 2.5 shows this behaviour.

The population means match the steady-state values for voltages and rates of an SSN with no noise (Fig 2.2).

(A) Voltage vs Time

(B) Rate vs Time



(C) spiking SSN spike count

(D) Visualizing spiking in spiking SSN using raster plot

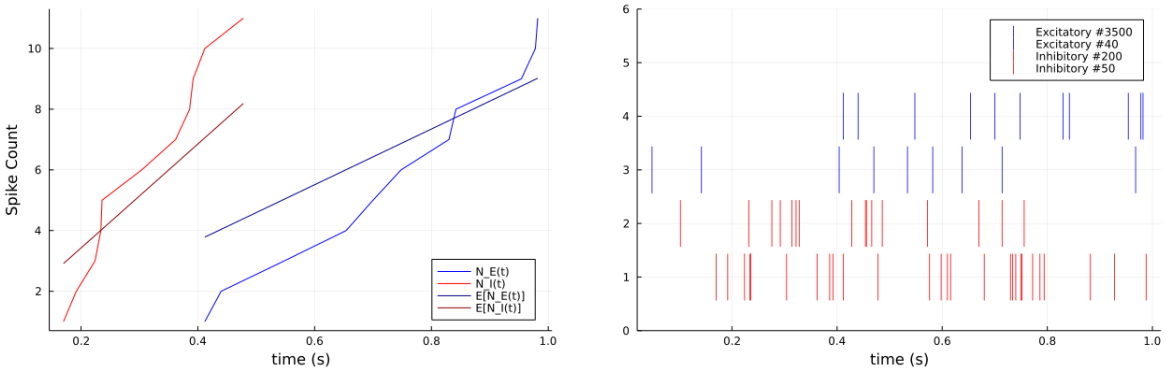


FIGURE 2.5: Generating spikes from a spiking SSN
We simulated a spiking SSN with $N_E = 4000$ excitatory neurons and $N_I = 1000$ inhibitory neurons. Each neuron i has $p_E N_E$ excitatory and $p_I N_I$ inhibitory presynaptic partners, assigned uniformly at random. The corresponding synaptic weights are chosen such that, for a given set of mean firing rates in the E and I populations, average E and I synaptic inputs to E and I cells matched the corresponding recurrent inputs in the rate-based model (Fig 2.2). (A) and (B) show the fluctuating behaviour of randomly chosen E and I neurons around the population means. (C) plots the total number of spikes generated so far at each time instant t. Since this is a heavy simulation, the graph shows a limited number of spikes, till $t = 1$ s. Speed of the simulation needs to be improved to be able to collect more data. (D) shows a raster plot to help visualize spiking.

2.3 Motivation for Hawkes

Using SSNs to model biological networks of neurons poses a number of limitations on the properties of the network accessible to us. An SSN has no intrinsic noise. So noise has to be added externally into the system to generate a more realistic model.

We cannot extract spike trains from the original SSN model. Additional mechanisms need to be included to the SSN to generate spiking.

To our knowledge, there is no analytic way to compute variability in SSNs. The linear version (linear rate model) does have an analytic way to compute covariance, but not SSNs. The covariance is a useful property as the effect of Spike Time Dependent Plasticity (STDP) rules in a network of neurons depends on the covariance between spike times, and on even higher order statistics. [Gjo+11]

Most theoretical studies on SSNs are based on mean-field approaches, where the network is reduced to two dimensions. Higher dimensional models are mostly used in simulations, and to our knowledge, an in-depth analysis of the same is not available.

Linear Hawkes processes compensate for all the above stated problems of SSNs. Also, significant amount of work has been published in the domain of Hawkes processes, making it easier to expand their existing applications in network models. Hawkes processes

- are stochastic processes, and hence have intrinsic noise.
- are point processes (defined in the next section). Each point in a Hawkes process can be easily interpreted as a spike in a neuron.
- It is possible to compute covariance [Haw71] and higher order statistics [JHR15] for Hawkes processes analytically.
- A large number of mutually exciting Hawkes processes can be easily modelled, and have extensive supporting theoretical studies.

Hence, Hawkes processes are a good candidate to replicate the behaviour of SSNs, while providing further insights and advantages. Studies have been conducted on the same by [Ock+17b]. We will be replicating some of these works and making the comparison more straightforwardly, thereby creating a preliminary resource for further investigations with the aim of using Hawkes processes as an alternative to Stabilized Supralinear Networks in STDP studies.

2.4 Hawkes Processes

Hawkes processes are stochastic point processes. They capture probabilistic event occurrences repeated in time. An important feature of Hawkes Processes is that they are self-exciting, which means that each arrival increases the rate of future arrivals for some period of time. A detailed explanation of Hawkes processes follows (definitions adapted from [LTP15]) :

Simple Point Process : A sequence of random variables $T = t_1, t_2, t_3, ...$, where $t_i \in [0, \infty)$, $0 \leq t_1 \leq t_2 \leq t_3...$ and the number of points in a bounded region is almost surely finite.

Count Process : A stochastic process $(N(t) : t \geq 0)$ that takes values in $\mathbb{N}$ and satisfies $N(0) = 0$, is almost surely finite, and is a right-continuous step function with increments of size $+1$.

A count process can be viewed as a cumulative count of the number of occurrences of an event, where the sequence of random occurrence times $(T = t_1, t_2, t_3, ...)$ is a point process. A point process (and its corresponding count process) is easy to describe and work with using its conditional intensity function (which is just a fancy word for rate of occurrences).

Conditional Intensity Function : Consider a counting process $N(.)$ with an associated history $H(t)$, its conditional intensity function :

$$\lambda^*(t) = \lim h \to 0 \frac{\mathbb{E}[N(t+h) - N(t)|H(t)]}{h} \tag{2.10}$$

is a non-negative function which relies only on information of $N(.)$ from the past.

Linear Hawkes Process : Consider a counting process $(N(t) : t \geq 0)$, with associated history $(H(t) : t \geq 0)$, that satisfies

$$\mathbb{P}(N(t+dt) - N(t) = m|H(t)) = \begin{cases} \lambda^*(t) + o(dt) & , m = 1 \\ o(dt) & , m > 1 \\ 1 - \lambda^*(t) + o(dt) & , m = 0 \end{cases}$$

where $o(dt)$ is negligible for small dt values. In our implementation of Hawkes Processes, we will be assuming dt to be small enough to have $0 \leq m \leq 1$. In other words, at each time step an event will occur either 0 or 1 time only.
Let the process' conditional intensity function be :

$$\lambda^*(t) = \lambda + \int_0^t \mu(t-u) dN(u) \tag{2.11}$$

where $\lambda > 0$ is the background intensity and $\mu : (0, \infty) \to [0, \infty)$ is the excitation function. Let $\mu(.) \neq 0$ to avoid the trivial case, that is, a homogeneous Poisson process. Such a process $N(.)$ is

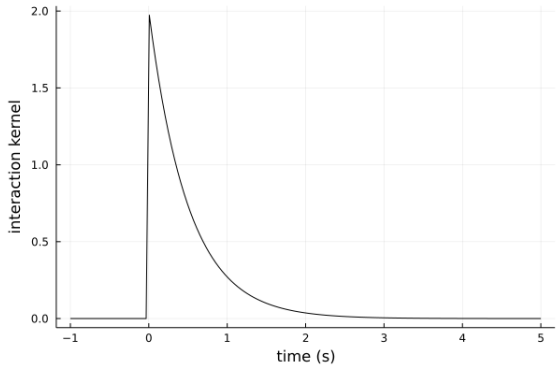
a linear Hawkes process. The point process corresponding to a conditional intensity function is generated using the "thinning algorithm" [LTP15].

Fig 2.6 simulates a self exciting linear Hawkes Process. (A) shows the excitation function which is taken to be an exponentially decaying function multiplied with a weight factor (w = 0.85).

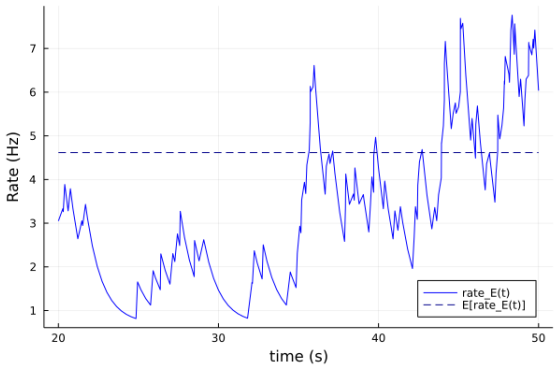
$$\mu(t) = \frac{1}{\tau} e^{\left(\frac{-t}{\tau}\right)} \implies \int_0^t \mu(t-u)dN(u) = \sum_{t_x} \mu(t-t_x) = \sum_{t_x} \frac{e^{(t_x-t)/\tau}}{\tau}$$

with τ = 1.0 s. (B) shows the conditional intensity function λ^*, with background intensity λ = 0.7 Hz. $\mathbb{E}[\lambda^*]$ was found to be 4.616 Hz. (C) visualizes the point process using a raster plot, where a spike at time instant t is represented by a vertical line at the x value t. The corresponding Count process is shown in (D) as an increasing step function.

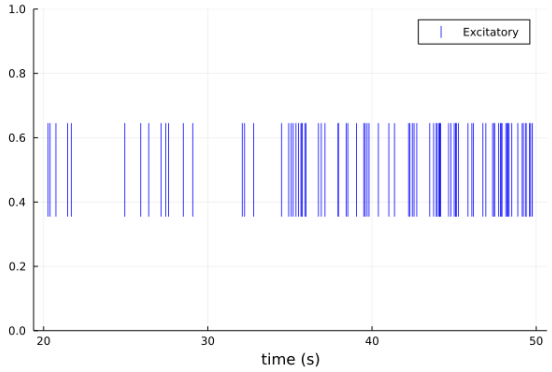
(A) Hawkes interaction kernel



(B) Conditional Intensity Function



(C) Point Process



(D) Count Process

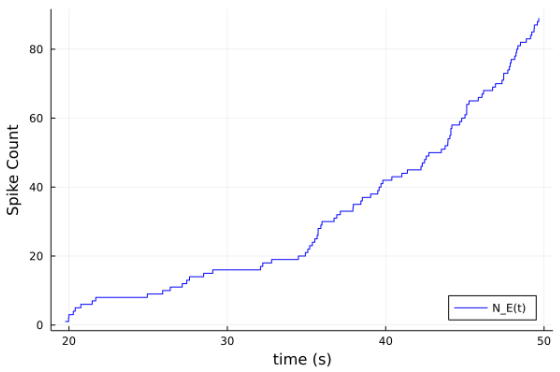


FIGURE 2.6: Simulating a Self Exciting, Linear Hawkes Process
(A) shows the excitation function. It is an exponentially decaying function ($\sum_x \frac{e^{(t_x-t)/\tau}}{\tau}$, with τ = 1.0) multiplied with the weight of the connection of the process with itself (w = 0.85). (B) shows the conditional intensity function λ^*, with background intensity λ = 0.7 Hz. $\mathbb{E}[\lambda^*]$ was found to be 4.616 Hz. (C) visualizes the point process using a raster plot, where a spike at time instant t is represented by a vertical line at the x value t. The corresponding Count process is shown in (D) as an increasing step function. Total number of spikes generated for the simulation = 90,000

The definition of Hawkes Processes can also be extended to model Mutually Exciting Point Processes.

Mutually Exciting Hawkes Processes : Consider a collection of m counting processes $N_1(.), ..., N_m(.)$ denoted $\mathbf{N}$. Say $t_{i,x} : i \in 1, ..., m, x \in \mathbb{N}$ are the observed random arrival times for each counting process. If for each $i = 1, ..., m$ then $N_i(.)$ has conditional intensity of the form :

$$\lambda_i^*(t) = \lambda_i + \sum_{j=1}^{m} \int_0^t \mu_j(t-u) dN_j(u) \tag{2.12}$$

for some $\lambda_i > 0$ and $\mu_i : (0, \infty) \to [0, \infty)$, then $\mathbf{N}$ is called a Mutually Exciting Hawkes Processes.

Setting the excitation function as exponentially decaying, we get :

$$\lambda_i^*(t) = \lambda_i + \sum_j \sum_x W_{ij} \frac{e^{(t_{jx}-t)/\tau_j}}{\tau_j}$$

In the succeeding sections, we will be comparing Hawkes Processes with SSNs. The event under consideration will be the spiking of a neuron. The point process, $t_{i,x} : i \in 1, ..., m, x \in \mathbb{N}$ will be the time of occurrence of the x^{th} spike of the i^{th} neuron. Its corresponding count process, $N_i(t)$ will be the spike count of neuron i at time t. Changing the variable names to assist the comparison. The conditional intensity function $\lambda_i^*(t)$ is renamed as $r_i^{hawkes}(t)$, which stands for the rate of spiking. The background intensity λ_i is renamed as h_i^{hawkes}, to represent the baseline rate, present because of non-zero external input to neuron i.

The excitation function μ is the product of a weight factor, W_{ij} with the interaction kernel $\sum_x \frac{e^{(t_{jx}-t)/\tau_j}}{\tau_j}$. This Interaction Kernel takes all the spike times of neuron j as input (t_{jx}), applies an exponentially decaying function on them and returns the sum of all these factors. The excitation function can be interpreted as the cumulative effect of all spikes of neuron j on neuron i. So the equation we will be using for a Hawkes Process is as follows :

$$r_i^{hawkes}(t) = h_i^{hawkes} + \sum_j W_{ij}^{hawkes} \sum_x \frac{e^{(t_{jx}-t)/\tau_j}}{\tau_j} \tag{2.13}$$

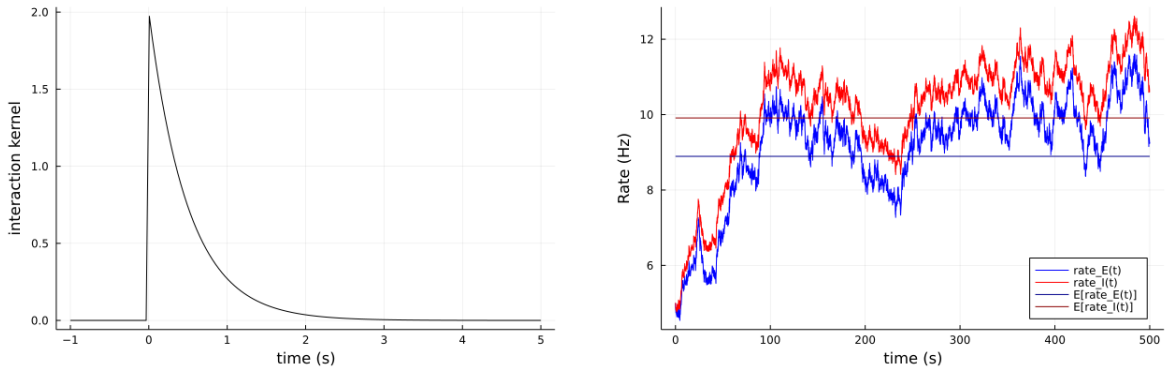
Fig 2.7 simulates two mutually exciting Hawkes processes. Let E and I be the two point processes under consideration. Values of variables used : $\tau_E = 20.0$ s; $\tau_I = 10.0$ s; $h_E = h_I = 5.0$ Hz. The simulation was run up till a total of 500,000 spikes were generated by the model. The mean rate of spiking of the two processes obtained : $\mathbb{E}[r_E(t)] = 8.892$ Hz and $\mathbb{E}[r_I(t)] = 9.911$ Hz. $W_{EE} = 1.25$, $W_{EI} = -0.65$, $W_{IE} = 1.2$, $W_{II} = -0.5$. The negative weights (W_{EI} and W_{II}) imply that process I had an inhibitory effect on both the processes.

Another important variation of the Hawkes process is created by adding a nonlinearity to the conditional intensity function.

Nonlinear Hawkes Process : Consider a counting process $N(.)$ with conditional intensity

(A) Hawkes interaction kernel

(B) Linear Hawkes Instantaneous Rate



(C) Linear Hawkes Spike Count

(D) Visualizing spiking using a Raster Plot

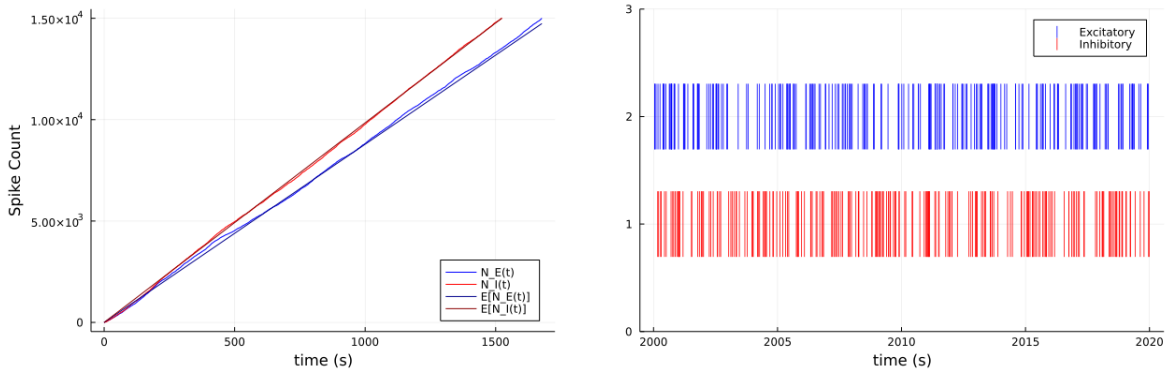


FIGURE 2.7: Simulating Two Mutually Exciting, Linear Hawkes Processes
(A) shows the excitation function, which is same as the self exciting 1D case. However, different time constants will be used : $\tau_E = 20.0$ s; $\tau_I = 10.0$ s. (B) shows the conditional intensity function λ^*, with $h_E = h_I = 5.0$ Hz. $\mathbb{E}[r_E] = 8.892$ Hz, $\mathbb{E}[r_I] = 9.911$ Hz. (C) visualizes the point processes using a raster plot, where a spike at time instant t is represented by a vertical line at the x value t. The corresponding Count processes are shown in (D). Total number of spikes generated = 500,000

function of the form :

$$r = \phi(h^{hawkes} + \sum_j W_j^{hawkes} \sum_i \frac{e^{(t_{ij}-t)/\tau_j^{hawkes}}}{\tau_j^{hawkes}}) \tag{2.14}$$

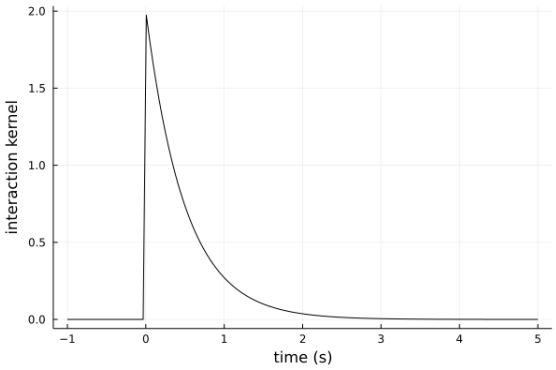
where $\phi : \mathbb{R} \to [0, \infty)$. Then $N(.)$ is a mutually exciting nonlinear Hawkes process. Selecting $\psi(x) = x$ reduces $N(.)$ to a linear Hawkes process. We will be taking $\phi(x) = x^2$.

$$r = (h^{hawkes} + \sum_j W_j^{hawkes} \sum_i \frac{e^{(t_{ij}-t)/\tau_j^{hawkes}}}{\tau_j^{hawkes}})^2 \tag{2.15}$$

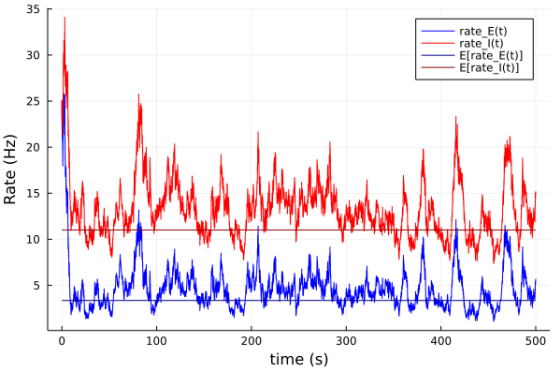
Fig 2.8 simulates two mutually nonlinear Hawkes processes using Eq. 2.15. Let E and I be the two point processes under consideration. Values of variables used : $\tau_E = 20.0$ ms; $\tau_E = 10.0$ ms; $h_E = h_I = 5.0$ Hz. Total number of spikes for which the simulation was run $(n\_spikes) =$ 500,000. The mean rate of spiking of the two processes obtained : $\mathbb{E}[r_E(t)] = 3.592$ Hz and $\mathbb{E}[r_I(t)] = 11.654$ Hz. $W_{EE} = 1.25$, $W_{EI} = -0.65$, $W_{IE} = 1.2$, $W_{II} = -0.5$. The negative

weights (W_{EI} and W_{II}) imply that process I had an inhibitory effect on both the processes.

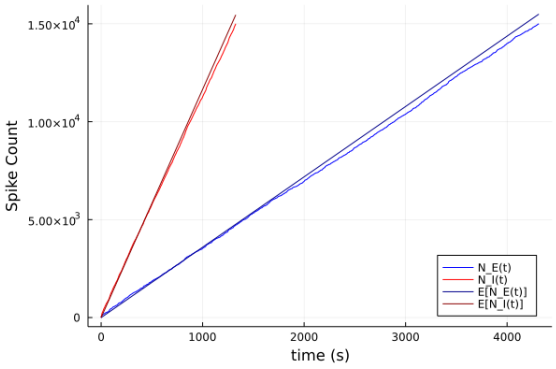
(A) Hawkes interaction kernel



(B) Non Linear Hawkes Instantaneous Rate



(C) Non Linear Hawkes Spike Count



(D) Visualizing spiking using a Raster Plot

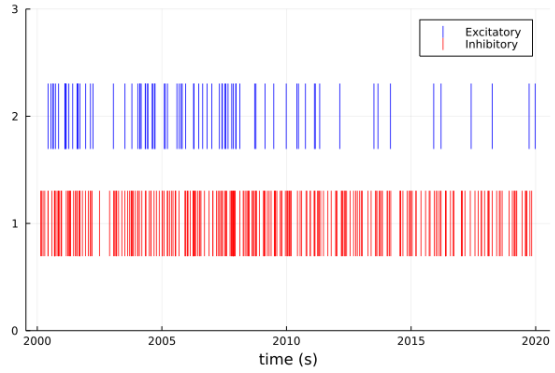


FIGURE 2.8: Simulating Two Mutually Exciting, Non Linear Hawkes Processes
This shows an interesting difference in the behaviour of linear and nonlinear Hawkes processes. The inhibitory process, i.e. the process with negative weights, generates spikes at a much larger rate than the excitatory process, i.e. process with positive weights. Values of variables used : $\tau_E = 20.0$ ms; $\tau_E = 10.0$ ms; $h_E = h_I = 5.0$ Hz. Total number of spikes for which the simulation was run $(n\_spikes) = 500{,}000$. The mean rate of spiking of the two processes obtained : $\mathbb{E}[r_E(t)] = 3.592$ Hz and $\mathbb{E}[r_I(t)] = 11.654$ Hz.
The interaction kernel is the same as the linear case, since the nonlinearity is added at a later stage.

Chapter 3

Comparisons

3.1 Comparing the Linear Rate Model with Linear Hawkes Processes

We first work on a simplified linear version of the problem.
To consider the linear rate model, we simply assign $n = 1$ in Eq 2.2, giving :

$$\tau_i \frac{\mathrm{d}V_i}{\mathrm{d}t} = -V_i + V_{rest} + h_i + \sum_j W_{ij} r(V_j) \tag{3.1}$$

$$r(V_j) = \alpha([V_j - V_{rest}]_+) \tag{3.2}$$

The input-output function in this case is a Rectified Linear Unit (ReLU) with output $rate = 0$ for input voltage below the resting potential (Fig 3.1).
Fig 3.2 shows the behaviour of a 2D linear rate model for a fixed external input (h = 5 mV), confirming that Eqs 3.1 and 3.2 takes the system to a stable point. For the comparison of the first property, i.e. rate, we will be using the steady state equation. Values of variables used : α = 0.3, $W = [1.25 - 0.65; 1.2 - 0.5]$, $\tau_E^{ratemodel} = 20.0$, $\tau_I^{ratemodel} = 10.0$. The stable points for voltages and rates are : $v\_excite = -63.948$ mV; $v\_inhibit = -63.758$ mV, $r\_excite = 1.816$ Hz and $r\_inhibit = 1.873$ Hz.
Fig 3.3 highlights the important distinction between an SSN and a linear rate model by showing the linear behaviour of Eqs 3.1 and 3.2 across different external inputs ($h\_max = 20.0$; $h\_min = 1.0$; $step = 1$).

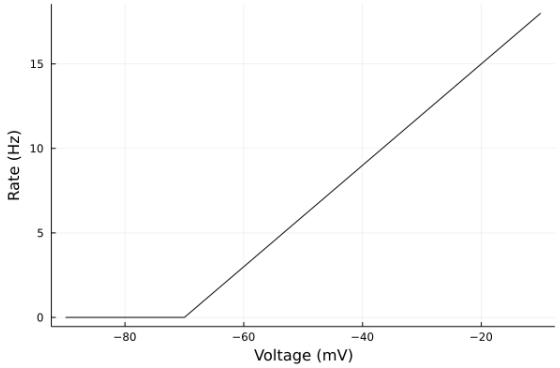


FIGURE 3.1: Linear Rate Model I/O Function $r = 0.3(v - v_{rest})$

(A) Voltage vs Time

(B) Rate vs Time

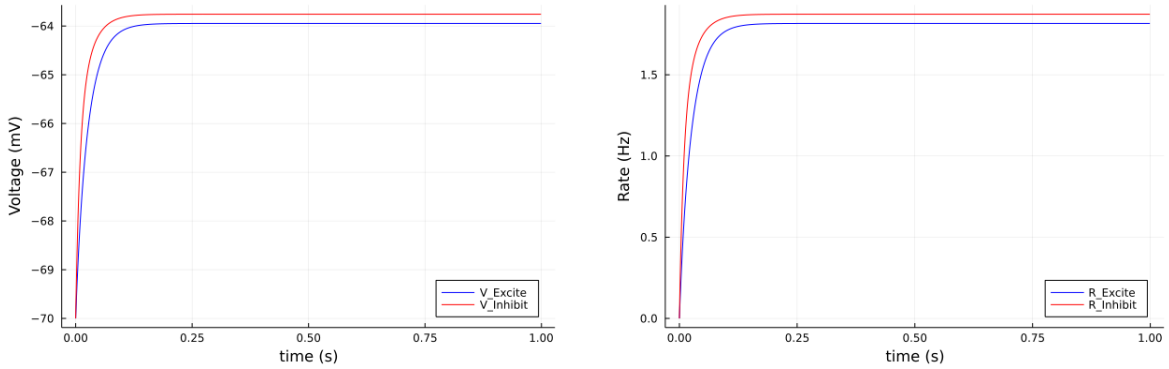


FIGURE 3.2: Linear Rate Model Voltages and Rates for a fixed external input, $h = 5mV$
This simulation confirms that Eqs 3.1 and 3.2 takes the system to a stable point. The stable points for voltages and rates are : $v\_excite = -63.948$ mV; $v\_inhibit = -63.758$ mV, $r\_excite = 1.816$ Hz and $r\_inhibit = 1.873$ Hz.

(A) Voltage vs h

(B) Rate vs h

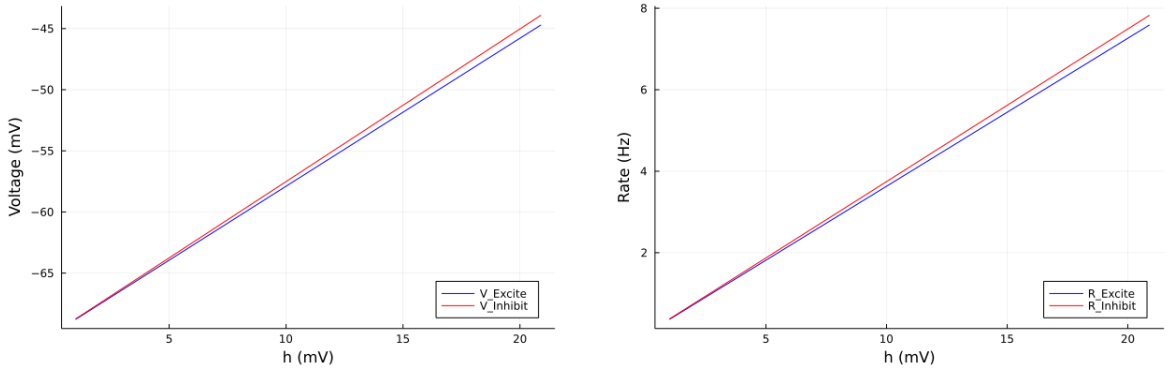


FIGURE 3.3: Linear Rate Model Voltages and Rates across different external input
This simulation highlights the important distinction between an SSN and a linear rate model by showing the "linear" behaviour of Eqs 3.1 and 3.2 across different external inputs ($h\_max = 20.0$; $h\_min = 1.0$; $step = 1$).

3.1.1 Comparing Rates

Our goal is to be able to use a linear Hawkes process as an alternative to the linear rate model, to get the spike trains and properties associated to the rate and spiking of the network of neurons. Hence, the **steady state of the RATE** of the linear rate model is going to be taken into consideration, and not the voltage. From the input/output equation, assuming $V \geq V_{rest}$ we have :

$$\frac{\mathrm{d}r_j}{\mathrm{d}t} = \alpha \frac{\mathrm{d}V_j}{\mathrm{d}t} \tag{3.3}$$

Changing the variable V_i to $r_i (= \alpha(V_i - V_{rest}))$:

$$\begin{aligned} \tau_i \frac{\mathrm{d}r_i}{\mathrm{d}t} &= \alpha(-V_i + V_{rest} + h_i + \sum_j W_{ij} r_j) \\ &= \alpha(-V_i + V_{rest}) + \alpha h_i + \sum_j \alpha W_{ij} r_j \\ &= -r_i + \alpha h_i + \sum_j \alpha W_{ij} r_j \end{aligned} \tag{3.4}$$

At the Steady State, $\frac{\mathrm{d}r_i}{\mathrm{d}t} = 0$. So,

$$r_i^{ss} = \alpha h_i + \sum_j \alpha W_{ij} r_j^{ss} \tag{3.5}$$

Comparing this steady state with Linear Hawkes :

$$r_i^{hawkes}(t) = h_i^{hawkes} + \sum_j W_{ij}^{hawkes} \sum_x \frac{e^{(t_{jx} - t)/\tau_j}}{\tau_j} \tag{3.6}$$

$$h_i^{hawkes} = \alpha h_i \qquad W_{ij}^{hawkes} = \alpha W_{ij} \tag{3.7}$$

This suggests that using αh as h^{hawkes} and αW as W^{hawkes} we obtain a Hawkes process with mean rate equal to the steady state rate of the corresponding linear rate model, and could potentially be used to simulate a network with the same response as the linear rate model, while giving us the additional benefits of a Hawkes Process.

Fig 3.4 compares the steady state rate of a 2D linear rate model with the corresponding mean rate of the 2D linear Hawkes Process. Values of the variables used : h = 5mV, $\alpha = 0.3$, $W = [1.25 - 0.65; 1.2 - 0.5]$, $\tau_E^{ratemodel} = 20.0$ ms, $\tau_I^{ratemodel} = 10.0$ ms, $h_{hawkes} = \alpha h$, $W_{hawkes} = \alpha W$, $tau_E^{hawkes} = 10.0$ s, $tau_I^{hawkes} = 5.0$ s, $n\_spikes = 500,000$. The voltages of the neurons in the linear rate model stabilize at : $v\_excite = -63.948$ mV; $v\_inhibit = -63.758$ mV. Their corresponding steady state rates are : $r\_excite = 1.816$ Hz and $r\_inhibit = 1.873$ Hz. The mean rates of the two Hawkes processes are : $r_E = 1.808$ Hz and $r_I = 1.866$ Hz. Rates of the two models match well, with an absolute error of 0.008 Hz and relative error of 0.004. Hence,

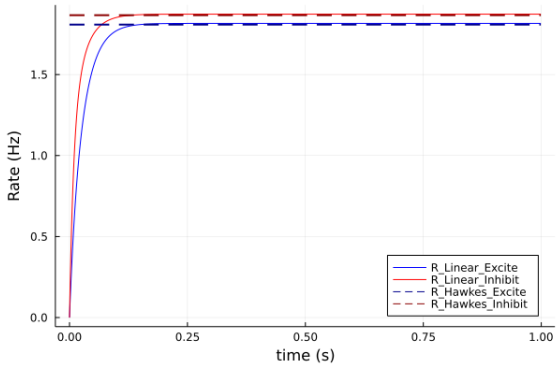


FIGURE 3.4: Comparing the steady state rate of a Linear Rate Model with the corresponding mean rate of the Linear Hawkes Process at a fixed value of external input h = 5 mV, Values of the variables used : h = 5mV, $h_{hawkes} = \alpha h$, $W_{hawkes} = \alpha W$. The voltages of the neurons in the linear rate model stabilize at : $v\_excite = -63.948$ mV; $v\_inhibit = -63.758$ mV. Their corresponding steady state rates are : $r\_excite = 1.816$ Hz and $r\_inhibit = 1.873$ Hz. The mean rates of the two Hawkes processes are : $r_E = 1.808$ Hz and $r_I = 1.866$ Hz. Hence, Rates of the two models match well.

the Hawkes processes defined above can generate spikes with an effective mean rate equal to the steady-state rate of the linear rate model!

Fig 3.5 compares the rates across different h values ($h\_max = 20.0$; $h\_min = 1.0$; $step = 1$). The rates for each steady-state in the plot of the linear model matches well with the mean rates of the two Hawkes processes, with input current = αh Hz. The absolute error has an increasing trend with increasing h, but the error values are still small enough for a wide range of h, with the maximum observed error = 0.04 Hz. This suggests that the tau^{hawkes} values chosen favour smaller h values. To investigate more, we plot the relative error and find that the relative error has no obvious dependence on h. The maximum relative error is 0.009. Hence, the linear Hawkes processes can generate spikes with mean rate equal to the steady-state rate of the linear rate model for different h values.

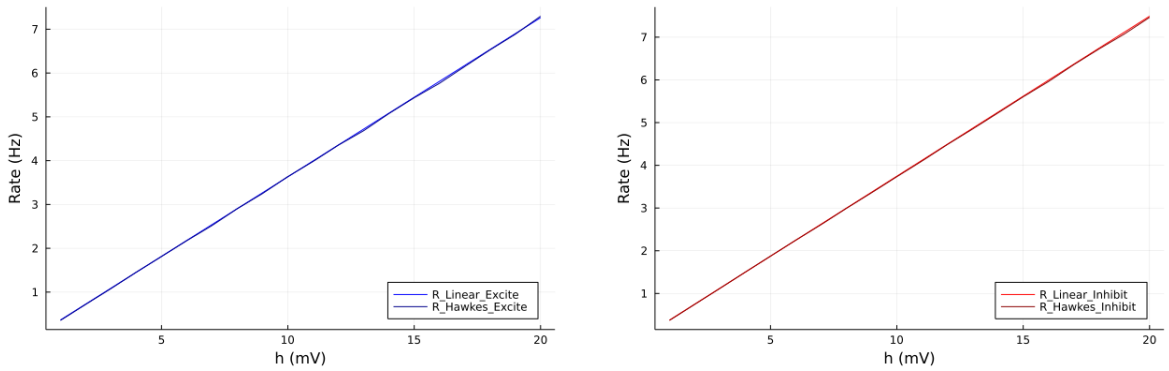
3.1.2 Comparing Variability

Another important property of a network of neurons is the variability in the spike trains. It provides useful information on the dependence of the possibility of spikings in a neuron on the past spikes of the same neuron neuron as well as the past spikes of the other neurons in the network.

We add external noise (modelled as a multivariate Ornstein-Uhlenbeck process) into the linear system to have control over variability in this model. No such external addition is necessary for Hawkes processes as they already have internal noise, and hence variability on account of being stochastic processes. We use the same 2D linear Hawkes process as described in the previous section. The 2D linear rate model has an additional noise term. Values of variables

(A) Excitatory Rate

(B) Inhibitory Rate



(C) Absolute Error

(D) Relative Error

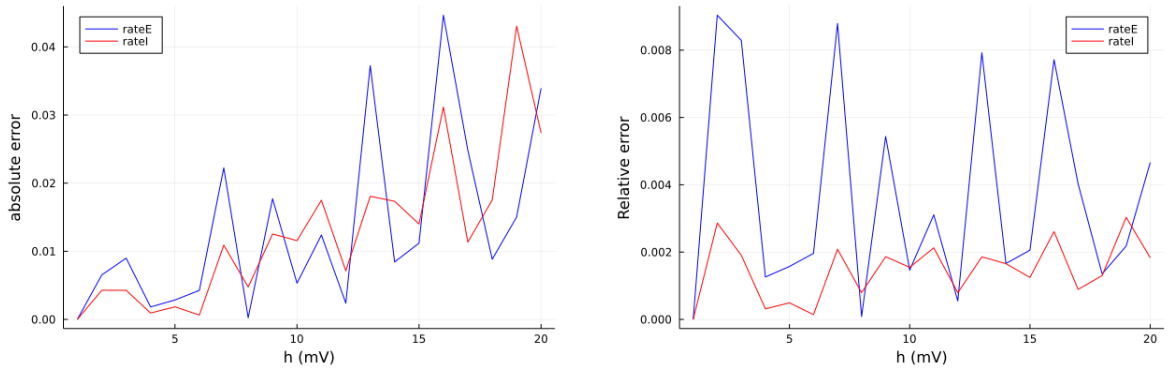


FIGURE 3.5: Comparing the steady state rate of a Linear Rate Model with corresponding mean rate of the Linear Hawkes Process across different external input values
The rates for each steady-state in the plot of the linear model match well with the mean rates of the Hawkes processes, with input current = αh Hz. Maximum absolute errror = 0.04 Hz, maximum relative error = 0.009. Hence, the linear Hawkes processes can generate spikes with mean rate equal to the steady-state rate of the linear rate model for different h values.

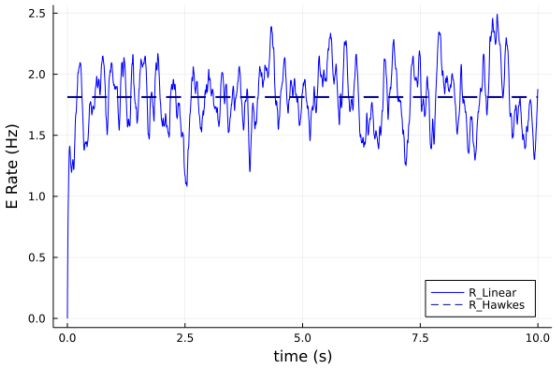
used : $h = 5$ mV, $\alpha = 0.3$, $V_{rest} = -70$ mV, $W = [1.25 - 0.65; 1.2 - 0.5]$, $\tau_E^{ratemodel} = 20.0$ ms, $\tau_I^{ratemodel} = 10.0$ ms, $h_{hawkes} = \alpha h$, $W_{hawkes} = \alpha W$, $tau_E^{hawkes} = 10.0$ s, $tau_I^{hawkes} = 5.0$ s, $n\_spikes = 500,000$.

The additional noise term is modelled with variable values : $mean_{wiener} = 0$, $variance_{wiener} = 0.6$, $\tau_{noise} = 50.0$ ms. The input covariance was chosen to be a diagonal matrix with values : $\Sigma^{noise} = [\alpha h, 0.0; 0.0, \alpha h]$, implying that both the neurons receive uncorrelated noise. Fig 3.6 shows the rates of the linear model at the steady-state with noise, over a time interval of 10 seconds. These rates are compared with the mean rates of spiking of the linear Hawkes processes : $\mathbb{E}[r_E^{ratemodel}] = 1.816$ Hz, $\mathbb{E}[r_I^{ratemodel}] = 1.874$ Hz, $\mathbb{E}[r_E^{hawkes}] = 1.813$ Hz, $\mathbb{E}[r_I^{ratemodel}] = 1.873$ Hz. There is a clear match in the rate values with an absolute error of 0.003 Hz and relative error of 0.002, giving a sanity check.

To find covariances in both the models, first the spike trains are divided into bins with equal time intervals. The rates for each of these individual bins is calculated, which is then used to calculate the covariance using the inbuilt Julia function. The covariances matrices calculated

are : $Cov^{ratemodel} = [1.820 - 0.045; -0.0451.845]$ and $Cov^{hawkes} = [1.8880.024; 0.0241.828]$. So, $Cov_{ee}^{ratemodel} = 1.820$, $Cov_{ee}^{hawkes} = 1.888$, $Cov_{ii}^{ratemodel} = 1.845$, $Cov_{ii}^{hawkes} = 1.828$, $Cov_{ei}^{ratemodel} = Cov_{ie}^{ratemodel} = -0.045$ and $Cov_{ei}^{hawkes} = Cov_{ie}^{hawkes} = 0.024$. The comparison shows that Cov_{ee} and Cov_{ii} values for the two models match well (absolute error = 0.017, relative error = 0.009). However, Cov_{ei} values don't match.

(A) Excitatory Rate vs time



(B) Inhibitory Rate vs time

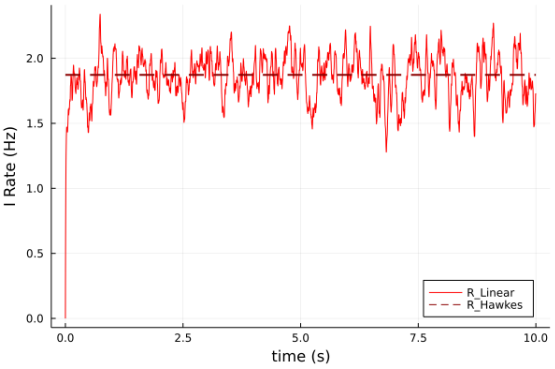


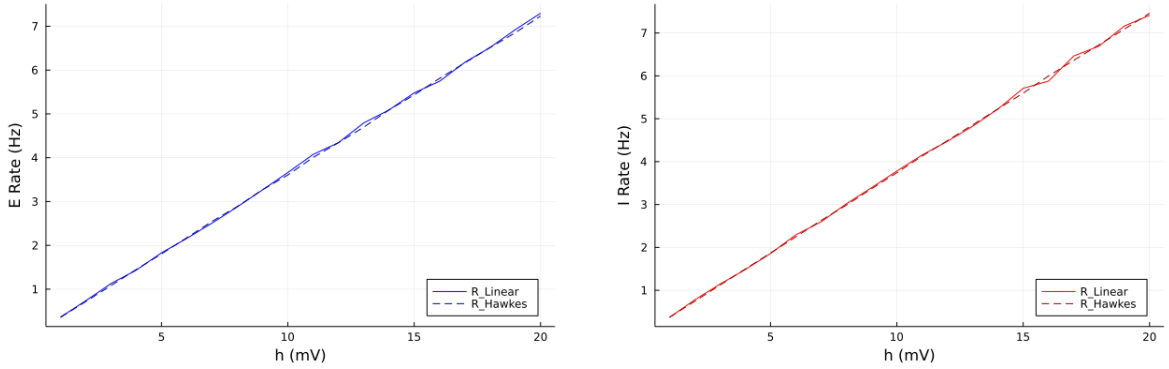
FIGURE 3.6: Linear Rate Model with noise vs Linear Hawkes
shows the rates of the two models over a time interval of 10 seconds. Comparing the two : $\mathbb{E}[r_E^{ratemodel}] = 1.816$ Hz, $\mathbb{E}[r_I^{ratemodel}] = 1.874$ Hz, $\mathbb{E}[r_E^{hawkes}] = 1.813$ Hz, $\mathbb{E}[r_I^{ratemodel}] = 1.873$ Hz. There is a clear match in the rate values with an absolute error of 0.003 Hz and relative error of 0.002.

To get a better picture, we collect more data points for this comparison by extending it across different h values. Fig 3.7 compares the mean rates of the Hawkes processes with the mean rates of the linear model with noise, for $h\_max = 20.0$; $h\_min = 1.0$; $step = 1$. It shows a good match between these values for both Excitatory and Inhibitory neuron, with maximum absolute error = 0.125 Hz and maximum relative error = 0.05. Hence, providing a sanity check.

Next, we divide the spike trains into equal time interval bins, calculate the rates for each of the individual bins and use the inbuilt Julia function to find the covariance matrices for both the models, at all h values. Fig 3.8 compares Cov_{ee} and Cov_{ii} values of the two models. The covariances match with a maximum absolute error = 0.6 and maximum relative error = 0.125. Both these covariances have a positive linear relationship with h. The fluctuations from a straight line in the linear model plots for both Cov_{ee} and Cov_{ii} could possibly be attributed to the external noise added to the system. A better analysis is possible by running simulations for each h multiple times and taking the average covariance values. This is yet to be tried, and has not been added to this report. Further investigation is needed. Although, the analysis so far provides a preliminary result, showing a good match in the covariance values Cov_{ee} and Cov_{ii} !

(A) Excitatory Rate vs external input

(B) Inhibitory Rate vs external input



(C) Absolute Error

(D) Relative Error

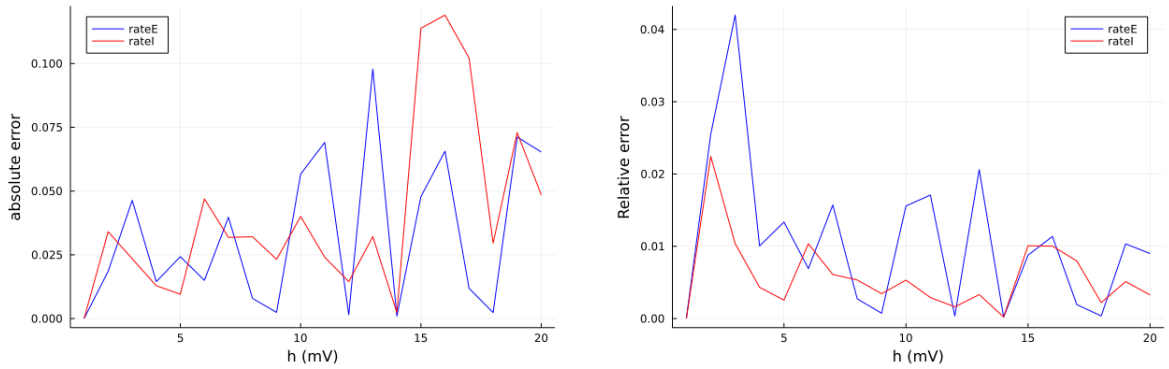
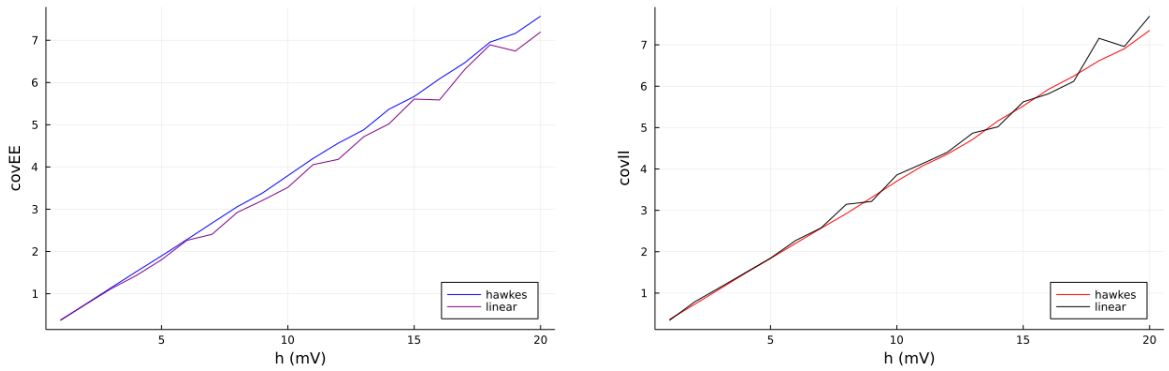


FIGURE 3.7: Comparing the steady state rate of a Linear Rate Model with noise with corresponding mean rate of the Linear Hawkes Process across different external input values This simulation shows a good match between the two models for both Excitatory and Inhibitory neuron rates, with maximum absolute error = 0.125 Hz and maximum relative error = 0.05

(A) Cov EE vs external input

(B) Cov II vs external input



(C) Absolute Error

(D) Relative Error

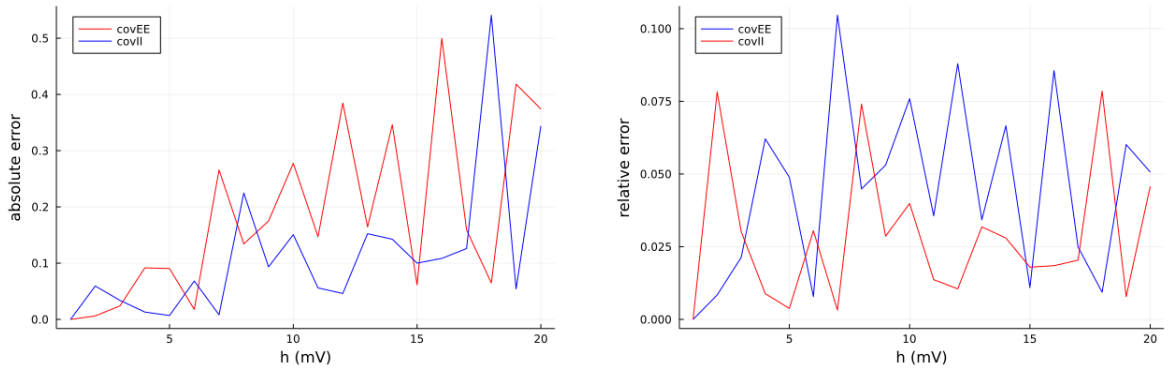
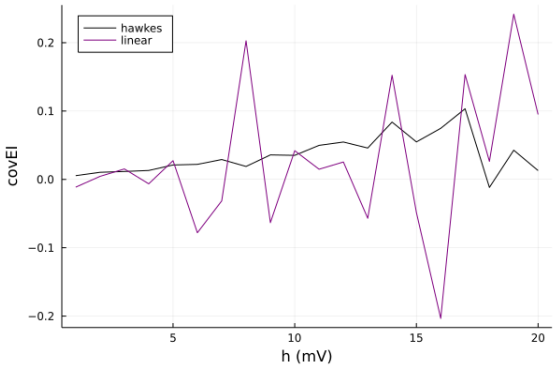
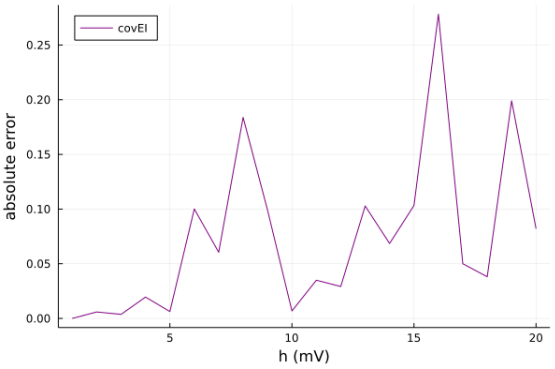


FIGURE 3.8: Comparing the variability of a Linear Rate Model with noise with corresponding variability of the Linear Hawkes Process across different external input values
Covariances of the two models match with a maximum absolute error = 0.6 and maximum relative error = 0.125. Both these covariances (Cov_{ee} and Cov_{ii}) have a positive linear relationship with h.

(A) Cov EI vs external input



(B) Absolute Error



(C) Relative Error

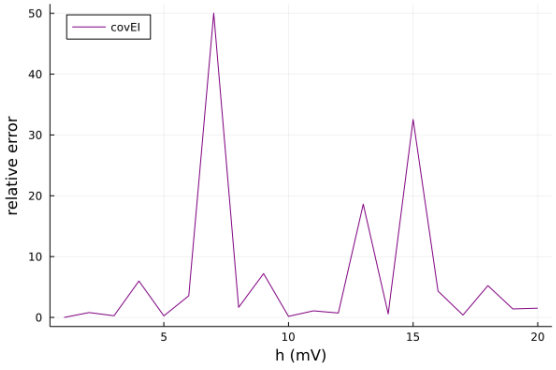
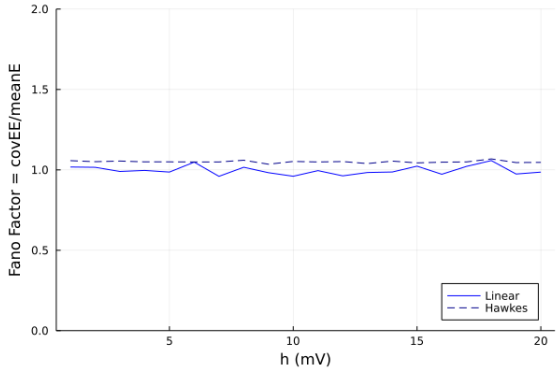


FIGURE 3.9: Comparing the variability of a Linear Rate Model with noise with corresponding variability of the Linear Hawkes Process across different external input values
It is difficult to comment on dependence of Cov_{ei} on h. Further investigation is needed.

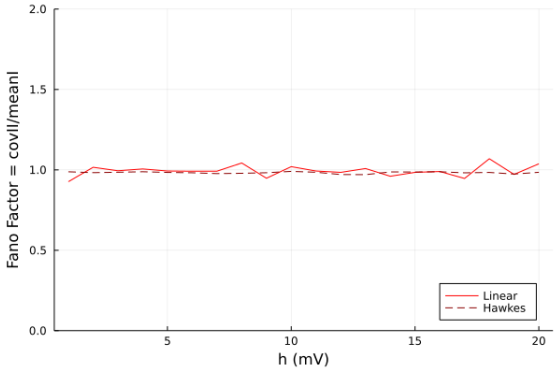
Fig 3.9 plots Cov_{ei} of the two models along h. This plot does not give a good result. It is also difficult to comment on dependence of Cov_{ei} on h, as $Cov_{ei}^{ratemodel}$ has huge variations for consecutive h values. However, this analysis gives a good idea on what work needs to be done. Firstly, we need to check whether the Cov_{ei} values found for the rate model match with the analytic results. Assess whether noise needs to be modelled differently to match with the analytic results. Find out whether and how can Hawkes processes model negative covariances.
Further investigation is needed.

The Fano Factor is another important property used to measure the variability of neuronal spike trains. It is defined as the variance-to-mean ratio of spike counts in a time window. We already have the variance and mean for a 2D linear rate model and 2D linear Hawkes from the previous section. Fig 3.10 plots the Fano Factor for both these models. It shows a good match for both excitatory and inhibitory neuron (maximum absolute error = 0.09, maximum relative error = 0.09). Fano Factor for linear models (both rate model and Hawkes) is clearly independent of h. Another interesting observation is that Fano Factor = 1 (approximately) for Hawkes and rate model, across all h values. A linear Hawkes process with a constant rate is essentially a Poisson process (mean = covariance). For our analysis, the instantaneous rate of Hawkes process keeps fluctuating. Although, the mean rate is constant, and also equal to the steady-state rate of the linear rate model, for a particular h.

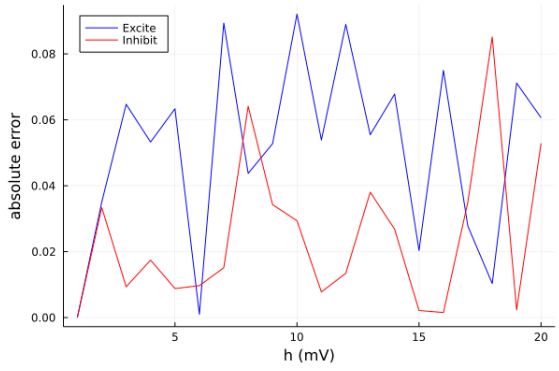
(A) Fano Factor EE vs external input



(B) Fano Factor II vs external input



(C) Absolute Error



(D) Relative Error

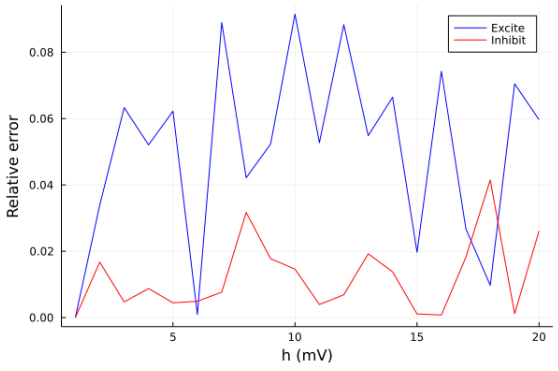


FIGURE 3.10: Comparing Fano Factor of the two models using covariance and mean found in the above analysis.

By extending the above analysis in the nonlinear regime, we can now start comparing SSNs with Hawkes processes. Two possible methods for the same are explored here. First, one could linearize an SSN **around its stable point**, and then compare this linear version with linear Hawkes processes using the method explained in the previous section. Second, one could directly compare the SSN with a nonlinear Hawkes process.

3.2 Approximation method

The sequence of steps for this method is :
SSN $\rightarrow$ Linear Approximation $\rightarrow$ Linear Hawkes. So we start with the SSN equation :

$$\tau_i \frac{\mathrm{d}V_i}{\mathrm{d}t} = -V_i + V_{rest} + h_i + \sum_j W_{ij} r(V_j) \tag{3.8}$$

$$r(V_j) = \alpha (V_j - V_{rest})^2 \tag{3.9}$$

Changing the variable V to $r(V)$:

$$\frac{\mathrm{d}r_j}{\mathrm{d}t} = 2\alpha (V_j - V_{rest}) \frac{\mathrm{d}V_j}{\mathrm{d}t} \tag{3.10}$$

$$\begin{aligned} \frac{\mathrm{d}r_i}{\mathrm{d}t} = F(r_i) &= 2\alpha (V_i - V_{rest})(-V_i + V_{rest} + h_i + \sum_j W_{ij} r_j) \frac{1}{\tau_i} \\ &= \{-2\alpha (V_i - V_{rest})^2 + 2\alpha (V_i - V_{rest})(h_i + \sum_j W_{ij} r_j)\} \frac{1}{\tau_i} \\ &= \{-2r_i + 2(\alpha r_i)^{\frac{1}{2}} (h_i + \sum_j W_{ij} r_j)\} \frac{1}{\tau_i} \end{aligned} \tag{3.11}$$

The Taylor expansion of the above equation is :

$$F(r_i) = F(r_i^{ss}) + \left. \frac{\partial \mathbf{F}(\mathbf{r})}{\partial r_i} \right|_{\mathbf{r^{ss}}} (r_i - r_i^{ss}) + \left. \frac{\partial^2 \mathbf{F}(\mathbf{r})}{\partial r_i^2} \right|_{\mathbf{r^{ss}}} \frac{(r_i - r_i^{ss})^2}{2!} + ... \tag{3.12}$$

where r_i^{ss} indicates the steady-state rate of neuron i.
Deriving the Taylor Expansion **up to first order** for one neuron at a time, to simplify variables. starting with the E-neuron.
To find the Jacobian terms,

$$\begin{aligned} \mathbf{F}_e = \frac{\mathrm{d}r_e}{\mathrm{d}t} &= \{-2r_e + 2(\alpha r_e)^{\frac{1}{2}} (h_e + \sum_j W_{ej} r_j)\} \frac{1}{\tau_e} \\ \mathbf{F}_e &= \frac{1}{\tau_e} (-2r_e + 2\alpha^{\frac{1}{2}} W_{ee} r_e^{\frac{3}{2}} + 2(\alpha r_e)^{\frac{1}{2}} (h_e + W_{ei} r_i)) \end{aligned} \tag{3.13}$$

$$\frac{\partial \mathbf{F}_e}{\partial r_e} = \{-2 + 3\alpha^{\frac{1}{2}} r_e^{\frac{1}{2}} W_{ee} + \alpha^{\frac{1}{2}} r_e^{\frac{-1}{2}} (h_e + W_{ei} r_i)\} \frac{1}{\tau_e} \tag{3.14}$$

$$\frac{\partial \mathbf{F}_e}{\partial r_i} = \{2\alpha^{\frac{1}{2}} r_e^{\frac{1}{2}} W_{ei}\} \frac{1}{\tau_e} \tag{3.15}$$

So, the Taylor Expansion becomes :

$$F(r_e) = F(r_e^{ss}) + \left.\frac{\partial \mathbf{F}_e}{\partial r_e}\right|_{r_e^{ss}} (r_e - r_e^{ss}) + \left.\frac{\partial \mathbf{F}_e}{\partial r_i}\right|_{r_i^{ss}} (r_i - r_i^{ss}) \tag{3.16}$$

First term in Taylor Expansion = $F(r_e^{ss}) = \frac{\mathrm{d}r_e}{\mathrm{d}t}$ at steady state = 0
For the first order term, adding Eqs 3.14 and 3.15 :

$$\begin{aligned} &\left.\frac{\partial \mathbf{F}_e}{\partial r_e}\right|_{r_e^{ss}} (r_e - r_e^{ss}) + \left.\frac{\partial \mathbf{F}_e}{\partial r_i}\right|_{r_i^{ss}} (r_i - r_i^{ss}) = \\ &\{(-2 + 3\alpha^{\frac{1}{2}} r_e^{\frac{1}{2}} W_{ee} + \alpha^{\frac{1}{2}} r_e^{\frac{-1}{2}} (h_e + W_{ei} r_i))(r_e - r_e^{ss}) + 2\alpha^{\frac{1}{2}} r_e^{\frac{1}{2}} W_{ei} r_i - r_i^{ss}\} \frac{1}{\tau_e} \end{aligned} \tag{3.17}$$

Separating terms multiplied to r_e, r_i to get the new weight matrix and new baseline current :

$$\begin{aligned} &\tau_e \left.\frac{\partial \mathbf{F}_e}{\partial r_e}\right|_{r_e^{ss}} (r_e - r_e^{ss}) + \tau_e \left.\frac{\partial \mathbf{F}_e}{\partial r_i}\right|_{r_i^{ss}} (r_i - r_i^{ss}) = \\ &- 2r_e + \{3(\alpha r_e^{ss})^{\frac{1}{2}} W_{ee} + (\frac{\alpha}{r_e^{ss}})^{\frac{1}{2}} (h_e + W_{ei} r_i^{ss})\} r_e \\ &+ \{2(\alpha r_e^{ss})^{\frac{1}{2}} W_{ei}\} r_i \\ &+ \{(-2 + 3(\alpha r_e^{ss})^{\frac{1}{2}} W_{ee} + (\frac{\alpha}{r_e^{ss}})^{\frac{1}{2}} (h_e + W_{ei} r_i^{ss}))(-r_e^{ss}) - 2(\alpha r_e^{ss})^{\frac{1}{2}} W_{ei} r_i^{ss}\} \end{aligned} \tag{3.18}$$

So, we can write the Taylor expansion as :

$$\frac{\mathrm{d}r_e}{\mathrm{d}t} = \{-2r_e + W_{ee}^* r_e + W_{ei}^* r_i + h_e^*\} \frac{1}{\tau_e} \tag{3.19}$$

where $W_{ee}^* = 3(\alpha r_e^{ss})^{\frac{1}{2}} W_{ee} + (\frac{\alpha}{r_e^{ss}})^{\frac{1}{2}} (h_e + W_{ei} r_i^{ss})$ and

$W_{ei}^* = 2(\alpha r_e^{ss})^{\frac{1}{2}} W_{ei}$ are the terms of the new weight matrix relevant for the E-neuron, and

$h_e^* = (-2 + 3(\alpha r_e^{ss})^{\frac{1}{2}} W_{ee} + (\frac{\alpha}{r_e^{ss}})^{\frac{1}{2}} (h_e + W_{ei} r_i^{ss}))(-r_e^{ss}) - 2(\alpha r_e^{ss})^{\frac{1}{2}} W_{ei} r_i^{ss}$ is the new baseline current acting on E-neuron.

A similar analysis of the I-neuron gives us :

$$\frac{\mathrm{d}r_i}{\mathrm{d}t} = \{-2r_i + W_{ii}^* r_i + W_{ie}^* r_e + h_i^*\} \frac{1}{\tau_i} \tag{3.20}$$

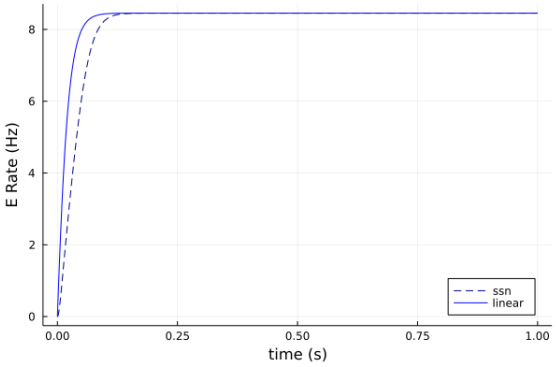
where $W_{ii}^* = 3(\alpha r_i^{ss})^{\frac{1}{2}} W_{ii} + (\frac{\alpha}{r_i^{ss}})^{\frac{1}{2}} (h_i + W_{ie} r_e^{ss})$ and

$W_{ie}^* = 2(\alpha r_i^{ss})^{\frac{1}{2}} W_{ie}$ are the terms of the new weight matrix relevant for the I-neuron, and

$h_i^* = (-2+3(\alpha r_i^{ss})^{\frac{1}{2}} W_{ii} + (\frac{\alpha}{r_i^{ss}})^{\frac{1}{2}}(h_i + W_{ie} r_e^{ss}))(-r_i^{ss}) - 2(\alpha r_i^{ss})^{\frac{1}{2}} W_{ie} r_e^{ss}$ is the new baseline current acting on I-neuron.

These equations give us the linear approximation of the SSN (Eqs 3.8 and 3.9).
Next, we simulate a comparison between an SSN and its linear approximation at a fixed external input ($h_E = h_I = 5$ mV). This is shown in Fig 3.11. The SSN in consideration is the same two dimensional network we have been using for all our simulations to avoid any complication. So, $\alpha = 0.3$, $W = [1.25 - 0.65; 1.2 - 0.5]$, $\tau_E = 20.0$, $\tau_I = 10.0$. The variable values for the linearly approximated system are calculated using Eq. 3.19. Both E and I neurons start at the resting potential ($V_{rest} = -70$ mV). As the simulation progresses, we can see that the SSN model and the linear model both reach the same steady state. The steady-state rate values are : $r\_excite_{ssn} = r\_excite_{linear} = 8.454$ Hz, $r\_inhibit_{ssn} = r\_inhibit_{linear} = 15.784$ Hz.
Fig 3.12 extends this comparison across different h values ($h\_max = 20.0$; $h\_min = 1.0$; $step = 1$). We can see that the steady state values for the SSN and its linear approximation match perfectly across the different h values. The non zero absolute error values and relative error values are small enough ($\mathcal{O}(10^{-13})$) to be attributed to computational errors.
Hence, we can positively say that our calculations are correct so far!
An important thing to note about Fig 3.12 : Equations 3.19 and 3.20 imply that the variables defining a linearly approximated network around the stable point of an SSN are dependent on W^{ssn}, h^{ssn}, and most importantly r_e^{ssn} and r_i^{ssn}. So, every point in the plot of the linearly approximated network stands for a DIFFERENT linear network. In other words, we are comparing a single SSN to its linearly approximated networks around each stable point, which are obviously different for different h values.

(A) Excitatory Rate vs time



(B) Inhibitory Rate vs time

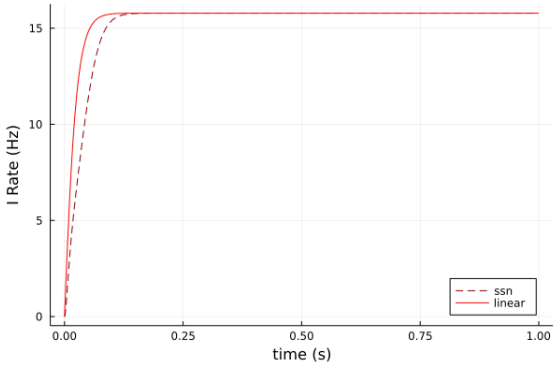
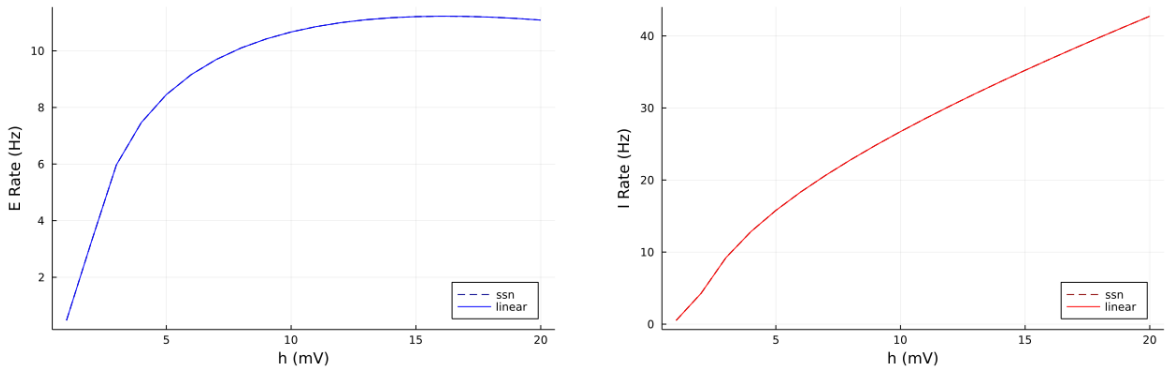


FIGURE 3.11: SSN → Linear Approximated Model, for h = 5 mV
Both E and I neurons start at the resting potential. As the simulation progresses, we can see that the SSN model and the linear model both reach the same steady state. The steady-state rate values are : $r\_excite_{ssn} = r\_excite_{linear} = 8.454$ Hz, $r\_inhibit_{ssn} = r\_inhibit_{linear} = 15.784$ Hz.

(A) Excitatory Rate vs external input

(B) Inhibitory Rate vs external input



(C) Absolute Error

(D) Relative Error

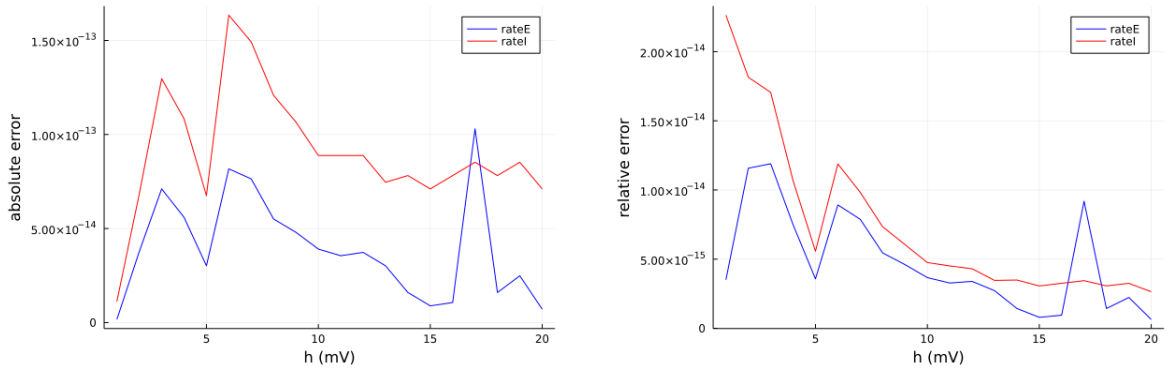


FIGURE 3.12: SSN → Linear Approximated Model
The steady-state values for both the models match well. Every point in the linear network's plot stand for a different linearly approximated network defined with variables dependent on the steady-state variables of the SSN for each h.

Moving on, we can now compare an SSN with a linear Hawkes processes via the two-step transformation :

SSN → Linear Approximation → Linear Hawkes

We start with a two dimensional E/I SSN described with the variables $W_{EE} = 1.25$, $W_{EI} = -0.65$, $W_{IE} = 1.2$, $W_{II} = -0.5$, $\tau_E = 20.0$, $\tau_I = 10.0$, $V_{rest} = -70$ mV with external input $h_E = h_I = 5.0$ mV.

$$\begin{aligned} \tau_i \frac{\mathrm{d}V_i}{\mathrm{d}t} &= -V_i + V_{rest} + h_i + \sum_j W_{ij} r(V_j) \\ r(V_j) &= \alpha (V_j - V_{rest})^2 \end{aligned} \tag{3.21}$$

Find its linearly approximated network by using Taylor Expansion around the stable point of rate variable.

$$\frac{\mathrm{d}r_i}{\mathrm{d}t} = \{-2r_i + \sum_j W^*_{ij} r_j + h^*_i\} \frac{1}{\tau_i} \tag{3.22}$$

This linear network is defined by the weight matrix $\mathbf{W}^* = f_1(\mathbf{W}, \mathbf{h}, \mathbf{r^{ss}})$, the baseline current $\mathbf{h}^* = f_2(\mathbf{W}, \mathbf{h}, \mathbf{r^{ss}})$ and time constants $\tau_E = 20.0$ and $\tau_I = 10.0$.

Finally, generate a 2D linear Hawkes model defined with the variables $\mathbf{W^{hawkes}} = \alpha\mathbf{W}^*$, $\mathbf{h^{hawkes}} = \alpha\mathbf{h}^*$ and $\tau_E^{hawkes} = 10.0$, $\tau_I^{hawkes} = 5.0$.

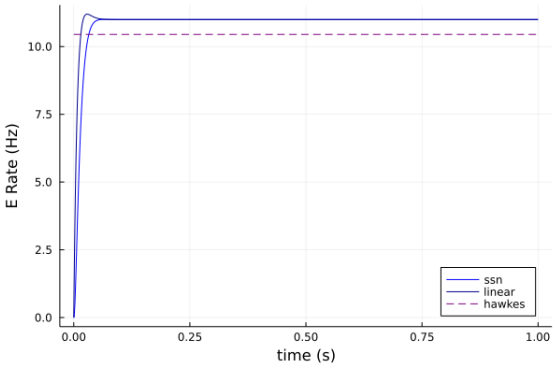
$$r_i^{hawkes}(t) = h_i^{hawkes} + \sum_j W_{ij}^{hawkes} \sum_x \frac{e^{(t_{jx}-t)/\tau_j^{hawkes}}}{\tau_j^{hawkes}} \tag{3.23}$$

Fig 3.13 shows this comparison for a fixed external input $h_E = h_I = 12$ mV. Values of variables used : $V_{rest} = -70$ mV, $\tau_E^{ssn} = 20$ ms, $\tau_I^{ssn} = 10$ ms, $\tau_E^{linearratemodel} = 10$ ms, $\tau_I^{linearratemodel} = 5$ ms, $\tau_E^{hawkes} = 10$ s, $\tau_I^{hawkes} = 5$ s, $W^{ssn} = [1.25 - 0.65; 1.2 - 0.5]$, $\alpha = 0.3$. The simulation gives a decent match in the rates with $rate_E^{ssn} = 10.998$ Hz, $rate_I^{ssn} = 30.296$ Hz, $rate_E^{linearratemodel} = 10.998$ Hz, $rate_I^{linearratemodel} = 30.296$ Hz, $rate_E^{hawkes} = 10.447$ Hz and $rate_I^{hawkes} = 28.964$ Hz.

Fig 3.14 extends this comparison across different h values ($h\_max = 20.0; h\_min = 1.0, step = 1$), showing a good match with maximum absolute error = 0.6 Hz and maximum relative error = 0.15.

An important thing to note about Fig 3.14 : as discussed earlier, we are comparing a single SSN to its linearly approximated networks around each stable point, which are obviously different for different h values. As a result, every point in the plot of linear Hawkes model also stands for a DIFFERENT Hawkes network.

(A) Excitation Rate vs time



(B) Inhibition Rate vs time

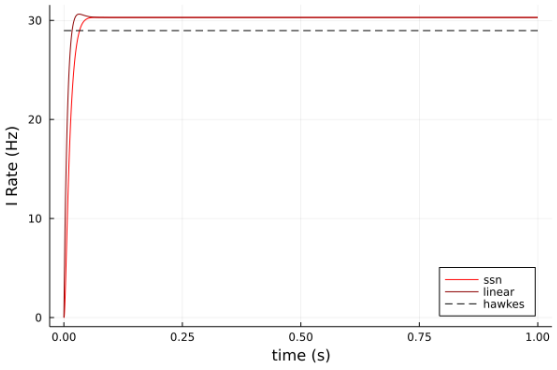
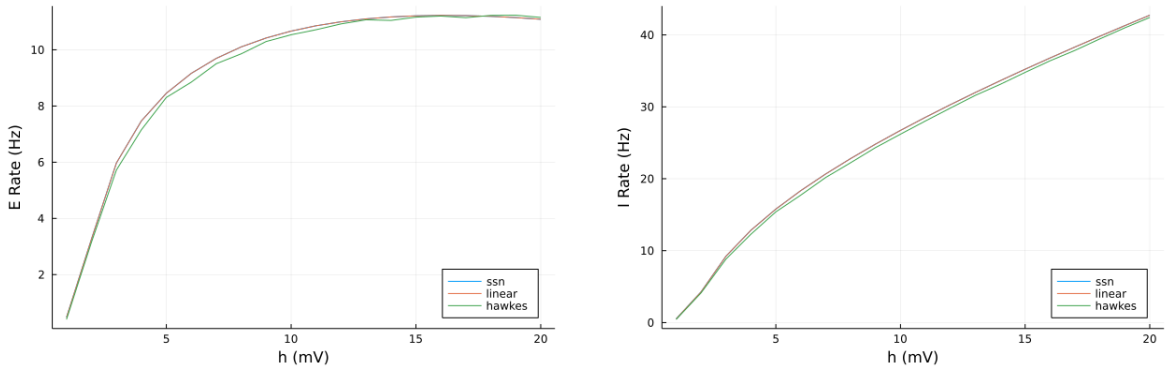


FIGURE 3.13: Comparing SSN, its linear approximation and Linear Hawkes for a fixed external input
shows the rates of the three models over a time interval of 1 seconds. The simulation gives a decent match in the rates with $rate_E^{ssn} = 10.998$ Hz, $rate_I^{ssn} = 30.296$ Hz, $rate_E^{linearratemodel} = 10.998$ Hz, $rate_I^{linearratemodel} = 30.296$ Hz, $rate_E^{hawkes} = 10.447$ Hz and $rate_I^{hawkes} = 28.964$ Hz.

(A) Excitatory Rate vs external input

(B) Inhibitory Rate vs external input



(C) Absolute Error

(D) Relative Error

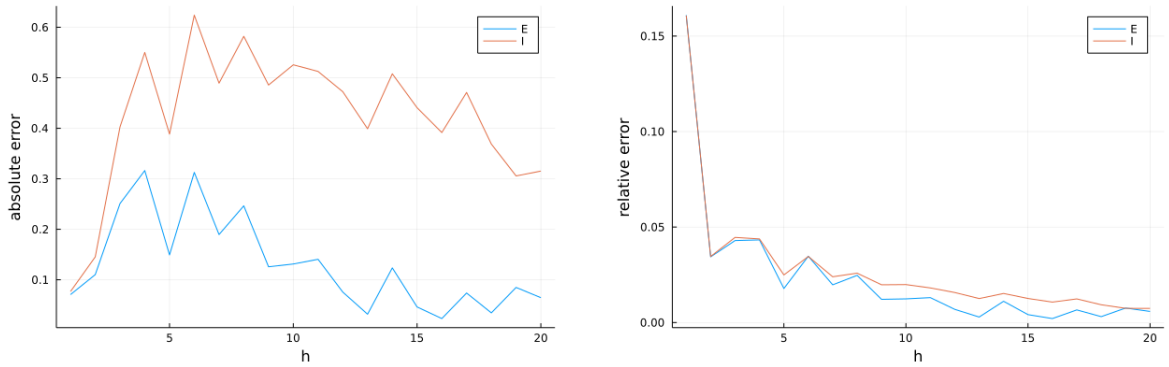


FIGURE 3.14: SSN → Linear Approximated Model → Linear Hawkes
This simulation shows a good match between the three models for both Excitatory and Inhibitory neuron rates, with maximum absolute error = 0.6 Hz and maximum relative error = 0.15. Hence, we have successfully found linear Hawkes processes with mean spiking rates equal to the steady-state rates of SSNs.

3.3 Direct Method

Here we explore the method of directly comparing the steady state of an SSN with Nonlinear Hawkes. The analysis is similar to its linear counterpart. As is customary, starting with the SSN equation :

$$\begin{aligned} \tau_i \frac{\mathrm{d}V_i}{\mathrm{d}t} &= -V_i + V_{rest} + h_i + \sum_j W_{ij} r(V_j) \\ r(V_j) &= \alpha (V_j - V_{rest})^n \end{aligned} \tag{3.24}$$

Changing the variable V to $r(V)$, since we will be considering the stable point of the rates of spiking of each individual unit in the SSN :

$$\frac{\mathrm{d}r_j}{\mathrm{d}t} = \alpha n (V_j - V_{rest})^{n-1} \frac{\mathrm{d}V_j}{\mathrm{d}t} \tag{3.25}$$

$$\begin{aligned} \tau_i \frac{\mathrm{d}r_i}{\mathrm{d}t} &= \alpha n (V_i - V_{rest})^{n-1} (-V_i + V_{rest} + h_i + \sum_j W_{ij} r_j) \\ &= -\alpha n (V_i - V_{rest})^n + \alpha n (V_i - V_{rest})^{n-1} (h_i + \sum_j W_{ij} r_j) \\ &= -n r_i + n \alpha^{\frac{1}{n}} r^{\frac{n-1}{n}} (h_i + \sum_j W_{ij} r_j) \end{aligned} \tag{3.26}$$

At the Steady State, $\frac{\mathrm{d}r_i}{\mathrm{d}t} = 0$. So,

$$\begin{aligned} r_i^{ss} &= \alpha^{\frac{1}{n}} r^{\frac{n-1}{n}} (h_i + \sum_j W_{ij} r_j^{ss}) \\ r_i^{\frac{1}{n} ss} &= \alpha^{\frac{1}{n}} (h_i + \sum_j W_{ij} r_j^{ss}) \\ r_i^{ss} &= (\alpha^{\frac{1}{n}} h_i + \sum_j \alpha^{\frac{1}{n}} W_{ij} r_j^{ss})^n \end{aligned} \tag{3.27}$$

Comparing this steady state with nonlinear Hawkes,

$$r_i^{hawkes}(t) = \left(h_i^{hawkes} + \sum_j W_{ij}^{hawkes} \sum_x \frac{e^{(t_{jx} - t)/\tau_j}}{\tau_j} \right)^n \tag{3.28}$$

$$h_i^{hawkes} = \alpha^{\frac{1}{n}} h_i \qquad W_{ij}^{hawkes} = \alpha^{\frac{1}{n}} W_{ij} \tag{3.29}$$

This suggests that using $\alpha^{\frac{1}{n}} h$ as h^{hawkes} and $\alpha^{\frac{1}{n}} W$ as W^{hawkes} could give us a Hawkes process with mean rate equal to the steady state rate of the corresponding SSN.

Fig 3.15 compares the steady state rate of a 2D SSN with corresponding mean rate of the 2D nonlinear Hawkes process. Values of the variables used : h = 5mV, $n = 2$, $\alpha = 0.3$, $W = [1.25 - 0.65; 1.2 - 0.5]$, $\tau_E^{ssn} = 20.0$ ms, $\tau_I^{ssnl} = 10.0$ ms, $h_{hawkes} = \sqrt{\alpha} h$, $W_{hawkes} = \sqrt{\alpha} W$,

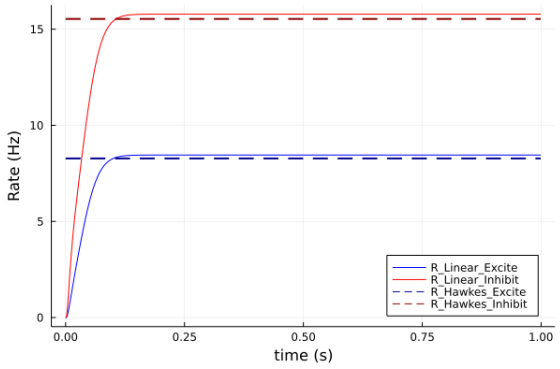
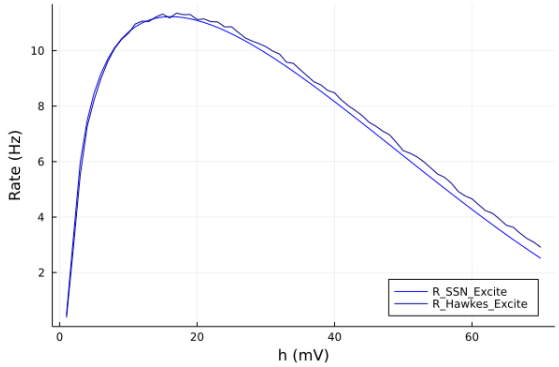


FIGURE 3.15: SSN vs NonLinear Hawkes, for a fixed external input h = 5 mV, Values of the variables used : h = 5mV, $h^{hawkes} = \sqrt{\alpha}h$, $W^{hawkes} = \sqrt{\alpha}W$. The voltages of the neurons in the SSN stabilize at : $v\_excite = -64.691$ mV, $v\_inhibit = -62.746$ mV. Their corresponding steady state rates are : $r\_excite = 8.454$ Hz and $r\_inhibit = 15.784$ Hz. The mean rates of the two Hawkes processes are : $r_E = 8.277$ Hz and $r_I = 15.536$ Hz. Hence, Rates of the two models match well.

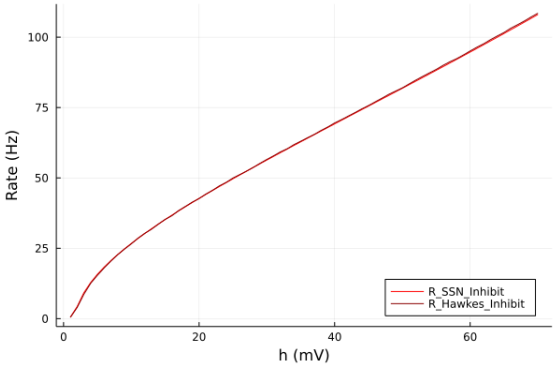
$tau_E^{hawkes} = 5.0$ s, $tau_I^{hawkes} = 2.0$ s, $n\_spikes = 500,000$. The voltages of the neurons in the SSN stabilize at : $v\_excite = -64.691$ mV; $v\_inhibit = -62.746$ mV. Their corresponding steady state rates are : $r\_excite = 8.454$ Hz and $r\_inhibit = 15.784$ Hz. The mean rates of the two nonlinear Hawkes processes were found to be : $r_E = 8.277$ Hz and $r_I = 15.536$ Hz. Rates of the two models match well, with an absolute error of 0.248 Hz and relative error of 0.016. Hence, the nonlinear Hawkes processes defined above can generate spikes with an effective mean rate equal to the steady-state rate of the SSN!

Fig 3.16 compares the rates across different h values ($h\_max = 70.0$; $h\_min = 1.0$; $step = 1$). The rates for each steady-state in the plot of the linear model matches well with the mean rates of the two Hawkes processes, with input current = $\sqrt{\alpha}h$ Hz. The absolute error appears to form a minima around $h = 10$ mV, with the maximum observed error = 0.6 Hz. This suggests that the tau^{hawkes} values chosen favours h values around 10 mV. To investigate more, we plot the relative error and find a similar observation. The maximum relative error is 0.14. Hence, the nonlinear Hawkes processes can generate spikes with mean rate equal to the steady-state rate of the SSN for different h values.

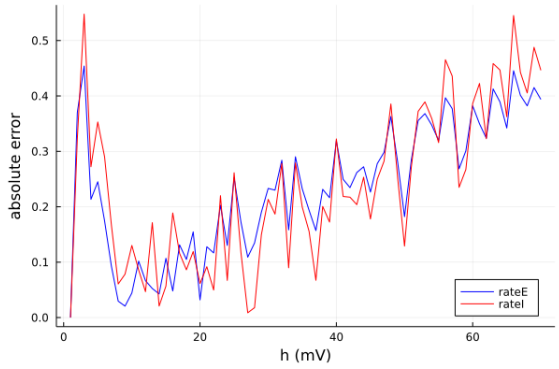
(A) Excitatory Rate



(B) Inhibitory Rate



(C) Absolute Error



(D) Relative Error

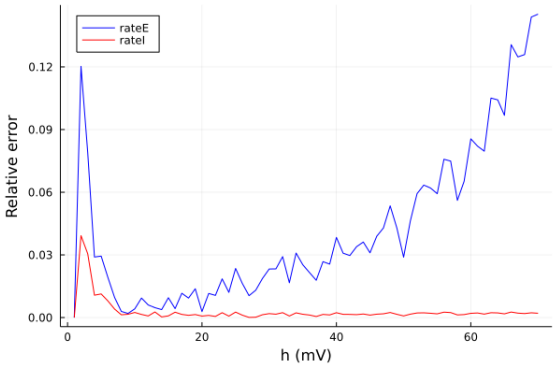


FIGURE 3.16: SSN vs NonLinear Hawkes
The rates for each steady-state in the plot of the SSN match well with the mean rates of the nonlinear Hawkes processes, with input current $h^{hawkes} = \sqrt{\alpha}h$ Hz. Maximum absolute error = 0.6 Hz, maximum relative error = 0.14. Hence, nonlinear Hawkes processes can generate spikes with mean rate equal to the steady-state rate of the SSN model for different h values, confirming that our analysis is moving the right direction.

Chapter 4

Conclusion and Further directions

This project studies the application of Hawkes processes in computational neuroscience. It takes a two dimensional E/I Stabilized Supralinear Network and finds relations between its defining variables with the defining variables of a Hawkes process.

We first work on the linear version of the problem, comparing the linear rate model with linear Hawkes. We find out that using $h^{hawkes} = \alpha h^{ratemodel}$ and $W^{hawkes} = \alpha W^{ratemodel}$ gives linear mutually exciting Hawkes processes with the same mean rate of spiking for each individual process as the steady-state of each neuron in the linear rate model. This result was successfully confirmed for different h values. Hence, we can generate spike trains with mean rate as the steady-state rate of a linear rate model using linear Hawkes processes.

The next step is of course to move on to second order statistics. So we compare the input-dependent variability of the spike trains of linear Hawkes with that of a linear rate model with noise. This provides good preliminary results, but the analysis is currently incomplete. Further investigation is needed to positively confirm the results obtained here.

Next, nonlinearity is added into the picture and SSNs are compared with Hawkes processes with the help of the linear analysis. This is done in two ways.
First, an SSN is linearized around its stable point, and the resultant linearly approximated network is compared with linear Hawkes. The rates of these three models, SSN, its linearly approximated networks around stable points and the corresponding linear Hawkes processes match well across different h values. Hence, we can successfully generate spike trains with mean rate as the steady-state rate of an SSN using linear Hawkes processes.
Second, a direct comparison is made between the steady-state rates of SSNs and mean spiking rates of nonlinear Hawkes processes using $h^{hawkes} = \sqrt{\alpha} h^{ratemodel}$ and $W^{hawkes} = \sqrt{\alpha} W^{ratemodel}$. This analysis gives a positive result.

An important paper relevant to this discussion is "Linking structure and activity in nonlinear spiking networks", by Ocker, Josic, Shea-Brown and Buice [Ock+17b]. This paper works with the nonlinear Hawkes in a general form, and computes its statistical properties. We have attempted to take a more simplified approach of relating Hawkes processes with linear rate models and SSNs at their stable points and building an easily understandable resource base for further exploration.

Further work on this would involve first completing the analysis of variability for the linear and nonlinear case. This will be followed by a study of motifs and plasticity rules using models of Hawkes processes in place of SSNs.
Synaptic plasticity causes changes in neural weights, based on the statistics of neural activity. This is of interest as plasticity regulates learning and memory in the brain, and helps in figuring out how a system can self-organize in a particular structure. Some motifs reinforce themselves due to plasticity and other motifs fade away. One can consider how plasticity rules modify and shape the connectivity of SSNs, and on what conditions can one find stationary solutions for the self-organized structure of neural weights arising from specific plasticity rules [OLKD15; MMG20].

Appendix A

Relevant Formulae and Algorithms used

• relative error between a and $b = \frac{2*abs(a-b)}{a+b}$

• Thinning Algorithm used to generate spikes in Hawkes processes [LTP15] :

```
1: procedure HawkesByThinning(T, lambda*(.))
2: require: lambda*(.) non-increasing in periods of no arrivals.
3: e ← 10^(-10) (some tiny value > 0).
4: P ← [], t ← 0.
5: while t < T do
6: Find new upper bound:
7: M ← lambda*(t + e).
8: Generate next candidate point:
9: E ← Exp(M), t ← t + E.
10: Keep it with some probability:
11: U ← Unif(0, M).
12: if t < T and U <= lambda*(t) then
13: P ← [P, t].
14: end if
15: end while
16: return P
17: end procedure
```

Bibliography

[AM19] Yashar Ahmadian and Kenneth D. Miller. "What is the dynamical regime of cerebral cortex?" In: *arXiv:1908.10101 [q-bio]* (Aug. 28, 2019). arXiv: 1908.10101. URL: http://arxiv.org/abs/1908.10101 (visited on 11/26/2020).

[ARM13] Yashar Ahmadian, Daniel B. Rubin, and Kenneth D. Miller. "Analysis of the Stabilized Supralinear Network". In: *Neural Computation* 25 (May 10, 2013), pp. 1994–2037. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00472. URL: https://doi.org/10.1162/NECO_a_00472 (visited on 01/29/2020).

[Ech+20] Rodrigo Echeveste et al. "Cortical-like dynamics in recurrent circuits optimized for sampling-based probabilistic inference". In: *Nature Neuroscience* (Aug. 10, 2020), pp. 1–12. ISSN: 1546-1726. DOI: 10.1038/s41593-020-0671-1. URL: https://www.nature.com/articles/s41593-020-0671-1 (visited on 08/16/2020).

[Gjo+11] Julijana Gjorgjieva et al. "A triplet spike-timing–dependent plasticity model generalizes the Bienenstock–Cooper–Munro rule to higher-order spatiotemporal correlations". In: *Proceedings of the National Academy of Sciences* 108.48 (2011), pp. 19383–19388. ISSN: 0027-8424. DOI: 10.1073/pnas.1105933108. eprint: https://www.pnas.org/content/108/48/19383.full.pdf. URL: https://www.pnas.org/content/108/48/19383.

[Haw71] Alan G. Hawkes. "Point Spectra of Some Mutually Exciting Point Processes". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 33 (1971), pp. 438–443. ISSN: 2517-6161. DOI: https://doi.org/10.1111/j.2517-6161.1971.tb01530.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1971.tb01530.x (visited on 04/25/2021).

[Hen+18] Guillaume Hennequin et al. "The Dynamical Regime of Sensory Cortex: Stable Dynamics around a Single Stimulus-Tuned Attractor Account for Patterns of Noise Variability". In: *Neuron* 98 (May 16, 2018), 846–860.e5. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2018.04.017. URL: http://www.sciencedirect.com/science/article/pii/S0896627318303258 (visited on 03/22/2019).

[JHR15] Stojan Jovanović, John Hertz, and Stefan Rotter. "Cumulants of Hawkes point processes". In: *Physical Review E* 91.4 (2015), p. 042802.

[LTP15] Patrick J. Laub, Thomas Taimre, and Philip K. Pollett. *Hawkes Processes*. 2015. arXiv: 1507.02822 [math.PR].

[MMG20] Lisandro Montangie, Christoph Miehl, and Julijana Gjorgjieva. "Autonomous emergence of connectivity assemblies via spike triplet interactions". In: *PLOS Computational Biology* 16 (May 8, 2020). Publisher: Public Library of Science, e1007835. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1007835. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007835 (visited on 04/12/2021).

[Ock+17a] Gabriel Koch Ocker et al. "From the statistics of connectivity to the statistics of spike times in neuronal networks". In: *Current Opinion in Neurobiology*. Computational Neuroscience 46 (Oct. 1, 2017), pp. 109–119. ISSN: 0959-4388. DOI: 10.1016/j.conb.2017.07.011. URL: http://www.sciencedirect.com/science/article/pii/S0959438817300740 (visited on 12/07/2020).

[Ock+17b] Gabriel Koch Ocker et al. "Linking structure and activity in nonlinear spiking networks". In: *PLOS Computational Biology* 13 (June 23, 2017). Publisher: Public Library of Science, e1005583. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005583. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005583 (visited on 05/17/2021).

[OLKD15] Gabriel Koch Ocker, Ashok Litwin-Kumar, and Brent Doiron. "Self-Organization of Microcircuits in Networks of Spiking Neurons with Plastic Synapses". In: *PLOS Comput Biol* 11 (Aug. 20, 2015), e1004458. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1004458. URL: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004458 (visited on 03/21/2016).

[RVM15] Daniel B. Rubin, Stephen D. Van Hooser, and Kenneth D. Miller. "The Stabilized Supralinear Network: A Unifying Circuit Motif Underlying Multi-Input Integration in Sensory Cortex". In: *Neuron* 85 (Jan. 21, 2015), pp. 402–417. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2014.12.026. URL: http://www.sciencedirect.com/science/article/pii/S0896627314011350 (visited on 02/01/2016).

[Yus15] Rafael Yuste. "From the neuron doctrine to neural networks". In: *Nature Reviews Neuroscience* 16 (Aug. 2015), pp. 487–497. ISSN: 1471-003X. DOI: 10.1038/nrn3962. URL: http://www.nature.com/nrn/journal/v16/n8/abs/nrn3962.html (visited on 11/24/2015).