

# Theta Model

Daniel Borrus

February 14th, 2020

## 1 Introduction

This is a network model of simple oscillators, here called neurons.

Consider a network of  $N$  neurons. The phase of a given neuron,  $i$ , in the network of  $N$  neurons is given by  $\theta_i$ , where

$$\dot{\theta}_i = \frac{1 - \cos \theta_i + [1 + \cos \theta_i] I_{\text{app}_i}}{\theta_\tau}. \quad (1)$$

See the phase diagram of the ODE below, to quickly understand the behavior.

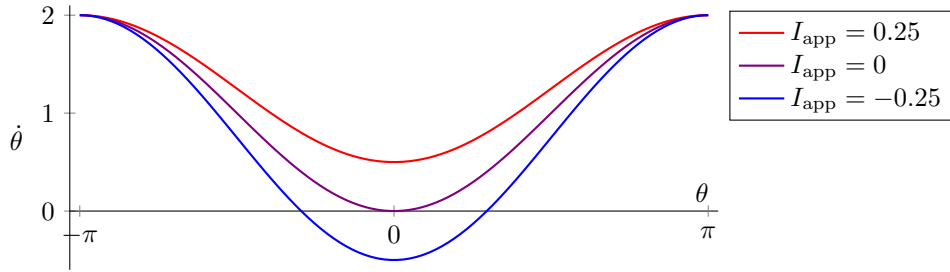


Figure 1: Plot of Equation 1 with two different  $I_{\text{app}}$ .  $\theta_\tau = 1$

When a neuron  $\theta_i$  crosses  $> \pi$  it “fires” and is reset back to  $-\pi$ . As the ODE has a period of  $2\pi$ , the dynamics are unaffected. As one can see from the phase diagram above, when  $I_{\text{app}}$  is negative, the neuron will be silent; when  $I_{\text{app}}$  is positive, the neuron will fire tonically.

A neuron  $i$  also has 3 additional ODEs, and probably 1 additional state variable.  $m_i$  and  $n_i$  describe the synaptic depression of a given neuron  $i$ , where ultimately the synaptic depression is  $y = 1 - n$ .

$$\dot{m}_i = -m_i/m_\tau \quad (2)$$

$$\dot{n}_i = \alpha_n m_i (1 - n_i) - n_i/n_\tau \quad (3)$$

After a burst ( $\theta_i > \pi$ ),  $m_i$  is stepped up a fraction of the way from its current value to  $m = 1$ ; the magnitude of this step is proportional to  $m_{\text{gain}}$ . Thus if a neuron “spikes” at time  $t$ , then

$$m_i(t+1) = m_i(t) + m_{\text{gain}}(1 - m_i(t))$$

.

The variable  $n$  rises more slowly, growing proportional to  $m$ . Thus synaptic depression comes on more gradually after a spike, and will grow with more rapid spiking, or activity. This allows synaptic depression ( $y = 1 - n$ ) to move on a slower time-scale than the next term, synaptic conductance ( $s$ ).

The variable  $s_i$  describes the amount of synaptic current a neuron generates for downstream neurons. It’s behavior is simple,

$$\dot{s}_i = -s_i/s_\tau \quad (4)$$

But similar to  $m_i$ , after a spike  $s_i$  immediately increases an amount proportional to  $s_{\text{gain}}$  so that

$$s_i(t+1) = m_i(t) + s_{\text{gain}}(1 - s_i(t))$$

. So after a spike, a neuron generates an “EPSP” that decays away with rate  $s_r$ .

Ultimately, synaptic depression ( $y$ ) and conductance ( $y$ ) determine the amount of “current” an upstream neuron generates for its target neurons. For each time step, a neuron’s  $I_{\text{app}_i}$  is calculated as

$$I_{\text{app}_i} = I_0 + d \sum_j s_j y_j \quad (5)$$

where  $I_0$  is the intrinsic  $I_{\text{app}}$  for each neuron (this can be the same for every neuron, or a distribution).  $d$  is the strength of the synaptic connection, a scalar.  $s_j$  is the synaptic conductance for all upstream neurons  $j$  from neuron  $i$ .  $y_j$  is the synaptic depression for all upstream neurons  $j$  from neuron  $i$ . Thus, Equation 5 reads: The applied current a neuron  $i$  receives at any time step is equal to an intrinsic applied current ( $I_0$ ), plus the strength of all upstream neuron synapses (a product of their synaptic conductance ( $s_j$ ) and recovery from synaptic depression ( $y_j$ )).

## 2 Network

An important question that we are very interested in is what the network topology should be. For the purpose of this short overview of the model, I will focus on two network structures.

The network structure we were using was an Erdos-Renyi random network with connectivity of  $p = 0.14$ . It’s worth pointing out, that for a network of 650 neurons, an ER network with  $p = 0.14$  leads to 91 connections per node. A bit much. Regardless, a network like this can produce robust oscillations. See section 3.

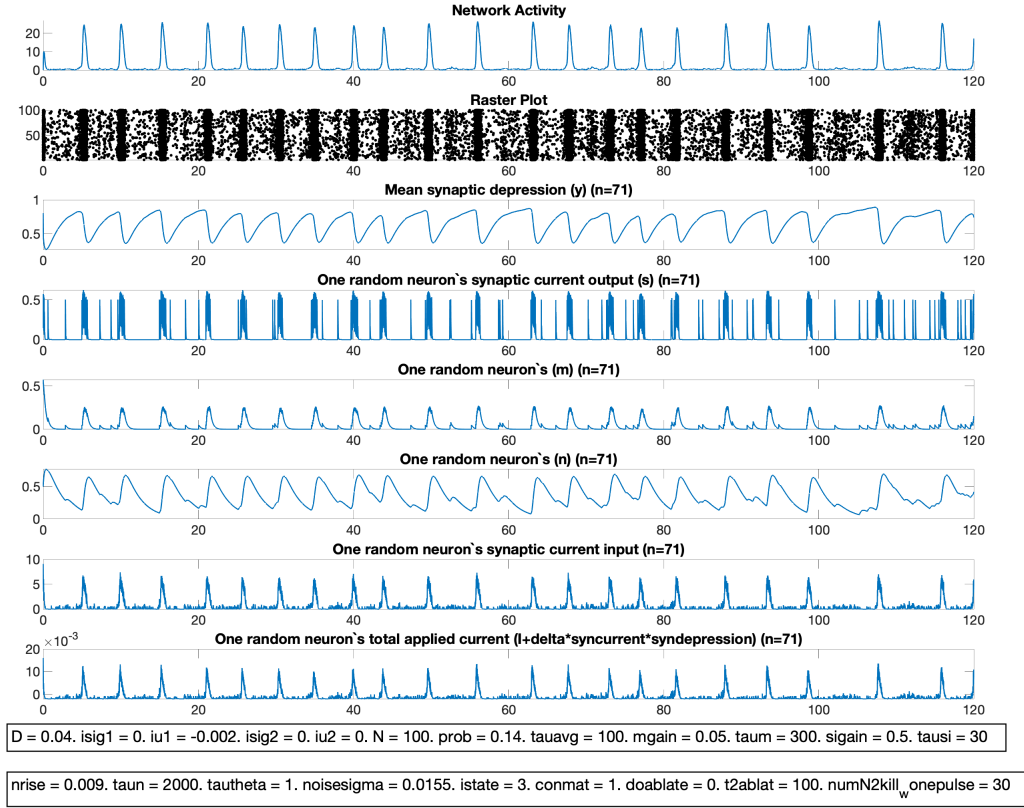
The next network I want to discuss is the 2-d lattice described in

Guerrier, Claire, John A. Hayes, Gilles Fortin, and David Holcman. “Robust Network Oscillations during Mammalian Respiratory Rhythm Generation Driven by Synaptic Dynamics.” *Proceedings of the National Academy of Sciences* 112, no. 31 (August 4, 2015): 9728–33. <https://doi.org/10.1073/pnas>

## 3 Simulation examples and network study

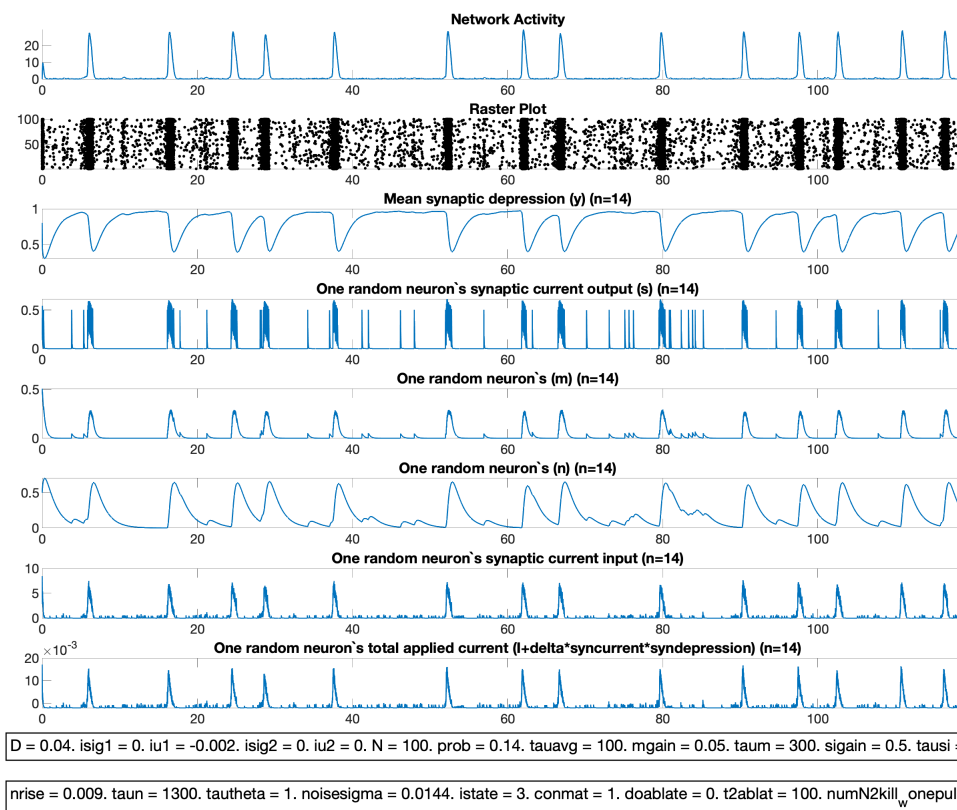
Let’s begin with the vanilla version of the model, with the ER network. The model is able to generate rhythmic bouts of spiking, where a single node spikes multiple times during a network-wide burst of activity. The neurons all have negative  $I_{\text{app}_i}$ , and are thus normally silent. Noise drives some of the nodes over threshold, and they begin to spike. This activity propagates throughout the network RAPIDLY and drives the upstroke of a burst. Synaptic depression begins to accumulate and the diminishing  $I_{\text{app}}$  terminates the burst. Synaptic depression must recover to some degree before another burst is possible. This last point, about recovery of synaptic depression and timing of the next burst, is an important one.

The model works in two different configurations. In one (with higher noise and slightly slower synaptic recovery), synaptic depression is not allowed to recover back to  $y = 1$  before the next burst is initiated. See the figure below.



In the top row, we see network activity, and below that a raster plot. But focus on the third row: the mean synaptic depression of the entire network. Notice it is not fully recovered before the next spike. As a result, the size of the following burst will be proportional to the duration of the preceding inter-burst interval [CITE]. I've tested this in the model myself, and do find this relationship. This dependency of burst size on preceding interval is not observed in the preB otC . And it has also been shown for the respiratory rhythm that synaptic depression appears to recover completely in the first half of the inter-burst interval [CITE]. So the above configuration of the network must be incomplete.

In the second configuration synaptic depression is allowed to recover, now noise drives the network to burst. But we can quickly see the network is behaving strangely. The problem is with the inter-event intervals. We find the intervals are exponentially distributed. This is because we are waiting for a random event that has some probability of happening at any given time step. Specifically, we are waiting for noise to drive us above some threshold. After waiting 5 minutes for a burst or an hour, the burst is not any more likely at the next moment. This behavior is also inconsistent with the real preB otC rhythm. So what gives!!?



at whats causing the burst.png

Well what if we were no longer waiting for a time-independent event, but a time-dependent one. An event that became more likely the longer you waited for it. It's possible that “a build up of excitability” in the network could take several seconds, or the second half of the inter-event interval. But the question is, “how much excitability can a dynamic network store?”

## 4 Where to store excitability in a network

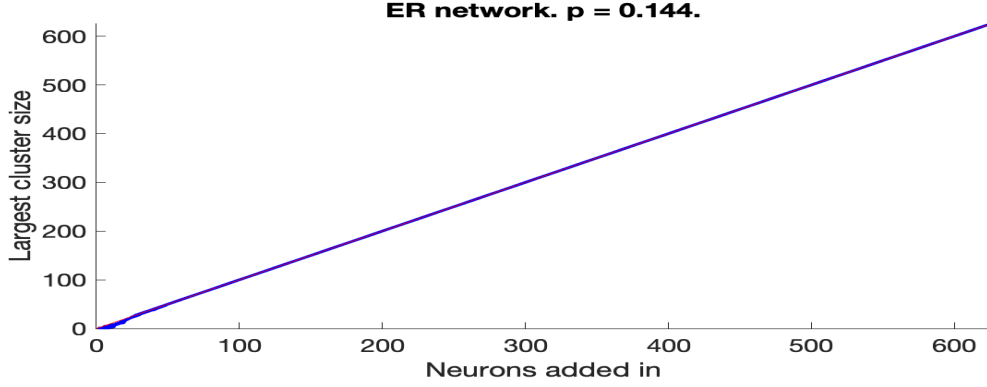
### 4.1 Network Topology

Our hypothesis is, under different network topologies, excitation can build and propagate through the network more slowly, and simply delay the network-wide burst that is inevitable. First, I looked to see at what rate the **giant component** comes online for different networks. The results were surprising. I will show two important examples here.

First, I build a network with some desired properties. Then I take all the nodes of the network offline. Next, one at a time, I reinstall a node and all its outgoing edges. After I add a node, I recalculate the largest component in the directed\* network.

Here is the result of this study for the ER network with  $p = 0.14$ :

\*The definition of the giant component in a **directed** graph requires more rigorous definition. The giant component we refer to here is the largest group of strongly connected components. In other words, if you start at node A, how many nodes can you pass through and still arrive back at A.



The size of the largest cluster (in blue) almost perfectly tracts the unity line  $y = x$ . That tells us, that anytime you add a random node into the network, it will likely already be a part of the giant component. This suggests there is minimal sub clustering of nodes. There are no secrets in the ER network of  $p = 0.14$ . What about other networks? Well one of the most extreme networks I found, when it comes to a giant component “coming online”, was from Guerrier 2015 [Real citation]. Their network is built on a 2-D lattice, where connections between neighbors is proportional to their absolute distance. For the math inclined, the probability of node  $i$  (at location  $(x_i, y_i)$ ) connecting to node  $j$  (at location  $(x_j, y_j)$ ) is

$$P_{i \rightarrow j} = e^{-d^2/(2s^2)} \quad (6)$$

where

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

In there case,  $s = 0.9$ , which leads to a mean in/out degree of  $\langle k_{\text{in/out}} \rangle \approx 3.9^\dagger$ .

---

<sup>†</sup>compare that to the ER network with 90 average connections.