Neural Signal Processing

Manu.Jayadharan

Department of Mathematics
IISER Mohali

NNMCB Internship 2015 Progress

Outline

1. Motivation
   - What is neural Signal Processing
   - Brain and Neuron Structure
   - Membrane Potential and Action Potential
2. Hodgkin Huxley Model
   - About the model
   - Simulation of the Model
3. Neuron Coupling
   - LIF(leaky integrate and fire) model
   - Simulations
4. Variable Topology Neural Network Simulator

Outline

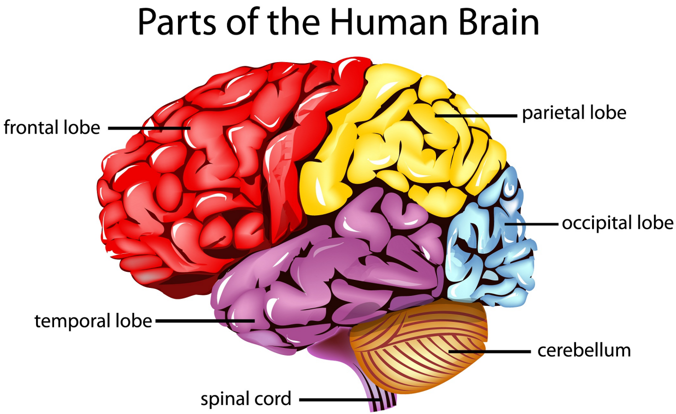
1. Motivation
   - What is neural Signal Processing
   - Brain and Neuron Structure
   - Membrane Potential and Action Potential
2. Hodgkin Huxley Model
   - About the model
   - Simulation of the Model
3. Neuron Coupling
   - LIF(leaky integrate and fire) model
   - Simulations
4. Variable Topology Neural Network Simulator

- Neurons can be can be considered as the most fundamental and core unit of the nervous system
- They are electrically excitable cells which helps in the propogation of information through electrical and chemical interactions within an organism
- Neurons can be sensory, motor or interconnecting
- Neurons interconnect to form neural network
- The sole purpose of neural signal processing is to understand how information is represented, transmitted and stored in the nervous system and develop the mechanism to record and interpret these signals

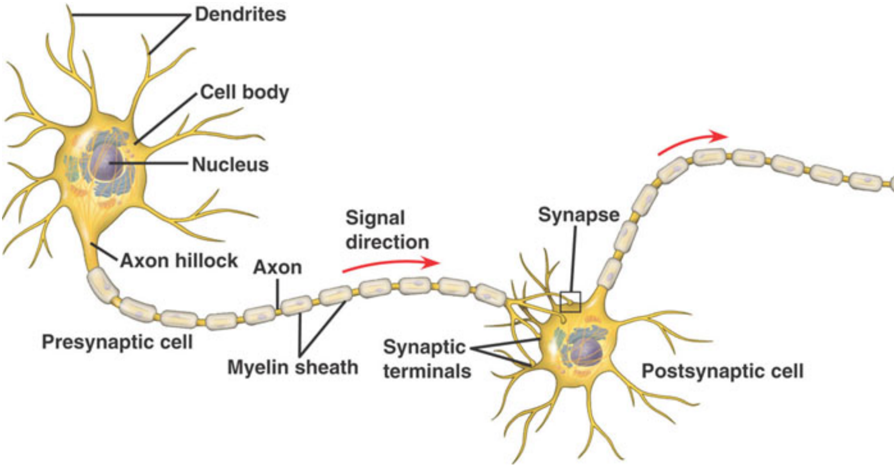
Outline

1. Motivation
   - What is neural Signal Processing
   - Brain and Neuron Structure
   - Membrane Potential and Action Potential
2. Hodgkin Huxley Model
   - About the model
   - Simulation of the Model
3. Neuron Coupling
   - LIF(leaky integrate and fire) model
   - Simulations
4. Variable Topology Neural Network Simulator

Humman brain consists of approximately 10 billion



neurons


Dendrites
Cell body
Nucleus
Axon hillock
Axon
Signal direction
Synapse
Presynaptic cell
Myelin sheath
Synaptic terminals
Postsynaptic cell

Outline

1. Motivation
   - What is neural Signal Processing
   - Brain and Neuron Structure
   - Membrane Potential and Action Potential
2. Hodgkin Huxley Model
   - About the model
   - Simulation of the Model
3. Neuron Coupling
   - LIF(leaky integrate and fire) model
   - Simulations
4. Variable Topology Neural Network Simulator

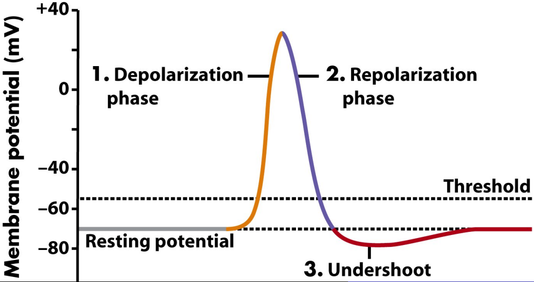
Membrane Potential

- Neurons usually maintain a difference in electric potential across its cell membrane
- Unequal distribution of charged ions(Na+,K+,Ca2+, cl- etc) accounts for this membrane potential difference
- Cell membrane has selectively permeable ion channels which facilitates the movement of corrensponding ions

$$V_m = V_{in} - V_{out}$$

Action Potential

- signals are processed across an excitable neuron in the form of propogating spikes or action potential.
- A slight decrease in membrane potential makes membrane more permeable to Na+ than K+, and a positive feedback loop facilitates further depolarization to form a sharp peak in the voltage across the membrane.
- Action potential is an all or nothing event.



Outline

1. Motivation
   - What is neural Signal Processing
   - Brain and Neuron Structure
   - Membrane Potential and Action Potential
2. Hodgkin Huxley Model
   - About the model
   - Simulation of the Model
3. Neuron Coupling
   - LIF(leaky integrate and fire) model
   - Simulations
4. Variable Topology Neural Network Simulator

H & H model

- The Hodgkin-Huxley Model is a mathematical model for the initiation and propogation of action potential across a neurons based on the dynamic behaviour of the conductance of different ions across the membrane
- It is a set of differential equations which is mostly derived from emperical data from different set of Voltage clamp experiments conducted on squid giant axon.
- Alan Lloyd Hodgkin and Andrew Huxley received the Nobel Prize in phhysiology or Medicine for the same

Differential Equations

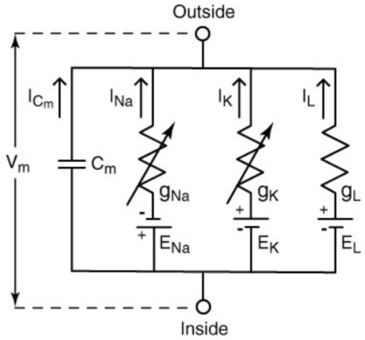
$$i_m = i_C + i_l + i_{Na} + i_K$$

where,

$$i_l = g_l(V_m - E_l)$$

$$i_{Na} = g_{Na}(V_m - E_{Na})$$

$$i_K = g_K(V_m - E_K)$$



$$i_m = C_m \frac{dV_m}{dt} + g_l(V_m - E_l) + g_{Na}(V_m - E_{Na}) + g_K(V_m - E_K)$$

$$g_{Na} = \bar{g}_{Na} m^3 h$$

$$g_K = \bar{g}_K n^4$$

Modelling of gating variables

The gating variables satisfy the nonlinear ordinary differential equations

$$\dot{m} = \frac{m_\infty(V) - m}{\tau_m(V)}, \qquad \dot{n} = \frac{n_\infty(V) - n}{\tau_n(V)}, \qquad \dot{h} = \frac{h_\infty(V) - h}{\tau_h(V)}.$$

The six functions $\tau_X(V)$ and $X_\infty(V)$, $X \in \{m, n, h\}$, are obtained from fits with experimental data. It is common practice to write

$$\tau_X(V) = \frac{1}{\alpha_X(V) + \beta_X(V)}, \qquad X_\infty(V) = \alpha_X(V)\tau_X(V)$$

The details of the final Hodgkin-Huxley description of nerve tissue are completed with:

$$\boxed{\begin{aligned}
\alpha_m(V) &= \frac{0.1(V+40)}{1-\exp[-0.1(V+40)]} & \alpha_h(V) &= 0.07\exp[-0.05(V+65)] \\
\alpha_n(V) &= \frac{0.01(V+55)}{1-\exp[-0.1(V+55)]} & \beta_m(V) &= 4.0\exp[-0.0556(V+65)] \\
\beta_h(V) &= \frac{1}{1+\exp[-0.1(V+35)]} & \beta_n(V) &= 0.125\exp[-0.0125(V+65)]
\end{aligned}}$$

$C = 1\mu\text{F cm}^{-2}$, $g_L = 0.3\text{mmho cm}^{-2}$, $g_K = 36\text{mmho cm}^{-2}$, $g_{Na} = 120\text{mmho cm}^{-2}$, $V_L = -54.402\text{mV}$, $V_K = -77\text{mV}$ and $V_{Na} = 50\text{mV}$. (All potentials are measured in mV, all times in ms and all currents in μA per cm^2).

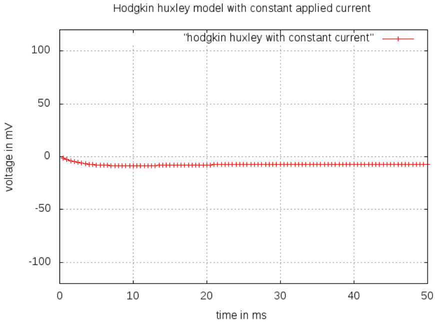
Outline

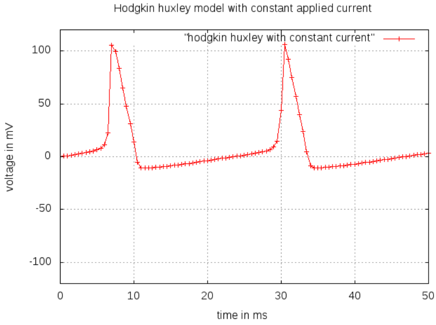
1. Motivation
   - What is neural Signal Processing
   - Brain and Neuron Structure
   - Membrane Potential and Action Potential
2. Hodgkin Huxley Model
   - About the model
   - Simulation of the Model
3. Neuron Coupling
   - LIF(leaky integrate and fire) model
   - Simulations
4. Variable Topology Neural Network Simulator

Simulation softwares and strategies

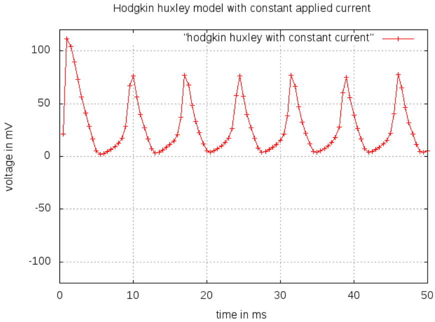
- Lot of open sourced and commercial advanced softwares are available for the simulation and visualisation of the Model none of which is being being for my simulations
- All codes are written using Fortran95 and all the packages and modules are self written.
- Simple Euler method is invoked for solving the set of coupled differential equations which is later improved using runge-kutta of order 4
- Facility for making an animation of plotting of the simulated data is incorporatd into the code with the help of gnu plot and ffmpeg.
- The complete code and animations are available at https://www.dropbox.com/sh/7bd3hueoui1kcfq/AAB2iR21nXP1owC tnhOmUf65a?dl=0

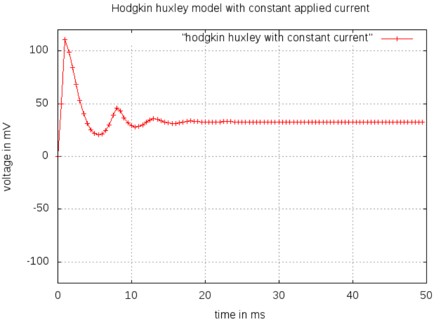
Simulation with current 0 and 4mA resp





Simulation with current 50mA and 100mA resp





Outline

1. Motivation
   - What is neural Signal Processing
   - Brain and Neuron Structure
   - Membrane Potential and Action Potential
2. Hodgkin Huxley Model
   - About the model
   - Simulation of the Model
3. Neuron Coupling
   - LIF(leaky integrate and fire) model
   - Simulations
4. Variable Topology Neural Network Simulator

- LIF model is a comparitively simple model in which the ionic conductances are not explicitley incorporated
- It is based on a simple RC circuit
- instead of modelling the spikes internally, in this model the spikes are artificially introduced when the membrane potential reaches a threshold votlage V_{th} and resetting V_m.
- The basic equation of an LIF model can be given by

$$C_m \frac{dV}{dt} = -g_L(V - E_L) + I_{app}$$

where if $V(t_*^-) = V_{th}$, then $V(t_*^+) = V_{reset}$.

Modelling the coupling

- The coupling between two neurons is modelled by incorporating an ohmic conductance between two neurons into the LIF model to get the following set of differential equations

$$\begin{cases} C_m \dfrac{dV_1}{dt} = -g_L(V_1 - E_L) + I_{app} + g_c(V_2 - V_1) \\ C_m \dfrac{dV_2}{dt} = -g_L(V_2 - E_L) + I_{app} + g_c(V_1 - V_2) \end{cases}$$

- A completely analytic solution for this set of equations can be done with the reduction of order using theory of weakly coupled oscillators.

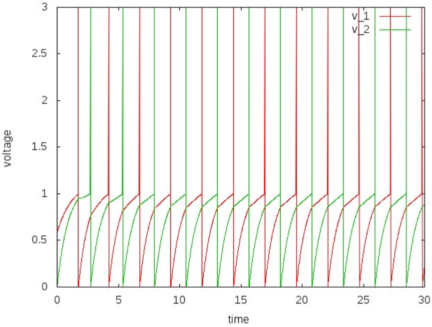
Outline

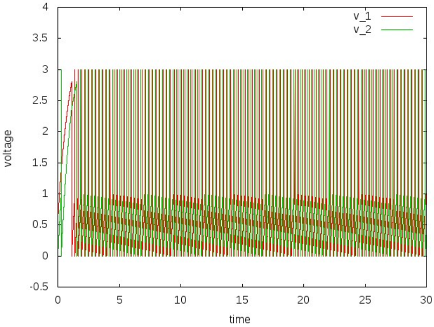
1. Motivation
   - What is neural Signal Processing
   - Brain and Neuron Structure
   - Membrane Potential and Action Potential
2. Hodgkin Huxley Model
   - About the model
   - Simulation of the Model
3. Neuron Coupling
   - LIF(leaky integrate and fire) model
   - Simulations
4. Variable Topology Neural Network Simulator

Software used for Simulation

- While analytic solution can be derived, numerical solution can be derived from the coupled differential equations to get an insight of the problem.
- Again all codes are written using Fortran95 and all the packages and modules are self written.
- Runge-kutta of order 4 is invoked for solving the set of coupled differential equations.
- Facility for making an animation of plotting of the simulated data is incorporatd into the code with the help of gnu plot and ffmpeg.
- The complete code and animations are available at https://www.dropbox.com/sh/qs4f3on3oo1fcvz/AAAu5XrPbo_nFdw3pGyEBUDea?dl=0

Simulations with weak and Strong applied current





Neural Network

- The complexity of Neural Networks arises not from the specialisation of neurons, but from their topology and nature of interconnection.
- Understanding how the topology of neural network affect signal propogation seems to be a very interesting filed of study as it might give some insight into how the same kind of action potential can be used to propogate complex iformation from the brain.

Mathematical Modelling

- Variable Topology Neural Network Simulator(VTNNS) is a simulator which is based on a Mathematical model that can simulate or mimic the behaviour of signal propogation in a neural network with any specified topology and synaptic relation.

$$\tau \frac{dV}{dt} = -(V - V_0) \tag{1}$$

$$\tau \frac{dV}{dt} = -(V - V_0) - g^{+}(t)(V - E^{+}) - g^{-}(t)(V - E^{-}) \tag{2}$$

$$\tau_s \frac{dg^i}{dt} = -g^i \tag{3}$$

Solution of the Equation

$$V(t) = -\rho(1-\tau_s, \tau_s g(t))\tau_s E_s g(t)$$

$$+exp(\tau_s(g(t)-g(0))-t)(V(0)+\rho(1-\tau_s, \tau_s g(0))\tau_s E_s g(0)$$

where, $\rho(a,b) = e^b x^{-a}\gamma(a,b)$ with $\gamma(a,b) = \int_0^b e^{-t} t^{a-1} dt$ which is the incompete gamma integral.

- Fast computing numerical libraries are available for efficient calculation of the incomplete gamma integrals.

VTNNS without time delay

1. Maintain a sorted table of firing time of various neurons in the network.
2. Updating the neuron after it fires: If a neuron is to be first at time t and if $t_0$is the last time of update of the neuron,
   1. $V \to V_{reset}$
   2. $g \to g * exp(\frac{-(t-t_0)}{\tau_s})$
3. Updating neuron j after i neuron fires: If a neuron fires at time t, then for each other neuron with last update time t_0,
   1. $V \to V(t - t_0)$
   2. $g \to g * exp(\frac{-(t-t_0)}{\tau_s})$
   3. $E_s \to \frac{gE_S + pw_{ij} + q|w_{ij}|}{g + |w_{ij}|}$ where $p = \frac{E^+ - E^-}{2}$and $q = E^+ + E^-$
   4. $g \to g + |w_{ij}|$
4. Update the firing time table of neurons

VTNNS with time delay

1. Maintain a sorted table of firing time of various neurons in the network.
2. Updating the neuron after it fires: If a neuron is to be first at time t and if $t_0$is the last time of update of the neuron,
    1. $V \rightarrow V_{reset}$
    2. $g \rightarrow g * exp(\frac{-(t-t_0)}{\tau_s})$
3. Updating neuron j after i neuron fires: If a neuron fires at time t, then for each other neuron with last update time t_0,
    1. $V \rightarrow V(t - t_0)$
    2. $g \rightarrow g * exp(\frac{-(t-t_0)}{\tau_s})$
    3. $E_s \rightarrow \frac{gE_S+pw_{ij}+q|w_{ij}|}{g+|w_{ij}|}$ where $p = \frac{E^+-E^-}{2}$and $q = \frac{E^++E^-}{2}$
    4. $g \rightarrow g + |w_{ij}|$
4. Update the firing time table of neurons

Spike tests

1. Check $E_s > V_t$, if yes then
2. Check $g_0 > g_{min}$, if yes then
3. Check $V(g_{min}) > V_{th}$

Neuron will fire iff the initial conditions passes all the three spike tests above

Coding for VTNNS

- Simulator with and without time lag in synaptic conductance is coded separatley . We have used the algorithms to make two VTNNS setup: one in which the synaptic interaction is very strong and a few neurons can be set to fire once and check the propogation of the signal in the network, the second in which one of the neurons in the network act as an oscilattor(source of signal) with a specified frequency
- All codes are written in *Fortran*95 considering the efficiency of the same to handle huge arrays and comutational efficiency while doing huge calculations. This give us an advantage to simulate large network within feasible computation time. Disadvantage being the difficulty in debugging and the code getting lengthy.

https://www.dropbox.com/sh/bkju25ktl9jip29/AABDjSxazoBQTcBQp13Xizema?dl=0

Star Topology

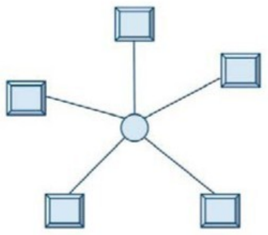
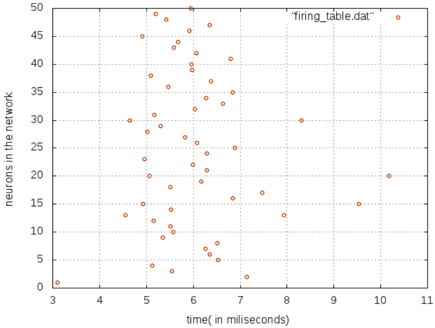
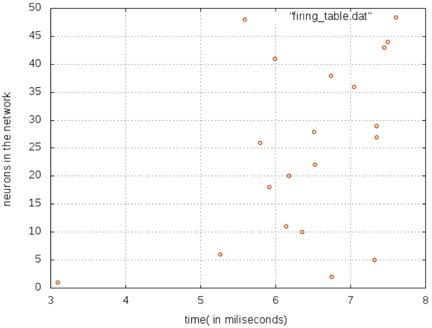
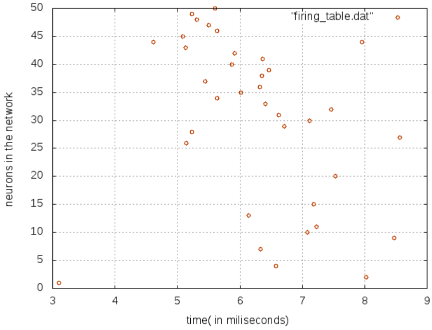


Figure: star topology


"firing_table.dat"
neurons in the network
time(in miliseconds)
0
5
10
15
20
25
30
35
40
45
50
3
4
5
6
7
8
9
10
11


"firing_table.dat"
neurons in the network
time(in miliseconds)
0
5
10
15
20
25
30
35
40
45
50
3
4
5
6
7
8



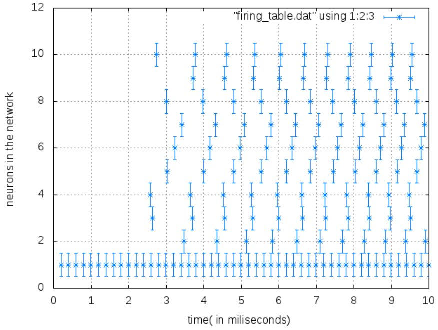


Figure: Raster plot for neural network with star topology simulated using VTNNS without time delay in propogation of spike(top left) and with time delay(top right). $w_{1j} = 1.2$(neuron 1 is connected to all other neurons with an excitory sypansis) .. Image on bottom left shows the same simulation with parameter w_{1j} increased to 1.6 for neurons 25 to 50

Fully connected Topology

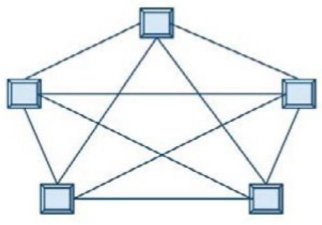


Figure: fully connected topology

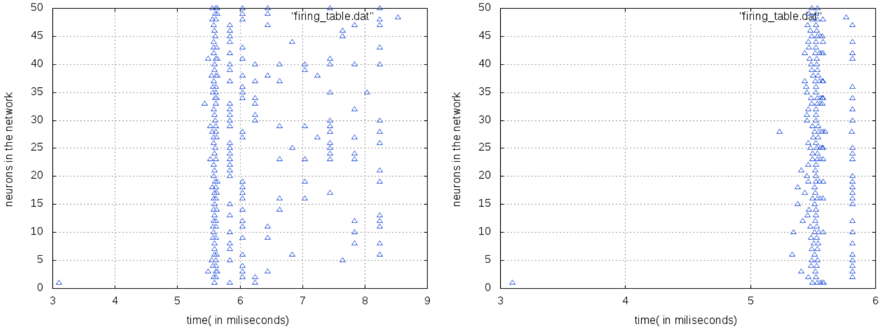


Figure: Raster plot for neural network with fully connected topology simulated using VTNNS without time delay in propogation of spike(top left) and with time delay(top right).

Ring Topology

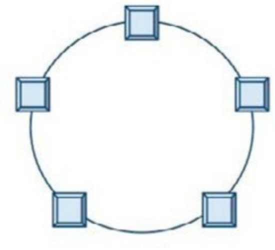


Figure: ring topology

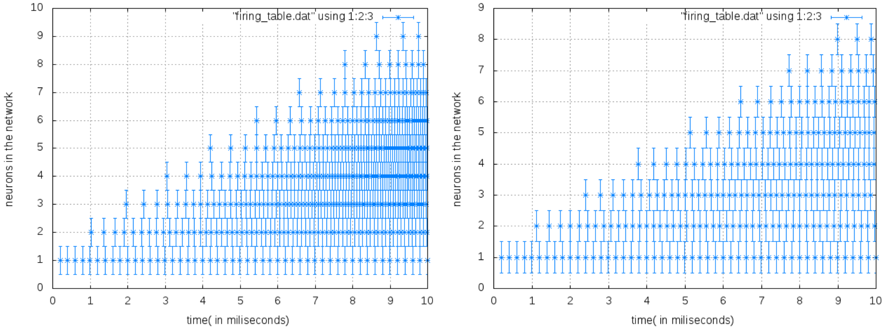


Figure: Raster plot for neural network with ring topology simulated using VTNNS without time delay in propogation of spike(left) and with time delay(right). $w_{i,i+1} = 1.0$ for $i = 1,..,9$ and $w_{10,1} = 1.0$(each neuron is connected to the next neuron in the order in a circle with an excitory sypansis)

Miscellaneous

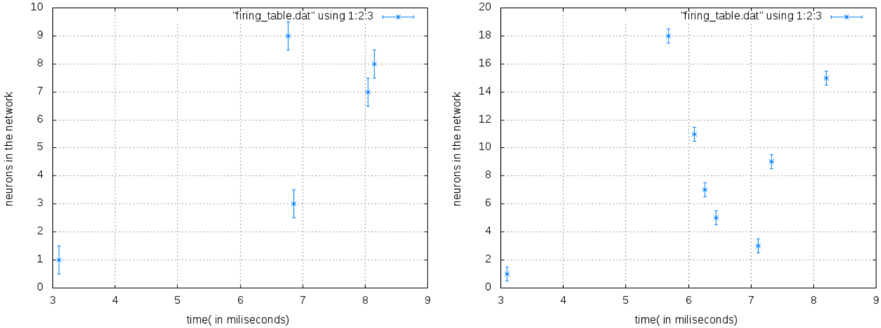


Figure: A wheel spoke kind of toplogy is considered by joining the peripheral neurons in a star shaped Neural netowork using inhibitory synapse.Raster plot of the same is simulated using VTNNS without time delay in propogation of spike(left) and with time delay(right). $w_{1j} = 1.0$(neuron 1 is connected to all other neurons with an excitory sypansis) and $w_{i,i+1} = -1.0$

Random Network

- VTNNS can be easily used to make a random network containing large number of neurons.
- Probabilistic distributions and functions could be used to define the synaptic relatin between these neurons and also spiking could also be introduced at various nodes following some particular distribution.

Improving the VTNNS

- Various kinds of synapstic transmitters can be incorporated in the model using multidimensional arrays for the weight matrix explaining synaptic relations.
- Synaptic relation between neurons in the network can be made to evolve with time which can be done by using functions to change entries of the weight matrix considering the current state of a neuron and the nature of the incomign spikes. This needs further work, proper experiments might be needed to fit functions which actually mimics the behaviour of the neural transmitters and their binding to various receptors in the synaptic junction.

- Each neuron or different section of neurons in the network can be assigned different synaptic time constants which again requires experiments to get an idea of how these values should be assigned to make it useful.
- A more probabilistic model can be used to model VTNNS in which case the whole algorithm need to be rewritten.
- I case of using VTNSS for some studies, various graphical interfaces can be used so that VTNNS will automatically generate an image of the toplogy that it is being simulated and also an animation which gives a clear view on how the neurons are behaving inside the simulated network.

- The Effects of Potassium Currents on the Synchronization of Electrically Coupled Neural Oscillators By Colin Baker Middleton
- Exact Simulation of Integrate-and-Fire Models with Synaptic Conductances, Romain Brette
- Lecture presentations of Dr.Caleb Kemere, Rice university
- Lecture notes on Hodgkin Huxley model by Dr.Stephen coombes, university of Nottingham