University of Waterloo
Faculty of Engineering
Department of System Design Engineering

SYDE 556 - Assignment 4
Nengo and Dynamics

200 University Avenue West
Waterloo, Ontario, Canada
N2L 3G1

Prepared by
Sugandha Sharma
ID: 20401693
userid: s72sharm
4B Electrical Engineering
19 March 2015

Table of Contents

List of Figures

1 Building an ensemble of neurons

Made a new model and inside that model made an ensemble of neurons with 100 neurons, representing a 1-dimensional space. The intercepts were chosen to be between -1 and 1, and the maximum firing rates to be between 100Hz and 200Hz. τRC = 0.02s and τref = 0.002s.

a) Tuning Curves

Plot of the tuning curves (firing rate of each neuron for different x values between -1 and 1)

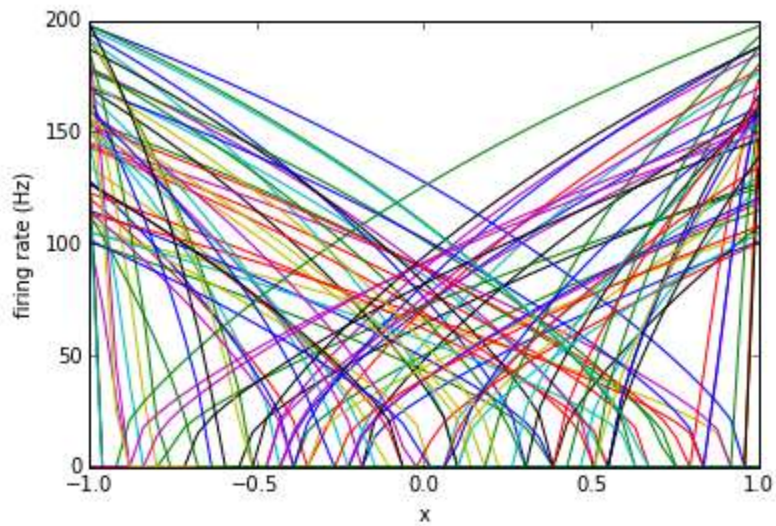

Figure 1. Plot of the tuning curves of 100 LIF neurons

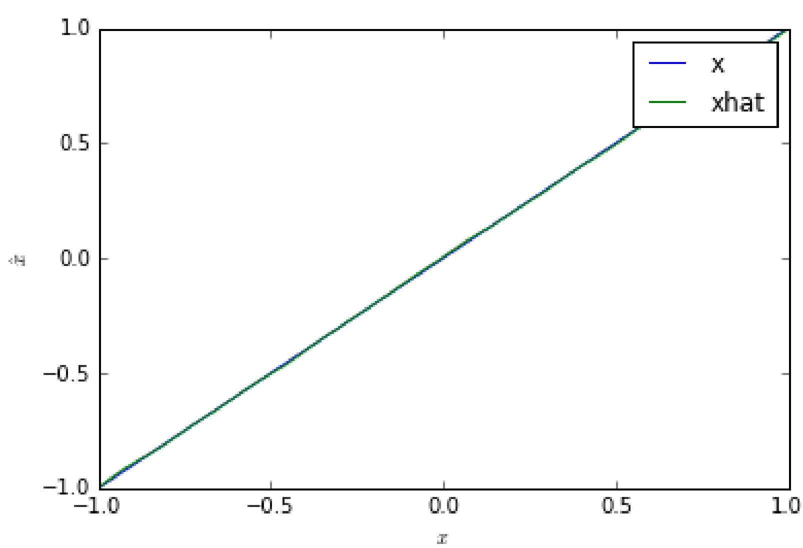

Figure 2. Representation Accuracy Plot

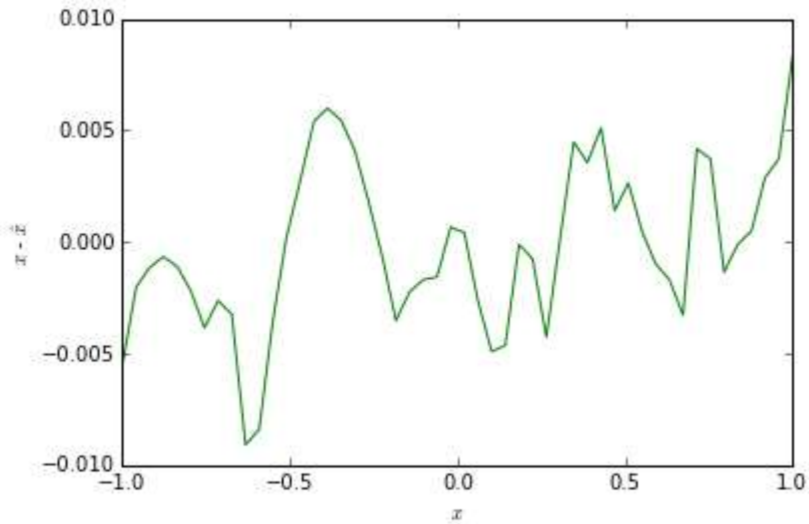

Figure 3. Representational Error Plot

RMSE with 100 neurons is 0.00527525

b) Effect of radius on RMSE

As the radius of the neurons increases, the RMSE is seen to increase linearly. This is because the radius is indicative of the amount of information which the neurons would represent. Hence as the radius is increased, the same number of neurons would need to represent more information and thus the Root mean squared error increases. In other words, 100 neurons would be better at representing information with radius = 1 than representing information with radius = 4 due to the increase in the amount of information to be represented. This effect can also be seen from the tuning curves which show a smaller change in firing rate for a change in x. This means that for a given change in stimulus, the firing rates do not change as much, so they do not represent the stimulus that accurately as the radius is increased.

Radius	**RMSE**
1	0.00591314
2	0.58566571
3	1.17266734
4	1.75979323

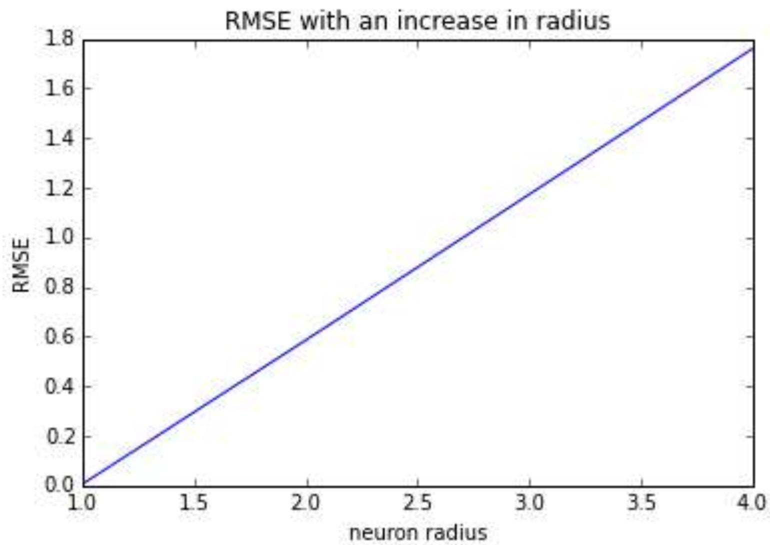

Figure 4. Effect of increasing radius on RMSE

c) Effect of tau_ref on RMSE

Tau_ref is the refractory period which is the amount of time it takes for an excitable membrane to be ready for a second stimulus once it returns to its resting state following excitation (recovery time). Thus basically, tau_ref represents the minimum time between consecutive spikes.

As tau_ref increases, the RMSE is found to increase in general. This is because as the refractory period of the neurons increases, the neurons saturate earlier and take more time to recover (slower/less linear response). This leads to a smaller change in the firing rate for a given change in in stimulus. Thus tau_ref sets an upper limit on the firing rate of neurons. The maximum firing rate of a neuron should be below the inverse refractory period.

With a lower tau_ref, smaller change in stimulus produce larger change in the firing rate and hence provides a more accurate representation.

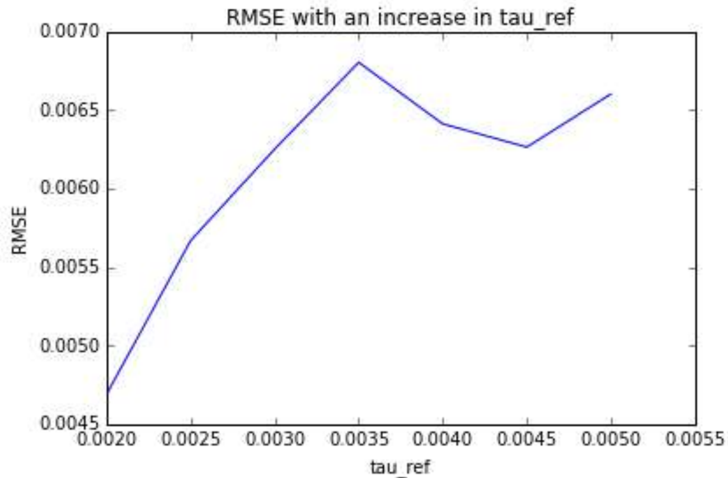

Figure 5. Effect of changing tau_ref on RMSE

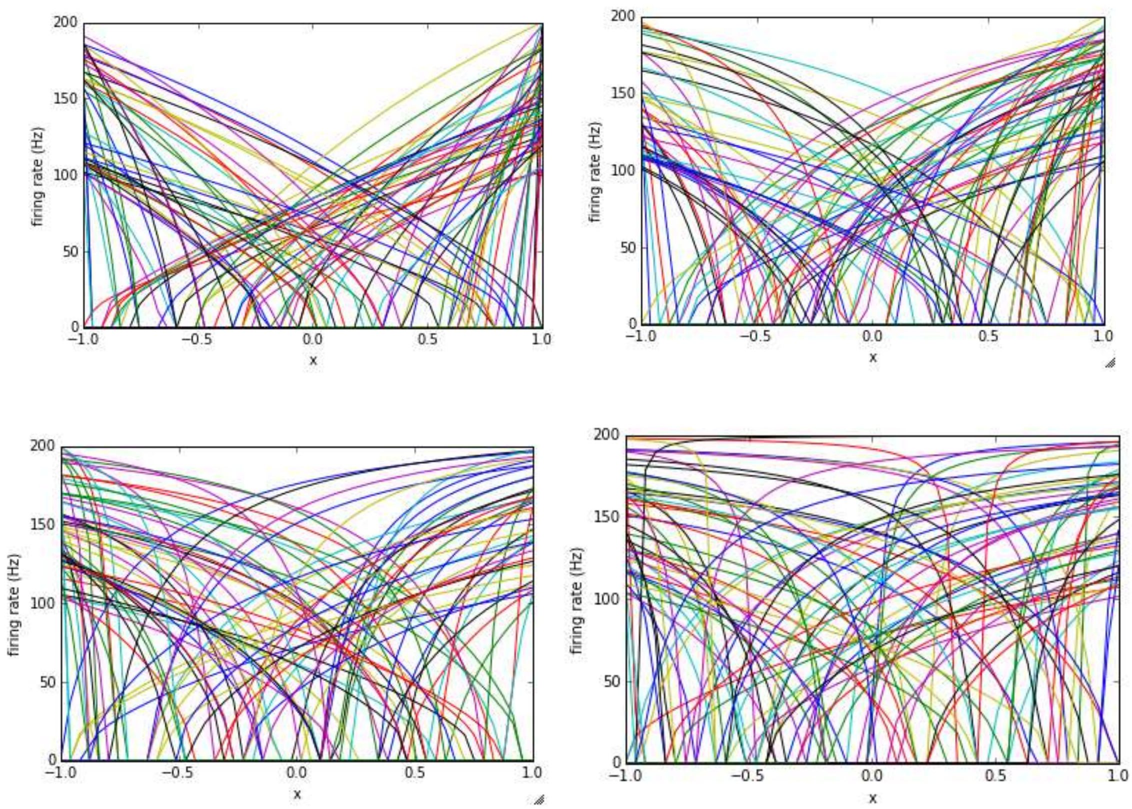

Figure 6. Tuning curves as tau_ref increases (from top left to bottom right)

The tuning curves show that the neuron response becomes slower/less linear as tau_ref increases as shown in figure 6. In other words there is a smaller change in the firing rate for a given change in the x as tau_ref is increased due to the increase in response time of the neurons.

d) Effect of tau_rc on RMSE

Tau_rc is the membrane RC time constant which is the product of the membrane's resistance and capacitance. As tau_rc increases, the slope of the LIF curve increases which means that there is greater change in the firing rate of the neurons for a given change in the stimulus. Thus the rmse is found to decrease in general as tau_rc increases since a longer RC time constant results in more linear/faster tuning curves. This results in a reduced error while fitting the response functions to a straight line (which is what we are doing when we find the decoders).

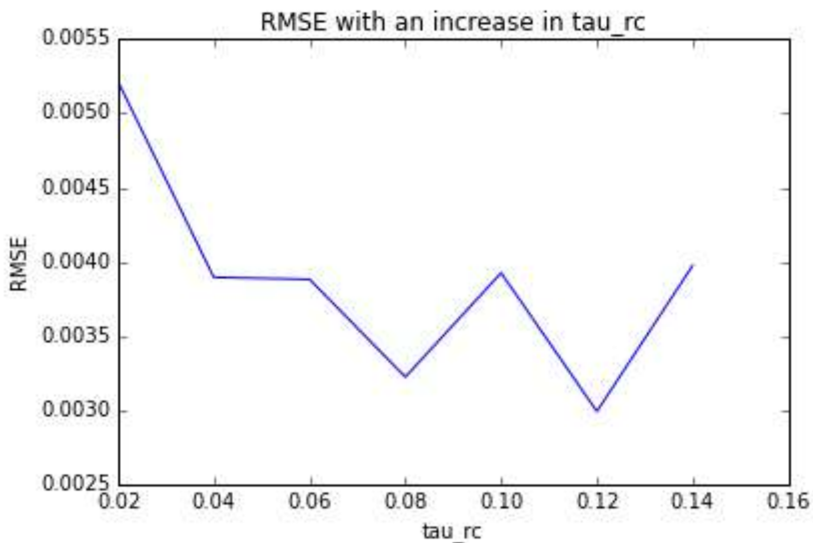

Figure 7. Effect of changing tau_rc on RMSE

The tuning curves show a faster/more linear response of the neurons with an increase in tau_rc. In other words, there is greater change in the firing rate of the neurons for a given change in the x when tau_rc is increased, thus resulting in a more accurate representation.

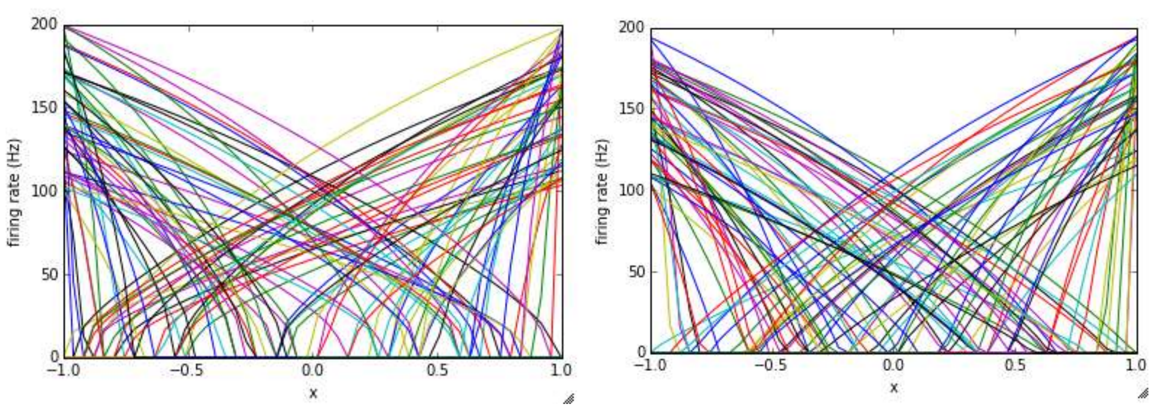

Figure 8. Tuning curves as tau_rc increases (from left to right)

2. Connecting Neurons

Made a second ensemble of neurons having the same parameters as the first ensemble of neurons (from the section 1), but have only 50 neurons in it. Connected the first ensemble to the second such that it computes the identity function, using a post-synaptic time constant of 0.01.

a) Response to a constant input

Provided an input to the system that is a value of 1 for 0.1s.

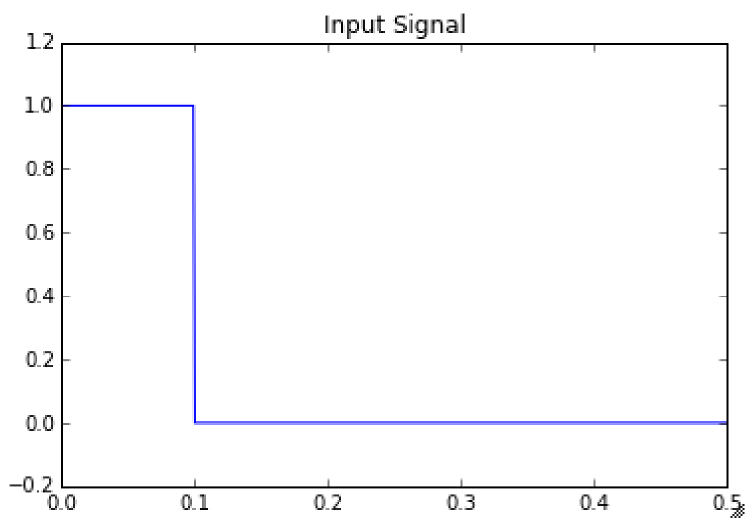

Figure 9. Constant input signal

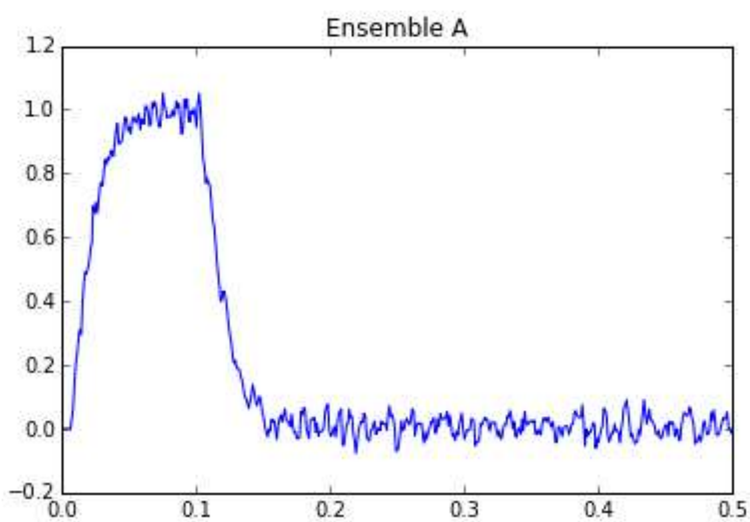

Figure 10. Decoded value from the first ensemble

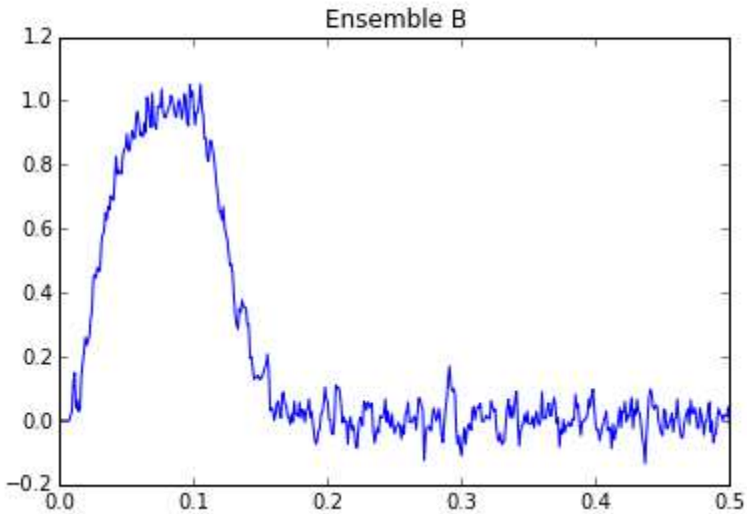

Figure 11. Decoded value from the second ensemble

b) Computing a transformation y = 1-2x

Provided an input to the system for 0.1s for computing the transformation y = 1 - 2*x.

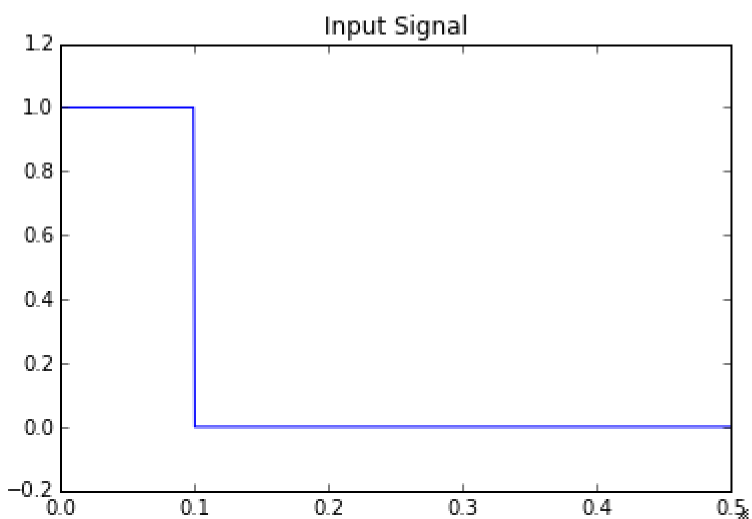

Figure 12. Constant Input Signal

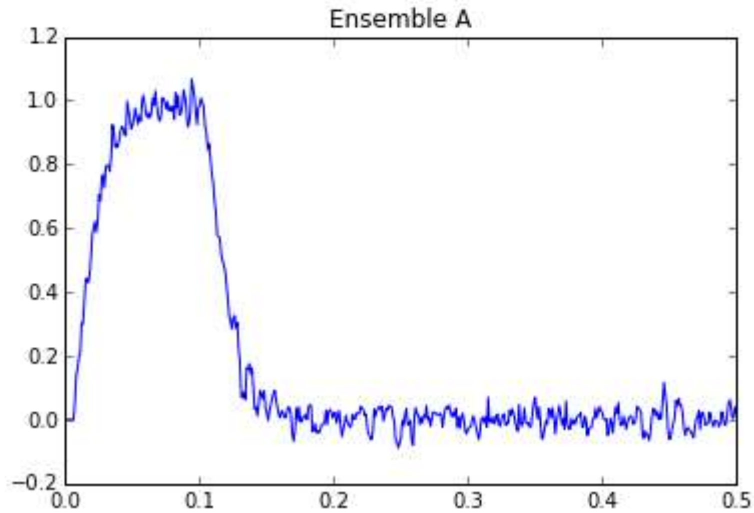

Figure 13. Decoded value from the first ensemble

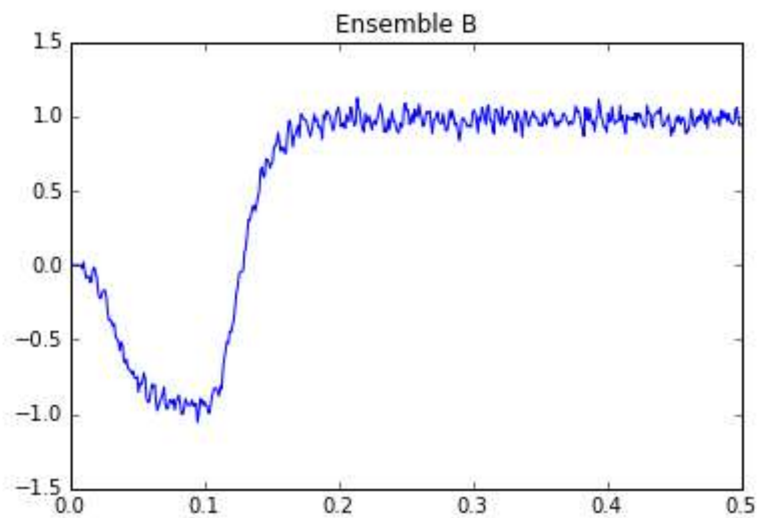

Figure 14. Decoded value from the second ensemble

3. Dynamics

Built a neural integrator which consists of one ensemble, one input, a connection from the input to the ensemble, and a connection from the ensemble back to itself. The ensemble has 200 neurons and the same parameters as in section 1. The post-synaptic time constant of the recurrent connection is 0.05, and the post-synaptic time constant of the input is 0.005.

a) Input and the value represented by the ensemble

The expected ideal result of the integral 0.9*x , i.e., the integral of a constant should be a linear function. The final value should be the area under the input signal which is (1-0.04) * 0.9 = 0.864.

Thus ideally, the result should linearly reach upto 0.864 and then stay constant. The result obtained shown in figure 15 is very close to the expected result, however the integral reaches up to around 0.9 instead of 0.864. Moreover, there is a slight drift since the neurons have a property of gradually forgetting the past inputs. Nevertheless, it can still be said that the value is being approximated quite well by the integrator.

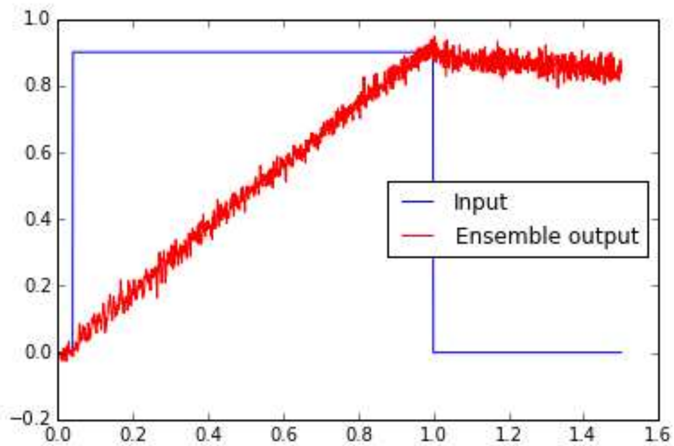

Figure 15. Input and the value represented by the ensemble

b) Changing the neural simulation to Rate Model

While using rate neurons, the system doesn't have to deal with the spikes and the noise that they add to the system. Thus using rate neurons, there is much less noise in the system as compared to section a). Due to less noise, the result obtained in this case seems to be closer to the ideal value of 0.864.

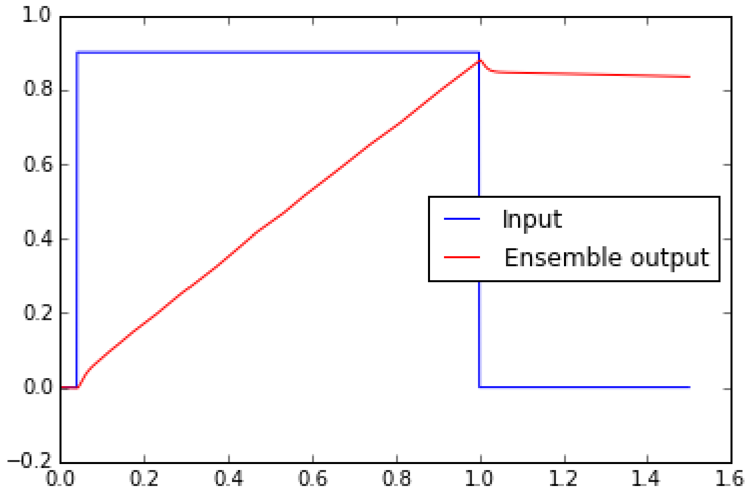

Figure 16. Input and the value represented by the ensemble using rate model

c) Returning to the Spiking Model

The expected ideal result of the integral is a linear function 0.9*x which should reach a final value equal to the area under the input signal which is (0.16-0.04) * 0.9 = 0.108 and then stay constant.

From figure 17, it can be seen that the final value of 0.108 is barely reached and there is a huge drift instead of the final value remaining constant. Thus, the results are worse than in section a). This is because the input was provided for a very short period of time in this case due to which the dynamics of the system had a hard time catching up with the fast impulse.

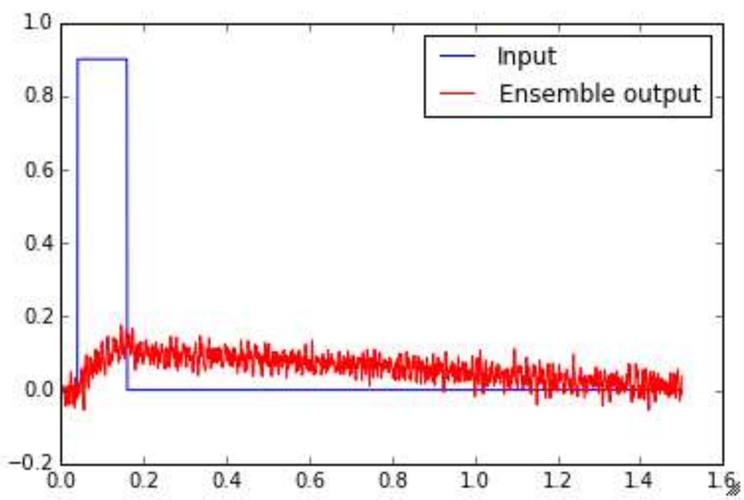

Figure 17. Input and the value represented by the ensemble using spiking model

d) Ramp Input

The ramp can be expressed by the equation y = 2x. Hence the expected ideal result of the integral is x^2 the final value should be equal to the area under the input signal which is 0.5* 0.45*0.9 = 0.2025. Thus the result should reach upto 0.2025 and then stay constant.

The result obtained shown in figure 18 is very close to the expected result since the ensemble output is a quadratic function in the beginning and ultimately settles around 0.2025. However, there is a slight drift since the neurons have a property of gradually forgetting the past inputs. Thus the ensemble ends up representing the final value which is approximately = 0.2025 since that is what the integral of the ramp evaluates to.

The ideal equation for the curve traced out by the ensemble is:

$$y = \begin{cases} x^2 & \text{for } 0 < x < 0.45 \\ 0.2025 & \text{for } x >= 0.45 \\ 0 & \text{Otherwise} \end{cases}$$

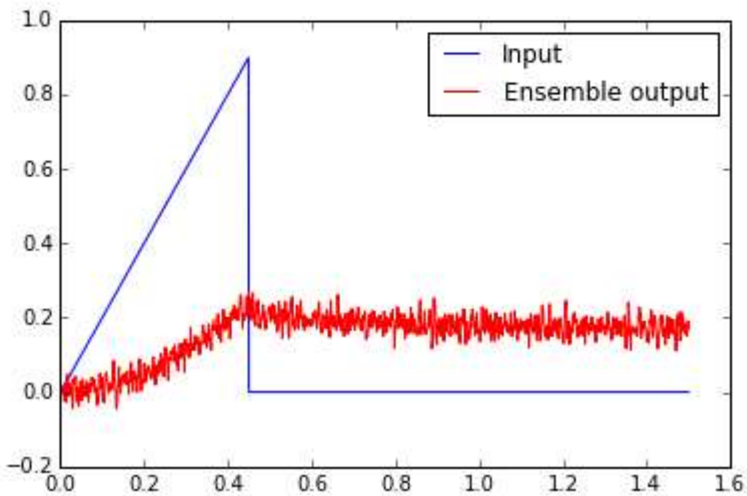

Figure 18. Ramp input and the value represented by the ensemble

e) Sinusoidal Input

Integration of 5*sin (5t) = (5/5) *-cos(5t) = - cos(5t). This should ideally be the value represented by the ensemble.

However, as shown in figure 19, the model doesn't approximate this equation very well. This is because the ensemble output is lagging behind the ideal value due to the delay caused by the post synaptic time constant. Thus the model's behaviour is not exactly the same as the expected ideal behaviour of an integrator. However, this lag in the output is expected for a biologically plausible model.

Figure 19 shows the decoded value from the ensemble using:

- The probe synapse = 0.001 (top plot)
- The probe synapse = 0.01 (bottom plot)

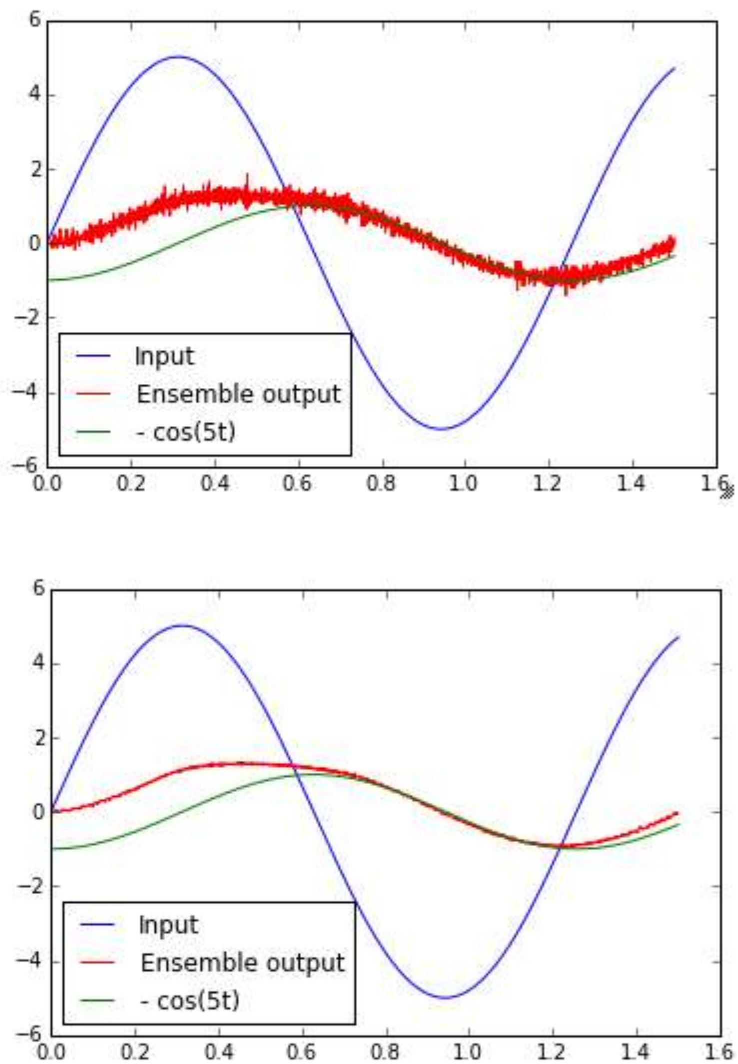

Figure 19. Sinusoidal input and the value represented by the ensemble

f) Non Linear Dynamical Systems

i) Rossler Attractor

I have modelled a non-linear dynamical system expressed by the following differential equations:

$$x'(t) = -(y(t) + z(t))$$

$$y'(t) = x(t) + ay(t)$$

$$z'(t) = b + x(t)z(t) - cz(t)$$

This system is known as Rossler Attractor System.

Using the mapping $x \Rightarrow x_0$, $y \Rightarrow x_1$, $z \Rightarrow x_2$, these equations can be written as follows:

$\dot{x}_0 = -(x_1 + x_2)$
$\dot{x}_1 = x_0 + a*x_1$
$\dot{x}_2 = b + x_0*x_2 - c*x_2 = b - x_2*(c-x_0)$

Expected Behaviour of the system: Figure 20 taken from
http://mathworld.wolfram.com/RoesslerAttractor.html

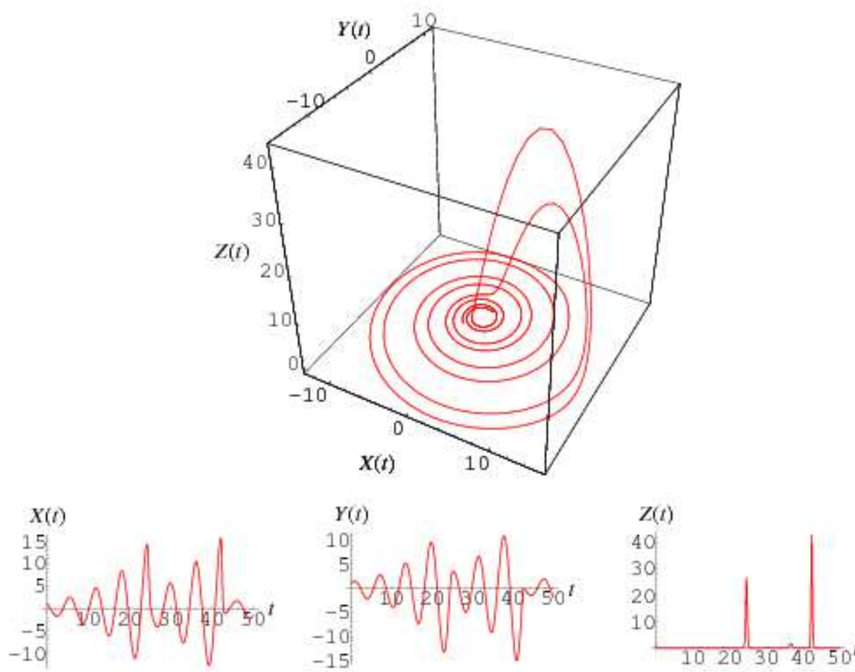

Figure 20. Expected Behaviour of a Rossler Attractor

It can be seen from case1 and case2 below that on selecting the right parameters, the system behaviour is similar to the expected behaviour i.e., the system acts as an attractor.

CASE 1:

Parameters Used:
tau = 0.1
a = 0.5
b = 1
c = 13

The system acts as an attractor as expected (shown in figure 21). The state value curves are also very similar to the expected curves shown in figure 20.

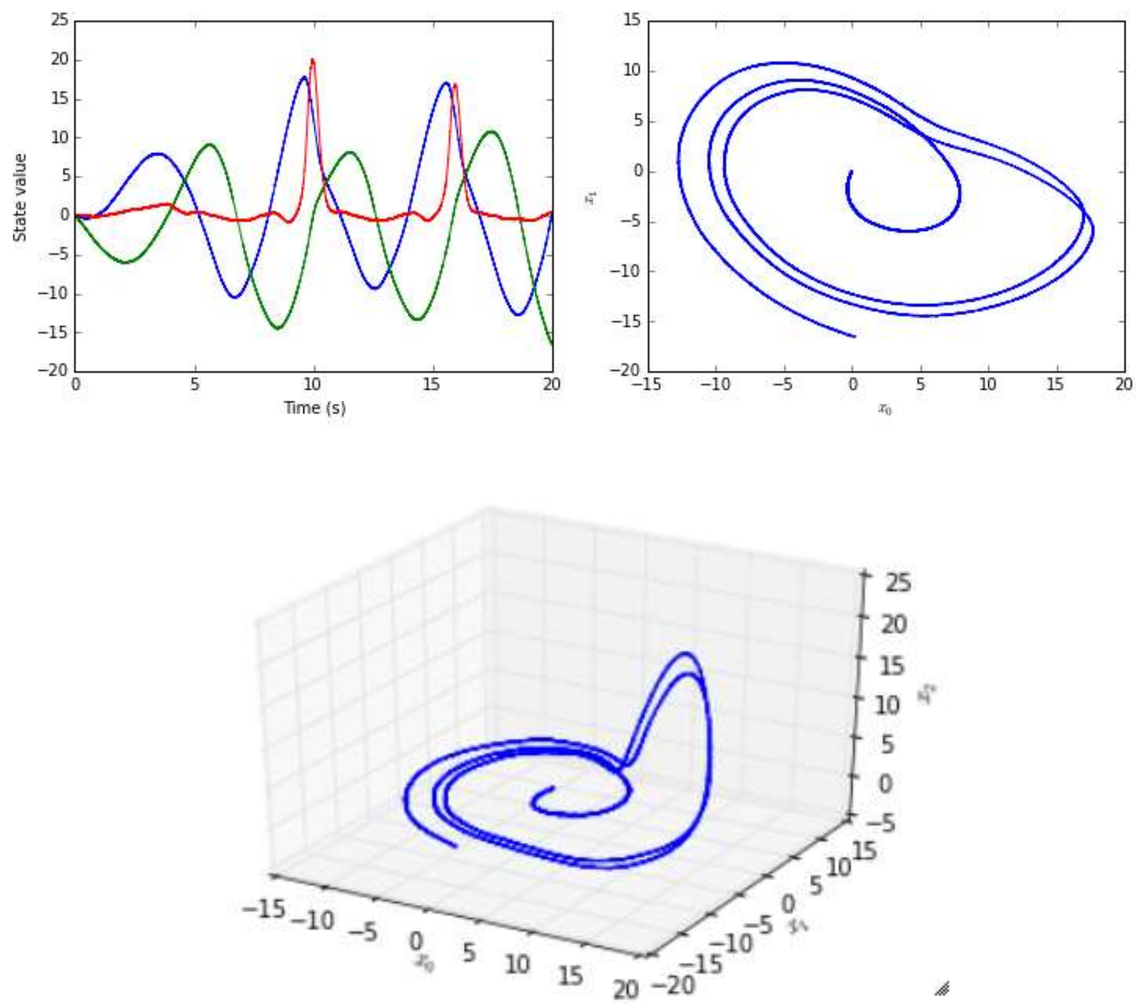

Figure 21. Rossler Attractor System : Case 1

CASE 2:

Parameters Used:
tau = 0.1
a = 0.5
b = 0.5
c = 5.7

The system acts as an attractor as expected (shown in figure 21).

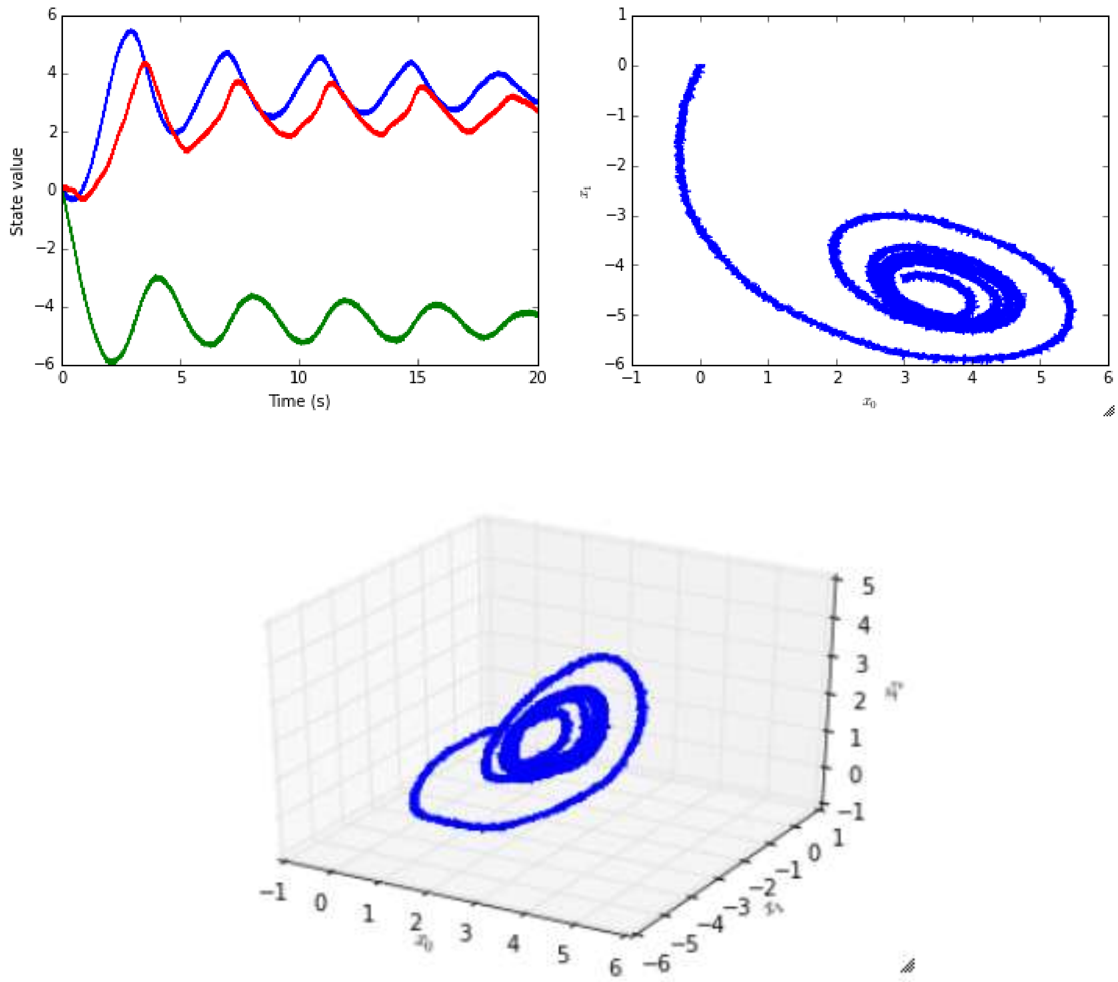

Figure 22. Rossler Attractor System: Case 2

ii) Double Integrator

I have created a double integrator by connecting two integrators together, such that the integral computed by the first integrator is in turn further integrated by the second integrator. The system was given a linear input and it was expected that the linear input would be turned into a ramp by the first integrator and further into a quadratic ouput by the second integrator.

From figure 22, it can be seen that the system works as expected (demonstrated for two different piecewise inputs). However, there is a significant lag between the two populations due to the post synaptic time constant.

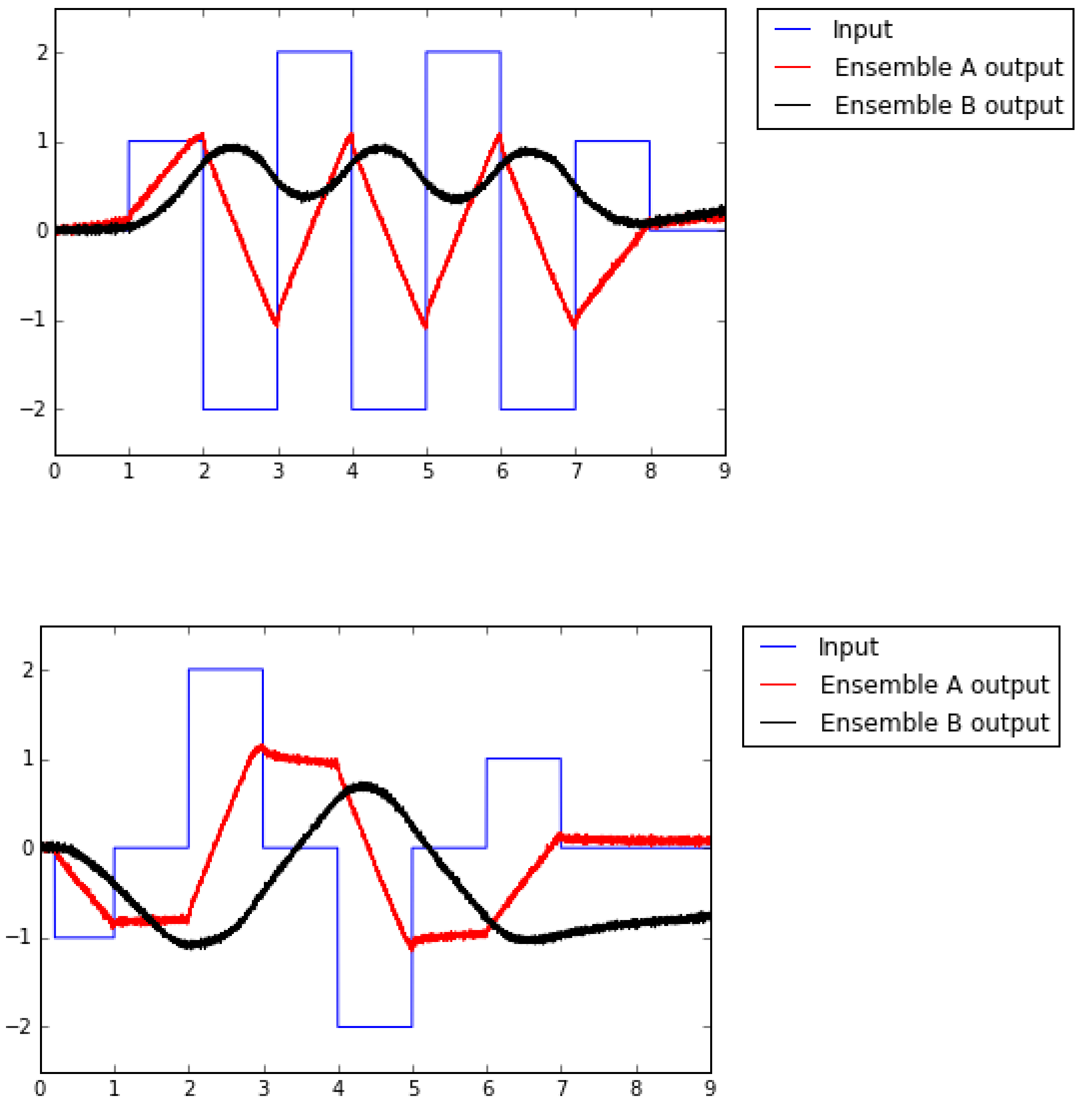

Figure 23. A Double Integrator