# Exercise 8: Temporal representation in spiking neurons

## 8.1 Using Nengo

1. Generate a new network. Generate a population (NEFEnsemble) of 2D 100 LIF neurons. Threshold range = [-1,1], maximum rate range = [200,500], $\tau_{RC} = .02$s, $\tau_{ref} = .001$s.

   (a) Hand in the population response curves, and report the MSE.

   (b) Generate 100 1D LIF neurons with the same parameters, and answer the following:

      i. What happens to MSE as the radius (and threshold range automatically) increases? Why?

      ii. What happens to the MSE and tuning curves if you set $\tau^{ref}$ to 5ms? Why?

      iii. What happens to the MSE and tuning curves as $\tau^{RC}$ changes? Why?

- Notes:
  - Right-clicking is your friend.
  - Introductory videos, tutorials, etc. can be found on nengo.ca.
  - Feel free to play with the various parameters, you should already know what they do.
  - Population response curves: right-click the population->plot->Constant Rate Responses.
  - To find the MSE, click on the NEF ensemble, open the inspector (top right magnifying glass), then click on the 'X' origin and look at the 'error'.
  - You can change the radius by editing 'radii' in the population inspector, the population will be instantly updated after saving the change.
  - For ii and iii, you need to generate new populations to determine the effects of parameter changes on MSE.
  - You can set the random seed in the last two questions so you know the population isn't changing in ways not governed by your parameter changes.
  - To set the random seed, open the script console and type 'from ca.nengo.math import PDFTools', then type 'PDFTools.setSeed(<seed>)'. You need to do this right before generating each population.
  - The console has a history and auto-complete (use arrow keys and tab). You can also create a python script and run it by opening it from the File menu.
  - There is context help for most configurable options that is accessible by right-click.

## 8.2 Dynamic transformations

1. Use Nengo to build a LIF neural integrator (see figure 8.6 on p. 232, i.e. one population). *Please be careful which time constants you use.* Inputs should be zero outside the specified range. Hand in plots of:

   (a) The linearity of the recurrent population (right-click->plot->plot distortion:X).

   (b) The response with a step input of .9 from t=[.04-1.0]. How do the spike and rate versions of this simulation compare?

   (c) The response with a step input of .9 from t=[.04-.16]. How does this compare to (b)? Why is it better or worse?

   (d) The response with a ramp input from 0 to .9 from t=[.04-.5]. How does it do, and why? What is the equation for the function traced out by the response from .04-.5s?

   (e) The response to $5\sin(5t)$. What should the output look like (write the equation)? Explain. Does spiking work in this case? Can you change any of the synaptic time constants to make the output better? Which, and how, and why does it work?

   Bonus: How do you get rid of the 'peak' at the beginning/end of step integration?

   Notes:

   - Population parameters as in the previous question, with 1-dimension.
   - To view all connection points of populations: right-click->origins and terminations->show all. The 'X' origin is the standard representation we talk about.
   - To connect populations: drag and drop a termination to add it to a population. Fill in the relevant information. Then drag and drop any origin to any termination. To change the parameters, right-click the termination or origin and choose 'inspector'.
   - To collect data you can use the interactive plots (graph button on the top right). In interactive plots, right-click on the element names to pull up various plots to resize, etc. You will want 'value' plots for these questions. You can save a layout by clicking the disk button in the bottom drop-down panel (click the triangle to drop it down).
   - Make the graphs big so you can see effects well. You can leave these graphs up as you change inputs, and the effects will show next time you reset and run.
   - Simulation parameters: Run for about 1.5s; The synaptic time constant *for the recurrent connection* must equal the 'weight' on the *input connection* (i.e., the B' matrix; i.e., $\tau B$) coming into the population (0.05s or so). The synaptic time constant *for the input connection* should be small (0.005s or so). The weight on the recurrent connection is the A' matrix.
   - Do these runs using 'rate mode' (i.e., rate neurons) unless otherwise stated. This can be changed by right-clicking on the network icon to change everything in the network, or one population at a time by right-clicking the population.

- To generate an input signal, create a new ‘Function Input’. When you first create it choose ~PiecewiseConstantFunction for b and c. Set the constructor arguments before creating the function. Here, p0 is the N times it changes, and p1 is the N+1 values it takes before/after those changes. So, for b, you have 2 columns in p0 equal to 0.04 and 1, and 3 columns in p1 equal to 0,0.9,0. I recommend using ’preview’ to see your functions and make sure they look right.
- Choose ‘User-defined Function’ for the ramp. Here you can type functions as math, where ‘x0’ is the time. So, e.g. ‘(x0-.5)*(x0>.5)’ will give a ramp from .5 to the end of the simulation. Your ramp should turn off.
- Running some of these simulations in ‘direct’ mode may help make the answers more obvious.
- To plot results, you can configure the amount of time shown, etc. in interactive plots in the bottom panel. These are the plots to hand in. You can save any screen as a .pdf using the bottom panel.