

Worksheet 5

Neural Circuits

Purpose.

To investigate how neurons can respond to input signals and generate output signals. To observe how they can add and multiply signals.

Depending on your installation you may need to load a couple of packages like this. You will find out when you run a script and it shouts at you.

```
pkg load signal
pkg load matgeom
```

1. Single Neuron with Pulse Input.

The neuron with a pulse input I and neuron state u_1 is shown in the box on the right. Note the neuron outputs its state, there is no output 'sigmoid' function. The ODE for the neuron is also shown. There is one parameter, the time constant "tau" τ .

The code to solve this ODE is in the file **rhs_SinglePulse.m**. To open this, scroll in the top left Octave window and double left-click to open. You should see the following line of code which looks like the above ODE,

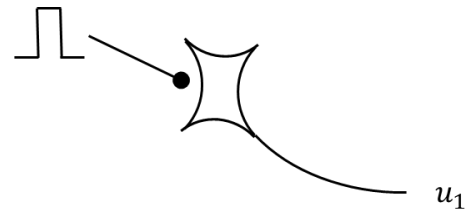
```
dudt(1) = (-u(1) + I)/tau1;
```

The equilibrium solution, obtained when $\frac{du_1}{dt} = 0$, is $u_1 = I$.

(a) To run the simulation type **SinglePulse** in Octave's command window. You will be asked for a few parameters. Here's some suggestions to get going.

Learning Outcome 1

Book Chapter 6



$$\frac{du_1}{dt} = \frac{1}{\tau}(-u_1 + I)$$

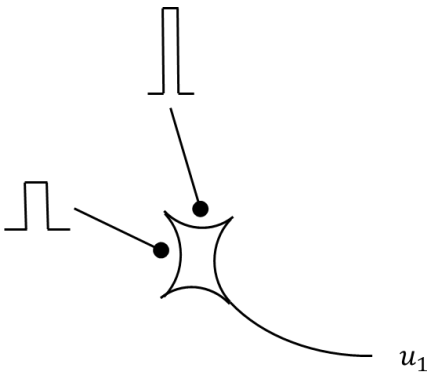
pulse start time	20
pulse end time	40
pulse height	2.5
tau	1

The computation will always end at time = 100 secs. Now let's do the following tests.

- (i) Check that the neuron's output will always rise to the pulse input value, including when this drops down to zero.
 - (ii) Check that for very short pulse lengths, the result in (i) does not hold. Why is this?
 - (iii) Investigate the neuron's behaviour for a range of values of tau. (If you want to overlay plots for comparison, then type **hold on** at the command line.
-

2. Adding two pulses

The circuit for adding two pulses is shown on the right together with the associated ODE.



$$\frac{du_1}{dt} = \frac{1}{\tau}(-u_1 + I_1 + I_2)$$

- (a) Run **TwoPulseAdd**, where the end time is 100 secs and tau is set to 0.5. Make sure the pulses overlap perfectly, i.e. they have the same start and end times. You should see the input values are added.
 - (b) Repeat for two pulses that only partially overlap. What do you observe? Can you explain this?
 - (c) Repeat for non-overlapping pulse. Can you explain what you see?
 - (d) Can you formulate a simple rule how this adding circuit should work?
 - (e) How would you modify the circuit and its ODE to *subtract* two pulses? Have a go; you will need to change the code in **rhs_TwoPulseAdd.m**.
-

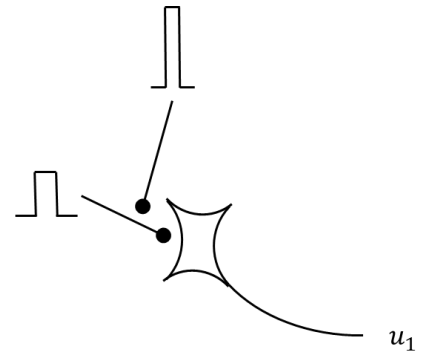
3. Multiplying Two Pulses.

Here we use the *shunting* effect of the output of one neuron on the input signal to the neuron in question. The circuit and its ODE are shown in the box on the right.

(a) Open up the script `rhs_TwoPulseMult.m` and complete the line of code

`dudt(1) =`

(b) Investigate the behaviour of this circuit, in the spirit of activity 2.



$$\frac{du_1}{dt} = \frac{1}{\tau}(-u_1 + I_1 * I_2)$$

4. Investigating the Sigmoid output function

Most neurons are equipped with an output function which limits the range of the output, typically this is in the range 0.0 – 1.0. This is useful, since we know the sizes of outputs when we construct circuits with multiple neurons. The *sigmoid* function which achieves this is as follows,

$$s(u) = \frac{1}{1 + e^{-(u-C)/D}}$$

There are two parameters; C fixes the ‘centre’ location of the curve, and D fixes how fast the curve changes.

Let’s take as ‘baseline’ parameters $C = 0.5$, $D = 0.05$ and find out what these parameters mean.

(a) Keep $C = 0.5$ and change D . Use the script **Sigmoid_Explore**. Summarise the effect of D .

(b) Now keep $D = 0.05$ and change C . You should see an interesting effect.

(c) Often, we may have a larger input range, say 0 – 500 which may come from some physical sensor. How would you choose C and D to sensibly deal with such a range?

You can experiment by changing the first line of code in the script **Sigma_Explore** to something like this.

```
x = [0:0.01:1]*500;
```

5. Advanced Reflexions on the Sigmoid.

We mentioned above that **C** fixes the centre of the sigmoid and **D** fixes the rate at which the curve rises. While it is true that D controls the rate of rise, it's not the whole truth, since this rate also depends on C.

One way of looking at this situation is to calculate the value of D so $s(u)$ rises to a certain value, let's take 0.25 which is a quarter of the maximum output. A little maths let's us find the value of D to do this. In the formula below u_1 is the input value we specify so the output is 0.25

$$D = (C - u_1) / \log 3$$

Say we have an input signal in the range 0.0 – 500.0. So we choose $C = 250$ to centre the curve. Let's require the output to be 0.5 for an input value $u_1 = 200$. Then we can calculate the value of D,

$$D = (250 - 200) / \log 3$$

which turns out to be 45.51. The box on the left shows the sigmoid for these values of C and D.

(a) Use the script **Sigma_Explore** to explore this idea.

