Anand Sekar

Professor Shea-Brown

AMATH 342

1/23/17

AMATH 342 HW 1

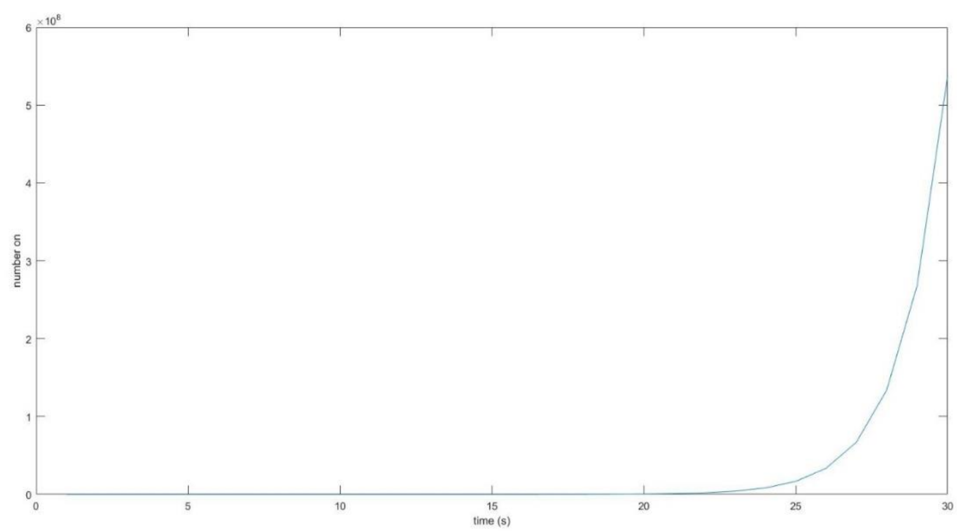
1. MATLAB TUTORIAL

EXERCISE 3.2

From the file neural_explosion.m

```
clear all;
%at time t=0 seconds, 1 neuron is "on" therefore
%the first value of number_on is 1
current_number_on = 1;
%declaring and initializing the vector
number_on = 1:1:30;
time = 1:1:30;
for s = 1:30
    number_on(s) = current_number_on;
    current_number_on = current_number_on * 2;
end

plot(time, number_on)
```

<u>Result:</u>



EXERCISE 4.1

From the file integrator_2.m

```
clear all;
thresh = 1.8;
sum = 0;
answer = 0;
for t = 1:1:10
    signal_vector = sin(t);
    sum = sum + signal_vector;
    if (sum >= thresh)
        disp('Time at which treshold is crossed:');
        disp(t);
        break
    end
end
```

<u>Result:</u>

```
Time at which treshold is crossed:

     3
```

EXERCISE 5.1

From the file rmatrix.m

```
function B = rmatrix(A, S, Z)
    B = A + S.*Z;
return;
```

Testing the code:

```
clear all;
A=ones(2,2);
S=0.5*eye(2);
Z=ones(2,2);
B=rmatrix(A,S,Z)
```

<u>Result:</u>

```
B =

    1.5000    1.0000
    1.0000    1.5000
```

2. SPIKE TRAIN ANALYSIS AND TUNING CURVES

1. CELL 1 RESPONSE AT 45 DEGREES

From the file analysis_1.m

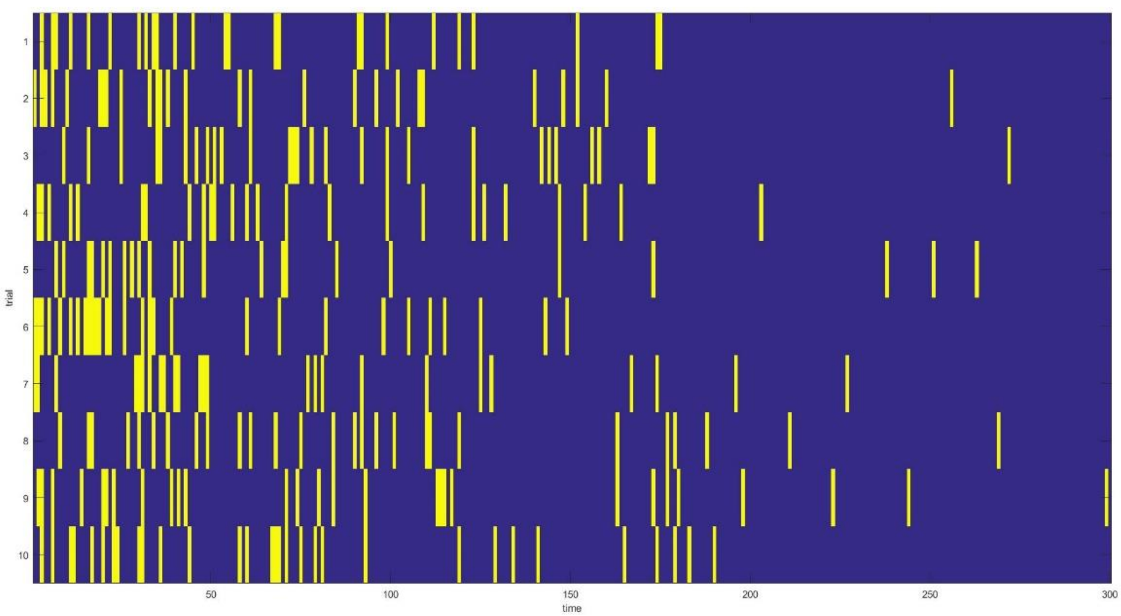
```
% Testing out using imagesc(spiketrain), figure, and
subplot commands
% Using 45 degrees, Cell 1, and 10 trials
clear all;
generate_noisy_data_cockroach;

figure;
imagesc(spiketrain)
xlabel('time')
ylabel('trial')

%Compute the average spike rate, and standard deviation
%Creating a vector for average firing rates across trials
trials = 10;
time = 300;
average_firing_rate = 1:trials;
for n = 1:trials
    current_sum = sum(spiketrain(n,:));
    current_rate = current_sum / time;
    average_firing_rate(n) = current_rate;
end
% disp('average firing rate:');
% disp(average_firing_rate);

total_average_firing_rate = mean(average_firing_rate);
disp('total average firing rate:');
disp(total_average_firing_rate);

standard_deviation = std(average_firing_rate);
disp('standard deviation:');
disp(standard_deviation);
```

Results:

```
>> analysis_1
Input the direction of your stimulus in degrees45
Which cell do you want to record from (1,2,3)1
How many repeated trials would you like to perform? 10
total average firing rate:
    0.0893


standard deviation:
    0.0078
```

For cell1 at 45° and ten trials, the average firing rates were consistently around 90 fires per second with a standard deviation of around 10 fires per second. Given this deviation, it's reasonable to assume that this data is accurate enough for our purposes.

2. EXAMINING RASTER PLOTS



The firing rates throughout each trial appear to be reducing as time goes on. This is consistent with all the cells at several different angles. This is a very simple and straightforward example of neural adaptation. Neural adaption is defined as change over time of a neuron's response to a constant stimulus. In this scenario, the neuron fires less as the constant stimulus of a single direction on the cricket's cercal ganglion is applied continuously through a short period of time – such as that of the span of the raster plot above.

3. TUNING CURVE

From the file analysis_3.m

The variable cell_num can be changed to test the three different cells. This script modifies the generate_noisy_data_cockroach.m script by adding a for loop to test every other degree from 0° to 45°.

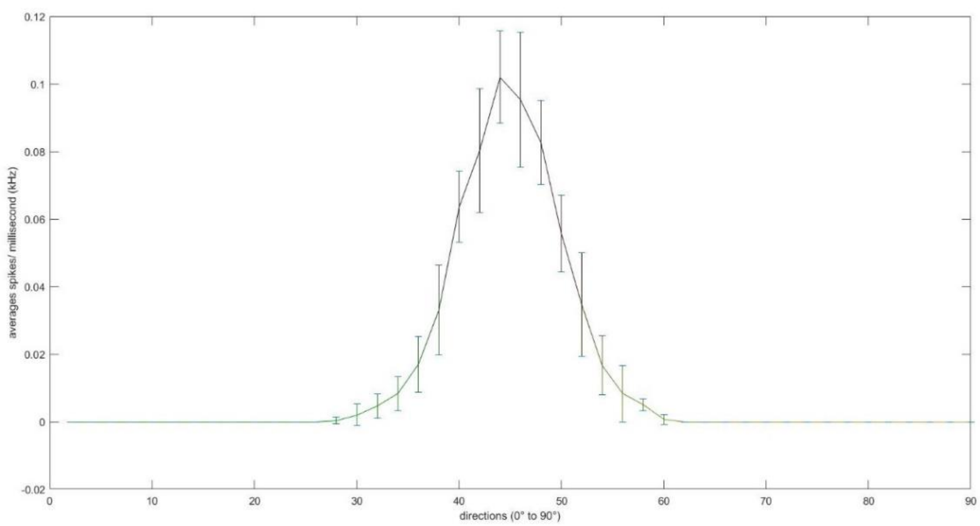
```
clear all;
%create vector for tuning curve
%testing for every 2 degrees
tuning_curve_averages = 1:1:45;
tuning_curve_direction = 1:1:45;
tuning_curve_std = 1:1:45;

for direction = 1:45

    stimDir = direction * 2;
    cell_num = 3;
    ntrials = 10;

    maxrate = 300; % 30 Hz max firing rate

    rate = maxrate*cockroach_tuning(stimDir, cell_num);
    tau = 100;       % adaptation time constant in msec
    nmsec = 300;     % number of milliseconds to record for
    times= 1:nmsec; % time units

    spiketrain = zeros(ntrials,nmsec);      % set up output data

    ratecurve = rate*exp(-times/tau)*.001;  % adapting rate function

    for j = 1:ntrials;

        for i = 1:nmsec;

            if(rand(1)<ratecurve(i)),
                spiketrain(j,i) = 1;
            end;

        end;

    end;
```

(code continued on next page)

```
%     at this point, you now have a spiketrain to analyze
%     get average firing rate and standard deviation
    average_firing_rate = 1:ntrials;
    for n = 1:ntrials
        current_sum = sum(spiketrain(n,:));
        current_rate = current_sum / nmsec;
        average_firing_rate(n) = current_rate;
    end

    total_average_firing_rate = mean(average_firing_rate);

    standard_deviation = std(average_firing_rate);

%     add averages and standard deviation to vector
    tuning_curve_averages(direction) = total_average_firing_rate;
    tuning_curve_std(direction) = standard_deviation;
end;

degrees_label = tuning_curve_direction * 2;
figure;
errorbar(degrees_label, tuning_curve_averages, tuning_curve_std);
xlabel('directions (0° to 90°)')
ylabel('averages spikes/ millisecond (kHz)')
```

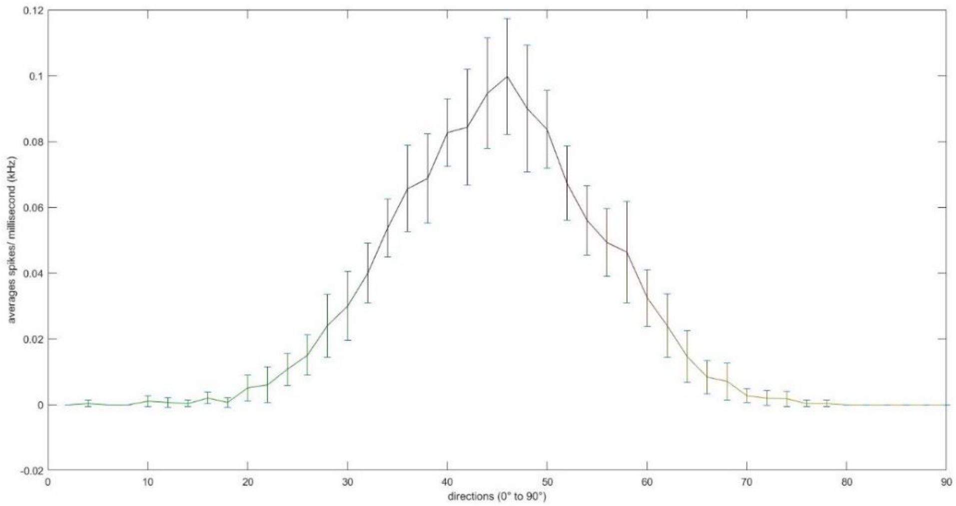
Results:

Cell 1:



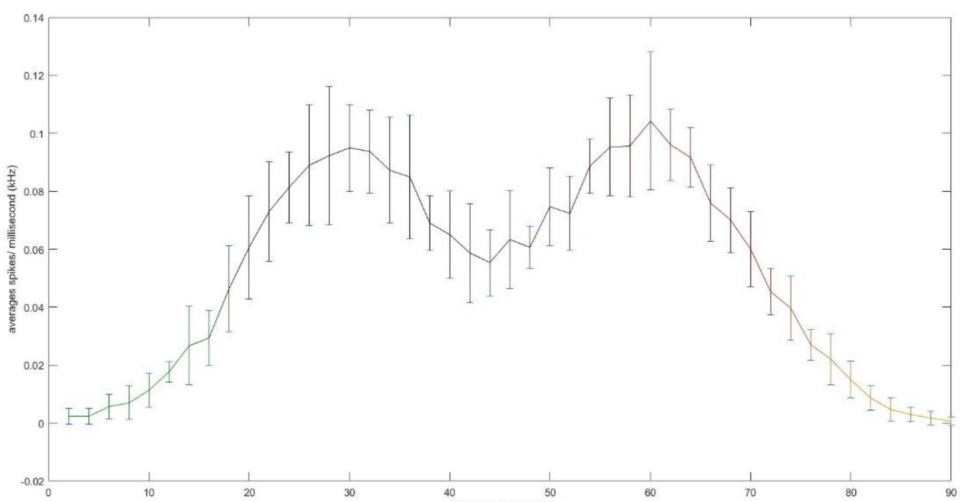
The tuning curve's average – at the highest point - tends to rest at approximately 45°. It seems symmetrical. There is a relatively steep decline in the curve on both sides of this point – indicating that this neuron fires more specifically for 45° with little range – which seems to be from just under 30° to just over 60°.

Cell 2:



This cell’s tuning curve’s average is also at approximately 45°. The difference between Cell 2 and Cell 1 is the steepness of the slopes on both sides of the average – they are both symmetrical, yet Cell 2 has a lesser slope. This neuron is not as “specific,” with firing when the hair is moved 45°, it has a range from 10° to 80°, with a seemingly constant slope from 20° to 70° at which it fires constantly more approaching the center at 45°. Both cells have the same “preferred orientation” (45°), however this description does not suffice – since we have identified the difference in the range around this preferred orientation.

Cell 3:



This cell seems to have two “preferred orientations,” at 30 and 60 degrees. Again, this graph demonstrates how “preferred orientation,” is insufficient, showing how there can be multiple preferences for a single neuron.

Each cell seems to exhibit these properties:

- A single or perhaps more “preferred orientations” at specific directions
- A somewhat steady/ consistent decline in firing rates at specific ranges around specific preferred directions