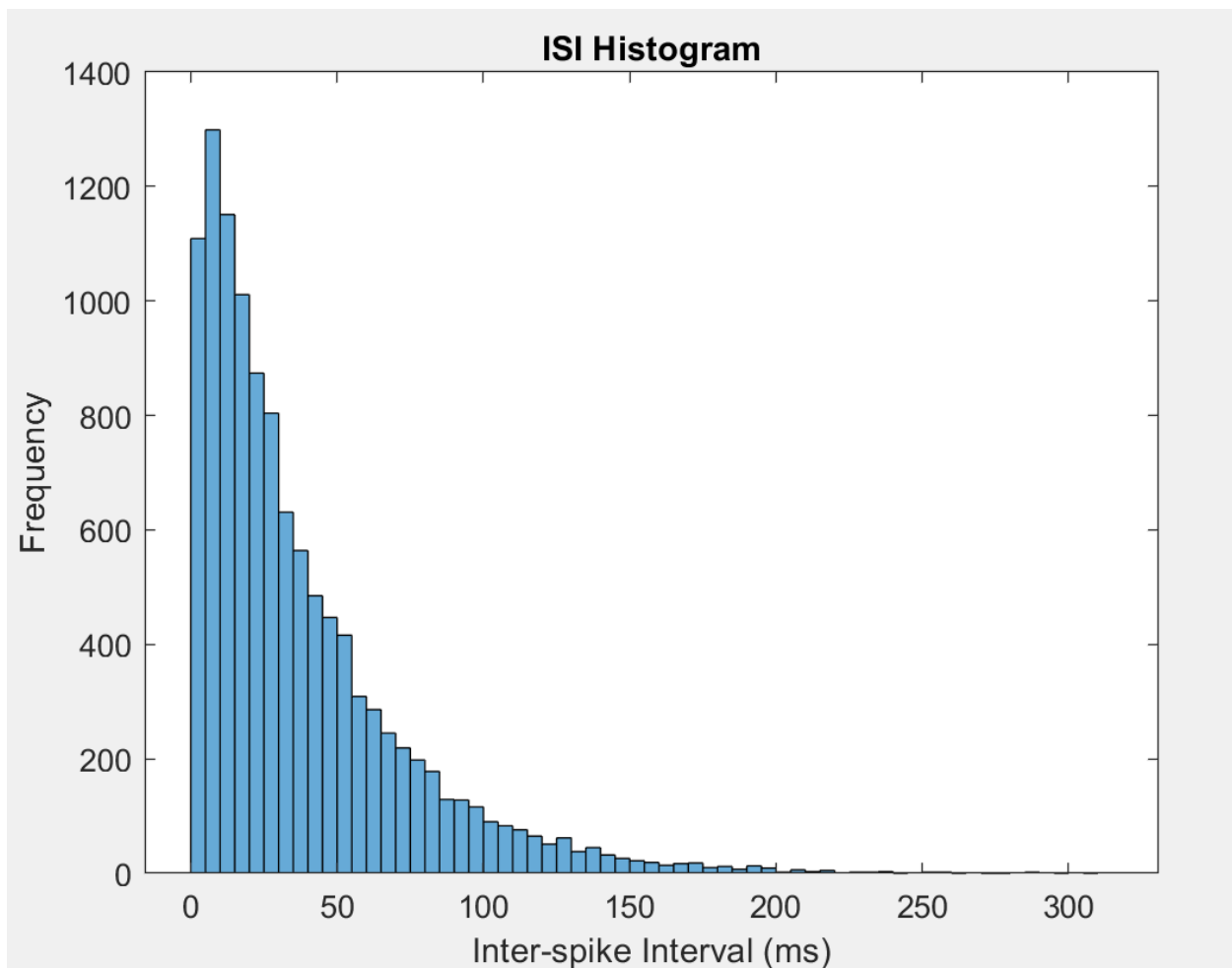


Jason Wong 1090292616 15/11/2017

%Note, if you would like to run the plots, please comment out the close(x) lines

%Exercise 2.1.1



```
clear all
close all
load Spike_data_1.mat
spikeData1=d;
load Spike_data_2.mat
spikeData2=d;
load Spike_data_3.mat
spikeData3=d;

ISI = [];
n_trials = 1000;
T = 500;

for k=1:n_trials
    spike_times = find(spikeData1(k,:) == 1);
    isi0 = diff(spike_times);
    ISI = [ISI,isi0];
end

figure(1)
histogram(ISI);
title('ISI Histogram')
```

```
ylabel('Frequency')
xlabel('Inter-spike Interval (ms)')
close(1)
```

%Exercise 2.1.2

```
%The neuron fires earlier rather than later.
%Because the curve fits inside a negative exponential curve, therefore the
%neuron is Poissonian and follows Poisson distribution. This is because
%neurons fire randomly in a fixed average rate.
```

%Exercise 2.1.3

```
%meanTime=36.9525ms, that means it takes on average 36.9525 milliseconds for
%the neuron to fire again
```

```
meanTime = mean(ISI);
```

%Exercise 2.1.4

```
%modeTime=2ms, that means the most common time it took for the neuron to
%fire again was 2ms
modeTime = mode(ISI);
```

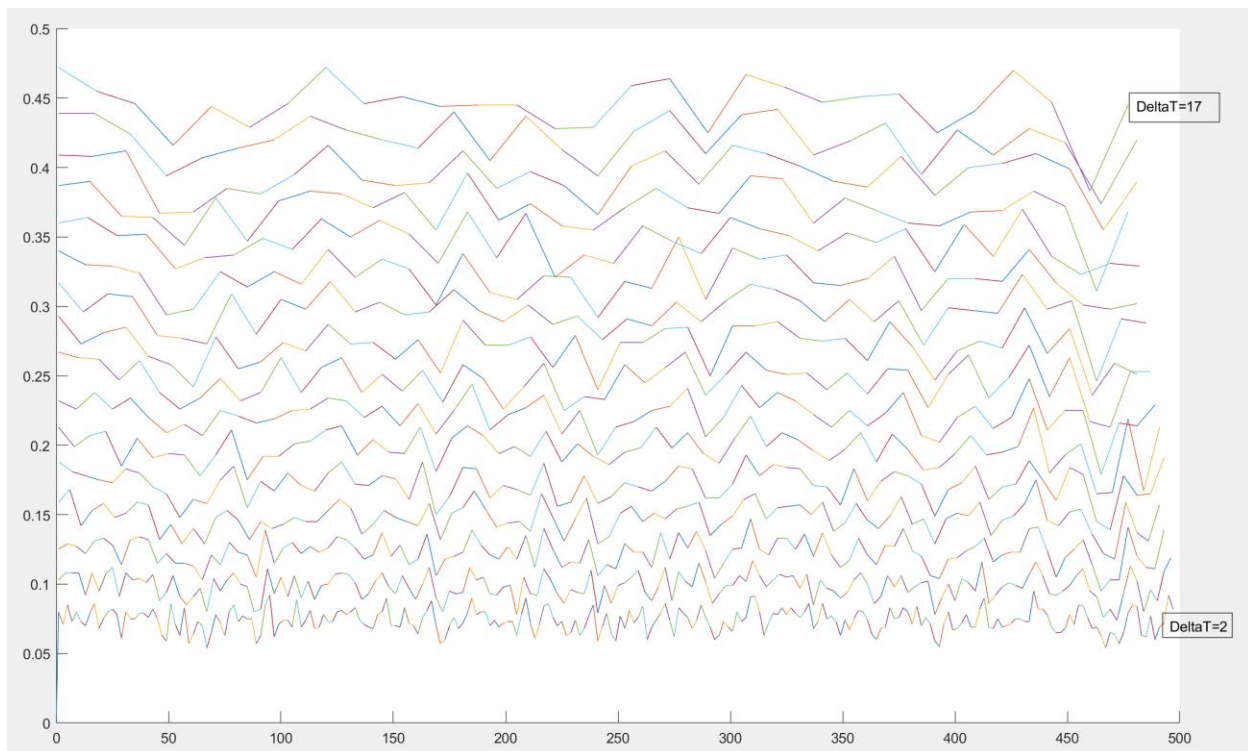
%Exercise 2.1.5

```
%meanTime(36.9525ms) is not equal to modeTime(2ms), because the average is
%not the same as the mode
```

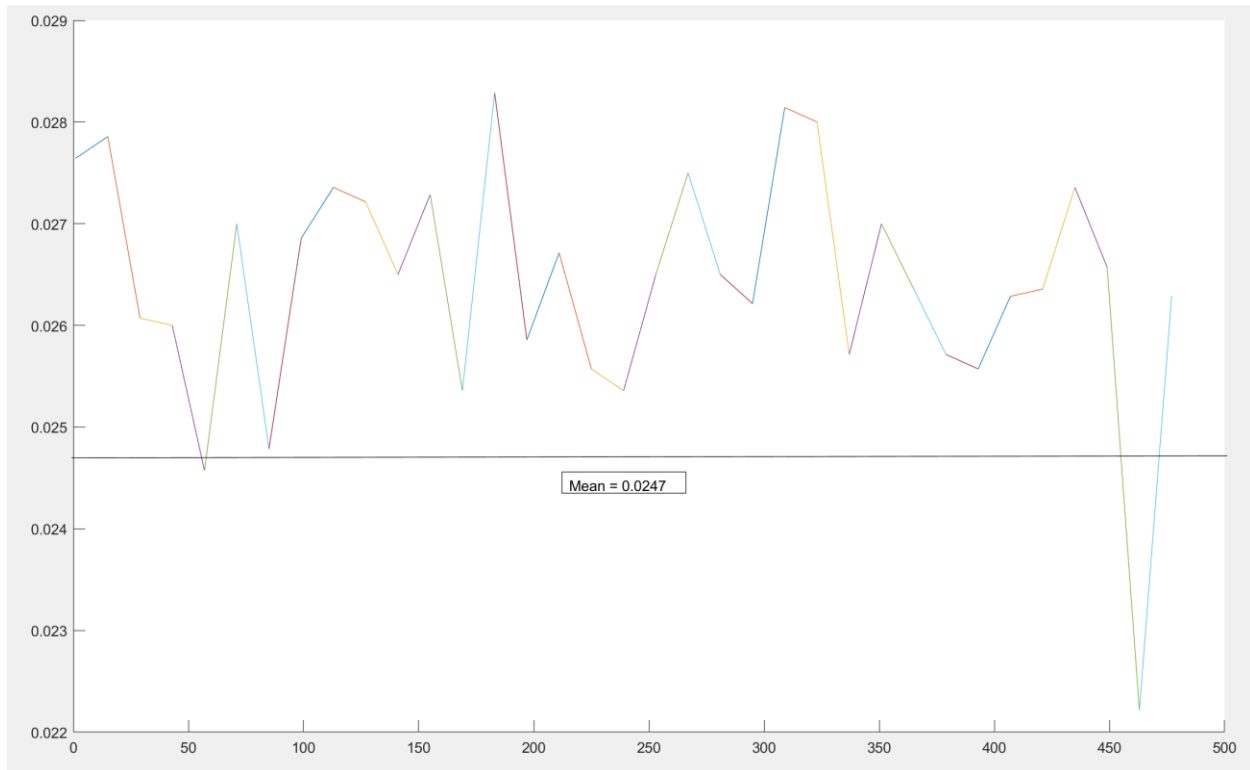
%Exercise 2.1.6

```
%I binned each set of spikes into bins seperated by deltaT, and by changing
%deltaT from 2 to 20ms, I could remove noise until a clear spike train
%emerged.
```

```
%Plotting deltaT from 2ms to 20ms
```



```
%Plotting deltaT=14 as it's the most clearest, note the ~12 spikes
```



```
%Given the ISI histogram, I can guess the shape of the spike train, this is
%because I can guess the average ISI using the histogram and then I will
%know the average intervals between each spike in the spike train.
figure(2)
prob_spike = sum(spikeData1,1)/n_trials;
prevDot = 0;
prevT = 0;
hold on
for deltaT=2:1:17
    for t=1:deltaT:500-deltaT
        sumSpikes=0;
        for i=t:1:t+deltaT
            sumSpikes=sumSpikes+prob_spike(i);
        end
        if t>prevT
            plot([prevT t], [prevDot sumSpikes]);
        end
        prevT = t;
        prevDot = sumSpikes;
    end
end
hold off
close(2);

figure(11)
prob_spike = sum(spikeData1,1)/n_trials;
hold on
for deltaT=14:1:14 %deltaT = 14 has the clearest spike train with n cloests to 12.344
    for t=1:deltaT:500-deltaT
        sumSpikes=0;
        for i=t:1:t+deltaT
            sumSpikes=sumSpikes+prob_spike(i);
        end
        sumSpikes=sumSpikes/deltaT
    end
end
```

```

        if t>prevT
            plot([prevT t], [prevDot sumSpikes]);
        end
        prevT = t;
        prevDot = sumSpikes;
    end
end
hold off
close(11);

```

%Exercise 2.2.1

```

%This sums up every spike, and divides it by n_trials for the average spike
%per trial
avgSpikePerTrial = 12.0750
avgSpikePerTrial = sum(sum(spikeData2))/n_trials;

```

%Exercise 2.2.2

```

%prob_Spike is an array of all the probability for a neuron to spike
%mean(prob_spike) gives the average firing rate probability
avgProb_Spike = 0.0241
prob_spike = sum(spikeData2)/n_trials;
avgProb_spike = mean(prob_spike);

```

%Exercise 2.2.3

```

%Average ISI is found by using the mean function on the ISI.
%meanTime = 35.6245

```

```

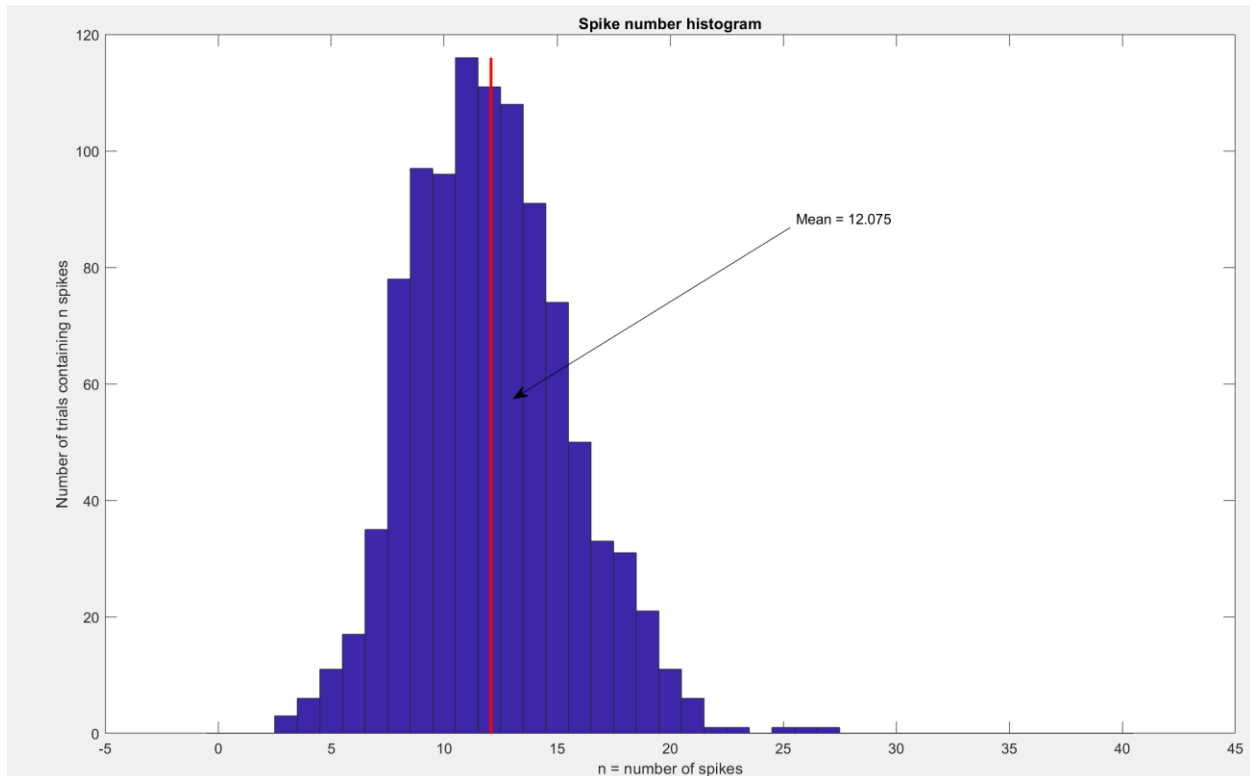
ISI2 = [];
n_trials = 1000;
T = 500;

for k=1:n_trials
    spike_times = find(spikeData2(k,:) == 1);
    isi0 = diff(spike_times);
    ISI2 = [ISI2, isi0];
end

meanTime = mean(ISI2);

```

%Exercise 2.2.4



```
%Code adapted from Zenas Chao's Week 5 class. I made it as fancy as
%possible
%n_avg = 12.075 spikes

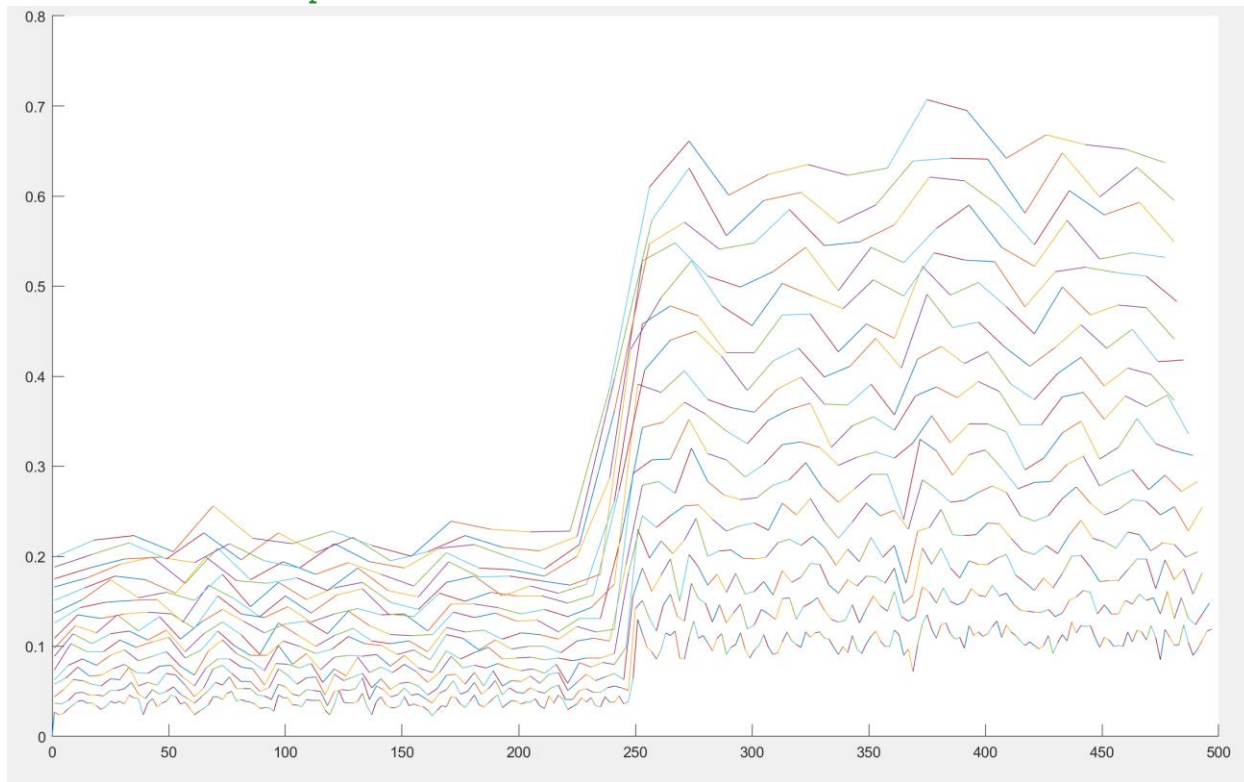
n_spikes_per_trial = sum(spikeData2,2);
n_avg = mean(n_spikes_per_trial);

figure(3)
hist(n_spikes_per_trial, [0:1:40])
xlabel('n = number of spikes');
ylabel('Number of trials containing n spikes');
title('Spike number histogram')
hold on
plot([n_avg, n_avg], [0, max(hist(n_spikes_per_trial, [0:1:40]))], 'r', 'LineWidth',
2)
x = [0.6 0.43];
y = [0.7 0.5];
annotation('textarrow',x,y,'String','Mean = 12.075') %Labels the mean
hold off
close(3)

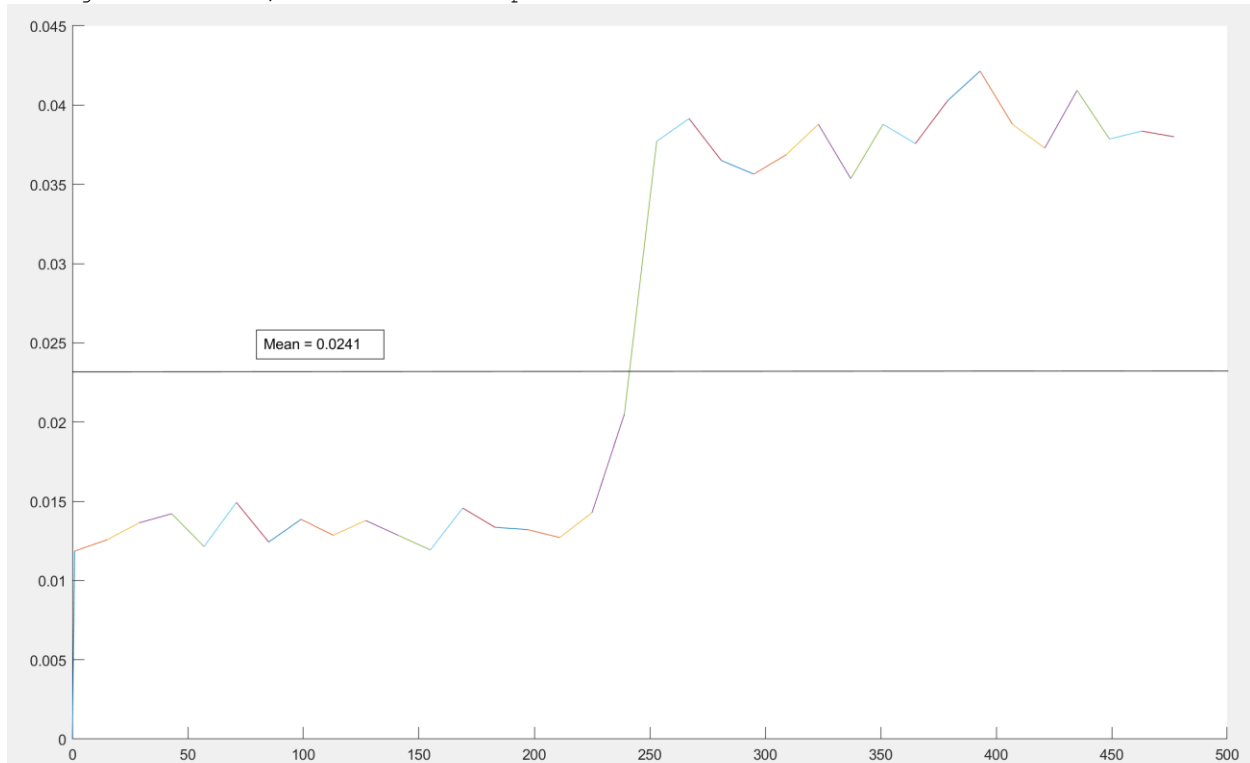
%Exercise 2.2.5
figure(4)
hold on
histogram(ISI2);
title('ISI Histogram')
xlabel('Inter-spike Interval (ms)');
ylabel('Frequency');
line([meanTime, meanTime], ylim, 'LineWidth', 1, 'Color','k'); %Plots the mean of the
histogram
x = [0.4 0.25];
y = [0.7 0.5];
annotation('textarrow',x,y,'String','Mean = 35.6245') %Labels the mean
hold off
close(4)
```

%Exercise 2.2.6

%Look at 2.1.6 for explanation of this code



%Using deltaT=14ms, notice the 12 spikes.



figure(5)
prob_spike = sum(spikeData2,1)/n_trials;

```

prevDot = 0;
prevT = 0;
hold on
for deltaT=2:1:17
    for t=1:deltaT:500-deltaT
        sumSpikes=0;
        for i=t:1:t+deltaT
            sumSpikes=sumSpikes+prob_spike(i);
        end
        if t>prevT
            plot([prevT t], [prevDot sumSpikes]);
        end
        prevT = t;
        prevDot = sumSpikes;
    end
end
hold off
close(5);

figure(10)
prob_spike = sum(spikeData2,1)/n_trials;
prevDot = 0;
prevT = 0;
hold on
for deltaT=14:1:14 %DeltaT = 14ms for the most clearest spike train
    for t=1:deltaT:500-deltaT
        sumSpikes=0;
        for i=t:1:t+deltaT
            sumSpikes=sumSpikes+prob_spike(i);
        end
        sumSpikes=sumSpikes/deltaT
        if t>prevT
            plot([prevT t], [prevDot sumSpikes]);
        end
        prevT = t;
        prevDot = sumSpikes;
    end
end
hold off
close(10);

```

%Exercise 2.2.7

%Average ISI: Data1 = 36.9525ms, Data2 = 35.6245ms (Exercise 2.1.3 and
 %Exercise 2.2.3 results)

%Data1 > Data 2

%Most common ISI: Data1 = 2ms 0.0251 occurrence rate, Data2 = 1ms 0.0315

%occurrence

%Occurrence rate found using numel(find(ISI == 2))/numel(ISI)

%Data 1 mode > Data 2 mode. Data 2 occurrence rate > Data 1 occurrence

%Average probability of spiking: Data1 = 0.0247, Data2 = 0.0241

%This is very similar so it can't be used to reliably differentiate between Data 1 and
 2

%Looking at the spike train of Data 2, the spike probability almost tripled
 %halfway through the trial. This indicates that the subject
 %underwent a stimulus event at the halfway point.

%For the spike train of Data 1, there wasn't such stimulus event, and the
 %spike train had a single average.

%Because Data 1 doesn't have a jump in spike probability which is present
 %in Data 2, my conclusion is that they display different activity.

%Exercise 2.3.8

```
%Activity in data2 has a massive jump halfway into the trial, most likely
%because of stimulation from an external source. Other than that there
%weren't a lot of unique identifiers.
```

%Exercise 2.3.1

```
%This sums up every spike, and divides it by n_trials for the average spike
%per trial
```

```
%avgSpikePerTrial = 13.4820
avgSpikePerTrial = sum(sum(spikeData3))/n_trials;
```

%Exercise 2.3.2

```
%prob_spike is an array of all the probability for a neuron to spike
%mean(prob_spike) gives the average firing rate probability
```

```
%avgProb_spike = 0.0270
prob_spike = sum(spikeData3)/n_trials;
avgProb_spike = mean(prob_spike);
```

%Exercise 2.3.3

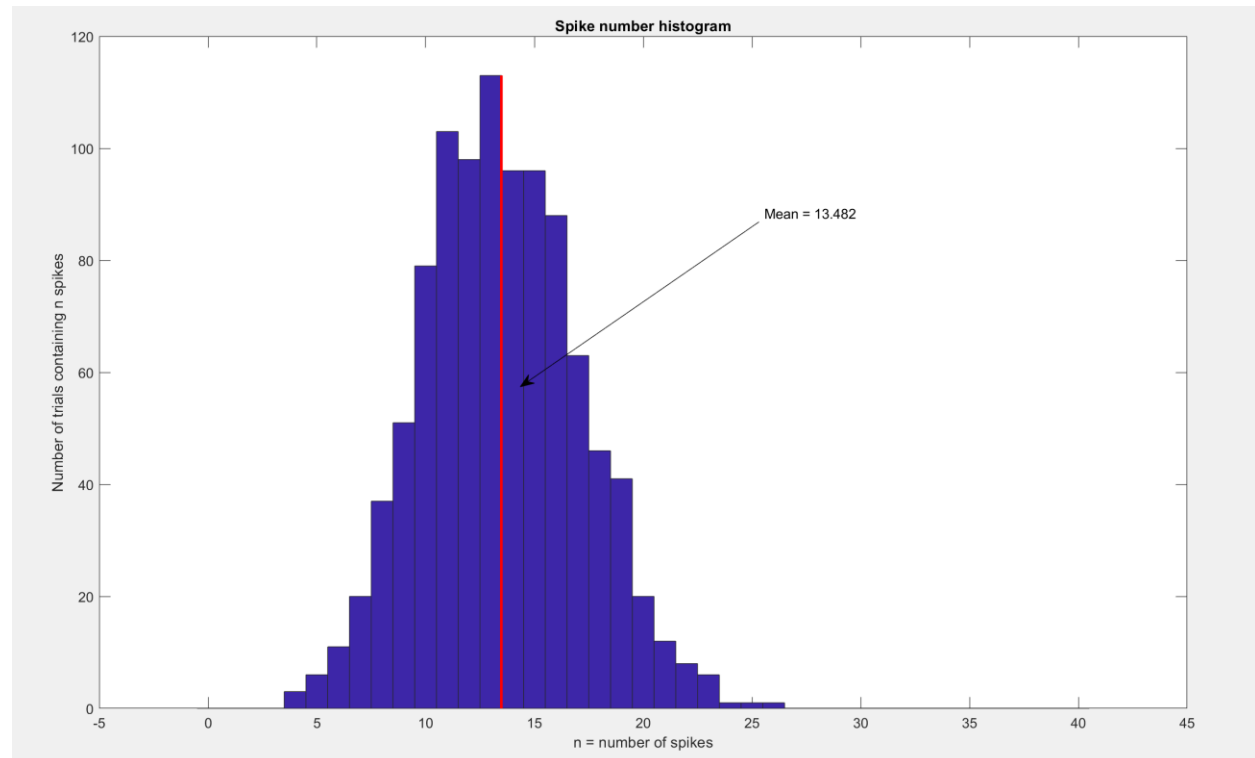
```
%Average ISI is found by using the mean function on the ISI.
%meanTime = 32.4087
```

```
ISI3 = [];
n_trials = 1000;
T = 500;

for k=1:n_trials
    spike_times = find(spikeData3(k,:) == 1);
    isi0 = diff(spike_times);
    ISI3 = [ISI3,isi0];
end
```

```
meanTime = mean(ISI3);
```

%Exercise 2.3.4



```
%Code adapted from Zenas Chao's Week 5 class. I made it as fancy as
%possible
```

```
n_spikes_per_trial = sum(spikeData3,2);
```

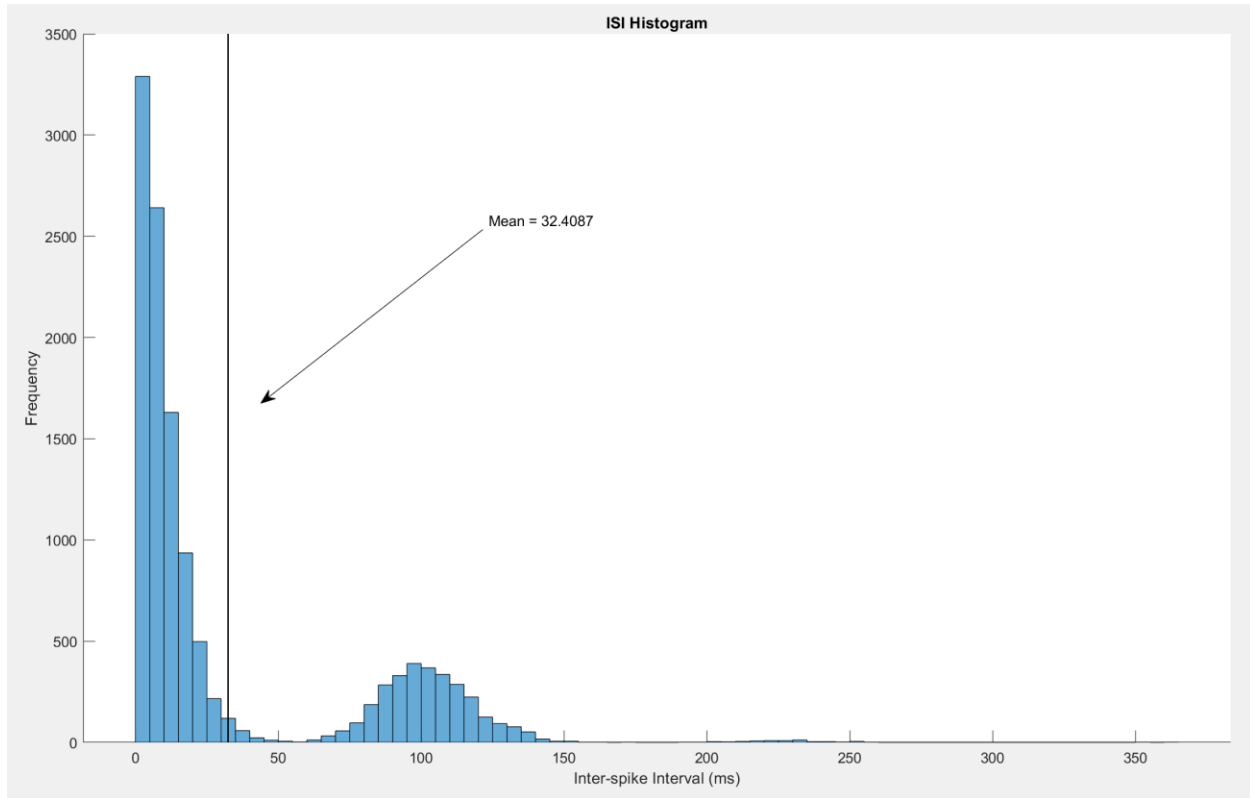


```

n_avg = mean(n_spikes_per_trial);
figure(6)
hist(n_spikes_per_trial, [0:1:40])
xlabel('n = number of spikes');
ylabel('Number of trials containing n spikes');
title('Spike number histogram')
hold on
plot([n_avg, n_avg], [0, max(hist(n_spikes_per_trial, [0:1:40]))], 'r', 'LineWidth',
2)
x = [0.6 0.43];
y = [0.7 0.5];
annotation('textarrow',x,y,'String','Mean = 13.482') %Labels the mean
hold off
close(6)

```

%Exercise 2.3.5

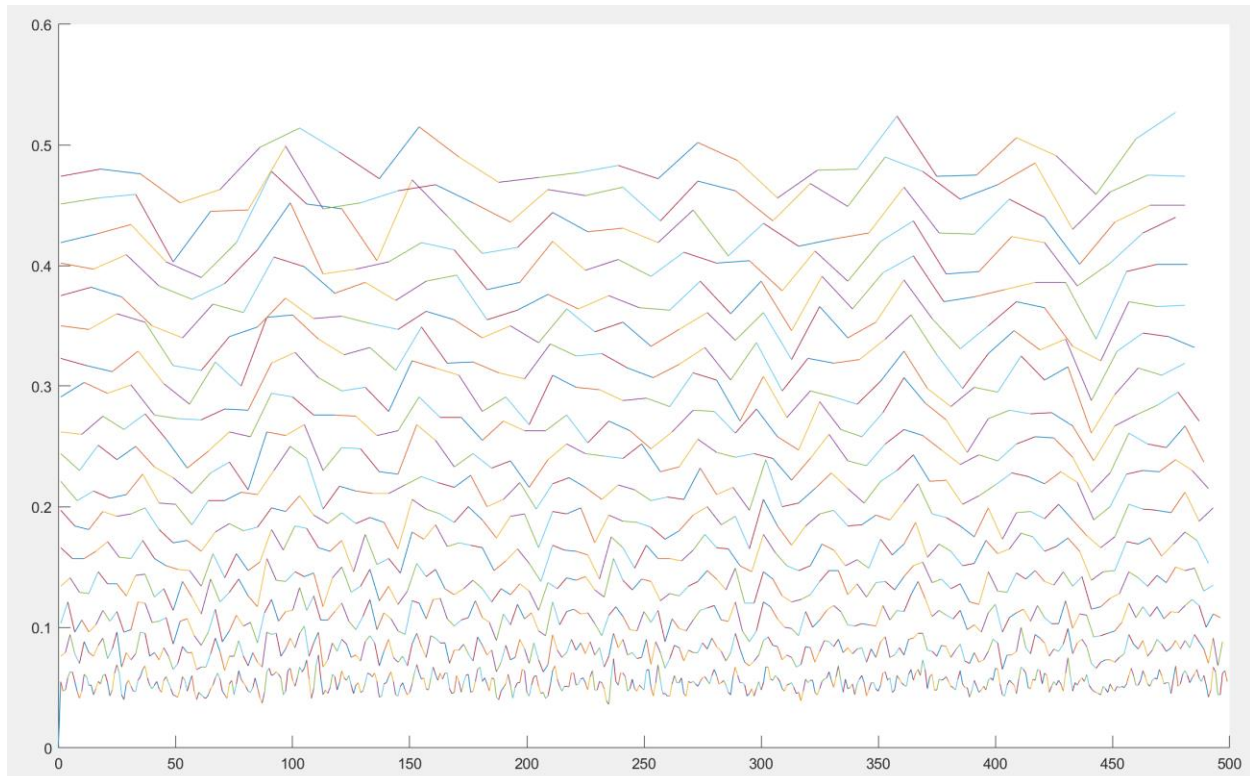


```

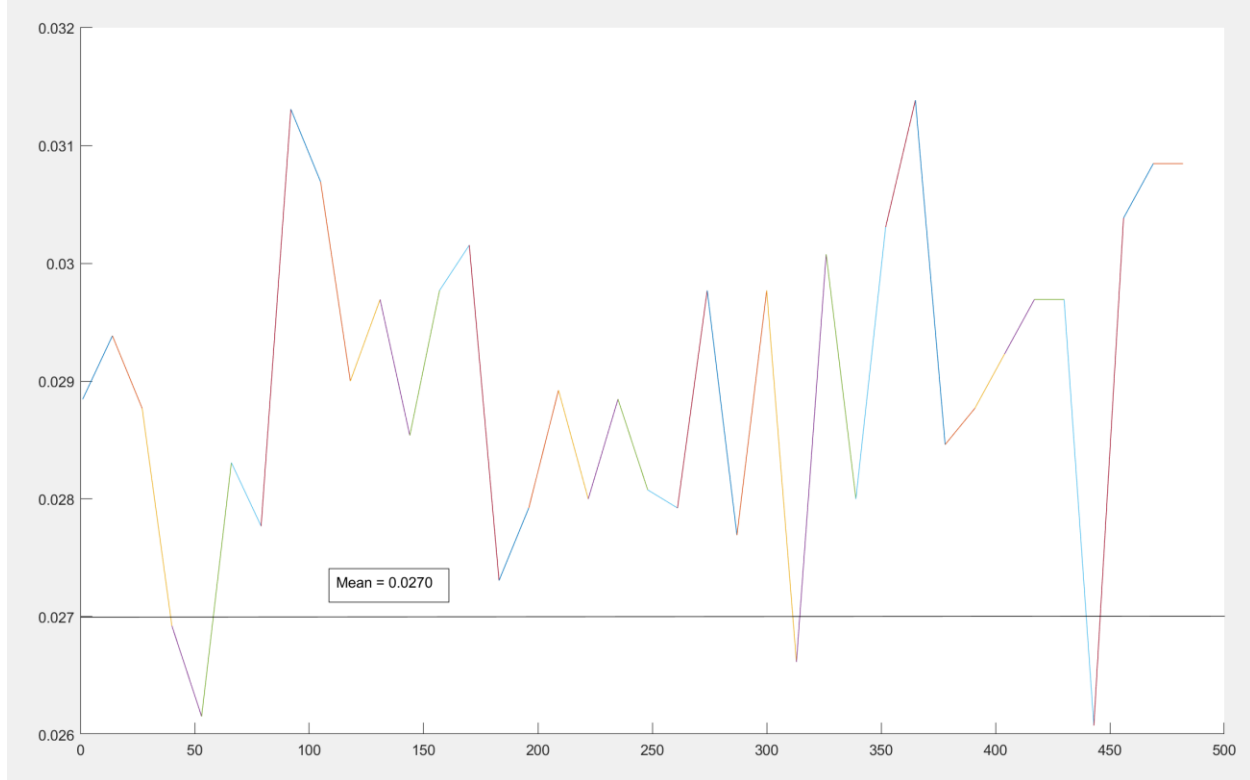
%Mean ISI = 32.4087
%There are two peaks, which allows us to separate data 3 from 1 and 2
figure(7)
hold on
histogram(ISI3);
title('ISI Histogram')
xlabel('Inter-spike Interval (ms)');
ylabel('Frequency');
line([meanTime, meanTime], ylim, 'LineWidth', 1, 'Color','k'); %Plots the mean of the
histogram
x = [0.4 0.25];
y = [0.7 0.5];
annotation('textarrow',x,y,'String','Mean = 32.4087') %Labels the mean
hold off
close(7)

```

%Exercise 2.3.6



```
%I binned each set of spikes into bins seperated by deltaT, and by changing
%deltaT from 2 to 20ms, I could remove noise until a clear spike train
%emerged.
%The deltaT is 13ms for the clearest line for data 3, notice the 13 visible spikes
```



figure(8)

```

prob_spike = sum(spikeData3,1)/n_trials;
prevDot = 0;
prevT = 0;
hold on
for deltaT=1:1:17 %Plotting for deltaT = 1 to 17
    for t=1:deltaT:500-deltaT
        sumSpikes=0;
        for i=t:1:t+deltaT
            sumSpikes=sumSpikes+prob_spike(i);
        end
        if t>prevT
            plot([prevT t], [prevDot sumSpikes]);
        end
        prevT = t;
        prevDot = sumSpikes;
    end
end
hold off
close(8)

figure(9)
hold on
for deltaT=[13] %Plotting for deltaT = 13 has the most clear spike train
    for t=1:deltaT:500-deltaT
        sumSpikes=0;
        for i=t:1:t+deltaT
            sumSpikes=sumSpikes+prob_spike(i);
        end
        sumSpikes=sumSpikes/deltaT
        if t>prevT
            plot([prevT t], [prevDot sumSpikes]);
        end
        prevT = t;
        prevDot = sumSpikes;
    end
end
hold off
close(9);

```

%Exercise 2.3.7

%The ISI histogram for data 3 is very different to those of 1 and 2, as it
 %has another peak at about 100ms whereas 1 and 2 doesn't have that peak.

%Looking at the spike train of Data 3, it is very similar to Data 1, but
 %the important difference is that the amplitude of the spikes in Data 3 are
 %much greater.

%Average ISI: Data1 = 36.9525ms, Data2 = 35.6245ms Data3 = 32.4087ms
 %Data1 > Data2 > Data3 in speed

%Most common ISI: Data1 = 2ms 0.0251 occurrence rate, Data2 = 1ms 0.0315
 %occurrence Data3 = 1ms 0.0731 occurrence rate
 %Occurance rate found using numel(find(ISI == 2))/numel(ISI)
 %Data 1 mode > Data 2 mode

%Average probability of spiking: Data1 = 0.0247, Data2 = 0.0241 Data3 =
 %0.0270
 %Data3 has a higher one than data1 and 2, so this can be used to
 %differentiate it out of the other two

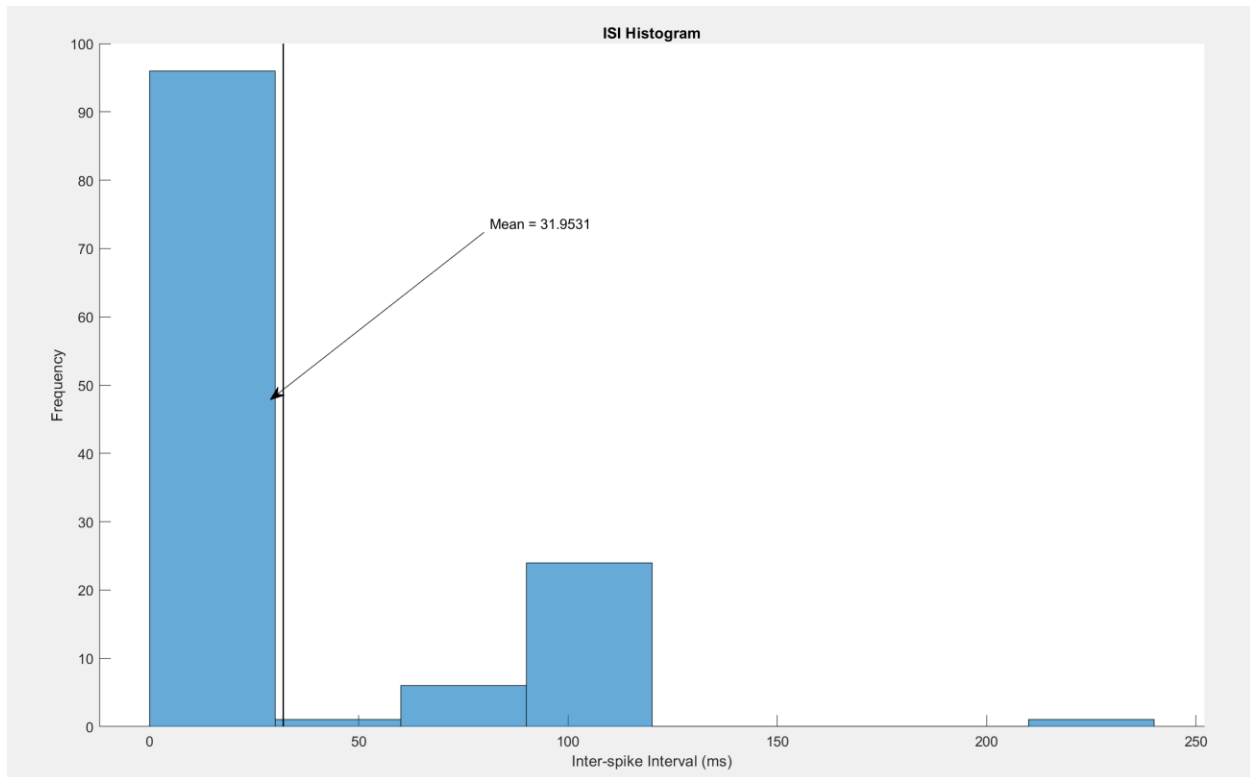
%Exercise 2.3.8

%Data 3 has a similar spike train to Data 1, the identifying feature for
 %this data set is that for the ISI histogram, there is a second peak at
 %100ms. The second peak may be because the neuron only fires at regular intervals.

%Exercise 2.4.1

```
%I will look at the ISI histogram and the spike train data, if the spike
%probability suddenly increases halfway through the trials, then it must be
%C2. If the ISI histogram has two peaks, then it must be C3. If it doesn't
%have two then it must be C1
load('Spike_data_test.mat')
spikeDataTest = d_test;
n_trials = 10;

%The ISI Histogram shows two peaks, so it must be C3
```

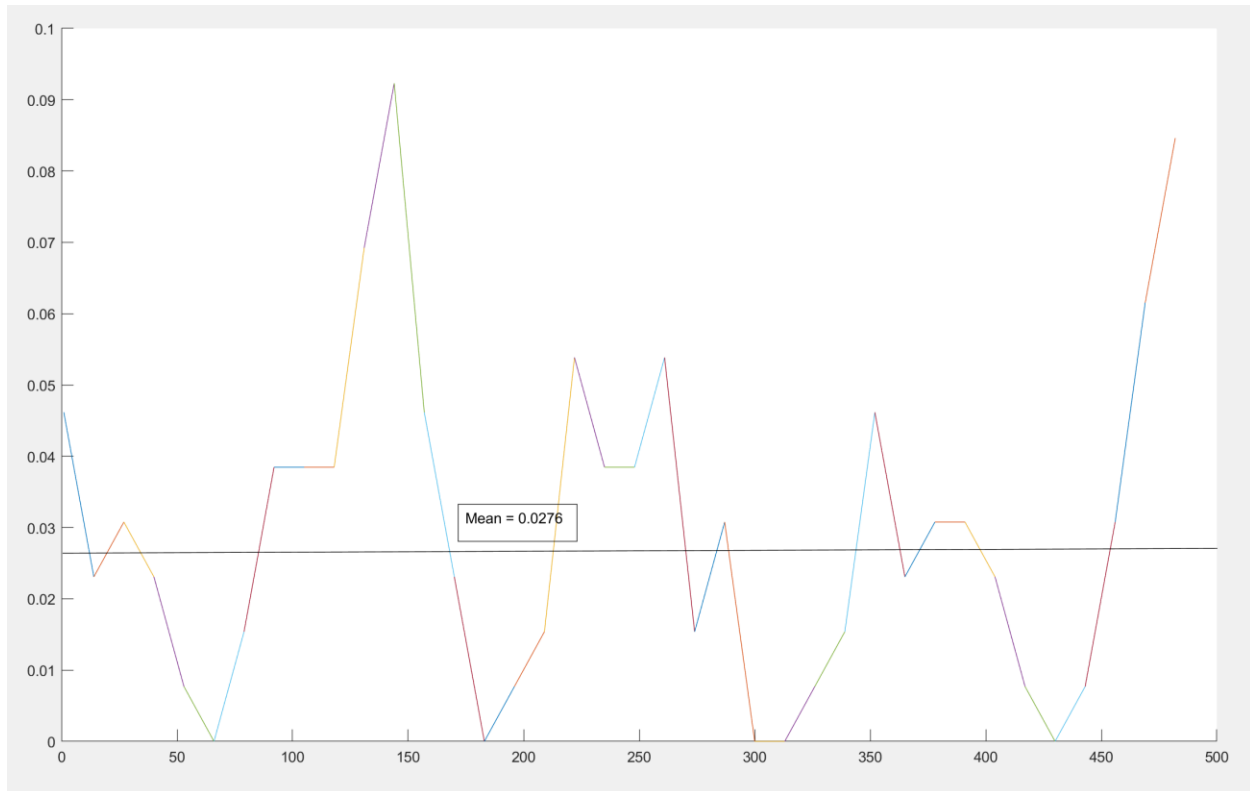


```
ISITest = [];
T = 500;

for k=1:n_trials
    spike_times = find(spikeDataTest(k,:) == 1);
    isi0 = diff(spike_times);
    ISITest = [ISITest,isi0];
end

figure(13)
hold on
histogram(ISITest);
title('ISI Histogram')
xlabel('Inter-spike Interval (ms)');
ylabel('Frequency');
line([meanTime, meanTime], ylim, 'LineWidth', 1, 'Color','k'); %Plots the mean of the
histogram
x = [0.4 0.25];
y = [0.7 0.5];
annotation('textarrow',x,y,'String','Mean = 32.4087') %Labels the mean
hold off
%close(13)

%Checking the spike train data, there doesn't seem to be an increase of
%spike probability halfway through the trial so it definitely isn't C2, also the mean
is very similar to C3 (0.0270) so it must be C3.
```



```
figure(15)
hold on
prob_spike = sum(spikeDataTest,1)/10;

for deltaT=[13] %Plotting for deltaT = 13ms has the most clear spike train
    for t=1:deltaT:500-deltaT
        sumSpikes=0;
        for i=t:1:t+deltaT
            sumSpikes=sumSpikes+prob_spike(i);
        end
        sumSpikes=sumSpikes/deltaT;
        if t>prevT
            plot([prevT t], [prevDot sumSpikes]);
        end
        prevT = t;
        prevDot = sumSpikes;
    end
end

hold off
%close(15);
```