Table of Contents

EE239AS HW #6

```
clc
clear
close all
% Collaborators: Vikranth, Yusi
```

Problem 1

```
load('/Users/Yusi/Documents/EE239AS/HW6/JR_2015-12-04_truncated2.mat');
```

Part A: Number of Trials

```
n_trials = length(R);
fprintf('Number of Trials: %d \n', n_trials)

% The number of trials performed by Monkey J is 506.
```

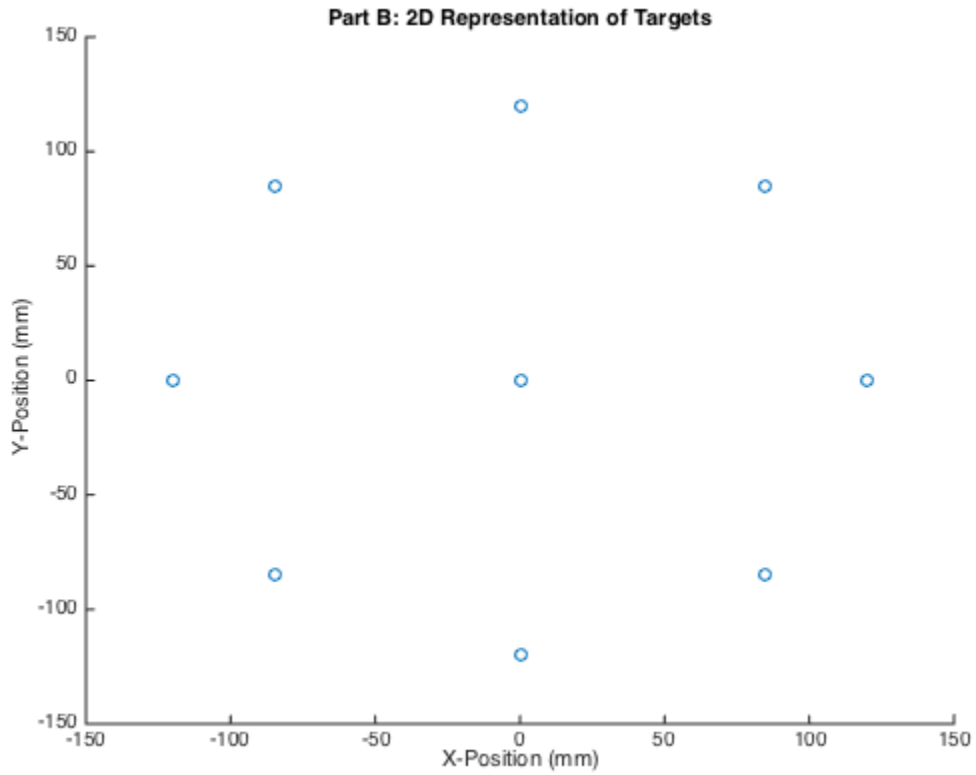
Number of Trials: 506

Part B: Number of Targets

```
targets = zeros(2,n_trials);

for i = 1:n_trials
    targets(:,i) = R(i).target(1:2);
end

u_targets = unique(targets', 'rows');
n_targets = length(u_targets);
fprintf('Number of Unique Targets: %d \n', n_targets)

figure(1)
```

```
scatter(u_targets(:,1), u_targets(:,2))
title('Part B: 2D Representation of Targets')
xlabel('X-Position (mm)')
ylabel('Y-Position (mm)')
```

Number of Unique Targets: 9



Part C: Number of Failed Trials

```
n_successes = 0;

for i = 1:n_trials
    n_successes = n_successes + R(i).isSuccessful;
end

n_fails = n_trials - n_successes;

fprintf('\nNumber of Failed Trials: %d \n', n_fails)
```

Number of Failed Trials: 0

Part D: Sampling Rate

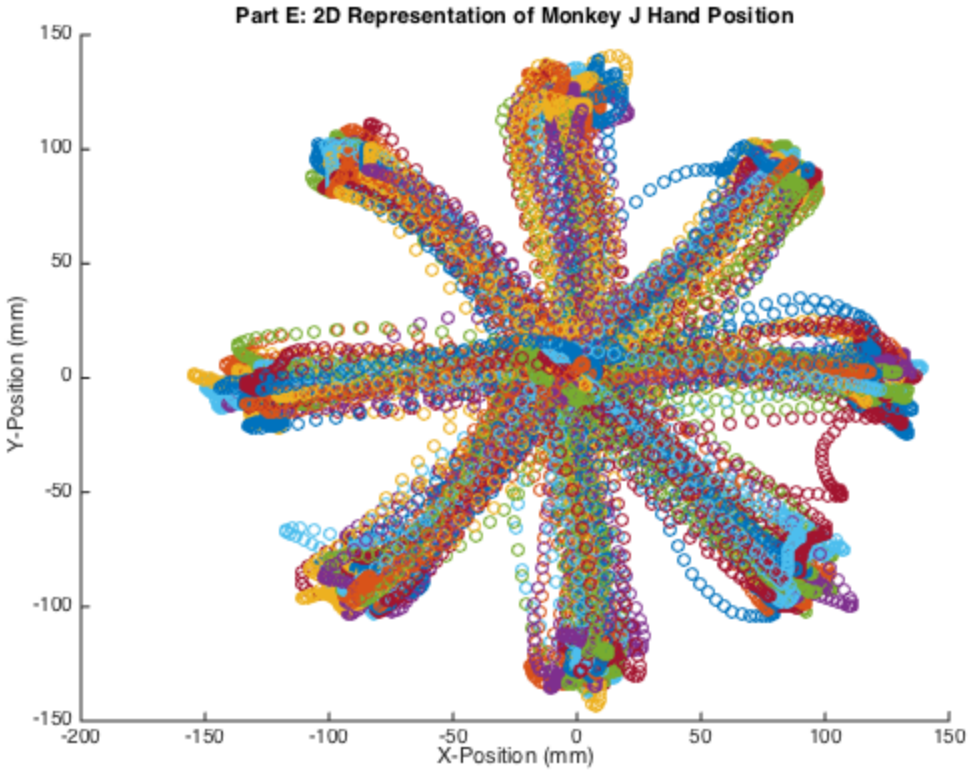
```
rate = zeros(1,n_trials);
```

```
for i = 1:n_trials
    rate(i) = 1000*length(unique(R(i).cursorPos','rows'))/length(R(i).cursorPos);
    % the rate is equal to the number of unique samples per trial divided
    % by the amount of time elapsed during that trial
end

rate_avg = mean(rate);
fprintf('\nAverage Sampling Rate: %2.2f Hz\n', rate_avg)

% No, the system does not sample at 1000 Hz. If it did, then the rate
% should be 1000 since there would be a unique data sample at every time
% point (1 ms). Sometimes the cursor does not move, which accounts for a
% slightly varing rate per trial. It samples at around 60 Hz.
```

Average Sampling Rate: 60.94 Hz

Part E: 2D Representation

```
figure(2)
hold on

for i = 1:n_trials
    u_pos = unique(R(i).cursorPos','rows');
    scatter(u_pos(:,1), u_pos(:,2))
end

hold off

title('Part E: 2D Representation of Monkey J Hand Position')
xlabel('X-Position (mm)')
ylabel('Y-Position (mm)')

% This 2D representation is what we would expect from the experiments. We
% see Monkey J's hand moving back and forth from the center target (0,0) to
% each of the eight targets in a star formation.
```



Part F

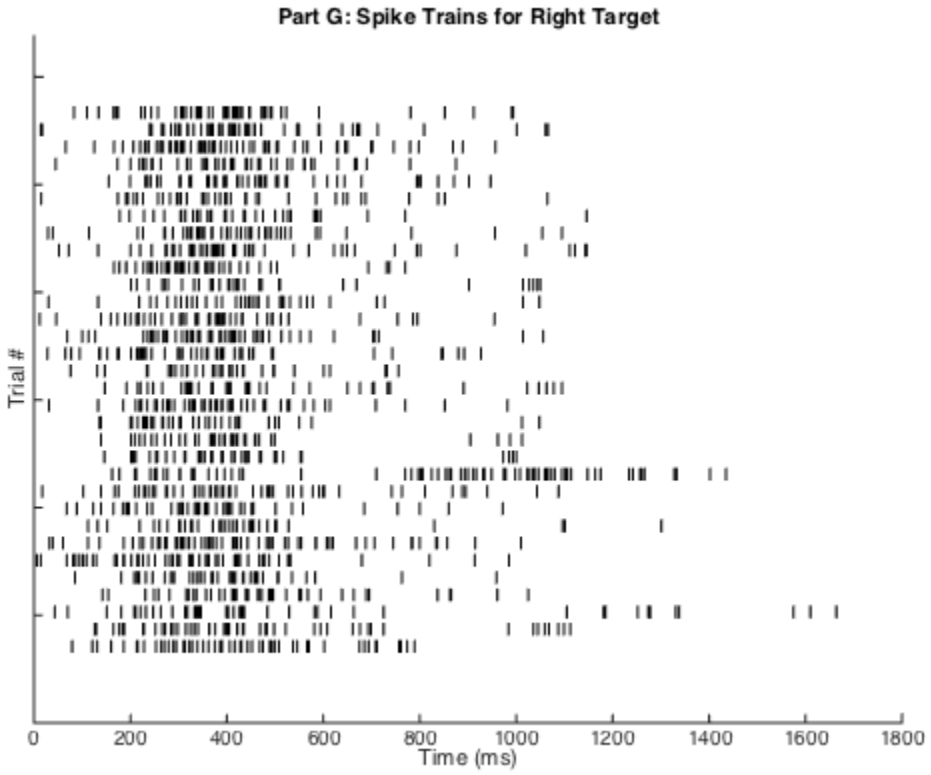
```
n_electrodes = size(full(R(1).spikeRaster),1);

fprintf('\nNumber of Electrode Channels: %d \n', n_electrodes)
```

```
Number of Electrode Channels: 96
```

Part G

```
raster_cell = {};

iter = 0;
for i = 1:n_trials
    if (isequal(R(i).target(1:2),[120; 0]))
        % check if the target is to the right
        iter = iter+1;
        % iterate if correct target
        full_raster = full(R(i).spikeRaster);
        % convert spike train to full matrix
        raster_cell{iter} = find(full_raster(17,:)>0);
        % find indices where there are spikes on electrode 17 and store into
        % a cell for plotRaster
    end
```

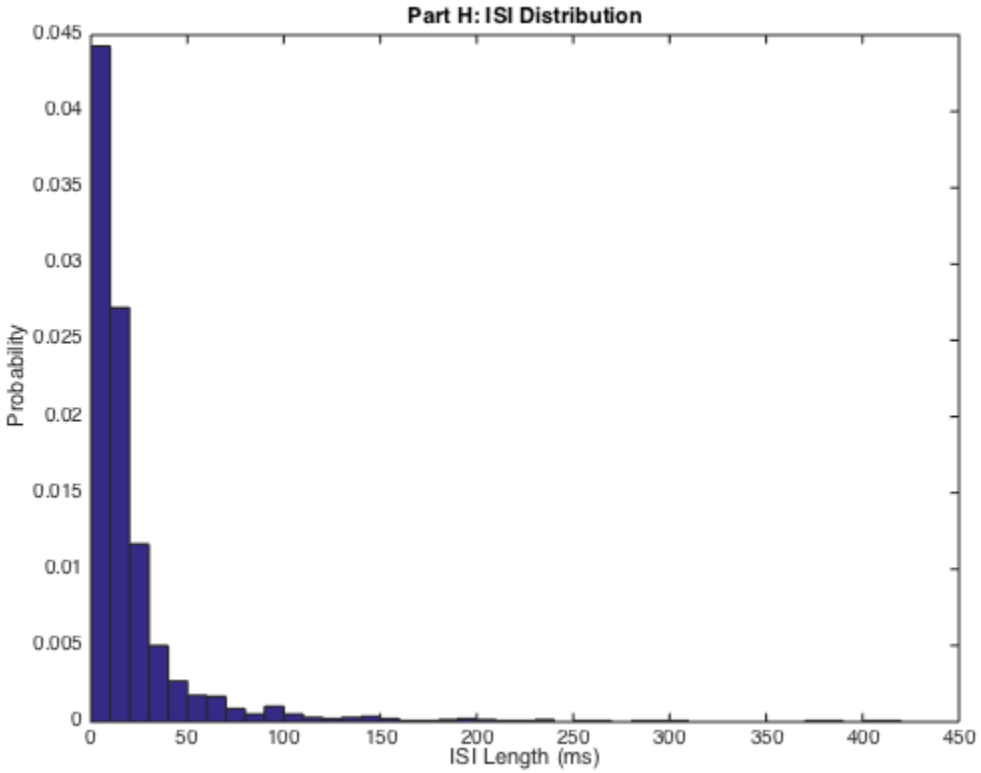
```
end

figure(3)
plotRaster(raster_cell)
ylabel('Trial #')
title('Part G: Spike Trains for Right Target')
```



Part H

```
ISI_dist = [];

for i = 1:length(raster_cell)
    ISI_dist = [ISI_dist, diff(raster_cell{i})];
end

bins = 0:10:max(ISI_dist);

figure(4)
ISI_hist = histcounts(ISI_dist,bins,'Normalization','pdf');
bar(bins(1:end-1), ISI_hist, 'histc')
title('Part H: ISI Distribution')
xlabel('ISI Length (ms)')
ylabel('Probability')

% The data is not spike sorted. We cannot see the refractory periods which
% should be present if each electrode was recording only one neuron.
```

```
% Instead, we see a superposition of several neurons on each electrode and
% no distinct refractory period in the ISI distribution.
```

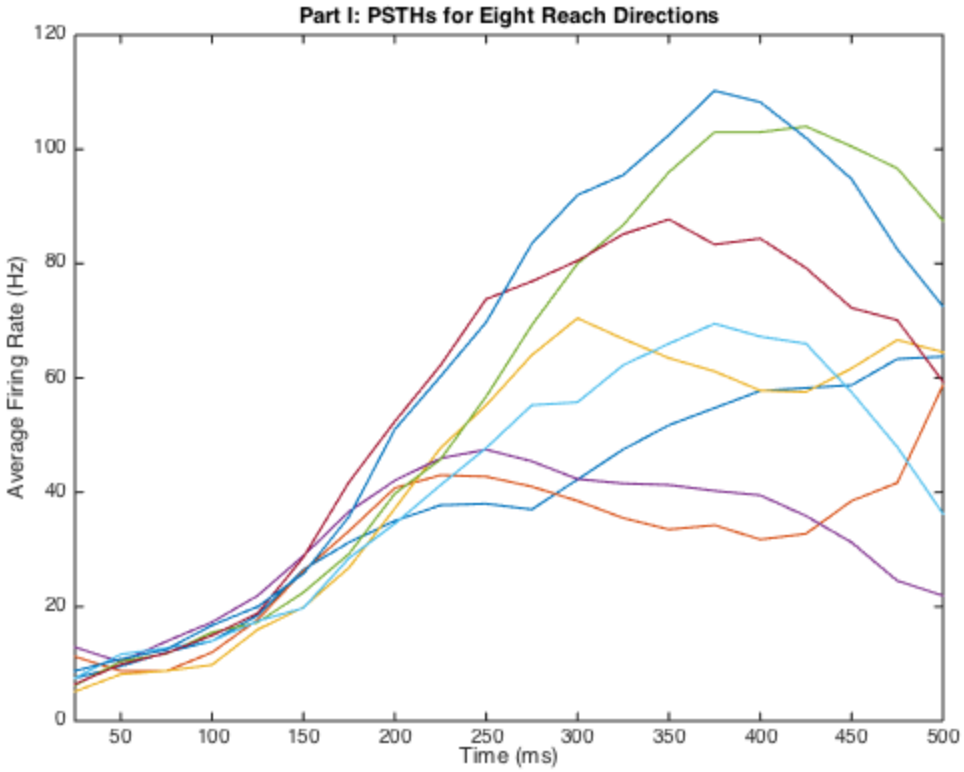


Part I

```
reach_raster = {};
reach_target = u_targets';

iter = zeros(1,n_targets);
reach_raster = cell(1,n_targets);

for i = 1:n_trials
    if (isequal(R(i).target(1:2),reach_target(:,1)))
        % remove if target is in the center
        iter(1) = iter(1)+1;
        % iterate if correct target
        full_reach_raster = full(R(i).spikeRaster);
        reach_raster{1}(:,:,iter(1)) = full_reach_raster(:,1:500);
        % convert spike train to full matrix and save first 500 ms
    end
    if (isequal(R(i).target(1:2),reach_target(:,2)))
        % remove if target is in the center
        iter(2) = iter(2)+1;
        % iterate if correct target
        full_reach_raster = full(R(i).spikeRaster);
        reach_raster{2}(:,:,iter(2)) = full_reach_raster(:,1:500);
```

```
    % convert spike train to full matrix and save first 500 ms
end
if (isequal(R(i).target(1:2),reach_target(:,3)))
    % remove if target is in the center
    iter(3) = iter(3)+1;
    % iterate if correct target
    full_reach_raster = full(R(i).spikeRaster);
    reach_raster{3}(:,:,iter(3)) = full_reach_raster(:,1:500);
    % convert spike train to full matrix and save first 500 ms
end
if (isequal(R(i).target(1:2),reach_target(:,4)))
    % remove if target is in the center
    iter(4) = iter(4)+1;
    % iterate if correct target
    full_reach_raster = full(R(i).spikeRaster);
    reach_raster{4}(:,:,iter(4)) = full_reach_raster(:,1:500);
    % convert spike train to full matrix and save first 500 ms
end
if (isequal(R(i).target(1:2),reach_target(:,5)))
    % remove if target is in the center
    iter(5) = iter(5)+1;
    % iterate if correct target
    full_reach_raster = full(R(i).spikeRaster);
    reach_raster{5}(:,:,iter(5)) = full_reach_raster(:,1:500);
    % convert spike train to full matrix and save first 500 ms
end
if (isequal(R(i).target(1:2),reach_target(:,6)))
    % remove if target is in the center
    iter(6) = iter(6)+1;
    % iterate if correct target
    full_reach_raster = full(R(i).spikeRaster);
    reach_raster{6}(:,:,iter(6)) = full_reach_raster(:,1:500);
    % convert spike train to full matrix and save first 500 ms
end
if (isequal(R(i).target(1:2),reach_target(:,7)))
    % remove if target is in the center
    iter(7) = iter(7)+1;
    % iterate if correct target
    full_reach_raster = full(R(i).spikeRaster);
    reach_raster{7}(:,:,iter(7)) = full_reach_raster(:,1:500);
    % convert spike train to full matrix and save first 500 ms
end
if (isequal(R(i).target(1:2),reach_target(:,8)))
    % remove if target is in the center
    iter(8) = iter(8)+1;
    % iterate if correct target
    full_reach_raster = full(R(i).spikeRaster);
    reach_raster{8}(:,:,iter(8)) = full_reach_raster(:,1:500);
    % convert spike train to full matrix and save first 500 ms
end
if (isequal(R(i).target(1:2),reach_target(:,9)))
    % remove if target is in the center
    iter(9) = iter(9)+1;
    % iterate if correct target
```

```
        full_reach_raster = full(R(i).spikeRaster);
        reach_raster{9}(:,:,iter(9)) = full_reach_raster(:,1:500);
        % convert spike train to full matrix and save first 500 ms
    end
end

dt = 25;
spike_count = cell(1,n_targets);
spike_avg = cell(1,n_targets);

spike_avg_smooth = cell(1,n_targets);

for i = 1:n_targets
    spike_count{i} = binFunc(reach_raster{i},dt);
    spike_avg{i} = sum(spike_count{i}(17,:,:),3)/iter(i);
    spike_avg_smooth{i} = smooth(spike_avg{i},5);
    % taking the average across all trials for each bin
    % iter(i) is the number of trials for each direction
end

figure(5)
t_plot = 25:dt:500;
% time axis
% convert y axis to firing rate (from # spikes per bin)
plot(t_plot, spike_avg_smooth{1}*1000/25, t_plot, spike_avg_smooth{2}*1000/25, ...
    t_plot, spike_avg_smooth{3}*1000/25,...
    t_plot, spike_avg_smooth{4}*1000/25, t_plot, spike_avg_smooth{6}*1000/25, ...
    t_plot, spike_avg_smooth{7}*1000/25,...
    t_plot, spike_avg_smooth{8}*1000/25, t_plot, spike_avg_smooth{9}*1000/25)
axis([25 500 0 120])
title('Part I: PSTHs for Eight Reach Directions')
xlabel('Time (ms)')
ylabel('Average Firing Rate (Hz)')

% tells you avg firing rate for electrode when reaching to target
% plotting lambda(t) for poisson process -- avg firing rate through time
% for 8 conditions
% can be used to generate spike
% neurons on each electrode are uncorrelated
% poisson process generated will be for that group of neurons -- not neuron
% specific
```



Published with MATLAB® R2014b