Table of Contents

EE239AS HW #6

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

Problem 3

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

% from part I of problem 1 (calculating spike counts for each direction)

n_trials = length(R);
n_electrodes = 96;
dt = 25;
```

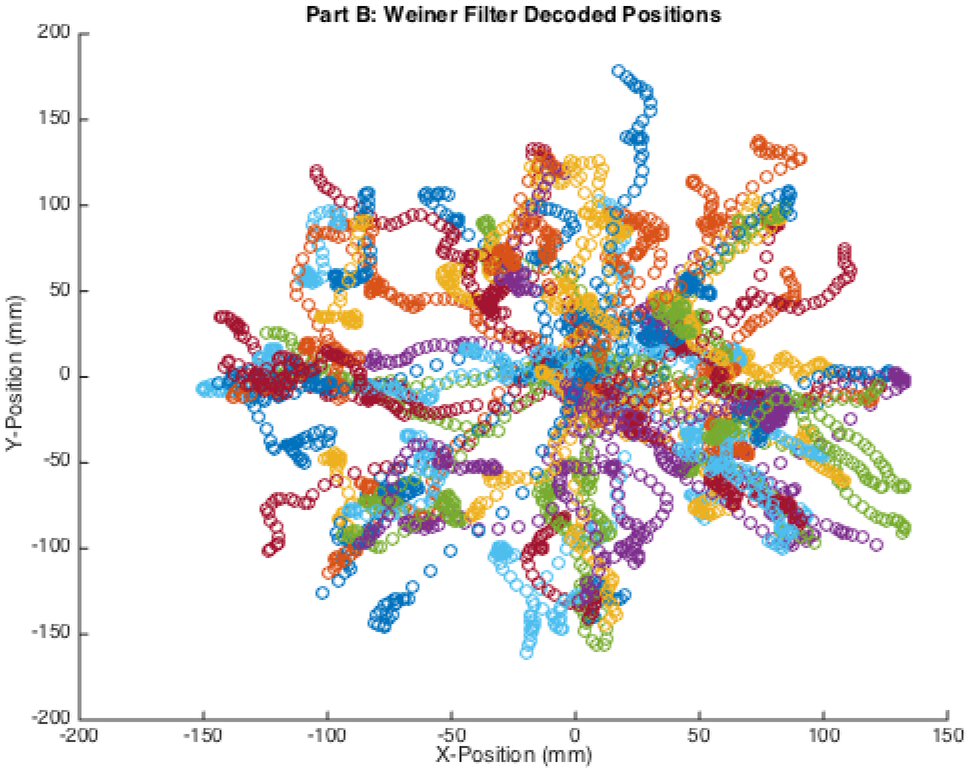
Part A: Y Matrix

```
Y = [R(1:400).spikeRaster];

Y_reach_raster = full(Y);
Y_bin = binFunc(Y_reach_raster, dt);
Y_bin = Y_bin(:, 1:end-1);
% get rid of the last bin which does not have 25 ms worth of data

hist_bins = 4;
Y_W = [];

for i = 0:hist_bins-1
    Y_W = [Y_W; Y_bin(:,hist_bins+1-i:end-i)];
end
% stack the sliding window of Y_bins
% have 5 because considering present bin and 100 ms (4 bins) worth of
% history

Y_W = [Y_W; ones(1,size(Y_W,2))];
fprintf('Dimensions of Y_W:\n')
disp(size(Y_W))
```

```
Dimensions of Y_W:
         385       16461
```

Part B: Fitting Weiner Filter

```
X = [R(1:400).cursorPos];
sample_ind = 1:25:length(X);

pos_bin = X(1:2,sample_ind);

X_bin = diff(pos_bin,1,2)/0.025;

X_W = X_bin(:,hist_bins+1:end);

L_W = X_W*pinv(Y_W);

% Decoding Using Weiner Filter

X_decode = cell(1, 106);
start_pos = zeros(2, 106);
X_test = cell(1, 106);
X_test_pos = cell(1,106);

Y_prev = full(R(400).spikeRaster);
% getting the previous (400th) trial's spiking data
Y_prev_bin = binFunc(Y_prev, dt);
% bin this spiking data
Y_init = [];

for i = 0:hist_bins-1
    Y_init = [Y_init; Y_prev_bin(:,(end-hist_bins+1-i):(end-i))];
end

% stack the sliding window of Y_bins
% have 5 because considering present bin and 100 ms (4 bins) worth of
% history
% take last 3 bins of the spiking data in order to calculate missing 3
% velocities (25, 50, 75 ms)

for i = 1:106
    start_pos(:,i) = R(400+i).cursorPos(1:2,1);
    Y_test = full(R(400+i).spikeRaster);
    Y_test_bin = binFunc(Y_test, dt);

    Y_W_test = [];

    for k = 0:hist_bins-1
        Y_W_test = [Y_W_test; Y_test_bin(:,hist_bins+1-k:end-k)];
    end

    Y_W_test = [Y_init, Y_W_test];
    Y_init = Y_W_test(:,end-hist_bins+1:end);
    % take last 3 bins of the spiking data of the previous trial in order to
    % calculate missing 3 velocities (25, 50, 75 ms) for the next trial

    % add the first three firing rates to the Y matrix
    Y_W_test = [Y_W_test; ones(1, size(Y_W_test,2))];
    % add ones in the last row

    X_decode{i} = L_W * Y_W_test;

    X_test{i} = X_decode{i}*0.025;

    X_test_pos{i}(:,1) = start_pos(:,i);

    for j = 2:length(X_test{i})+1
        X_test_pos{i}(:,j) = X_test_pos{i}(:,j-1) + X_test{i}(:,j-1);
    end

    % 2 x time point x trial

end

figure(1)
hold on
for i = 1:106
    scatter(X_test_pos{i}(1,:), X_test_pos{i}(2,:))
end
hold off
title('Part B: Weiner Filter Decoded Positions')
xlabel('X-Position (mm)')
ylabel('Y-Position (mm)')
```



Part C: Mean-Square Error

```
errors = cell(1,106);
mean_errors = zeros(2,106);

for i = 1:106
    pos_vec = R(400+i).cursorPos(1:2,:);
    sample_ind_pos = [1:25:length(pos_vec), length(pos_vec)];
    pos_vec = pos_vec(:, sample_ind_pos);
    errors{i} = (pos_vec - X_test_pos{i}).^2;
    mean_errors(:,i) = mean(errors{i},2);
end

mean_error_all = sum(mean(mean_errors,2));

fprintf('\nWeiner Filter Mean Square Error: %4.2f\n', mean_error_all)

% The Weiner filter performs better than the optimal linear estimator. In
% the optimal linear estimator decodes only using the present value of the
% firing rate. The Weiner filter uses history to improve this decoding.
```

```
Weiner Filter Mean Square Error: 3455.18
```

Published with MATLAB® R2014b