Homework #2
Due 3 March 2022

---

1. You are recording the activity of a neuron, which is spiking according to a Poisson process with rate λ. At some point during your experiment, the recording equipment breaks down and begins dropping spikes randomly with probability p.

(a) **10 points** Show that the spike train recorded using the broken equipment is a Poisson process.

Bernoulli Process:

$$\begin{aligned}
n &= \#\text{ of discrete time steps} \\
p &= \text{probability of spike dropping} \\
q &= 1-p = \text{probability of spike recorded} \\
\lambda &= nq \rightarrow (\#\text{ of spikes})(\text{chance of success}) \\
q &= \frac{\lambda}{n} \\
B(q,n) &= \mathbb{P}(X=k)\binom{n}{x} q^k (p)^{n-k} \\
\lim_{n\to\infty} \mathbb{P}(X=k) &= \lim_{n\to\infty} \frac{n!}{k!(n-k)!}\left(\frac{\lambda}{n}\right)^k \left(1-\frac{\lambda}{n}\right)^{n-k} \\
&= \left(\frac{1}{k!}\right)\left(\lambda^k\right) \lim_{n\to\infty} \frac{n!}{(n-k)!}\left(\frac{1}{n^k}\right)\left(1-\frac{\lambda}{n}\right)^{n-k} \\
&= \left(\frac{\lambda^k}{k!}\right) \lim_{n\to\infty} \frac{n!}{(n-k)!}\left(\frac{1}{n^k}\right)\left(1-\frac{\lambda}{n}\right)^{n}\left(1-\frac{\lambda}{n}\right)^{-k} \\
&= \left(\frac{\lambda^k}{k!}\right) \lim_{n\to\infty} \left(\frac{n}{n}\right)\left(\frac{n-1}{n}\right)\ldots\left(\frac{n-k+1}{n}\right) e^{-\lambda}(1) \\
&= \left(\frac{\lambda^k}{k!}\right)(1)\left(e^{-\lambda}\right)(1) \\
\mathbb{P}(\lambda,k) &= \left(\frac{\lambda^k e^{-\lambda}}{k!}\right) \rightarrow \text{pdf of Poisson distribution} \\
nq &= n(1-p) = \lambda s \rightarrow \text{as }\#\text{ of spikes} \rightarrow \infty, \text{ chance of success} \rightarrow 0
\end{aligned}$$

(b) **(1 point)** What is the rate of the Poisson process?

$$\begin{aligned}
\lambda_{\text{recorded}} &= \lambda(1-p) \\
\lambda_{\text{dropped}} &= \lambda p
\end{aligned}$$

(c) **(2 points)** What is the distribution of the number of spikes dropped within a $\mathcal{T}$ second interval?

$$\begin{aligned} \mathcal{T} &= \text{time interval} \\ \lambda_{\text{recorded}} &= \lambda(1-p) \\ p(n;\lambda) &= \frac{e^{-\lambda}\lambda^n}{n!} \text{ for } n = 0, 1, ... \\ \lambda &= \lambda\frac{\text{spikes}}{\cancel{\text{sec}}}(p)\mathcal{T}\cancel{\text{sec}} \\ &= \lambda p\mathcal{T} \text{ spikes} \\ p(n;\lambda p\mathcal{T}) &= \frac{e^{-\lambda p\mathcal{T}}\,(\lambda p\mathcal{T})^n}{n!} \text{ for } n = 0, 1, ... \end{aligned}$$

(d) **(2 points)** Given that N spikes were recorded in the $\mathcal{T}$ second interval, how does that change your answer to (c)?

Since the intervals are independent, so knowing how many spikes (N) were recorded in the $\mathcal{T}$ interval won't help you gain any new information in the future. Thus, the answer shouldn't change from (c).

2. Homogenous Poisson process

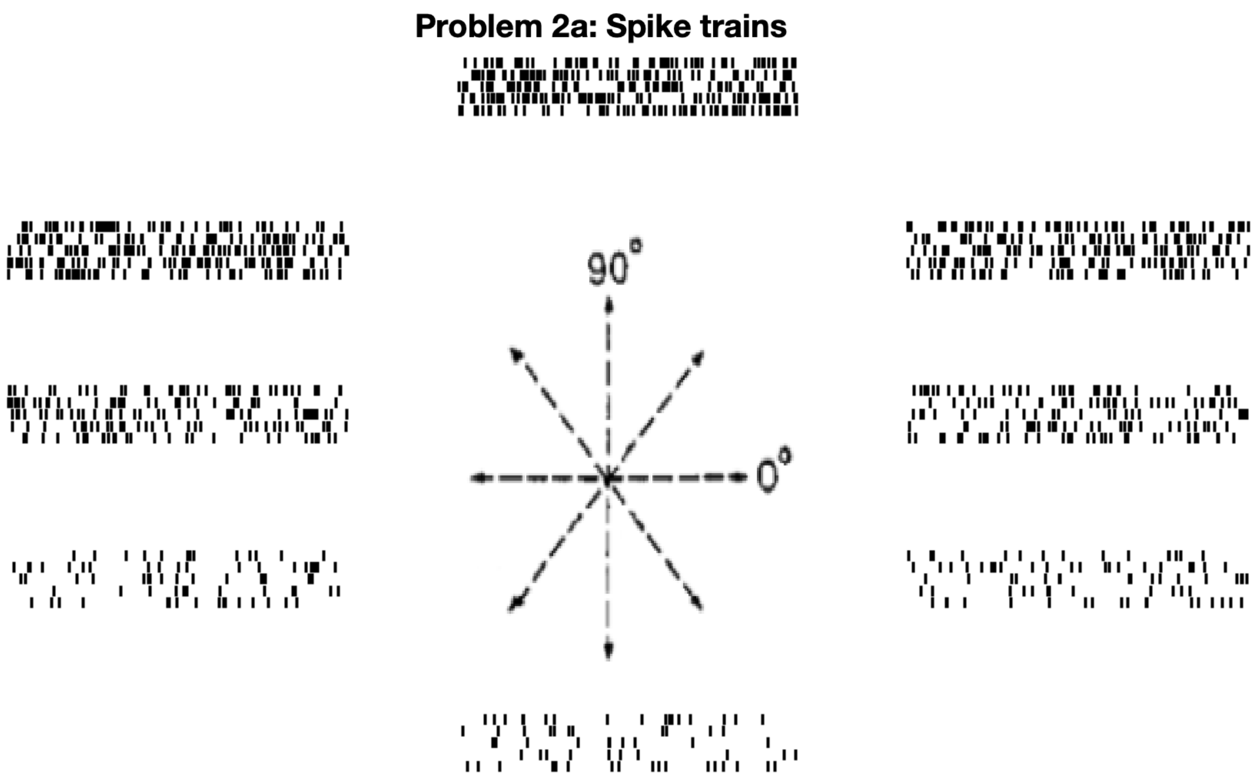
We will consider a simulated neuron that has a cosine tuning curve described in equation (1.15) in TN^1:

$$\lambda(s) = r_0 + (r_{\max} - r_0)\cos(s - s_{\max})$$

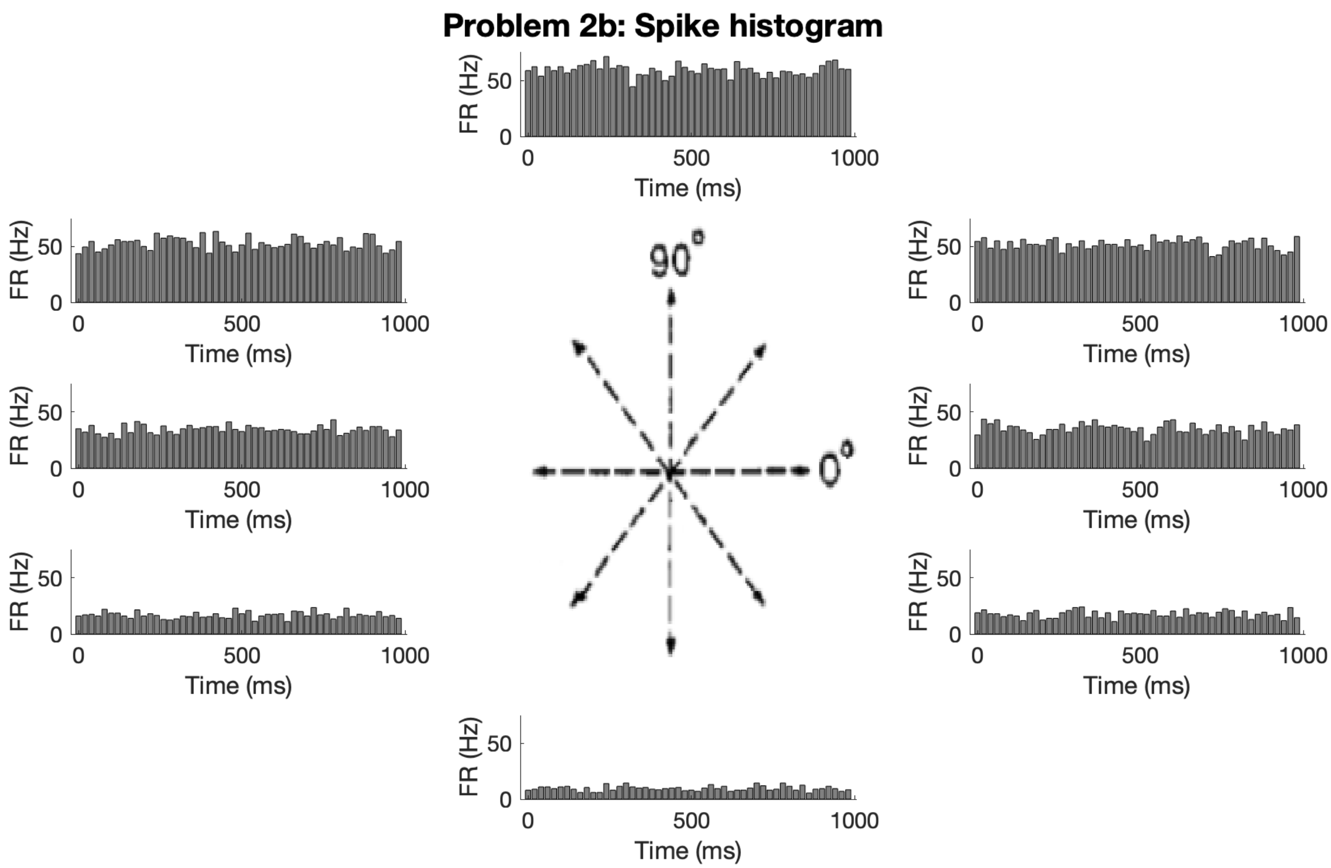
where λ is the firing rate (in spikes per second), s is the reaching angle of the arm, $x_{\max}$ is the reaching angle associated with the maximum response $r_{\max}$, and r_0 is an offset that shifts the tuning curve up from the zero axis. Let $r_0 = 35$, $r_{\max} = 60$, and $s_{\max} = \pi/2$.

(a) **(5 points)** Spike trains

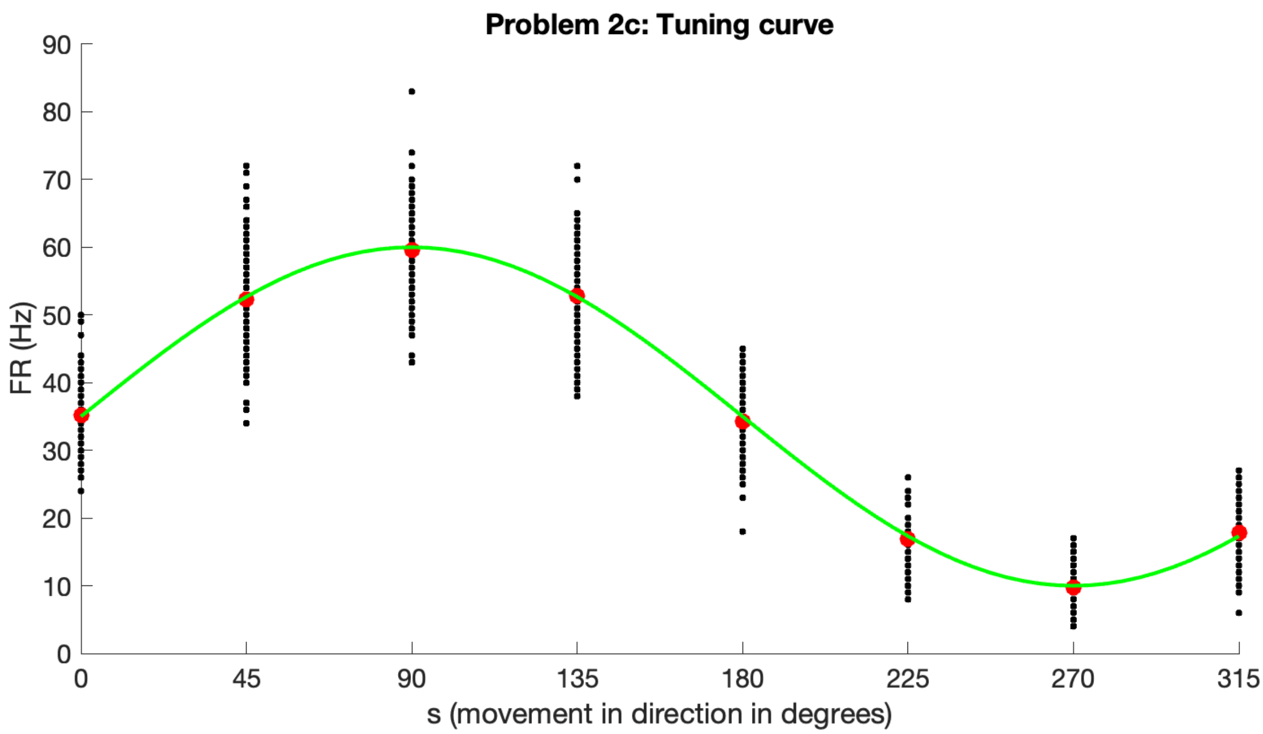
For each of the following reach angles ($s = k * \pi/4$, where $k = 0, 1, ..., 7$)), generate 100 spike trains according to a homogenous Poisson process. Each spike train should have a duration of 1 second. Plot 5 spike trains for each reaching angle in the same format as shown in Figure 1.6(A) in TN.



(b) **(5 points)** Spike histogram
For each reaching angle, find the spike histogram by taking spike counts in non- overlapping 20 ms bins, then averaging across the 100 trials. Plot the 8 resulting spike histograms around a circle, as in part (a). The spike histograms should have firing rate (in spikes / second) as the vertical axis and time (in msec, not time bin index) as the horizontal axis. The bar command in Matlab can be used to plot histograms.

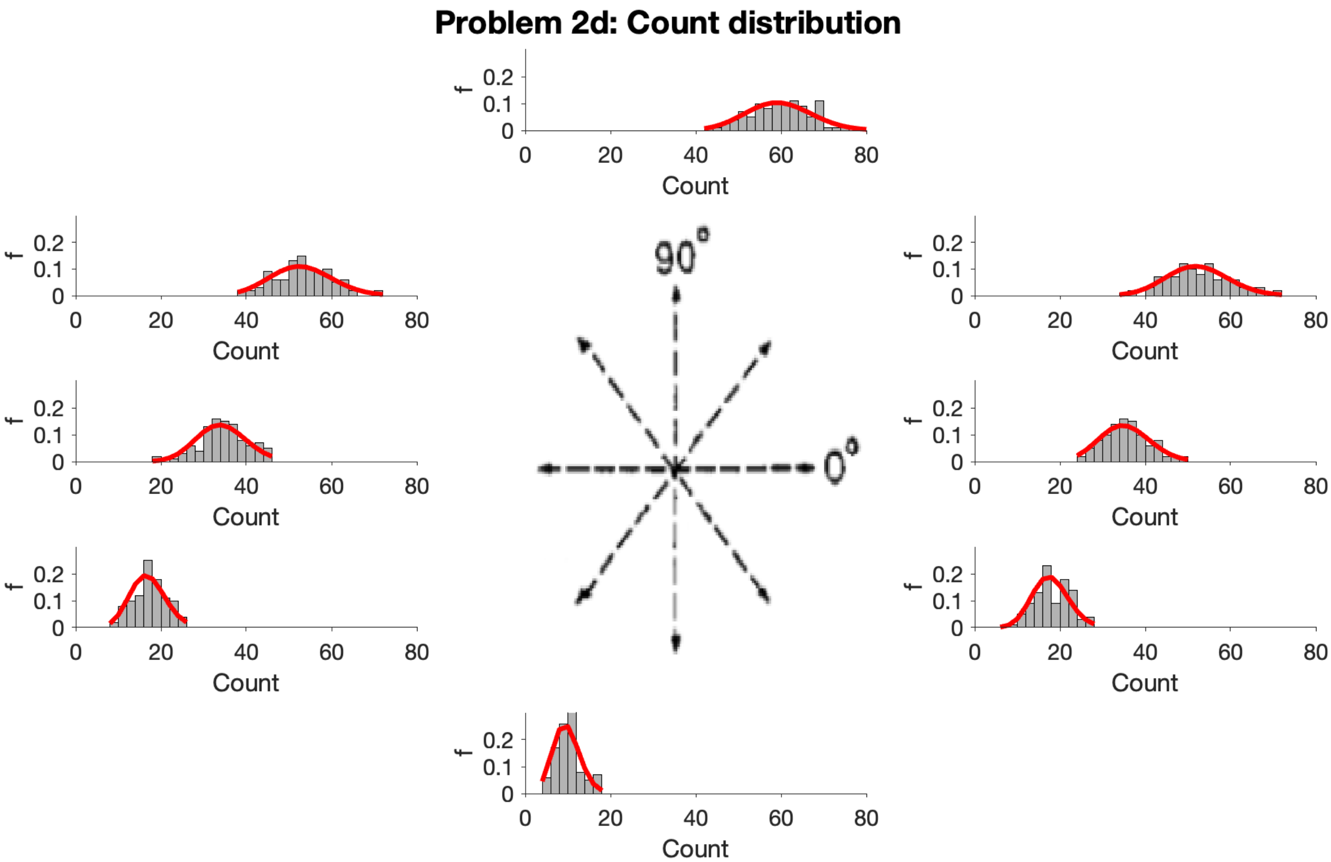


(c) **(5 points)** Tuning curve
For each trial, count the number of spikes across the entire trial. Plots these points on the axes shown in Figure 1.6(B) in TN. There should be 800 points in the plot (but some points may be on top of each other due to the discrete nature of spike counts). For each reaching angle, find the mean firing rate across the 100 trials, and plot the mean firing rate using a red point on the same plot. Now, plot the tuning curve (defined in (1)) of this neuron in green on the same plot. Do the mean firing rates lie near the tuning curve?



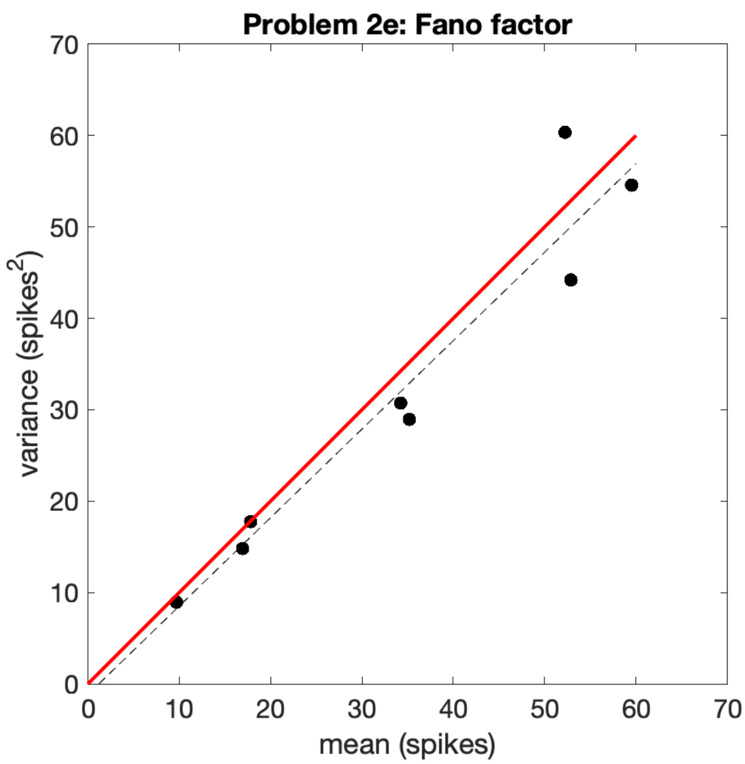
Yes.

(d) **(5 points)** Count distribution
For each reaching angle, plot the *normalized* distribution (i.e., normalized so that the area under the distribution equals one) of spike counts (using the same counts from part (c)). Plot the 8 distributions around a circle, as in part (a). Fit a Poisson distribution to each empirical distribution and plot it on top of the corresponding empirical distribution. Are the empirical distributions well-fit by Poisson distributions?



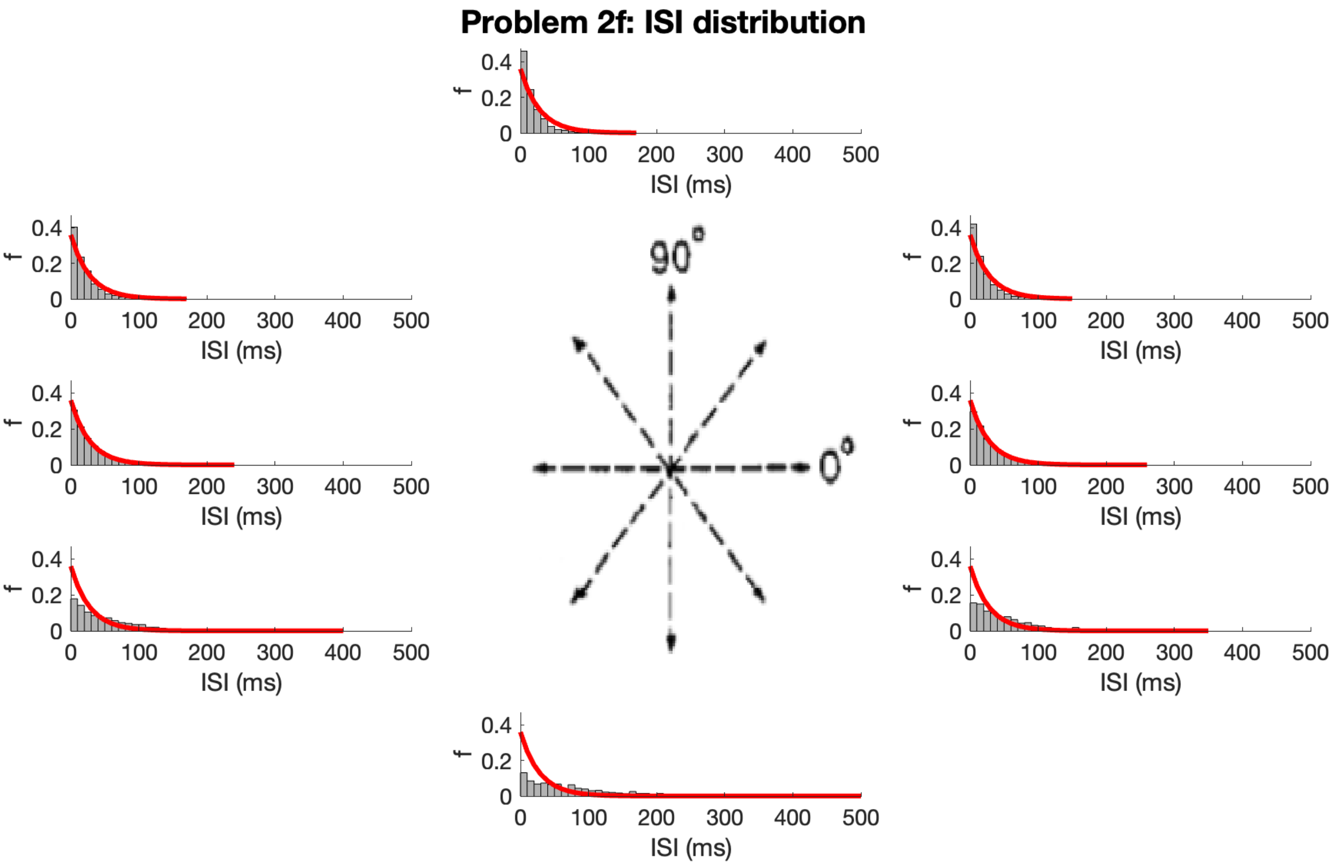
Yes.

(e) **(5 points)** Fano factor
For each reaching angle, find the mean and variance of the spike counts across the 100 trials (using the same spike counts from part (c)). Plot the obtained mean and variance on the axes shown in Figure 1.14(A) in TN. There should be 8 points in this plot – one per reaching angle. Do these points lie near the 45 deg diagonal, as would be expected of a Poisson distribution?



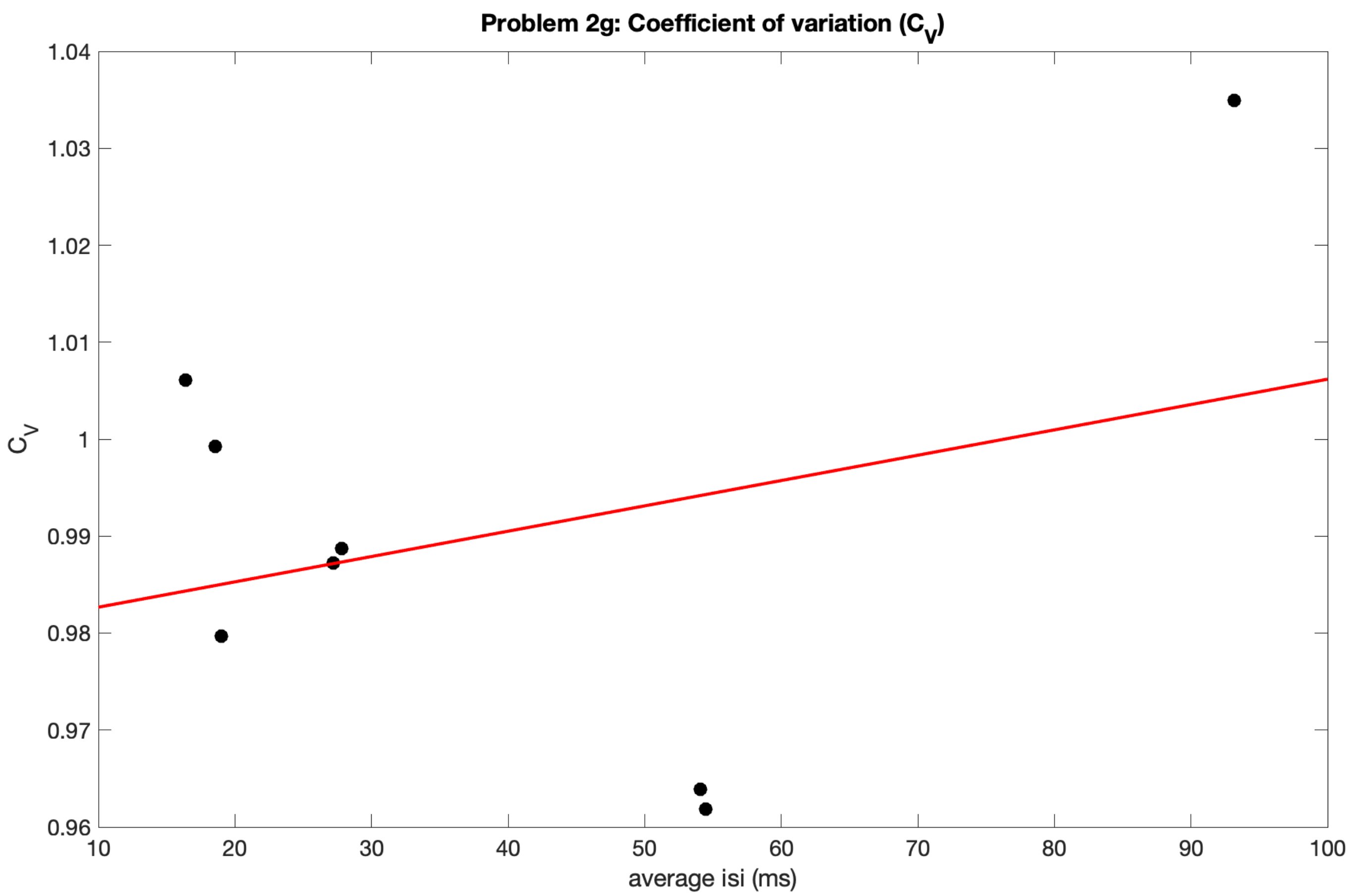
They lie very close to the 45 deg diagonal (red line)

(f) **(5 points)** Interspike interval (ISI) distribution
For each reaching angle, plot the normalized distribution of ISIs. Plot the 8 distributions around a circle, as in part (a). Fit an exponential distribution to each empirical distribution and plot it on top of the corresponding empirical distribution. Are the empirical distributions well-fit by exponential distributions?



Yes.

(g) **(5 points)** Coefficient of variation (C_V)
For each reaching angle, find the average ISI and C_V of the ISIs. Plot the resulting values on the axes shown in Figure 1.16 in TN. There should be 8 points in this plot. Do the C_V values lie near unit, as would be expected of a Poisson process?



Yes, the variation in C_V on the y-axis is very small.

3. Inhomogenous Poisson process

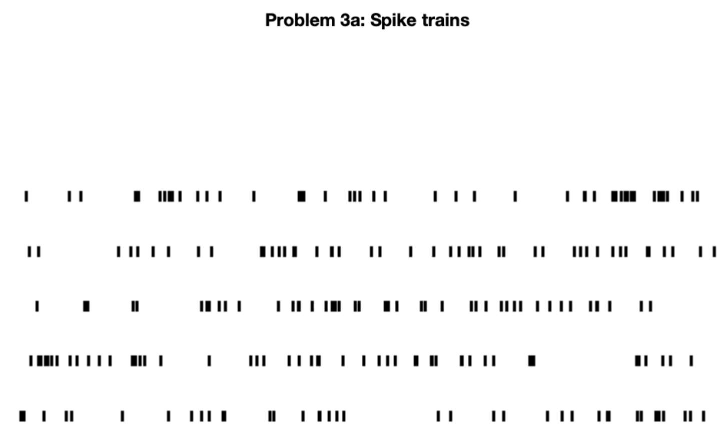
In this problem, we will use the same simulated neuron as in Problem 2, but now the reaching angle s will be time-dependent with the following form:

$$s(t) = t^2 * \pi$$

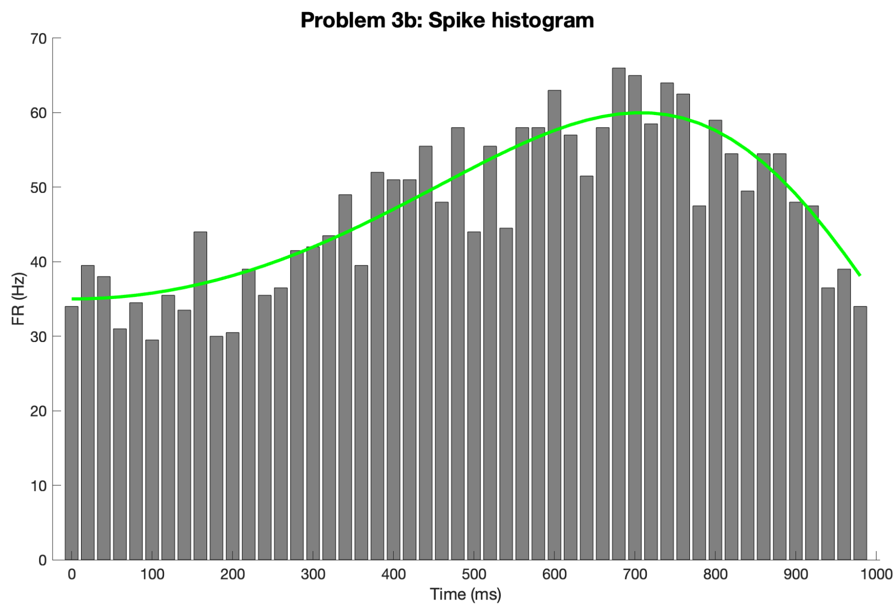
where t ranges between 0 and 1 second.

(a) **(5 points)** Spike trains

Generate 100 spike trains, each 1 second in duration, according to an inhomogeneous Poisson process with a firing rate profile defined by (1) and (2). Plot 5 of the generated spike trains.

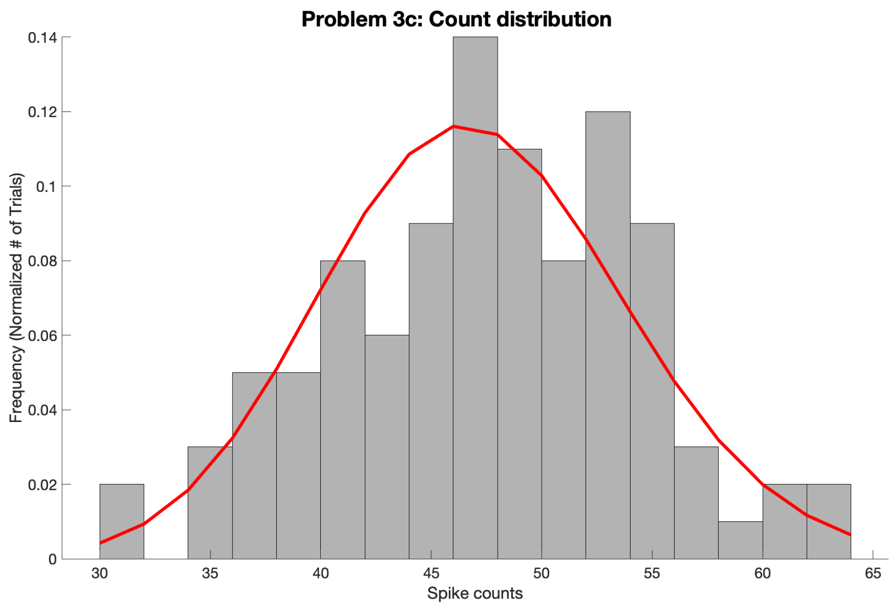


(b) **(5 points)** Spike histogram
Plot the spike histogram by taking spike counts in non-overlapping 20 ms bins, then averaging across the 100 trials. The spike histogram should have firing rate (in spikes / second) as the vertical axis and time (in msec, not time bin index) as the horizontal axis. Plot the expected firing rate profile defined by (1) and (2) on the same plot. Does the spike histogram agree with the expected firing rate profile?



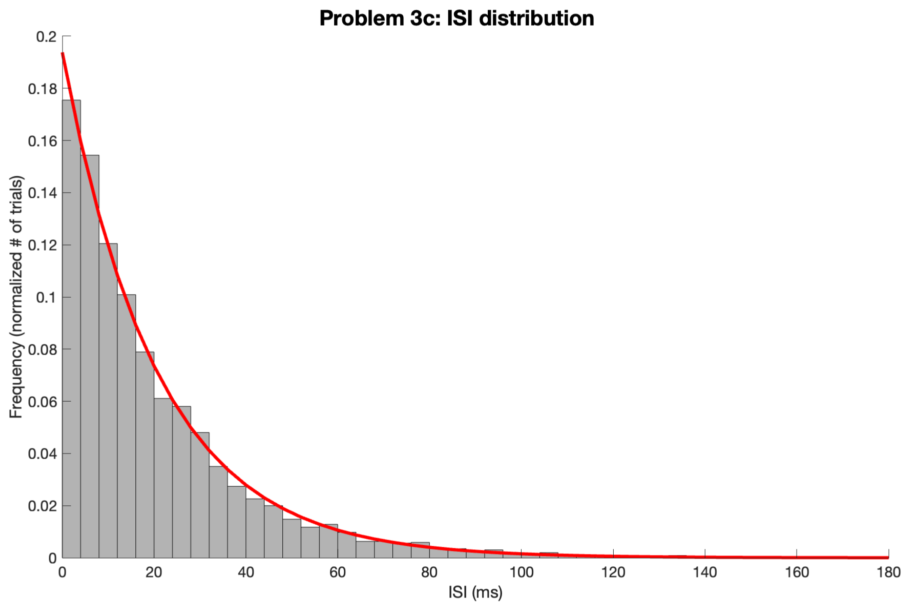
Yes.

(c) **(5 points)** Count distribution
For each trial, count the number of spikes across the entire trial. Plot the nor- malized distribution of spike counts. Fit a Poisson distribution to this empirical distribution and plot it on top of the empirical distribution. Should we expect the spike counts to be Poisson-distributed?



Kind-of, the spike rate is now changing at each time step so this is harder to fit.

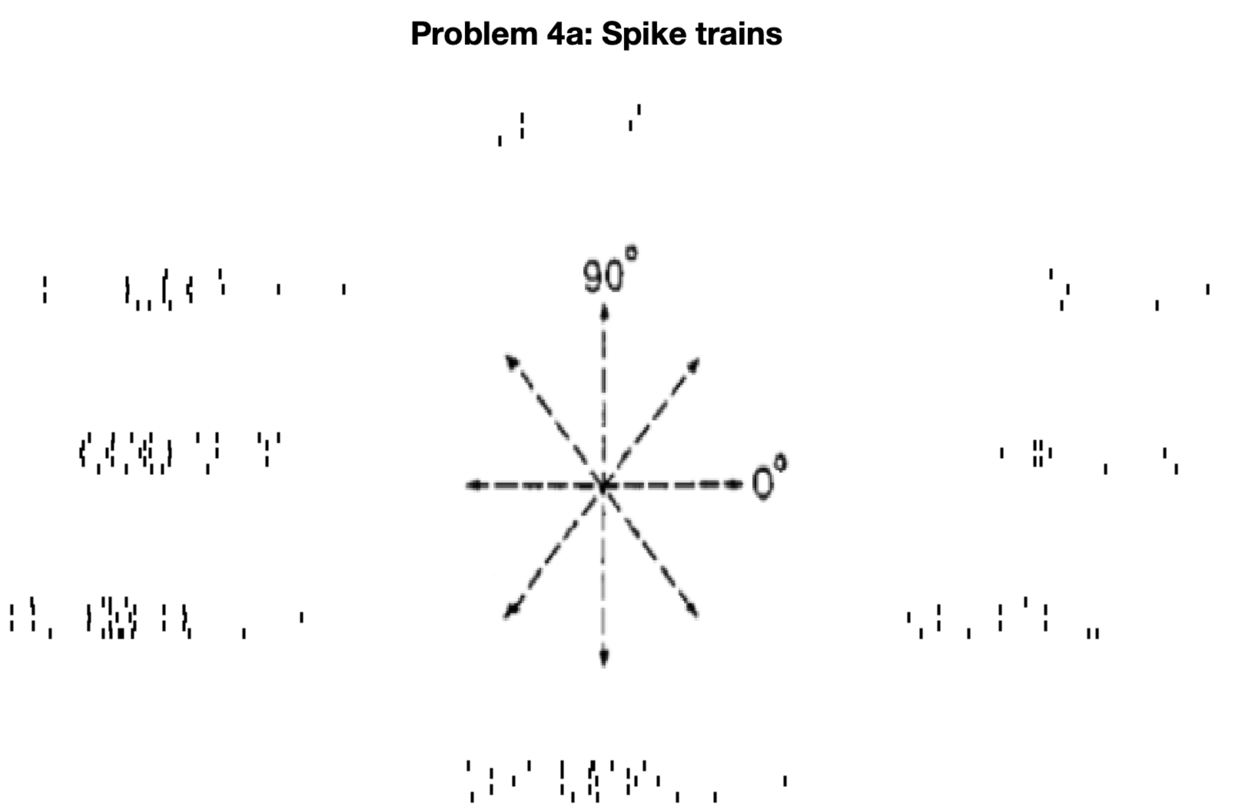
(d) **(5 points)** ISI distribution
Plot the normalized distribution of ISIs. Fit an exponential distribution to the empirical distribution and plot it on top of the empirical distribution. Should we expect the ISIs to be exponentially-distributed?



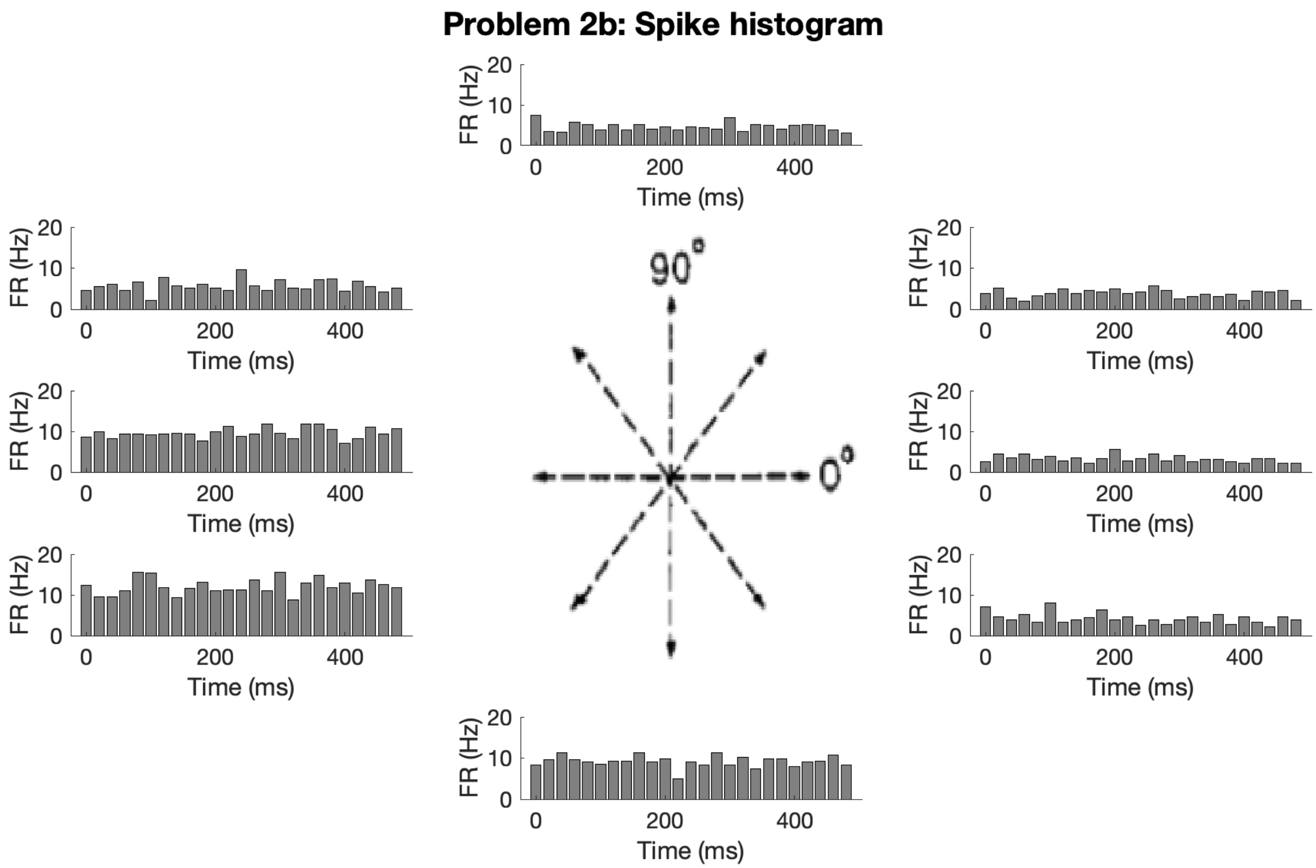
Kind-of, the exponential fits the ISIs better than the poisson fit the counts.

4. Real neural data
We will analyze real neural data recorded using a 100-electrode array in premotor cortex of a macaque monkey. The dataset can be found on the Canvas course website under "Files → Data sets → ps2 data.mat".
The following describes the data format. The .mat file has a single variable named **trial**, which is a structure of dimensions (182 trials) x (8 reaching angles). The structure contains spike trains recorded from a single neuron while the monkey reached 182 times along each of the 8 different reaching angles (where the trials of different reaching angles were interleaved). The spike train the the nth trial of the kth reaching angle is contained in **trial(n,k).spikes**, where **n = 1,...,182** and **k = 1,...,8**. The indices k = 1,...,8 corespond with reaching angles $\frac{3}{180}\pi, \frac{70}{180}\pi, \frac{110}{180}\pi, \frac{150}{180}\pi, \frac{190}{180}\pi, \frac{230}{180}\pi, \frac{310}{180}\pi, \frac{350}{180}\pi$, respectively. The reaching angles are not evenly spaced around the circle due to experimental constraints that are beyond the scope of this problem set.
A spike train is represented as a sequence of zeros and ones, where time is discretized in 1 ms steps. A zero indicates that the neuron did not spike in the 1 ms bin, whereas a one indicates that the neuron spiked once in the 1 ms bin. Due to the refractory period, it is not possible for a neuron to spike more than once within a 1 ms bin. Each spike train is 500 ms long and is, thus, represented by a 1×500 vector.

(a) **(5 points)** Spike trains
Plot 5 spike trains for each reaching angle in the same format as shown in Figure 1.6(A) in TN.



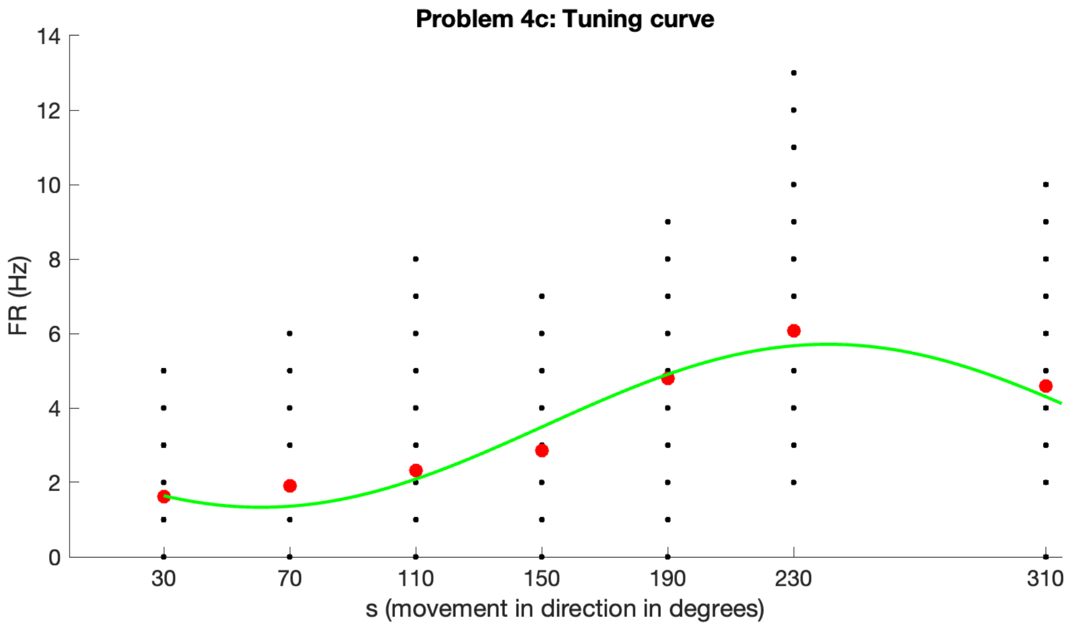
(b) **(5 points)** Spike histogram
For each reaching angle, find the spike histogram by taking spike counts in non- overlapping 20 ms bins, then averaging across the 182 trials. The spike histograms should have firing rate (in spikes / second) as the vertical axis and time (in msec, not time bin index) as the horizontal axis. Plot the 8 resulting spike histograms around a circle, as in part (a).



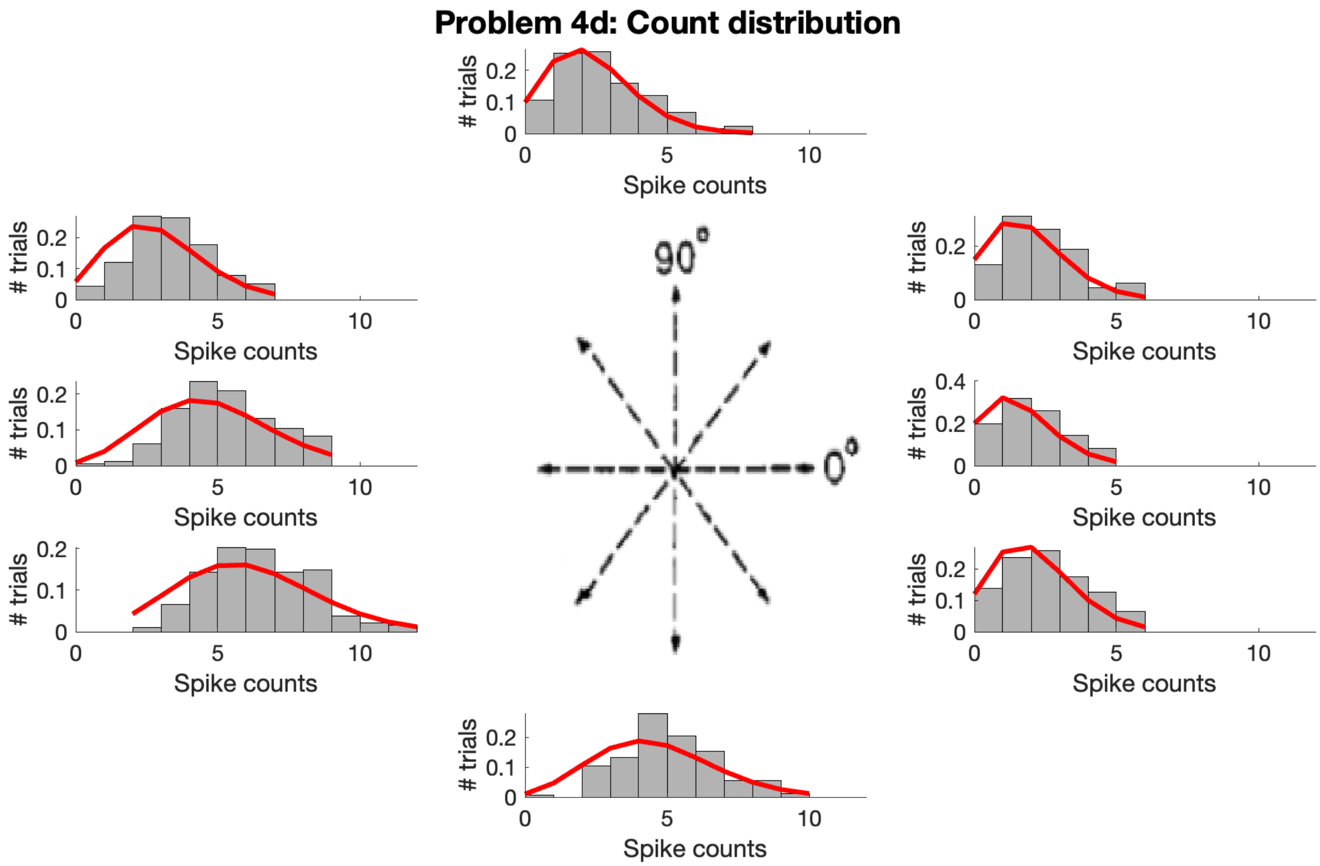
(c) **(5 points)** Tuning curve
For each trial, count the number of spikes across the entire trial. Plots these points on the axes shown in Figure 1.6(B) in TN. There should be $182 * 8$ points in the plot (but some points may be on top of each other due to the discrete nature of spike counts). For each reaching angle, find the mean firing rate across the 182 trials, and plot the mean firing rate using a red point on the same plot. Then, fit the cosine tuning curve (1) to the 8 red points by minimizing the sum of squared errors

$$\sum_{i=1}^{8} \left(\lambda(s_i) - r_0 - (r_{\max} - r_0)\cos\left(s_i - s_{\max}\right)\right)^2$$

with respect to the parameters r_0, $r_{\max}$, and $s_{\max}$. (Hint: this can be done using linear regression.) Plot the resulting tuning curve of this neuron in green on the same plot.

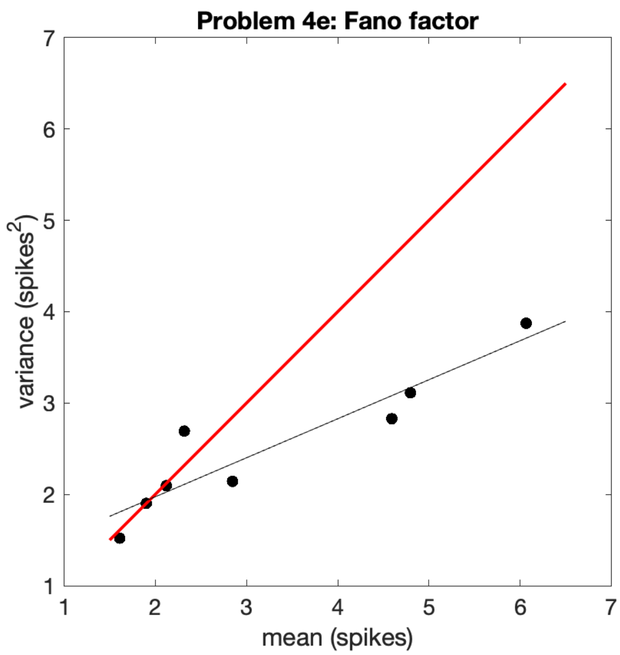


(d) **(5 points)** Count distribution
For each reaching angle, plot the normalized distribution of spike counts (using the same counts from part (c)). Plot the 8 distributions around a circle, as in part (a). Fit a Poisson distribution to each empirical distribution and plot it on top of the corresponding empirical distribution. Why might the empirical distributions differ from the idealized Poisson distributions?



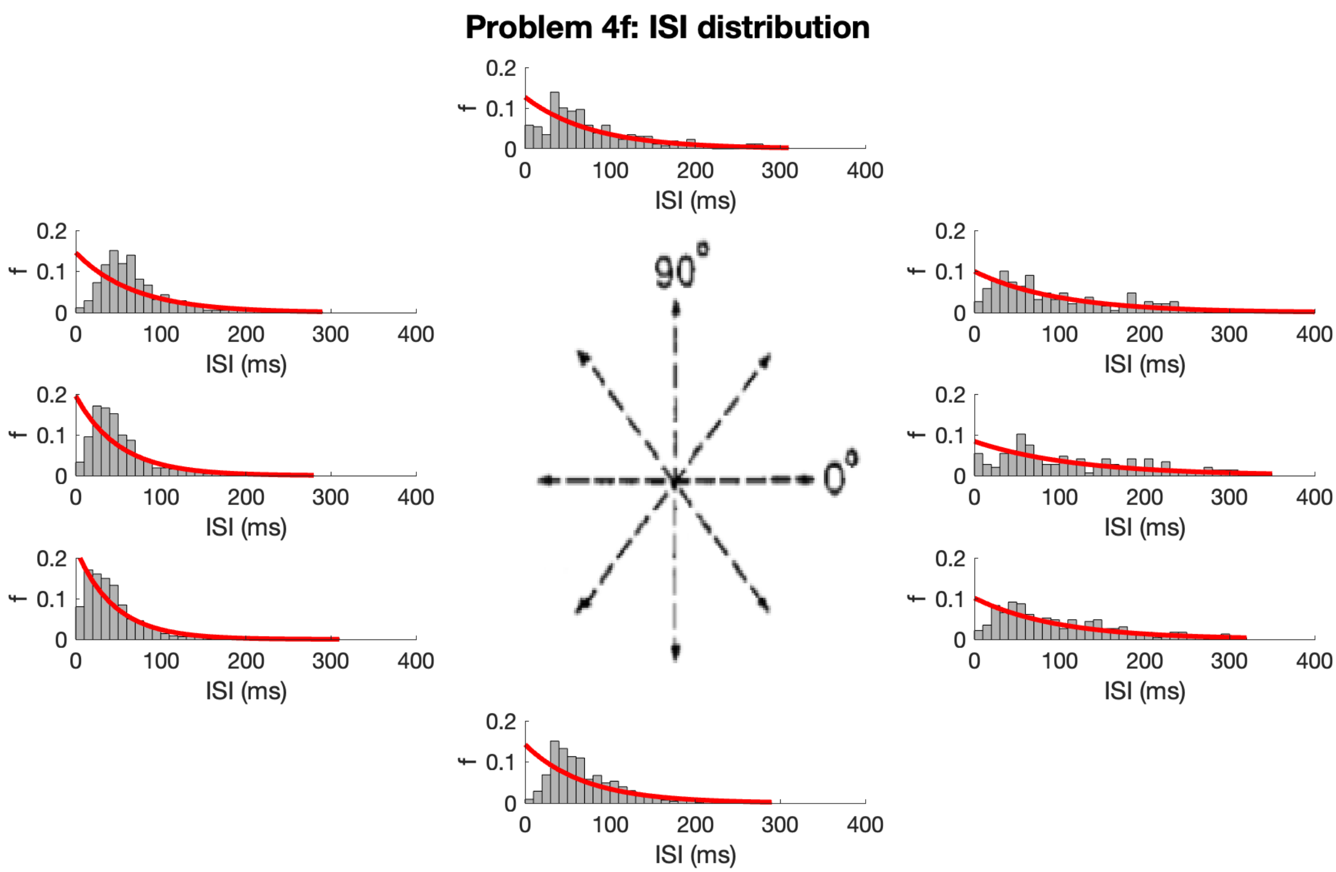
There is more variability in spike rate and this data is real, not idealized.

(e) **(5 points)** Fano factor
For each reaching angle, find the mean and variance of the spike counts across the 182 trials (using the same spike counts from part (c)). Plot the obtained mean and variance on the axes shown in Figure 1.14(A) in TN. There should be 8 points in this plot – one per reaching angle. Do these points lie near the 45 deg diagonal, as would be expected of a Poisson distribution?



It lies somewhat close to the 45 deg diagonal, not as good as the idealized data.

(f) **(5 points)** Interspike interval (ISI) distribution
For each reaching angle, plot the normalized distribution of ISIs. Plot the 8 distributions around a circle, as in part (a). Fit an exponential distribution to each empirical distribution and plot it on top of the corresponding empirical distribution. Why might the empirical distributions differ from the idealized exponential distributions?



ISIs from idealized data don't account for refractory periods.

A spike can't occur or is very unlikely to occur after a very short amount of time (1-5ms).

Contents


```
clear all;
```

```
Error using dbstatus
File: /Users/kendranoneman/School/22-spring/nsp/hw/hw2/hw2.m Line: 127 Column: 24
Incorrect use of '=' operator. Assign a value to a variable using '=' and compare values for equality using '=='.
```

Problem 2: Homogeneous Poisson Process

constant spike rate

```
k = 0:7;
s = (pi/4)*k; % reaching angle of arm

r0 = 35;
rmax = 60;
smax = pi/2;

rateS = num2cell(r0 + (rmax - r0)*cos(s - smax),1)';

% (a) Spike trains
n = 100; % # of spike trains
T = 1;
rate = cellfun(@(r) num2cell(poissrnd(r*T,n,1)), rateS, 'uni', 0); % number of spikes in interval [0, T]
Tn = [];
for i=1:length(k)
    Tn = [Tn cellfun(@(t) sort(rand(t,1)), rate{i,1}, 'uni',0)]; % spike times
end

% Plot 5 random spike trains for each reaching angle
rows = randi([1 size(Tn,1)],5,1);
Tn_plot = Tn(rows,:);
sublocs = {9,6,2,4,7,10,14,12};
f2a = figure;
f2a.Position = [100, 100, 1000, 600];
subplot(5,3,[5, 8, 11])
image( imread('f1.png') );
axis off
set(gcf,'color','w')
for p=1:length(k)
    subplot(5,3,sublocs{p})
    for ss=1:5
        x = Tn_plot{ss,p};
        y = ss*ones(1,length(x));
        plot(x,y,'k|','LineWidth',2);
        hold on;
    end
    axis padded
    axis off
    box off
    set(gcf,'color','w');
    xlim([0 1])
    ylim([-5 5])
end
sgtitle('Problem 2a: Spike trains','FontSize',20,'FontWeight','bold');
saveas(f2a,'figs/prob2a.png');

% (b) Spike histogram
```

```
binWidth = 20; % ms
bins =  [0:(binWidth)/1000:1]; % s
for a = 1:8
    spks = Tn(:,a);
    cnts = [];
    for t = 1:length(spks)
        thisTrial = spks{t};
        [N,~] = histcounts(thisTrial,bins);
        cnts = [cnts; N];
    end
    mnCnts{a} = mean(cnts,1)*(1000/binWidth);
end

sublocs = {9,6,2,4,7,10,14,12};
f2b = figure;
f2b.Position = [100, 100, 1000, 600];
subplot(5,3,[5, 8, 11])
image( imread('f1.png') );
axis off
set(gcf,'color','w')
for p=1:length(k)
    subplot(5,3,sublocs{p})
    x = bins(1:end-1)*1000; % ms
    bar(x,mnCnts{p},'FaceColor',[0.5 0.5 0.5])
    axis padded
    set(gcf,'color','w');
    xlabel('Time (ms)');
    ylabel('FR (Hz)');
    ylim([0 75])
    set(gca,'FontSize',14)
    box off
end
sgtitle('Problem 2b: Spike histogram','FontSize',20,'FontWeight','bold');
saveas(f2b,'figs/prob2b.png');

% (c) Tuning curve
kTun = 0:0.1:7;
sTun = (pi/4)*kTun; % reaching angle of arm
tunCurve = num2cell(r0 + (rmax - r0)*cos(sTun - smax),1)';

f2c = figure;
f2c.Position = [100, 100, 800, 400];
totCnts = cellfun(@length,Tn);
x = s*(180/pi);
xTun = sTun*(180/pi);
for tt=1:size(totCnts,1)
    plot(x,totCnts(tt,:),'k.','MarkerSize',10)
    hold on
end
plot(x,mean(totCnts,1),'r.','MarkerSize',30)
plot(xTun,cell2mat(tunCurve),'g-','LineWidth',2)
xticks(x);
xlabel('s (movement in direction in degrees)');
ylabel('FR (Hz)');
xlim([0 315]);
set(gcf,'color','w');
set(gca,'FontSize',14)
box off
title('Problem 2c: Tuning curve');
saveas(f2c,'figs/prob2c.png');

% (d) Count distribution
sublocs = {9,6,2,4,7,10,14,12};
f2d = figure;
f2d.Position = [100, 100, 1000, 600];
subplot(5,3,[5, 8, 11])
image( imread('f1.png') );
axis off
set(gcf,'color','w')
```

```
for p=1:length(k)
    subplot(5,3,sublocs{p})
    bw = 2;
    h = histogram(totCnts(:,p), 'normalization', 'probability','BinWidth',bw,'FaceColor',[0.5 0.5 0.5]);
    hold on
    edges = h.BinEdges;=
    y = poisspdf(floor(edges), mean(totCnts(:,p)));
    plot(edges,y.*2,'r-','LineWidth',3);
    axis padded
    box off
    set(gca, 'TickDir', 'out','FontSize',18)
    set(gcf,'color','w');
    xlabel('Count');
    ylabel('f');
    xlim([0 80]);
    ylim([0 0.3]);
    set(gca,'FontSize',14)
end
sgtitle('Problem 2d: Count distribution','FontSize',20,'FontWeight','bold');
saveas(f2d,'figs/prob2d.png');

% (e) Fano factor
mnTotCnts = mean(totCnts,1);
varTotCnts = var(totCnts);

f2e = figure;
plot(mnTotCnts,varTotCnts,'k.','MarkerSize',20);
hold on;
hline = refline(1,0);
hline.Color = 'r';
hline.LineWidth = 2;
lline = lsline;
lline.LineStyle = '--';
xlim([0 70])
ylim([0 70])
axis square
xlabel('mean (spikes)');
ylabel('variance (spikes^2)');
set(gcf,'color','w');
set(gca,'FontSize',14)
title('Problem 2e: Fano factor');
saveas(f2e,'figs/prob2e.png');

% (f) ISI distribution
isi = cellfun(@(t) diff(t), Tn, 'uni', 0);

for j=1:size(isi,2)
    isi_cat = [];
    for i=1:size(isi,1)
        isi_cat = [isi_cat; isi{i,j}];
    end
    isiAll{j} = isi_cat;
end

sublocs = {9,6,2,4,7,10,14,12};
f2f = figure;
f2f.Position = [100, 100, 1000, 600];
subplot(5,3,[5, 8, 11])
image( imread('f1.png') );
axis off
set(gcf,'color','w')
for p=1:8
    subplot(5,3,sublocs{p})
    bw = 10;
    h = histogram(isiAll{p}.*1000,'Normalization','probability','BinWidth',bw,'FaceColor',[0.5 0.5 0.5]);
    hold on
    edges = h.BinEdges;
    y = exppdf(edges,mean(isiAll{1}).*1000);
    plot(edges,y.*bw,'r-','LineWidth',3);
```

```
    axis padded
    box off
    set(gcf,'color','w');
    xlabel('ISI (ms)');
    ylabel('f');
    xlim([0 500]);
    ylim([0 0.47])
    set(gca,'FontSize',14)
end
sgtitle('Problem 2f: ISI distribution','FontSize',20,'FontWeight','bold');
saveas(f2f,'figs/prob2f.png');

% (g) Coefficient of variation (Cv)
mnISI = cellfun(@mean, isiAll);
cvISI = cell2mat(cellfun(@(y) sqrt(var(y))/mean(y), isiAll, 'uni', 0));

f2g = figure;
f2g.Position = [100, 100, 1000, 600];
plot(mnISI.*1000,cvISI,'k.','MarkerSize',30);
hold on;
hline = refline;
hline.Color = 'r';
hline.LineWidth = 2;
xlabel('average isi (ms)');
ylabel('C_V');
set(gcf,'color','w');
set(gca,'FontSize',14)
title('Problem 2g: Coefficient of variation (C_V)');
saveas(f2g,'figs/prob2g.png');
```

Problem 3: Inhomogeneous Poisson process

changing spike rate

```
t = 0:0.01:1;
st = t.^2 * pi; % reaching angle of arm

r0 = 35;
rmax = 60;
smax = pi/2;

rateS = num2cell(r0 + (rmax - r0)*cos(st - smax),1)';
rateMax = max(cell2mat(rateS));

% (a) Spike trains
n = 100; % # of spike trains
T = 1;
rate = num2cell(poissrnd(rateMax*T,n,1));

% thinning
Tns_thin = cell(n,1);
for i=1:length(rate)
    tns = sort(rand(rate{i},1));
    tns_keep = [];
    for j=1:length(tns)
        U = unifrnd(0,1);
        rateCheck = r0 + (rmax - r0)*cos(((tns(j).^2)*pi) - smax);
        if U < rateCheck/rateMax % keep the spike
            tns_keep = [tns_keep; tns(j)];
        end
    end
    Tns_thin{i} = tns_keep;
end

% (a) Plot 5 random spike trains
rows = randi([1 size(Tns_thin,1)],5,1);
Tn_plot = Tns_thin(rows,:);
```

```
f3a = figure;
%f3a.Position = [100, 100, 1000, 400];
for ss=1:5
    x = Tn_plot{ss,1};
    y = ss*ones(1,length(x));
    plot(x,y,'k|','LineWidth',2);
    hold on
end
axis padded
axis off
set(gcf,'color','w');
xlim([0 1])
ylim([-2 8])
title('Problem 3a: Spike trains');
saveas(f3a,'figs/prob3a.png');

% (b) Spike histogram
binWidth = 20; % ms
bins =  [0:(binWidth)/1000:1]; % s
cnts = [];
for t = 1:length(Tns_thin)
    thisTrial = Tns_thin{t};
    [N,~] = histcounts(thisTrial,bins);
    cnts = [cnts; N];
end
mnCnts = mean(cnts,1)*(1000/binWidth);

f3b = figure;
f3b.Position = [100, 100, 1000, 600];
x = bins(1:end-1)*1000; % ms
bar(x,mnCnts,'FaceColor',[0.5 0.5 0.5])
hold on;
xx = x./1000;
yy = r0 + (rmax - r0)*cos(((xx.^2)*pi) - smax);
plot(x,yy,'g-','LineWidth',3);
set(gcf,'color','w')
axis padded
box off
xlabel('Time (ms)');
ylabel('FR (Hz)');
%ylim([0 75])
set(gca,'FontSize',14)

sgtitle('Problem 3b: Spike histogram','FontSize',20,'FontWeight','bold');
saveas(f3b,'figs/prob3b.png');

% (c) Count distribution
totCnts = cellfun(@length,Tns_thin);

f3c = figure;
f3c.Position = [100, 100, 1000, 600];
bw = 2;
h = histogram(totCnts, 'normalization', 'probability','BinWidth',bw,'FaceColor',[0.5 0.5 0.5]);
hold on
edges = h.BinEdges;
lambdahat = poissfit(totCnts);
y = poisspdf(floor(edges), lambdahat);
plot(edges,y.*bw,'r-','LineWidth',3);
%axis off
xlabel('Spike counts');
ylabel('Frequency (Normalized # of Trials)')
box off
set(gcf,'color','w');
set(gca,'FontSize',14)
sgtitle('Problem 3c: Count distribution','FontSize',20,'FontWeight','bold');
saveas(f3c,'figs/prob3c.png');

% (d) ISI distribution
isi = cellfun(@(t) diff(t), Tns_thin, 'uni', 0);
```

```
isi_cat = [];
for i=1:length(isi)
    isi_cat = [isi_cat; isi{i}];
end

f3d = figure;
f3d.Position = [100, 100, 1000, 600];
bw = 4;
h = histogram(isi_cat.*1000, 'normalization', 'probability','BinWidth',bw,'FaceColor',[0.5 0.5 0.5]);
hold on
edges = h.BinEdges;
%muhat = expfit(isi_cat.*1000);
y = exppdf(floor(edges),mean(isi_cat).*1000);
plot(edges,y.*bw,'r-','LineWidth',3);
xlabel('ISI (ms)');
ylabel('Frequency (normalized # of trials)');
xlim([0 180])
%axis padded
box off
set(gcf,'color','w');
set(gca,'FontSize',14)
sgtitle('Problem 3c: ISI distribution','FontSize',20,'FontWeight','bold');
saveas(f3d,'figs/prob3d.png');
```

Problem 4: Real neural data

trial --> (182 trials) x (8 reaching angles) single neuron

```
load ps2_data.mat


% (a) Spike trains
rows = randi([1 size(trial,1)],5,1);
trial_plot = trial(rows,:);
sublocs = {9,6,2,4,7,10,14,12};
f4a = figure;
f4a.Position = [100, 100, 1000, 600];
subplot(5,3,[5, 8, 11])
image( imread('f1.png') );
axis off
set(gcf,'color','w')
for p=1:size(trial,2)
    subplot(5,3,sublocs{p})
    for ss=1:5
        x = find(trial_plot(ss,p).spikes==1);
        y = ss*(trial_plot(ss,p).spikes(x));
        plot(x,y,'k|','LineWidth',2);
        hold on;
    end
    axis padded
    axis off
    set(gcf,'color','w');
    xlim([0 500])
    ylim([-5 10])
end
sgtitle('Problem 4a: Spike trains','FontSize',20,'FontWeight','bold');
saveas(f4a,'figs/prob4a.png');

% (b) Spike histogram
binWidth = 20; % ms
bins =  [0:binWidth:500]; % s
mnCnts = cell(1,8);
for a = 1:size(trial,2)
    spks = {trial(:,a).spikes}.';
    cnts = [];
    for t = 1:length(spks)
        thisTrial = spks{t};
        N = sum(reshape(thisTrial,25,20),2)';
```

```
        cnts = [cnts; N];
    end
    mnCnts{a} = mean(cnts,1)*(1000/binWidth);
end

sublocs = {9,6,2,4,7,10,14,12};
f4b = figure;
f4b.Position = [100, 100, 1000, 600];
subplot(5,3,[5, 8, 11])
image( imread('f1.png') );
axis off
set(gcf,'color','w')
for p=1:8
    subplot(5,3,sublocs{p})
    x = bins(1:end-1); % ms
    bar(x,mnCnts{p},'FaceColor',[0.5 0.5 0.5])
    axis padded
    set(gcf,'color','w');
    xlabel('Time (ms)');
    ylabel('FR (Hz)');
    ylim([0 20])
    set(gca,'FontSize',14)
    box off
end
sgtitle('Problem 2b: Spike histogram','FontSize',20,'FontWeight','bold');
saveas(f4b,'figs/prob4b.png');

% (c) Tuning curve
totCnts = cell(size(trial));
for a = 1:size(trial,2)
    spks = {trial(:,a).spikes}.';
    for t = 1:length(spks)
        thisTrial = spks{t};
        N = sum(thisTrial);
        totCnts{t,a} = N;
    end
end
totCnts = cell2mat(totCnts);

r0 = 35;
rmax = 60;
smax = pi/2;

s = [30 70 110 150 190 230 310 350]; % reaching angle of arm
srad = s*(pi/180);
tunCurve = num2cell(r0 + (rmax - r0)*cos(srad - smax),1)';
mnTotCnts = mean(totCnts,1);

% linear least squares fit
x = s';
y = mnTotCnts';

yu = max(y); yl = min(y);
yr = (yu-yl); yz = y-yu+(yr/2);
zs = x(yz(:) .* circshift(yz(:),[1 0]) <= 0);     % Find zero-crossings
per = 2*mean(diff(zs));                           % Estimate period
ym = mean(y);                                     % Estimate offset
fit = @(b,x)  b(1) + (b(2) - b(1))*cos((x)*(pi/180) - b(3));
fcn = @(b) sum((fit(b,x) - y).^2);                              % Least-Squares cost function
ss = fminsearch(fcn, [yr;  per;  -1;  ym]);                      % Minimise Least-Squares
xp = linspace(min(x),max(x));

f4c = figure;
f4c.Position = [100, 100, 800, 400];
for tt=1:size(totCnts,1)
    plot(s,totCnts(tt,:),'k.','MarkerSize',10)
    hold on
end
xticks(s)
```

```
box off
plot(s,mnTotCnts,'r.','MarkerSize',30)
plot(xp,fit(ss,xp),'g-','LineWidth',2);
xlabel('s (movement in direction in degrees)');
ylabel('FR (Hz)');
xlim([0 315]);
set(gcf,'color','w');
set(gca,'FontSize',14)
title('Problem 4c: Tuning curve');
saveas(f4c,'figs/prob4c.png');

% (d) Count distribution
sublocs = {9,6,2,4,7,10,14,12};
f4d = figure;
f4d.Position = [100, 100, 1000, 600];
subplot(5,3,[5, 8, 11])
image( imread('f1.png') );
axis off
set(gcf,'color','w')
for p=1:8
    subplot(5,3,sublocs{p})
    bw = 1;
    h = histogram(totCnts(:,p),'Normalization','probability','BinWidth',bw,'FaceColor',[0.5 0.5 0.5]);
    hold on
    edges = h.BinEdges;
    y = poisspdf(floor(edges), mean(totCnts(:,p)));
    plot(edges,y.*bw,'r-','LineWidth',3);
    box off
    set(gcf,'color','w');
    xlabel('Spike counts');
    ylabel('# trials');
    xlim([0 12]);
    set(gca,'FontSize',14)
end
sgtitle('Problem 4d: Count distribution','FontSize',20,'FontWeight','bold');
saveas(f4d,'figs/prob4d.png');

% (e) Fano factor
mnTotCnts = mean(totCnts,1);
varTotCnts = var(totCnts);

f4e = figure;
plot(mnTotCnts,varTotCnts,'k.','MarkerSize',20);
hold on;
hline = refline(1,0);
hline.Color = 'r';
hline.LineWidth = 2;
lsline
axis square
xlim([1 7])
ylim([1 7])
xlabel('mean (spikes)');
ylabel('variance (spikes^2)');
set(gcf,'color','w');
set(gca,'FontSize',14)
title('Problem 4e: Fano factor');
saveas(f4e,'figs/prob4e.png');

% (f) ISI distribution
isiAll = cell(1,8);
for a = 1:size(trial,2)
    spks = {trial(:,a).spikes}.';
    tns = cellfun(@(x) diff(find(x==1)), spks,'uni',0);
    isi = [];
    for t = 1:length(tns)
        thisTrial = tns{t};
        if ~isempty(thisTrial)
            isi = [isi; thisTrial'];
        end
```

```
    end
    isiAll{a} = isi;
end


sublocs = {9,6,2,4,7,10,14,12};
f4f = figure;
f4f.Position = [100, 100, 1000, 600];
subplot(5,3,[5, 8, 11])
image( imread('f1.png') );
axis off
set(gcf,'color','w')
for p=1:8
    subplot(5,3,sublocs{p})
    bw = 10; %ms
    h = histogram(isiAll{p},'Normalization','probability','BinWidth',bw,'FaceColor',[0.5 0.5 0.5]);
    hold on;
    edges = h.BinEdges;
    y = exppdf(floor(edges),mean(isiAll{p}));
    plot(edges,y.*bw,'r-','LineWidth',3);
    box off
    set(gcf,'color','w');
    xlabel('ISI (ms)');
    ylabel('f');
    xlim([0 400]);
    ylim([0 0.2]);
    set(gca,'FontSize',14)
end
sgtitle('Problem 4f: ISI distribution','FontSize',20,'FontWeight','bold');
saveas(f4f,'figs/prob4f.png');
```

Published with MATLAB® R2021a