
```
% Tutorial 3.1
% Email: jlindbloom@smu.edu

clear all
close all

% Define parameters.
global e_leak r_membrane c_membrane v_threshold v_reset e_k
delta_g_sra tau_sra g_leak a delta_th b v_max
e_leak = -70e-3;
r_membrane = 100e6;
c_membrane = 100e-12;
g_leak = 10e-9;
v_threshold = -50e-3;
v_reset = -80e-3;
v_max = 10e-3;
e_k = -80e-3;
delta_g_sra = 1e-9;
tau_sra = 150e-3;
a = 2e-9;
delta_th = 2e-3;
b = 0;

% Setup time vector.

global dt
dt = 0.01e-3;
t = 0:dt:100;

sigma = 50e-12;
I_app = normrnd(0, sigma/(sqrt(dt)), 1, length(t));

i_sra = zeros(1, length(t));
v = zeros(1, length(t));
v(1) = e_leak;

[v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app);

spikes = find(spikes);
inter_spikes = zeros(1, length(spikes)-1);
for n = 1:length(inter_spikes)
    inter_spikes(n) = spikes(n+1) - spikes(n);
end
inter_spikes = dt*inter_spikes;

f1 = figure;
figure(f1);
histogram(inter_spikes, 25)
title("ISI Histogram (1a)");
xlabel("Time (seconds)");
ylabel("Count");
```

```

cov = std(inter_spikes)/mean(inter_spikes);
fprintf("1.a. The Coefficient of Variation is %f\n", cov);

nspikes = zeros(1, 1000);
tspikes = dt*spikes;

for n = 1:length(tspikes)
    temp = fix(tspikes(n)/(100e-3));
    nspikes(temp+1) = nspikes(temp+1) + 1;
end

fano = ((std(nspikes))^2)/(mean(nspikes));
fprintf("1.a. The Fano Factor is %f\n", fano);

windows = 10e-3: 10e-3:1;
varfano = zeros(1,length(windows));

for n = 1:length(windows)
    nspikes = zeros(1, fix(100/windows(n)));
    for m = 1:length(tspikes)
        temp = fix(tspikes(m)/(windows(n)));
        try
            nspikes(temp+1) = nspikes(temp+1) + 1;
        end
    end
    fano = ((std(nspikes))^2)/(mean(nspikes));
    varfano(n) = fano;
end

f2 = figure;
figure(f2);
scatter(windows, varfano)
title("Fano Factor vs. Window Size (1a)");
xlabel("Window Size (ms)");
ylabel("Fano Factor");

b = 1e-9;
[v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app);

spikes = find(spikes);
inter_spikes = zeros(1, length(spikes)-1);
for n = 1:length(inter_spikes)
    inter_spikes(n) = spikes(n+1) - spikes(n);
end
inter_spikes = dt*inter_spikes;

f3 = figure;
figure(f3);
histogram(inter_spikes, 25)
title("ISI Histogram (1b)");
xlabel("Time (seconds)");
ylabel("Count");

```

```

cov = std(inter_spikes)/mean(inter_spikes);
fprintf("1.b. The Coefficient of Variation is %f\n", cov);

nspikes = zeros(1, 1000);
tspikes = dt*spikes;

for n = 1:length(tspikes)
    temp = fix(tspikes(n)/(100e-3));
    nspikes(temp+1) = nspikes(temp+1) + 1;
end

fano = ((std(nspikes))^2)/(mean(nspikes));
fprintf("1.b. The Fano Factor is %f\n", fano);

windows = 10e-3: 10e-3:1;
varfano = zeros(1,length(windows));

for n = 1:length(windows)
    nspikes = zeros(1, fix(100/windows(n)));
    for m = 1:length(tspikes)
        temp = fix(tspikes(m)/(windows(n)));
        try
            nspikes(temp+1) = nspikes(temp+1) + 1;
        end
    end
    fano = ((std(nspikes))^2)/(mean(nspikes));
    varfano(n) = fano;
end

f4 = figure;
figure(f4);
scatter(windows, varfano)
title("Fano Factor vs. Window Size (1b)");
xlabel("Window Size (ms)");
ylabel("Fano Factor");

% 1.c.

% 0 nA.

b = 0;
sigma = 20e-12;
I_app = normrnd(0, sigma/(sqrt(dt)), 1, length(t));

[v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app);

spikes = find(spikes);
inter_spikes = zeros(1, length(spikes)-1);
for n = 1:length(inter_spikes)
    inter_spikes(n) = spikes(n+1) - spikes(n);
end
inter_spikes = dt*inter_spikes;

```

```

f5 = figure;
figure(f5);
histogram(inter_spikes, 25)
title("ISI Histogram (1c, +0 nA)");
xlabel("Time (seconds)");
ylabel("Count");

cov = std(inter_spikes)/mean(inter_spikes);
fprintf("1.c. (+0 nA) The Coefficient of Variation is %f\n", cov);

nspikes = zeros(1, 1000);
tspikes = dt*spikes;

for n = 1:length(tspikes)
    temp = fix(tspikes(n)/(100e-3));
    nspikes(temp+1) = nspikes(temp+1) + 1;
end

fano = ((std(nspikes))^2)/(mean(nspikes));
fprintf("1.c. (+0 nA) The Fano Factor is %f\n", fano);

% 0.1 nA.

% 0 nA.

b = 0;
sigma = 20e-12;
I_app = normrnd(0, sigma/(sqrt(dt)), 1, length(t)) + 0.1e-9;

[v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app);

spikes = find(spikes);
inter_spikes = zeros(1, length(spikes)-1);
for n = 1:length(inter_spikes)
    inter_spikes(n) = spikes(n+1) - spikes(n);
end
inter_spikes = dt*inter_spikes;

f6 = figure;
figure(f6);
histogram(inter_spikes, 25)
title("ISI Histogram (1c, +0.1 nA)");
xlabel("Time (seconds)");
ylabel("Count");

cov = std(inter_spikes)/mean(inter_spikes);
fprintf("1.c. (+0.1 nA) The Coefficient of Variation is %f\n", cov);

nspikes = zeros(1, 1000);
tspikes = dt*spikes;

for n = 1:length(tspikes)
    temp = fix(tspikes(n)/(100e-3));
    nspikes(temp+1) = nspikes(temp+1) + 1;

```

```

end

fano = ((std(nspikes))^2)/(mean(nspikes));
fprintf("1.c. (+0.1 nA) The Fano Factor is %f\n", fano);

% 0.2 nA.

% 0 nA.

b = 0;
sigma = 20e-12;
I_app = normrnd(0, sigma/(sqrt(dt)), 1, length(t)) + 0.2e-9;

[v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app);

spikes = find(spikes);
inter_spikes = zeros(1, length(spikes)-1);
for n = 1:length(inter_spikes)
    inter_spikes(n) = spikes(n+1) - spikes(n);
end
inter_spikes = dt*inter_spikes;

f7 = figure;
figure(f7);
histogram(inter_spikes, 25)
title("ISI Histogram (1c, +0.2 nA)");
xlabel("Time (seconds)");
ylabel("Count");

cov = std(inter_spikes)/mean(inter_spikes);
fprintf("1.c. (+0.2 nA) The Coefficient of Variation is %f\n", cov);

nspikes = zeros(1, 1000);
tspikes = dt*spikes;

for n = 1:length(tspikes)
    temp = fix(tspikes(n)/(100e-3));
    nspikes(temp+1) = nspikes(temp+1) + 1;
end

fano = ((std(nspikes))^2)/(mean(nspikes));
fprintf("1.c. (+0.2 nA) The Fano Factor is %f\n", fano);

%{
Comments:

(1.b)

```

When $b=0$, the ISI histogram resembles the canonical distribution, or at least a distribution with decaying probabilities for larger inter-spike intervals - when $b=1e-9$, the ISI hitogram is very different and instead resembles a normal

distribution centered at an inter-spike interval of 180 milliseconds.

The plots for Fano Factor vs. Window Size are also very different.

When $b=0$, it is hard to discern any trend since the results are noisy and very across repeated trials. On the other hand, when $b=1e-0$ there is a clear relationship that as the window size increases, the fano factor decreases and seems to approach a horizontal asymptote. This trend occurs because when the window size is very small, the number of spikes in windows is either 1 or 0, and since this vector is fairly sparse the mean is very close to the variance resulting in a fano factor close to 1.

(1.c)

Increasing the constant input current term skews the ISI distribution slightly towards an increased inter-spike interval, and also greatly increases the spike counts in increased bins - this makes sense, since firing rate increases with input current, leading to more spikes. Additionally, both the coefficient of variation and the fano factor decreased when the constant input current term was increased.

```
%}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function Definitions:

function [v_simulated, spike_vec, i_sra_simulated] = aelifsim(v,
    i_sra, t, i_applied)
% Simulates the membrane potential with given inputs.
global v_threshold v_reset dt b v_max c_membrane delta_th e_leak
    g_leak a tau_sra
v_simulated = zeros(1, length(v));
v_simulated(1) = v(1);
spike_vec = zeros(1, length(v));
i_sra_simulated = zeros(1, length(i_sra));
i_sra_simulated(1) = i_sra(1);
temp = length(v)-1;
for n = 1:temp
    if ( v_simulated(n) > v_max )
        v_simulated(n) = v_reset;
        i_sra_simulated(n) = i_sra_simulated(n) + b;
        spike_vec(n) = 1;
    end

    step1 = dt*( g_leak*(e_leak-v_simulated(n) +
        delta_th*exp((v_simulated(n)-v_threshold)/delta_th) ) ...
        - i_sra_simulated(n) + i_applied(n))/c_membrane;

    step2 = dt*( a*(v_simulated(n) - e_leak) - i_sra_simulated(n) )/
tau_sra;
```

```

        v_simulated(n+1) = v_simulated(n) + step1;
        i_sra_simulated(n+1) = i_sra_simulated(n) + step2;

    end

end

function [result] = expandbin(initial_vector, initial_width,
    final_width)
% Downsamples the input vector.
old_length = length(initial_vector);
ratio = round(final_width/initial_width);

new_length = ceil(old_length/ratio);
result = zeros(1, new_length);

for n = 1:new_length-1
    result(n) = mean(initial_vector( (n-1)*ratio+1 : n*ratio ) );
end

result(new_length) = mean(initial_vector( (new_length-1)*ratio :
    new_length ) );
end

function [sta, tcorr] = STA(Iapp, spikes, dt, tminus, tplus)
% Computes the spike-triggered average.

if (~exist('tminus'))
    tminus = 75e-3;
end
if (~exist('tplus'))
    tminus = 25e-3;
end

% Set nminus and nplus.
nminus = ceil(tminus/dt);
nplus = ceil(tplus/dt);

% Define tcorr.
tcorr = -dt*nminus:dt:dt*nplus;

% Setup sta vector.
sta = zeros(1,nminus+nplus+1);

% Find spikes.
spike_indices = find(spikes);
nspikes = length(spikes);

% Loop over spike vector.
for n=1:length(spike_indices)

    index = spike_indices(n);

```

```

    % Check that indices never step outside of the bounds.

    if (index-nminus) < 1
        left = 1;
    else
        left = index-nminus;
    end

    if (index+nplus) > length(Iapp)
        right = length(spike_indices);
    else
        right = index+nplus;
    end

    % Add Iapp to sta vector.
    for i=left:right
        sta(i-index+nminus+1) = sta(i-index+nminus+1) + Iapp(i)/
nspikes;
    end
end

% Divide sta vector by the number of spikes.
sta = sta/nspikes;

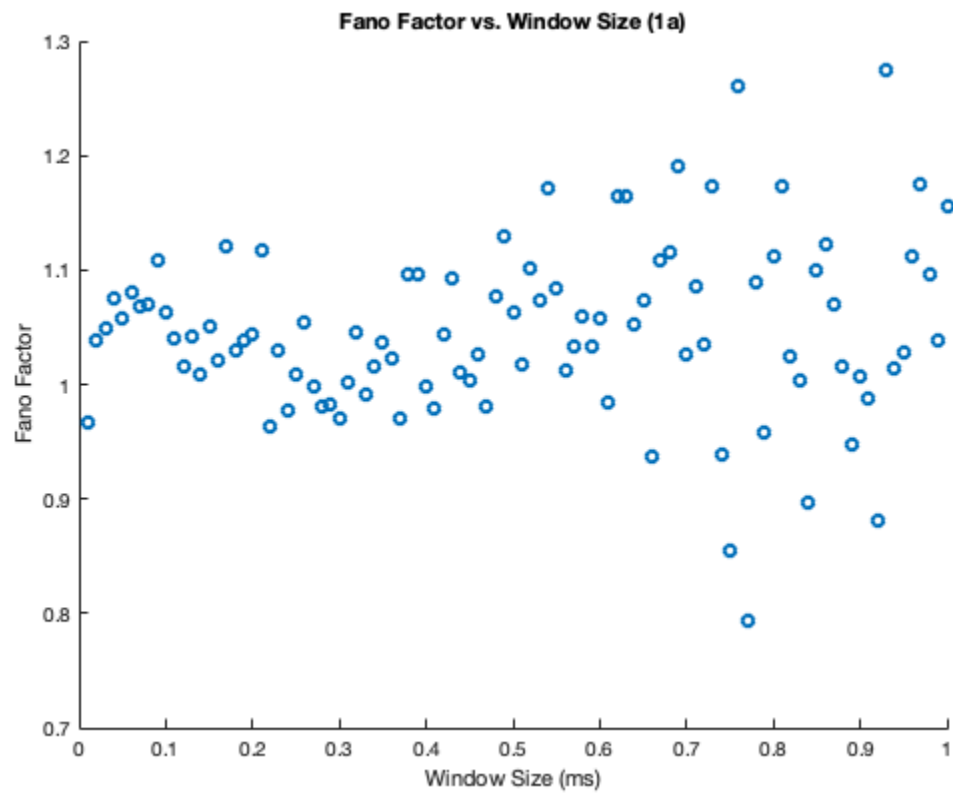
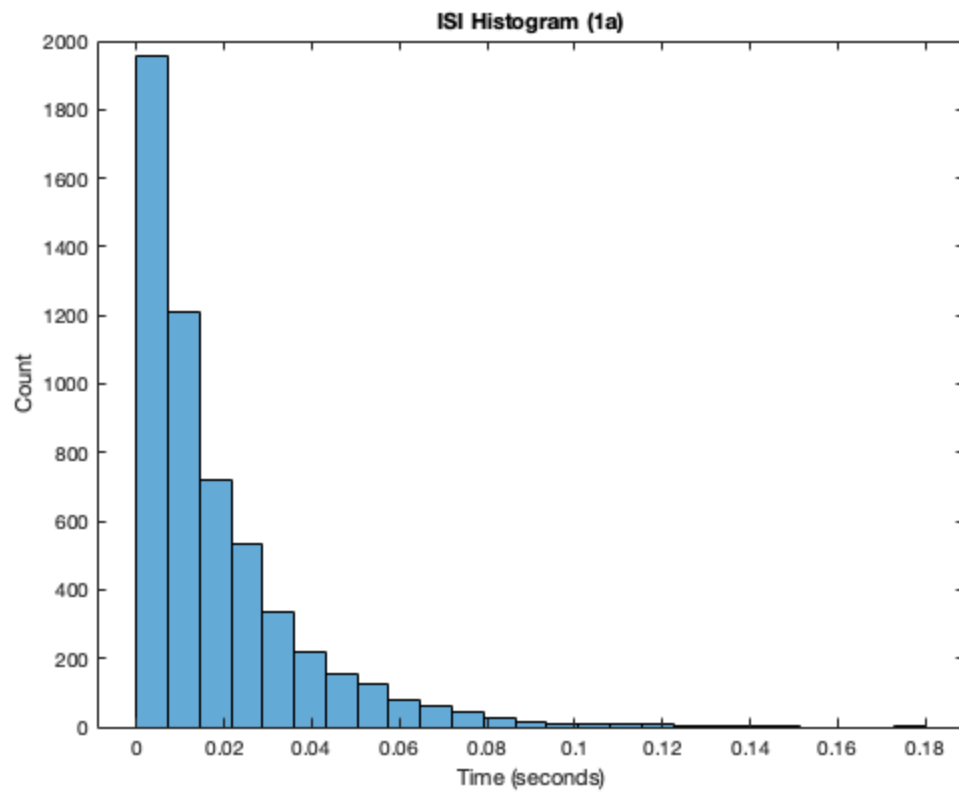
end

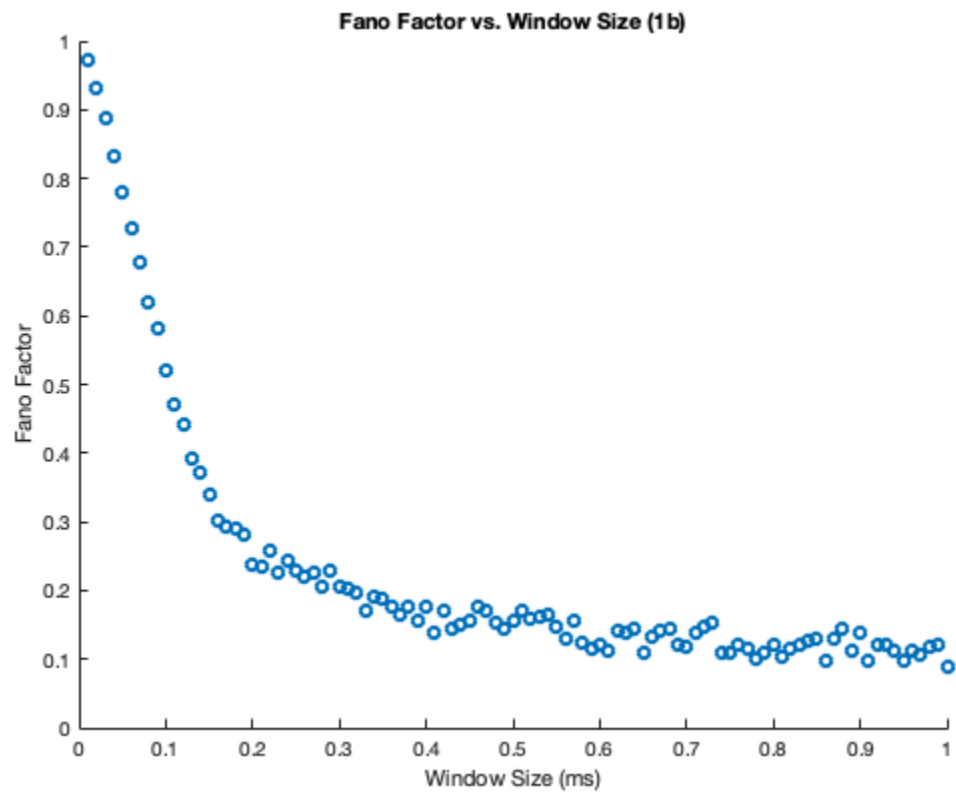
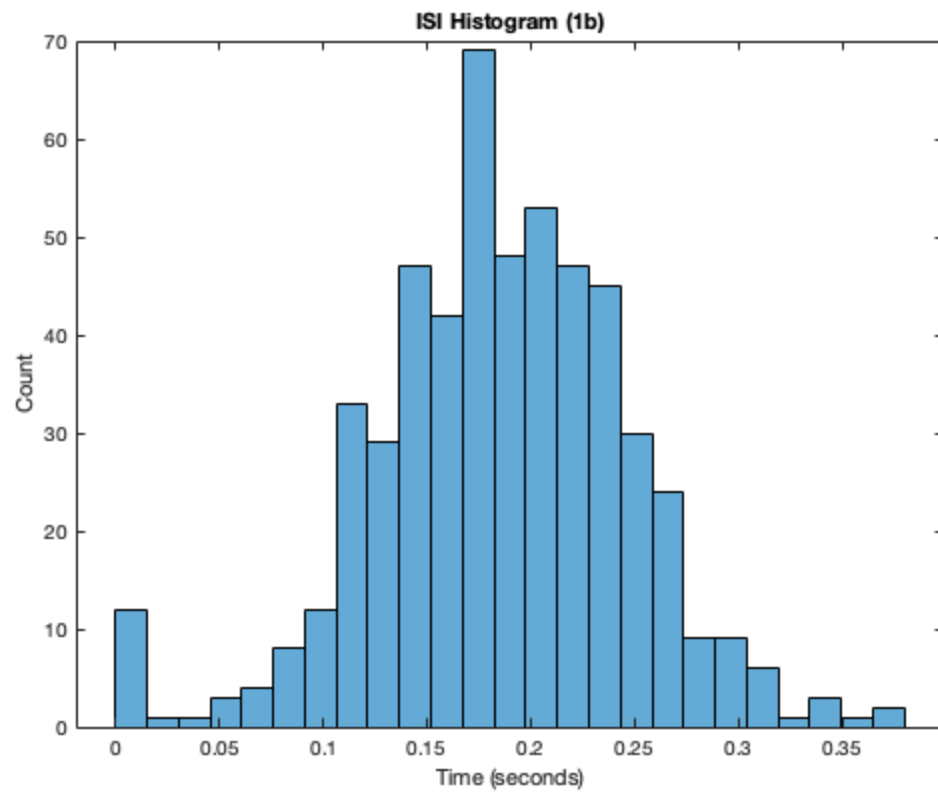
```

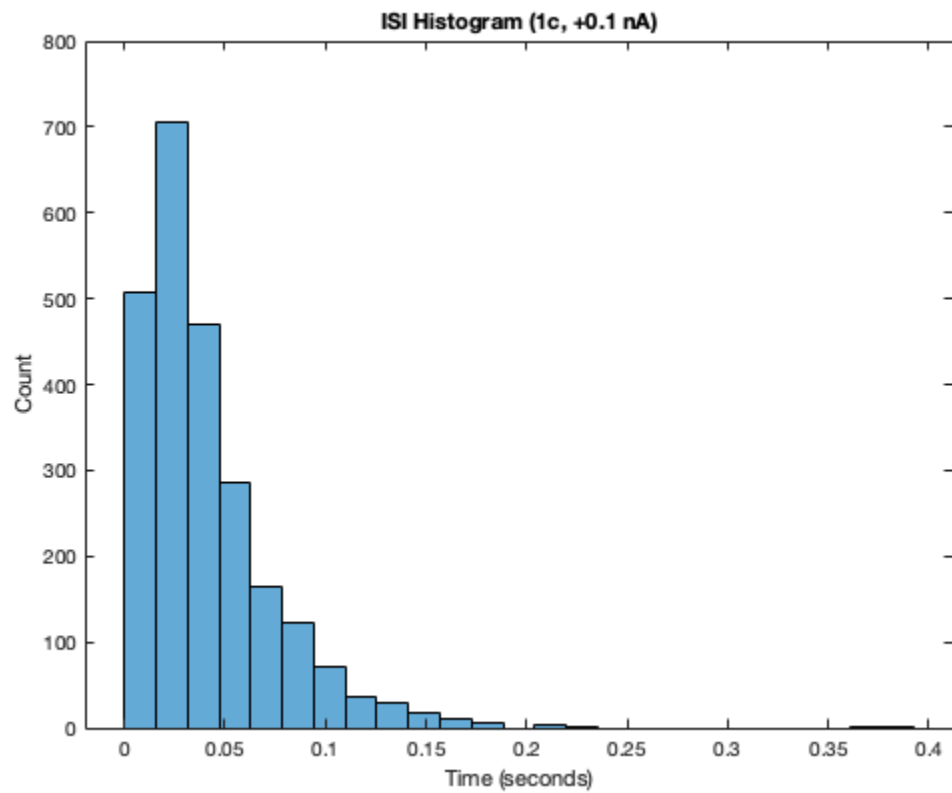
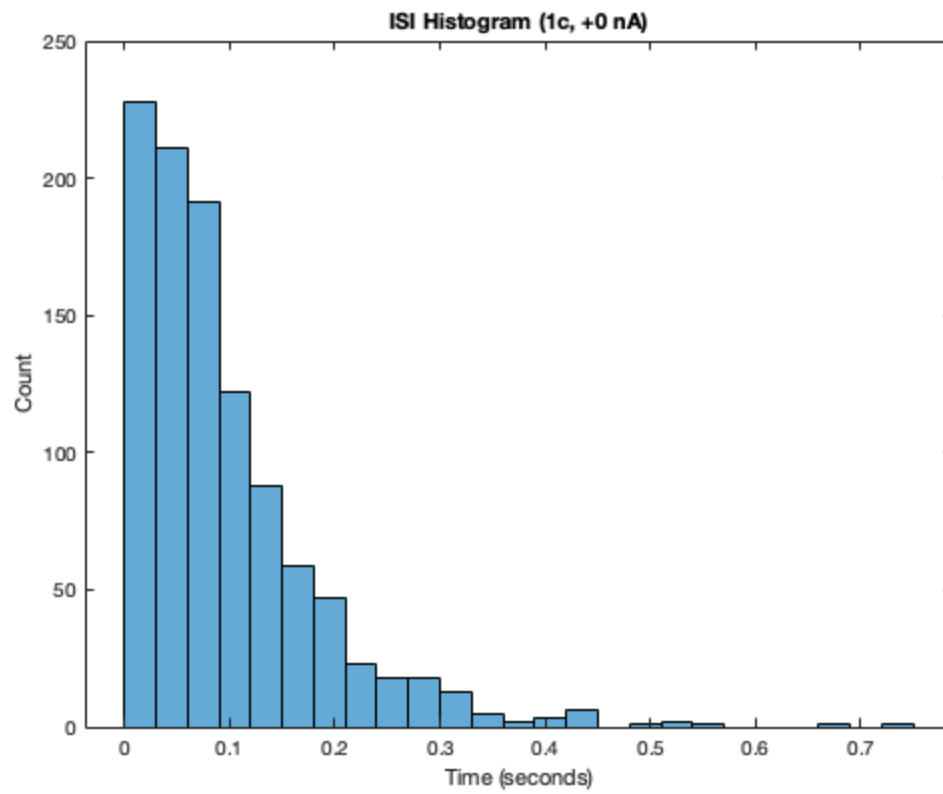
```

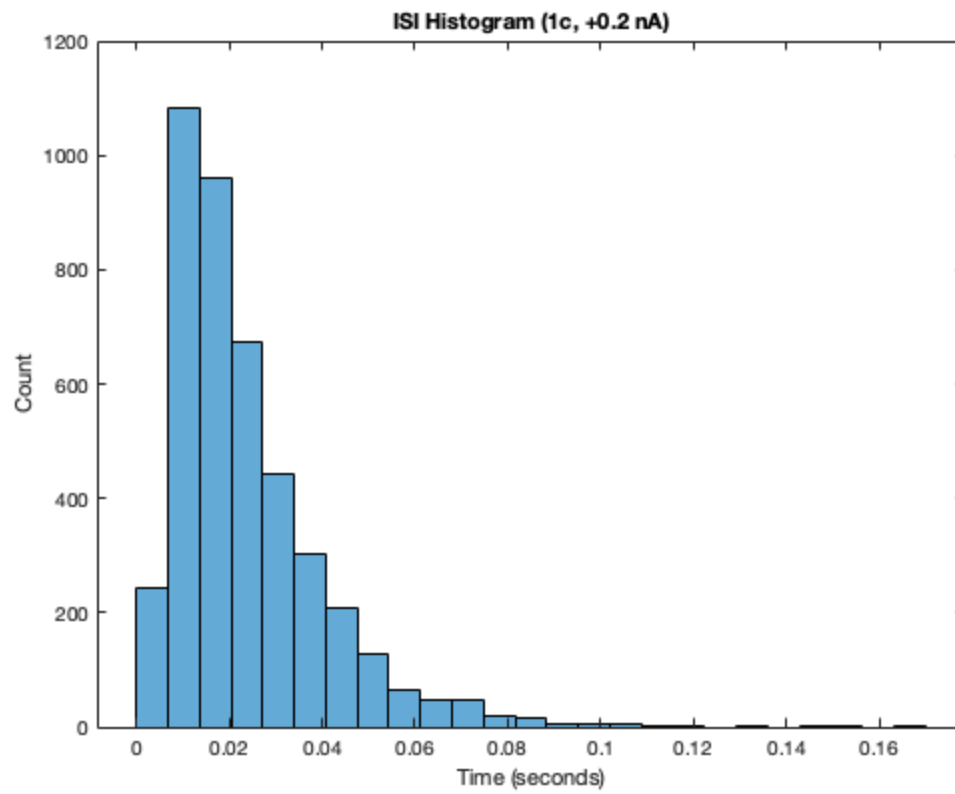
1.a. The Coefficient of Variation is 1.046675
1.a. The Fano Factor is 1.062127
1.b. The Coefficient of Variation is 0.332571
1.b. The Fano Factor is 0.519779
1.c. (+0 nA) The Coefficient of Variation is 0.915220
1.c. (+0 nA) The Fano Factor is 0.832070
1.c. (+0.1 nA) The Coefficient of Variation is 0.815558
1.c. (+0.1 nA) The Fano Factor is 0.674538
1.c. (+0.2 nA) The Coefficient of Variation is 0.693589
1.c. (+0.2 nA) The Fano Factor is 0.510963

```







Published with MATLAB® R2018a