
```
% Tutorial 3.3
% 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:0.5;

sigma = 20e-12;
numtrials = 1000;

no_stim_sims_spikes = zeros(1, numtrials);
stim_sims_spikes = zeros(1, numtrials);

% Run simulation and get spike-number data.
for n=1:1000
    i_sra = zeros(1, length(t));
    v = zeros(1, length(t));
    v(1) = e_leak;
    I_app_no_stim = normrnd(0, sigma/(sqrt(dt)), 1, length(t));
    I_app_with_stim = normrnd(0, sigma/(sqrt(dt)), 1, length(t)) +
0.1e-9;
    [v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app_no_stim);
    no_stim_sims_spikes(n) = sum(find(spikes));
    [v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t,
I_app_with_stim);
    stim_sims_spikes(n) = sum(find(spikes));
end

% Process data.
no_stim_mean_firing_rate = mean(no_stim_sims_spikes/0.5);
```

```

with_stim_mean_firing_rate = mean(stim_sims_spikes/0.5);

fprintf("Without stimulus, the mean firing rate over 0.5 seconds is %f
spikes/second. \n",...
    no_stim_mean_firing_rate );
fprintf("With stimulus, the mean firing rate over 0.5 seconds is %f
spikes/second. \n",...
    with_stim_mean_firing_rate );

nmax = max(max(stim_sims_spikes), max(no_stim_sims_spikes));
edges = [0:10000:nmax];
[no_stim_hist, edges] = histcounts(no_stim_sims_spikes, edges);
[with_stim_hist, edges] = histcounts(stim_sims_spikes, edges);

% Plot # of observations vs. spike count.
f1 = figure;
figure(f1);
stairs(no_stim_hist);
hold on
stairs(with_stim_hist);
title("# of Observations vs. Spike Count (0.5 s)");
xlabel("Spike Count");
ylabel("# of Observations");
legend("No Stimulus", "With Stimulus");

% Plot probability greater vs. spike count.
f2 = figure;
figure(f2);
stairs(cumsum(no_stim_hist, 'reverse')/numtrials);
hold on
stairs(cumsum(with_stim_hist, 'reverse')/numtrials);
title("Probability Greater vs. Spike Count (0.5 s)");
xlabel("Spike Count");
ylabel("Probability Greater");
legend("No Stimulus", "With Stimulus");

% Plot ROC curve.
f3 = figure;
figure(f3);
spikenums = [0:nmax];
truepositives = zeros(1, length(spikenums));
falsepositives = zeros(1, length(spikenums));
for n=1:length(truepositives)
    temp = stim_sims_spikes > spikenums(n);
    truepositives(n) = sum(temp)/numtrials;
    temp = no_stim_sims_spikes > spikenums(n);
    falsepositives(n) = sum(temp)/numtrials;
end
plot(falsepositives, truepositives);
title("ROC Curve (0.5 s)");
xlabel("Probability of False Positive");
ylabel("Probability of True Positive");

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Repeat all of the above,
% except with a duration of
% 0.2 seconds.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Setup time vector.

global dt
dt = 0.01e-3;
t = 0:dt:0.2;

sigma = 20e-12;
numtrials = 1000;

no_stim_sims_spikes = zeros(1, numtrials);
stim_sims_spikes = zeros(1, numtrials);

% Run simulation and get spike-number data.
for n=1:1000
    i_sra = zeros(1, length(t));
    v = zeros(1, length(t));
    v(1) = e_leak;
    I_app_no_stim = normrnd(0, sigma/(sqrt(dt)), 1, length(t));
    I_app_with_stim = normrnd(0, sigma/(sqrt(dt)), 1, length(t)) +
    0.1e-9;
    [v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app_no_stim);
    no_stim_sims_spikes(n) = sum(find(spikes));
    [v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t,
    I_app_with_stim);
    stim_sims_spikes(n) = sum(find(spikes));
end

% Process data.
no_stim_mean_firing_rate = mean(no_stim_sims_spikes/0.5);
with_stim_mean_firing_rate = mean(stim_sims_spikes/0.5);

fprintf("Without stimulus, the mean firing rate over 0.2 seconds is %f
spikes/second. \n",...
    no_stim_mean_firing_rate );
fprintf("With stimulus, the mean firing rate over 0.2 seconds is %f
spikes/second. \n",...
    with_stim_mean_firing_rate );

nmax = max(max(stim_sims_spikes), max(no_stim_sims_spikes));
edges = [0:2000:nmax];
[no_stim_hist, edges] = histcounts(no_stim_sims_spikes, edges);
[with_stim_hist, edges] = histcounts(stim_sims_spikes, edges);

% Plot # of observations vs. spike count.
f4 = figure;
figure(f4);
stairs(no_stim_hist);
hold on

```

```

stairs(with_stim_hist);
title("# of Observations vs. Spike Count (0.2 s)");
xlabel("Spike Count");
ylabel("# of Observations");
legend("No Stimulus", "With Stimulus");

% Plot probability greater vs. spike count.
f5 = figure;
figure(f5);
stairs(cumsum(no_stim_hist, 'reverse')/numtrials);
hold on
stairs(cumsum(with_stim_hist, 'reverse')/numtrials);
title("Probability Greater vs. Spike Count (0.2 s)");
xlabel("Spike Count");
ylabel("Probability Greater");
legend("No Stimulus", "With Stimulus");

% Plot ROC curve.
f6 = figure;
figure(f6);
nmax = max(max(stim_sims_spikes), max(no_stim_sims_spikes));
spikenums = [0:nmax];
truepositives = zeros(1, length(spikenums));
falsepositives = zeros(1, length(spikenums));
for n=1:length(truepositives)
    temp = stim_sims_spikes > spikenums(n);
    truepositives(n) = sum(temp)/numtrials;
    temp = no_stim_sims_spikes > spikenums(n);
    falsepositives(n) = sum(temp)/numtrials;
end
plot(falsepositives, truepositives);
title("ROC Curve (0.2 s)");
xlabel("Probability of False Positive");
ylabel("Probability of True Positive");

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Repeat all of the above,
% except with specified
% modification.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Setup time vector.

global dt
dt = 0.01e-3;
t = 0:dt:0.5;

sigma1 = 20e-12;
sigma2 = 5e-12;
numtrials = 1000;

no_stim_sims_spikes = zeros(1, numtrials);
stim_sims_spikes = zeros(1, numtrials);

```

```

% Run simulation and get spike-number data.
for n=1:1000
    i_sra = zeros(1, length(t));
    v = zeros(1, length(t));
    v(1) = e_leak;
    I_app_no_stim = normrnd(0, sigma1/(sqrt(dt)), 1, length(t));
    I_app_with_stim = normrnd(0, sigma2/(sqrt(dt)), 1, length(t)) +
    0.5e-9;
    [v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t, I_app_no_stim);
    no_stim_sims_spikes(n) = sum(find(spikes));
    [v_sim, spikes, i_sra_sim] = aelifsim(v, i_sra, t,
    I_app_with_stim);
    stim_sims_spikes(n) = sum(find(spikes));
end

% Process data.
no_stim_mean_firing_rate = mean(no_stim_sims_spikes/0.5);
with_stim_mean_firing_rate = mean(stim_sims_spikes/0.5);

fprintf("Without stimulus, the mean firing rate over 0.5 seconds is %f
spikes/second. \n",...
    no_stim_mean_firing_rate );
fprintf("With stimulus, the mean firing rate over 0.5 seconds is %f
spikes/second. \n",...
    with_stim_mean_firing_rate );

nmax = max(max(stim_sims_spikes), max(no_stim_sims_spikes));
edges = [0:10000:nmax];
[no_stim_hist, edges] = histcounts(no_stim_sims_spikes, edges);
[with_stim_hist, edges] = histcounts(stim_sims_spikes, edges);

% Plot # of observations vs. spike count.
f7 = figure;
figure(f7);
stairs(no_stim_hist);
hold on
stairs(with_stim_hist);
title("# of Observations vs. Spike Count (0.5 s)");
xlabel("Spike Count");
ylabel("# of Observations");
legend("No Stimulus", "With Stimulus");

% Plot probability greater vs. spike count.
f8 = figure;
figure(f8);
stairs(cumsum(no_stim_hist, 'reverse')/numtrials);
hold on
stairs(cumsum(with_stim_hist, 'reverse')/numtrials);
title("Probability Greater vs. Spike Count (0.5 s)");
xlabel("Spike Count");
ylabel("Probability Greater");
legend("No Stimulus", "With Stimulus");

```

```

% Plot ROC curve.
f9 = figure;
figure(f9);
spikenums = [0:nmax];
truepositives = zeros(1, length(spikenums));
falsepositives = zeros(1, length(spikenums));
for n=1:length(truepositives)
    temp = stim_sims_spikes > spikenums(n);
    truepositives(n) = sum(temp)/numtrials;
    temp = no_stim_sims_spikes > spikenums(n);
    falsepositives(n) = sum(temp)/numtrials;
end
plot(falsepositives, truepositives);
title("ROC Curve (0.5 s)");
xlabel("Probability of False Positive");
ylabel("Probability of True Positive");

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 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
end

```

```

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

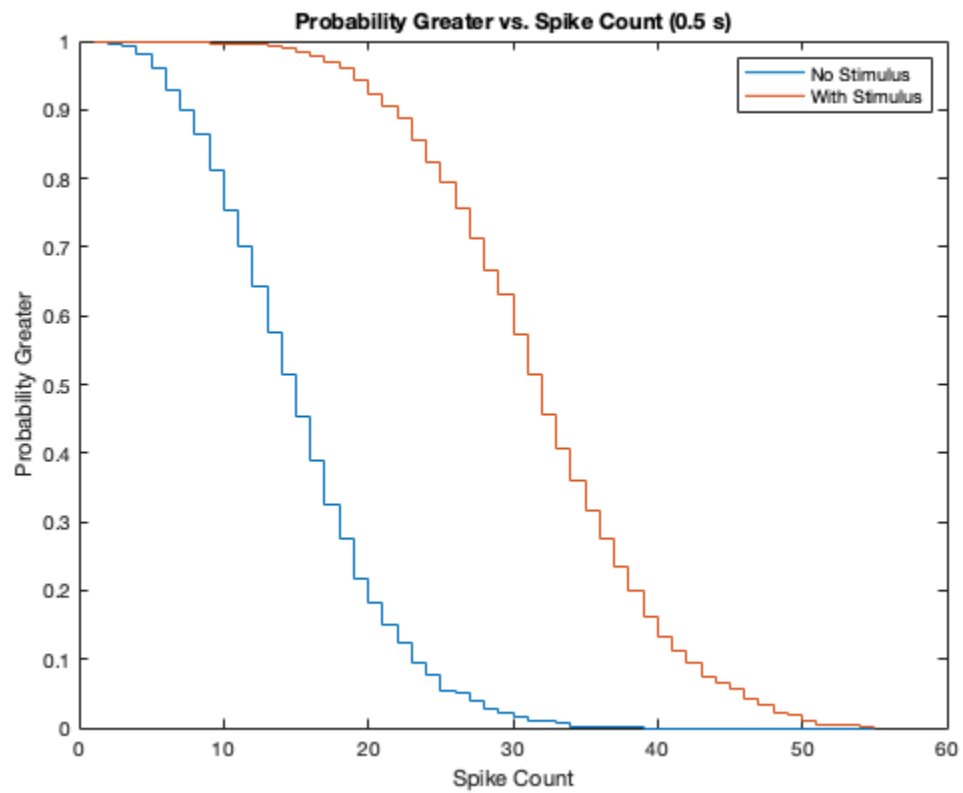
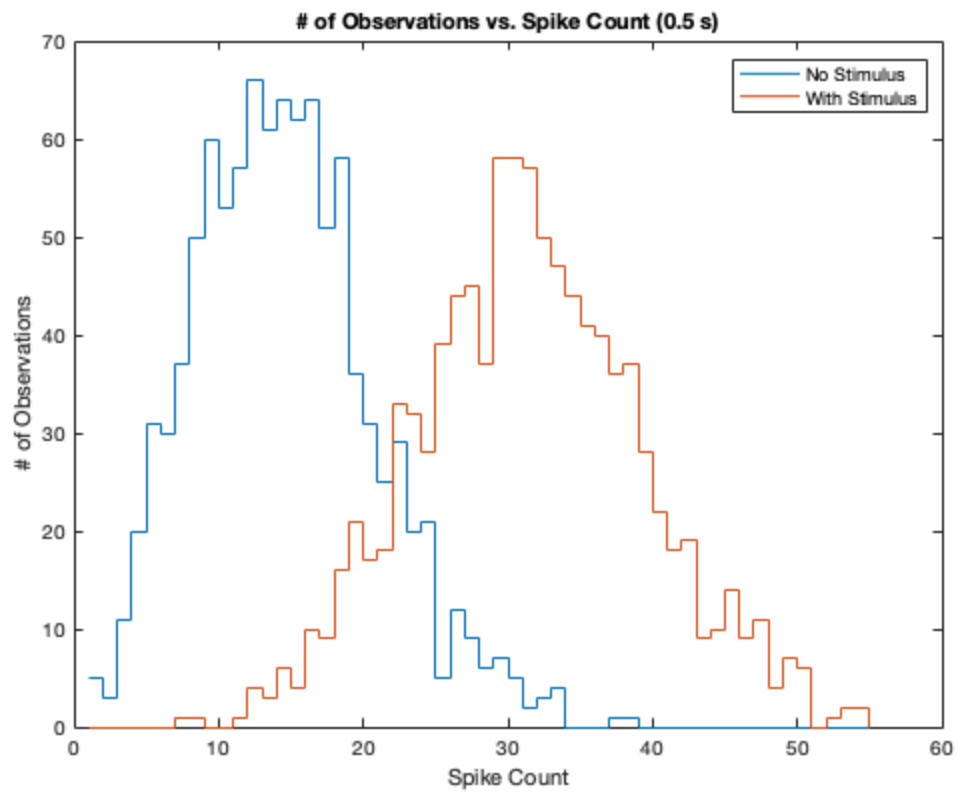
    % Add Iapp to sta vector.
    for i=left:right
        sta(i-index+nminus+1) = sta(i-index+nminus+1) + Iapp(i)/
nspikes;
    end
end

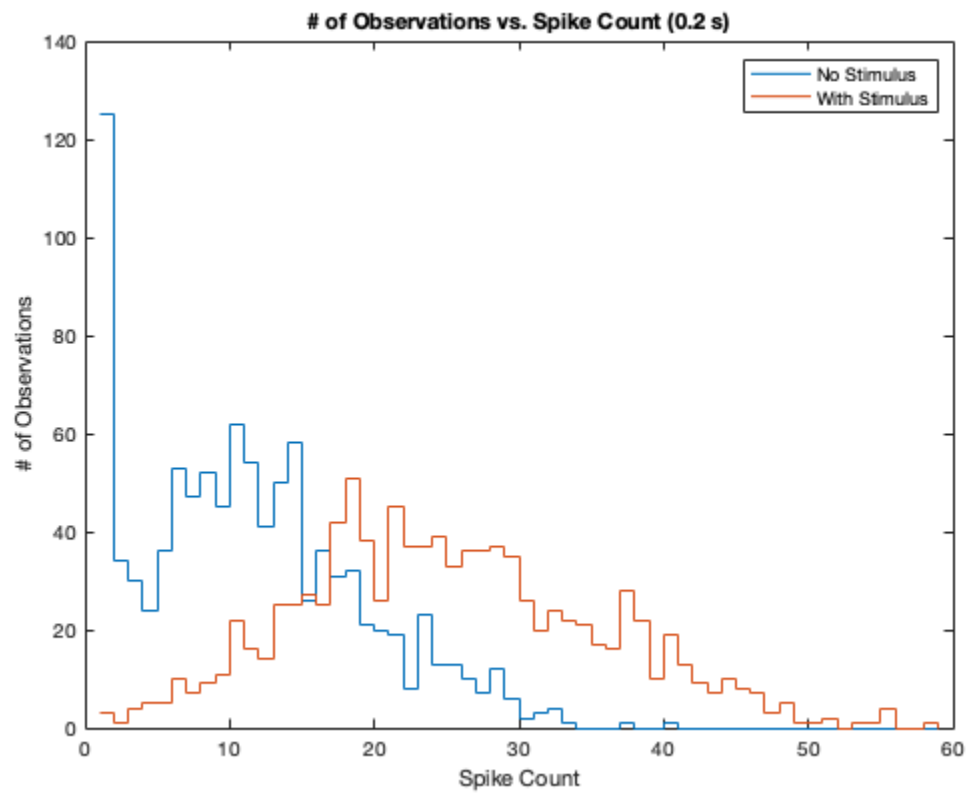
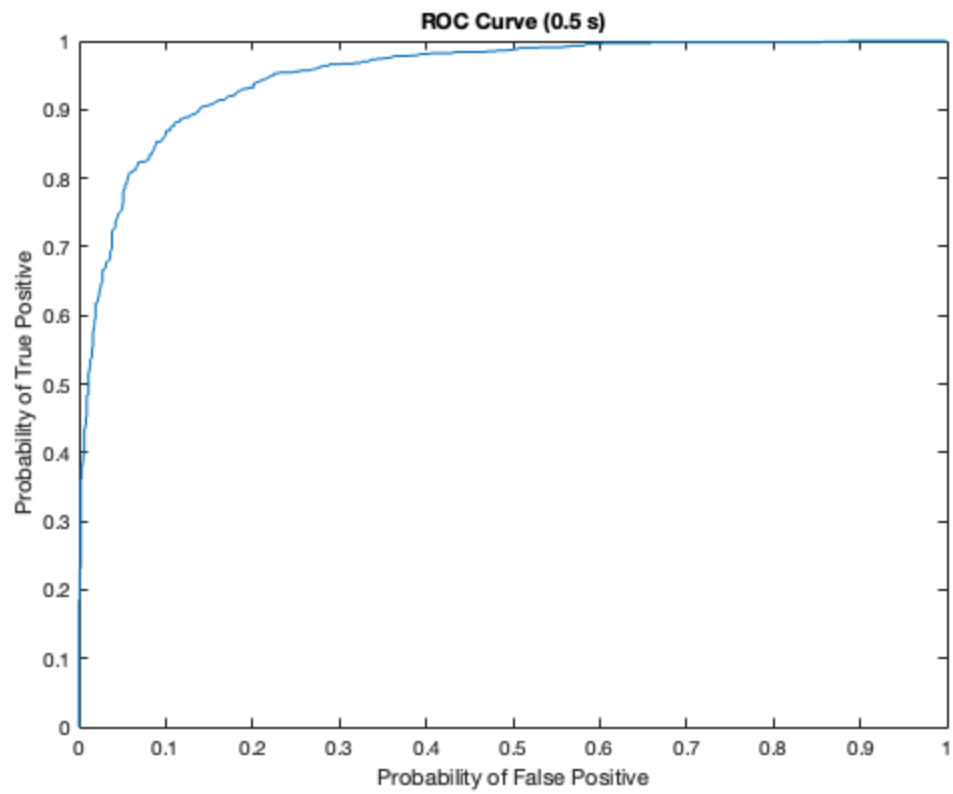
% Divide sta vector by the number of spikes.
sta = sta/nspikes;

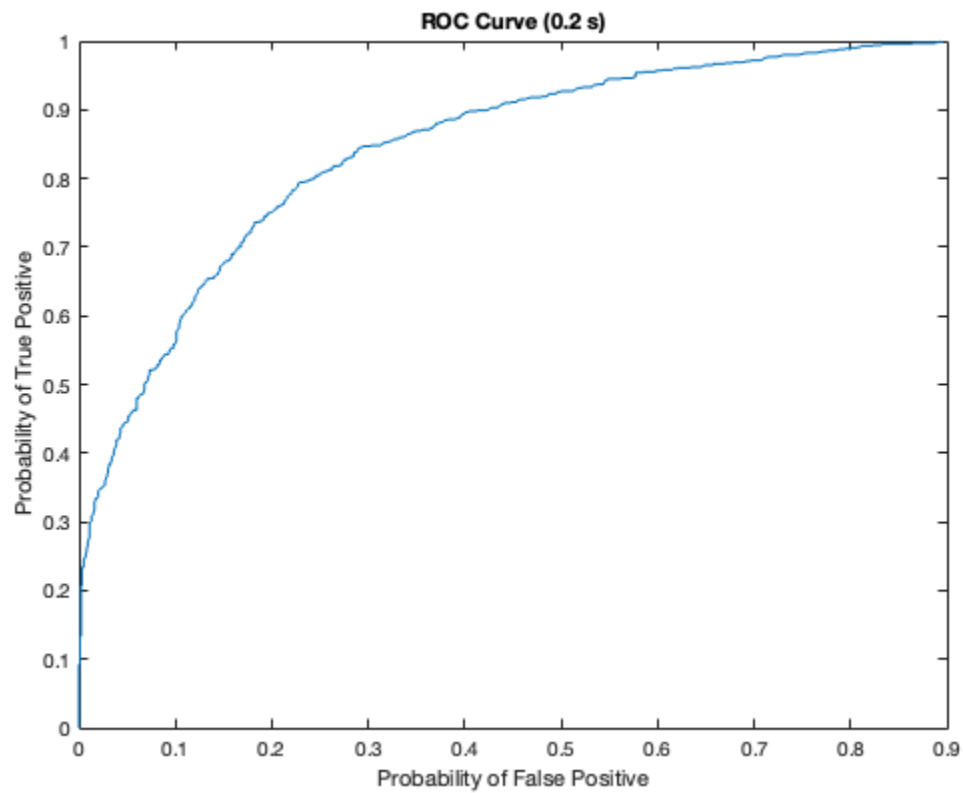
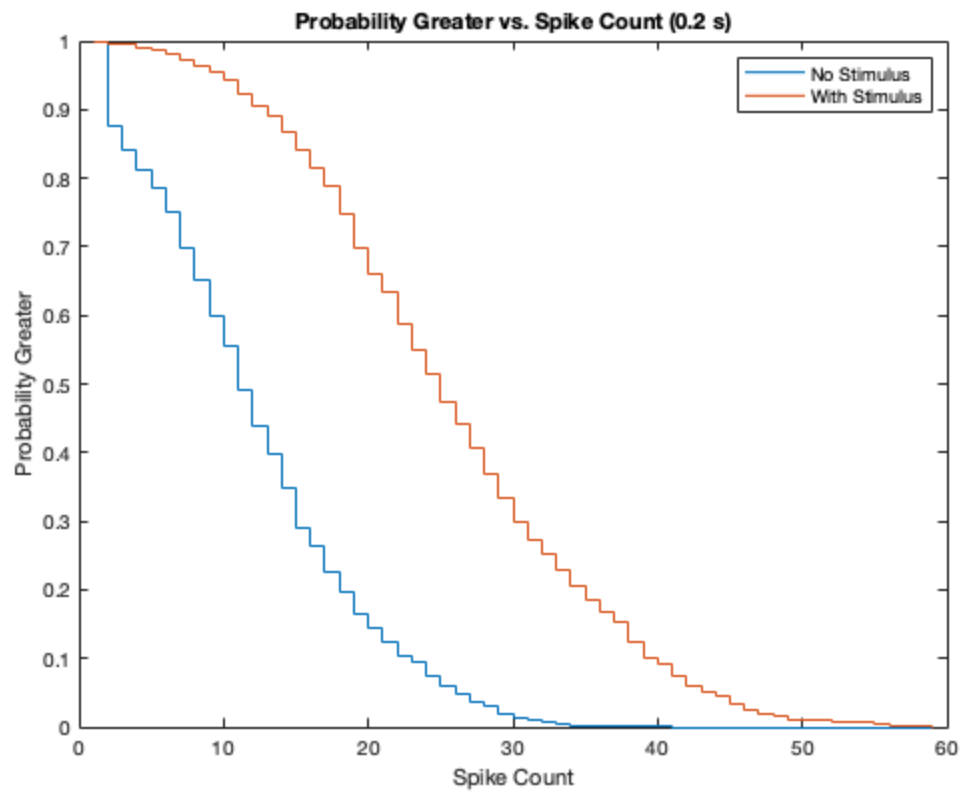
end

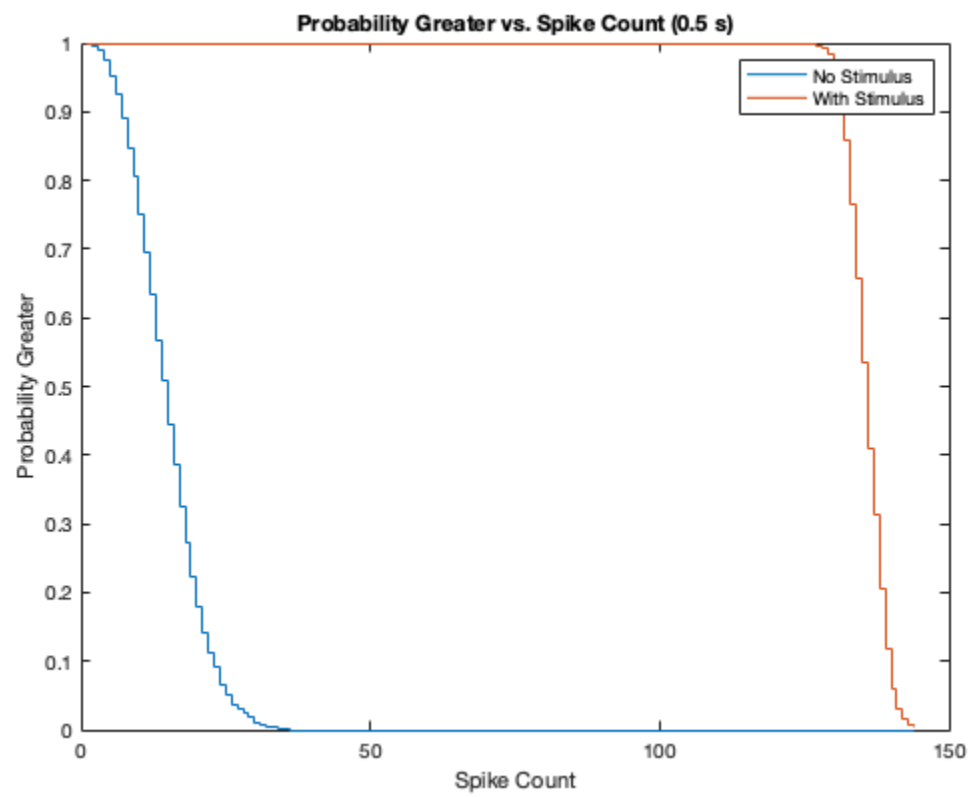
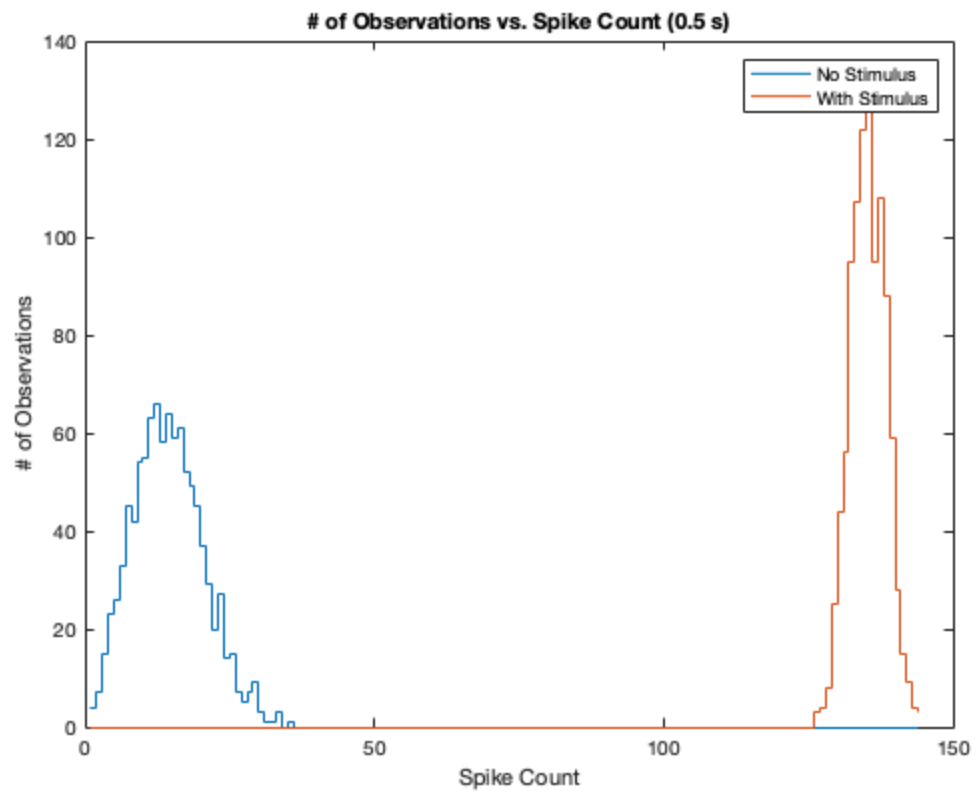
Without stimulus, the mean firing rate over 0.5 seconds is
    272887.364000 spikes/second.
With stimulus, the mean firing rate over 0.5 seconds is 609684.540000
    spikes/second.
Without stimulus, the mean firing rate over 0.2 seconds is
    42464.292000 spikes/second.
With stimulus, the mean firing rate over 0.2 seconds is 96981.986000
    spikes/second.
Without stimulus, the mean firing rate over 0.5 seconds is
    269216.952000 spikes/second.
With stimulus, the mean firing rate over 0.5 seconds is 2686986.716000
    spikes/second.

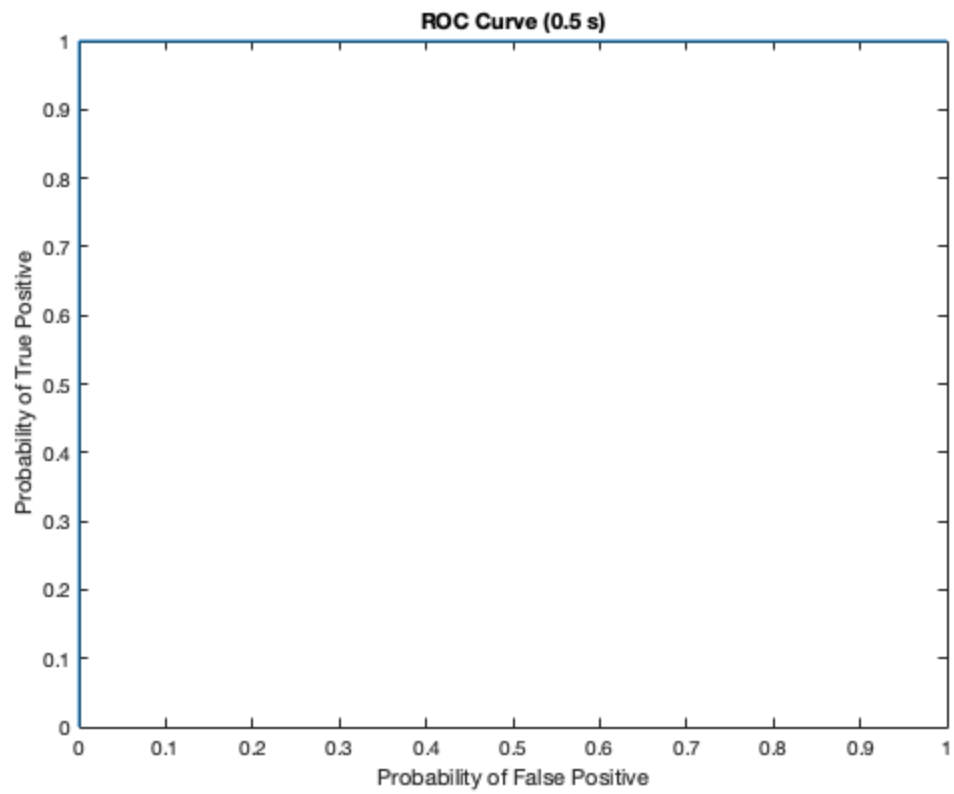
```









Published with MATLAB® R2018a