LIAF Neuron Model

In this experiment, we are going to simulate a neuron which has V_{rest} = -65 mV, spiking threshold of -50 mV, decay constant = 20 ms, V_{reset} = -70 mV, and time bins of 10 ms. We are going to use the following leaky integrate-and-fire model to simulate where I(t) is the input current:

$$\frac{dV(t)}{dt} = -\frac{1}{\tau}\left[V(t) - V_{rest}\right] + I(t)$$

PART I: Membrane Potential Implementation

The following function takes in the initial potential of the neuron, the total time of simulation, and the input sequence, one for each time bin. It will return the sequence of potentials for each time bin, and a binary sequence indicating the spikes, one bit for each time bin.

```
function [v, s] = mbr_pot(v0, time, input)
% Parameters: initial potential(v0), total time(time), input sequence(input) one for each
% time bin
% Return: the membrane potential sequence(v) and the binary spike sequence(s) of the neuron
    delta_t = 0.1;       % Time bin
    v = zeros(1, time / delta_t + 1);       % Membrane potential
    s = zeros(1, time / delta_t + 1);       % Binary spike indicator
    v_rest = -65;
    threshold = -50;
    v_reset = -70;
    decay_const = 20;

    v(1, 1) = v0;        % Simulate from t = 0
    count = 2;
    for t = delta_t : delta_t : time
        prev_v = v(1, count - 1);
        if prev_v > threshold
            v(1, count) = v_reset;
            s(1, count) = 1;         % Spike, reset
        elseif prev_v < v_reset
            v(1, count) = v_reset;  % Reset
        else
            v(1, count) = prev_v - delta_t
                 * (1 / decay_const * (prev_v - v_rest) - input(count));
        end
        count = count + 1;
    end

end
```

PART II: Test Function with Constant Inputs

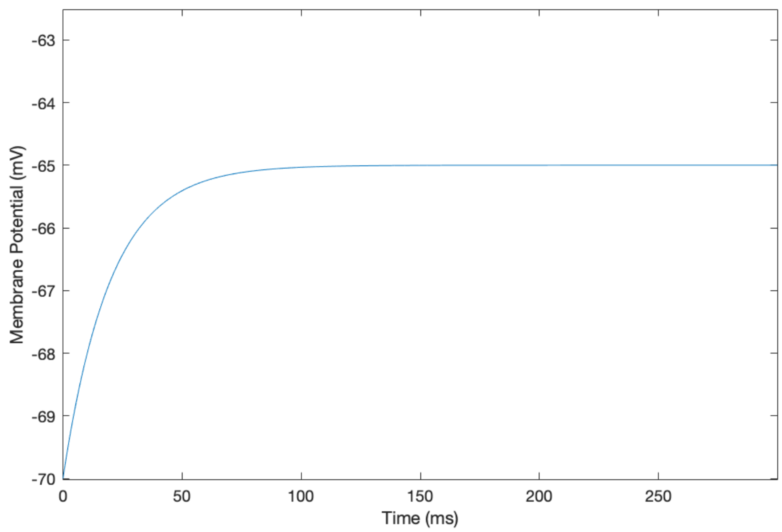
To test the function, I use the following script to plot the graph, if taking I(t)= 0, V(0) = -70 mV, and time = 300 ms:

```
input = zeros(1, 3001);      % Create a sequence of 3001 constant input which has value 0
[v, s] = mbr_pot(-70, 300, input);      % Run the simulation for 300 ms
t = [0 : 0.1 : 300];       % Create t from 0 to 300 with time bins of 0.1 ms
plot(t, v);       % Plot v vs. t
```

Here gives results on 3 different simulations:

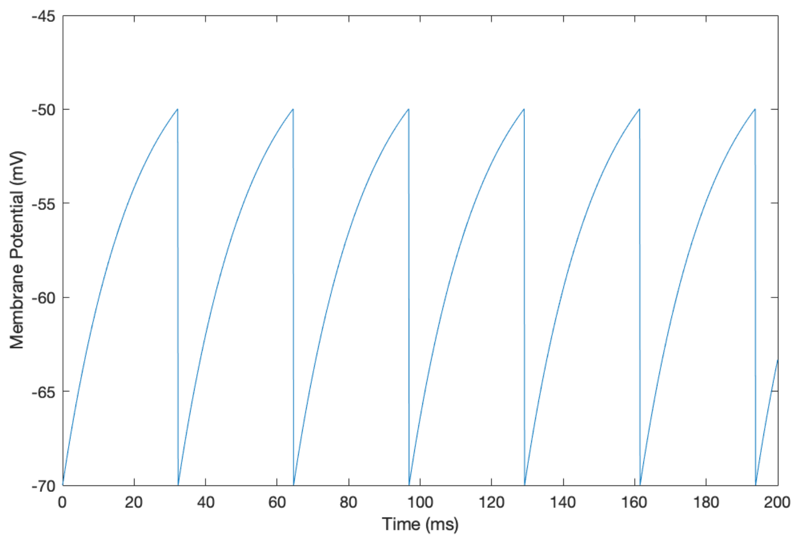
1. I(t) = 0, V(0) = -70 mV, time = 300 ms.

The membrane potential moves closer to V_{rest} from V(0) as expected, forming an asymptote curve.



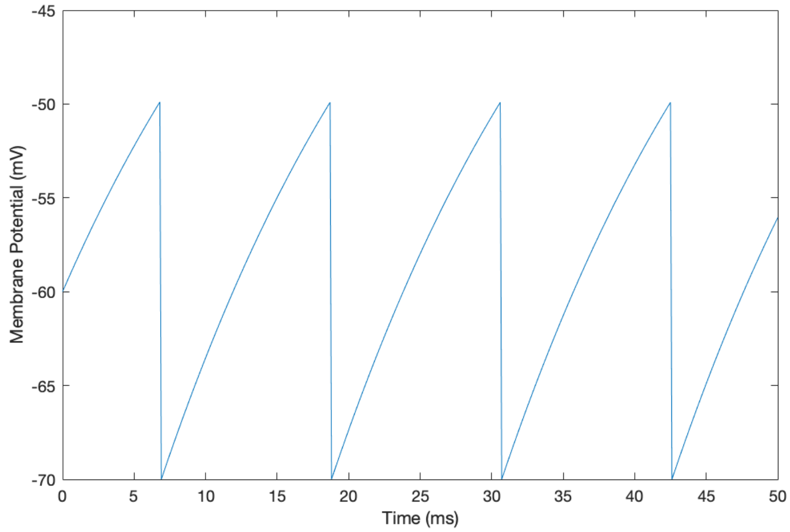
2. I(t) = 1, V(0) = -70 mV, time = 200 ms.

The potential gradually increases from V(0), reaches threshold and jumps back to V_{reset} and repeats.



3. I(t) = 2, V(0) = -60 mV, time = 50 ms.

The potential increases from V(0), reaches threshold and jumps back to V_{reset}. We can see that as constant input increases, the spike rate increases.



PART III: Spike Rate with Constant Inputs

In this part the input is still constant $I(t) = I_0$, but we change I_0 to find its relationship with the spike rate. The following function takes in the sequence of constant inputs and simulates on each for 1 second from initial potential at -70 ms, then calculate the spike rate for each.

```
function r = spk_rate(input)
% Parameter: sequence of constant inputs
% Return the spike rate corresponding to each input
    r = zeros(1, length(input));
    v0 = -70;
    time = 1000;    % Simulate for 1 s
    delta_t = 0.1;

    for index = 1 : length(input)   % Loop for each constant input
         % Construct the sequence of constant input
        i_vec = ones(1, time / delta_t + 1) * input(index);
        [~ , s] = mbr_pot(v0, time, i_vec);
        count = 0;      % Count number of spikes
        for j = s
            if j == 1
                count = count + 1;
            end
        end
        r(1, index) = count / time * 1000;     % Spike rate = count / time (#/s)
    end
end
```

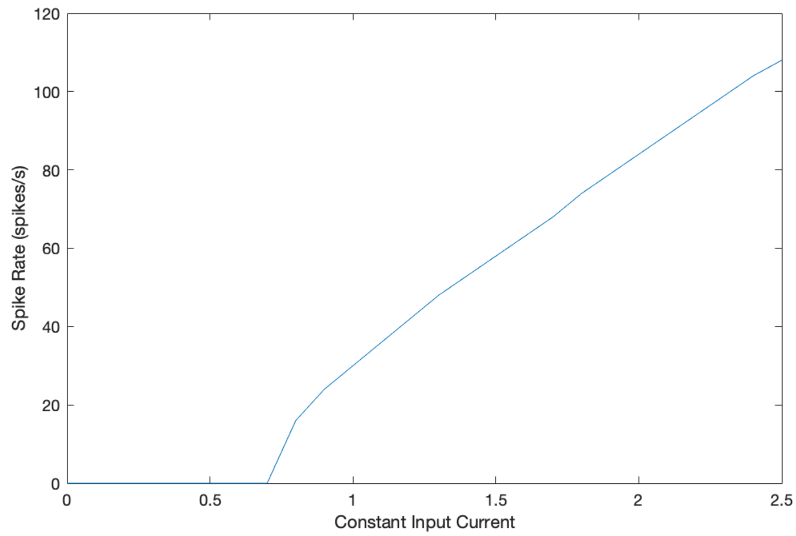
I use the following script to plot the relations between input and the spike rate. After some attempts, I found the input at which the spike rate = 100 spikes / s is about 2.3. So I'm going to use I_0 ranging from 0 to 2.5 with increment of 0.1.

```
input = [0 : 0.1 : 2.5];

r = spk_rate(input);

plot(input, r);
```

From the data I found the minimum value for I_0 for which V(t) reaches the spiking threshold is 0.7. The graph shows that the rate remains 0 until I_0 reaches 0.7, then starts to grow almost linearly as I_0 increases.



PART IV: ISI Distribution and Membrane Potential with Variable Inputs

For variable inputs, I used the following function to generate a Gaussian distributed input sequence. In this part I will choose several standard deviation values (σ_I) for the input and examine its relation to inter-spike-interval (ISI) length distribution.

```
function i = gen_input(i0, std_dev, size)
% Generate a sequence of inputs according to the Gaussian distribution of
% i0(mean) and given standard deviation, of the given size
    i = normrnd(i0, std_dev, [1, size]);
end
```

The following function calculates the ISI of given input sequence.

```
function isi = get_ISI(input, size)
% Parameter: input sequence and size of ISI wanted
% Return: sequence of ISI's with given size
    isi = zeros(1, size);
    v0 = -70;
    delta_t = 0.1;
    time = (length(input) - 1) * delta_t;

    [~, s] = mbr_pot(v0, time, input);  % Simulate

    interval = 0;       % Count the intervals passed between spikes
    count = 1;      % Count the number of ISI's recorded
    for index = 1 : length(s)
        if count > size
            break
        end

        if s(1, index) == 0
            interval = interval + 1;
        else                                % Spike detected
            interval = interval + 1;
            isi(1, count) = interval * delta_t;     % Record time from last spike
            interval = 0;
            count = count + 1;
        end
    end
end
```

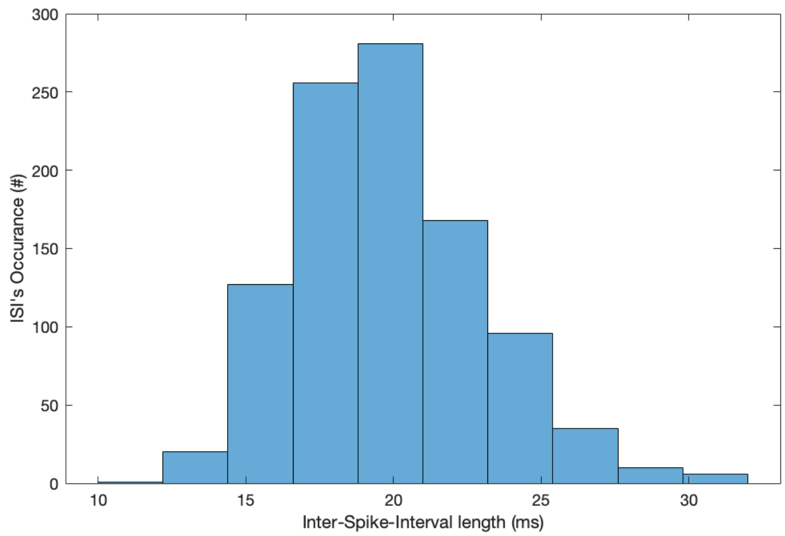
From PART III, we can find the I_0 for which the spike rate will be about 50 spikes/s. This input is 1.35. Thus for the ISI distribution data gathering, the mean of the input distribution I_0 will be set to 1.35. I collected 1000 ISI data for each σ_I, including 2, 5, 20. I use the following script for, for instance, $\sigma_I = 2$.

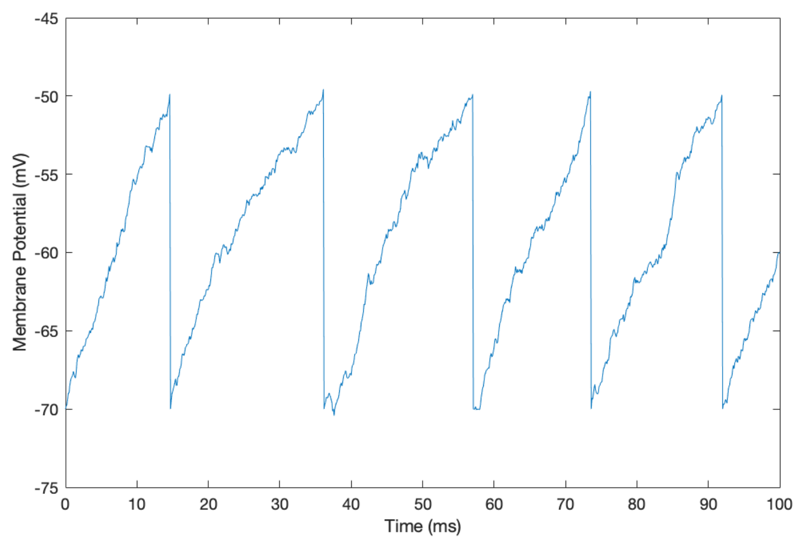
```
% Frist generate input of mean 1.35 and standard deviation of 2, of size
% 1000001 in case not enough
```

```
input = gen_input(1.35, 2, 1000001);
isi = get_ISI(input, 1000);      % Get ISI sequence of size 1000
histogram(isi, 10);
cv = std(isi) / mean(isi);      % Get coefficient of variation
```

I used time span 100 ms to better visually show the change of membrane potential. I also calculated the coefficient of variation (CV) for each ISI dataset. Here is the data I collected for different standard deviations:

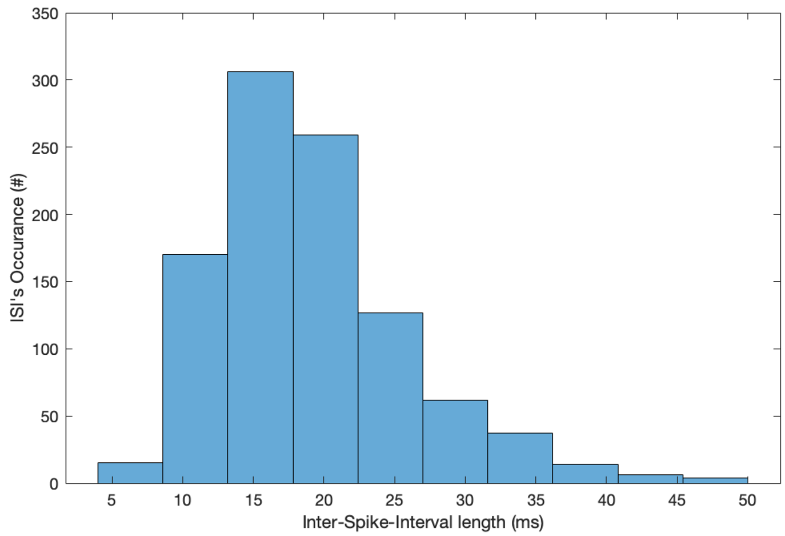
1. $\sigma_I = 2$. CV = 0.1456.
   The ISI distribution is a beautiful Gaussian distribution ranging from 10 ms to 30 ms. The membrane potential is affected by the noise but still shows a good shape of our model.

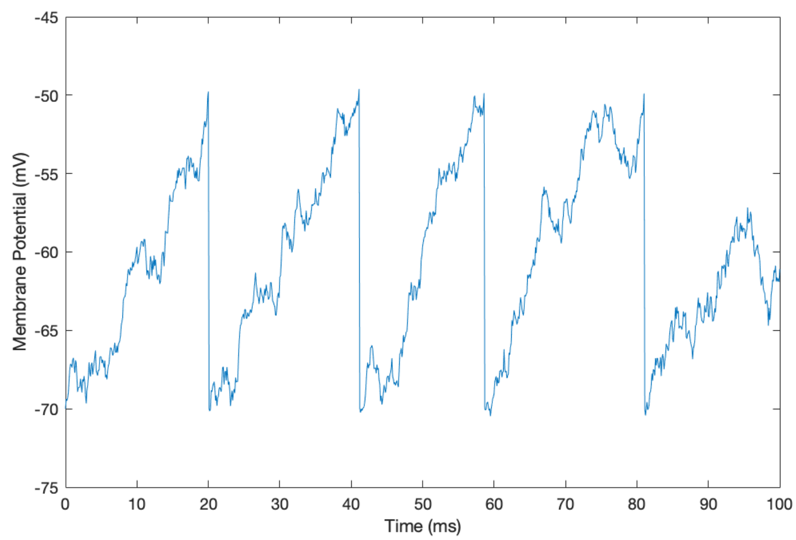




2. $\sigma_I = 5$. CV = 0.3557.

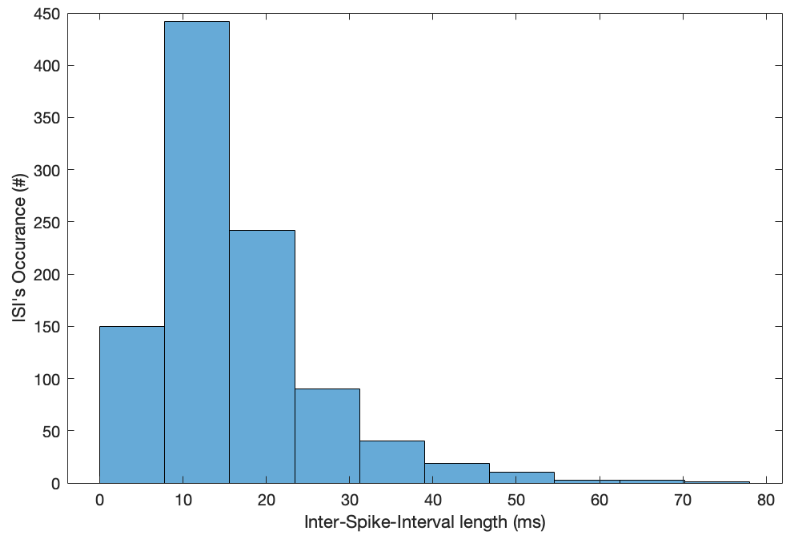
We see that the ISI data become sparser as σ_I increases, ranging from 5 ms to 50 ms. The membrane potential is clearly affected by more noise, but there are still 4 spikes in 100 ms.

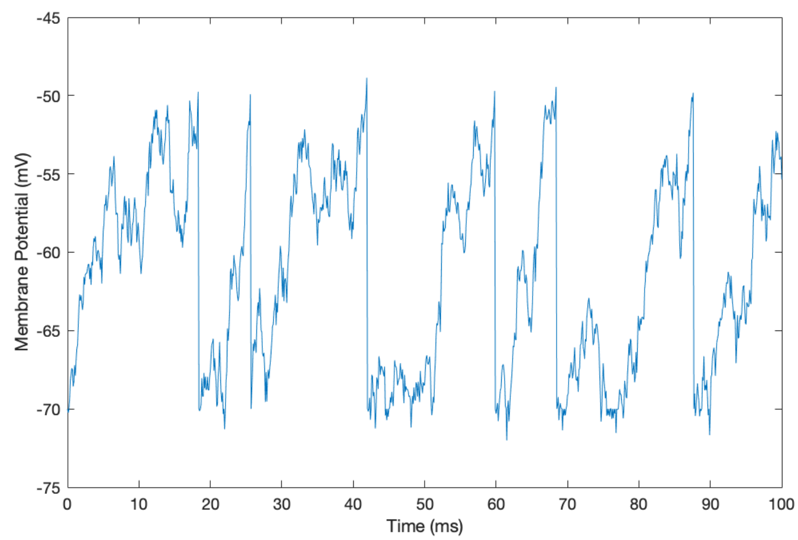




3. σ_I = 10. CV = 0.5946.

Now the ISI distribution becomes too sparse that it ranges from 0 ms to almost 80 ms. The membrane potential is greatly affected by the noise and has 6 spikes in 100 ms. Also, it needs to reset itself for multiple times because V becomes lower than the V_{reset} sometimes.





PART V: ISI Coefficient of Variation and Spike Rate vs Input Standard Deviation

In this part I will vary σ_I and find how it affects the CV of ISI and the spike rate. The following two functions both take in the sigma sequence, the mean of inputs, and the time we want to simulate for.

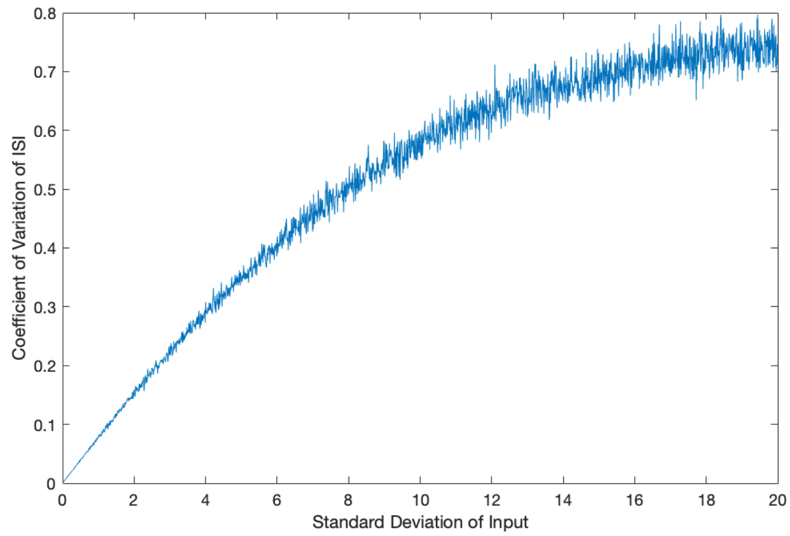
```
function cv = cv_from_sigma(i0, sigma, time)
% Parameters: mean of input(i0), sequence of standard deviations of
% input(sigma), simulation time for each sigma
% Returns: the CV of ISI's for each corresponding sigma
    cv = zeros(1, length(sigma));
    delta_t = 0.1;
    for index = 1 : length(sigma)    % Simulate for each
        input = gen_input(i0, sigma(1, index), time / delta_t + 1);      % Generate enough
input to simulate
        isi = get_ISI(input, 1000);       % Here calls the simulation, collects 1000 ISI data
        cv(1, index) = std(isi) / mean(isi);
    end
end

function r = spk_rate_from_sigma(i0, sigma, time)
% Parameters: mean of input(i0), sequence of standard deviations of
% input(sigma), simulation time for each sigma
% Returns: the spiking rate for each corresponding sigma
    r = zeros(1, length(sigma));
    delta_t = 0.1;
    for index = 1 : length(sigma)        % Simulate for each sigma
        input = gen_input(i0, sigma(1, index), time / delta_t + 1); % Generate enough input
        [~, s] = mbr_pot(-70, time, input);      % Simulate
        count = 0;       % Count number of spikes each simulation
        for j = s
            if j == 1
                count = count + 1;
            end
        end
        r(1, index) = count / time * 1000;       % Record spiking rate
    end
end
```

I still uses $I_0 = 1.35$ for a 50 spikes/s. I choose sigma in the range of 0 to 20, with 0.01 intervals. I simulated each for 100 seconds, and here's the graph I got:

1. CV vs σ_I.

The CV becomes greater and sparser as σ_I increases. But it becomes asymptote which I did not expect. The longer I simulate, the less sparser CV becomes.



2. Spike Rate vs σ_I.

Similarly, the spike rate increases greatly as σ_I increases and becomes more sparse. Again, the longer time I run the simulation, the less the error (variation) the spike rate becomes