Neuro 120 Homework 2: Data Analysis

William Schmitt and Will Drew

Due: Thursday 18 October 2018

1 Question 1: Auditory Neuroplasticity

1.1 Raster Plot of Single-Unit Activity

To plot the raster plot, we wrote the following code:

```
stimulus_start_times = 0:1/6:(60); % In seconds
%% Part A
% Make raster plot
figure(1);
hold on
for i = 1:(length(stimulus_start_times)-1)
    spikes_in_window = spikes_single_unit((spikes_single_unit > ...
        stimulus_start_times(i)) & (spikes_single_unit < ...
        stimulus_start_times(i + 1)));
    spikes_normalized = (spikes_in_window - stimulus_start_times(i))';
    trial_num = ones(1, length(spikes_normalized))*i;
    plot([spikes_normalized; spikes_normalized], [trial_num; trial_num-1],'k',
        'LineWidth',3)
end
xlim([0,0.167])
ylim([0,360])
yticks(0:30:360)
xlabel('Time (s)');
ylabel('Trial Number');
title('Response of a Single-Unit to the Exposure Stimulus');
```

This produces the raster plot shown in Figure 1, which clearly shows that the neuron responds consistently to the stimulus about 0.03 seconds into the start of the trial.

1.2 Gaussian Kernel Firing Rate Estimate

We calculate the estimate of the firing rate of this single-unit neuron by writing the following code:

```
%% Part B
% Create gaussian filter
figure(2);
```

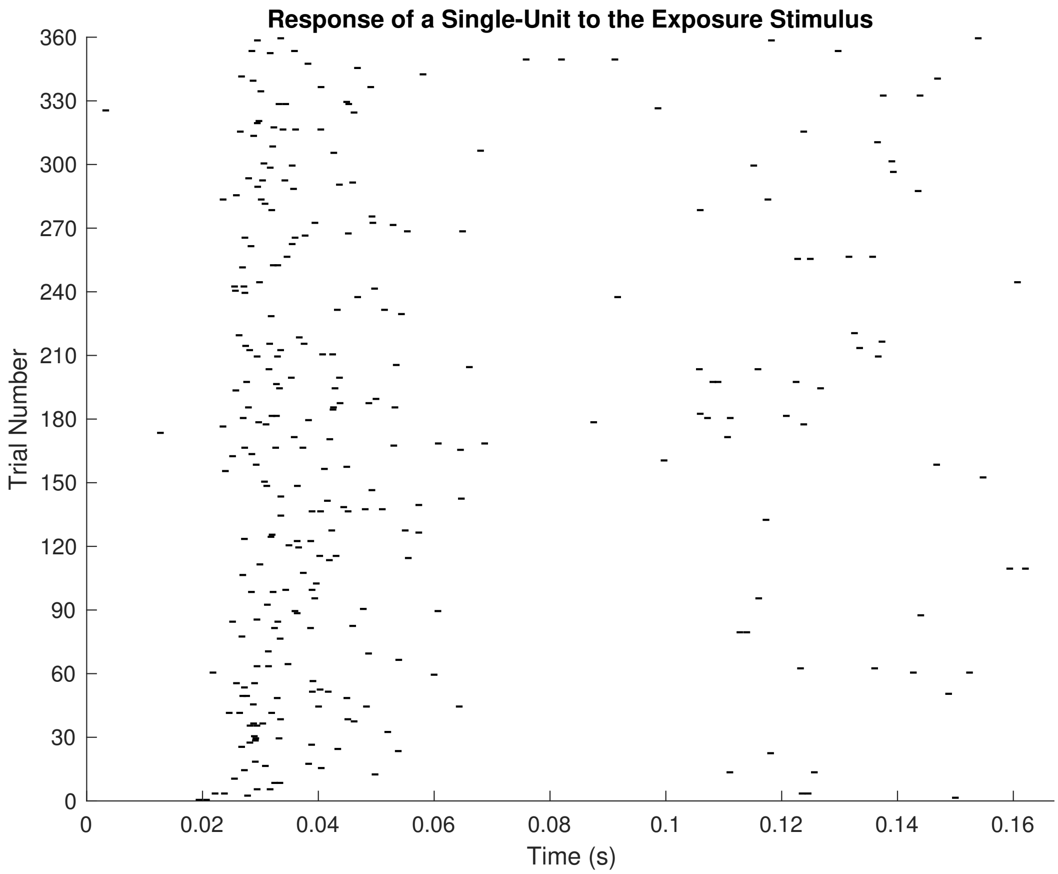

Figure 1: Raster Plot of Single-Unit Activity.

```
x=[0:0.0001:1/6];
avg_dist = zeros(1,length(x));
for i = 1:(length(stimulus_start_times)-1)
    spikes_in_window = spikes_single_unit((spikes_single_unit > ...
        stimulus_start_times(i)) & (spikes_single_unit < ...
        stimulus_start_times(i + 1)));
    spikes_normalized = (spikes_in_window - stimulus_start_times(i))';
    trial_num = ones(1, length(spikes_normalized))*i;
    norm = zeros(1,length(x));
    for j=1:length(spikes_normalized)

        norm = norm + normpdf(x,spikes_normalized(j),0.005);
    end
    avg_dist = avg_dist+norm;
end
plot(x,avg_dist./360)
xlabel('Time (s)');
ylabel('Firing Rate (Hz)');
```

This code simply places a Gaussian distribution with a standard deviation of 0.005s centered at the location of each spike and averages these distributions over all stimulus trials. This work produces Figure 2, which shows that the firing rate increases dramatically around 0.03s into the trial, going from a baseline firing rate of about 5 Hz to a peak of just over 25 Hz.

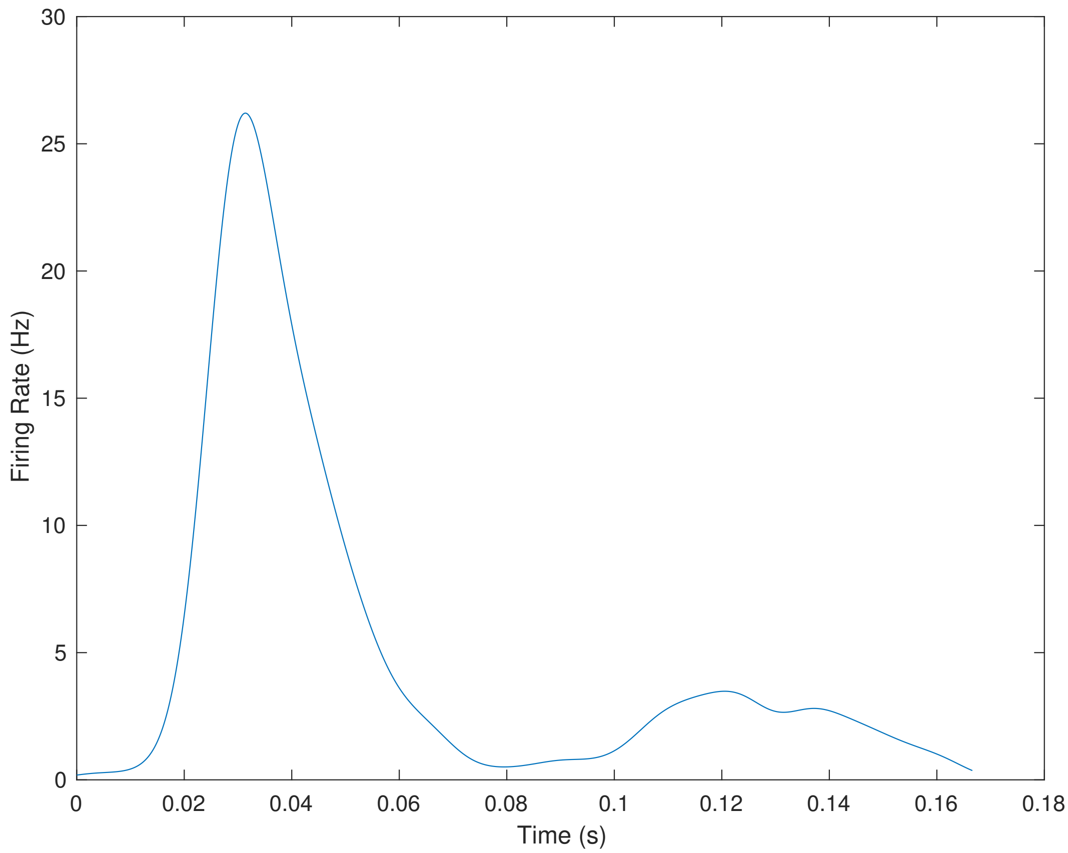

Figure 2: Gaussian Estimation of Firing Rate Using $\sigma = 0.005s$.

1.3 Gaussian Kernel Parameter Variation

We next alter the σ parameter of the Gaussian kernel from above and generate Figures 3 and 4 which shows the effect of this change. We used the following code to make these figures.

```
%% Part C
% sigma = 50ms, 0.5ms
sigma = [0.05,0.0005]
for k = 1:2
    figure(k+2);
    x=[0:0.0001:1/6];
    avg_dist = zeros(1,length(x));
    for i = 1:(length(stimulus_start_times)-1)
        spikes_in_window = spikes_single_unit((spikes_single_unit > ...
            stimulus_start_times(i)) & (spikes_single_unit < ...
            stimulus_start_times(i + 1)));
```

```
        spikes_normalized = (spikes_in_window - stimulus_start_times(i))';
        trial_num = ones(1, length(spikes_normalized))*i;
        norm = zeros(1,length(x));
        for j=1:length(spikes_normalized)
            norm = norm + normpdf(x,spikes_normalized(j),sigma(k));
        end
        avg_dist = avg_dist+norm;
    end
    plot(x,avg_dist./360)
    xlabel('Time (s)');
    ylabel('Firing Rate (Hz)');
end
```

We can see from these figures that a very small standard deviation results in a quite noisy curve with large, quick oscillations, which intuitively, does not seem to describe the behavior of the neuron well as it is unlikely the firing rate would change that quickly. Conversely, a very large standard deviation results in a very smooth, wide curve that also does not seem to describe the data well as it predicts a large firing rate at the start of the experiment, where the raster plot indicates there is very little activity.

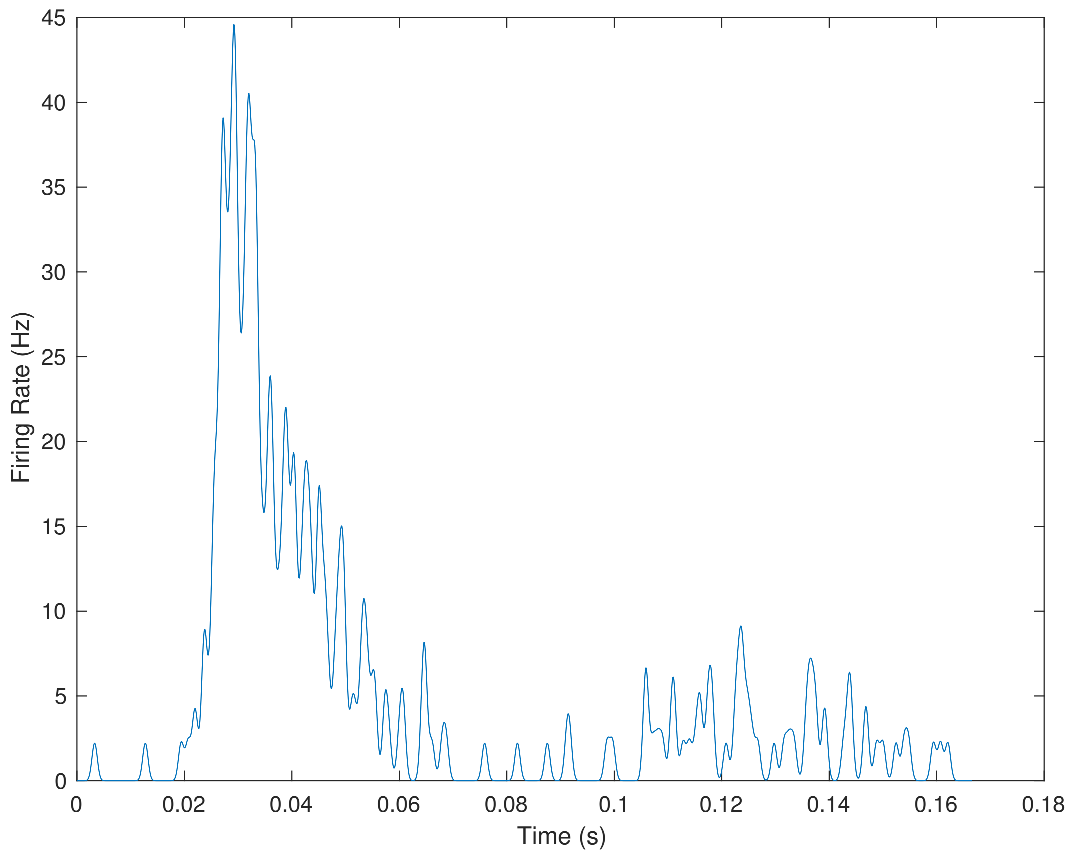

Figure 3: Gaussian Estimation of Firing Rate Using $\sigma = 0.0005s$.

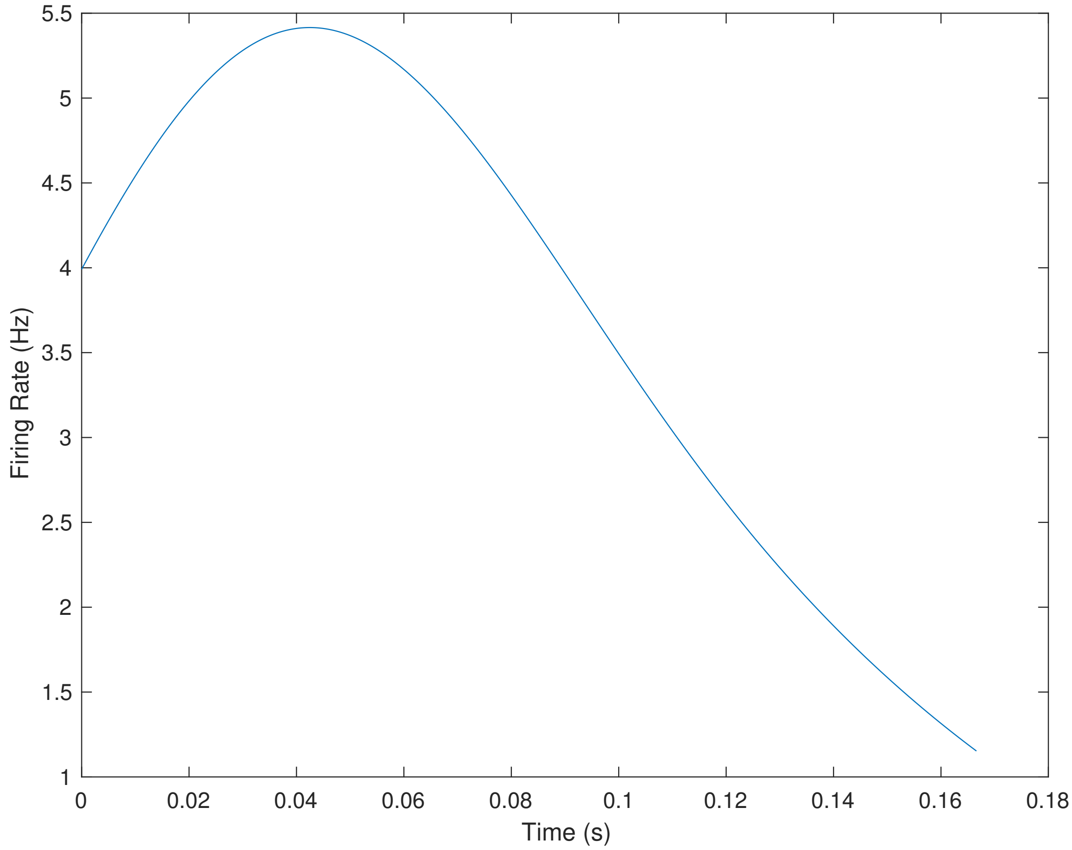

Figure 4: Gaussian Estimation of Firing Rate Using $\sigma = 0.05s$.

1.4 Grand Average Post-Stimulus Time Histogram

We compute the grand average poststimulus time histogram using the following code:

```
%% Part D
figure(5);
hold on;
nbins = length(0:0.005:(1/6))
neural_data = {spikes_control, spikes_exp};
total_counts = {zeros(1, nbins), zeros(1, nbins)};
num_neurons = {N_control, N_exp};
for condition = 1:2
    for i = 1:(length(stimulus_start_times)-1)
        spikes_in_window = neural_data{condition}((neural_data{condition} > ...
            stimulus_start_times(i)) & (neural_data{condition} < ...
            stimulus_start_times(i + 1)));
        spikes_normalized = (spikes_in_window - stimulus_start_times(i))';
        total_counts{condition} = total_counts{condition} + hist(spikes_normalized,
            nbins);
    end
    total_counts{condition} =
```

```
        (total_counts{condition}/(360*num_neurons{condition}*.005));
end
bar([total_counts{2}', total_counts{1}'])
```

This code generates Figure 5, which shows us the same trend we saw in the previous figures. The stimulus evokes a larger response from the population of neurons about 0.03 seconds after the onset of a trial.

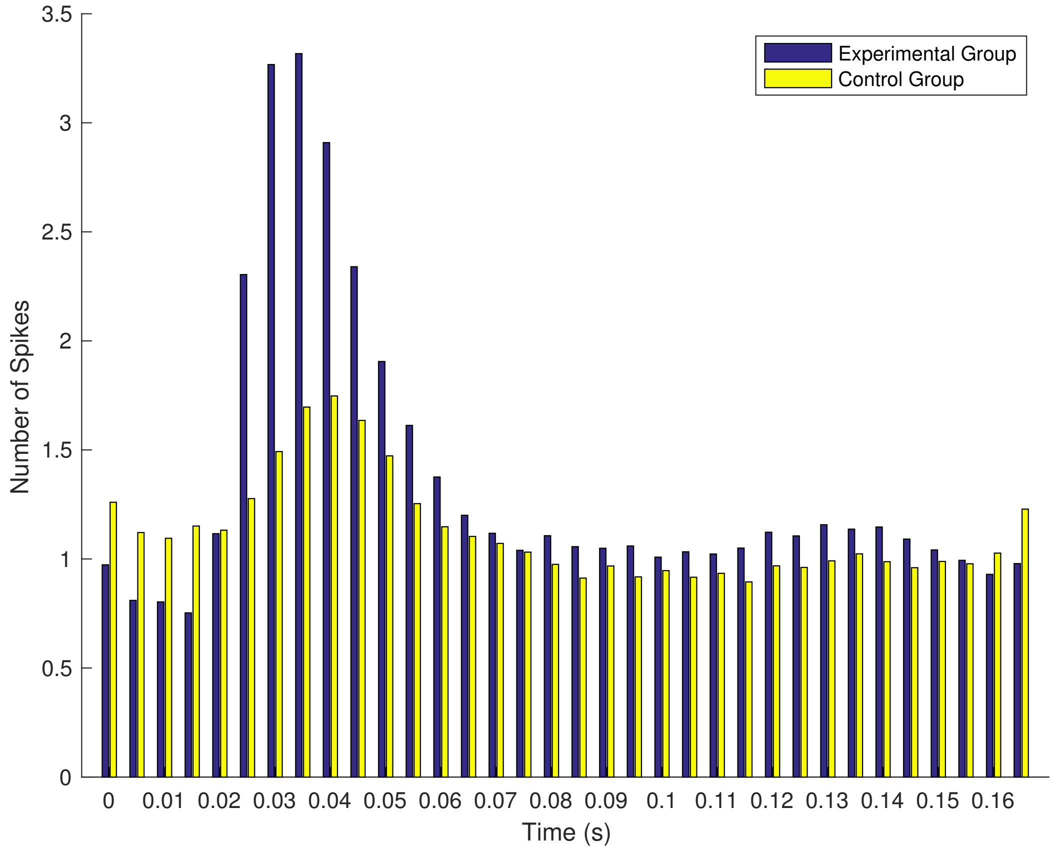

Figure 5: Grand Average Post-Stimulus Time Histogram of two Populations of Neurons. Bin width was 5ms.

1.5 Differences between Experimental and Control Groups

Looking at Figure 5, we see that the experimental group has a stronger response (i.e. more spikes) during the peak firing activity (about 0.02s to 0.05s after the onset of the trial) of the neurons. However, the control neuron population still exhibits this increase, so therefore it is just that the experimental group of neurons became more selective to the exposure stimulus while the precision of the response did not appear to change. Finally, we can see that there is very little difference in the activity of the two groups (beyond some higher activity in the control group at the start and end of a trial) outside of this peak firing range.

1.6 Spike-Triggered Average for Neuron

We generate a spike-triggered average using the following code:

```
clear all

% Load DMR stimulus specrogram and spiking responses from one neuron
load dmr_experiment

% Plot spectrogram of stimulus
figure(1)
plot_spectrogram(stim_spectrogram, stim_time, stim_freq)

%% Generate STA
figure(2)
t_past = 125; % in ms
t_future = 125; % in ms
sampling_rate = mean(median(diff(stim_time)));
sta_time = (-t_past/1000):sampling_rate:(t_future/1000);
sta_freq = stim_freq;

sta_spectro = zeros(38, 51);
for i = 1:982
    spike_index = floor(spikes(i)/0.005);
    for j = 1:38
        sta_spectro(j,:) =
            sta_spectro(j,:)+stim_spectrogram(j,spike_index-25:spike_index+25);
    end
end

sta_spectro = sta_spectro/982;

plot_spectrogram(sta_spectro, sta_time, sta_freq)
colorbar;
```

This code generates Figure 6, which plots the spike-triggered average spectrogram from the DMR stimulus.

1.7 Frequency Selectivity of Neuron

The neuron appears to be most sensitive to a frequency of 7127 Hz.

1.8 Neuronal Stimulus Preference

According to the STA, the neuron's peak response would be higher to a brief tone pip. This is because if the neurons were responsive to constant tones, we would expect longer stretches of spikes instead of short bursts. The tone pip should be about 30ms long to evoke the largest response.

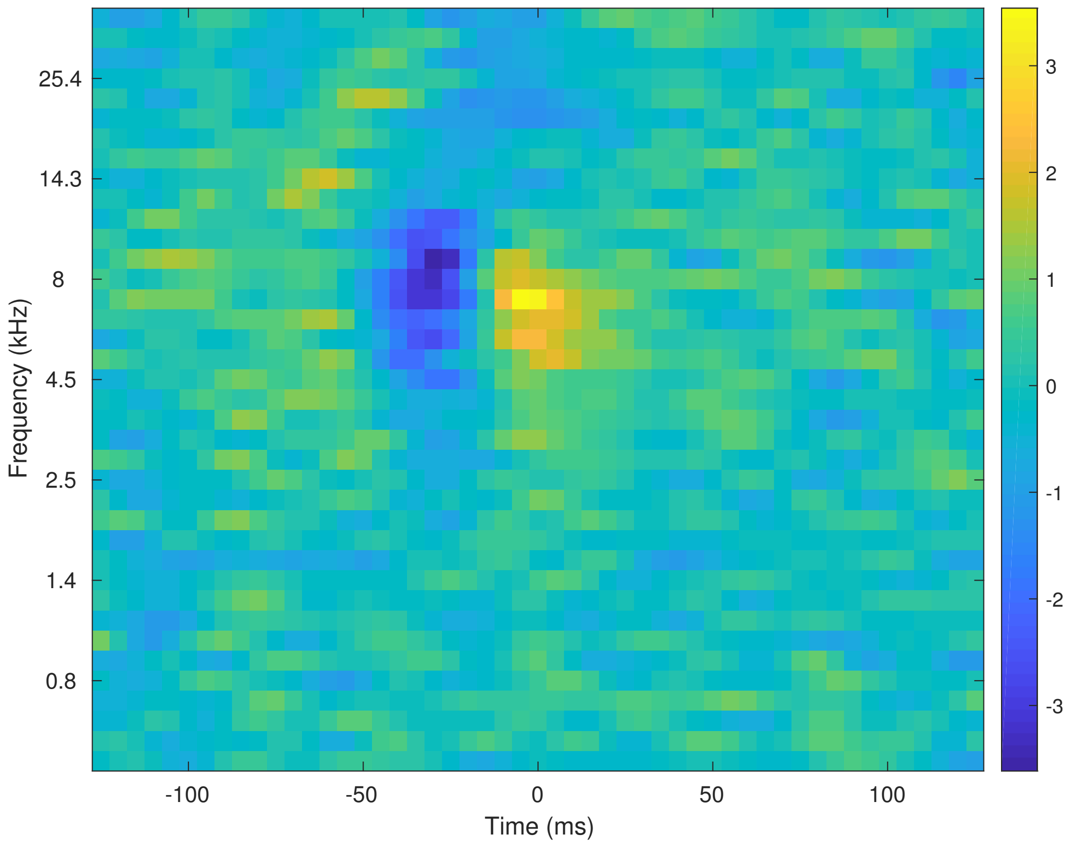

Figure 6: Spectrogram of spike-triggered average from DMR stimulus. Time range: ± 125 ms.

1.9 Potential Stimulus Correlations

We plot the stimulus correlation matrix using the following code:

```
%% Plot stimulus correlation matrix
figure(3);
plot_spectrogram(stim_spectrogram*stim_spectrogram', 38, stim_freq);
colorbar;
```

This code generates Figure 7, which appears to be close to an identity matrix.

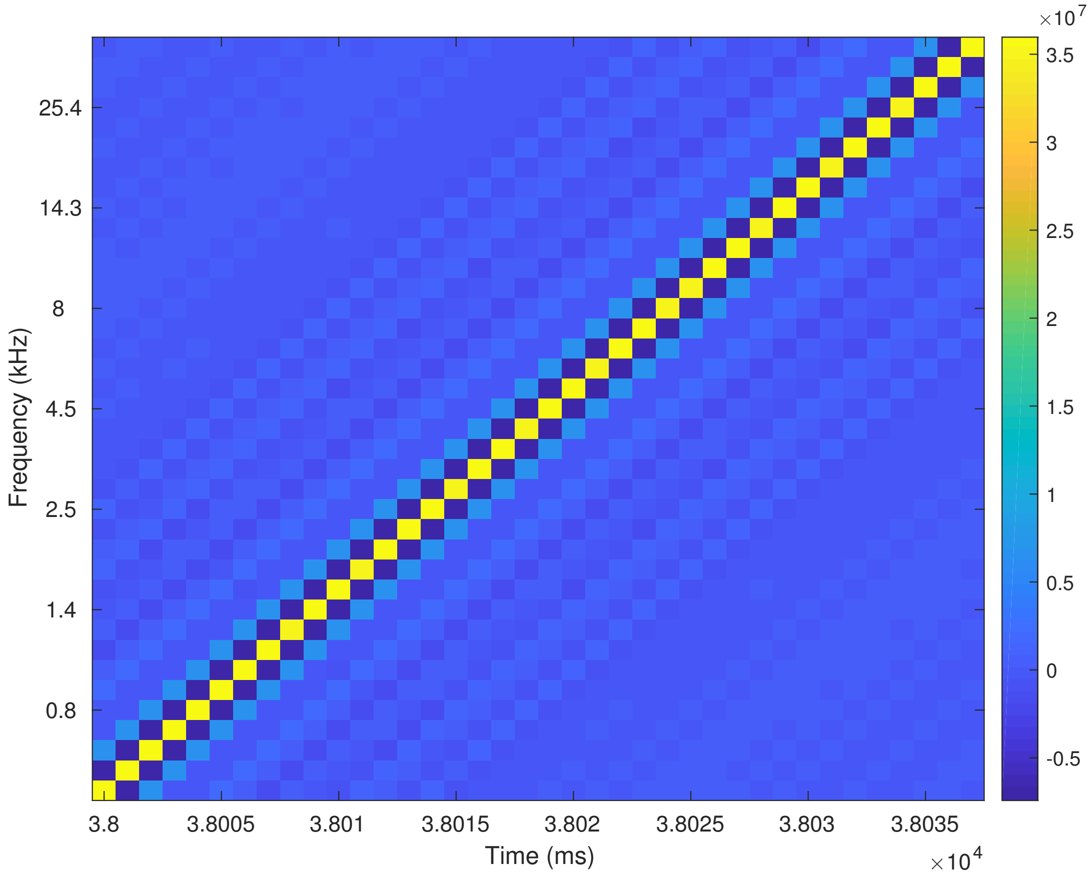

Figure 7: Spectrogram of stimulus correlation matrix.

2 Random Neural Networks and Overfitting in Regression

2.1 Linear Regression with Regularization

2.2 Overdetermined Networks

2.3 Underdetermined Networks

2.4 Label Noise and Regularization Parameters

2.5 Reflections

3 Image Demixing

3.1 Covariance Matrix and PCA

3.1.1 De-meaning the data

3.1.2 Covariance Matrix Function

3.1.3 Eigenvectors and values of covariance matrix

3.1.4 Time Required for Computation