Trey Bradley

Table of Contents

October 2, 2018 ; MIR - Assignment 1_Implementation 4

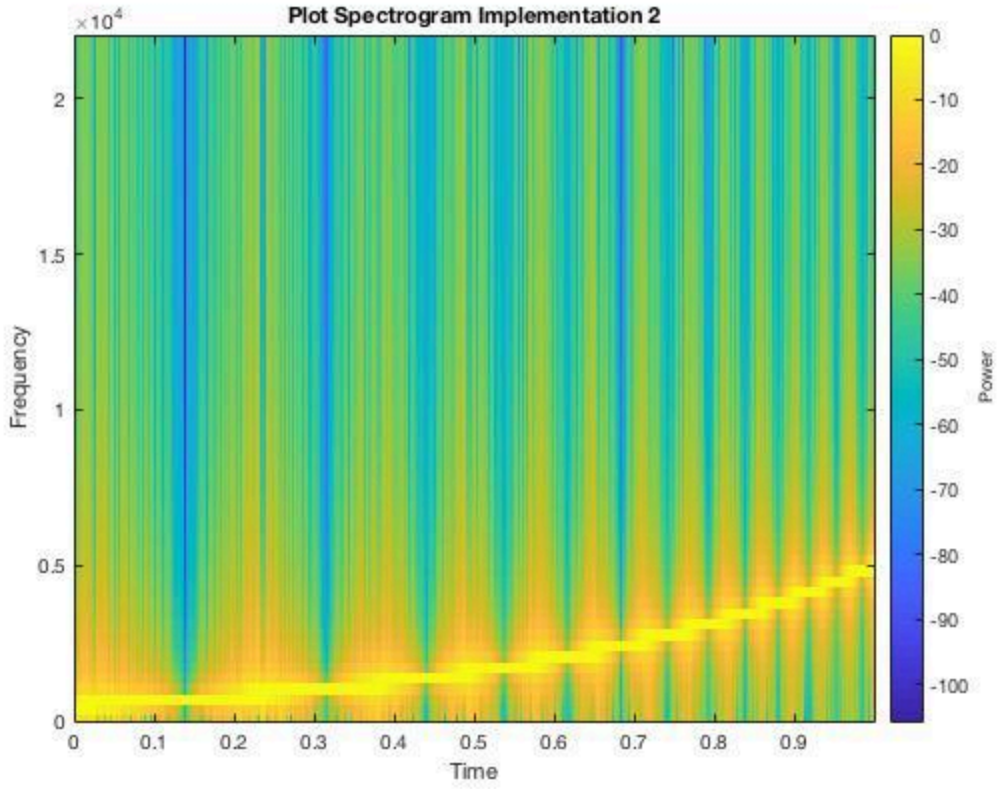
Implementation 1 (Sound)

```
f_min = 500;
f_max = 5000;
fs = 44100;
duration = 1;
a = .95;
x_t = sine_sweep(f_min, f_max, fs, duration, a);
```

Analysis (A):

MATLAB's Spectrogram Function (plot_spectrogram) with Default Parameters This figure illustrates the computation of the spectrogram of a sine sweep signal according to the parameters described above in the Implementation 1 (Sound) section. With a nfft size at this level, the frequency resolution is approximately 344Hz = fs/nfft. The frequency resolution can be visualized by a poor representation of the variance in frequency, which results from fewer windows, and fewer nfft points to represent frequency bands.

```
N = 128;
win_size = N;
hop_size = N/2;
win_type = 'rect';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);
```

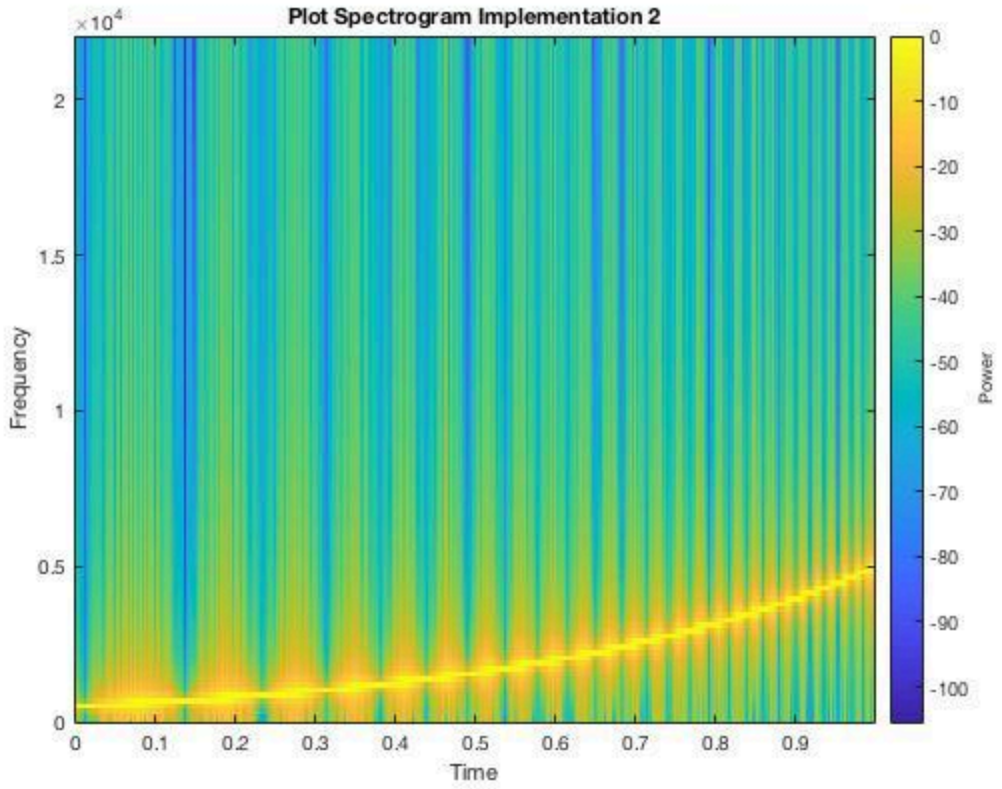


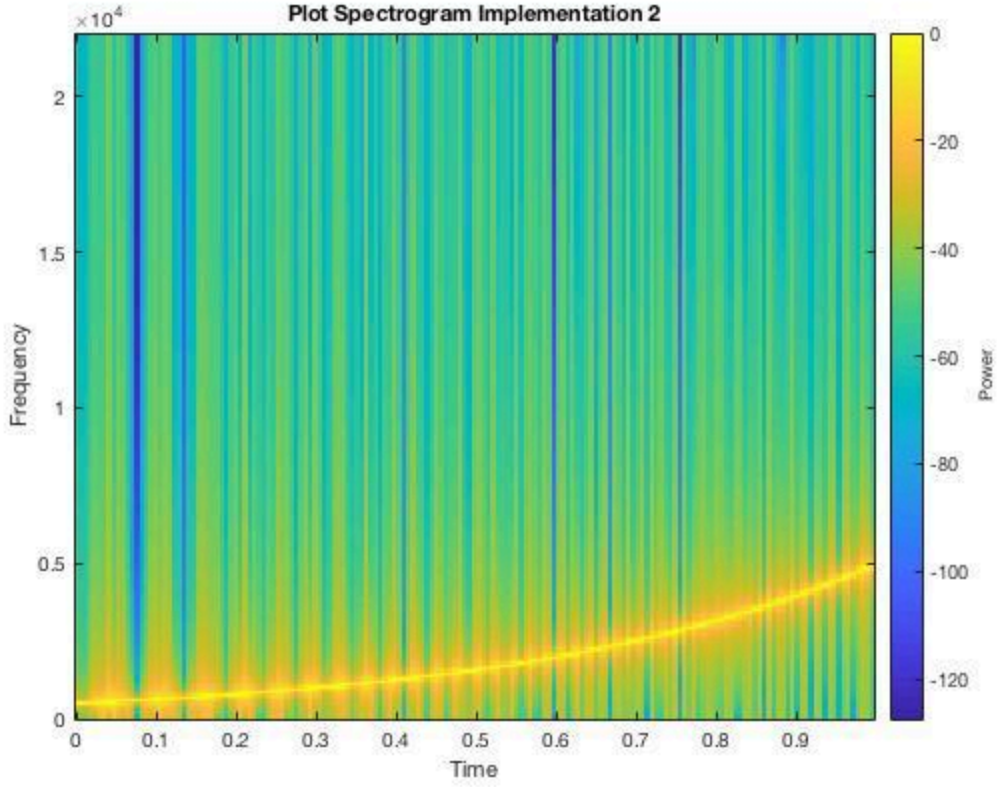
Analysis (B):

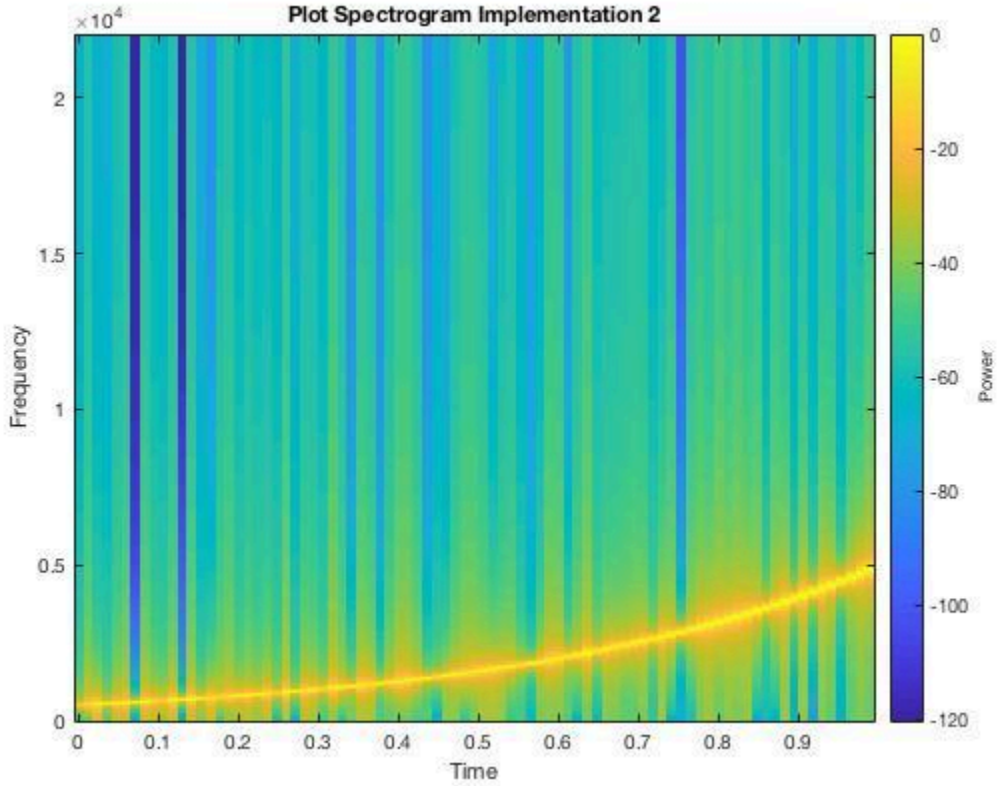
MATLAB's Spectrogram Function with N = 256, 512, and 1024 This plot illustrates the tradeoff between frequency resolution and time resolution. As the window size increases (greater segments of time that we operate the fft on) we see less time resolution and greater frequency resolution, which is clear from the smoother sweep function. The larger window size and larger nfft size (which is defaulted to nfft = window size) also results in greater frequency resolution since we now are taking more nfft points per frequency bin.

```
N = 256;
win_size = N;
hop_size = N/2;
win_type = 'rect';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);

N = 512;
win_size = N;
hop_size = N/2;
win_type = 'rect';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);

N = 1024;
win_size = N;
hop_size = N/2;
win_type = 'rect';
```

```
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);
```



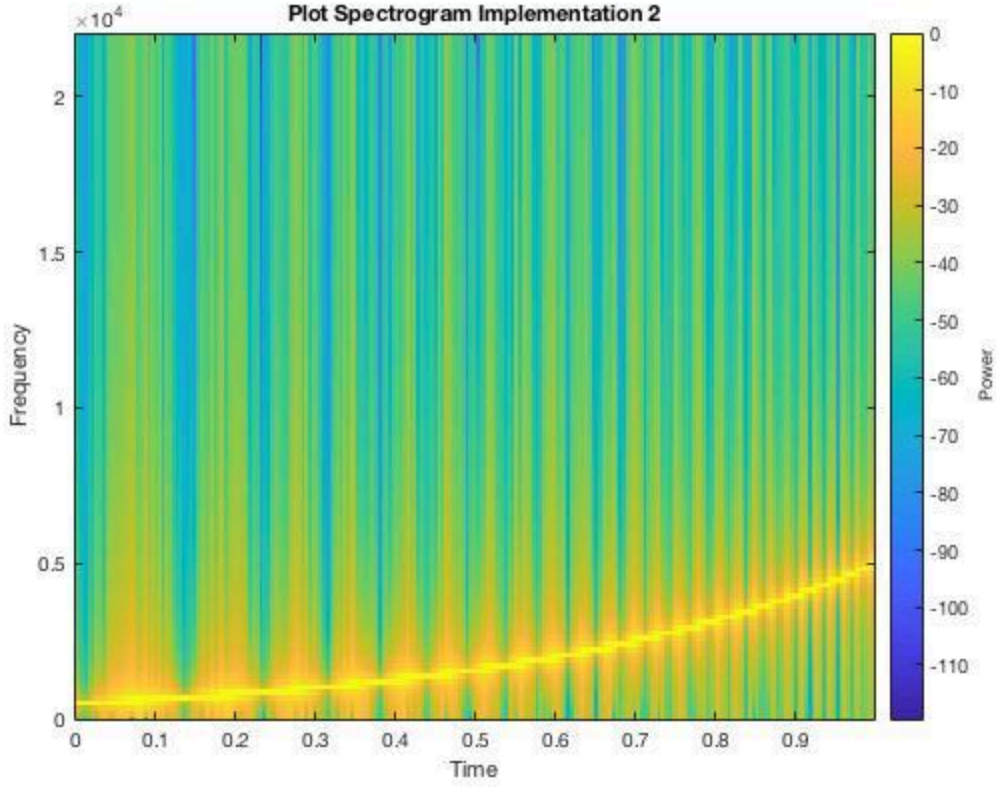

Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-20
-40
-60
-80
-100
-120

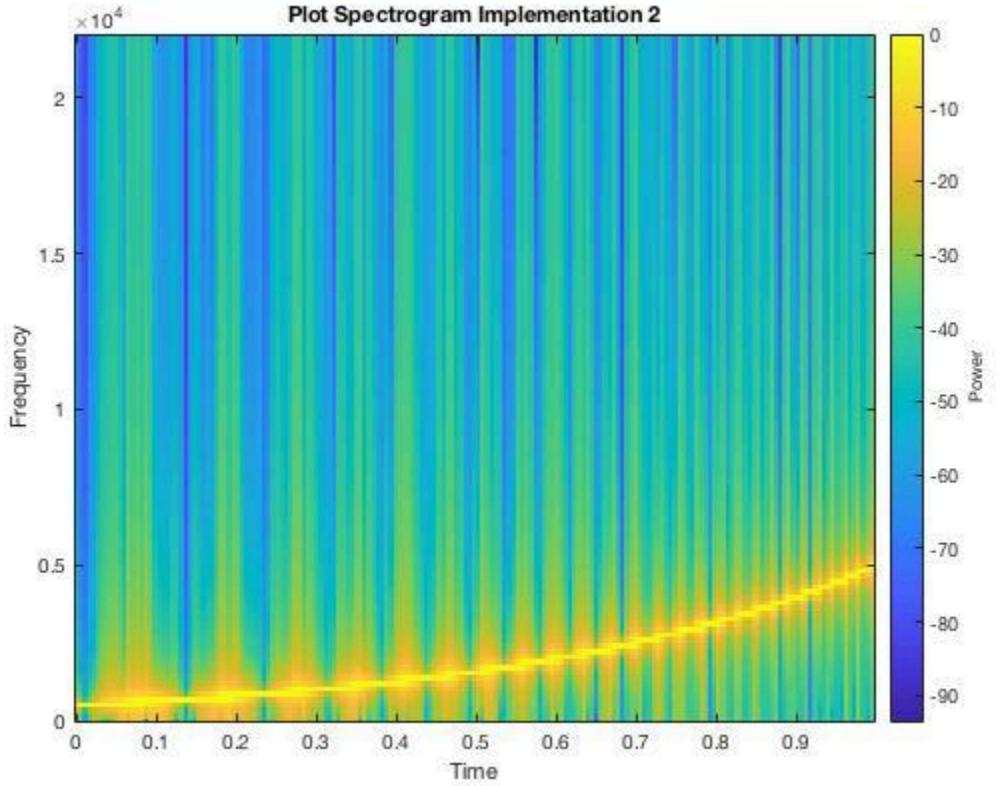

Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-20
-40
-60
-80
-100
-120

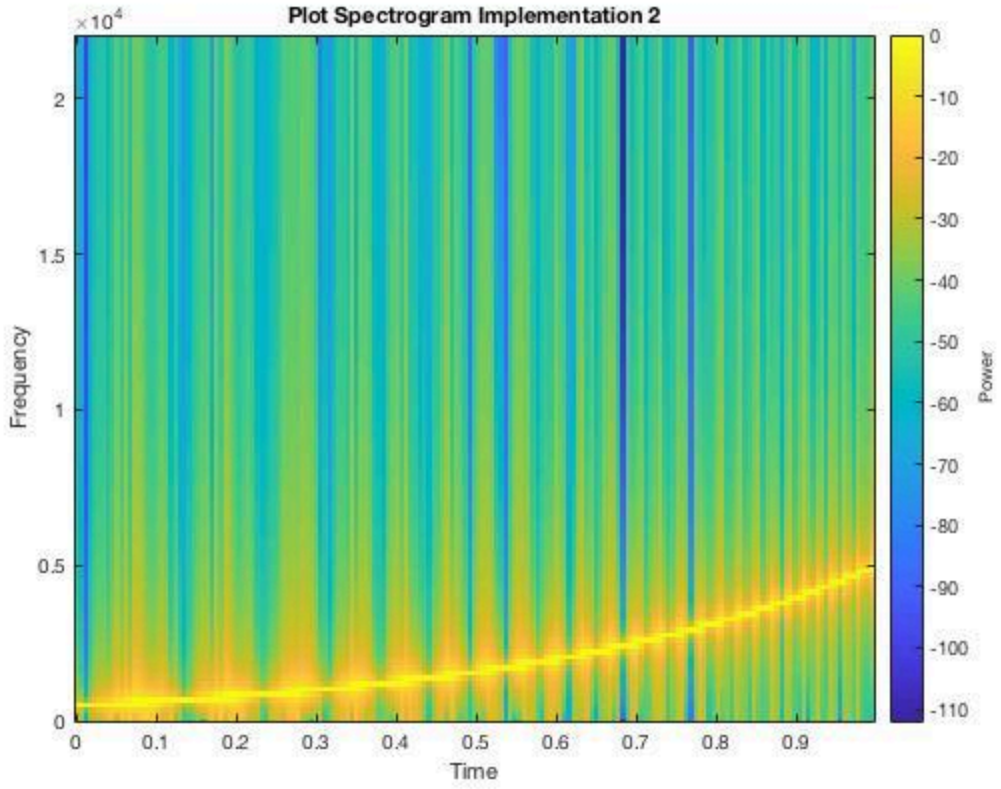
Analysis (C):

MATLAB's Spectrogram Function, where h = N/4, h = N/16, h = N/32, where the window size N = 256 With a fixed window size and decreasing overlap index?while not affecting frequency resolution?has an affect on the level of amplitude modulation that sideband frequencies undergo, which can be visualized by the amount of energy in the higher frequency ranges. When hop size is small (where h=N/32), the effect of the windows is increased, where side band information is being attenuated more. Less attenuation of the sidebands occurs when hop size is large, as more spectral content is being leaked (not captured by windows). This results in higher energy at the sidebands (since they are not being attenuated as much), visualized by the increase in yellow color as h increases.

```
N = 256;
win_size = N;
hop_size = N/4;
win_type = 'rect';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);

N = 256;
win_size = N;
hop_size = N/16;
win_type = 'rect';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);

N = 256;
win_size = N;
hop_size = N/32;
win_type = 'rect';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);
```


Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110

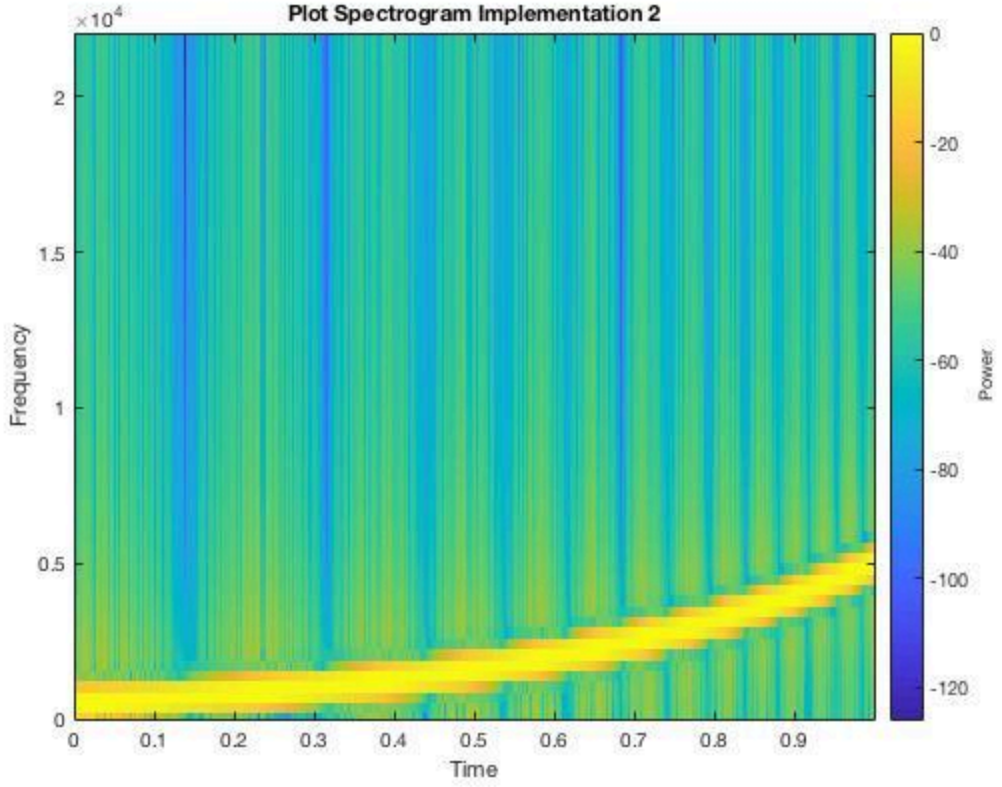

Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-10
-20
-30
-40
-50
-60
-70
-80
-90

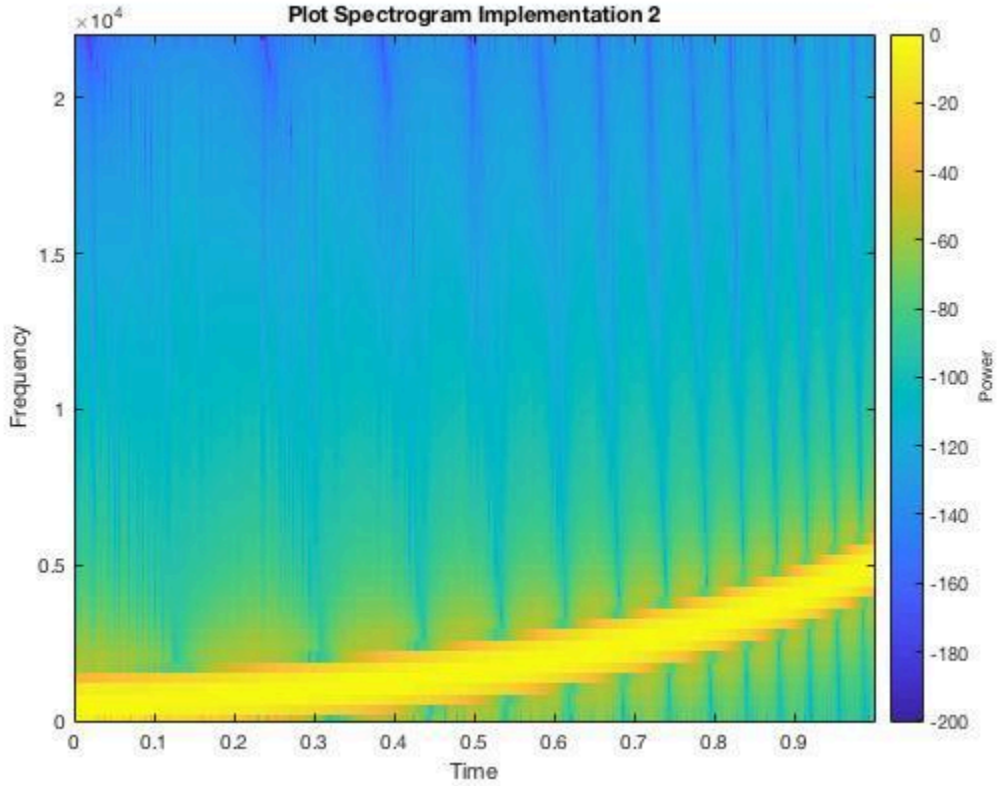


Analysis (D):

MATLAB's Spectrogram Function, where window type = hamming, blackman Window type specifies the shape that the window function takes, which will determine how sideband information will be attenuated, if at all. Will the energies of side band samples be affected by the windows or not. In the hamming window case, we see a thinner sine sweep with more spectral leakage, since the window?s side lobes are larger, attenuating the signals sideband frequencies less. Since the blackman function takes a narrow parabolic shape, far more attenuation occurs, resulting in less spectral leakage and a thicker sine sweep.

```
N = 128;
win_size = N;
hop_size = N/2;
win_type = 'hamming';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);

N = 128;
win_size = N;
hop_size = N/2;
win_type = 'blackman';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);
```


Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-20
-40
-60
-80
-100
-120


Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200

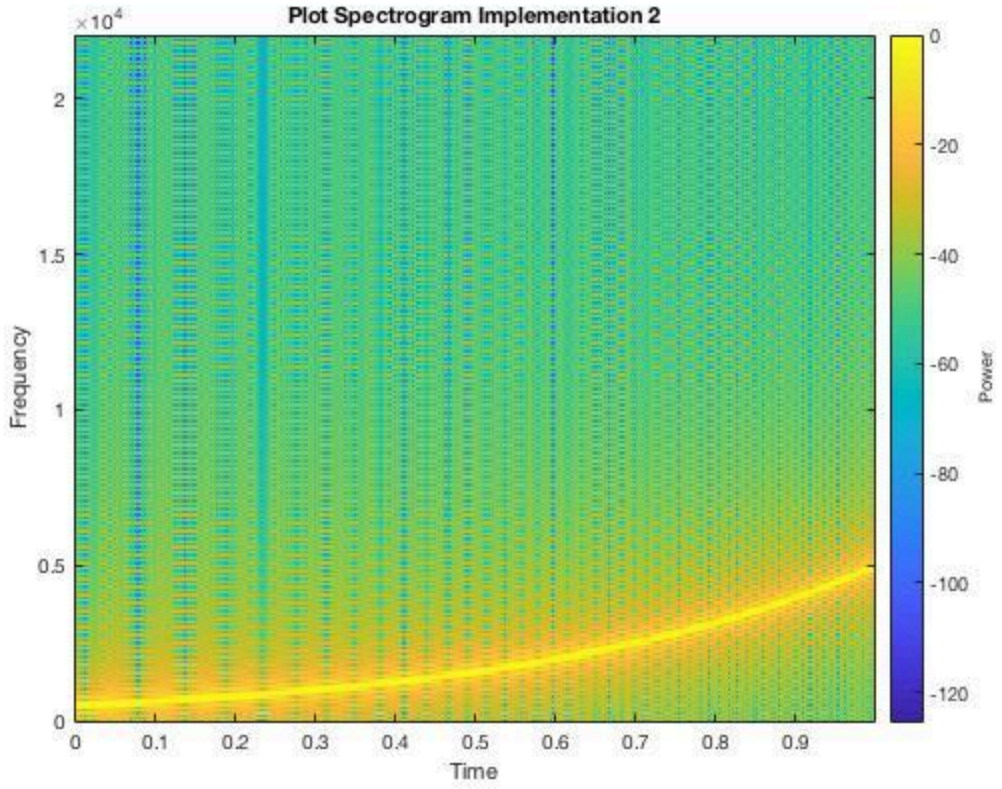
Analysis (E):

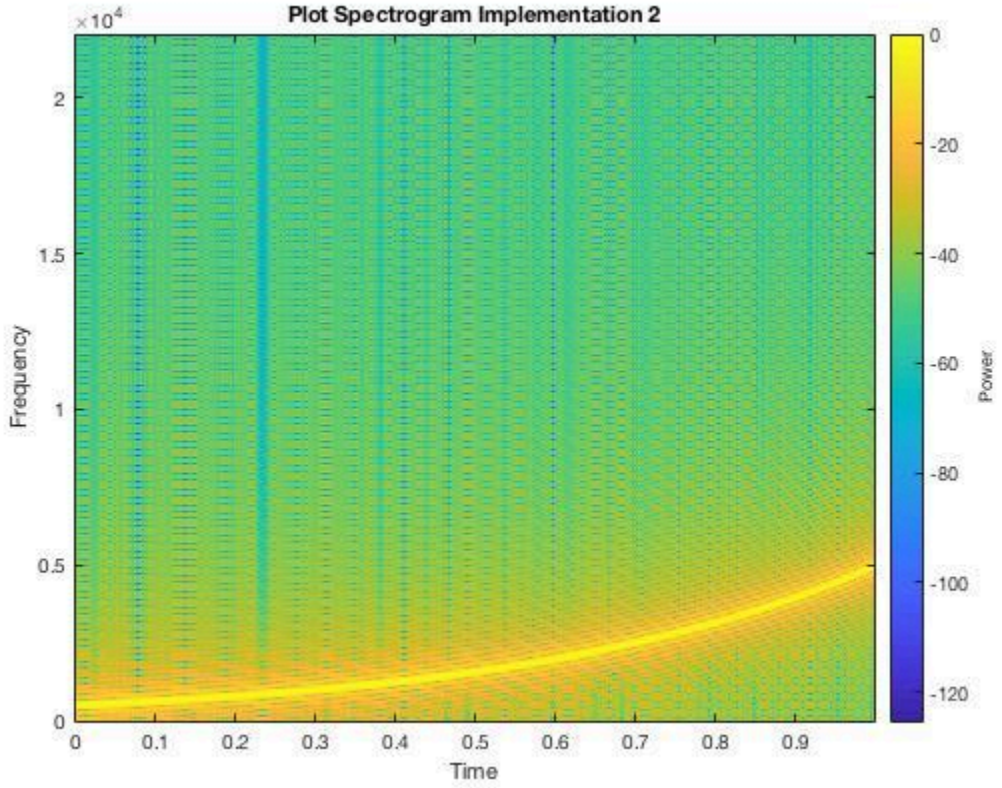
MATLAB's Spectrogram Function, where fft length = 2N, 4N, 8N, where the window size N = 256 FFT length determines frequency resolution and illustrates the effect of zero padding, as nfft exceeds window size, zero padding by the difference. First off, the larger fft size results in greater frequency resolution, per frequency resolution = fs/nfft. Zero padding, were zeros are added to each frequncy bin, further increases frequncy resolution.

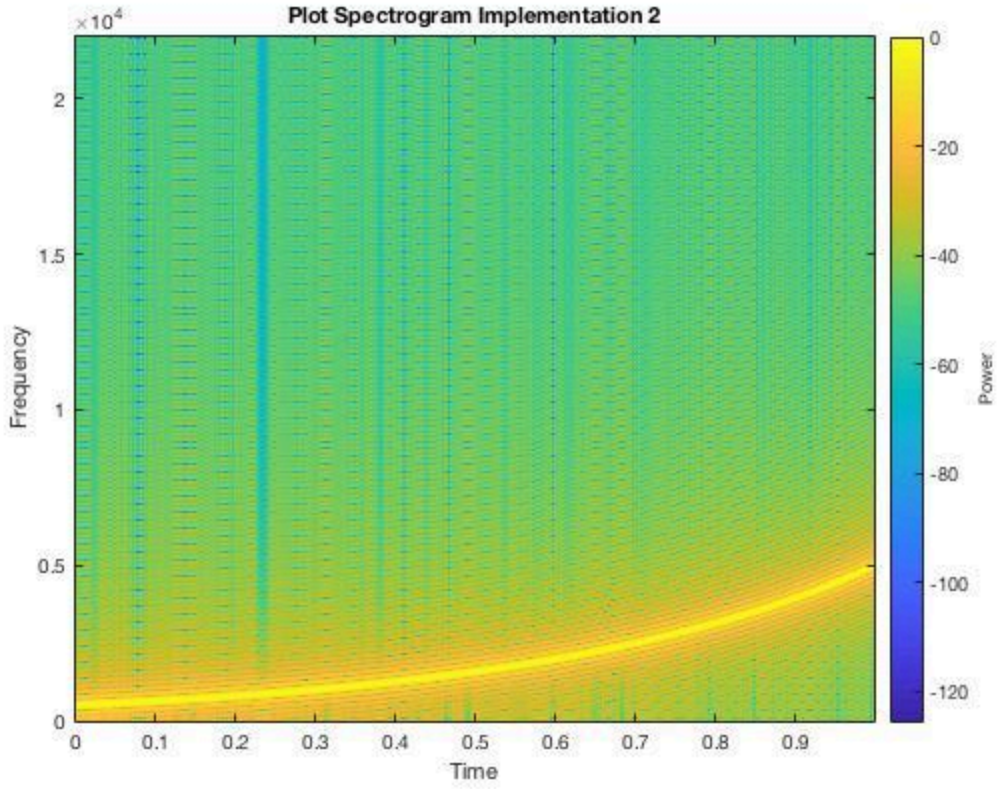
```
N = 256;
win_size = N;
hop_size = N/2;
win_type = 'rect';
nfft = 2*N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);
```

```
N = 256;
win_size = N;
hop_size = N/2;
win_type = 'rect';
nfft = 4*N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);
```

```
N = 256;
win_size = N;
hop_size = N/2;
win_type = 'rect';
nfft = 8*N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);
```


Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-20
-40
-60
-80
-100
-120

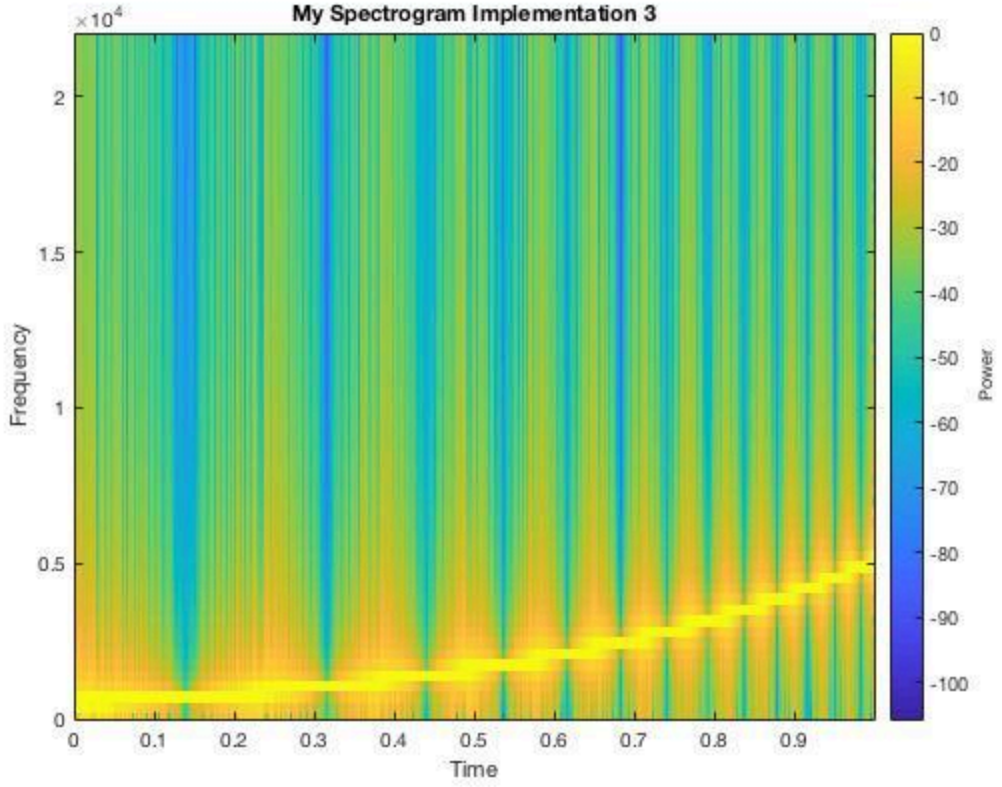

Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-20
-40
-60
-80
-100
-120

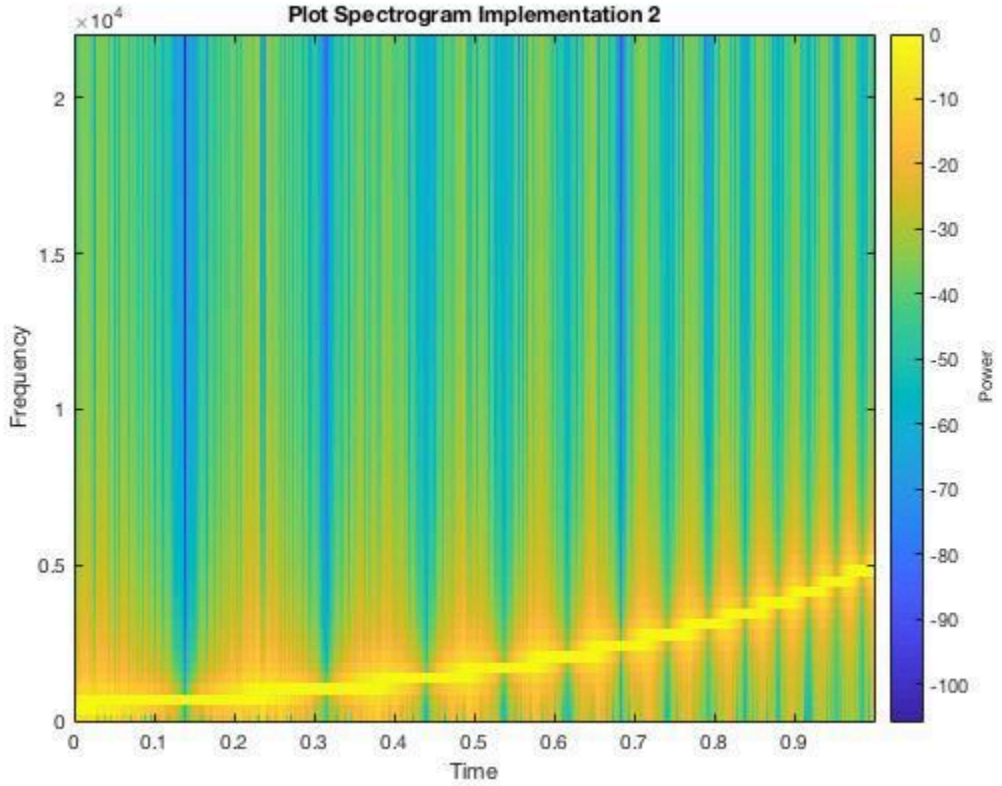


Analysis (F):

MATLAB's Spectrogram Function AND Spectrogram by Matrix Multiplication The default parameters were analyzed here. The results in these two methods are equal.

```
N = 128;
window = rectwin(N);
noverlap = N/2;
nfft = N;
[S, F, T] = my_spectrogram(x_t,window, noverlap, nfft, fs);

win_size = N;
hop_size = N/2;
win_type = 'rect';
nfft = N;
plot_spectrogram(x_t,win_size, hop_size, win_type, fs, nfft);
```


My Spectrogram Implementation 3
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100


Plot Spectrogram Implementation 2
×10^4
Frequency
Time
Power
0
0.5
1
1.5
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100

Published with MATLAB® R2018a