ECE594B

Windowing

April 7, 2021

Student:	Ivan Arevalo
Perm Number:	5613567
Email:	ifa@ucsb.com

Department of Electrical and Computer Engineering, UCSB

0 Motivation

The main objective of this assignment is to study different window functions and their spectral characteristics.

1 Window Functions

Figure 1 shows the 32 point rectangular, sine, hann, and 31 point triangular window in time domain.

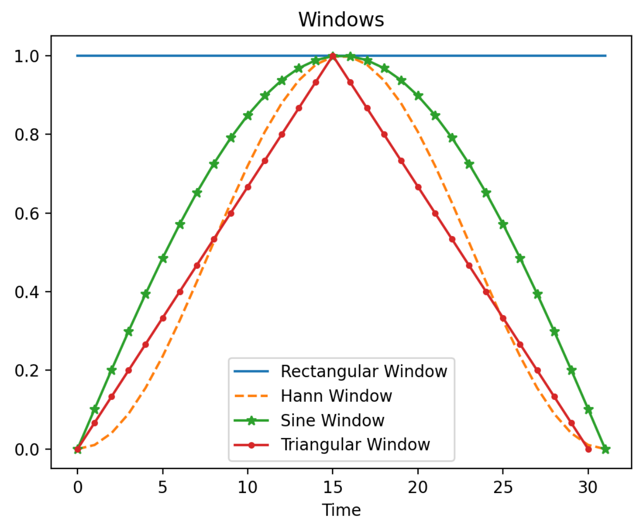

Figure 1: Time-domain windows

We can now evaluate the 1024-point fft of these windows and observe its characteristics.

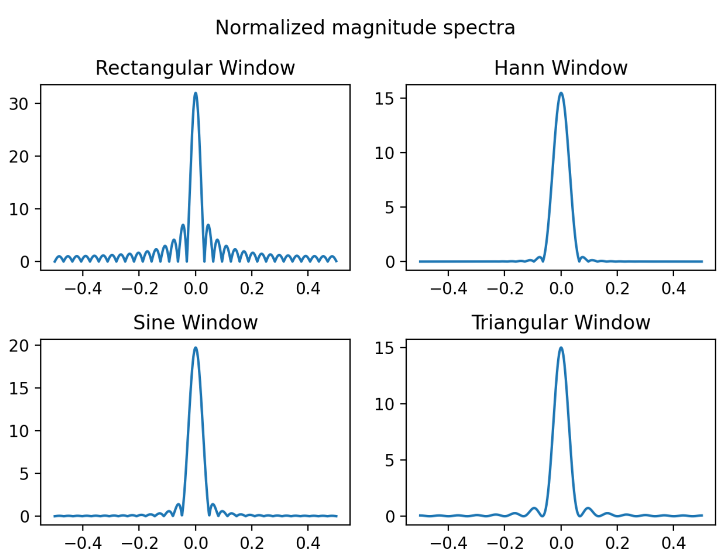

Figure 2: Frequency-domain windows

We can further observe these subtleties by using a logarithmic magnitude scale as shown in figure 3.

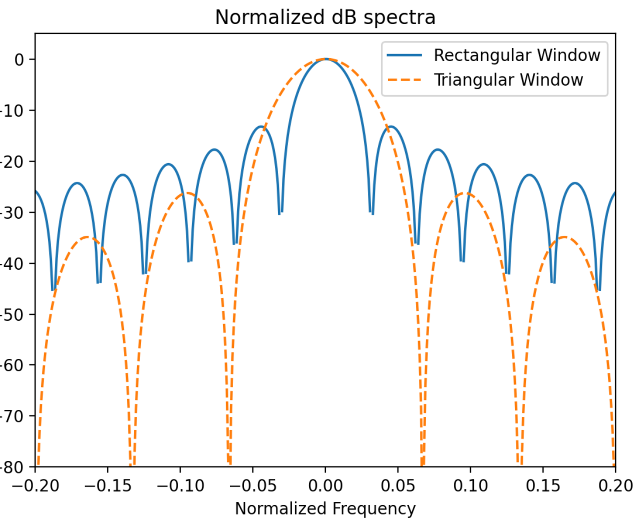

Figure 3: Frequency-domain in dB

Comparing the main lobes of the rectangular and triangular windows, we can observed that the main lobe for the triangular window is twice as wide as that of the rectangular window. We measure the main lobe for the rectangular and triangular window to be about 0.06 and 0.12 respectively. Furthermore, the magnitude of the first side lobe for the rectangular and triangular window is -13.5 dB and -26.3 dB respectively.

Repeating these same steps for the sine and hann windows we arrive at figure 4. Figure 4 shows the frequency-domain for the sine and hann windows.

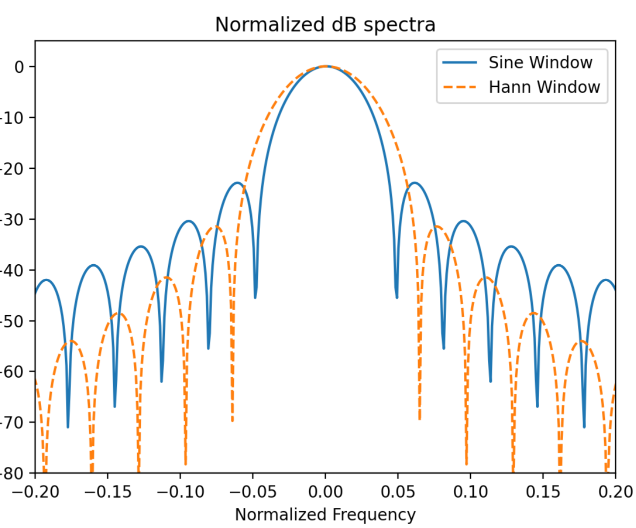

Figure 4: Time-domain windows

We observe that the main lobe of the sine and hann window are about 0.09 and 0.13 respectively, with a first side lobe magnitude of -23.3 dB and -31.9 dB.

Last, we repeat these steps for the Hamming and Blackman windows. Figure 5-7 show the time-domain and spectral plots for these windows.

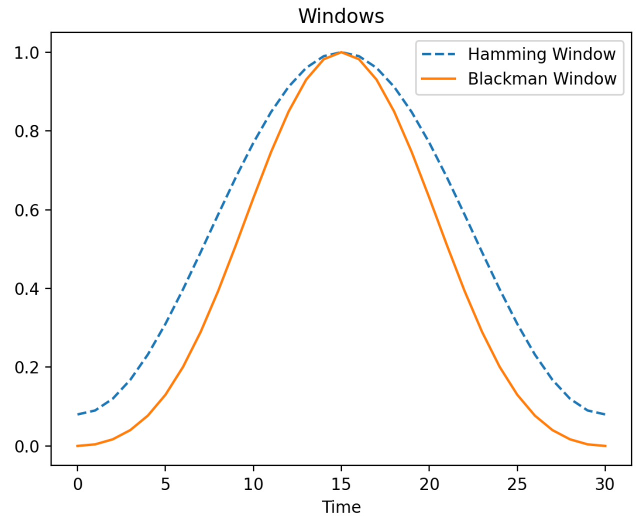

Figure 5: Time-domain windows

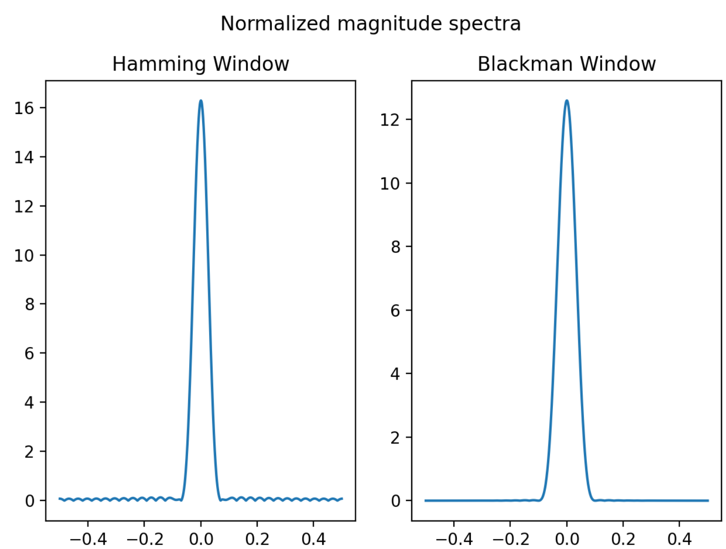

Figure 6: Frequency-domain

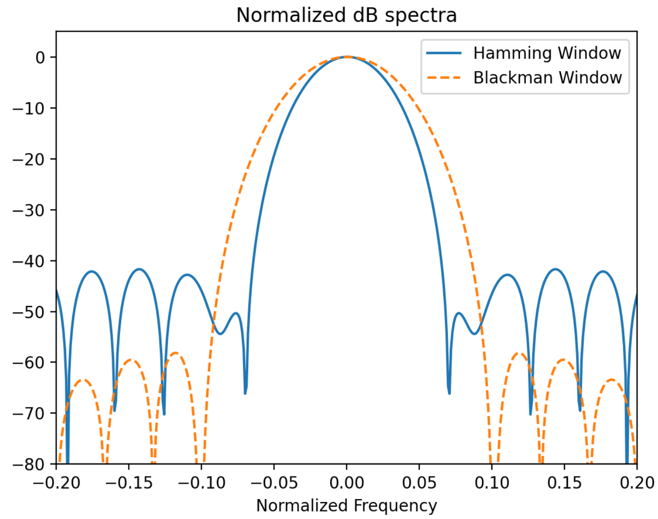

Figure 7: Frequency-domain in dB

Observing figure 7, we can measure the main lobe to be about 0.14 and 0.20 for the hamming and blackman windows respectively. Furthermore, the first side-lobe magnitude is -43 dB and -58 dB for Hamming and Blackman window respectively.

2 Conclusion

We conclude that the rectangular window has the sharpest peak but with highest side lobes and slow decay. The Hann window has a wider main lobe than the Sine window with lowe side lobes. And finally, the blackman window proved to have the widest mainlobe with the lower sidelobes of all windows. The main take-away is that there is always a tradeoff for having lower sidelobes in the form of wider main lobes.

```
import numpy as np
import matplotlib.pyplot as plt


class Window():
    def __init__(self, N):
        self.N = N
        self.window = np.arange(self.N)

    def rectangular_window(self):
        return np.ones(self.N)

    def triangular_window(self):
        window = ( (self.N - 1)/2 - np.abs(self.window - (self.N -1)/2) ) * 2/(self.N -1)
        return window

    def sine_window(self):
        window = np.sin(np.pi * self.window / (self.N -1))
        return window

```

```
    def hann_window(self):
        window = 1/2 * (1 - np.cos(2*np.pi*self.window / (self.N -1) ) )
        return window

    def hamming_window(self):
        window = 0.54 - 0.46 * np.cos(2*np.pi*self.window / (self.N -1) )
        return window

    def blackman_window(self):
        window = 0.42 - 0.5 * np.cos(2*np.pi*self.window / (self.N -1) ) +
            0.08*np.cos(4*np.pi*self.window / (self.N -1) )
        return window


if __name__ == '__main__':

    # Generate windows
    N_odd = 31 # Triangular window_even
    N_even = 32 # All other windows
    window_even = Window(N_even)
    window_odd = Window(N_odd)
    rectangular_window = window_even.rectangular_window()
    hann_window = window_even.hann_window()
    sine_window = window_even.sine_window()
    triangular_window = window_odd.triangular_window()
    hamming_window = window_odd.hamming_window()
    blackman_window = window_odd.blackman_window()

    # Plotting windows on same graph.
    fig = plt.figure()
    plt.plot(rectangular_window, label='Rectangular Window')
    plt.plot(hann_window, '--', label='Hann Window')
    plt.plot(sine_window, '*-', label='Sine Window')
    plt.plot(triangular_window, '.-', label='Triangular Window')
    plt.legend()
    plt.title("Windows"), plt.xlabel('Time')

    fig2 = plt.figure()
    plt.plot(hamming_window, '--', label='Hamming Window')
    plt.plot(blackman_window, label='Blackman Window')
    plt.legend()
    plt.title("Windows"), plt.xlabel('Time')
    plt.show()


    # Plot normalized magnitude spectrum for each window
    num_fft_pts = 1024
    rectangular_window_fft = np.fft.fftshift(np.fft.fft(rectangular_window, num_fft_pts))
    hann_window_fft = np.fft.fftshift(np.fft.fft(hann_window, num_fft_pts))
    sine_window_fft = np.fft.fftshift(np.fft.fft(sine_window, num_fft_pts))
    triangular_window_fft = np.fft.fftshift(np.fft.fft(triangular_window, num_fft_pts))
    hamming_window_fft = np.fft.fftshift(np.fft.fft(hamming_window, num_fft_pts))
    blackman_window_fft = np.fft.fftshift(np.fft.fft(blackman_window, num_fft_pts))
    norm_freq = np.linspace(-0.5, 0.5, num_fft_pts)
```

```
fig3, axs = plt.subplots(2,2)
axs[0, 0].plot(norm_freq, abs(rectangular_window_fft)), axs[0,
    0].set_title('Rectangular Window')
axs[0, 1].plot(norm_freq, abs(hann_window_fft)), axs[0, 1].set_title('Hann Window')
axs[1, 0].plot(norm_freq, abs(sine_window_fft)), axs[1, 0].set_title('Sine Window')
axs[1, 1].plot(norm_freq, abs(triangular_window_fft)), axs[1,
    1].set_title('Triangular Window')
plt.suptitle("Normalized magnitude spectra")
fig3.tight_layout()
plt.show()

fig4, axs = plt.subplots(1, 2)
axs[0].plot(norm_freq, abs(hamming_window_fft)), axs[0].set_title('Hamming Window')
axs[1].plot(norm_freq, abs(blackman_window_fft)), axs[1].set_title('Blackman Window')
plt.suptitle("Normalized magnitude spectra")
fig4.tight_layout()
plt.show()

# Plot normalized dB spectrum of rectangular and triangular windows
fig5 = plt.figure()
plt.plot(norm_freq, 20*np.log10( abs(rectangular_window_fft)/max(
    abs(rectangular_window_fft) ) ), label='Rectangular Window')
plt.plot(norm_freq, 20*np.log10( abs(triangular_window_fft)/max(
    abs(triangular_window_fft) ) ), '--', label='Triangular Window')
plt.legend()
plt.title("Normalized dB spectra"), plt.xlabel('Normalized Frequency')
plt.xlim(-0.2, 0.2), plt.ylim(-80, 5)
plt.show()

# Plot normalized dB spectrum of rectangular and triangular windows
fig6 = plt.figure()
plt.plot(norm_freq, 20 * np.log10(abs(sine_window_fft) / max(abs(sine_window_fft))),
    label='Sine Window')
plt.plot(norm_freq, 20 * np.log10(abs(hann_window_fft) / max(abs(hann_window_fft))),
    '--', label='Hann Window')
plt.legend()
plt.title("Normalized dB spectra"), plt.xlabel('Normalized Frequency')
plt.xlim(-0.2, 0.2), plt.ylim(-80, 5)
plt.show()

# Plot normalized dB spectrum of rectangular and triangular windows
fig7 = plt.figure()
plt.plot(norm_freq, 20 * np.log10(abs(hamming_window_fft) /
    max(abs(hamming_window_fft))), label='Hamming Window')
plt.plot(norm_freq, 20 * np.log10(abs(blackman_window_fft) /
    max(abs(blackman_window_fft))), '--', label='Blackman Window')
plt.legend()
plt.title("Normalized dB spectra"), plt.xlabel('Normalized Frequency')
plt.xlim(-0.2, 0.2), plt.ylim(-80, 5)
plt.show()
```