Investigating the Effects of Windowing on Discrete Fourier Transform Analysis

Coy Zhu

Trinity College

November 16, 2024

Abstract

This report was written as part of the Cambridge University Engineering Part IB Integrated Coursework extended exercise. This report will tackle the effects of windowing in frequency analysis. The report will be structured as follows:

- *DFT theory.*
- *common windows, with their benefits and drawbacks.*
- *analysis of generated data to confirm the research.*

The analysis will involve the following steps:

1. *Generate a linear combination of sine waves of known frequency, i.e.* $A \sin \alpha t + B \sin \beta t$.
2. *Control 1 - DFT of signal measured over an integer number of periods.*
3. *Control 2 - DFT of signal measured over a non-integer number of periods.*
4. *Multiply signal by a window and perform DFT of signal measured over a non-integer number of (pre-windowed) periods.*
5. *Repeat the above step for 2 other windows.*

Using the spectral data acquired from the above steps, features and characteristics will be discussed, and some conclusions will be made.

1 Theory

1.1 Fourier Transforms, Discrete-time Fourier Transforms and Discrete Fourier Transforms

1.1.1 The Continuous Fourier Transform

We make our starting point on this journey the standard Fourier series, which is easy to derive via the concept of frequency bases,

$$x(t) \triangleq \sum_{-\infty}^{\infty} C_n e^{\mathrm{j}n\omega_0 t}, \quad C_n = \frac{1}{T}\int_{-T/2}^{T/2} x(t)e^{-\mathrm{j}n\omega_0 t}\mathrm{d}t, \tag{1}$$

where T is the period of $x(t)$ and $\omega_0 = 2\pi/T$. Firstly, let us derive the Fourier transform. We will do this by taking $\lim_{T\to\infty}$, effectively removing the assumption of periodicity in $x(t)$, and turning $n\omega_0 = \omega$ into a continuous spectrum.

$$C_n = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} x(t)e^{-\mathrm{j}\omega t}\mathrm{d}t = \mathrm{d}\omega \int_{-\infty}^{\infty} x(t)e^{-\mathrm{j}\omega t}\mathrm{d}t, \tag{2}$$

By definition, $C_n \triangleq X(\omega)\mathrm{d}\omega$. This will be justified shortly. As such we arrive at the Fourier transform,

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\cdot e^{-\mathrm{j}\omega t}\mathrm{d}t. \tag{3}$$

This equation has some interesting implications, in particular about periodic functions. Consider $x(t) = Ae^{\mathrm{j}(\omega_1 t+\varphi)}$,

$$X(\omega) = \int_{-\infty}^{\infty} Ae^{\mathrm{j}(\omega_1 t+\varphi)}\cdot e^{-\mathrm{j}\omega t}\mathrm{d}t = Ae^{\mathrm{j}\varphi}\int_{-\infty}^{\infty} e^{\mathrm{j}(\omega_1-\omega)t}\mathrm{d}t = Ae^{\mathrm{j}\varphi}\cdot\delta\left(\omega-\omega_1\right), \tag{4}$$

where $\delta(t)$ is the Dirac delta function. We can confirm that it truly is a delta function by considering the peak complex amplitude of $x(t)$ being where all the individual harmonics superpose "constructively", i.e.

$$x_{\max} = \int_0^{\infty} X(\omega)\mathrm{d}\omega. \tag{5}$$

In this case, since there is only one harmonic, assuming delta function form of Fourier transform, this evaluates to $Ae^{\mathrm{j}\varphi}$, as expected. Building on these foundations, by the theory of Fourier series, any periodic function such that $x(t) = x(t+T)$ can be constructed as a series as shown in equation (1), where $C_n \in \mathbb{C}$ and $\omega_0 = 2\pi/T$. Since the Fourier transform is intrinsically linear, the principle of superposition must apply, giving

$$X(\omega) = \sum_{n=-\infty}^{\infty} C_n\delta\left(\omega - n\omega_0\right). \tag{6}$$

This above analysis justifies the definition $C_n \triangleq X(\omega)\mathrm{d}\omega$.

1.1.2 The Discrete-Time Fourier Transform

All of this above analysis gives us what we expect from the ideal transform on periodic functions: *a series of delta functions, scaled by their complex amplitudes.* However, computing the ideal transform is impossible, since it assumes an infinite density of samples over an infinite period of measurement. Let us make some adjustments. To make the sampling density finite, we introduce a sample rate f_s and angular sample rate $\omega_s = 2\pi f_s$, and a corresponding sampling period $T_s = 1/f_s$. Since we are dealing with only discrete time steps, we will assume that each of our samples correspond to times $t = nT_s$. To model this sample train, we will make use of the delta function to form an impulse train, such that each sample is an impulse.

$$x_s(t) \triangleq \sum_{n=-\infty}^{\infty} x(t)\delta(t - nT_s), \tag{7}$$

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s). \tag{8}$$

By taking the Fourier transform (applying the integration and then using the sifting property of delta trains) of the above equation, we acquire the Discrete-time Fourier Transform,

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} x(nT_s)e^{-\mathrm{j}\omega nT_s}. \tag{9}$$

We can define a discrete sequence of time-domain samples, $x_n = x(nT_s)$,

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-\mathrm{j}\omega nT_s}. \tag{10}$$

Remark: ω is still a continuous variable.
Another key observation to make here is that X_s is periodic in ω with a period of $2\pi/T_s = \omega_s$, which is related to aliasing, which is derived in the appendix.

1.1.3 The Discrete Fourier Transform

In this report, the DFT has been placed after the CFT since it represents the process of going from the ideal, incomputable CFT to the practical DFT. The CFT requires an infinite sampling rate, and an infinite period of measurement, the DTFT only allows for finite sampling rates, whereas DFT is finite in both departments. However, to derive the DFT, we merely discretize the Fourier series.

$$C_k = \frac{1}{T}\int_0^T x(t)e^{-\mathrm{j}k\omega_0 t}\mathrm{d}t, \tag{11}$$

$$X_k = \frac{1}{NT_s}\sum_{n=0}^{N-1} x(nT_s)e^{-\mathrm{j}k\frac{2\pi}{NT_s}nT_s}T_s = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{-\mathrm{j}\frac{2\pi k}{N}n}, \tag{12}$$

where X_k is an approximation of the Fourier transform of x at $\omega = k\omega_s/N$. Intuitively, if we only put in N pieces of information, we cannot acquire more than N pieces of data back out, so it is impossible to work out a full spectrum range. However, due to aliasing, we only need to evaluate the DFT for $0 \le \omega < \omega_s/2$.

1.2 Windows

The DFT assumes that the signal it analyses is periodic, and figuratively wraps the function around from X_{N-1} back to X_0, which can generate a jump discontinuity. Jumps result in lots of high-frequency noise, and spectral leakage, where the associated energy is distributed across the whole frequency spectrum. Window functions help to taper the time-domain signals at either end, by modulating onto the signals. However, if done too sharply, kinks can also cause the same effects as a jump discontinuity. Therefore, there are a variety of smooth and cleverly devised window functions to a) minimise spectral leakage and b) minimise computation complexity. These two features are compromised for the signal being windowed. The amount of spectral leakage can be quantified by the fall-off of side lobes and the width of the main lobe of the CFT of the windows. Greater fall-off and a thinner main lobe is better. Below, the windows being investigated are shown:

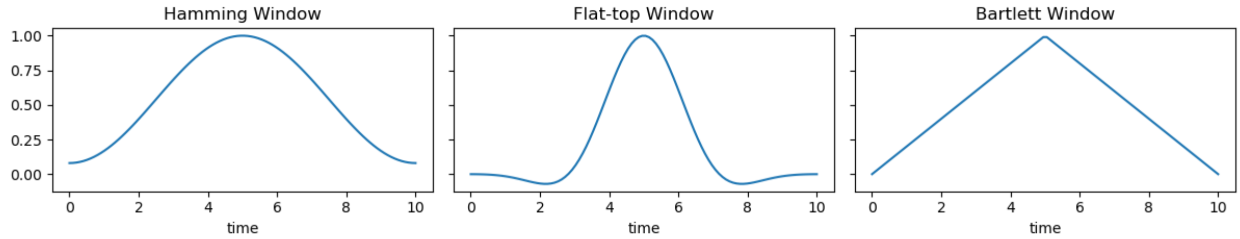

Figure 1: Three different windows.

The equations for each of the three windows are listed below for $n \in [0, N]$

$$x_{\text{Hamming},n} = a_0 + (1 - a_0)\cos\left(\frac{2\pi n}{N}\right) \tag{13}$$

$$x_{\text{Flat-top},n} = a_0 - a_1\cos\left(\frac{2\pi n}{N}\right) + a_2\cos\left(\frac{4\pi n}{N}\right) - a_3\cos\left(\frac{6\pi n}{N}\right) + a_4\cos\left(\frac{8\pi n}{N}\right) \tag{14}$$

$$x_{\text{Bartlett},n} = 1 - \left|\, \frac{2n}{N} - 1 \,\right| \tag{15}$$

The Hamming window is known for having a thin main lobe, and very good first side-lobe fall-off. This is shown below. The Flat-top compromises the Hamming window's main lobe width for a taller peak. Finally, the Bartlett window has easy computation, but has the widest main lobe and the least side-lobe fall-off, so should have the worst spectral leakage.

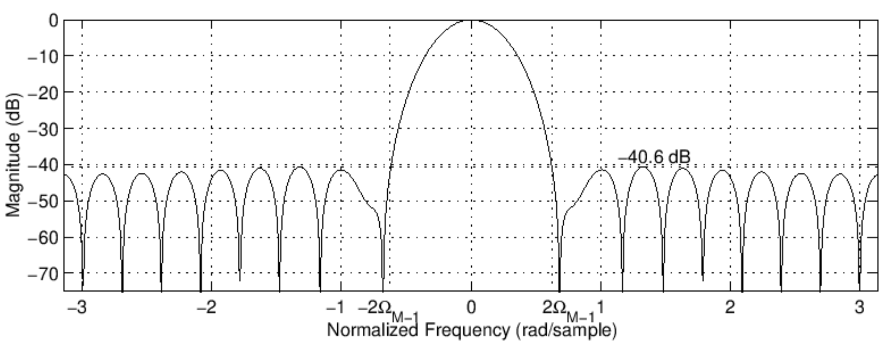

Figure 2: Hamming Window Frequency Spectrum

2 Analysis

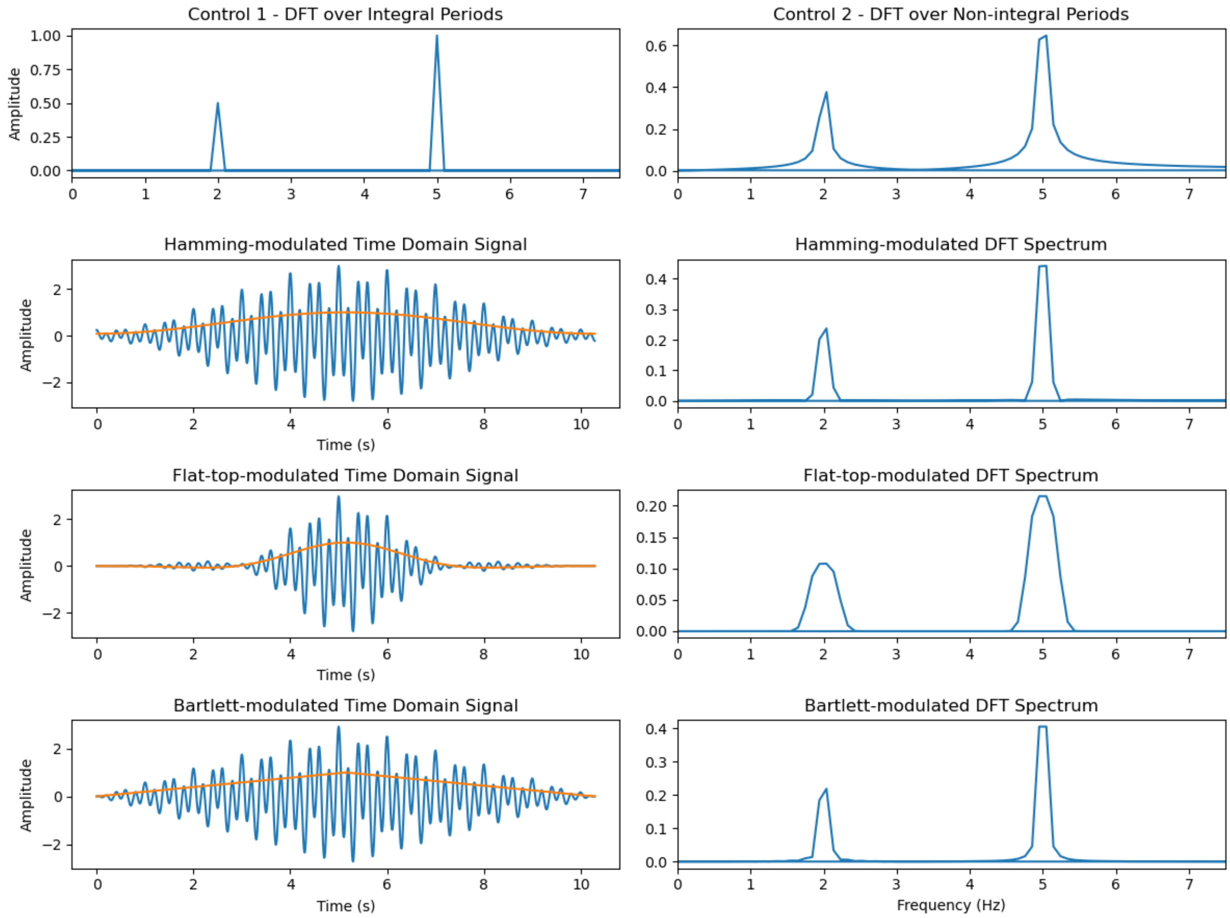

Figure 3: Results. The input signal is $x(t) = 2\cos(10\pi t) + \cos(4\pi t)$.

From the graphs, we can see in the top left, Control 1 gives us effectively a perfect DFT for the input signal with minimum spectral leakage, since x_0 and x_{N-1} were designed to have no jump discontinuities. Note the amplitudes are halved because there are also negative frequency peaks, which have been omitted. Control 2 adds in a jump discontinuity, and the effects are immediately obvious, with non-zero values "leaking" away from the true frequency peaks. Also, note that the frequency peaks have fallen.

The Hamming-modulated signal seems to perform the best, causing the fastest signal drop-off either side of the frequency peaks, minimising spectral leakage. On the other hand, the Flat-top-modulated signal forms very wide peaks as expected but does not improve the peak height, going against literature. The Bartlett-modulated signal sits in a middle ground between the other two, which is also unexpected.

It should be noted that all three windowed signals concentrate energy towards the peaks and reduce the energy spread across the frequency spectrum, relative to Control 2, demonstrating the benefits of windowing. In addition, the relative amplitudes of the input signal frequency components were preserved, though their values fell, since window-modulating reduces average time-domain amplitude.

Performing DFTs on window functions can create the illusions of no spectral leakage depending on phase, i.e. all data points lie in troughs, so has been omitted from this short report.

3 Appendix

3.1 Aliasing and Nyquist Frequency

Since the unit impulse train is periodic with T_s, we may represent it as a Fourier series, which is given by

$$\sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} e^{\mathrm{j}\frac{2n\pi}{T_s}t} = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} e^{\mathrm{j}n\omega_s t} \tag{16}$$

Here, let us consider the sample sequence as a modulated impulse train,

$$\implies x_s(t) = \sum_{n=-\infty}^{\infty} x(t)\delta(t - nT_s) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} x(t)e^{\mathrm{j}n\omega_s t}. \tag{17}$$

Now let us compute the Fourier transform of the above equation,

$$X_s(\omega) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s), \tag{18}$$

i.e. X_s is obtained by replicating X every ω_s in the frequency domain. This gives rise to aliasing and is why the Nyquist frequency is the highest frequency that can be represented in a discretised Fourier transform.

3.2 Python Code

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

omega1 = 2*np.pi*5
omega2 = 2*np.pi*2
f_s = 100
dt = 1 / f_s

# Control 1
tCtrl1 = np.arange(0, 10, dt)
N1 = 10/dt
xCtrl1 = 2 * np.cos(omega1*tCtrl1) + np.cos(omega2*tCtrl1)
fCtrl1 = np.fft.fftfreq(tCtrl1.size, d = dt)
XCtrl1 = np.fft.fft(xCtrl1)/N1

# Control 2
tCtrl2 = np.arange(0, 10.3, dt)
N2 = 10.3/dt
xCtrl2 = 2 * np.cos(omega1*tCtrl2) + np.cos(omega2*tCtrl2)
fCtrl2 = np.fft.fftfreq(tCtrl2.size, d = dt)
XCtrl2 = np.fft.fft(xCtrl2)/N2
```

```
# Hamming Window
hammingWind = np.hamming(tCtrl2.size)
xHamming = xCtrl2 * hammingWind
XHamming = np.fft.fft(xHamming)/N2
#XHamWind = np.fft.fft(hammingWind)
# Flat-top Window
flattopWind = scipy.signal.windows.flattop(tCtrl2.size)
xFlattop = xCtrl2 * flattopWind
XFlattop = np.fft.fft(xFlattop)/N2
# Bartlett Window
bartlettWind = np.bartlett(tCtrl2.size)
xBartlett = xCtrl2 * bartlettWind
XBartlett = np.fft.fft(xBartlett)/N2

# Display
fig, ax = plt.subplots(4, 2, figsize = (12,9))

ax[0][0].plot(fCtrl1, np.abs(XCtrl1))
ax[0][0].set_ylabel("Amplitude")
ax[0][0].set_title("Control 1 - DFT over Integral Periods")
ax[0][0].set_xlabel("Frequency (Hz)")
ax[0][0].set_xlim(0, 7.5)
ax[0][1].plot(fCtrl2, np.abs(XCtrl2))
ax[0][1].set_title("Control 2 - DFT over Non-integral Periods")
ax[0][1].set_xlim(0, 7.5)
ax[1][0].plot(tCtrl2, xHamming, tCtrl2, hammingWind)
ax[1][0].set_ylabel("Amplitude")
#ax[1][0].set_xlabel("Time (s)")
ax[1][0].set_title("Hamming-modulated Time Domain Signal")
ax[1][1].plot(fCtrl2, np.abs(XHamming))
ax[1][1].set_title("Hamming-modulated DFT Spectrum")
ax[1][1].set_xlim(0, 7.5)
ax[2][0].plot(tCtrl2, xFlattop, tCtrl2, flattopWind)
ax[2][0].set_ylabel("Amplitude")
#ax[2][0].set_xlabel("Time (s)")
ax[2][0].set_title("Flat-top-modulated Time Domain Signal")
ax[2][1].plot(fCtrl2, np.abs(XFlattop))
ax[2][1].set_title("Flat-top-modulated DFT Spectrum")
ax[2][1].set_xlim(0, 7.5)
ax[3][0].plot(tCtrl2, xBartlett, tCtrl2, bartlettWind)
ax[3][0].set_ylabel("Amplitude")
ax[3][0].set_xlabel("Time (s)")
ax[3][0].set_title("Bartlett-modulated Time Domain Signal")
ax[3][1].plot(fCtrl2, np.abs(XBartlett))
ax[3][1].set_xlabel("Frequency (Hz)")
ax[3][1].set_title("Bartlett-modulated DFT Spectrum")
ax[3][1].set_xlim(0, 7.5)

fig.tight_layout()
fig.show()
```