Digital Signal Processing Course Project Frequency Spectrum Analysis Based on DFT

Zexi Huang[*]
Yingcai Honors College

December 25, 2016

Abstract

Spectral analysis is an important application of digital signal processing. Here, we present a digital spectrum analysis scheme that clearly identifies the frequency components of a linear combination of sinusoids with relatively close frequencies but distinct amplitudes. In an effort to obtain the best resolution as well as saving computational efforts, the choices of the type and length of window function and length of DFT are discussed in details and a special method by avoiding spectrum leakage is highlighted.

1 Introduction

Determining in the discrete-time domain the frequency contents of a continuous-time signal, more commonly known as spectral analysis, is widely used in signal procssing and analysis. The rationale is that it is very common for information to be encoded in the sinusoids that form a signal, no matter whether it is naturally occurring or created by human. In this case, the shape of the time domain waveform is of little importance, while the frequency, phase and amplitude of the component sinusoids are of interest. The general procedure for spectral analysis is shown in Figure 1.

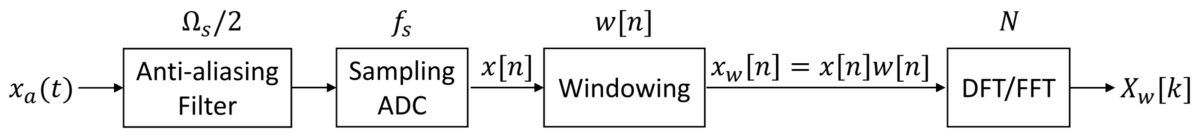

Figure 1: Block diagram of spectral analysis.

Here, we simulate the process of spectral analysis with MATLAB for a signal composed of sinusoids with relatively close frequencies but distinct amplitudes. As we will see later, for such a signal, the type and length of the window function as well as the length of DFT in the

[*]Correspondence should be addressed to Z. Huang. E-mail: Eitima@163.com. Student No. 2014030302014

procedure is of great significance to the final results, as the sinusoids with small amplitudes can be easily submerged in the side-lobes of sinusoids with much bigger amplitudes.

The remainder of this report is organized as follows: In the following section, we illustrate the procedure of spectral analysis step by step. Section 3 presents the numerical experiments for a specific signal. Finally, we give a short discussion and conclude in Section 4.

2 Problem Formulation

2.1 Anti-aliasing Filter, Sampling and Analog-to-Digital Converter

Before a real-life analog filter can be processed by a digital spectrum analyzer, it must be converted to discrete form. Anti-aliasing filter, sampling module and analog-to-digital converter are used for that purposes. Since we focus on computer-based simulation of the process, the signal is considered without noise and what we need to ensure is the sampling frequency, which follows Nyquist-Shannon sampling theorem [1]:

$$f_s > 2f_H \tag{1}$$

where f_s denotes the sampling frequency and f_H is the highest frequency that exist in original signal $x_a(t)$.

The output of these modules is the discretized signal $x[n]$, which is ready for further digital processing.

2.2 Windowing

Theoretically speaking, $x[n]$ can now be transformed into its frequency domain and the spectrum can be readily available now. However, in practice, several considerations discourage us from doing so:

- Computational capability constrains. The real-life signal can be infinitely long. Finding the DFT of such signal is computationally impossible.
- Memory storage constrains. For a very long signal, the spatial cost for storaging the complete set of data is overwhelming.
- Real-time requirements. For an incessant signal, often, the spectrum of interest is a short period of it. Instead of waiting for the whole signal to be transmitted, we need real-time spectrum of it.

Windowing is a method to solve these problems. By multiplying the original signal $x[n]$ with a causal and finite-length window function $w[n]$,

$$w[n] = \begin{cases} w_s[n], 0 \leqslant n \leqslant M-1 \\ 0, \text{otherwise} \end{cases} \tag{2}$$

we have

$$x_w[n] = x[n]w[n] \tag{3}$$

which is also finite-length and causal. Of course, its DTFT,

$$X_w(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})W(e^{j(\omega-\theta)})e^{j\theta}d\theta \neq X(e^{j\omega}) \tag{4}$$

as $W(e^{j\omega})$ can't be perfectly rectangular. To illustrate, some of the common windon functions and their gain responses are shown in Figure 2.

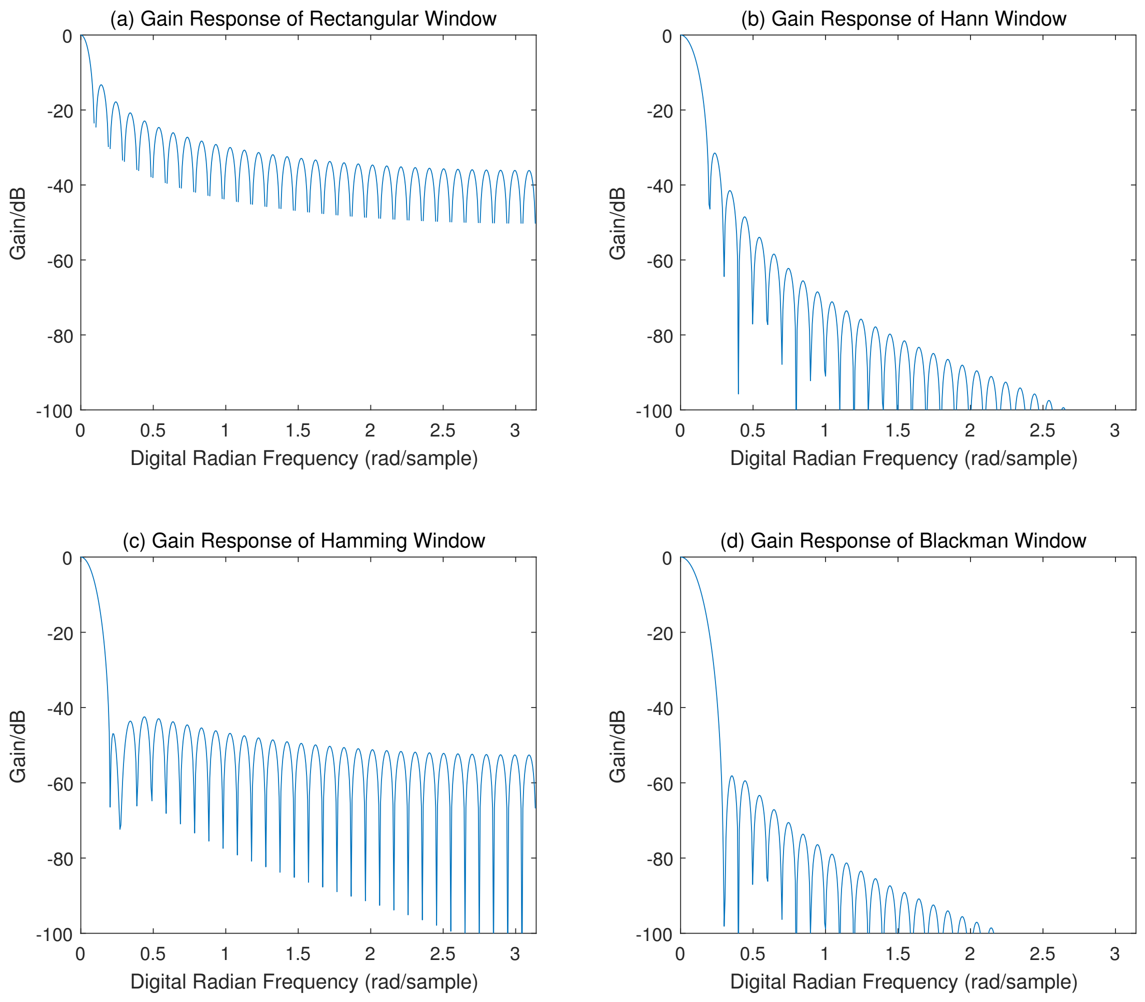

Figure 2: Gain response of some common windows ($M = 64$). (a) Rectangular window ($w_s[n] = 1$). (b) Hann window ($w_s[n] = 0.5[1 + \cos(\pi(n - 0.5M)/M)]$). (c) Hamming window ($w_s[n] = 0.54 + 0.46\cos(\pi(n - 0.5M)/M)$). (d) Blackman window ($w_s[n] = 0.42 + 0.5\cos(\pi(n - 0.5M)/M) + 0.08\cos(2\pi(n - 0.5M)/M)$).

Depending on the properties of the window functions choosen, the output spectrum may distort to different extent. Since our objective is to identify all the frequency components in a linear combination of sinusoids with close frequencies but distinct amplitudes, the key properties of interest are the main lobe width Δ_{ML} and relative sidelobe level A_{sl}. Δ_{ML} is critical to whether we could identify components with close frequencies. Large Δ_{ML} leads to component with a large amplitude surpresses its nearby weak component, thus a small Δ_{ML} is desired. For A_{sl}, the smaller it is, the more easier for the component with relatively small amplitude to be detected. However, as shown in Table 1, for a fixed M, thses two desires can't be achieved at the same time, indicating a trade-off must be made. On the other hand, for a certain type of window, incresing its length leads to smaller main lobe width, but requires more computational efforts.

Table 1: Properties of some common windows.

Types of Window	Main Lobe Width Δ_{ML}	Relative Sidelobe Level A_{sl}
Rectangular	$4\pi/M$	$13.3dB$
Hann	$8\pi/M$	$31.5dB$
Hamming	$8\pi/M$	$42.7dB$
Blackman	$12\pi/M$	$58.1dB$

Specially, if the signal under consideration only consists of several sinusoids signals with frequencies rational numbers, whose ideal frequency response are pulses, their sidelobes can be completely surpressed if there is no spectrum leakage (see next subsection). Under this circumstance, rectangular window is far superior to other windows since it doesn't distort the signal in time domain, resulting the ideal pulse frequency responses.

2.3 Discrete Fourier Transform

After appropriate windowing, $x_w[n]$ is ready to be to tranformed into its frequency domain. DSP modules implements this by using FFT (**F**ast **F**ourier **T**ransform, [2]) or other popular algorithms like Goertzel's [3]. No matter what methods are used, the length of the FFT N is of great importance. Distortion or misrepresentation can happen due to spectrum leakage and picket fence effect of inapproriate choice of length.

In general, longer length leads to better results since it can cover more points in frequency domain and has higher possibility of identifying peak but more computational efforts.

As mentioned in previous subsection, for signal consisting of only rational-valued frequency sinusoids, avoiding the spectrum leakage directly leads to total surpression of sidelobes, thus identifying the frequency components without additional computational efforts.

To be specific, consider an sampled sinusoid with rational frequency $f_0 = \frac{a}{b}$,

$$s[n] = \sin(\frac{2\pi f_0}{f_s}n) = \sin(\frac{2\pi a}{bf_s}n), 0 \leqslant n \leqslant M-1 \tag{5}$$

Its period is given by

$$P = b \times \text{numerator of } \frac{f_s}{a} \text{after reduction} \tag{6}$$

Since DFT of a signal can be viewed as the evaluation of DFS for the periodic form (with period N) of that signal, by choosing

$$N = kP \tag{7}$$

The frequency response would become an ideal pulse since the periodic form of $s[n]$, $s_p[n]$, is

$$s_p[n] = \sin(\frac{2\pi f_0}{f_s}n), -\infty \leqslant n \leqslant +\infty \tag{8}$$

Recall that if sidelobes exist, we need to increse M in order to decrease the main lobe width. But in this case, N can be surprisingly small and M is only required to be no less than N. Therefore a huge amount of computational efforts can be saved.

3 Numerical Experiments

3.1 Problem and Enviroment

In numerical experiments, the signal under consideration is

$$\begin{aligned} x(t) = 10\sin(2\pi \times 64t) + \sin(2\pi \times 250\frac{1}{3}t) + 20\sin(2\pi \times 256t) \\ +3\sin(2\pi \times 260t) + 10\sin(2\pi \times 512t) \end{aligned} \tag{9}$$

which is composed with five frequency components at $64Hz$, $250\frac{1}{3}Hz$, $256Hz$, $260Hz$, $512Hz$, with respective amplitudes 10, 1, 20, 3, 10. The challenge is to correctly identify the $250\frac{1}{3}Hz$, $256Hz$ and $260Hz$ components. Their frequencies are close while their amplitudes, 1, 20, 3 are distinct.

The simulation environment for the whole process is MATLAB 9.0.0.341360 (R2016a), and all sources codes used in this simulation and listed in the appendix are of that syntax. This includes the spectrum analyzer given window type, window length and DFT length *SpectrumAnalyzer.m* and several other auxiliary functions.

3.2 Sampling

Since the highest frequency component in $x(t)$ is $512Hz$,

$$f_s > 1024Hz \tag{10}$$

Here, $f_s = 1200Hz$ is arbitrarily choosen for simplicity. The sampled signal $x[n]$ is

$$x[n] = 10\sin(\frac{8\pi}{75}n) + \sin(\frac{751\pi}{1800}n) + 20\sin(\frac{32\pi}{75}n) + 3\sin(\frac{13\pi}{30}n) + 10\sin(\frac{64\pi}{75}n) \tag{11}$$

3.3 Avoiding Spectrum Leakage

Noting that $x[n]$ is composed of sinusoids with rational-valued frequencies, we consider the method of avoiding spectrum leakage with rectangular window and $M = N$ for sake of computational efforts. Denote the five sinusoids in Equation 11 as $x_1[n]$, $x_2[n]$, $x_3[n]$, $x_4[n]$ and $x_5[n]$ respectively. Their periods are computed as

$$P_1 = 75, P_2 = 3600, P_3 = 75, P_4 = 60, P_5 = 75 \tag{12}$$

and the period for the signal

$$P = LCM\{P_1, P_2, P_3, P_4, P_5\} = 3600 \tag{13}$$

An naive idea is to choose the length of DFT $N = 3600$. In this case, the spectrum leakage of the whole signal would be avoided, resulting ideal pulse responses for all frequency components, as shown in Figure 3.

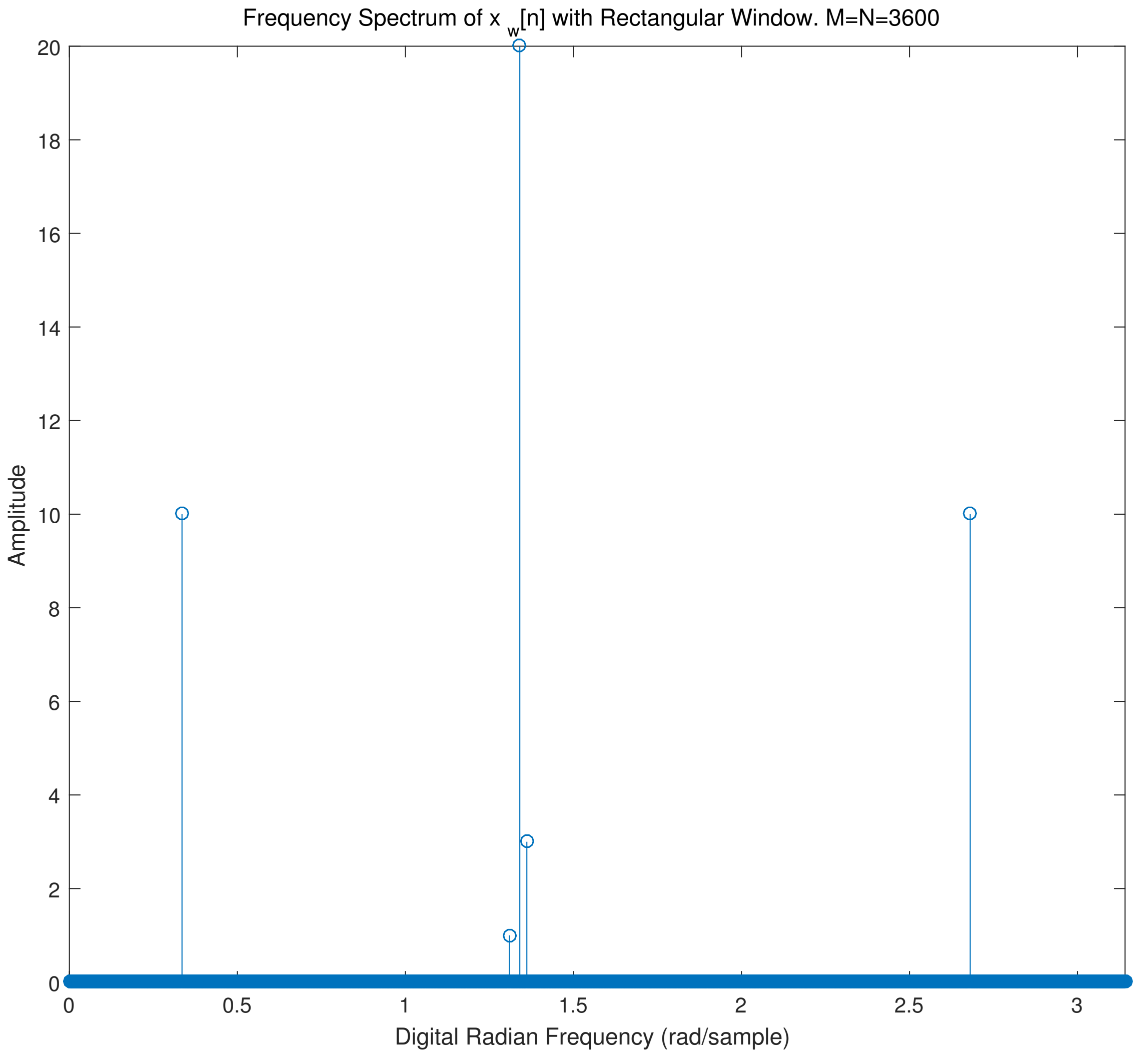

Figure 3: Frequency spectrum for $x_w[n]$ with rectangular window. $M = N = 3600$.

However, with such big values for M and N, other window functions also yields acceptable results as show in Figure 4.

Therefore, to show superiority of the avoiding spectrum leakage method, we must try to shorten the length. If we loosen our restrict on periodicity, using

$$N = P' = LCM\{P_1, P_3, P_4, P_5\} = 300 \tag{14}$$

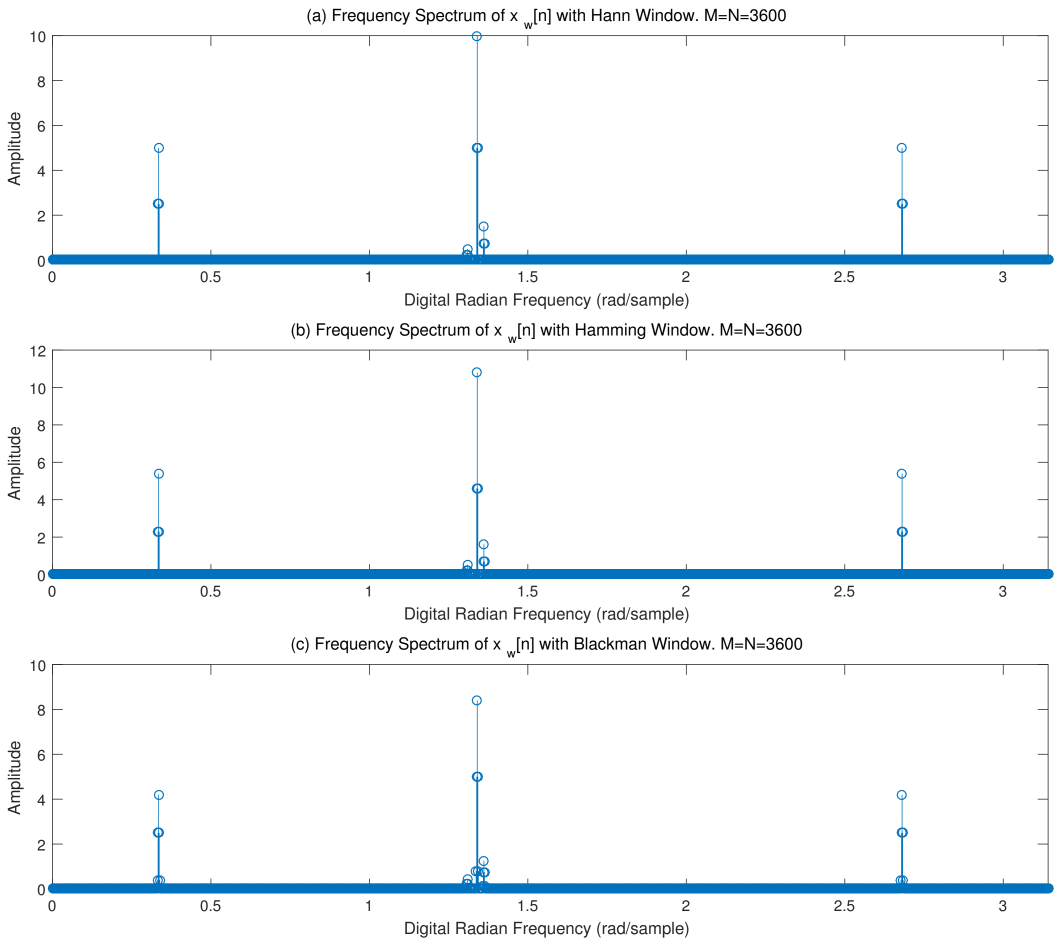

Figure 4: Frequency spectrum for $x_w[n]$ with (a) Hann window, (b) Hamming window, and (c) Blackman window. $M = N = 3600$.

although $x_2[n]$ would be nonperiodic, the effect of its sidelobe is limited as it has the smallest amplitude compared to other component. The result is shown in Figure 5. As explained, only $x_2[n]$ introduces its sidelobe due to spectrum leakage. And it is unnecessary to argue that the peak of $x_2[n]$ may arise from the sidelobe of other components since there is no such thing at all. Another observation that further confirms the successful detection is that the amplitudes of the two stems near that of $x_3[n]$, are of a ratio approximately $1 : 3$, which is consistent with $x_2[n]$ vs. $x_4[n]$ in Equation 11.

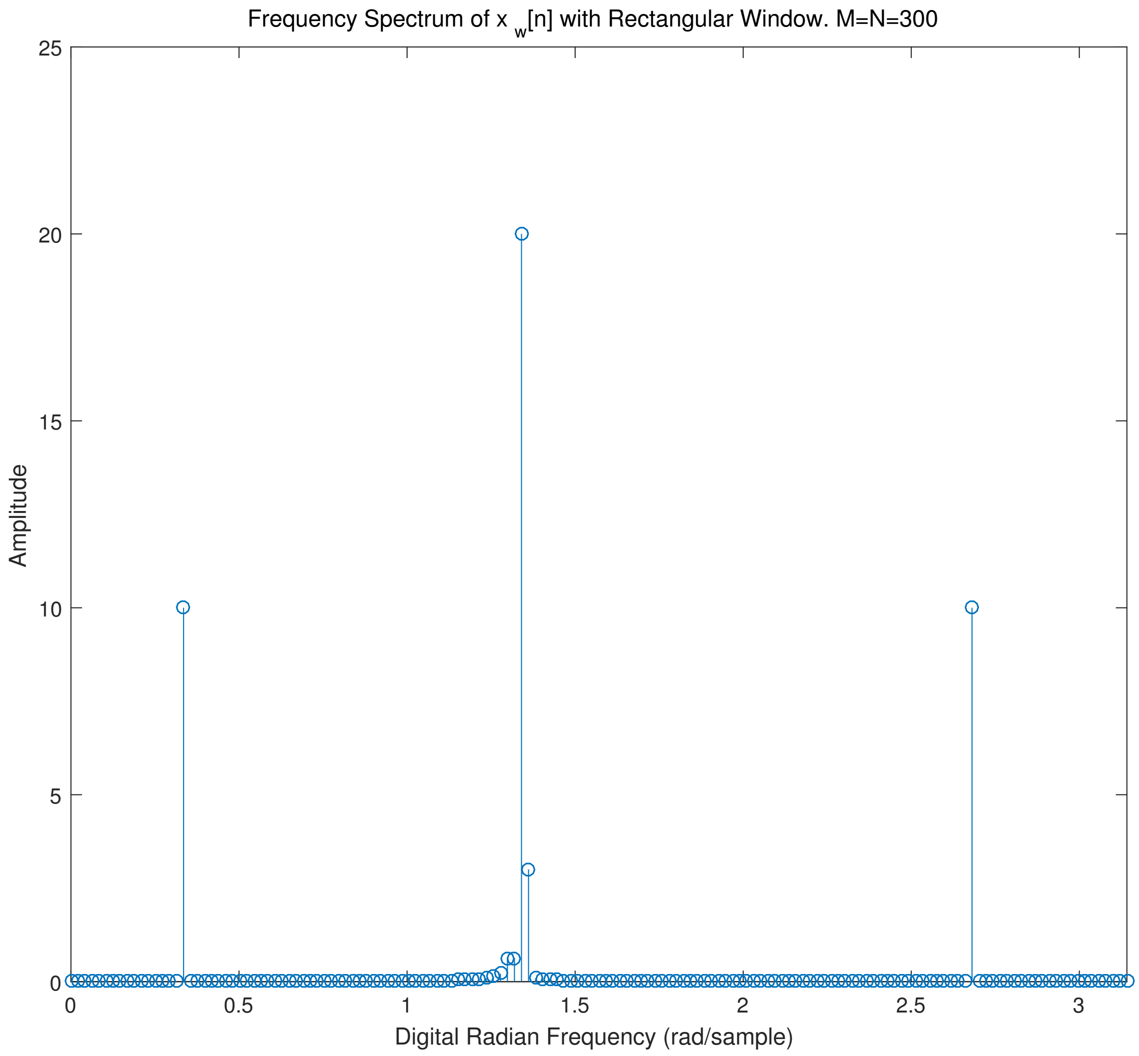

Figure 5: Frequency spectrum for $x_w[n]$ with rectangular window. $M = N = 300$.

This time, other window functions face complete failures in resolving $x_2[n]$, $x_3[n]$ and $x_4[n]$, as shown in Figure 6. Their window lengths are not sufficiently big, resulting in full concealment of $x_2[n]$ and $x_4[n]$.

A more ambitious idea is to let

$$N = P_3 = 75 \tag{15}$$

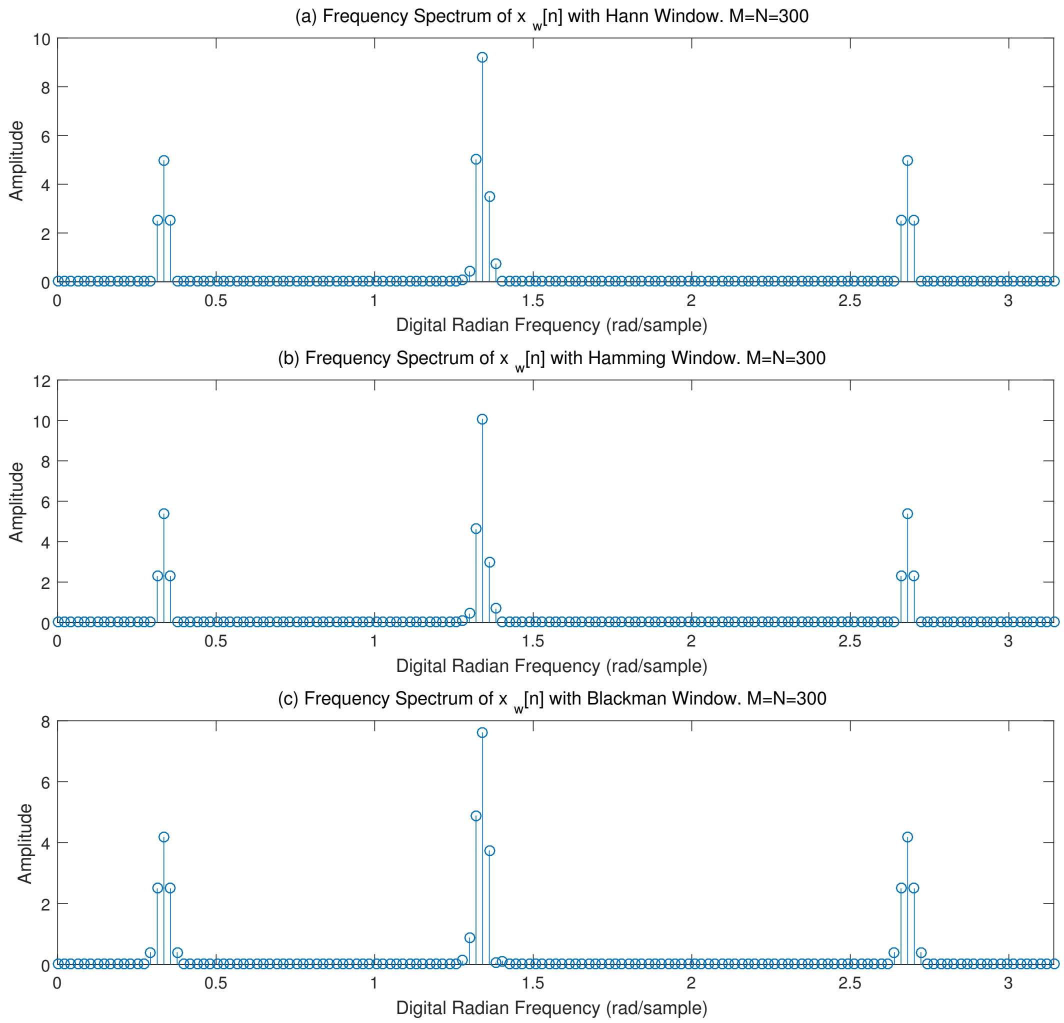

Figure 6: Frequency spectrum for $x_w[n]$ with (a) Hann window, (b) Hamming window, and (c) Blackman window. $M = N = 300$.

which only avoid the spectrum leakage of $x_3[n]$ (as well as $x_1[n]$ and $x_5[n]$, but they are of little importance as to the concealment of $x_2[n]$ and $x_4[n]$). The results are undesirable, however. Since there is no guarantee for spectrum leakage avoidance of $x_2[n]$ and $x_4[n]$, picket fence effect comes in as limted DFT samples are unable to detect the mainlobe peak. In addition, sidelobes of $x_2[n]$ and $x_4[n]$ would cover each other, resulting in aliasing spectrum and concealment for both, althouth $x_3[n]$ renders no influence on them at all. It's the same case when $M = N = 150$. Both cases are illustrated in Figure 7.

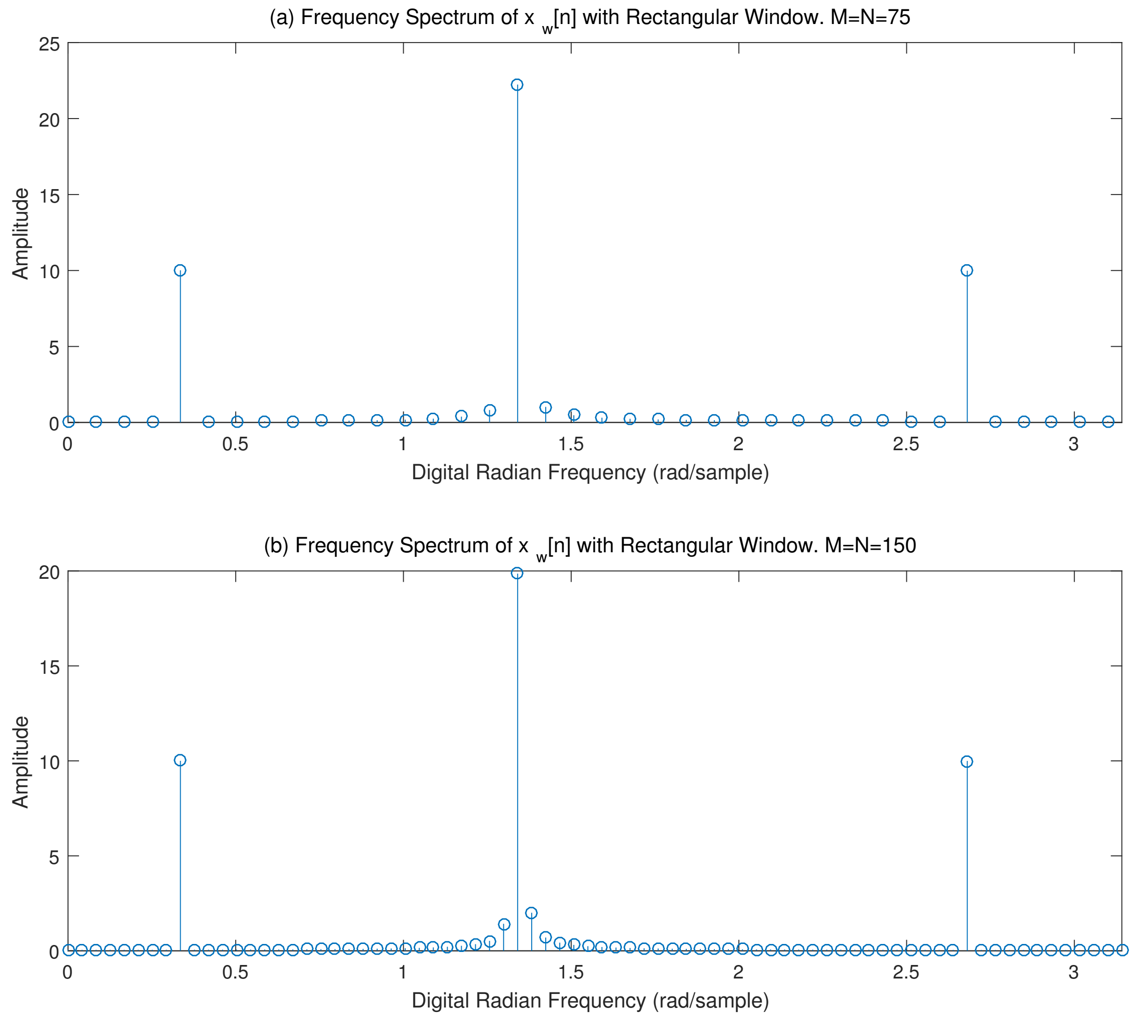

Figure 7: Frequency spectrum for $x_w[n]$ with rectangular window. (a) $M = N = 75$, (b) $M = N = 150$.

Therefore, for the signal given in Equation 9, rectangular window with $M = N = 300$ is the best option consiering both resolution and computational costs.

3.4 General Case

In practice, the exact frequency of each component may be a long float number. Under such circumstance, perfect spectrum avoidance is unavailable (the period P can be enormous). We simulate this process by letting N not equal to any multiple of P_1, P_2, P_3, P_4 or P_5. The test-and-check process is tedious and neglected here. The final test result is that the rectangular window with $M = N = 580$ may be a threshold between clear dectection with right amplitude ratio and right frequency location and failure. The result is shown in Figure 8, with comparision to other windows under same condition.

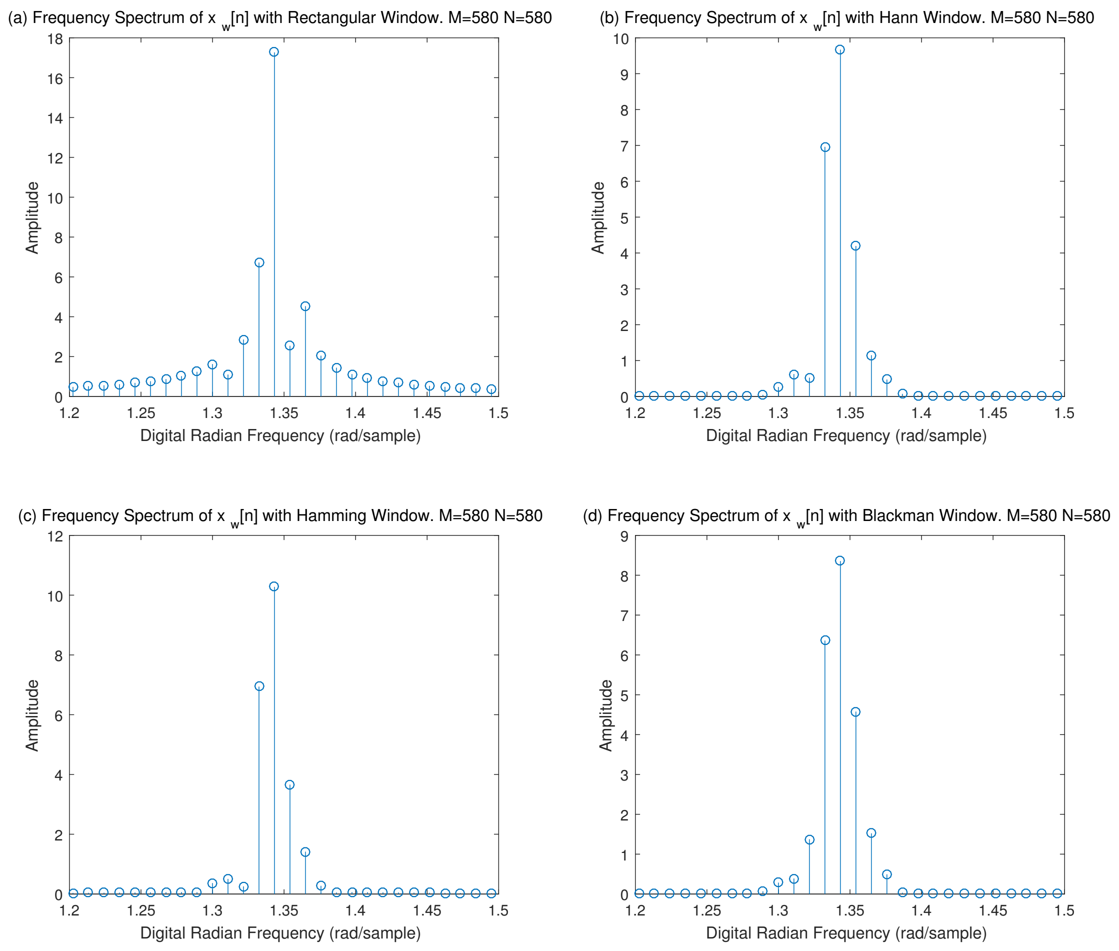

Figure 8: Frequency spectrum for $x_w[n]$ with (a) Rectangular window, (b) Hann window, (c) Hamming window, and (d) Blackman window. $M = N = 580$.

4 Concluding Remarks

In this report, we present a digital spectrum analysis scheme that clearly identifies the frequency components of a linear combination of sinusoids with relatively close frequencies but distinct amplitudes. For rational-valued frequency components, a special method, spectrum leakage avoidance, is introduced, which to a huge extent, saves computational efforts. Real-life general case is also discussed. The whole project is insightful for newcomers in the field of signal processing.

References

[1] Harry Nyquist. Certain topics in telegraph transmission theory. 1928.

[2] James Cooley and John Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):271–301, 1965.

[3] Gerald Goertzel. An algorithm for the evaluation of finite trigonometric series. *The American Mathematical Monthly*, 65(1):34–35, 1958.

A Appendix

A.1 Main Function: Frequency Analyzer

- *FrequencyAnalyzer.m.*

```
%{
FrequencyAnalyzer.m
Draw the frequency spectrum of the given signal, with different
    parameters
of windows and DFT.
Zexi Huang
Dec.22 2016
%}
function FrequencyAnalyzer(windowType,M,N)
%windowType: type of the window choosen, 'R' for Rectangular, 'N' for
%haNn, 'M' for haMming, 'B' for Blackman.
%M: length of the window.
%N: length of DFT.
%f0: number of frequency components detected.
%f: detected frequency.

%Generate the sampled signal.
n=0:1:M-1;
x=10*sin(2*pi*64/1200*n)+sin(2*pi*(250)/1200*n)+20*sin(2*pi*256/1200*n)
    +3*sin(2*pi*260/1200*n)+10*sin(2*pi*512/1200*n);

%Generate the window function.
switch windowType
    case 'R'
        w=rectwin(M);
    case 'N'
        w=hann(M);
    case 'M'
        w=hamming(M);
    case 'B'
        w=blackman(M);
end

%The windowed signal.
x_w=x.*w';
X_W=fft(x_w,N)/(N/2);
X_W=abs(X_W);
omega=0:2*pi/N:2*pi*(1-1/N);

%Draw the figure for results.
stem(omega,X_W);
xlim([0 pi]);
xlabel('Digital Radian Frequency (rad/sample)');
ylabel('Amplitude');
```

A.2 Auxiliary Functions

- *AvoidingLeakageCompareFigure.m.*

```
%{
AvoidingLeakageCompareFigure.m
Draw figures for the avoiding spectrum leakage method with comparison.
Zexi Huang
Dec.22 2016
%}
function AvoidingLeakageCompareFigure(M)
%M: length of DFT and window function.

subplot(3,1,1);
FrequencyAnalyzer('N',M,M);
title(['(a) Frequency Spectrum of x_w[n] with Hann Window. M=N=',num2str(
    M)]);

subplot(3,1,2);
FrequencyAnalyzer('M',M,M);
title(['(b) Frequency Spectrum of x_w[n] with Hamming Window. M=N=',
    num2str(M)]);

subplot(3,1,3);
FrequencyAnalyzer('B',M,M);
title(['(c) Frequency Spectrum of x_w[n] with Blackman Window. M=N=',
    num2str(M)]);
```

- *AvoidingLeakageFigure.m.*

```
%{
AvoidingLeakageFigure.m
Draw figures for the avoiding spectrum leakage method.
Zexi Huang
Dec.22 2016
%}
function AvoidingLeakageFigure(M)
%M: length of DFT and window function.

FrequencyAnalyzer('R',M,M);
title(['Frequency Spectrum of x_w[n] with Rectangular Window. M=N=',
    num2str(M)]);
```

- *AvoidingLeakageFigureTwo.m.*

```
%{
AvoidingLeakageFigureTwo.m
Draw figures for the avoiding spectrum leakage method for special
comparision.
Zexi Huang
Dec.22 2016
%}
function AvoidingLeakageFigureTwo

subplot(2,1,1);
FrequencyAnalyzer('R',75,75);
title('(a) Frequency Spectrum of x_w[n] with Rectangular Window. M=N=75')
    ;

subplot(2,1,2);
FrequencyAnalyzer('R',150,150);
title('(b) Frequency Spectrum of x_w[n] with Rectangular Window. M=N=150'
    );
```

- *CompareDifferentParameters.m.*

```
%{
CompareDifferentParameters.m
Draw figures for comparision of different windows, without considering
    the
spectrum leakage avoidance.
Zexi Huang
Dec.23 2016
%}
function CompareDifferentParameters(M,N)
%M: length of the window.
%N: length of DFT.

subplot(2,2,1);
FrequencyAnalyzer('R',M,N);
xlim([1.2 1.5]);
title(['(a) Frequency Spectrum of x_w[n] with Rectangular Window. M=',
    num2str(M),' N=',num2str(N)]);

subplot(2,2,2);
FrequencyAnalyzer('N',M,N);
xlim([1.2 1.5]);
```

```
title(['(b) Frequency Spectrum of x_w[n] with Hann Window. M=',num2str(M)
    ,' N=',num2str(N)]);

subplot(2,2,3);
FrequencyAnalyzer('M',M,N);
xlim([1.2 1.5]);
title(['(c) Frequency Spectrum of x_w[n] with Hamming Window. M=',num2str
    (M),' N=',num2str(N)]);

subplot(2,2,4);
FrequencyAnalyzer('B',M,N);
xlim([1.2 1.5]);
title(['(d) Frequency Spectrum of x_w[n] with Blackman Window. M=',
    num2str(M),' N=',num2str(N)]);
```

- *WindowDrawer.m.*

```
%{
WindowDrawer.m
Drawing the spectrum of given window funcitons.
Zexi Huang
Dec.22 2016
%}
function WindowDrawer(M)
%M: Length of each window.

%Rectangular window.
subplot(2,2,1);
w=rectwin(M);
[W,omega]=freqz(w,1,8*M);
W=abs(W);
W=W/max(W);
plot(omega,20*log10(W));
axis([0 pi -100 0]);
title('(a) Gain Response of Rectangular Window');
xlabel('Digital Radian Frequency (rad/sample)');
ylabel('Gain/dB');

%Hann window.
subplot(2,2,2);
w=hann(M);
[W,omega]=freqz(w,1,8*M);
W=abs(W);
W=W/max(W);
plot(omega,20*log10(W));
axis([0 pi -100 0]);
title('(b) Gain Response of Hann Window');
xlabel('Digital Radian Frequency (rad/sample)');
ylabel('Gain/dB');

%Hamming window.
subplot(2,2,3);
w=hamming(M);
[W,omega]=freqz(w,1,8*M);
```

```
W=abs(W);
W=W/max(W);
plot(omega,20*log10(W));
axis([0 pi -100 0]);
title('(c) Gain Response of Hamming Window');
xlabel('Digital Radian Frequency (rad/sample)');
ylabel('Gain/dB');

%Blackman window.
subplot(2,2,4);
w=blackman(M);
[W,omega]=freqz(w,1,8*M);
W=abs(W);
W=W/max(W);
plot(omega,20*log10(W));
axis([0 pi -100 0]);
title('(d) Gain Response of Blackman Window');
xlabel('Digital Radian Frequency (rad/sample)');
ylabel('Gain/dB');
```