Laboratory Report of Digital Signal Processing

Lab2.Fourier Analysis

Name:Lu Junyi

No:517261910033

Date:2021.4.21

Score:______________

SHANGHAI JIAOTONG UNIVERSITY
SJTU-ParisTech Elite Institute of Technology

Contents

1 Introduction

Objectives:1. Continuous-time (CT) signals 2. Fourier analysis of CT signals 3. Windowing effects on Fourier transform 4. Discrete Fourier transform and its fast algorithm 5. Application of DFT
The main objectives of this assignment are to master the theory of Fourier transform by exercising, include FS,FT, DTFT, DFT and FFT and know how to realize these algorithm in Matlab. And lead to the reflection about the reason why we proposed these transform and the difference between DSP in theory and DSP in practice.

2 Exercise

2.1 Fourier series of a CT periodic signal

Question1

In Fig1,I create a section of the sine signal $x(t) = A\sin(2\pi f_0 t + \phi)$ where $f_0 = 4, A = 3, \phi = \pi/12$ and plot it over $t = [-1, 1]$ with time interval 0.001.

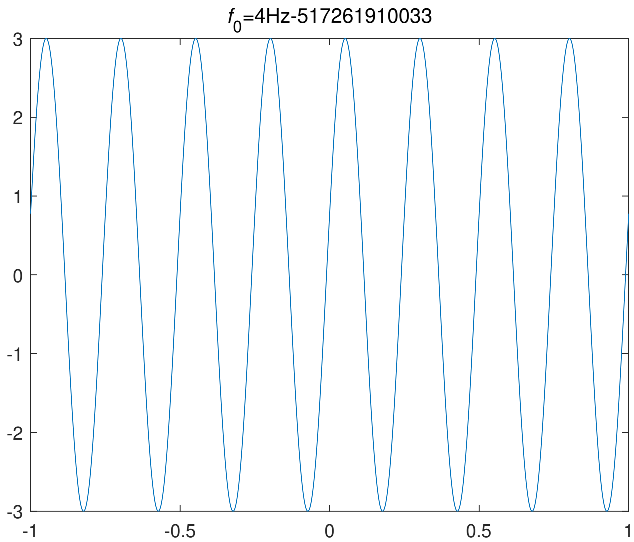

Fig 1. Signal x(t)

Then I calculate the Continuous-time Fourier series (CTFS) coefficients $X(k)$ of $x(t)$ with $k = -5:5$ and plot the frequency spectrum of f in Fig2

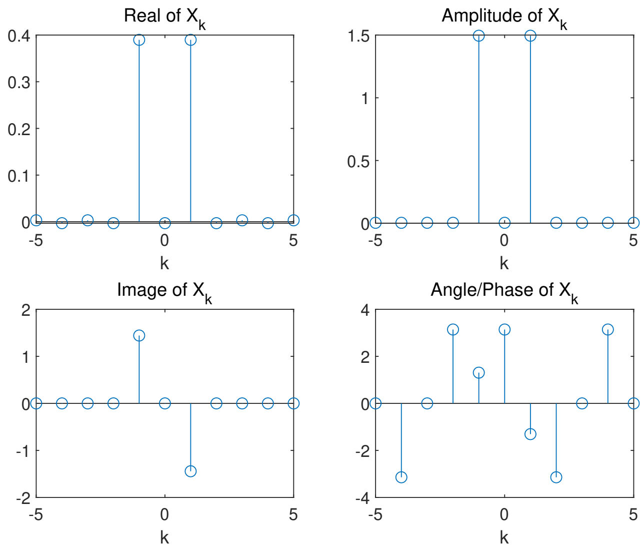

Fig 2. coefficients X(k) of signal x(t)

In order to calculate the coefficients $X(k)$ of $x(t)$, I write a function **ctfs**. In fact, I transform the integration of $x(t)$ into sum, because Matlab deal with continuous signal in forme discrete.

```
function X_k=ctfs(x,t,k,f0)
    T0=1/f0;
    w0=2*pi*f0;
    window=find(t>=-T0/2&t<=T0/2);
    t_window=t(window);
    x_window=x(window);
    dt=T0/length(window);
    X_k=zeros(1,length(k));

    for idx=1:length(k)
        X_k(idx)=1/T0*sum(x_window.*exp(-1j*w0*k(idx)*t_window))*dt;
    end
end
```

Question2

The Parseval's formula for continuous-time periodic signal is:

$$\frac{1}{T}\int_T |x(t)|^2 \, \mathrm{d}t = \sum_{k=-\infty}^{+\infty} |a_k|^2$$

where a_k is the Fourier coefficient of $x(t)$, and T the period of the signal. Here I calculate the integration of $x(t)$ in time domain and then the sum of $X(K)$ in frequency domain and get:

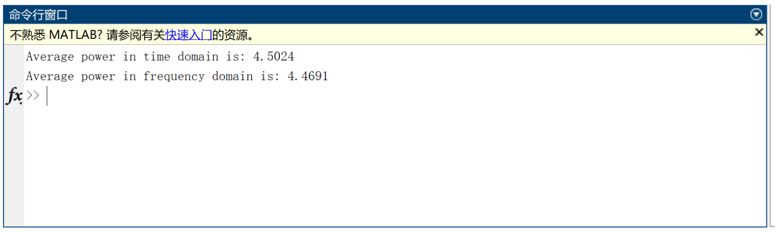

Fig 3. Average power in time/frequency domain

After comparison ,we found that the average power in time domain is close to that in frequency domain,so the Parsval's formula for periodic signals is confirmed.

2.2 Fourier transform of a CT signal

Question1

In Fig4, I create a section of the gate function $y(t)$ and plot it with $D = 8, H = 3$ over $t = [-10, 10]$ with time interval 0.001.

$$y(t) = \begin{cases} H & , -D/2 \le t \le D/2 \\ 0 & , t < -D/2, t > D/2 \end{cases}$$

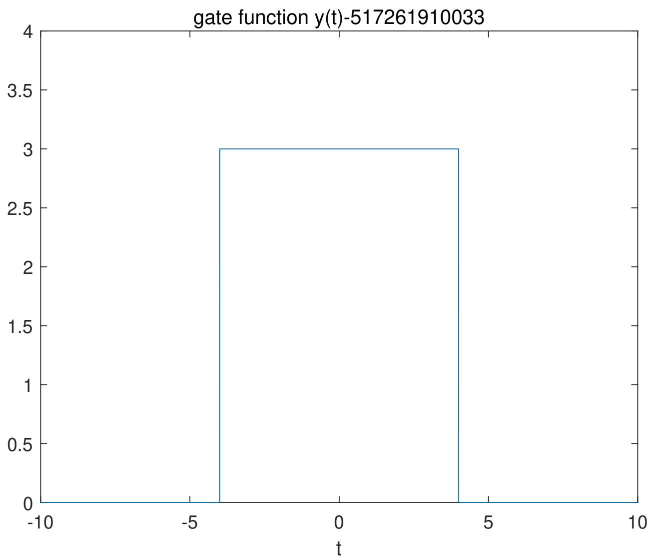

Fig 4. Signal y(t)

Question2

Here I calculate the CTFT of $y(t)$ and show the modules and phases of $Y(\omega)$ with $\omega = -10\pi \sim 10\pi$:

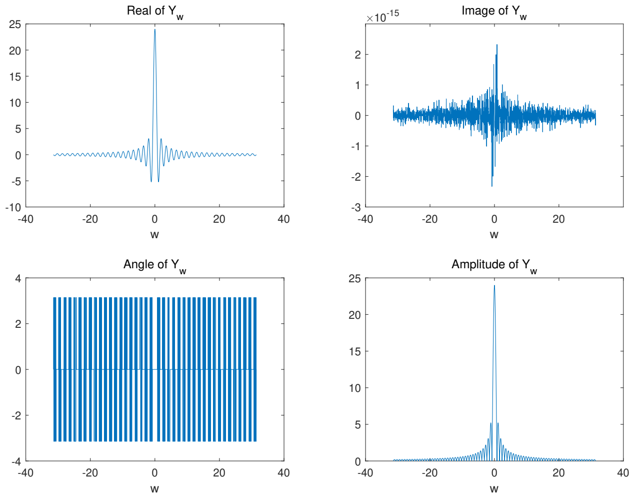

Fig 5. coefficients X(k) of signal x(t)

In order to calculate the coefficients $Y(w)$ of $y(t)$, I write a function **ctft**.

```
function X_jw = ctft(x,t,w)
    dt=(max(t)-min(t))/(length(t)-1);
    X_jw=zeros(1,length(w));
    for idx=1:length(w)
        X_jw(idx)=sum(x.*exp(-1j*w(idx)*t))*dt;
    end
end
```

Question3

Parseval's Formula for Nonperiodic Signals

$$\int_{-\infty}^{\infty} |xt|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |Xj\omega|^2 d\omega = \int_{-\infty}^{\infty} |Xf|^2 df$$

Here I calculate the integration of $y(t)$ in time domain and then the integration of $Y(w)$ in frequency domain and get:

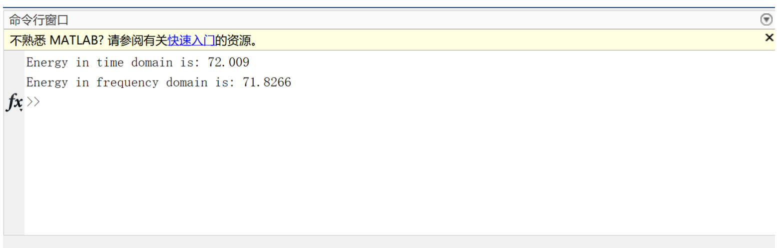

Fig 6. Energy in time/frequency domain

After comparison ,we found that the energy in time domain is close to that in frequency domain,so the Parsval's formula for non-periodic signals is confirmed.

2.3 Windowing effects on FT

Question1

I create the continuous-time harmonics $x(t)$ with $f_s = 1000, f_1 = 17, A_1 = 1.5, \varphi_1 = 0, \Delta f = 2, \quad f_2 = f_1 + \Delta f, A_2 = 0.15, \varphi_2 = 0, \mathrm{D} = 0.8$ and $0 \leq t \leq \mathrm{D}$, according to equation below and plot it in Fig 7:

$$x(t) = A_1 \sin(2\pi f_1 t + \varphi_1) + A_2 \sin(2\pi f_2 t + \varphi_2)$$

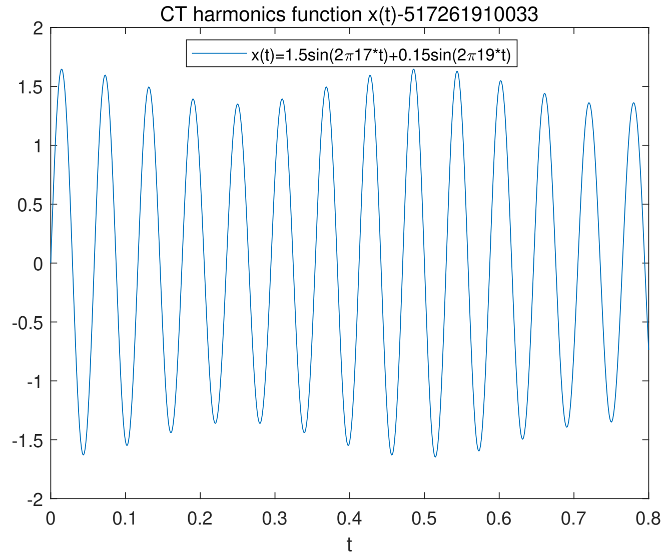

Fig 7. x(t) with D=0.8 in time domain

Question2

The FT results in theory of $x(t)$ is:

$$\begin{aligned}\mathcal{F}_t[x(t)](f) &= \mathcal{F}_t[\ \mathrm{A}_1 \sin(2\pi \mathrm{f}_1 t) + \mathrm{A}_2 \sin(2\pi \mathrm{f}_2 t)](f) \\ &= \int_0^{\infty} (\mathrm{A}_1 \sin(2\pi \mathrm{f}_1 t) + \mathrm{A}_2 \sin(2\pi \mathrm{f}_2 t)) e^{i2\pi f t} dt \\ &= \{\ \mathrm{A}_1 \delta(f - \mathrm{f}_1) -\ \mathrm{A}_1 \delta(\mathrm{f}_1 + f) +\ \mathrm{A}_2 \delta(f - \mathrm{f}_2) -\ \mathrm{A}_2 \delta(\mathrm{f}_2 + f)\} \frac{j}{2}\end{aligned} \tag{1}$$

As we can see ,in the equation above, $t \in [0, +\infty]$. The number of samples is not limited. However, in actual spectrum computations, two additional approximations must be made:

- Finite number of samples $x(nT)$ in calculations. This approximation leads to the concept of a time window and the related effects of smearing and leakage of the spectrum.
- A finite set of frequencies f at which to evaluate $\hat{X}(f)$.

And keeping only a finite number of samples, say $x(nT), 0 \leq n \leq N-1$ requires a so called time-windowing process. The time duration of the record samples is $T_N = NT$, where T is sample interval.

For rectangle windows,which is a gate function, the FT result is:

$$X(j\omega) = \mathcal{F}(G_{T1}(t))(\omega) = \int_{-T_1}^{T_1} e^{-j\omega t} dt = \frac{2\sin(\omega T_1)}{\omega} = 2T_1 Sa(\omega T_1)$$

All The FT in practical of $x(t)$ in Q2,Q3 can be regard as the FT of $x(t) \cdot G_{T1}(t)$ with $T1 = D, D \in \{0.8, 2, 5\}$. And the multiply with a sinusoidal function can be regard as a frequency shifting:

$$x(t)e^{\pm j\omega_0 t} \Leftrightarrow X[j(\omega \mp \omega_0)]$$

So the FT result in practical of $Asin(w_0 t)$ is:

$$F(G_{T_1}(t)) A\sin(\omega_0 t))(\omega) = -j\frac{A}{2} \cdot T_1 \left[\frac{\sin(\omega-\omega_0)T_1}{(\omega-\omega_0)T_1} - \frac{\sin(\omega+\omega_0)T_1}{(\omega+\omega_0)T_1}\right]$$

So in Q2,Q3 and Q4(add rectwin), we need to divide the amplitude of $X(f)$ and $X_w(f)$ with $T1 = D$ to get $A_1/2$ and $A_2/2$

I plot $X(f)$ in Fig 8:

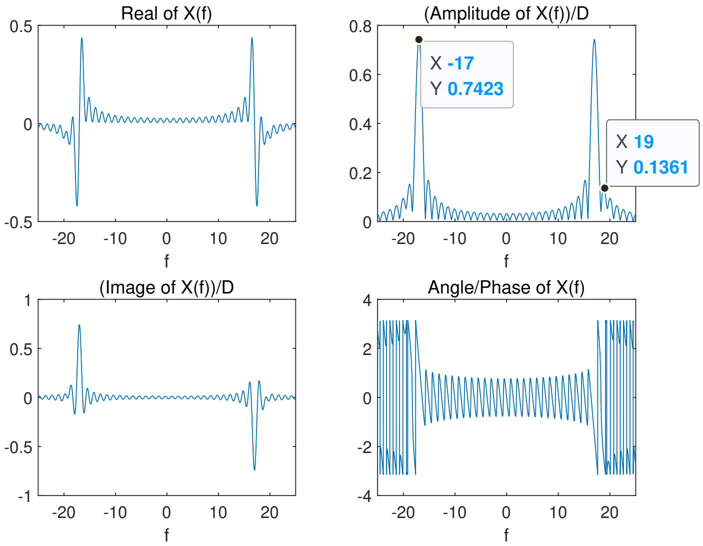

Fig 8. X(f) with D=0.8 in frequency domain

In the spectrum:

- for the first signal harmonic whose frequencies is $f_1 = 17Hz$,the amplitude is $\frac{A_1}{2} = 0.7423$
- for the second signal harmonic whose frequencies is $f_2 = 19Hz$,the amplitude is $\frac{A_2}{2} = 0.1361$

As we can see, The amplitude $\frac{A_2}{2} = 0.1361$is very different from that of original parameters,which is 0.075. The reason is that the windowing process has two major effects:

- reduce the frequency resolution of the computed spectrum
- introduce spurious high-frequency components into the spectrum, referred to as "frequency leakage".

Due to the frequency leakage,the sidelobes of the stronger signal are confused with the weaker signal, so the amplitude of signal with $f_2 = 19Hz$ become quite different.

Question3

I truncate the signals with $D = 2$ and show the windowed signals $x_1(t)$ in Fig 9.Then I calculate and plot the FT of this signals in Fig10:

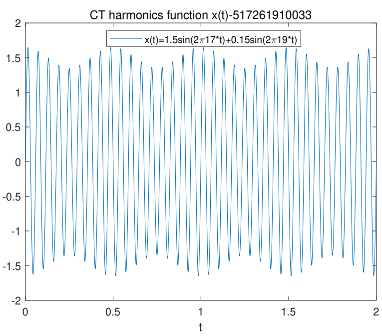

Fig 9. $x_1(t)$ with D=2

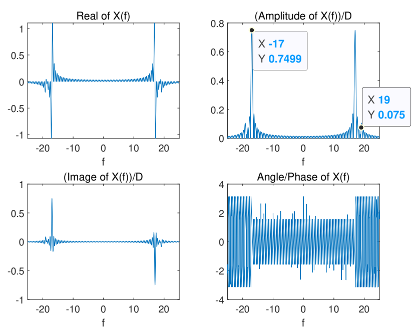

Fig 10. $X_1(f)$ with D=2

As we can see in the spectrum:

- for the first signal harmonic whose frequencies is $f_1 = 17Hz$,the amplitude is $\frac{A_1}{2} = 0.7499$
- for the second signal harmonic whose frequencies is $f_2 = 19Hz$,the amplitude is $\frac{A_2}{2} = 0.075$

I truncate the signals with $D = 5$ and show the windowed signals $x_2(t)$ in Fig 11.Then I calculate and plot the FT of this signals in Fig12:

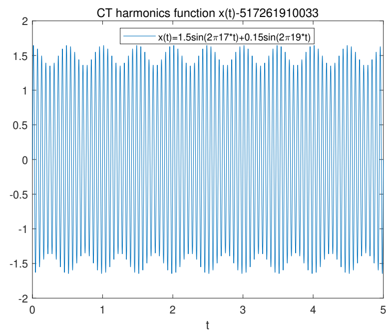

Fig 11. $x_2(t)$ with D=5

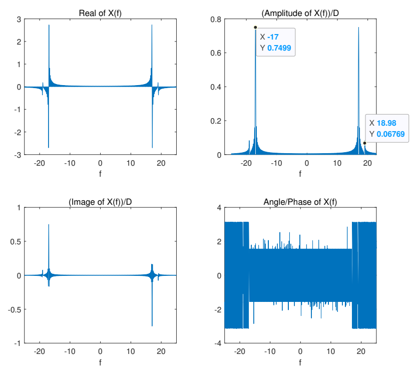

Fig 12. $X_2(f)$ with D=5

As we can see in the spectrum:

- for the first signal harmonic whose frequencies is $f_1 = 17Hz$,the amplitude is $\frac{A_1}{2} = 0.7499$
- for the second signal harmonic whose frequencies is $f_2 = 19Hz$,the amplitude is $\frac{A_2}{2} = 0.0676$

Question4

I apply the rectangle window with width D = 5 to truncate the signals and show the windowed signals $x_{\mathrm{w}}(t)$. in Fig 13 .Then I calculate and plot the modulus of FT of $x_{\mathrm{w}}(t)$ in Fig 14:

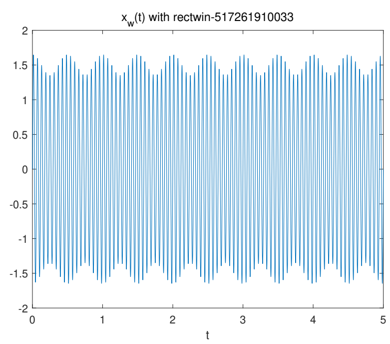

Fig 13. $x_2(t)$ with D=5,add rectangle window

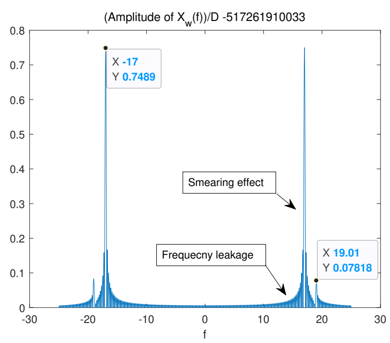

Fig 14. $X_2(f)$ with D=5,add rectangle window

In digital signal processing, the process of truncating data is also known as windowing. Truncating the data directly is equivalent to multiplying the signal with an impulse signal. The impulse signal is also called a rectangular window. It is natural to add a rectangular window to the data. But notice that the rectangular window is discontinuous at both the beginning and the end, i.e. the signal starts and ends abruptly.

Physically it is well understood that this sudden change in the time domain means that there are very high frequency components in the frequency domain. Obviously, as long as these high-frequency components are suppressed, the sidelobe level can be reduced, thus effectively suppressing the leakage of the sub-flap. Therefore, from the point of view of suppressing the sidelobe, it is intuitively clear that the flatter the beginning and end of the window function, the better the suppression of the sidelobe should be.

The rectangular window is obviously not a good window from the point of view of sidelobe leakage.

The standard technique for suppressing the sidelobes is to use a non-rectangular window —a window that cuts off to zero less sharply and more gradually than the rectangular one.

So I apply the hamming window with width D = 5 to truncate the signals:

$$w(n)=0.54-0.46\cos\left(\frac{2\pi n}{N}\right)\qquad n=0,1,\cdots,N-1$$

$$w(n)=0.54+0.46\cos\left(\frac{2\pi n}{N}\right)\qquad n=-\frac{N}{2},\cdots,0,\cdots,\frac{N}{2}$$

The FT result of Hamming window is:

$$W\left(\mathrm{e}^{\mathrm{i}\omega}\right)=0.54U(\omega)+0.23U\left(\omega-\frac{2\pi}{N}\right)+0.23U\left(\omega+\frac{2\pi}{N}\right)$$

$$B=1.3\Delta\omega,\quad B_0=\frac{8\pi}{N},\quad A=-43\text{ dB},\quad D=-6\text{ dB/oct}$$

So we need to divide the amplitude of $X_w(f)$ with $0.54*D$ with $D=5$ to get $A_1/2$ and $A_2/2$

I show the windowed signals $x_{\mathrm{w}}(t)$. in Fig 15 .Then I calculate and plot the modulus of FT of $x_{\mathrm{w}}(t)$ in Fig 16:

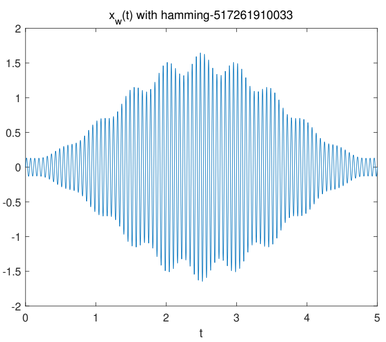

Fig 15. $x_2(t)$ with D=5,add hamming window

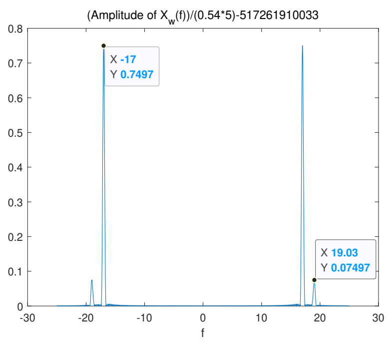

Fig 16. $X_2(f)$ with D=5,add hamming window

We can see in Fig 16 that after adding Hamming window to the signal, the Leakage effect and Smearing effect have been greatly weakened.

The higher the side-lobe level of the window function is, the narrower the corresponding main-lobe width is. In order to suppress the frequency leakage, it is often desired that the main-lobe width is small and the side-lobe level is low at the same time. However, these two indicators are a contradiction, it is difficult to meet the requirements at the same time. In the actual signal processing process,we need to choose a compromise according to the application of the background. Frequency resolution is usually the same as the trend of the main-lobe width, that is, the wider the main-lobe, the lower the frequency resolution.

The resolvability condition that the two sinusoids appear as two distinct ones is that their frequency separation Δf be greater than the mainlobe width:

$$\Delta f \geq \Delta f_w = \frac{f_s}{L} \quad \text{(frequency resolution)}$$

The minimum number of samples required to achieve a desired frequency resolution Δf can be determined. The smaller the desired separation, the longer the data record.

Question5

I note the two harmonics of $x(t)$ as $x_a(t)$ and $x_b(t)$:

$$x_a(t) = A_1 \sin(2\pi f_1 t + \varphi_1)$$

$$x_b(t) = A_2 \sin(2\pi f_2 t + \varphi_2)$$

Then I plot the FS results of the two harmonics of $x(t)$ together in Fig 17:

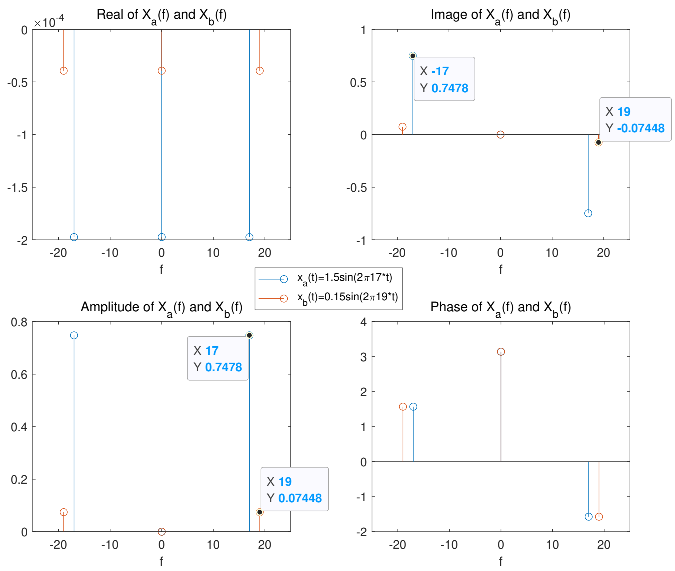

Fig 17. FS of $x_a(t)$ and $x_b(t)$

However, the FT results of $x(t)$ is not so good in comparision with the FS result:

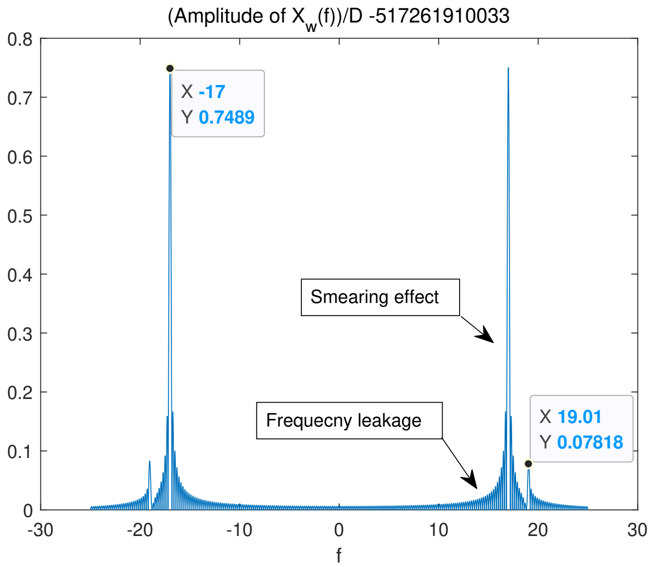

Fig 18. Frequency leakage and Smearing Effect

We can see that the two effects are:

- Smearing effect:Due to the small time span, the width of the peaks in the frequency domain is very large, so the peaks in close proximity are superimposed on each other, i.e., the frequency domain resolution is small.
- Frequency leakage: the spectrum has some peaks that should not exist.

2.4 Discrete Time Fourier Transform and Discrete Fourier Transform

2.4.1 DTFT and DFT

DTFT

I create a triangular wave $x[n]$in equation below and plot in Fig 19:

$$x[n] = \begin{cases} -|n| + 5, |n| \leq 5 \\ 0, 6 \leq |n| \leq 15 \end{cases}$$

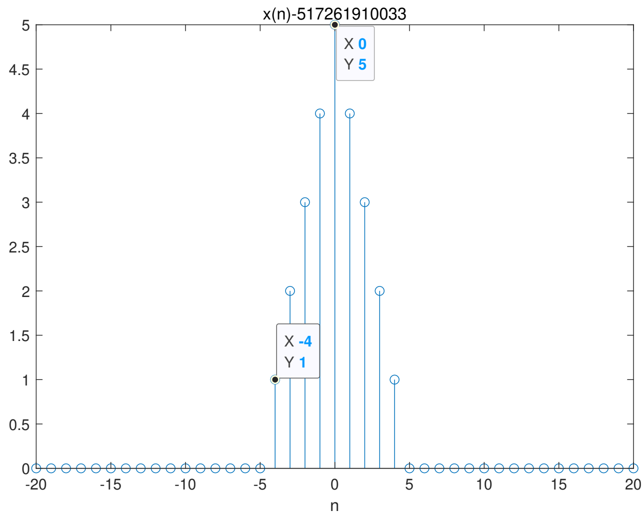

Fig 19. x[n] in time domain

I calculate the DTFT of $x[n]$ in the Nyquist frequency range $[-1/2, 1/2]$:

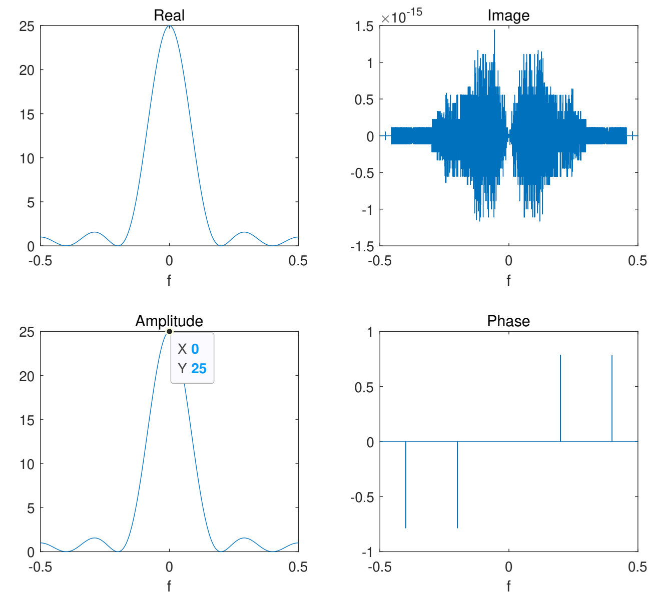

Fig 20. DTFT of x[n]

iDTFT

What's more, I calculate the inverse of DTFT and plot the result in Fig 21, the algorithm in detail can be obtain in Appendix.

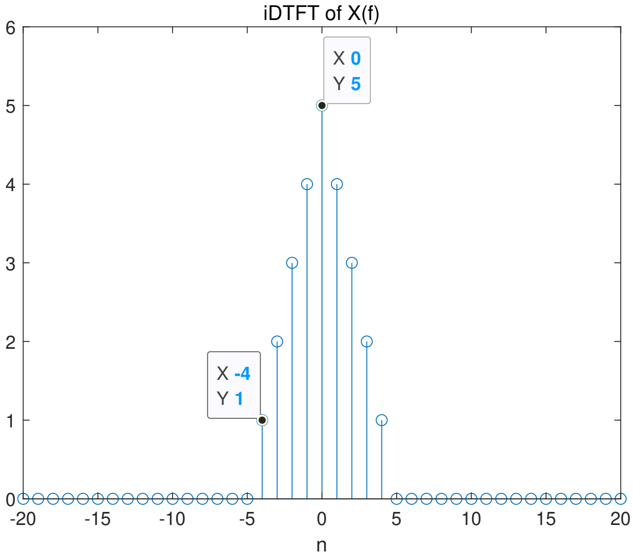

Fig 21. iDTFT of $X[e^{jw}]$

The result of iDTFT has some small errors compared with the original signal, which is due to the fact that the result of iDTFT is complex and its imaginary part is not zero.

DFT

I calculate the DFT of $x[n]$ in the Nyquist frequency range $[-1/2, 1/2]$:

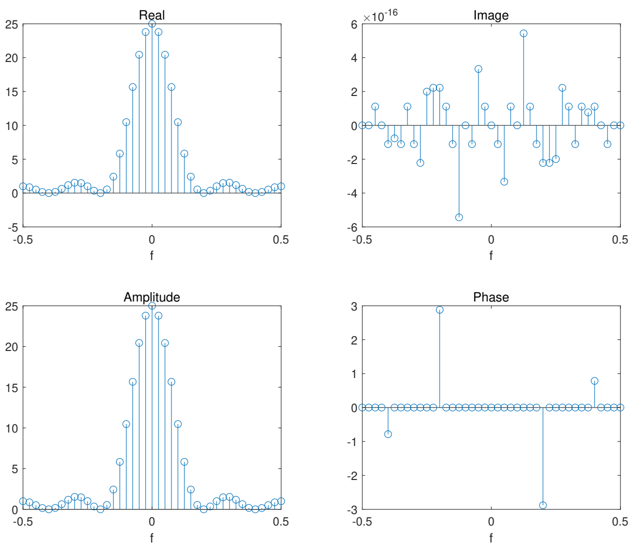

Fig 22. DFT of x[n]

The computational resolution is defined as the spacing between the DFT frequencies at which the DTFT is computed,and note as Bin width in DFT, in rads/sample:

$$\Delta\omega_{\text{bin}} = \frac{2\pi}{N}$$

In this problem, I selected the time domain sampling point n in the range [-20,20], so N=41, so the computation resolution here is:

$$\Delta\omega_{\text{bin}} = \frac{2\pi}{41}$$

2.4.2 DFT and FFT

Plot y[n] and yw[n]

I define a cosine signal $y[n]$, as shown in equation below, where $f_0 = 1/16, A = 2.5$, and define a rectangle window $w[n]$, as shown in equation below. The truncated cosine signal $yw[n] = y[n] \cdot w[n]$ has a length of $N+1$, where $N = 16$

$$y[n] = A\cos(2\pi f_0 n)$$
$$w[n] = \begin{cases} 1, 0 \le n \le N \\ 0, \text{ others} \end{cases}$$

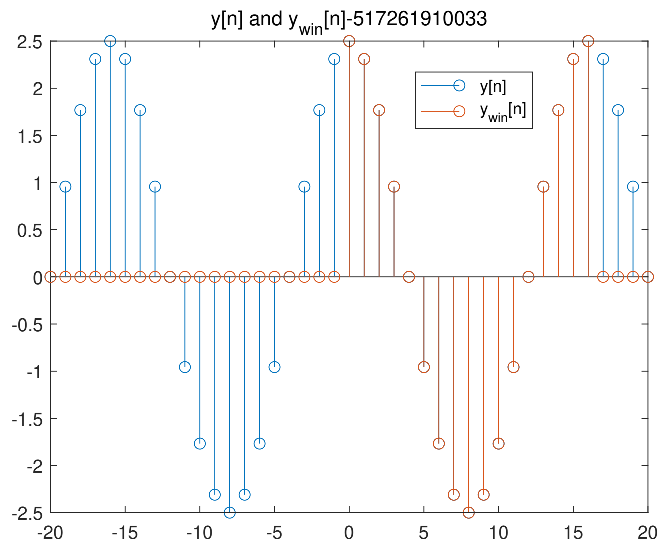

Fig 23. $y[n]$ and $y_{win}[n]$ in time domain

Calculate and show DFT and FFT

I calculate and show DFT and fft (MATLAB function) in a Nyquist interval in radians $[0, 2\pi]$.

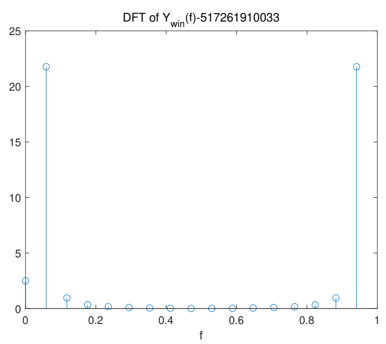

Fig 24. $Y_{win}[n]$ by DFT

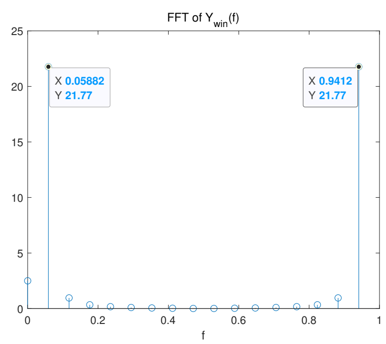

Fig 25. $Y_{win}[n]$ by FFT

Investigate the computational time of DFT and fft

In order to investigate the computational time of DFT and fft with respect to the window length,I use a tic-toc function to evaluate the computational time with $N = [10, 50, 100, 500, 1000, 5000]$ and get the table below:

N	10	50	100	500	1000	5000
time of DFT	$2.23 \cdot 10^{-3}$s	$1.17 \cdot 10^{-3}$s	$1.05 \cdot 10^{-3}$s	$1.08 \cdot 10^{-2}$s	$3.39 \cdot 10^{-2}$s	44.28s
time of FFT	$9.58 \cdot 10^{-5}$s	$6.17 \cdot 10^{-5}$s	$5.43 \cdot 10^{-5}$s	$1.53 \cdot 10^{-4}$s	$3.77 \cdot 10^{-4}$s	$3.03 \cdot 10^{-3}$s

I show the curve of computational time with respect to N in figure below:

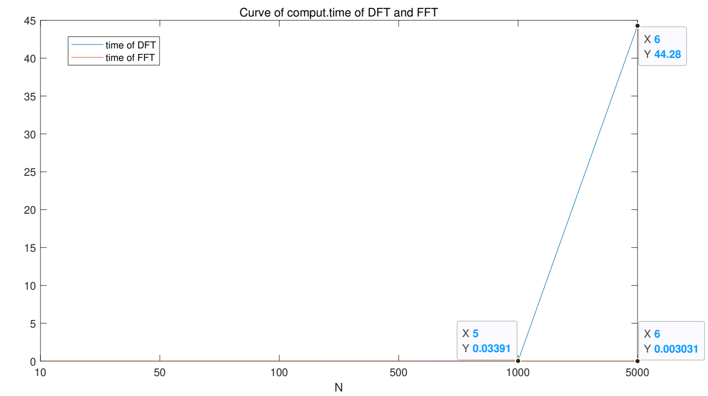

Fig 26. The computation time of DFT and FFT

we can see that FFT is faster than DFT when N is large.

Investigate the relationship between fft algorithm and the power of 2 points

In order to investigate the relationship between fft algorithm and the power of 2 points, use a tic-toc function to evaluate the computational time with $N = [1000, 2^{10}, 2000, 2^{11}]$.

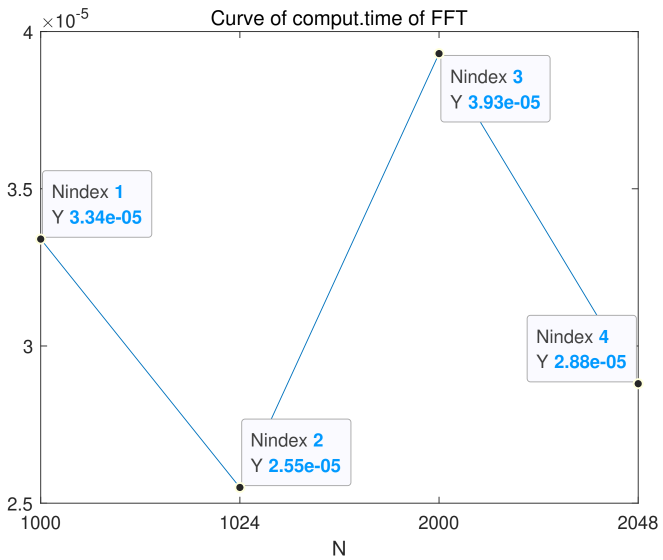

Fig 27. The computation time of FFT

We find that when N is the power of 2, the computation of fft spends less time.

2.5 Application of DFT

The ideal electricity wave is a harmonic signal with the frequency of $f_0 = 50$ Hz and the amplitude of $A_0 = 220$ V, written as $E_0 = A_0 \sin(2\pi f_0 t)$, denoted as $e_0 = A_0@f_0$. Due to the interference of some power inverters, the practical electrical waveform is distorted and then includes the fundamental frequency $e_1 = 225@50$, the high frequency harmonics $e_2 = 3@101$, $e_3 = 2.5@140$ and inter-harmonics $\mathrm{e}_4 = 0.5@55, \mathrm{e}_5 = 1.5@104$.

$$E_n(t) = \sum_{i=1}^{n} e_i$$

we want to detect the frequency components accurately in the distorted wave signal, so I simulate the detecting process as follows

First of all, First, we define some variables to facilitate the explanation later:

$$T_{length} : \text{the length of window}$$

$$Fs : \text{the sampling frequency}$$

$$L : \text{the number of sample points in time domain}$$

$$n : \text{the number of sample points in frequncy domain}$$

Question.a

I construct a uniform rectangle window with a length of 0.1 s and a sampling frequency of 1000 Hz to truncate a time sequence x[n] from equation above, I use a matrix to represent the signal E_5 truncated by the rectangle window:

```
Fs = 1000;                    % Sampling frequency
T = 1/Fs;                     % Sampling period
% L = 1000;                   % Length of signal
L = 100;
t = (0:L-1)*T;                % Time vector

e1 = 225*sin(2*pi*50*t);           % First row wave
e2 = 3*sin(2*pi*101*t);            % Second row wave
e3 = 2.5*sin(2*pi*140*t);          % Third row wave
e4 = 0.5*sin(2*pi*55*t);
e5 = 1.5*sin(2*pi*104*t);

E5=e1+e2+e3+e4+e5;

X = [e1; e2; e3; e4; e5];
```

Then I plot E_5 and its components e_i in Figure below:

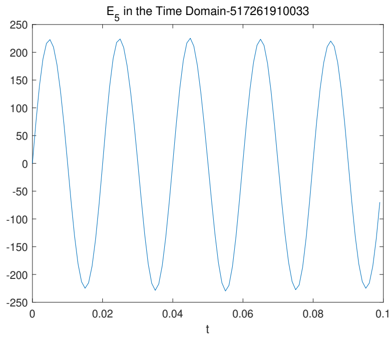

Fig 28. E_5 truncated in time domain by rectwin,length=0.1s

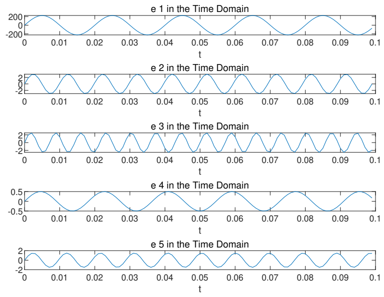

Fig 29. e_i truncated in time domain by rectwin,length=0.1s

Question.b

I calculate the frequency amplitude of E_5 by using DTFT at the supposed frequencies, then I divide the the result of DTFT with L,the number of sample points in time domain,to get the proper amplitude:

```
df1 = 0.01;
f1 = 0:df1:Fs-df1;
n1=length(f1);

E5_f_DTFT=zeros(1,length(f1));
for ii=1:length(f1)
    E5_f_DTFT(ii)=sum(E5.*exp(-1j*2*pi*f1(ii)*t));
end

%%
E5_P2_DTFT=abs(E5_f_DTFT/L);
E5_P1_DTFT=E5_P2_DTFT(:,1:n1/2+1);
E5_P1_DTFT(:,2:end-1) = 2*E5_P1_DTFT(:,2:end-1);

%%
figure
plot(0:df1:(Fs/2-df1),E5_P1_DTFT(1:n1/2))
title('Amplitude of E_5 in the Frequency Domain(DTFT)')
```

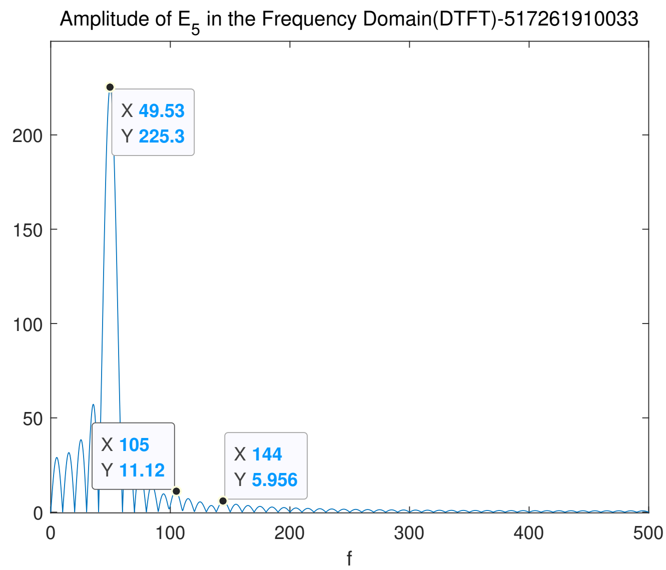

Fig 30. E_5 in frequency domain(DTFT)
rectwin,length=0.1s

As we can see, it's hard to find these frequency components with this spectra, What's more, the frequency error and the amplitude error are large. So we decide to use DFT and FFT

Question.c

rectwin, length=0.1s

First,I use the algorithm of DFT below to calculate the frequency amplitude:

```
df = Fs/n;
f = 0:df:Fs-df;

E5_f_DTFT=zeros(1,length(f));
for ii=1:length(f)
    E5_f_DTFT(ii)=sum(E5.*exp(-1j*2*pi*f(ii)*t));
end

%%
E5_P2_DFT=abs(E5_f_DTFT/L);=
E5_P1_DFT=E5_P2_DFT(:,1:n/2+1);=
E5_P1_DFT(:,2:end-1) = 2*E5_P1_DFT(:,2:end-1);=

%%
figure
plot(0:df:(Fs/2-df),E5_P1_DFT(1:n/2))
title('Amplitude of E_5 in the Frequency Domain(DFT)-517261910033')
```

And I found that the result of my DFT is the same with the function fft() of Matlab, so in the rest part of my report, I will only show the result of FFT,you can verify the result with code in Appendix

I plot the FFT result of E_5 and e_i in figures below:

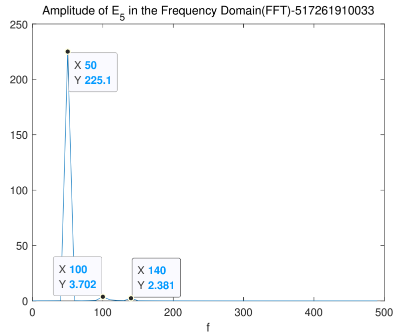

Fig 31. E_5 in frequcny domain(FFT)
rectwin,length=0.1s

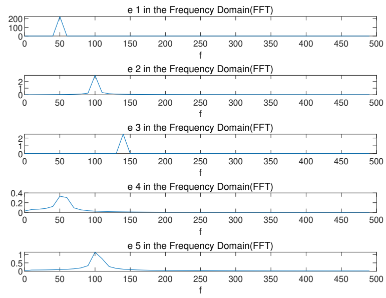

Fig 32. e_i in frequcny domain(FFT)
rectwin,length=0.1s

The result of FFT is much better than DTFT, but still has error with amplitude and frequency.

hamming,length=0.1s

In this part, I try to use a window with same length but differrent shape, and the code in detail can be obtain in Appendix

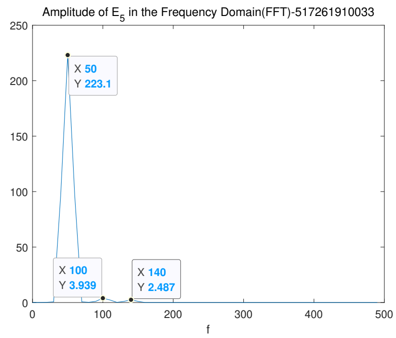

Fig 33. E_5 in frequcny domain(FFT)
hamming,length=0.1s

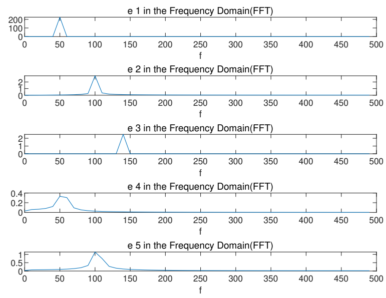

Fig 34. e_i in frequcny domain(FFT)
hamming,length=0.1s

Unfortunately, it seems the result of FFT dosen't become better, However, we found that the result of DTFT with Hamming is much more clear, the influence of frequency leakage is weakened.

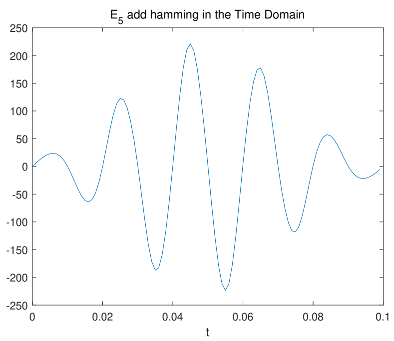

Fig 35. E_5 trunacted in time domain
hamming,length=0.1s

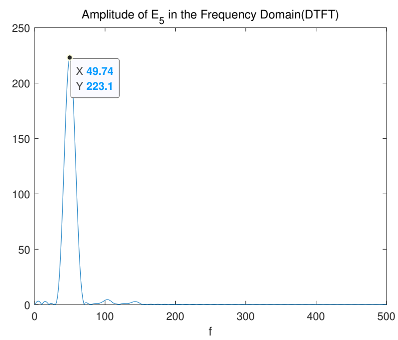

Fig 36. E_5 in frequcny domain(DTFT)
hamming,length=0.1s

rectwin,length=1s

In this part, I try to use a window with same shape but much longer (from $T_{length} = 0.1s$ to $T_{length} = 1s$). So the number of sample points in time domain(L) increase to 1000 and the number of sample points in frequency domain(n) increase to 1000.

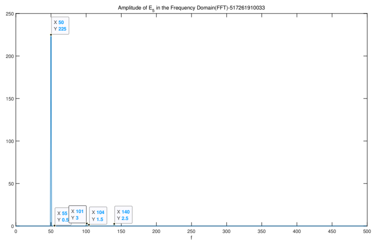

Fig 37. E_5 in frequcny domain(FFT) rectwin,length=1s

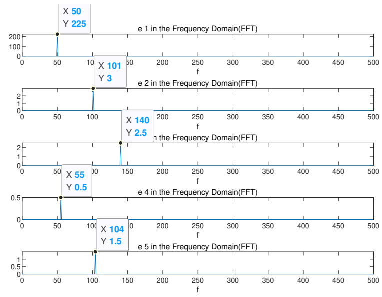

Fig 38. e_i in frequncy domain(FFT) rectwin,length=1s

The frequency and amplitude in this spectra are accurate. If T_{length} increase to 1s, the peak of spectra will become more intense.

What's more, If I change n into the power of 2(by using the command "n = 2nextpow2(L)") and let Matlab do zero padding to the original signal, the computation time will become less, and the spectra become smoother, but the error of frequency and amplitude become greater:

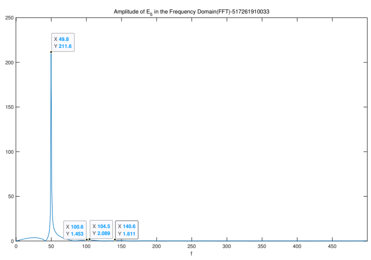

Fig 39. E_5 in frequcny domain(FFT) rectwin,length=1s,n=1024

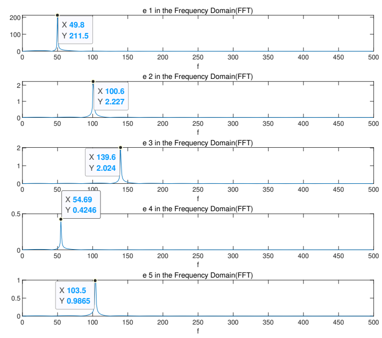

Fig 40. e_i in frequncy domain(FFT) rectwin,length=1s,n=1024

Question.d

From Question.c, we can see that when Fs is fixed, with the increase of the length of windows , the error in frequency and amplitude of FFT become less and the spectra become clear.That's because the resolution is determined by the equation below:

$$\Delta f = \frac{F_s}{\text{number of points to do fft}}$$

When the sampling frequency F_s is fixed, we can augment the resolution by augment the number of points to do fft. However, the process of zero padding will increase error in frequency and amplitude, so only increase n is useless if we want to augment the resolution, we need to increase L,the number of sample points in time domain.

3 Summary

From this assignment, we found the biggest difference between DSP in theory and DSP in practice is thatthe data of signal in limited and discrete. In fact, he digital signal processing is always like a convolution of a rectangle window with the discrete sampling points of the signal infinity original ,which result in frequency leakage and smearing effect. So we introduce the process of window adding and also DTFT,DFT to solve these problems. In order to increase the speed of DFT's computation, we introduce FFT.

A Appendix

A.1 Ex.1

```
clc;clear;close all
%% Parameter and continuous signal
A=3;
fs=1000;
dt=1/fs;
phi=pi/12;
f0=4;
T0=1/f0;
t=-4*T0:dt:4*T0;
x=A*sin(2*pi*f0*t+phi);

figure
plot(t,x)
str=['{\itf}_0=',num2str(f0),'Hz'];
title(str);
%% Calculate Fourier series
kk=-5:1:5;
X_k=ctfs(x,t,kk,f0);
%% Show teh Fourier Series
figure
subplot(221)
stem(kk,real(X_k))
title('Real of X_k')
xlabel('k')
subplot(223)
stem(kk,imag(X_k))
title('Image of X_k')
xlabel('k')
subplot(222)
stem(kk,abs(X_k))
title('Amplitude of X_k')
xlabel('k')
subplot(224)
stem(kk,angle(X_k))
title('Angle/Phase of X_k')
xlabel('k')
%% Average Power
window=find(t<=T0/2&t>=-T0/2);
x_window=x(window);
```

```
P_t=1/T0*sum(abs(x_window).^2)*dt;
disp(['Average power in time domain is: ',num2str(P_t)])
P_f=sum(abs(X_k).^2);
disp(['Average power in frequency domain is: ',num2str(P_f)])
```

A.2 Ex.2

```
clc;clear;close all
%% Parameter and continuous signal
D=8;
H=3;
dt=0.001;
t=-10:dt:10;
y=H.*(t>=-D/2&t<=D/2)+0.*(t<-D/2&t>D/2);

figure
plot(t,y)
ylim([0,H+1])
xlabel('t')
title('gate function y(t)-517261910033')
%% Calculate Fourier Transform
dw=0.01*pi;
w=-10*pi:dw:10*pi;
Y_w=ctft(y,t,w);
%% Show the Fourier Transform
figure
subplot(221)
plot(w,real(Y_w))
title('Real of Y_w')
xlabel('w')
subplot(222)
plot(w,imag(Y_w))
title('Image of Y_w')
xlabel('w')
subplot(223)
plot(w,angle(Y_w))
title('Angle of Y_w')
xlabel('w')
subplot(224)
plot(w,abs(Y_w))
title('Amplitude of Y_w')
xlabel('w')
```

```
%% Parsval formula
dw=(max(w)-min(w))/(length(w)-1);
E_t=sum(abs(y).^2*dt);
disp(['Energy in time domain is: ',num2str(E_t)])
E_w=(1/(2*pi))*sum(abs(Y_w).^2*dw);
disp(['Energy in frequency domain is: 1',num2str(E_w)])
```

A.3 Ex.3

```
clc;clear;close all
%% Parameter and CT harmonics function x(t)
fs=1000;
dt=1/fs;

f1=17;
A1=1.5;
phi1=0;

deltaf=2;

f2=f1+deltaf;
A2=0.15;
phi2=0;

% D=0.8;
% D=2;
D=5;
t=0:dt:D;% time domain

f=-25:0.001:25;% frequency domain

x1=A1*sin(2*pi*f1*t+phi1);
x2=A2*sin(2*pi*f2*t+phi2);
x=x1+x2;


%% x(t)
figure
plot(t,x)
xlabel('t')
legend('x(t)=1.5sin(2\pi17*t)+0.15sin(2\pi19*t)')
title('CT harmonics function x(t)-517261910033')
```

```
%% FT of x(t)
X_f=ctft_f(x,t,f);

figure
subplot(221)
plot(f,real(X_f))
title('Real of X(f)')
xlabel('f')
subplot(223)
plot(f,imag(X_f)/D)
title('(Image of X(f))/D')
xlabel('f')
subplot(222)
plot(f,abs(X_f)/D)
title('(Amplitude of X(f))/D')
xlabel('f')
subplot(224)
plot(f,angle(X_f))
title('Angle/Phase of X(f)')
xlabel('f')

%% Add rectwin to x(t) with D=5 and plot the FT
% x_rectangle = boxcar(length(t));
% x_win=x.*x_rectangle';
% figure
% plot(t,x_win)
% xlabel('t')
% title('x_w(t) with rectwin -517261910033')
%
% % FT of x_win(t)
% X_win_f=ctft_f(x_win,t,f);
%
% figure
% plot(f,abs(X_win_f)/D)
% title('(Amplitude of X_w(f))/D -517261910033')
% xlabel('f')
%% Add hamming to x(t) with D=5 and plot the FT
x_hamming = hamming(length(t));
x_win=x.*x_hamming';

figure
plot(t,x_win)
xlabel('t')
```

```
title('x_w(t) with hamming-517261910033')

% FT of x_win(t)
X_win_f=ctft_f(x_win,t,f);

figure
plot(f,abs(X_win_f)/(0.54*5))
title('(Amplitude of X_w(f))/(0.54*5)-517261910033')
xlabel('f')
%% FS of x(t)'s harmonic
kk=-5:1:5;
X1_k=ctfs(x1,t,kk,f1);
X2_k=ctfs(x2,t,kk,f2);

figure
subplot(221)
stem(kk*f1,real(X1_k))
title('Real of X_a(f) and X_b(f)')
xlabel('f')
hold on
stem(kk*f2,real(X2_k))
xlim([-25,25])
hold off
subplot(222)
stem(kk*f1,imag(X1_k))
title('Image of X_a(f) and X_b(f)')
xlabel('f')
hold on
stem(kk*f2,imag(X2_k))
xlim([-25,25])
hold off
subplot(223)
stem(kk*f1,abs(X1_k))
title('Amplitude of X_a(f) and X_b(f)')
xlabel('f')
hold on
stem(kk*f2,abs(X2_k))
xlim([-25,25])
hold off
subplot(224)
stem(kk*f1,angle(X1_k))
title('Phase of X_a(f) and X_b(f)')
xlabel('f')
```

```
hold on
stem(kk*f2, angle(X2_k))
xlim([-25,25])
hold off
legend('x_a(t)=1.5sin(2\pi17*t)','x_b(t)=0.15sin(2\pi19*t)')
```

ctft f

```
function X_f = ctft(x,t,f)
    dt=(max(t)-min(t))/(length(t)-1);
    X_f=zeros(1,length(f));
    for idx=1:length(f)
        X_f(idx)=sum(x.*exp(-1j*2*pi*f(idx)*t))*dt;
    end
end
```

A.4 Ex.4

Question1

```
clc;clear;close all;
%% 4.1: DTFT and DFT
n=-20:1:20;
x=(-abs(n)+5).*(abs(n)<=5)+0.*(abs(n)<=15 & abs(n)>=6);

figure
stem(n,x)
xlabel('n')
title('x(n)-517261910033')
%% %% DTFT
df1=0.0001;
ff1=-1/2:df1:1/2-df1;
X_f_DTFT=zeros(1,length(ff1));
for ii=1:length(ff1)
    X_f_DTFT(ii)=sum(x.*exp(-1j*2*pi*ff1(ii)*n));
end
figure
subplot(221)
plot(ff1, real(X_f_DTFT))
xlabel('f')
title('Real')
subplot(222)
plot(ff1, imag(X_f_DTFT))
xlabel('f')
```

```
title('Image')
subplot(223)
plot(ff1,abs(X_f_DTFT))
xlabel('f')
title('Amplitude')
subplot(224)
plot(ff1,angle(X_f_DTFT))
xlabel('f')
title('Phase')
%% %% iDTFT
d=zeros(1,length(n));
for ii=1:length(n)
    d(ii)=sum(X_f_DTFT.*exp(1j*2*pi*n(ii)*ff1))*df1;

end
figure
stem(n,abs(d))
xlabel('n')
title('iDTFT of X(f)')
%% %% DFT
N=length(n);
df2=1/(N-1);
ff2=-1/2:df2:1/2;
X_f_DFT=zeros(1,length(ff2));
for ii=1:length(ff2)
    X_f_DFT(ii)=sum(x.*exp(-1j*2*pi*ff2(ii)*n));
end
figure
subplot(221)
stem(ff2,real(X_f_DFT))
xlabel('f')
title('Real')
subplot(222)
stem(ff2,imag(X_f_DFT))
xlabel('f')
title('Image')
subplot(223)
stem(ff2,abs(X_f_DFT))
xlabel('f')
title('Amplitude')
subplot(224)
stem(ff2,angle(X_f_DFT))
xlabel('f')
```

```
title('Phase')
```

Question2.a,2.b

```
clc;clear;close all
%% Definition of fs
fs=1;
%% Definition of signal
n=-20:1:20;


N=16;
dt=1/fs;
t=0:dt:N*dt;


f0=1/16;
A=2.5;
y=A*cos(2*pi*f0*n);
w=1.*(n>=0 & n<=N)+0.*(n<=0 & n>=N);
ywin=y.*w;


figure
stem(n,y)
xlim([-20,20])
hold on
stem(n,ywin)
hold off
legend('y[n]','y_{win}[n]')
title('y[n] and y_{win}[n]-517261910033')

ywin=ywin(find(w));
%% DFT and FFT
%% %% DFT

L=N;
df=fs/(L+1);
f=0:df:1-df;

Ywin_f_DFT=zeros(1,length(f));
tic
for ii=1:length(f)
    Ywin_f_DFT(ii)=sum(ywin.*exp(-1j*2*pi*f(ii)*t));
```

```
end
toc

figure
stem(f,abs(Ywin_f_DFT))
title('DFT of Y_{win}(f)-517261910033')
xlabel('f')
%% %% FFT
tic
Ywin_f_FFT=fft(ywin);
toc

figure
stem(f,abs(Ywin_f_FFT))
title('FFT of Y_{win}(f)-517261910033')
xlabel('f')
```

Question2.c

```
clc;clear;close all;
%% Compute time
fs=1;

Nval=[10 50 100 500 1000 50000];
Nind=[1 2 3 4 5 6];
T_DFT=zeros(1,length(Nval));
T_FFT=zeros(1,length(Nval));

for jj=1:length(Nval)

    N=Nval(jj);

    dt=1/fs;
    t=0:dt:N*dt;

    f0=1/16;
    A=2.5;
    ywin=A*cos(2*pi*f0*t);



    L=N;

    df=fs/(L+1);
```

```
        f=0:df:1-df;
        %DFT
        Ywin_f_DFT=zeros(1,length(f));
        tic
        for ii=1:length(f)
            Ywin_f_DFT(ii)=sum(ywin.*exp(-1j*2*pi*f(ii)*t));
        end
        T_DFT(jj)=toc;

        %FFT
        tic
        Ywin_f_FFT=fft(ywin);
        T_FFT(jj)=toc;
end
figure
plot(Nind,T_DFT)
hold on
plot(Nind,T_FFT)
xlabel('N')
legend('time of DFT','time of FFT')
title('Curve of comput.time of DFT and FFT--517261910033')
```

Question2.d

```
clc;clear;close all;
%% Compute time
fs=1;

Nval=[1000 2^10 2000 2^11];
Nind=[1 2 3 4];

T_FFT=zeros(1,length(Nval));

for jj=1:length(Nval)

        N=Nval(jj)-1;

        dt=1/fs;
        t=0:dt:N*dt;

        f0=1/16;
        A=2.5;
        ywin=A*cos(2*pi*f0*t);

```

```


    L=N;

    df=fs/(L+1);
    f=0:df:1-df;

    %FFT
    tic
    Ywin_f_FFT=fft(ywin);
    T_FFT(jj)=toc;
end
figure
plot(Nind,T_FFT)
xlabel('N')
title('Curve of comput.time of FFT-517261910033')
```

A.5 Ex.5

Add rectwin

```
clc;clear; close all;
%%
Fs = 1000;                    % Sampling frequency
T = 1/Fs;                     % Sampling period
L = 1000;                   % Length of signal
% L = 100;
t = (0:L-1)*T;                % Time vector

e1 = 225*sin(2*pi*50*t);          % First row wave
e2 = 3*sin(2*pi*101*t);           % Second row wave
e3 = 2.5*sin(2*pi*140*t);         % Third row wave
e4 = 0.5*sin(2*pi*55*t);
e5 = 1.5*sin(2*pi*104*t);

E5=e1+e2+e3+e4+e5;

X = [e1; e2; e3; e4; e5];
%%
figure
plot(t(1:100),E5(1:100))
title('E_5 in the Time Domain-517261910033')
xlabel('t')
```

```
figure
for i = 1:5
    subplot(5,1,i)
    plot(t(1:100),X(i,1:100))
    title(['e ',num2str(i),' in the Time Domain'])
    xlabel('t')
end

%%
df1 = 0.01;
f1 = 0:df1:Fs-df1;
n1=length(f1);

E5_f_DTFT=zeros(1,length(f1));
for ii=1:length(f1)
    E5_f_DTFT(ii)=sum(E5.*exp(-1j*2*pi*f1(ii)*t));
end

%% %%
E5_P2_DTFT=abs(E5_f_DTFT/L);
E5_P1_DTFT=E5_P2_DTFT(:,1:n1/2+1);
E5_P1_DTFT(:,2:end-1) = 2*E5_P1_DTFT(:,2:end-1);

%% %%
figure
plot(0:df1:(Fs/2-df1),E5_P1_DTFT(1:n1/2))
title('Amplitude of E_5 in the Frequency Domain(DTFT)-517261910033')
xlabel('f')
%%
n = 2^nextpow2(L);
% n=L;

%%

df = Fs/n;
f = 0:df:Fs-df;

E5_f_DTFT=zeros(1,length(f));
for ii=1:length(f)
    E5_f_DTFT(ii)=sum(E5.*exp(-1j*2*pi*f(ii)*t));
end

```

```
%% %%
E5_P2_DFT=abs(E5_f_DTFT/L);
E5_P1_DFT=E5_P2_DFT(:,1:n/2+1);
E5_P1_DFT(:,2:end-1) = 2*E5_P1_DFT(:,2:end-1);

%% %%
figure
plot(0:df:(Fs/2-df),E5_P1_DFT(1:n/2))
title('Amplitude of E_5 in the Frequency Domain(DFT)-517261910033')
xlabel('f')

%%

E5_f_FFT=fft(E5,n);

dim = 2;
Y = fft(X,n,dim);

%% %%
E5_P2_FFT=abs(E5_f_FFT/L);
E5_P1_FFT=E5_P2_FFT(:,1:n/2+1);
E5_P1_FFT(:,2:end-1) = 2*E5_P1_FFT(:,2:end-1);

P2 = abs(Y/L);
P1 = P2(:,1:n/2+1);
P1(:,2:end-1) = 2*P1(:,2:end-1);

%% %%
figure
plot(0:(Fs/n):(Fs/2-Fs/n),E5_P1_FFT(1:n/2))
title('Amplitude of E_5 in the Frequency Domain(FFT)-517261910033')
xlabel('f')
figure
for i=1:5
    subplot(5,1,i)
    plot(0:(Fs/n):(Fs/2-Fs/n),P1(i,1:n/2))
    title(['e ',num2str(i),' in the Frequency Domain(FFT)'])
    xlabel('f')
end
```

Add hamming

```
clc;clear; close all;
%%
```

```
Fs = 1000;                    % Sampling frequency
T = 1/Fs;                     % Sampling period
% L = 1000;                   % Length of signal
L = 100;
t = (0:L-1)*T;                % Time vector

e1 = 225*sin(2*pi*50*t);          % First row wave
e2 = 3*sin(2*pi*101*t);           % Second row wave
e3 = 2.5*sin(2*pi*140*t);         % Third row wave
e4 = 0.5*sin(2*pi*55*t);
e5 = 1.5*sin(2*pi*104*t);

E5=e1+e2+e3+e4+e5;

X = [e1; e2; e3; e4; e5];
%%
figure
plot(t(1:100),E5(1:100))
title('E_5 in the Time Domain')
xlabel('t')

figure
for i = 1:5
    subplot(5,1,i)
    plot(t(1:100),X(i,1:100))
    title(['e ',num2str(i),' in the Time Domain'])
    xlabel('t')
end
%%
E5_win=window(@hamming,L);
E5=E5.*E5_win';
k=1.852;
figure
plot(t,E5)
title('E_5 add hamming in the Time Domain')
xlabel('t')
%%
df1 = 0.01;
f1 = 0:df1:Fs-df1;
n1=length(f1);

E5_f_DTFT=zeros(1,length(f1));
for ii=1:length(f1)
```

```
        E5_f_DTFT(ii)=sum(E5.*exp(-1j*2*pi*f1(ii)*t));
end

%%
E5_P2_DTFT=abs(E5_f_DTFT*k/L);
E5_P1_DTFT=E5_P2_DTFT(:,1:n1/2+1);
E5_P1_DTFT(:,2:end-1) = 2*E5_P1_DTFT(:,2:end-1);

%%
figure
plot(0:df1:(Fs/2-df1),E5_P1_DTFT(1:n1/2))
title('Amplitude of E_5 in the Frequency Domain(DTFT)')
xlabel('f')

%%
% n = 2^nextpow2(L);
n=L;

%%

df = Fs/n;
f = 0:df:Fs-df;

E5_f_DTFT=zeros(1,length(f));
for ii=1:length(f)
        E5_f_DTFT(ii)=sum(E5.*exp(-1j*2*pi*f(ii)*t));
end

%%
E5_P2_DFT=abs(E5_f_DTFT*k/L);
E5_P1_DFT=E5_P2_DFT(:,1:n/2+1);
E5_P1_DFT(:,2:end-1) = 2*E5_P1_DFT(:,2:end-1);

%%
figure
plot(0:df:(Fs/2-df),E5_P1_DFT(1:n/2))
title('Amplitude of E_5 in the Frequency Domain(DFT)-517261910033')
xlabel('f')


%%

E5_f_FFT=fft(E5,n);
```

```
dim = 2;
Y = fft(X,n,dim);

%%
E5_P2_FFT=abs(E5_f_FFT*k/L);
E5_P1_FFT=E5_P2_FFT(:,1:n/2+1);
E5_P1_FFT(:,2:end-1) = 2*E5_P1_FFT(:,2:end-1);

P2 = abs(Y/L);
P1 = P2(:,1:n/2+1);
P1(:,2:end-1) = 2*P1(:,2:end-1);

%%
figure
plot(0:(Fs/n):(Fs/2-Fs/n),E5_P1_FFT(1:n/2))
title('Amplitude of E_5 in the Frequency Domain(FFT)-517261910033')
xlabel('f')

figure
for i=1:5
    subplot(5,1,i)
    plot(0:(Fs/n):(Fs/2-Fs/n),P1(i,1:n/2))
    title(['e ',num2str(i),' in the Frequency Domain(FFT)'])
    xlabel('f')
end
```