Lab 4 Fourier Transform

Objective

- Learn the Fourier transform of continuous-time signal with MATLAB.
- Analyze the signals with Fourier Transform.
- Analyze the LTI system with system models.

Content

As for a periodic rectangular pulse signal, $\omega = \frac{2\pi}{T}$, as T increases, the density of the spectrum line increases, as shown in Figure 1.

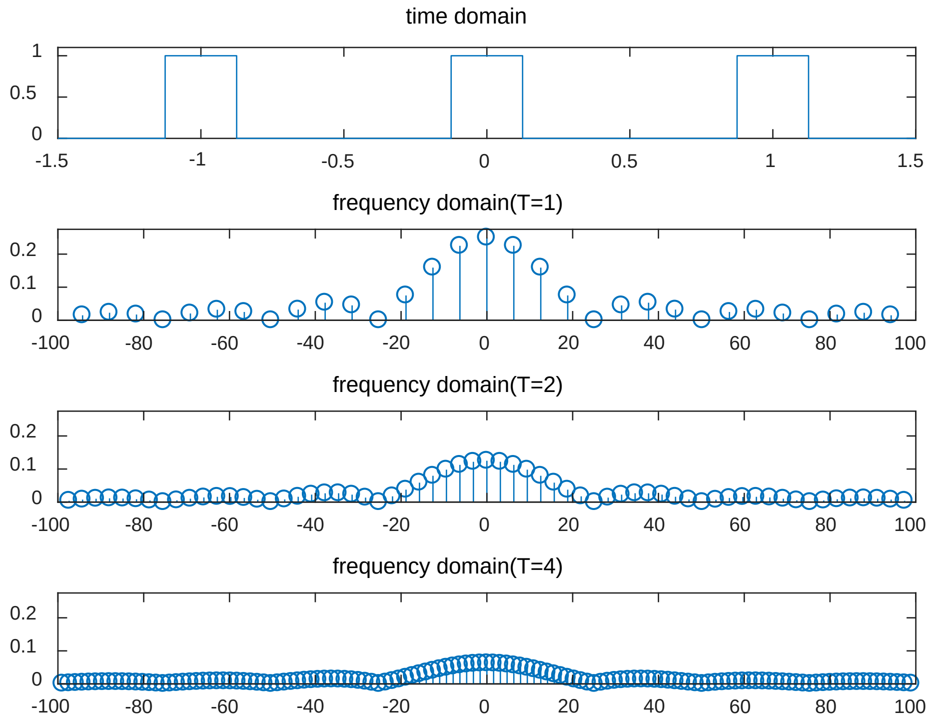


Figure 1 Influence of T

When $T \rightarrow \infty$, the periodic signal changes to aperiodic signal, and the spectrum line changes from discrete ones to continuous ones. However, the amplitude of the spectrum line also gets smaller, while the relative difference still exists. So for aperiodic signals, it is necessary to remove the effect of T in the frequency domain. As shown in Figure 2.

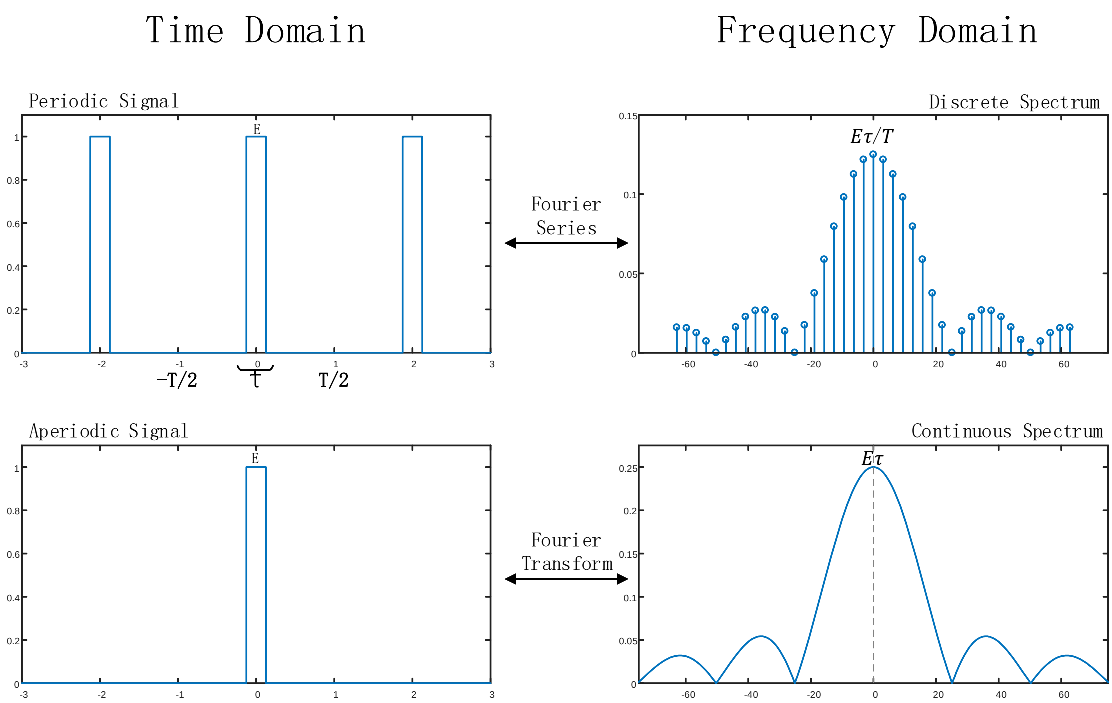


Figure 2 Fourier Series and Fourier Transform

A time signal f(t) is a function of time t, where $-\infty \leq t \leq \infty$. This signal may be represented in the frequency domain using the Fourier transform.

Signal Analysis

Fourier Transform and Inverse Fourier Transform with Symbolic Method

The Fourier transform of f(t) is defined as:

$$\mathrm{F}(j\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}\,dt$$

And the inverse Fourier transform is defined as:

$$\mathrm{f(t)} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(j\omega)e^{j\omega t}\,\mathrm{d}\omega$$

For signals that can be expressed by an expression, use the function **fourier** and **ifourier** to do the Fourier transfer and inverse Fourier transfer with MATLAB. Both **fourier** and **ifourier** are symbolic methods. The format of the two functions is listed in Table 1.

Table 1 Format of fourier and ifourier

Format	Description
F=fourier(f)	Do Fourier transform of function f with variable x. The result F is the function with variable w.
F=fourier(f,v)	Do Fourier transform of function f with variable x. The result F is the function with variable v.

F=fourier(f,u,v)	Do Fourier transform of function f with variable u. The result F is the function with variable v.
f=ifourier(F)	Do inverse Fourier transform of function F with variable w. The result f is the function with variable x.
f = ifourier(F,u)	Do inverse Fourier transform of function F with variable w. The result f is the function with variable u.
f = ifourier(F,v,u)	Do inverse Fourier transform of function F with variable v. The result f is the function with variable u.

Use function **abs** and **angle** to plot the amplitude-frequency and phase-frequency.
Example: Do the Fourier transfer of single-sided exponential signal.

$$f(t) = \begin{cases} e^{-2t}, t > 0 \\ 0, t < 0 \end{cases}$$

```
syms t
a = 2;
f = exp(-(a*t))*heaviside(t);
subplot(3,1,1);fplot(f);axis([-3,3,-0.1,1.1]);grid on;
F = fourier(f);
subplot(3,1,2);fplot(abs(F));axis([-10,10,0,.6]);grid on;
subplot(3,1,3);fplot(angle(F));axis([-10,10,-pi/2,pi/2]);grid on;
```

Example: Do the inverse Fourier transfer of $F(j\omega) = \frac{1}{1+\omega^2}$.

```
syms t w
ifourier(1/(1+w^2),t);
```

Fourier Transform with Numeric Method

Fourier Transform with Function fft

Function fft is used to do Fourier transform in the numeric method. Format fft(x) computes the discrete Fourier transform of x using a fast Fourier transform algorithm. Format fft(x,n) returns the n-point DFT.
Example: x=cos(2*pi*10.*t)+2sin(2*pi*15.*t)+3cos(2*pi*20.*t)). Observe the signal in frequency domain.

```
N = 256;
Fs = 100;
t = [0:N-1]./Fs;
x = 1*cos(2*pi*10.*t) + 2*sin(2*pi*15.*t) + 3*cos(2*pi*20.*t);
X = fft(x);
plot(abs(X))
```

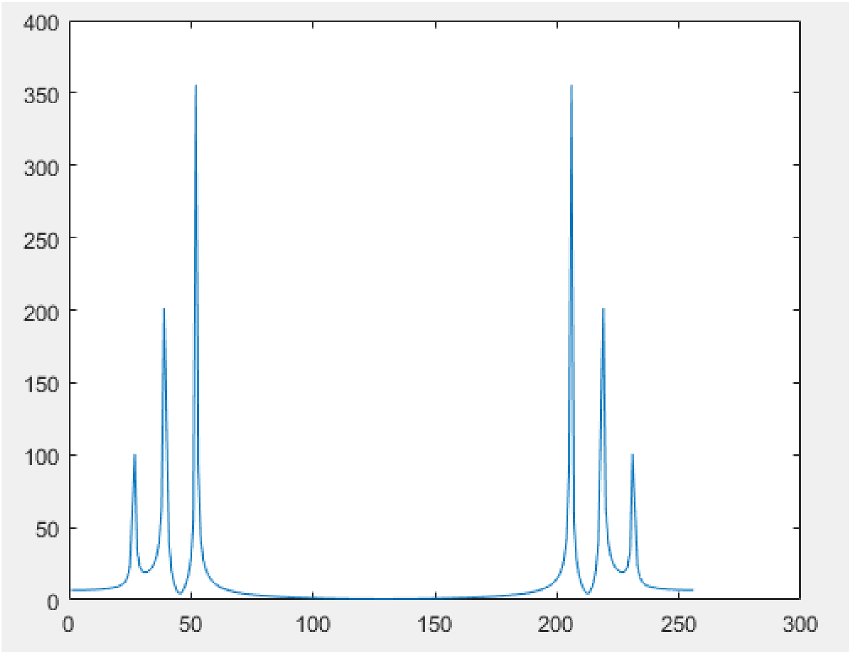
1. Figure 3 (a) shows the graphics of X and (b) shows the value. From (b), it is easy to find out that the 1^{th} and the 129^{th} items are real numbers, and the items at the sides of 129^{th} are conjugated. This is because MATLAB present frequency components from f=0 to f=Fs (from DC to the sampling frequency). However, we are only interested in the range of 0 to Fs/2. So only the 1^{th}

to 128th items (N/2) are valuable.

2. The corresponding frequency to the X(n) is calculated by $f(n) = \frac{n-1}{N}Fs$, n = 1,2, … , N, where $\frac{Fs}{N}$ is the frequency resolution.

3. As mentioned in 1 and 2, the effective observation items are from 1th to $\frac{N}{2}$th, so the corresponding frequency is from 0 to $\frac{Fs}{2}$. So this is the basis for choosing a suitable Fs.

4. According to the characters of function fft, it is recommended that N prefers 2^n.



(a)

```
1 至 5 列
 0.0665 + 0.0000i   0.0665 + 0.0015i   0.0665 + 0.0029i   0.0667 + 0.0044i   0.0668 + 0.0060i
6 至 10 列
 0.0670 + 0.0076i   0.0673 + 0.0092i   0.0676 + 0.0109i   0.0679 + 0.0128i   0.0683 + 0.0147i
...
121 至 125 列
-0.0067 - 0.0018i  -0.0066 - 0.0015i  -0.0066 - 0.0013i  -0.0065 - 0.0011i  -0.0065 - 0.0009i
126 至 130 列
-0.0065 - 0.0007i  -0.0064 - 0.0004i  -0.0064 - 0.0002i  -0.0064 + 0.0000i  -0.0064 + 0.0002i
131 至 135 列
-0.0064 + 0.0004i  -0.0065 + 0.0007i  -0.0065 + 0.0009i  -0.0065 + 0.0011i  -0.0066 + 0.0013i
...
251 至 255 列
 0.0673 - 0.0092i   0.0670 - 0.0076i   0.0668 - 0.0060i   0.0667 - 0.0044i   0.0665 - 0.0029i
256 列
 0.0665 - 0.0015i
```

(b)

Figure 3 fft Result

5. Amplitude can be got by using function abs(). To get right amplitude, the 1st element needs

to be divided by N due to the calculation method of fft. The other elements need to be divided by N first, and then multiplied by 2 to transfer the energy at the right part of the 129th element to the left part.

As we know the amplitude and frequency of x is 1@10Hz, 2@15Hz, 3@20Hz. Let's change the code to:

```
clear
N = 256;
Fs = 100; dt = 1/Fs;
df = Fs/N;
f = [0:N-1]*df;
t = [0:N-1]*dt;
x=1*cos(2*pi*10.*t)+2*sin(2*pi*15.*t)+3*cos(2*pi*20.*t));
X = fft(x);
Xabs = abs(X);
Xabs = 2*Xabs/N;
Xabs(1) = Xabs(1)/2;
Xabs = Xabs(1:N/2);
f = f(1:N/2);
plot(f,Xabs)
```

The result is show in Figure 4.

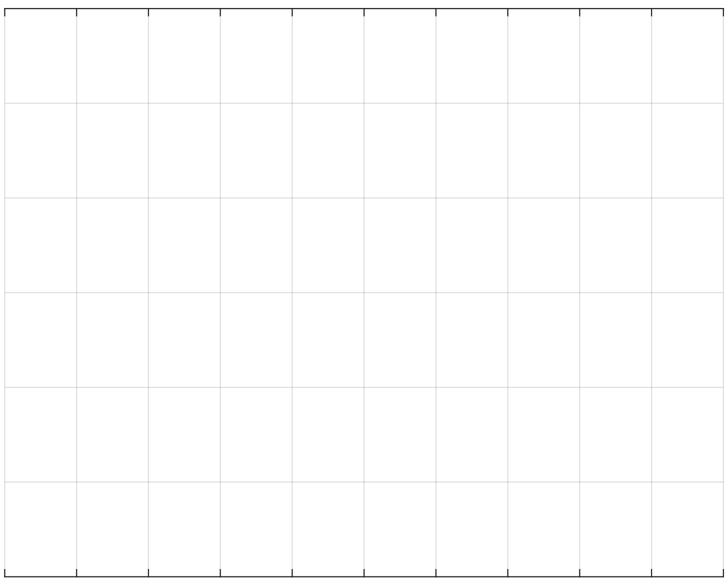


Figure 4 Adjustment

6. There is still some deviation of amplitude caused by the spectrum leak. To reduce the spectrum leak, the frequency of signal should be an integer multiple of the frequency resolution. That is $f = \frac{Fs}{N} * k$. Adjust Fs and N to make the equation hold. Some adjustment of Fs and N is made in Table 2. Plot them to view the effect. The result is shown in Figure 5.

Table 2 Adjusting of Fs and N

k=f*N/Fs,N=256			
	1@10Hz	2@15Hz	3@20Hz
Fs=120	21.33	32	42.67
Fs=160	16	24	32
k=f*N/Fs,Fs=100			
	1@10Hz	2@15Hz	3@20Hz
N=260	26	39	52
N=250	25	37.5	50

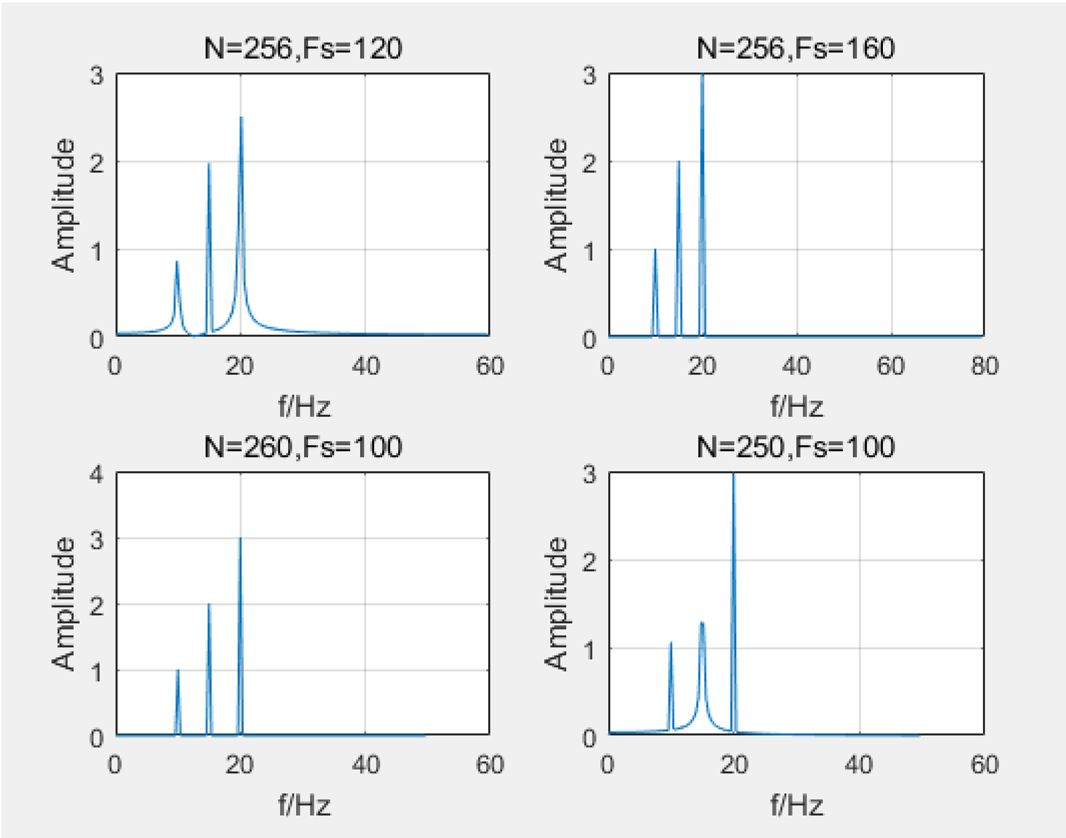


Figure 5 Adjust Fs and N

About Using N and dt

Function fft is an efficient DFT algorithm. It is an operation of discrete signals. The definition of DFT is:

$$X_{DFT}(k) = \sum_{n=0}^{N-1} x(n) e^{-jk\frac{2\pi}{N}n} = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad k = 0, \cdots, N-1$$

When performing fft on a continuous signal, the continuous signal needs to be sampled first. For a continuous periodic signal x(t) with a period T_0, its Fourier series is like:

$$\mathrm{X}(\mathrm{jk}w_0) = \frac{1}{T_0} \int_0^{T_0} x(t) e^{-jkw_0 t} dt$$

After discretizing T_0 with N point, we define: $T_s = \frac{T_0}{N}$, $\mathrm{t} = \mathrm{k}T_s$, $dt = T_s$, $T_0 = \mathrm{N}T_s$. Thus

$$\mathrm{X(jk}w_0) = \frac{1}{T_0}\int_0^{T_0} x(t)e^{-jkw_0 t}dt = \frac{T_s}{T_0}\sum_{n=0}^{N-1} x(nT_s)e^{-jk\frac{2\pi}{T_0}nT_s}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} x(nT_s)e^{-jk\frac{2\pi}{N}nT_s} = \frac{1}{N}X_{DFT}(k) = X_T(k)$$

So when using function fft to find out the Fourier transform of a continuous periodic function, we need to divide Fn by N.

As for a continuous aperiodic signal x(t), we have $\mathrm{X(jw)} = \int_{-\infty}^{+\infty} x(t)e^{-jwt}dt$. Divide the signal according to the following Figure 6, we get

$$\mathrm{X(jw)} = \int_{-\infty}^{+\infty} x(t)e^{-jwt}dt = \sum_{n=0}^{N-1}\int_{nT_s}^{(n+1)T_s} x(t)e^{-jwt}dt$$

$$= \sum_{n=0}^{N-1} x(nT_s)e^{-jw\cdot nT_s}\cdot T_s \quad = T_s \cdot X_{DFT}(k)$$

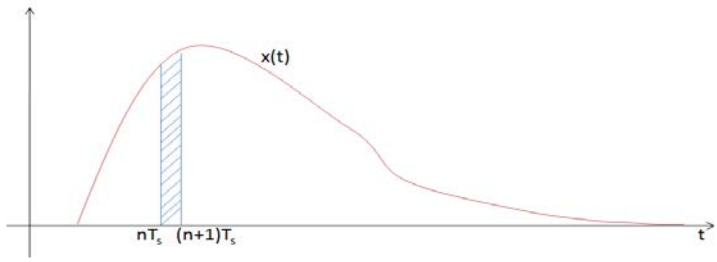


Figure 6 Divide the signal by Ts

So when using function fft to find out the Fourier transform of a continuous aperiodic function, we need to multiply Fn by dt.

Direct Fourier Transform

In addition to do the Fourier transform with function fft, sometimes we can also use matrix operations to do the Fourier transform directly. For aperiodic signal,

$$\mathrm{F}(j\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}\,dt = \lim_{\tau\to 0}\sum_{n=-\infty}^{\infty} f(n\tau)e^{-jwn\tau}\tau$$

is true, if τ is small enough. Since the signal is time-limited, N can be taken as the border of n. After discretizing ω, we have

$$F(w_k) = \tau\sum_{n=-N}^{n=N} f(n\tau)e^{-j\omega_k n\tau}, -\mathrm{N} \le \mathrm{n} \le \mathrm{N}, -\mathrm{M} \le \mathrm{k} \le \mathrm{M}$$

$$f_s = \frac{1}{\tau}$$

$$\omega_k = \frac{2\pi f_s}{M}k = \frac{2\pi}{M\tau}k$$

Which can be expressed in the matrix as follow:

$$\mathrm{F}(\mathrm{w}_k) = \begin{bmatrix} F(w_{-M}) \\ \vdots \\ F(w_0) \\ \vdots \\ F(w_M) \end{bmatrix} = \tau \cdot \begin{bmatrix} f(-N\tau)\cdot e^{-jw_{-M}(-N\tau)} + \cdots + f(0\cdot\tau)\cdot e^{jw_{-M}(0\cdot\tau)} + \cdots + f(N\tau)\cdot e^{-jw_{-M}(N\tau)} \\ \vdots \\ f(-N\tau)\cdot e^{-jw_0(-N\tau)} + \cdots + f(0\cdot\tau)\cdot e^{jw_0(0\cdot\tau)} + \cdots + f(N\tau)\cdot e^{-jw_0(N\tau)} \\ \vdots \\ f(-N\tau)\cdot e^{-jw_M(-N\tau)} + \cdots + f(0\cdot\tau)\cdot e^{jw_M(0\cdot\tau)} + \cdots + f(N\tau)\cdot e^{-jw_M(N\tau)} \end{bmatrix}$$

$$= \tau \cdot \begin{bmatrix} f(-N\tau) & \cdots & f(0\tau) & \cdots & f(N\tau) \end{bmatrix} \cdot \begin{bmatrix} e^{-jw_{-M}(-N\tau)} & \cdots & e^{-jw_0(-N\tau)} & \cdots & e^{-jw_M(-N\tau)} \\ \vdots & & \vdots & & \vdots \\ e^{-jw_{-M}(0\bullet\tau)} & \cdots & e^{-jw_0(0\cdot\tau)} & \cdots & e^{-jw_M(0\cdot\tau)} \\ \vdots & & \vdots & & \vdots \\ e^{-jw_{-M}(N\tau)} & \cdots & e^{-jw_0(N\tau)} & \cdots & e^{-jw_M(N\tau)} \end{bmatrix}$$

$$= \tau \cdot \begin{bmatrix} f(-N\tau) & \cdots & f(0\tau) & \cdots & f(N\tau) \end{bmatrix} \cdot e^{-j\cdot\begin{bmatrix} -N\tau \\ \vdots \\ 0\cdot\tau \\ \vdots \\ N\tau \end{bmatrix}\cdot\begin{bmatrix} w_{-M} & \cdots & w_0 & \cdots & w_M \end{bmatrix}}$$

Example: Find out the Fourier transform of

$$f(t) = G(t) = \begin{cases} 1, |t| \le 1 \\ 0, |t| > 1 \end{cases}$$

```
tao = 0.02;
t = -2:tao:2;
f = heaviside(t+1)-heaviside(t-1);
M = 500; k=-M:M;
w1=2*pi*10;   % 10 indicates the range of the frequency to observe
W=k*w1/M;     % w1/M determines the resolution of the frequency

F = f*exp(-1i*t'*W)*tao;
F = abs(F);

subplot(2,1,1); plot(t,f);
xlabel('t'); ylabel('f(t)');
title('f(t)=u(t+1)-u(t-1)');
subplot(2,1,2); plot(W,F);
xlabel('w'); ylabel('F(w)');
title('Fourier transform of f(t)');
```

Tips:

1. You can use function fftshift to shift zero-frequency component to center of spectrum. In this way, you will obtain a bilateral amplitude spectrum that will retain all the signal energy. Therefore, there is no need to adjust the amplitude of the spectrum.
2. To find out the N that satisfy $\mathrm{N} = 2^n$, use the function nextpow2().
   Example: N=7, to find out the closest integer to 7, which is also the power of 2.

```
N=7
```

```
N = 7
```

```
N=2^nextpow2(N)
```

```
N = 8
```

3. Functions used to read and play audio is listed in Table 3.
   Table 3 Audio Function

format	description
[y,Fs]=audioread(FileName)	FileName:the audio to be read. y: sampled data. Fs: sample rate.
[y,Fs]=audioread(FileName,[Start,End])	Only the samples from Start to End is returned.
sound(y, Fs)	Play vector as sound.

System Analysis

As we mentioned in Lab 2, a dynamic system can be represented by a linear differential equation with constant coefficients, which is shown as below:

$$\sum_{i=0}^{N} a_i y^{(i)}(t) = \sum_{j=0}^{M} b_j x^{(j)}(t)$$

Do Fourier transform on both sides of the differential equation. So the frequency response of the system can be described as:

$$\mathrm{H(jw)} = \frac{Y(jw)}{X(jw)} = \frac{b_M(jw)^M + b_{M-1}(jw)^{M-1} + \cdots + b_1(jw) + b_0}{a_N(jw)^N + a_{N-1}(jw)^{N-1} + \cdots + a_1(jw) + a_0} = \frac{B(jw)}{A(jw)}$$

Function freqs is used to analyze H(jw), the format of which is:

$\mathrm{h = freqs(b, a, w)}$ or $\mathrm{[h, w] = freqs(b, a, n)}$

$\mathrm{b} = [b_M, b_{M-1}, \cdots, b_1, b_0], \mathrm{a} = [a_N, a_{N-1}, \cdots, a_1, a_0]$

w is the range of the frequency domain to observe, which is expressed as angular frequencies in rad/s.

n is the frequency point to compute the frequency response.

Example: the frequency response of an analog filter is like:

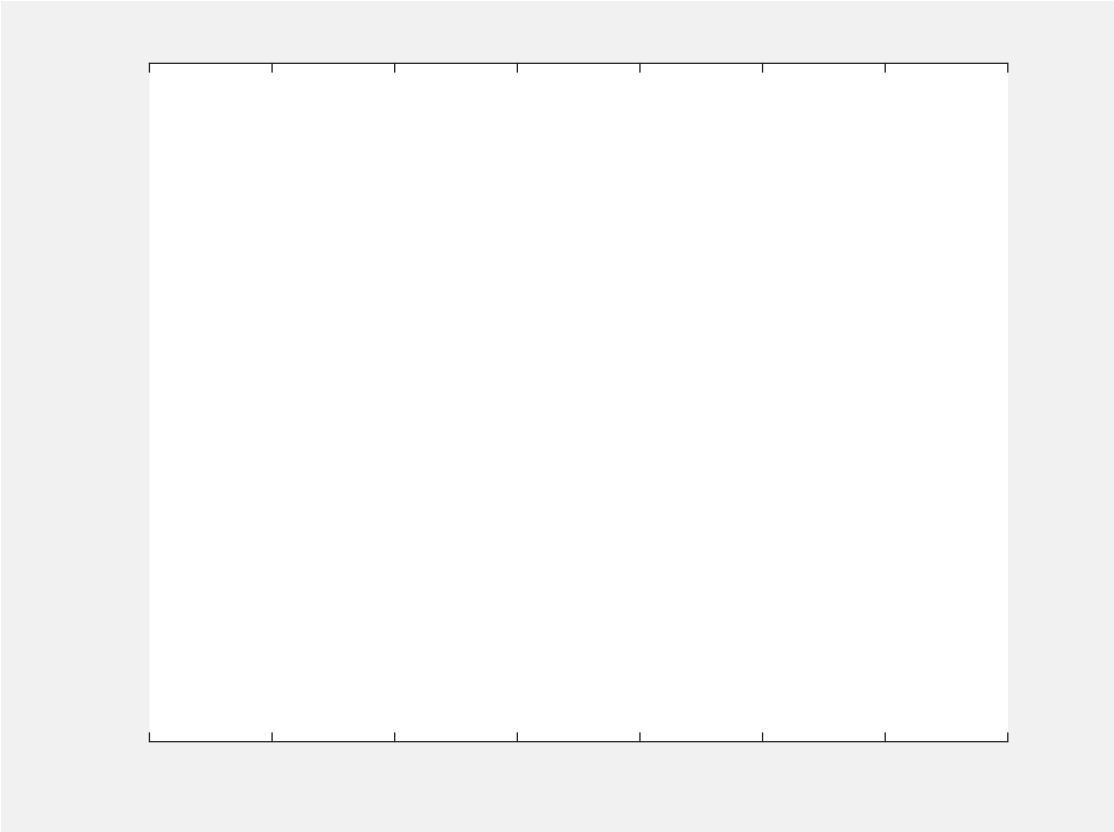
$$\mathrm{H(jw)} = \frac{b_4(jw)^4 + b_3(jw)^3 + b_2(jw)^2 + b_1(jw)^1 + b_0}{a_5(jw)^5 + a_4(jw)^4 + a_3(jw)^3 + a_2(jw)^2 + a_1(jw) + a_0}$$

$$\begin{aligned}\mathrm{b} &= [b_4, b_3, b_2, b_1, b_0] \\ &= [1.53116389e + 03, -1.29990890e - 09, 7.32176217e \\ &\quad + 12, -2.03715033e + 00, 7.71381999e + 21]\end{aligned}$$

$$\begin{aligned}\mathrm{a} &= [a_5, a_4, a_3, a_2, a_1, a_0] \\ &= [1, 3.47913978\mathrm{e} + 04, 1.87590501\mathrm{e} + 09, 4.03313474\mathrm{e} + 13, 7.97671668\mathrm{e} \\ &\quad + 17, 7.71381999\mathrm{e} + 21]\end{aligned}$$

So its frequency response can be calculated as:

```
b=[1.53116389e+03,-1.29990890e-09,7.32176217e+12,-
2.03715033e+00,7.71381999e+21];
a=[1,3.47913978e+04,1.87590501e+09,4.03313474e+13,7.97671668e+17,7.71
381999e+21];
w=linspace(1,14000*2*pi,1000);
H=freqs(b,a,w); plot(w/(2*pi)/1000, abs(H)); title('Magnitude
Response');xlabel('kHz');
```



Obviously it a low-pass filter.