신호 및 시스템
Project #2

전기전자공학부 2018142023 조성민

1. Properties of Fourier Transform

1) Plot the cosine signal x_1 with $f_1 = 0.5Hz$ and the cosine signal x_2 with $f_2 = 10Hz$ in both time and frequency domain.

2) Prove the properties of Fourier transform using the x_1 and x_2 signal in time and frequency domain. You can choose any constant value.

i. Linearity

$$a x_1(t) + b x_2(t) \leftrightarrow a X_1(f) + b X_2(f)$$

ii. Scale Change

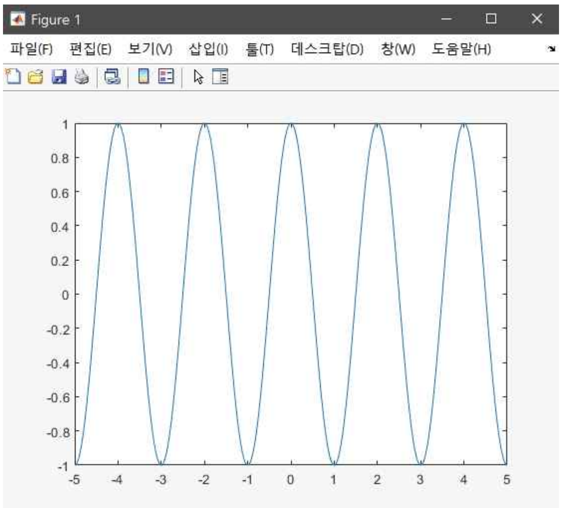
$$x_1(\alpha t) \leftrightarrow \frac{1}{|\alpha|} X_1(\frac{f}{\alpha})$$

1-1

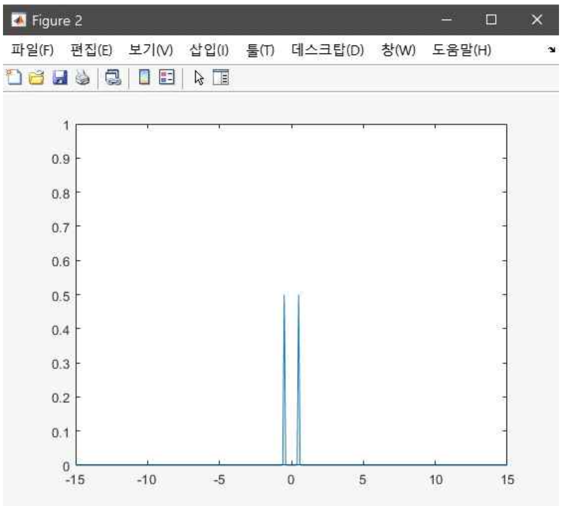
<코드>

```
fs=1000;                              % Sampling frequency 설정
t1=-5:1/fs:5;                         % x1(t)의 범위
f1=0.5;                               % f1=0.5Hz 설정
L1=length(t1);                        % x1(t) 범위의 길이
x1=cos(2*pi*f1*t1);                   % x1(t) 설정

X1=fft(x1)./L1;                       % x1(t)의 FT 및 Normalization
P1=abs(fftshift(X1));                 % X1(f) 재배열
f1=linspace(-fs/2, fs/2, L1);         % P1(f)의 범위

figure(1)                             % x1(t) 그래프
plot(t1,x1)

figure(2)                             % P1(f) 그래프
plot(f1,P1)
xlim([-15 15])                        % 그래프 x축 표시 제한
ylim([0 1])                           % 그래프 y축 표시 제한


t2=-0.25:1/fs:0.25;                   % x2(t)의 범위
f2=10;                                % f2=10Hz 설정
L2=length(t2);                        % x2(t) 범위의 길이
x2=cos(2*pi*f2*t2);                   % x2(t) 설정

X2=fft(x2)./L2;                       % x2(t)의 FT 및 Normalization
P2=abs(fftshift(X2));                 % X2(f) 재배열
f2=linspace(-fs/2, fs/2, L2);         % P2(f)의 범위

figure(3)                             % x2(t) 그래프
plot(t2,x2)

figure(4)                             % P2(f) 그래프
plot(f2,P2)
xlim([-300 300])                      % 그래프 x축 표시 제한
ylim([0 1])                           % 그래프 y축 표시 제한
```

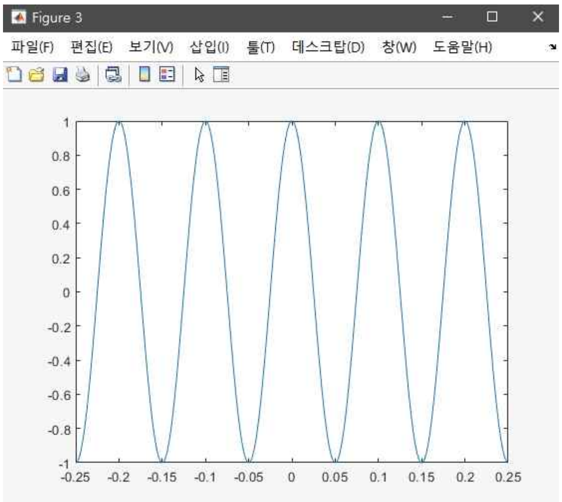
<결과>



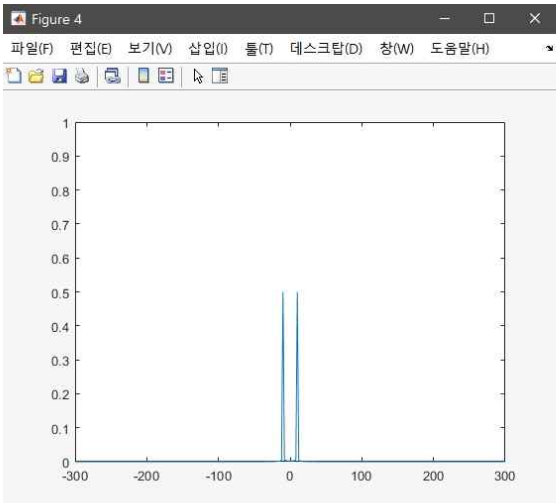
$$x_1(t) = \cos(2\pi f_1 t)$$



$$X_1(f) = \frac{1}{2}\delta(f - f_1) + \frac{1}{2}\delta(f + f_1)$$



$$x_2(t) = \cos(2\pi f_2 t)$$



$$X_2(f) = \frac{1}{2}\delta(f - f_2) + \frac{1}{2}\delta(f + f_2)$$

-우선 모든 문제에서 sampling frequency를 넉넉하게 잡았다.

-Matlab 기능 중 고속 푸리에 변환 함수 fft()를 사용했는데, 이 함수 자체로는 원하는 주파수에서 원하는 값을 얻을 수가 없어 normalization과 주파수의 재배열 과정을 거쳤다.

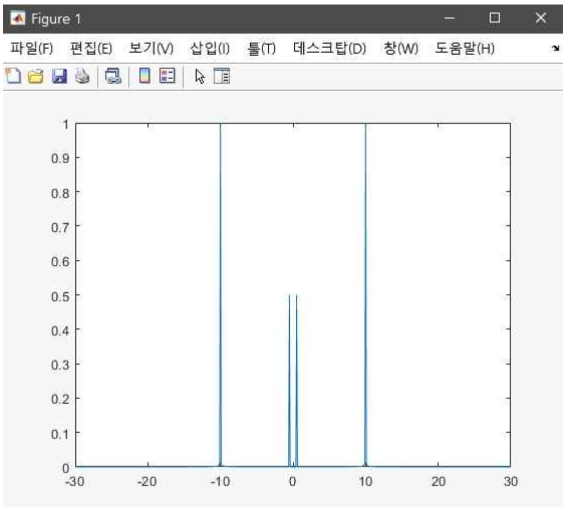
-Fourier Transform Theorems 중 modulation 이론이 있는데, 식 $x(t)\cos(2\pi f_0 t) \leftrightarrow \frac{1}{2}X(f - f_0) + \frac{1}{2}X(f + f_0)$에서 $x(t) = 1$이라고 설정하면 $1 \leftrightarrow \delta(f)$ 관계를 통해 $\cos(2\pi f_0 t) = \frac{1}{2}\delta(f - f_0) + \frac{1}{2}\delta(f + f_0)$임을 알 수 있고, matlab을 통해 f_1, f_2에 맞게 잘 구현했음을 확인할 수 있다.

1-2-Linearity

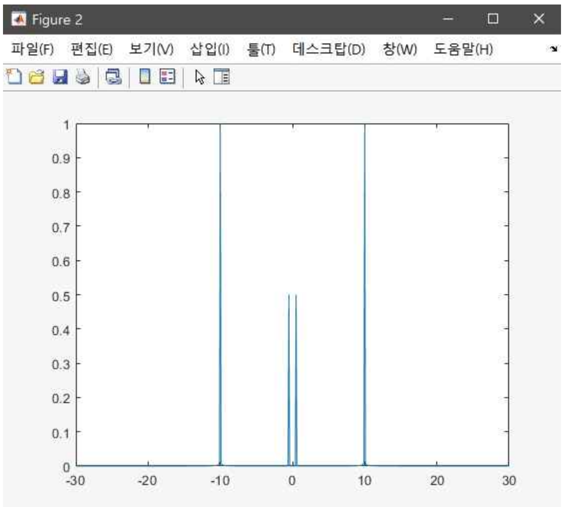
<코드>

```
fs=1000;                          % Sampling frequency 설정
a=1;                              % 상수a=1 설정
b=2;                              % 상수b=2 설정
t=-5:1/fs:5;                      % x1(t), x2(t)의 범위
f1=0.5;                           % f1=0.5Hz 설정
f2=10;                            % f2=10Hz 설정
L=length(t);                      % x1(t), x2(t) 범위의 길이
x1=cos(2*pi*f1*t);                % x1(t) 설정
x2=cos(2*pi*f2*t);                % x2(t) 설정
x=a*x1+b*x2;                      % x(t)=ax1(t)+bx2(t) 설정

X1=fft(x1)./L;                    % x1(t)의 FT 및 Normalization
P1=abs(fftshift(X1));             % X1(f) 재배열
X2=fft(x2)./L;                    % x2(t)의 FT 및 Normalization
P2=abs(fftshift(X2));             % X2(f) 재배열
X=fft(x)./L;                      % x(t)의 FT 및 Normalization
PX=abs(fftshift(X));              % X(f) 재배열
f=linspace(-fs/2, fs/2, L);       % P1(f), P2(f), PX(f)의 범위

figure(1)                         % PX(f) 그래프
plot(f,PX)
xlim([-30 30])                    % 그래프 x축 표시 제한
ylim([0 1])                       % 그래프 y축 표시 제한

figure(2)                         % aP1(f)+bP2(f) 그래프
plot(f,a*P1+b*P2)
xlim([-30 30])                    % 그래프 x축 표시 제한
ylim([0 1])                       % 그래프 y축 표시 제한
```

<결과>



$ax_1(t)+bx_2(t)\,FT$



$aX_1(f)+bX_2(f)$

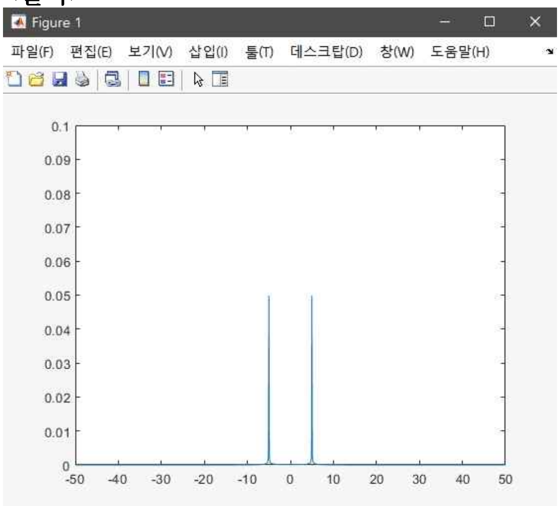
-Fourier Transform Theorems 중 linearity 이론은 homogenity와 additivity, 즉 superposition을 만족한다.
-$ax_1(t)+bx_2(t)$함수를 만들고 푸리에 변환을 한 것과 $x_1(t)\leftrightarrow X_1(f)$, $x_2(t)\leftrightarrow X_2(f)$ 과정을 거친 후 각각 a와 b를 곱하고 더한 것이 동일한 그래프를 나타내는 것을 보아 matlab으로 linearity 이론이 입증되었음을 알 수 있다.

1-2-Scale Change

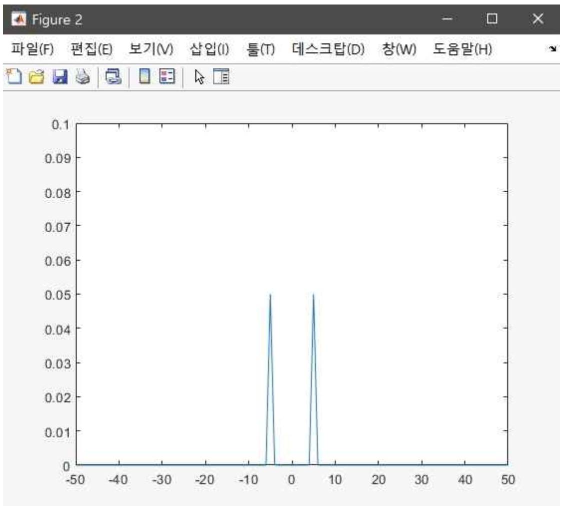
<코드>

```
fs1=1000;                      % Sampling frequency 설정1
fs2=100;                       % Sampling frequency 설정2
a=10;                          % 상수a=10 설정
f=0.5;                         % f=0.5Hz 설정
t1=-5:1/fs1:5;                 % x(t)의 범위
t2=-5:1/fs2:5;                 % xa(t)의 범위
L1=length(t1);                 % x(t) 범위의 길이
L2=length(t2);                 % xa(t) 범위의 길이
x=cos(2*pi*f*t1);              % x(t) 설정
xa=cos(2*pi*a*f*t2);           % xa(t) 설정

X=fft(x)./L1;                  % x(t)의 FT 및 Normalization
PX=abs(fftshift(X));           % X(f) 재배열
f1=linspace(-fs1/2, fs1/2, L1);  % PX(f)의 범위
XA=fft(xa)./L1;                % xa(t)의 FT 및 Normalization
PXA=abs(fftshift(XA));         % XA(f) 재배열
f2=linspace(-fs2/2, fs2/2, L2);  % PXA(f)의 범위

figure(1)                      % PXA(f) 그래프
plot(f2,PXA)
xlim([-50 50])                 % 그래프 x축 표시 제한
ylim([0 0.1])                  % 그래프 y축 표시 제한

figure(2)                      % PX(f/a)/a 그래프
plot(f1.*a,PX./a)
xlim([-50 50])                 % 그래프 x축 표시 제한
ylim([0 0.1])                  % 그래프 y축 표시 제한
```

<결과>



$x_1(at)\,FT$



$\frac{1}{|a|}X_1(\frac{f}{a})$

-Sampling frequency를 2개로 설정한 이유는 $x(t)$가 $x(at)$로 변함에 따라 t축의 간격 또한 같이 바꾸고 동일한 frequency에서 같은 값을 갖도록 하기 위함이다.

-$x_1(at)$함수를 만들고 푸리에 변환을 한 것과 $x_1(t) \leftrightarrow X_1(f)$ 과정을 거친 후 a를 f축에 곱하고 $X_1(\frac{f}{a})$에 나눈 것과 동일한 그래프를 나타내는 것을 보아 matlab으로 scale change 이론이 입증되었음을 알 수 있다.

2. **Fourier Transform of Sinc Function**

1) Plot the $x(t) = sinc(t)$ in both time and frequency domain. Provide a discussion on the result.

2) Prove the properties of Fourier transform using the x signal in time and frequency domain. You can choose any constant value.

   i. **Modulation ($x(t) = sinc(t)$)**

   $$x(t)cos(2\pi f_0 t) \leftrightarrow \frac{1}{2}X(f - f_0) + \frac{1}{2}X(f + f_0)$$

   ii. **Scale Change**

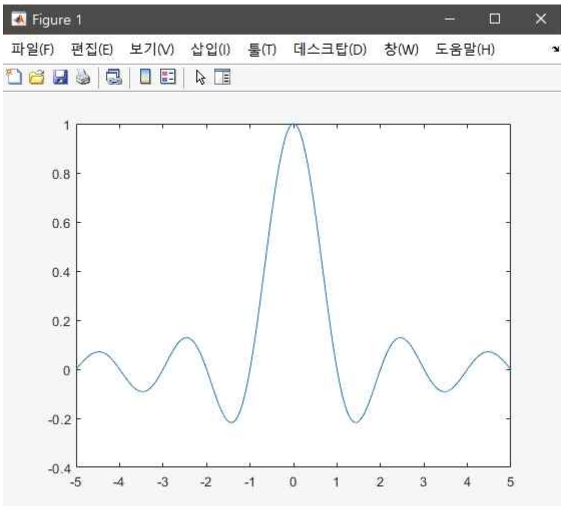
   $$x(\alpha t) \leftrightarrow \frac{1}{|\alpha|}X(\frac{f}{\alpha})$$

2-1

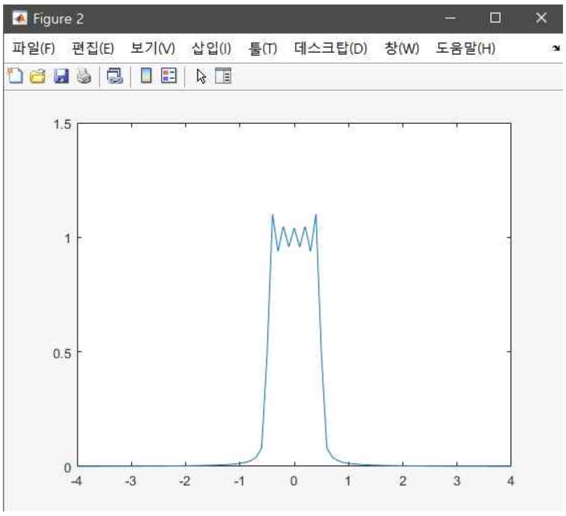
<코드>

```
fs=1000;                              % Sampling frequency 설정
t=-5:1/fs:5;                          % x(t)의 범위
L=length(t);                          % x(t) 범위의 길이
x=sinc(t);                            % x(t) 설정

X=fft(x)./fs;                         % x(t)의 FT 및 Normalization
P=abs(fftshift(X));                   % X(f) 재배열
f=linspace(-fs/2, fs/2, L);           % P(f)의 범위

figure(1)                             % x(t) 그래프
plot(t,x)

figure(2)                             % P(f) 그래프
plot(f,P)
xlim([-4 4])                          % 그래프 x축 표시 제한
ylim([0 1.5])                         % 그래프 y축 표시 제한
```

<결과>



$x(t) = \mathrm{sinc}(t)$



$X(f) = \Pi(f)$

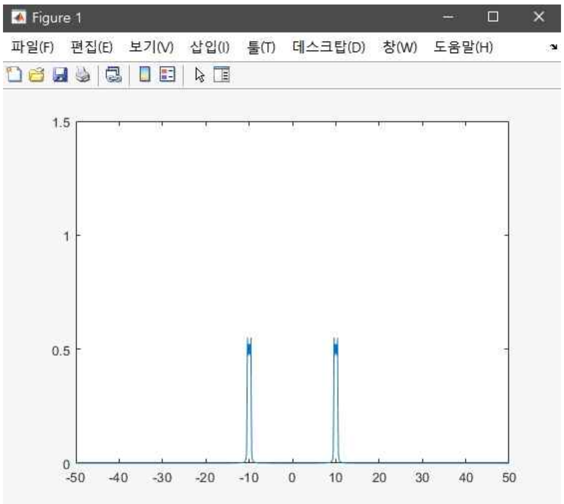
-푸리에 변환 중 $\mathrm{sinc}(t) \leftrightarrow \Pi(f)$ 관계가 성립하는데, 불연속 신호에서 발생하는 노이즈 현상인 Gibbs phenomenon으로 인해 -1과 1 사이에서 불안정한 1값을 가지는 것을 확인할 수 있다.

2-2-Modulation

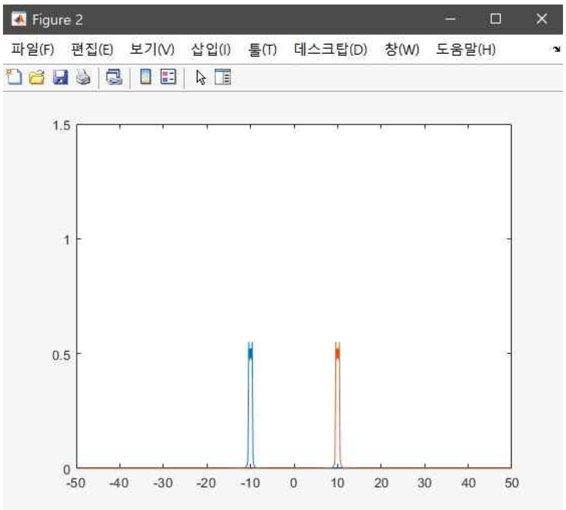
<코드>

```
fs=1000;                          % Sampling frequency 설정
f0=10;                            % f0=10Hz 설정
t=-5:1/fs:5;                      % x(t)의 범위
L=length(t);                      % x(t) 범위의 길이
x=sinc(t);                        % x(t) 설정
xc=x.*cos(2*pi*f0*t);             % xc(t) 설정

X=fft(x)./fs;                     % x(t)의 FT 및 Normalization
P=abs(fftshift(X));               % X(f) 재배열
XC=fft(xc)./fs;                   % xc(t)의 FT 및 Normalization
PC=abs(fftshift(XC));             % XC(f) 재배열
f=linspace(-fs/2, fs/2, L);       % P(f), PC(f)의 범위

figure(1)                         % PC(f) 그래프
plot(f,PC)
xlim([-50 50])                    % 그래프 x축 표시 제한
ylim([0 1.5])                     % 그래프 y축 표시 제한

figure(2)                         % P(f-f0)/2+P(f+f0)/2 그래프
plot(f-f0,P./2)                   % P(f-f0)/2
hold on
plot(f+f0,P./2)                   % P(f+f0)/2
xlim([-50 50])                    % 그래프 x축 표시 제한
ylim([0 1.5])                     % 그래프 y축 표시 제한
```

<결과>



$x(t)\cos(2\pi f_0 t)\,FT$



$\frac{1}{2}X(f-f_0)+\frac{1}{2}X(f+f_0)$

-Euler's equation으로 식 $\cos(2\pi f_0 t)=\frac{1}{2}e^{j(2\pi f_0 t)}+\frac{1}{2}e^{-j(2\pi f_0 t)}$이 성립한다.

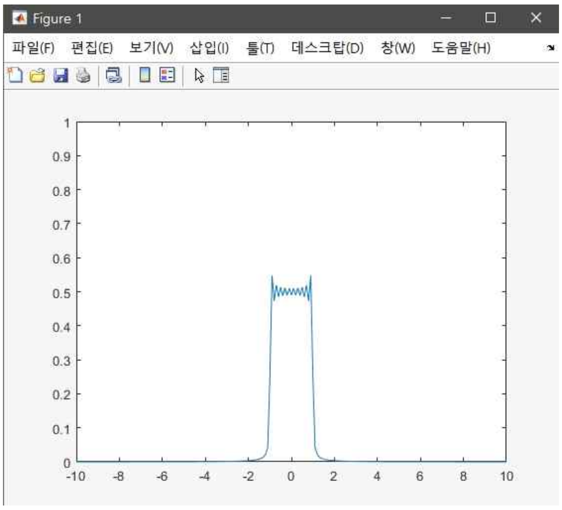
-$x(t)\cos(2\pi f_0 t)$함수를 만들고 푸리에 변환을 한 것과 $x(t)\leftrightarrow X(f)$ 과정을 거친 후 수동으로 f축에 f_0를 덧셈, 뺄셈하고 2로 나눈 것이 동일한 그래프를 나타내는 것을 보아 matlab으로 modulation 이론이 입증되었음을 알 수 있다.

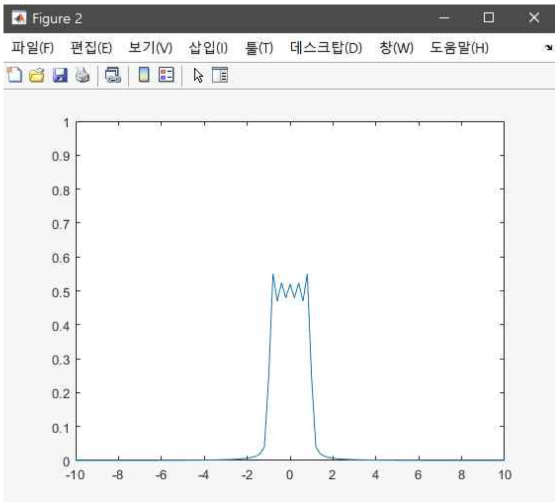
2-2-Scale Change

<코드>

```
fs1=1000;                          % Sampling frequency 설정1
fs2=500;                           % Sampling frequency 설정2
a=2;                               % 상수a=2 설정
t1=-5:1/fs1:5;                     % x(t)의 범위
t2=-5:1/fs2:5;                     % xa(t)의 범위
L1=length(t1);                     % x(t) 범위의 길이
L2=length(t2);                     % xa(t) 범위의 길이
x=sinc(t1);                        % x(t) 설정
xa=sinc(a*t2);                     % xa(t) 설정

X=fft(x)./fs1;                     % x(t)의 FT 및 Normalization
PX=abs(fftshift(X));               % X(f) 재배열
f1=linspace(-fs1/2, fs1/2, L1);    % PX(f)의 범위
XA=fft(xa)./fs2;                   % xa(t)의 FT 및 Normalization
PXA=abs(fftshift(XA));             % XA(f) 재배열
f2=linspace(-fs2/2, fs2/2, L2);    % PXA(f)의 범위

figure(1)                          % PXA(f) 그래프
plot(f2,PXA)
xlim([-10 10])                     % 그래프 x축 표시 제한
ylim([0 1])                        % 그래프 y축 표시 제한

figure(2)                          % PX(f/a)/a 그래프
plot(f1.*a,PX./a)
xlim([-10 10])                     % 그래프 x축 표시 제한
ylim([0 1])                        % 그래프 y축 표시 제한
```

<결과>





$x(at)\,FT$ $\qquad\qquad$ $\frac{1}{|a|}X(\frac{f}{a})$

-문제 1-2에서 matlab으로 입증했던 scale change 이론과 동일하게, $x(at)$함수를 만들고 푸리에 변환을 한 것과 $x(t) \leftrightarrow X(f)$ 과정을 거친 후 a를 f축에 곱하고 $X(\frac{f}{a})$에 나눈 것과 동일한 그래프를 나타내는 것을 보아 matlab으로 scale change 이론이 입증되었음을 알 수 있다.

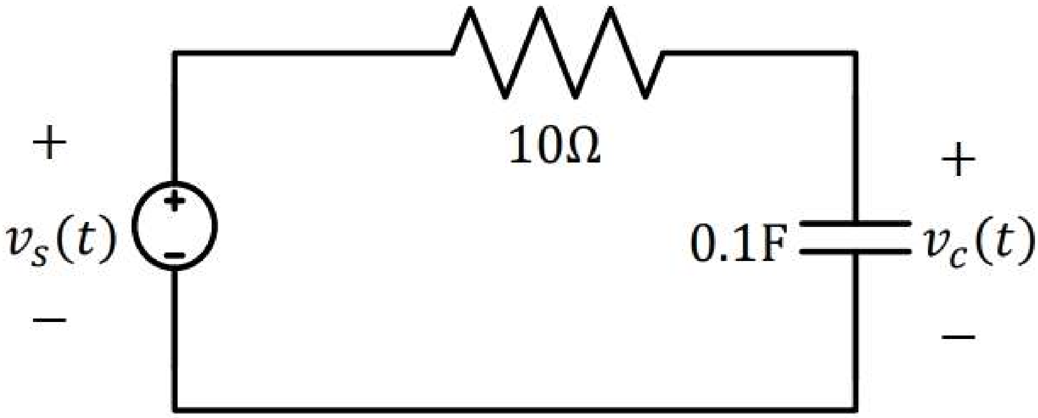
3. Laplace Transform

1) Show and prove these transform pairs using MATLAB.

   i. $u(t) \leftrightarrow 1/s$

   ii. $y(t) = [e^{t} + e^{-5t} - 2\cos(2t)]u(t) \leftrightarrow Y(s) = \frac{-4s^2+18s+16}{s^4+4s^3-s^2+16s-20}$

2) Consider the RC filter shown in below.



   i. Find and plot the output voltage $v_c(t)$ for $t \geq 0$, when input signal $v_s(t) = u(t)$.

   ii. Find and plot the transfer function in time domain and the bode plot.

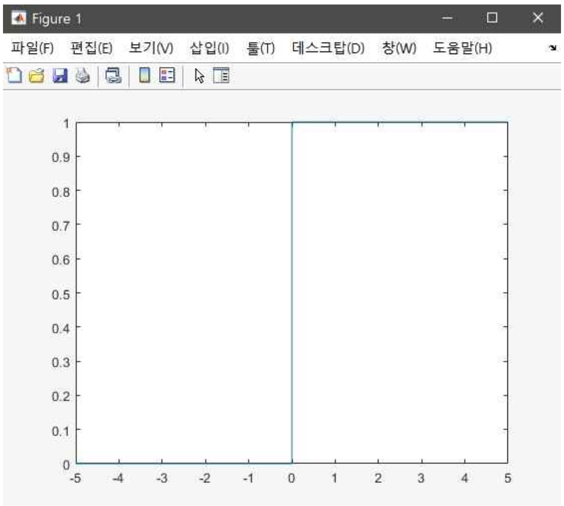
   iii. Plot the pole-zero plots and determine whether the system is stable, marginally stable, or unstable.

3-1-1

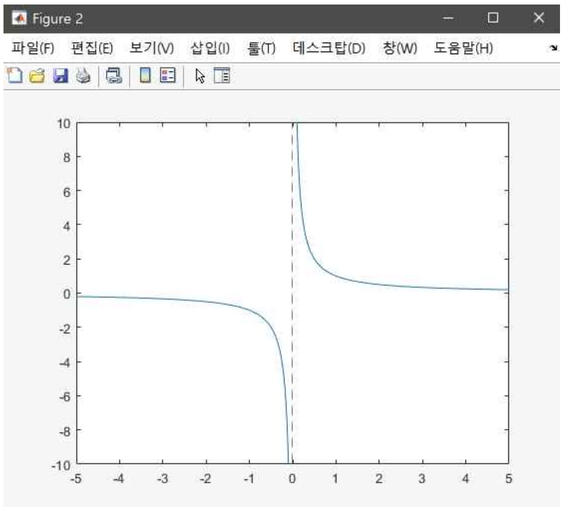
<코드>

```
syms t
u=heaviside(t);       % u(t) 설정
U=laplace(u);         % U(f) 설정

figure(1)             % u(t) 그래프
fplot(u)

figure(2)             % U(f) 그래프
fplot(U)
```

<결과 & command window>



$u(t)$



$1/s$

```
>> syms t
>> u=heaviside(t);
>> laplace(u)

ans =

1/s
```

command window

$$\mathcal{L}\{u(t)\} = \int_{-\infty}^{\infty} u(t)e^{-st}dt = \int_{0}^{\infty} e^{-st}dt = \left[-\frac{e^{-st}}{s}\right]_{0}^{\infty} = \frac{1}{s}$$

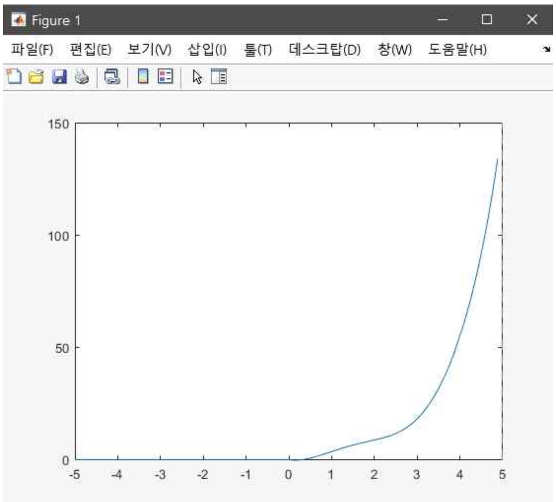
-Command window에 나타난 식과 손으로 직접 유도한 식이 동일하므로 $u(t) \leftrightarrow 1/s$ 관계가 입증되었음을 알 수 있다.

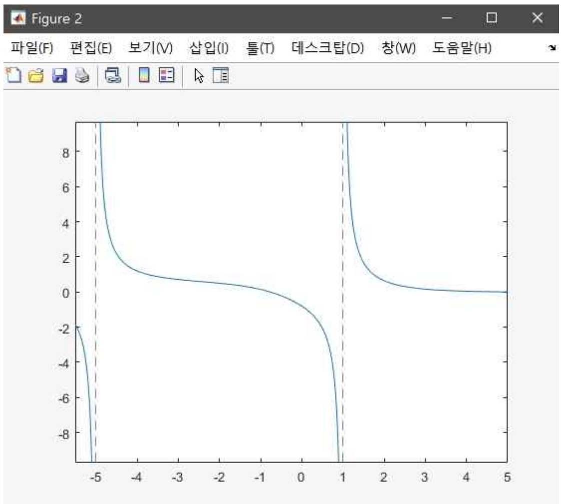
3-1-2

<코드>

```
syms t
y=(exp(t)+exp(-5*t)-2*cos(2*t))*heaviside(t);     % y(t) 설정
Y=laplace(y);                                      % Y(f) 설정

figure(1)                                          % y(t) 그래프
fplot(y)

figure(2)                                          % Y(f) 그래프
fplot(Y)
```

<결과 & command window>





$$y(t) = [e^{t} + e^{-5t} - 2\cos(2t)]u(t) \qquad Y(s) = \frac{-4s^2 + 18s + 16}{s^4 + 4s^3 - s^2 + 16s - 20}$$

```
>> syms t
>> y=(exp(t)+exp(-5*t)-2*cos(2*t))*heaviside(t);
>> laplace(y)

ans =

1/(s - 1) + 1/(s + 5) - (2*s)/(s^2 + 4)
```

command window

$$y(t) = [e^{t} + e^{-5t} - 2\cos(2t)]\,u(t)$$

$$\mathcal{L}\{y(t)\} = \int_0^\infty e^{(1-s)t}\,dt + \int_0^\infty e^{(-5-s)t}\,dt - \left\{\int_0^\infty e^{(j2-s)t}\,dt + \int_0^\infty e^{(-j2-s)t}\,dt\right\}$$

$$= \left[\frac{e^{(1-s)t}}{1-s}\right]_0^\infty + \left[\frac{e^{(-5-s)t}}{-5-s}\right]_0^\infty - \left\{\left[\frac{e^{(j2-s)t}}{j2-s}\right]_0^\infty + \left[\frac{e^{(-j2-s)t}}{-j2-s}\right]_0^\infty\right\}$$

$$= \frac{1}{s-1} + \frac{1}{s+5} - \left\{-\frac{1}{j2-s} + \frac{1}{j2+s}\right\}$$

$$= \frac{1}{s-1} + \frac{1}{s+5} - \frac{2s}{s^2+4}$$

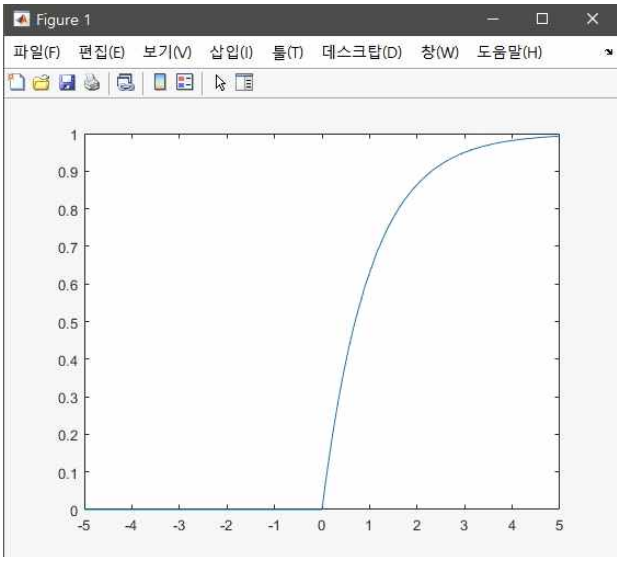
-Command window에 나타난 식과 손으로 직접 유도한 식이 동일하므로 $[e^{t} + e^{-5t} - 2\cos(2t)]u(t) \leftrightarrow \frac{-4s^2 + 18s + 16}{s^4 + 4s^3 - s^2 + 16s - 20} = \frac{1}{s-1} + \frac{1}{s+5} - \frac{2s}{s^2+4}$ 관계가 입증되었음을 확인할 수 있다.

3-2-1

<코드>

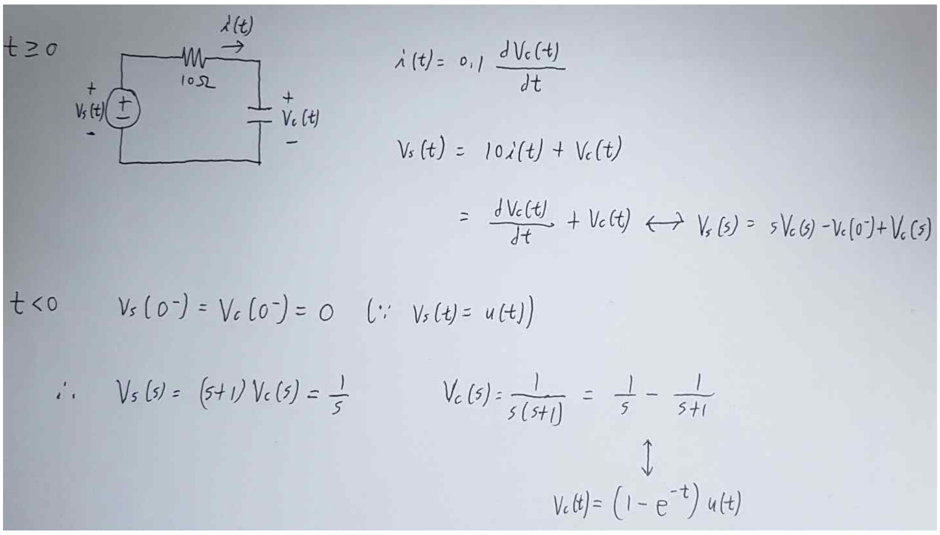
```
syms t
vc=(1-exp(-t))*heaviside(t);      % vc(t) 설정

figure(1)                         % vc(t) 그래프
fplot(vc)
```

<결과>



$v_c(t) = (1 - e^{-t})u(t)$

-$v_c(t)$를 유도하는 과정은 아래와 같다.

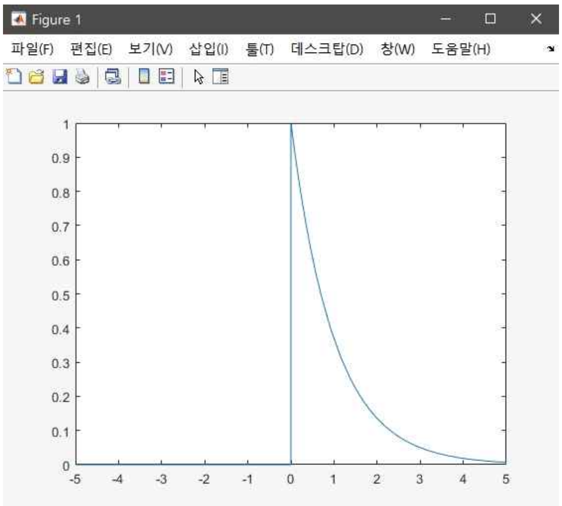


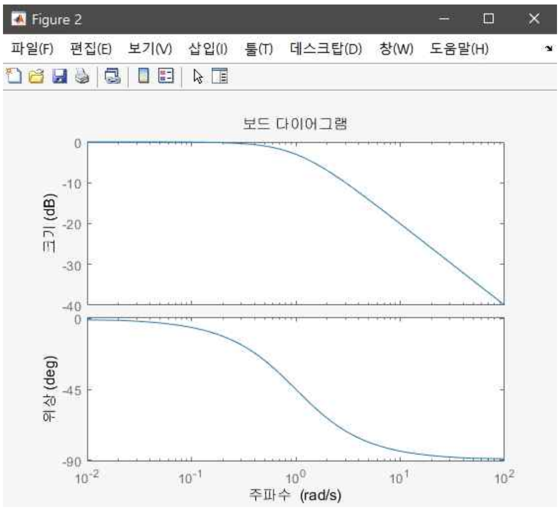
3-2-2

<코드>

```
syms t
h=exp(-t)*heaviside(t);          % h(t) 설정
H=tf([1],[1 1]);                 % H(s) 설정

figure(1)                        % h(t) 그래프
fplot(h)

figure(2)                        % H(s) bode plot
bode(H)
```

<결과>





$h(t)=e^{-t}u(t)$ $\qquad$ $H(s)=\dfrac{1}{s+1}$

$$-H(s)=\frac{V_c(s)}{V_s(s)}=\frac{\frac{1}{s}-\frac{1}{s+1}}{\frac{1}{s}}=1-\frac{s}{s+1}=\frac{1}{s+1}$$

$-v_c(t)=v_s(t)*h(t)\leftrightarrow V_c(s)=V_s(s)H(s)$ (LT convolution 이론)

$-H(s)=\dfrac{1}{s+1}\leftrightarrow h(t)=e^{-t}u(t)$ ($\because$ LT s shift 이론)

-Magnitude Plot: $w=1(rad/s)$에서 pole값을 가지며 크기가 $-3dB$로 줄어든다.

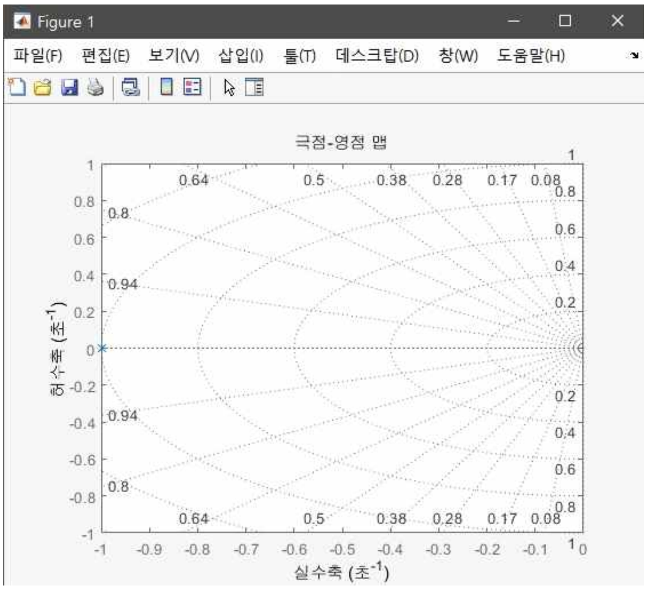
-Phase Plot: $w=1(rad/s)$를 중심으로 0° 부터 90° 까지 octave당 45° 씩 감소한다.

3-2-3

<코드>

```
H=tf([1],[1 1]);        % H(s)=1/(s+1) 설정
pzmap(H)                % H(s) pole-zero plot
grid on
```

<결과>



-The system is defined by the transfer function $H(s) = \dfrac{1}{s+1}$.

-Pole이 LHP(Left Half Plane)인 s=-1에 위치해 있으므로 stable이다.

4. Sampling Theorem and Nyquist Rate

Given sinusoidal function $myfunc(t)$, which has three different frequency components cosine function.

1) Plot the signal in time domain. ($x(t) = myfunc(t)$)

2) Plot the Fourier transform of your own function. $X(f)$, in frequency domain. And you must analyze it using mathematical method and answer what is the $myfunc(t)$.

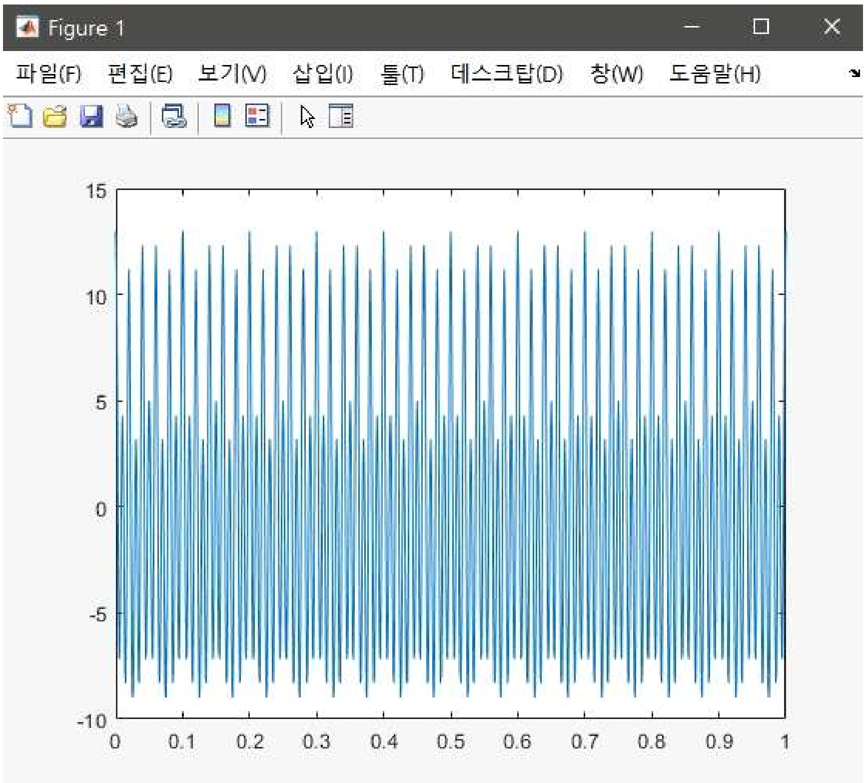
3) Do inverse Fourier transform of $X(f)$ using built-in function $\boldsymbol{ifft}()$. You can get $myfunc(t)$ through inverse Fourier transform. Your signal may not reconstruct exactly. If your signal reconstructs exactly write the result and analyze it.

Q 4-1

<코드>

```
t=0:0.001:1;           % x(t)의 범위
x=myfunc(t);           % x(t) 설정

figure(1)              % x(t) 그래프
plot(t,x)
```

<결과>



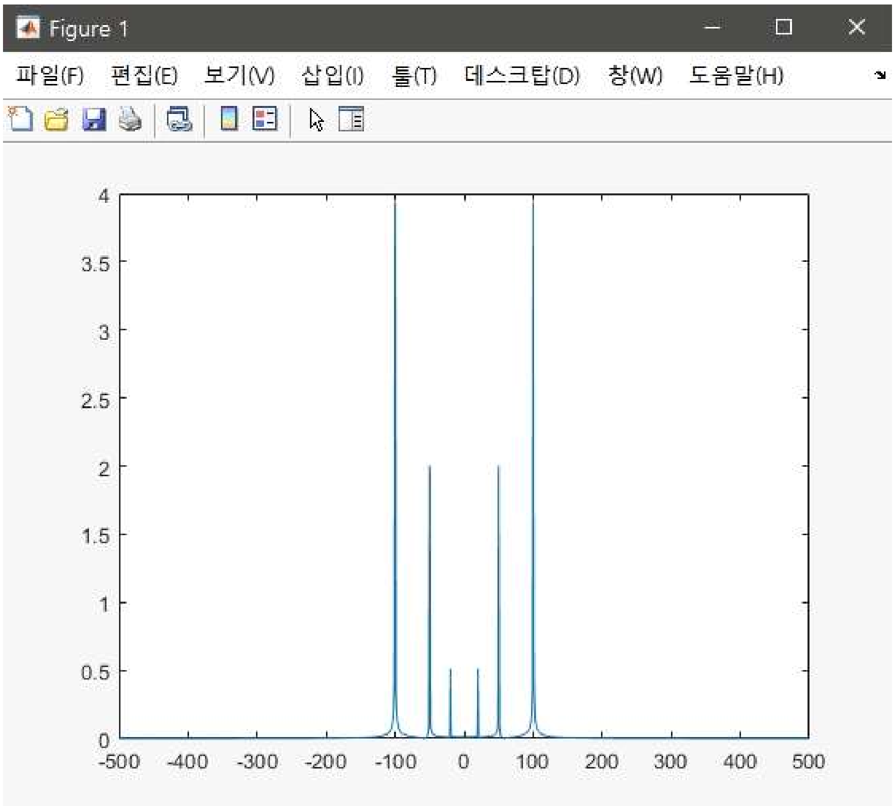
-미지의 myfunc(t) 그래프이다.

Q 4-2

<코드>

```
fs=1000;                    % Sampling frequency 설정
t=0:1/fs:1;                 % x(t)의 범위
L=length(t);                % x(t) 범위의 길이
x=myfunc(t);                % x(t) 설정

X=fft(x)./L;                % x(t)의 FT 및 Normalization
P=abs(fftshift(X));         % X(f) 재배열
f=linspace(-fs/2,fs/2,L);   % P(f)의 범위

figure(1)                   % P(f) 그래프
plot(f,P)
```

<결과>



-$X(f)=\frac{1}{2}\delta(f-20)+\frac{1}{2}\delta(f+20)+2\delta(f-50)+2\delta(f+50)+4\delta(f-100)+4\delta(f+100)$

-Fourier Transform Theorems 중 modulation 이론에 따라

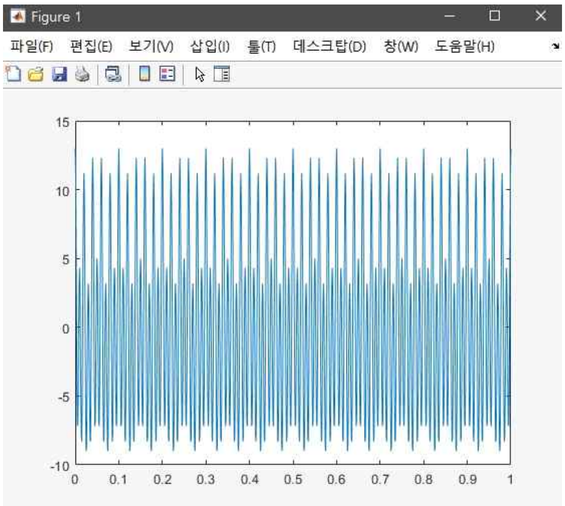
$\frac{1}{2}\delta(f-20)+\frac{1}{2}\delta(f+20)\leftrightarrow\cos(2\pi\times20\times t)$, $2\delta(f-50)+2\delta(f+50)\leftrightarrow4\cos(2\pi\times50\times t)$, $4\delta(f-100)+4\delta(f+100)\leftrightarrow8\cos(2\pi\times100\times t)$이므로, myfunc(t)인 x(t)는 $x(t)=\cos(40\pi t)+4\cos(100\pi t)+8\cos(200\pi t)$이다.

Q 4-3

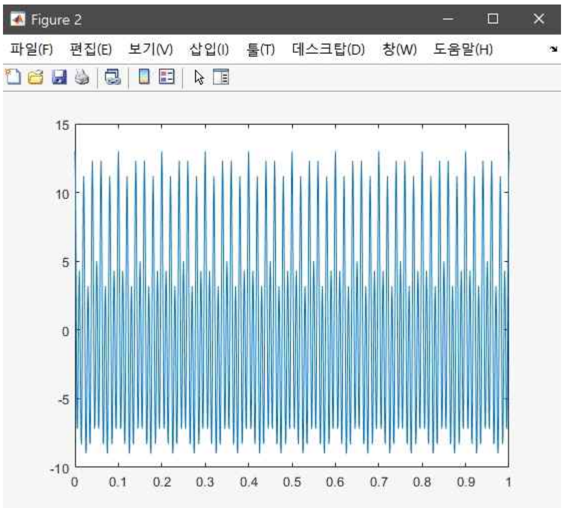
<코드a>

```
fs=1000;                 % Sampling frequency 설정
t=0:1/fs:1;              % x(t)의 범위
L=length(t);             % x(t) 범위의 길이
x=myfunc(t);             % x(t) 설정

X=fft(x)./L;             % x(t)의 FT 및 Normalization
Xi=ifft(X).*L;           % X(t)의 IFT

figure(1)                % x(t) 그래프
plot(t,x)
ylim([-10 15])           % 그래프 y축 표시 제한

figure(2)                % Xi(t) 그래프
plot(t,Xi)
ylim([-10 15])           % 그래프 y축 표시 제한
```

<결과a>



$x(t)$

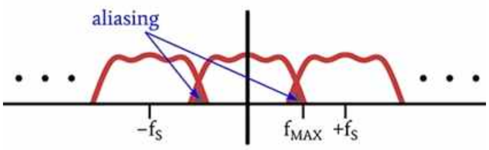


$X(f)$ IFT

-본래 아날로그 함수 $x(t)$와 내장함수 ifft()를 사용하여 X(f)의 inverse laplace transform이 동일하다는 것을 확인할 수 있다.

-Nyquist Rate: 신호의 최고 주파수의 2배 속도를 의미한다.($2f_{max}$)

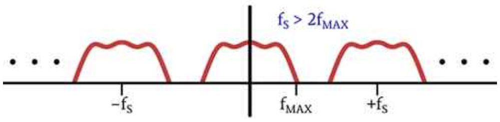
-Shannon's Sampling Theorem: 시스템이 Nyquist Rate을 초과하는 속도로 아날로그 신호를 균일하게 sampling하면 sampling을 통해 생성된 이산 값에서 본래 아날로그 신호를 완벽하게 복구할 수 있다.



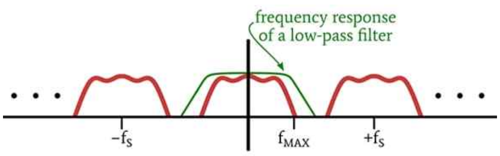
-Sampling frequency가 $2f_{max}$ 미만일 경우 subspectra가 겹치는 aliasing이 발생한다.

-Sampling 속도에 상관없이 A/D(analog to digital) 변환을 할 때마다 aliases가 생성되는데, sampling된 데이터에서 aliases가 아날로그 신호의 일부가 되면서 D/A 변환을 하면 본래 신호와 동일하지 않게 된다.

-따라서 아날로그 신호를 완벽히 복구하려면 aliasing이 발생하지 않도록 해야 한다.



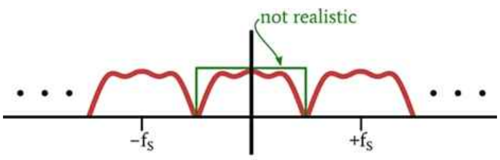
-Nyquist Rate을 초과하여 sampling을 하더라도 aliases는 여전히 존재하지만 authentic spectrum과 aliased spectra 사이에 간격이 있게 된다.



-신호의 재구성은 적절한 sampling frequency에서만 시작되며, low-pass filtering이 기본 요구사항이 된다.

-D/A 변환을 한 후 low-pass filter를 적용하여 aliases를 제거하고 본래 아날로그 신호로 복구할 수 있게 되는데, 이때 이 필터를 reconstruction filter라고 부른다.

-추가적인 내용으로, digital sampling을 통해 aliases를 제거할 수는 없으며, reconstruction filter는 anti-aliasing filter(A/D 변환 전에 적용됨)가 아니다.



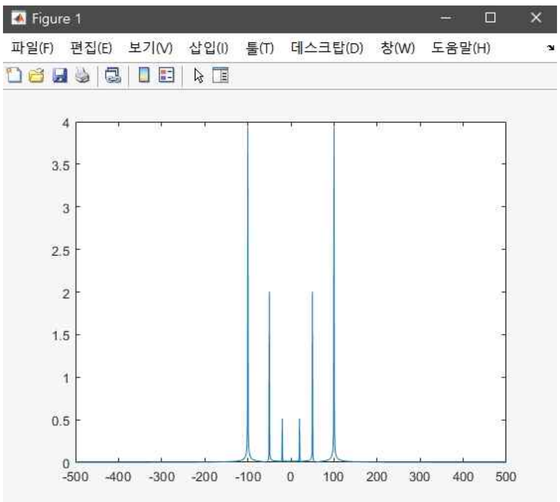
-물리적 영역에서는 이상적인 수학적 영역에서처럼 brick-wall response가 나타나지 않게 된다. 따라서 효과적인 reconstruction filter를 구축하기 위해 sampling frequency를 Nyquist Rate보다 훨씬 크게 지정하는 oversampling를 사용한다.

<코드b>

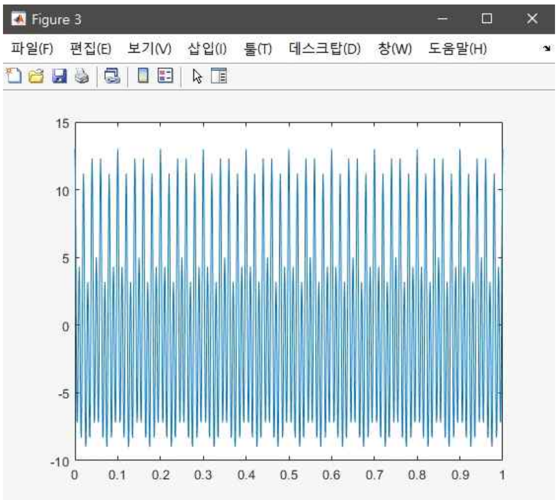
```
fs1=1000;                               % Sampling frequency 설정1
fs2=150;                                % Sampling frequency 설정2
t1=0:1/fs1:1;                           % x1(t)의 범위
t2=0:1/fs2:1;                           % x2(t)의 범위
L1=length(t1);                          % x1(t) 범위의 길이
L2=length(t2);                          % x2(t) 범위의 길이
x1=myfunc(t1);                          % x1(t) 설정
x2=myfunc(t2);                          % x2(t) 설정

X1=fft(x1)./L1;                         % x1(t)의 FT 및 Normalization
X2=fft(x2)./L2;                         % x2(t)의 FT 및 Normalization
X1i=ifft(X1).*L1;                       % X1(f)의 IFT
X2i=ifft(X2).*L2;                       % X2(f)의 IFT
P1=abs(fftshift(X1));                   % X1(f) 재배열
P2=abs(fftshift(X2));                   % X2(f) 재배열
f1=linspace(-fs1/2,fs1/2,L1);           % P1(f)의 범위
f2=linspace(-fs2/2,fs2/2,L2);           % P2(f)의 범위

figure(1)                               % P1(t) 그래프
plot(f1,P1)

figure(2)                               % P2(t) 그래프
plot(f2,P2)

figure(3)                               % X1i(t) 그래프
plot(t1,X1i)

figure(4)                               % X2i(t) 그래프
plot(t2,X2i)
```

<결과b>

-Sampling frequency=1000

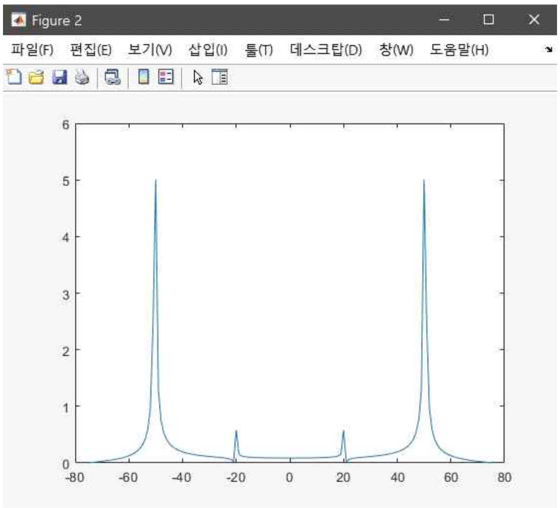


$X(f)$

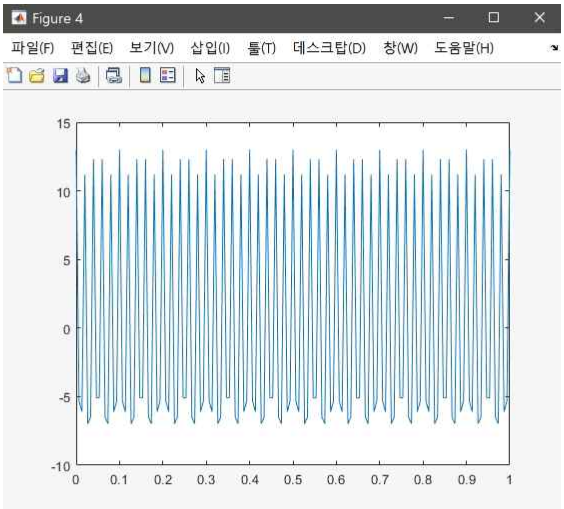


$x(t)$

-Sampling frequency=150



$X(f)$



$x(t)$

-Sampling frequency를 Nyquist Rate인 200보다 높게 1000으로 설정했을 경우 본래 $x(t)$가 원래대로 복구되는 것을 확인할 수 있다.
-하지만 Sampling frequency를 Nyquist Rate인 200보다 낮게 150으로 설정한 경우 $X(f)$의 결과가 제대로 나타나지 않은 것을 알 수 있고, 이에 따라 본래의 $x(t)$가 정상적으로 복구되지 않았음을 확인할 수 있다.

참고문헌

-https://www.allaboutcircuits.com/technical-articles/the-nyquistshannon-sampling-theorem-exceeding-the-nyquist-rate/

Appendix

```
% Q 1-1
fs=1000;                              % Sampling frequency 설정
t1=-5:1/fs:5;                         % x1(t)의 범위
f1=0.5;                               % f1=0.5Hz 설정
L1=length(t1);                        % x1(t) 범위의 길이
x1=cos(2*pi*f1*t1);                   % x1(t) 설정

X1=fft(x1)./L1;                       % x1(t)의 FT 및 Normalization
P1=abs(fftshift(X1));                 % X1(f) 재배열
f1=linspace(-fs/2, fs/2, L1);         % P1(f)의 범위

figure(1)                             % x1(t) 그래프
plot(t1,x1)

figure(2)                             % P1(f) 그래프
plot(f1,P1)
xlim([-15 15])                        % 그래프 x축 표시 제한
ylim([0 1])                           % 그래프 y축 표시 제한


t2=-0.25:1/fs:0.25;                   % x2(t)의 범위
f2=10;                                % f2=10Hz 설정
L2=length(t2);                        % x2(t) 범위의 길이
x2=cos(2*pi*f2*t2);                   % x2(t) 설정

X2=fft(x2)./L2;                       % x2(t)의 FT 및 Normalization
P2=abs(fftshift(X2));                 % X2(f) 재배열
f2=linspace(-fs/2, fs/2, L2);         % P2(f)의 범위

figure(3)                             % x2(t) 그래프
plot(t2,x2)

figure(4)                             % P2(f) 그래프
plot(f2,P2)
xlim([-300 300])                      % 그래프 x축 표시 제한
ylim([0 1])                           % 그래프 y축 표시 제한
```

```
% Q 1-2-Linearity
fs=1000;                              % Sampling frequency 설정
a=1;                                  % 상수a=1 설정
b=2;                                  % 상수b=2 설정
t=-5:1/fs:5;                          % x1(t), x2(t)의 범위
f1=0.5;                               % f1=0.5Hz 설정
f2=10;                                % f2=10Hz 설정
L=length(t);                          % x1(t), x2(t) 범위의 길이
x1=cos(2*pi*f1*t);                    % x1(t) 설정
x2=cos(2*pi*f2*t);                    % x2(t) 설정
x=a*x1+b*x2;                          % x(t)=ax1(t)+bx2(t) 설정

X1=fft(x1)./L;                        % x1(t)의 FT 및 Normalization
P1=abs(fftshift(X1));                 % X1(f) 재배열
X2=fft(x2)./L;                        % x2(t)의 FT 및 Normalization
P2=abs(fftshift(X2));                 % X2(f) 재배열
X=fft(x)./L;                          % x(t)의 FT 및 Normalization
PX=abs(fftshift(X));                  % X(f) 재배열
f=linspace(-fs/2, fs/2, L);           % P1(f), P2(f), PX(f)의 범위

figure(1)                             % PX(f) 그래프
plot(f,PX)
xlim([-30 30])                        % 그래프 x축 표시 제한
ylim([0 1])                           % 그래프 y축 표시 제한

figure(2)                             % aP1(f)+bP2(f) 그래프
plot(f,a*P1+b*P2)
xlim([-30 30])                        % 그래프 x축 표시 제한
ylim([0 1])                           % 그래프 y축 표시 제한
```

```
% Q 1-2-Scale Change
fs1=1000;                              % Sampling frequency 설정1
fs2=100;                               % Sampling frequency 설정2
a=10;                                  % 상수a=10 설정
f=0.5;                                 % f=0.5Hz 설정
t1=-5:1/fs1:5;                         % x(t)의 범위
t2=-5:1/fs2:5;                         % xa(t)의 범위
L1=length(t1);                         % x(t) 범위의 길이
L2=length(t2);                         % xa(t) 범위의 길이
x=cos(2*pi*f*t1);                      % x(t) 설정
xa=cos(2*pi*a*f*t2);                   % xa(t) 설정

X=fft(x)./L1;                          % x(t)의 FT 및 Normalization
PX=abs(fftshift(X));                   % X(f) 재배열
f1=linspace(-fs1/2, fs1/2, L1);        % PX(f)의 범위
XA=fft(xa)./L1;                        % xa(t)의 FT 및 Normalization
PXA=abs(fftshift(XA));                 % XA(f) 재배열
f2=linspace(-fs2/2, fs2/2, L2);        % PXA(f)의 범위

figure(1)                              % PXA(f) 그래프
plot(f2,PXA)
xlim([-50 50])                         % 그래프 x축 표시 제한
ylim([0 0.1])                          % 그래프 y축 표시 제한

figure(2)                              % PX(f/a)/a 그래프
plot(f1.*a,PX./a)
xlim([-50 50])                         % 그래프 x축 표시 제한
ylim([0 0.1])                          % 그래프 y축 표시 제한
```

```
% Q 2-1
fs=1000;                               % Sampling frequency 설정
t=-5:1/fs:5;                           % x(t)의 범위
L=length(t);                           % x(t) 범위의 길이
x=sinc(t);                             % x(t) 설정

X=fft(x)./fs;                          % x(t)의 FT 및 Normalization
P=abs(fftshift(X));                    % X(f) 재배열
f=linspace(-fs/2, fs/2, L);            % P(f)의 범위

figure(1)                              % x(t) 그래프
plot(t,x)

figure(2)                              % P(f) 그래프
plot(f,P)
xlim([-4 4])                           % 그래프 x축 표시 제한
ylim([0 1.5])                          % 그래프 y축 표시 제한
```

```
% Q 2-2-Modulation
fs=1000;                              % Sampling frequency 설정
f0=10;                                % f0=10Hz 설정
t=-5:1/fs:5;                          % x(t)의 범위
L=length(t);                          % x(t) 범위의 길이
x=sinc(t);                            % x(t) 설정
xc=x.*cos(2*pi*f0*t);                 % xc(t) 설정

X=fft(x)./fs;                         % x(t)의 FT 및 Normalization
P=abs(fftshift(X));                   % X(f) 재배열
XC=fft(xc)./fs;                       % xc(t)의 FT 및 Normalization
PC=abs(fftshift(XC));                 % XC(f) 재배열
f=linspace(-fs/2, fs/2, L);           % P(f), PC(f)의 범위

figure(1)                             % PC(f) 그래프
plot(f,PC)
xlim([-50 50])                        % 그래프 x축 표시 제한
ylim([0 1.5])                         % 그래프 y축 표시 제한

figure(2)                             % P(f-f0)/2+P(f+f0)/2 그래프
plot(f-f0,P./2)                       % P(f-f0)/2
hold on
plot(f+f0,P./2)                       % P(f+f0)/2
xlim([-50 50])                        % 그래프 x축 표시 제한
ylim([0 1.5])                         % 그래프 y축 표시 제한
```

```
% Q 2-2-Scale Change
fs1=1000;                             % Sampling frequency 설정1
fs2=500;                              % Sampling frequency 설정2
a=2;                                  % 상수a=2 설정
t1=-5:1/fs1:5;                        % x(t)의 범위
t2=-5:1/fs2:5;                        % xa(t)의 범위
L1=length(t1);                        % x(t) 범위의 길이
L2=length(t2);                        % xa(t) 범위의 길이
x=sinc(t1);                           % x(t) 설정
xa=sinc(a*t2);                        % xa(t) 설정

X=fft(x)./fs1;                        % x(t)의 FT 및 Normalization
PX=abs(fftshift(X));                  % X(f) 재배열
f1=linspace(-fs1/2, fs1/2, L1);       % PX(f)의 범위
XA=fft(xa)./fs2;                      % xa(t)의 FT 및 Normalization
PXA=abs(fftshift(XA));                % XA(f) 재배열
f2=linspace(-fs2/2, fs2/2, L2);       % PXA(f)의 범위

figure(1)                             % PXA(f) 그래프
plot(f2,PXA)
xlim([-10 10])                        % 그래프 x축 표시 제한
ylim([0 1])                           % 그래프 y축 표시 제한

figure(2)                             % PX(f/a)/a 그래프
plot(f1.*a,PX./a)
xlim([-10 10])                        % 그래프 x축 표시 제한
ylim([0 1])                           % 그래프 y축 표시 제한
```

```
% Q 3-1-1
syms t
u=heaviside(t);         % u(t) 설정
U=laplace(u);           % U(f) 설정

figure(1)               % u(t) 그래프
fplot(u)

figure(2)               % U(f) 그래프
fplot(U)
```

```
% Q 3-1-2
syms t
y=(exp(t)+exp(-5*t)-2*cos(2*t))*heaviside(t);     % y(t) 설정
Y=laplace(y);                                     % Y(f) 설정

figure(1)                                         % y(t) 그래프
fplot(y)

figure(2)                                         % Y(f) 그래프
fplot(Y)
```

```
% Q 3-2-1
syms t
vc=(1-exp(-t))*heaviside(t);    % vc(t) 설정

figure(1)                       % vc(t) 그래프
fplot(vc)
```

```
% Q 3-2-2
syms t
h=exp(-t)*heaviside(t);         % h(t) 설정
H=tf([1],[1 1]);                % H(s) 설정

figure(1)                       % h(t) 그래프
fplot(h)

figure(2)                       % H(s) bode plot
bode(H)
```

```
% Q 3-2-3
H=tf([1],[1 1]);        % H(s)=1/(s+1) 설정
pzmap(H)                % H(s) pole-zero plot
grid on
```

```
% Q 4-1
t=0:0.001:1;            % x(t)의 범위
x=myfunc(t);            % x(t) 설정

figure(1)               % x(t) 그래프
plot(t,x)
```

```
% Q 4-2
fs=1000;                      % Sampling frequency 설정
t=0:1/fs:1;                   % x(t)의 범위
L=length(t);                  % x(t) 범위의 길이
x=myfunc(t);                  % x(t) 설정

X=fft(x)./L;                  % x(t)의 FT 및 Normalization
P=abs(fftshift(X));           % X(f) 재배열
f=linspace(-fs/2,fs/2,L);     % P(f)의 범위

figure(1)                     % P(f) 그래프
plot(f,P)
```

```
% Q 4-3-a
fs=1000;                      % Sampling frequency 설정
t=0:1/fs:1;                   % x(t)의 범위
L=length(t);                  % x(t) 범위의 길이
x=myfunc(t);                  % x(t) 설정

X=fft(x)./L;                  % x(t)의 FT 및 Normalization
Xi=ifft(X).*L;                % X(t)의 IFT

figure(1)                     % x(t) 그래프
plot(t,x)
ylim([-10 15])                % 그래프 y축 표시 제한

figure(2)                     % Xi(t) 그래프
plot(t,Xi)
ylim([-10 15])                % 그래프 y축 표시 제한
```

```
% Q 4-3-b
fs1=1000;                         % Sampling frequency 설정1
fs2=150;                          % Sampling frequency 설정2
t1=0:1/fs1:1;                     % x1(t)의 범위
t2=0:1/fs2:1;                     % x2(t)의 범위
L1=length(t1);                    % x1(t) 범위의 길이
L2=length(t2);                    % x2(t) 범위의 길이
x1=myfunc(t1);                    % x1(t) 설정
x2=myfunc(t2);                    % x2(t) 설정

X1=fft(x1)./L1;                   % x1(t)의 FT 및 Normalization
X2=fft(x2)./L2;                   % x2(t)의 FT 및 Normalization
X1i=ifft(X1).*L1;                 % X1(f)의 IFT
X2i=ifft(X2).*L2;                 % X2(f)의 IFT
P1=abs(fftshift(X1));             % X1(f) 재배열
P2=abs(fftshift(X2));             % X2(f) 재배열
f1=linspace(-fs1/2,fs1/2,L1);     % P1(f)의 범위
f2=linspace(-fs2/2,fs2/2,L2);     % P2(f)의 범위

figure(1)                         % P1(t) 그래프
plot(f1,P1)

figure(2)                         % P2(t) 그래프
plot(f2,P2)

figure(3)                         % X1i(t) 그래프
plot(t1,X1i)

figure(4)                         % X2i(t) 그래프
plot(t2,X2i)
```