

Signals and Systems

EEE2060.03-00

Project 2 Report

- Sampling -

Name : 함태완

Project 2

Signals and Systems (EEE2060.03-00), Yonsei university

함 태완¹

wani7@yonsei.ac.kr

Problem 1

(1) Problem 1-(a)

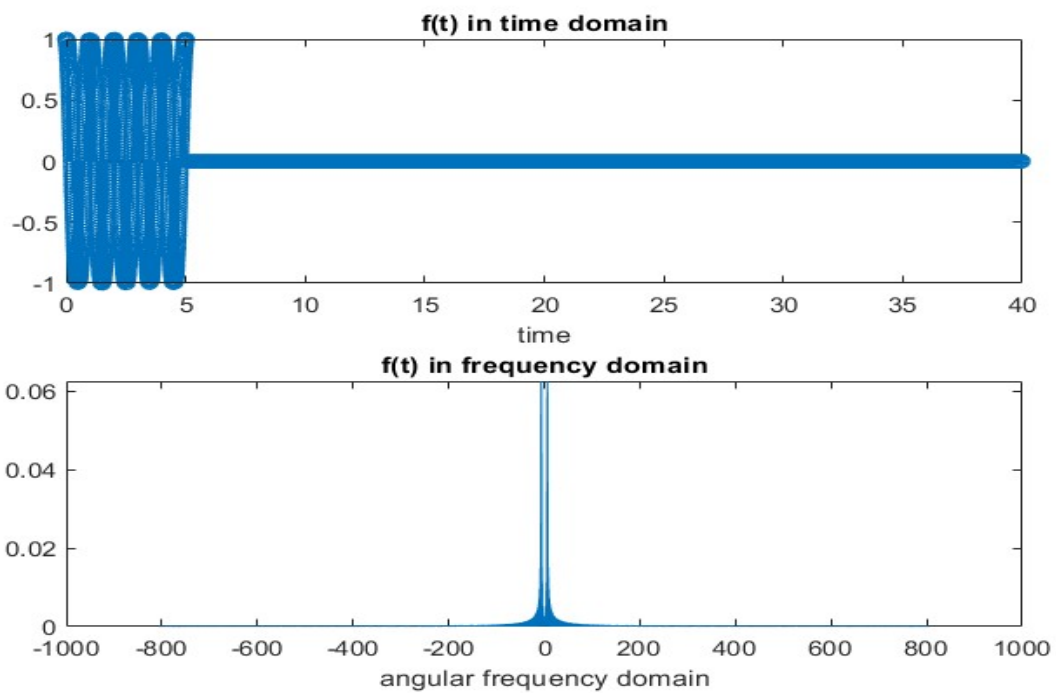


Figure 1 $f(t)$ in time and frequency domain

T가 5인 $f(t)$ 는 Figure 1과 같이 그려지는 것을 확인할 수 있다. Discrete 함수이기 때문에 plot이 아닌 stem을 이용해주었다.

(2) Problem 1-(b)

¹ 전기전자공학부 학부 재학생

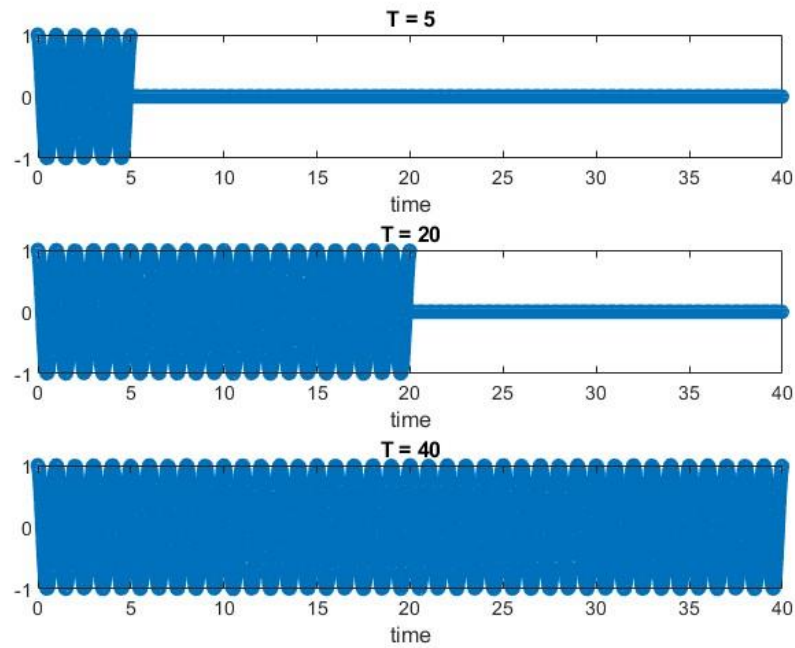


Figure 2 $f(t)$ in time domain when $T=5,20,40$

T 가 커질수록 time domain에서 \cos 부분의 비율이 커지는 것을 확인할 수 있다. 결과는 Figure 2와 같다. frequency domain에서의 plot은 Figure 3과 같다. T 가 작을수록 sinc 함수의 개형이 나타나는 것을 확인할 수 있다. T 가 커질수록 이상적인 \cos 함수의 푸리에 변환과 같게 된다. 이상적인 \cos 함수의 푸리에 변환은 Figure 4와 같다.

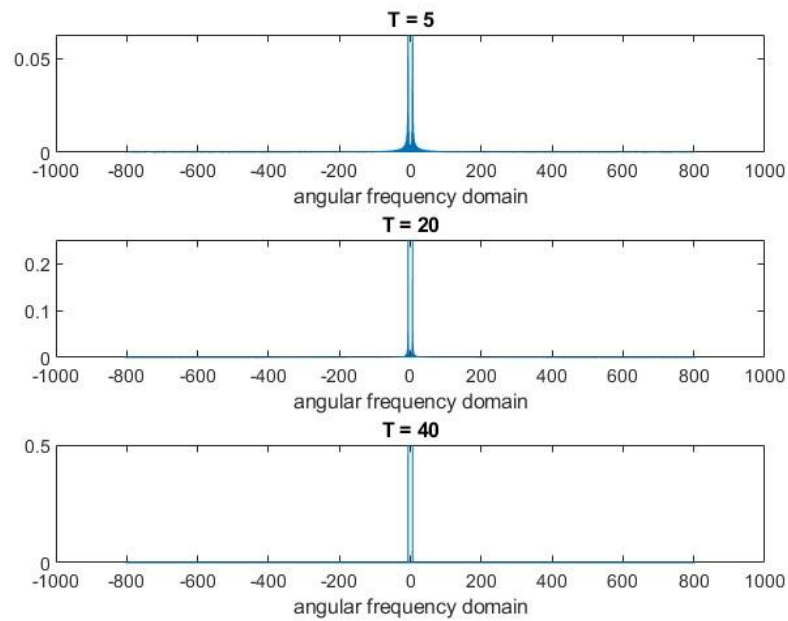


Figure 3 $f(t)$ in frequency domain when $T=5,20,40$

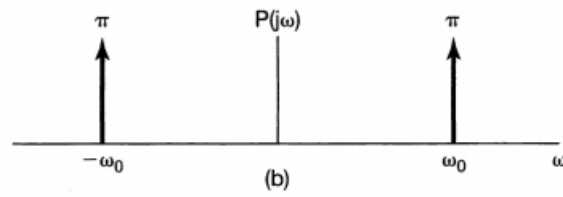


Figure 4 Ideal plot of $\cos w_0 t$ in frequency domain²

T가 작을수록 sinc 함수의 개형이 나타나는 이유는 time-limited 되었다는 것은 time domain에서 $f(t)$ 에 rectangular 함수를 곱한 것과 같기 때문이다. time domain에서의 rectangular 함수는 푸리에 변환을 거치면 frequency domain에서는 sinc 함수가 된다. Multication property에 의해서, time domain에서의 곱은 frequency domain에서의 convolution과 같게 된다. 따라서, time domain에서 T가 작을수록, 더 time-limited 될수록 frequency domain에서는 이상적인 cos 함수의 푸리에 변환이 아닌 sinc 함수가 같이 나타나는 것이다.

Problem 2

Nyquist theorem에 의하면, aliasing이 일어나지 않기 위한 w_s 의 범위는 아래와 같다.

$$w_s > 2w_1 \cdots (1)$$

주어진 조건에 따르면, w_1 은 200π 이므로 w_s 는 400π 보다 크면 된다. Figure 5는 w_s 가 600π 로 가정하고 plot한 것이다. Figure 6는 w_s 가 200π 로 가정하고 plot한 것이다. Figure 5에서의 w_s 는 식 (1)을 만족하는 반면, Figure 6에서의 w_s 는 식(1)을 만족하지 못한다.

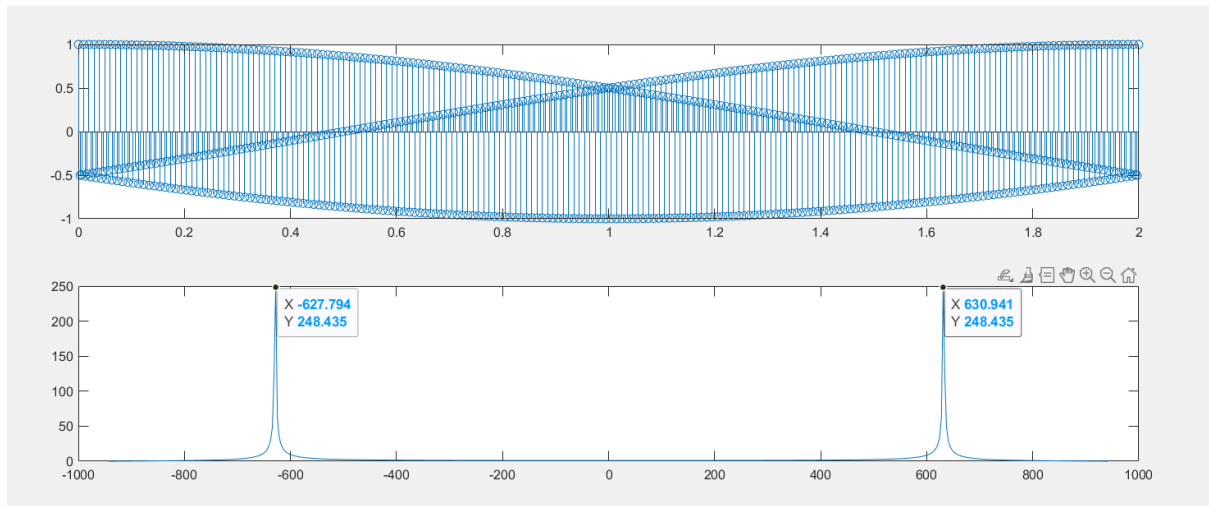


Figure 5 sampled $m(t)$ in time and frequency domain when $w_s=600\pi$

² Oppenheim, Signals and Systems (Pearson, 1996), 323.

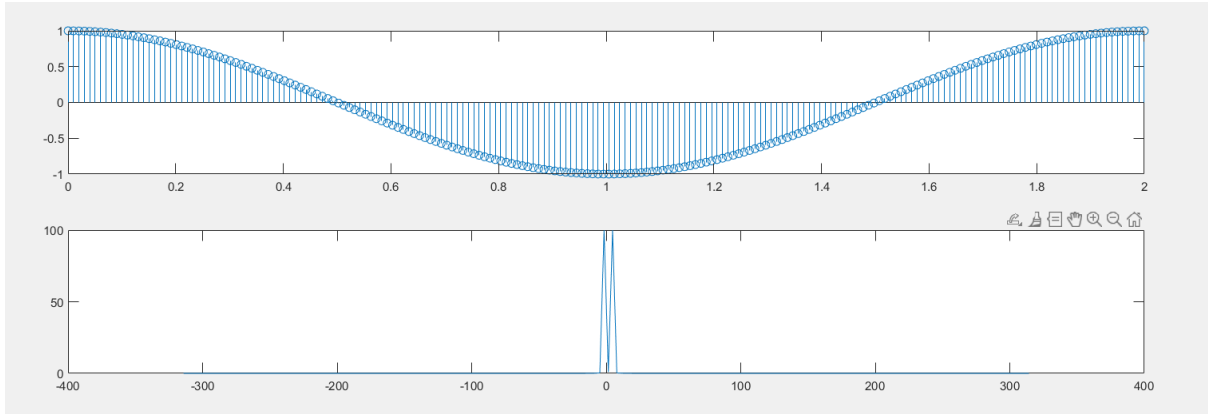


Figure 6 Sampled $m(t)$ in time and frequency domain when $w_s=200\pi$

Figure 6에서 볼 수 있듯이 w_s 가 식 (1)을 만족하지 못했기 때문에 aliasing이 일어나고 있다.

Problem 3

(1) Problem 3-(a)

문제에서 주어진 $x(t)$, $c(t)$, $r(t)$ 를 각각 time domain, frequency domain에서 plot하면 Figure 7, 8과 같다. $r(t)$ 는 $x(t)$ 와 $c(t)$ 의 곱이다. 따라서, 푸리에 변환의 multiplication property에 의해서 frequency domain에서의 $r(t)$ 는 $x(t)$ 의 푸리에 변환과 $c(t)$ 의 푸리에 변환의 convolution과 같다.

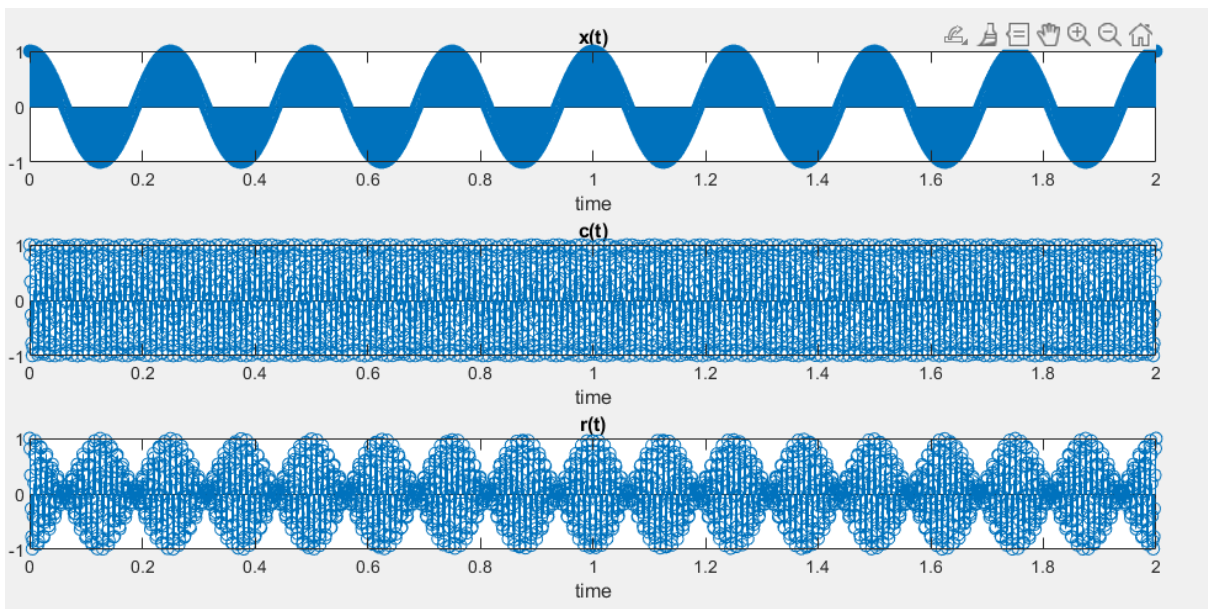


Figure 7 Plot in time domain

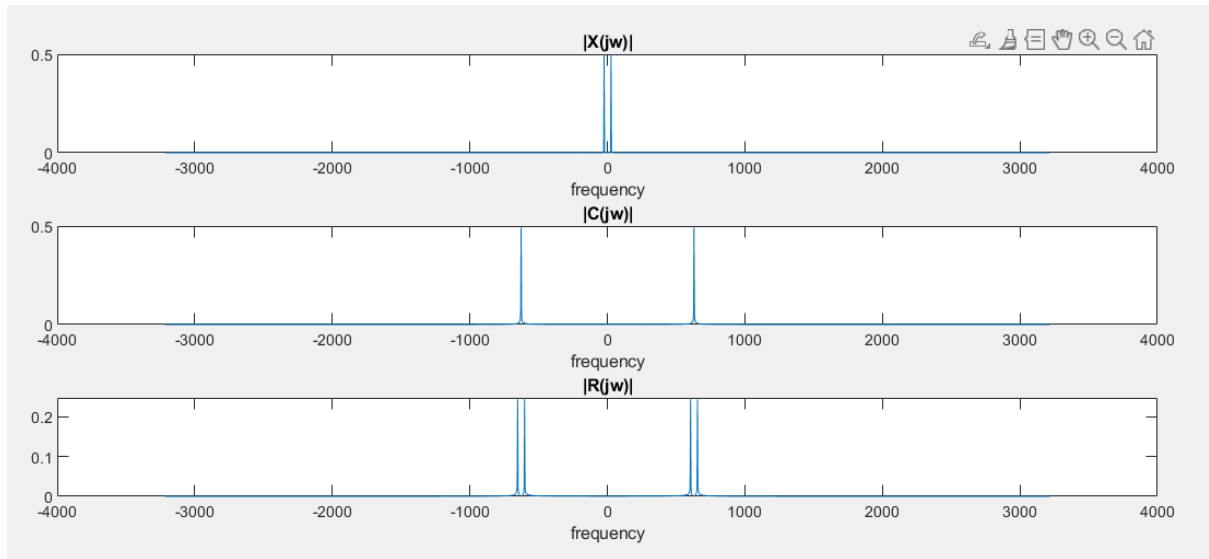


Figure 8 Plot in frequency domain

Figure 8에서 $|R(jw)|$ 의 gain이 0.248...으로 $|R(jw)|$ 와 $|C(jw)|$ 의 gain인 0.5보다 절반인 것을 알 수 있다. 그 이유는 modulation property 때문에 그렇다. modulation property에 의하면 아래와 같은 관계가 성립된다.

$$x(t) \xleftrightarrow{FT} X(w)$$

$$x(t)\cos\omega_0 t \xleftrightarrow{FT} \frac{1}{2}(X(\omega - \omega_0) + X(\omega + \omega_0))$$

따라서 gain이 modulation 과정에서 절반이 된 것이다.

(2) Problem 3-(b)

주어진 $d(t)$ 는 (a)에서 만든 $r(t)$ 에 $c(t)$ 를 곱한 신호이다. 이를 frequency domain에서 그리면 Figure 9과 같다.

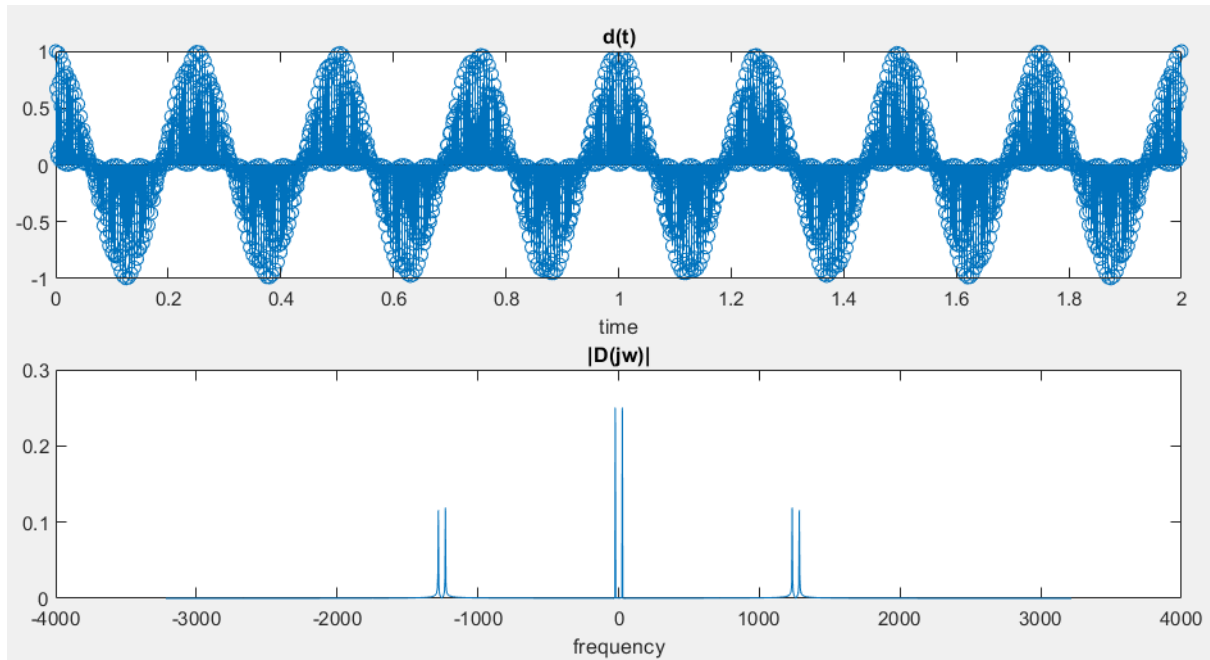


Figure 9 Signal d(t) in time and frequency domain

(3) Problem 3-(c)

d(t)가 x(t)가 다른 점은 Figure 8과 Figure 9을 비교하면 직관적으로 느낄 수 있다. d(t)의 경우 high frequency에도 피크가 있으며 x(t)와 유사한 위치에 있는 피크의 gain이 x(t)보다 절반이다. 그 이유는 d(t)를 만드는 과정에 있다. d(t)는 r(t)에 cos 함수를 곱한 것이므로 결과적으로는 x(t)의 관점에서 cos 함수를 두 번 곱한 것이 되는 것이다. 이에 modulation property가 두 번 적용된 것이다.

d(t)를 x(t)로 복원하기 위해서는 두 번의 modulation 과정에서 발생된 high frequency에 있는 피크를 제거하기 위한 low-pass filter(LPF)를 적용하면 된다. 이 때, gain 또한 복원해야 하므로 아래와 같이 LPF를 frequency domain에서 설계하였다.

$$H(jw) = \begin{cases} 2 & -51 < \omega < 55 \\ 0 & \text{else} \end{cases}$$

설계한 LPF를 이용하여 신호를 복원한 결과는 Figure 10과 같다. 의도했던 대로, high frequency가 제거되고 gain이 복구되었다.

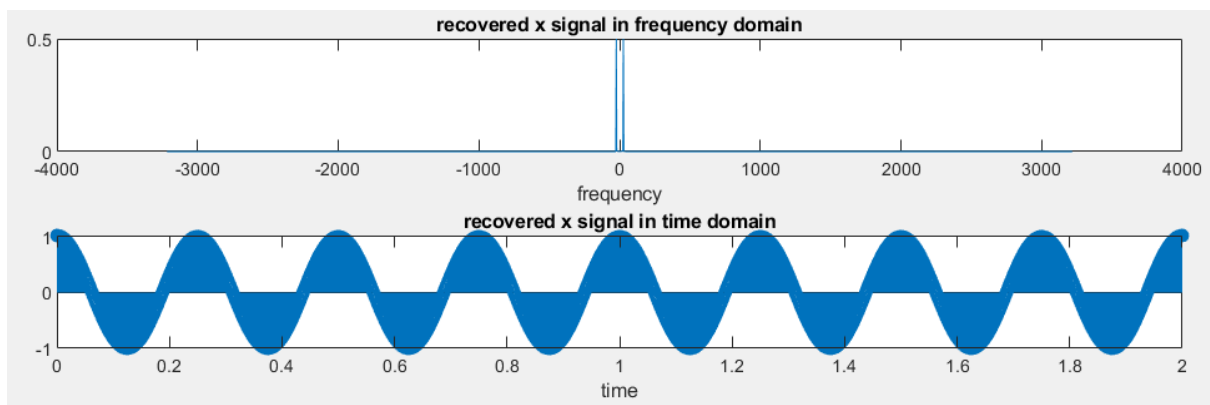


Figure 10 LPF and recovered x signal

Problem 4

(1) Problem 4-(a)

주어진 신호를 time domain과 frequency domain에서 plot 하면 Figure 11과 같다. 해당 신호는 ω 가 200~300, 400~600일때만 다른 주파수를 갖는 cos 함수가 존재한다. 이는 time-limited된 함수가 두 개 존재하는 것으로 생각할 수 있으며 Problem 1의 사례를 참고 할 수 있다. 예상대로 frequency domain에서 sinc 함수의 형태가 나타난다.

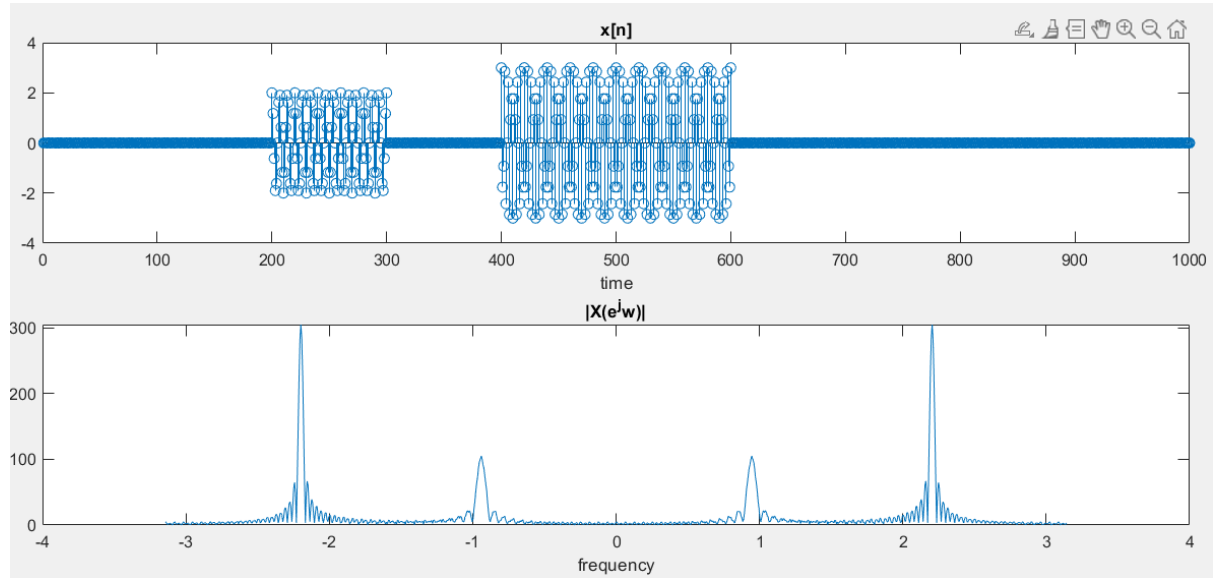


Figure 11 $x[n]$ in time and frequency domain

(2) Problem 4-(b)

구하려는 신호 $y[n]$ 은 $x[n]$ 과 $h_1[n]$ 의 convolution이다. FT의 convolution property에 의해서 time domain에서의 convolution은 frequency domain에서의 multiplication과 같다. $y[n]$ 을 구하기 위해서 frequency domain에서 곱한 다음, ifft를 이용하여 time domain에서의 $y[n]$ 을 구했다. Figure 12에서 결과를 확인할 수 있다. Figure 12의 첫번째 plot은 $h_1[n]$ 을 time domain에서 plot한 것이다. 이러지는 두 plot은 $y[n]$ 을 time domain과 frequency domain에서 각각 plot 한 것이다.

$H_1(j\omega)$ 를 time domain에서 살펴보면 아래와 같다.

$$H_1(e^{j\omega}) = 0.5e^{-j400\omega} \xleftrightarrow{FT} h_1[n] = 0.5\delta(t - 400)$$

따라서, 이론적으로 $y_1[n]$ 은 $x[n]$ 을 400만큼 time shift하고 gain을 절반으로 줄인 형태가 된다. 하지만 Figure 12를 보면, 400만큼의 time shift는 예상대로 되었지만, gain은 거의 1/4만큼 줄은 것을 확인할 수 있다. 이는 $Y(e^{j\omega})$ 를 ifft 하는 과정에서 발생한 오류라고 예상된다.

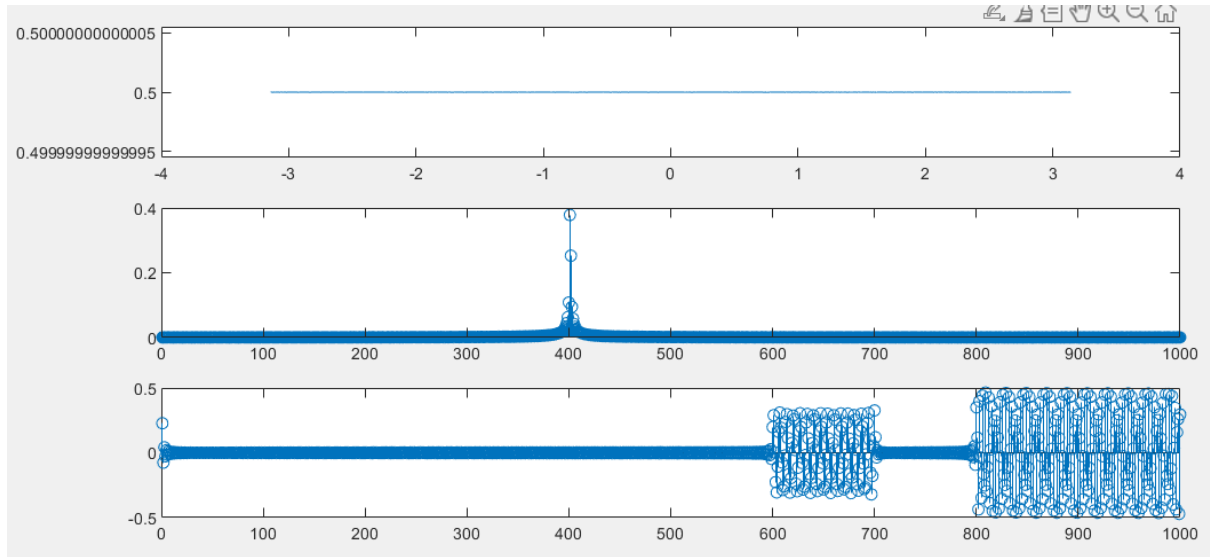


Figure 12 $h_1[n]$ 과 $y[n]$ 의 plot

(3) Problem 4-(c)

$H_2(e^{jw})$ 는 $-0.5\pi \sim 0.5\pi$ 에서는 e^{-j400w} 이고, 이외에서는 $0.5e^{-j400w}$ 이다. 이를 (b)와 같은 방법으로 $y[n]$ 을 구한 후 plot 하면 Figure 13과 같은 결과를 확인할 수 있다. $x[n]$ 에서 n 이 200~300인 \cos 함수의 frequency는 대략 -1 과 1 부근에서 형성되어 있다. 따라서 $H_2(e^{jw})$ 의 e^{-j400w} 의 영향을 받으리라고 예상할 수 있다. $x[n]$ 의 $2\cos(0.3\pi n)$ 은 gain을 유지한 채 400 만큼 time shift 될 것이다. $x[n]$ 에서 n 이 400~600인 \cos 함수는 $H_2(e^{jw})$ 의 $0.5e^{j400w}$ 의 영향을 받으리라고 예상할 수 있다. $x[n]$ 의 $3\cos(0.7\pi n)$ 은 gain이 절반인 채 -400 만큼 time shift를 할 것이다. 위의 추측들이 Figure 13에서 사실임을 확인할 수 있다.

다만, Figure 12에서 $n=800$ 부근에서 예상을 벗어난 peak가 있는 것을 확인할 수 있다. 이는 $X(e^{jw})$ 의 다른 두 \cos 함수의 sinc 함수 때문에 발생한 것이라고 추측된다. sinc 함수는 limited 되어 있지 않고 이론상 무한하게 값이 존재하므로 $H_2(e^{jw})$ 를 적용하는 과정에서 $n=800$ 부근의 peak가 발생한 것이다.

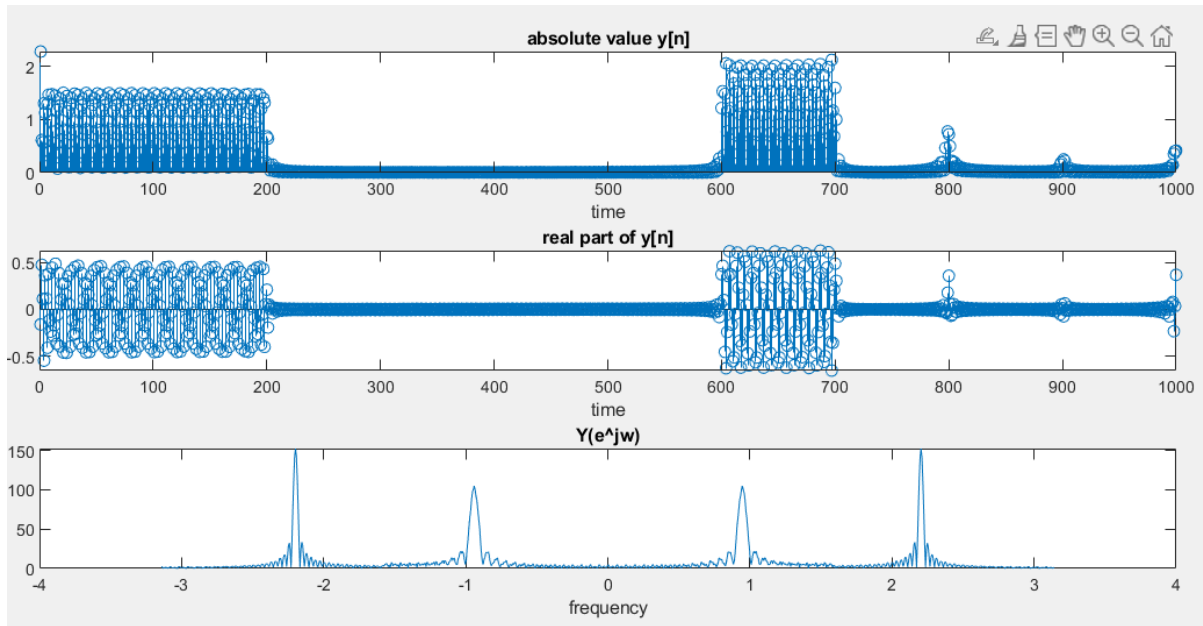


Figure 13 $y[n]$ 의 plot

Source Code

* 해당 프로젝트에서는 MATLAB R2022a가 사용되었다.

(1) Problem 1

```
% Project 2
% - Problem 1
% - Author : Taewan Ham

[t_1, t_2, samples] = make_domain(5);
t = [t_1, t_2]; % time domain
w = linspace(-pi*256, pi*256, samples); % angular frequency domain

x_t = [cos(2*pi*t_1), 0*t_2]; % sampled function in time domain
x_f = fftshift(fft(x_t)); % sampled function in frequency domain

% Problem 1-(a)
figure(1);
subplot(2,1,1); stem(t, x_t); xlabel('time'); title('f(t) in time domain');
subplot(2,1,2); plot(w, abs(x_f)/samples); xlabel('angular frequency domain'); title('f(t) in frequency domain');

% Problem 1-(b)
[t_1_20, t_2_20, samples_20] = make_domain(20); % T = 20
[t_1_40, t_2_40, samples_40] = make_domain(40); % T = 40

t_20 = [t_1_20, t_2_20]; % time domain with T = 20
t_40 = [t_1_40, t_2_40]; % time domain with T = 40
w_20 = linspace(-pi*256, pi*256, samples_20); % angular frequency domain with T = 20
w_40 = linspace(-pi*256, pi*256, samples_40); % angular frequency domain with T = 40
```

```

x_t_20 = [cos(2*pi*t_1_20), 0*t_2_20]; % sampled function in time domain
x_f_20 = fftshift(fft(x_t_20)); % sampled function in frequency domain
x_t_40 = [cos(2*pi*t_1_40), 0*t_2_40]; % sampled function in time domain
x_f_40 = fftshift(fft(x_t_40)); % sampled function in frequency domain

figure(2);
subplot(3,1,1); stem(t, x_t); xlabel('time'); title('T = 5');
subplot(3,1,2); stem(t, x_t_20); xlabel('time'); title('T = 20');
subplot(3,1,3); stem(t, x_t_40); xlabel('time'); title('T = 40');

figure(3);
subplot(3,1,1); plot(w, abs(x_f)/samples); xlabel('angular frequency domain'); title('T = 5');
subplot(3,1,2); plot(w, abs(x_f_20)/samples_20); xlabel('angular frequency domain'); title('T = 20');
subplot(3,1,3); plot(w, abs(x_f_40)/samples_40); xlabel('angular frequency domain'); title('T = 40');

function [t_1, t_2, samples] = make_domain(T)
    r = T / (40); % ratio
    samples = 256 * 40; % 256 (samples per second) * 40 seconds

    t_1 = linspace(0, T, samples*r);
    t_2 = linspace(T, 40, samples*(1-r));
end

```

(2) Problem 2

```

% Project 2
% - Problem 2
% - Author : Taewan Ham

% sampling frequency = w_s
% w_s > 2*w1 = 2*2*pi*100
% Let sampling frequency w_s = 6*pi*100 (300Hz)

w_s = 2*pi*100;
samples = 2*w_s / (2*pi); % number of samples
n = linspace(0, 2, samples); % sampled time domain
w = linspace(-w_s*0.5, w_s*0.5, samples); % frequency domain

m_t = cos(2*pi*100*n);
m_f = fftshift(fft(m_t)); % ft

figure(1);
subplot(2,1,1); stem(n, m_t); % signal in time domain
subplot(2,1,2); plot(w, abs(m_f)); % signal in frequency domain

```

(3) Problem 3

```

% Project 2
% - Problem 3

```

```

% - Author : Taewan Ham

w_s = 2*pi*1024;
Hz = w_s / (2*pi);
samples = 2*Hz; % number of samples
n = linspace(0, 2, samples); % sampled time domain
w = linspace(-w_s*0.5, w_s*0.5, samples);

x_t = cos(2*pi*4*n); % signal x
c_t = cos(2*pi*100*n); % signal c
r_t = x_t.*c_t; % signal r

x_f = fftshift(fft(x_t));
c_f = fftshift(fft(c_t));
r_f = fftshift(fft(r_t));

% Problem 3-(a)
figure(1);
subplot(3,1,1); stem(n, x_t); xlabel('time'); title('x(t)'); % signals in time domain
subplot(3,1,2); stem(n, c_t); xlabel('time'); title('c(t)');
subplot(3,1,3); stem(n, r_t); xlabel('time'); title('r(t)');

figure(2);
subplot(3,1,1); plot(w, abs(x_f)/samples); xlabel('frequency'); title('|X(jw)|'); % signals in
frequency domain
subplot(3,1,2); plot(w, abs(c_f)/samples); xlabel('frequency'); title('|C(jw)|');
subplot(3,1,3); plot(w, abs(r_f)/samples); xlabel('frequency'); title('|R(jw)|');

% Problem 3-(b)
d_t = r_t.*c_t; % signal d
d_f = fftshift(fft(d_t));

figure(3);
subplot(2,1,1); stem(n, d_t); xlabel('time'); title('d(t)'); % time domain
subplot(2,1,2); plot(w, abs(d_f)/samples); xlabel('frequency'); title('|D(jw)|'); % frequency
domain

% Problem 3-(c)
fil_h = [zeros(1,1007) 2*ones(1,35) zeros(1,1006)]; % Lowpass filter w = -51~55
recov_x_f = d_f.*fil_h; % recovered signal
recov_x_t = ifft(ifftshift(recov_x_f));

figure(4);
subplot(3,1,1); plot(w, abs(fil_h)/samples); xlabel('frequency'); title('Lowpass filter');
subplot(3,1,2); plot(w, abs(recov_x_f)/samples); xlabel('frequency'); title('recovered x signal in
frequency domain');
subplot(3,1,3); stem(n, recov_x_t); xlabel('time'); title('recovered x signal in time domain');

```

(4) Problem 4

```

% Project 2
% - Problem 4
% - Author : Taewan Ham

n = 1 : 1000; % 1000 points

```

```

w = linspace(-pi, pi, 1000);

x_n = [0*[1:199], 2*cos(0.3*pi*[200:300]), 0*[301:399], 3*cos(0.7*pi*[400:600]), 0*[601:1000]];
x_f = fftshift(fft(x_n));

% Problem 4-(a)
figure(1);
subplot(2,1,1); stem(n, x_n); xlabel('time'); title('x[n]');
subplot(2,1,2); plot(w, abs(x_f)); xlabel('frequency'); title('|X(e^jw)|');

% Problem 4-(b)
h1_f = 0.5*(exp(-j*400*w));
y_f = x_f.*h1_f;
h1_t = ifft(ifftshift(h1_f));
y_t = ifft(ifftshift(y_f));
figure(2);
subplot(3,1,1); plot(w, abs(h1_f)); % h1 的 frequency domain plot
subplot(3,1,2); stem(n, abs(h1_t)); % h1 的 time domain plot
subplot(3,1,3); stem(n, real(y_t)); % signal y in time domain

% Problem 4-(c)
h2_f = [0.5*(exp(j*400*w(1:249))), exp(-j*400*w(250:751)), 0.5*exp(j*400*w(752:1000))]; % h2 in
frequency domain
y2_f = x_f.*h2_f; % y2 in frequency domain
h2_t = ifft(ifftshift(h2_f));
y2_t = ifft(ifftshift(y2_f));

figure(3);
subplot(3,1,1); stem(n, abs(y2_t)); xlabel('time'); title('absolute value y[n]'); % gain
subplot(3,1,2); stem(n, real(y2_t)); xlabel('time'); title('real part of y[n]'); % real
subplot(3,1,3); plot(w, abs(y2_f)); xlabel('frequency'); title('Y(e^jw)');

```