임베디드
신호처리 실습

Sampling 실습 결과보고서

전자공학부 임베디드시스템

2014146004 김 민 섭

1.1 임펄스열 함수 구현

임펄스열은 기본적으로 다음과 같이 정의된다.

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nTs)$$

위 함수를 메틀랩에서 구현시키는 것이 이번 실습의 주요 목표과제 중 하나이다. 메트랩으로 구현하기 위해서 다음과 같은 입력과 출력을 설정하였다.

- 입력

시작시간 : t1[sec]
끝 시간의 최댓값 : t2[sec]
임펄스열의 전체길이 : N
표본화 주파수 : fs[Hz]

- 출력

시간 : t[sec]
임펄스열 신호 : p

본격적으로 임펄스열 함수를 구현하기 이전에 임펄스열이 무엇인지를 한번 살펴보고자 한다. 임펄스열은 말 그대로 임펄스 신호가 열지어 있는 것이다. 임펄스 함수는 시간이동이 없을때 '0'의 값에서만 '1'의 값을 가지는 함수이다. 이것이 열지어 있다는 의미이다. 얼마나 자주 열지어 있는가를 말하는 것이 주기가 될 것이며 1/주기 는 주파수가 될 것이다. 이제 임펄스열이 어떤 것인지에 대한 이해가 이루어졌으니 어떻게 함수를 만들어 나갈 것인지에 대해서 말해보도록 하겠다.

우리는 임펄스열 함수를 작성할때 해결해야 할 것을 크게 3가지로 바라보았다.

- **1이 찍혀야 할 시간은 정확히 언제인가?**
- **1과 1사이에 0은 어떻게 찍을 것인가?**
- **전체 시간에서 1이 나올 위치는 어디인가?**

위 세가지 질문에 답을 구해 나아가면서 구현 함수를 작성하였다. 이에 대해서 순서대로 살펴보고자 한다.

- **1이 찍혀야 할 정확한 시간 값 찾기.**

1이 찍혀야 할 정확한 시간값을 찾기 위해서 시작시간(t1)과 끝시간의 최댓값(t2)를 활용하였다. 그리고 주기 주기를 구하여 주기마다 1이 찍혀야 할 정확한 시간 값을 설정하였다. 구현 코드는 다음과 같다.

```
Ts = 1/fs;
value_time = [t1 : Ts : t2];
```

의외로 간단하게 첫번째 질문에 대한 해답은 다음 코드를 통해서 구할 수 있었다. 하지만 이를 구하는 것은 다음의 질문들을 해결할 때 있어서 굉장히 중요하게 작용을 하게되었다.

- **1과 1사이에 ‘0’의 값을 채워 넣기.**

1과 1사이의 값을 구하였으니 이제 그 사이에 들어가야 할 ‘0’의 개수를 맞추고 집어 넣어주어야 한다. 우선 전체 길이 N을 기준으로 1과 1사이에 몇개의 ‘0’ 값들이 들어가야 하는지 계산하였다. 이는 N 을 ‘1’이 나오는 갯수로 나누면 쉽게 구할 수 있다. 1과 1사에에 몇개의 ‘0’이 나오는지 알아냈다면 그 사이에 집어 넣는 과정을 진행해야 한다. 이 과정 이후부터는 우리가 작성한 코드를 보고 설명하도록 하겠다.

```
between_time = N/(length(value_time)) //1과 1사이에 0 이 들어가는 갯수
between_time = ceil(between_time) // 갯수이므로 정수가 나와야 함(올림)
D=[];              //빈 행렬 생성

for k=2:length(value_time)
    mid = linspace(value_time(k-1),value_time(k),between_time+1); //(1) 하단 보충 설명
    D=[D , mid];
    D = unique(D); //(2) 하단 보충 설명
end

total_time = D(:,1:length(D)-1); //t2는 포함시키지 않으려 -1까지 잡았다.
```

(1)은 예시를 들어서 설명하는것이 더 이해도가 높을 것같아 예시를 들어서 설명해 보도록 하겠다. 예를 들어 value_time(k-1) 의 값이 ‘1’ 이고 value_time(k)의 값이 2이며 between_time+1 = 6 라고 가정해 보겠다. 이렇게 된다면 mid = [1, 1.2, 1.4, 1.6, 1.8, 2.0] 이란 행렬이 생성될 것이다. between_time 에 1을 더해준 목적은 1부터 2.0까지 2를 포함하지 않고 5개의 사이값을 넣기 위함이다. 그런데 계속해서 이와 같은 방식으로 만든다면 겹치는 구간이 생기게 된다. 이를 해결하기 위해서 (2)을 실행해 주었다. 이는 유일한 값만 남기게 하는 함수이다. 이 과정을 거치게 되면서 우리는 전체시간을 구할 수 있었다.

- **전체시간에서 1이 나올 위치를 찾아서 1의 값 대입해 주기.**

지금까지 1이 나와야할 시간을 구하고 그 사이에 추가적으로 시간을 채워줌으로서 전체시간을 구하였다. 이제 간단한 작업만 남았다. 1이 나와야 할 시간의 순서에 1의 값을 대입해 주면 된다. 우리가 작성한 코드를 먼저 확인한 후에 계속 설명하도록 하겠다.

```
value = [zeros(1, length(total_time))];  // 전체 시간 길이만큼의 0행렬을 만들어 주었다.

for i = 1:length(total_time)
    for j=1:length(value_time)
        if total_time(i) == value_time(j)
            value(i) =1;
        end
    end
end
```

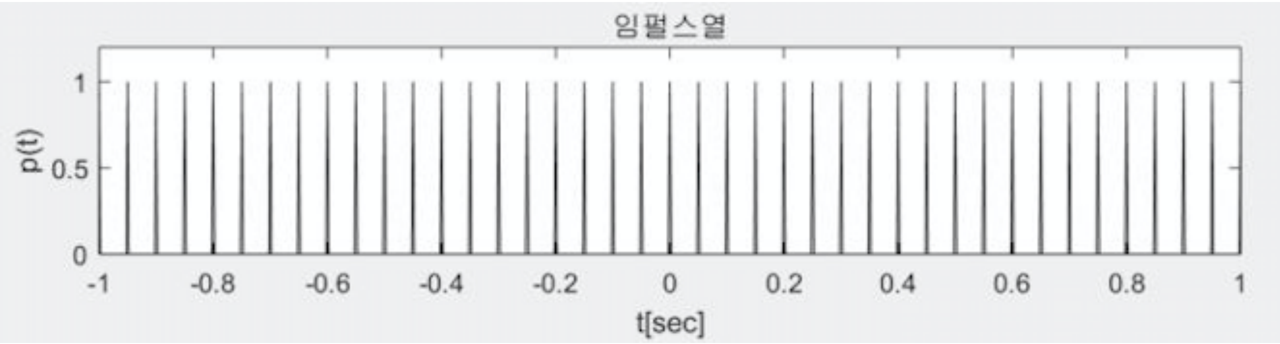
위의 코드를 보면 알 수 있듯이 1이 나와야 할 시간과 전체 시간이 일치할때에 value 값에다가 '1'을 대입해 줌으로서 코드를 완성시킬 수 있었다. 전체 코드는 다음과 같다.

```
function [t , p] = myfun_impulse (t1, t2,N, fs)
Ts = 1/fs;
value_time = [t1 : Ts : t2];

between_time = N / (length(value_time))
between_time = ceil(between_time)
D=[];

for k=2:length(value_time)
   mid = linspace(value_time(k-1), value_time(k), between_time+1);
   D=[D , mid];
   D = unique(D);
end

total_time = D( : ,1:length(D)-1);
value = [ zeros(1, length( total_time ) ) ];
for i = 1:length( total_time )
   for j=1:length( value_time )
      if total_time(i) == value_time(j)
         value(i) =1;
      end
   end
end

t = total_time(:,1:N);
p = value(:,1:N);
```

1.2 구현한 임펄스열을 다음 파라미터에 대해 구하고 그래프에 표시하여라

– t_1 = –5, t_2 = 5
– f_s = 20 Hz
– N = 4096

위 파라미터에 대해서 메틀랩을 활용하여서 임펄스열을 구현해 보았다. 사용함수는 1.1에서 만든 임펄스열 함수를 사용하였다.



1.3 임펄스열 p(t)의 Fourier Transform을 손으로 계산하라.

$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

$$T = T_s \ \left(\tfrac{1}{20}\right) \quad f = \frac{1}{T_s} \ (20) \quad w_0 = 2\pi f = \underline{40\pi}$$

$$C_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)\, e^{-jk2\pi f t}\, dt$$

$p(t)$ 를 집어넣어 보면

$$C_n = \frac{1}{T_s}\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} p(t)\, e^{-jn2\pi f t}\, dt \Rightarrow$$ n번째 주파수 스펙트럼의 크기!

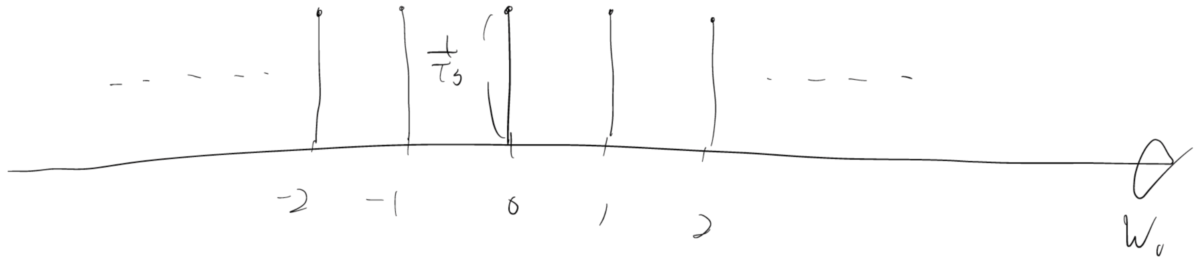
$$C_0 = \frac{1}{T_s}\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} \delta(t)\, dt = \frac{1}{T_s}$$

$$C_1 = \frac{1}{T_s}\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} \delta(t - T_s)\, e^{-j2\pi f t}\, dt \quad \frac{1}{T_s} e^{-j2\pi f} = \frac{1}{T_s} e^{-jw_0}$$

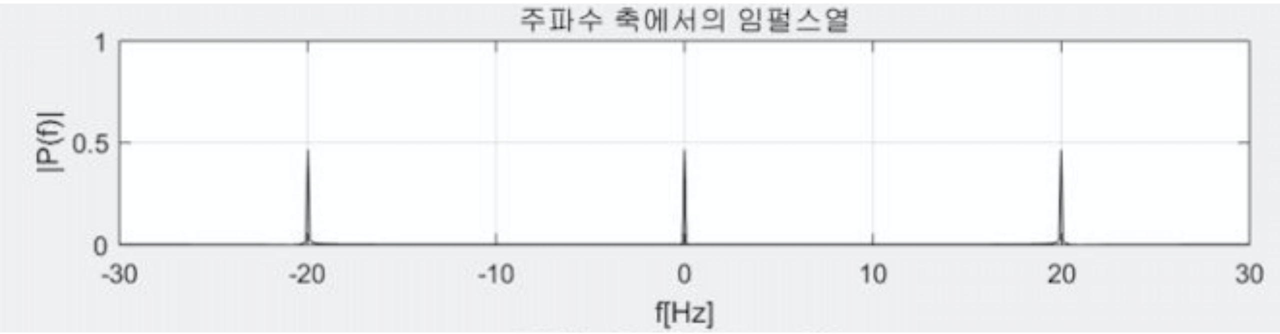
$$C_2 = \frac{1}{T_s}\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} \delta(t - 2T_s)\, e^{-2j2\pi f t}\, dt = \frac{1}{T_s} e^{-2j2\pi f} = \frac{1}{T_s} e^{-2jw_0}$$

$$C_n = \frac{1}{T_s} e^{-njw_0}$$

1.3 임펄스열 p(t)의 크기 스펙트럼을 (MATLAB으로) 구하여 그래프에 표시하고 손으로 계산한 Fourier Transform과 비교하라.



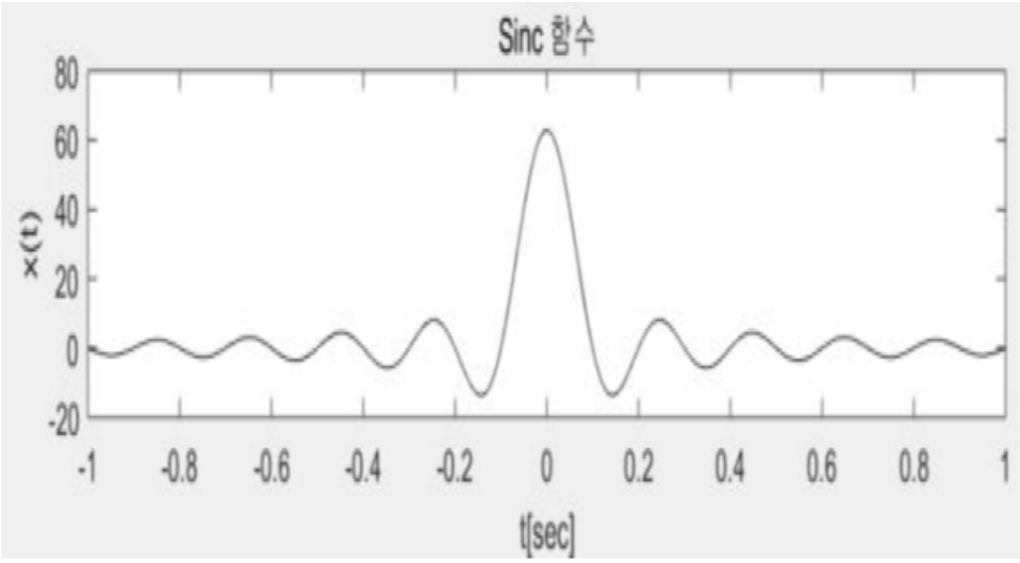
위 사진은 손으로 p(t) 의 크기 스펙트럼을 표현한 것이다. 크기는 1/Ts 이고 이것이 w0마다 반복된다. W0 = 2πfs이다. 1.2 의 파라미터 에서는 크기는 20 이 될 것이고 간격은 20hz 간격으로 나오게 될 것이다.



위 그래프는 Matlab을 사용하여 1.2의 임펄스열 주파수축으로 가져온 것이다. 변환 함수는 강의안에 있는 my_fun_SA 를 사용하였다. Matlab 으로 확인하여 보니 20Hz의 간격마다 임펄스열이 나오는 것은 동일하나 크기는 20이 아니라 0.5가 되는 것을 확인 할 수 있었다.

2.1 다음과 같은 신호 x(t)를 발생시키고 그래프로 표시하라.

x(t) = $\tau sinc(\tau t/2\pi)$ t = 실습 1에서 구한 t사용 τ =20π



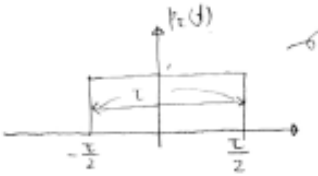
위 그래프는 Matlab을 활용하여 그린 그래프이며 매트랩 내부 함수인 sinc() 함수를 사용하여서 구현하였다.

2.2 신호 x(t)의 Fourier transform 을 손으로 계산하라.

$x(t) = \tau \operatorname{sinc}\left(\frac{\tau t}{2\pi}\right)$

$\mathcal{F}(x(t)) = \mathcal{F}\left(\tau \operatorname{sinc}\left(\frac{\tau t}{2\pi}\right)\right)$

$\Rightarrow \quad X(w) = \int_{-\infty}^{\infty} x(t) e^{-jwt}\, dt$



구형파를 $P_\tau(t)$ 로 정의

$$\mathcal{F}(P_\tau(t)) = P_\tau(w) \Rightarrow \int_{-\infty}^{\infty} P_\tau(t) e^{-jwt}\, dt = \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} P_\tau(t) e^{-jwt}\, dt$$

$$= \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} e^{-jwt}\, dt$$

$$= -\frac{1}{jw}\left[e^{-jwt}\right]_{-\frac{\tau}{2}}^{\frac{\tau}{2}} = -\frac{1}{jw}\left\{e^{-jw\frac{\tau}{2}} - e^{jw\frac{\tau}{2}}\right\}$$

$$= -\frac{1}{jw}\left\{\cos\left(\frac{w\tau}{2}\right) - j\sin\left(\frac{w\tau}{2}\right) - \cos\left(\frac{w\tau}{2}\right) - j\sin\left(\frac{w\tau}{2}\right)\right\}$$

$$= -\frac{1}{jw}\left\{-2j \sin\left(\frac{w\tau}{2}\right)\right\}$$

$$= \frac{2\sin\left(\frac{w\tau}{2}\right)}{w} = \tau \cdot \frac{\sin\left(\frac{w\tau}{2}\right)}{\frac{w\tau}{2}}$$

$$= \tau \operatorname{sinc}\left(\frac{\tau w}{2\pi}\right)$$

Duality 성질

$x(t) \xrightarrow{F.T} X(w)$

$X(t) \xrightarrow{F.T} 2\pi x(-w)$

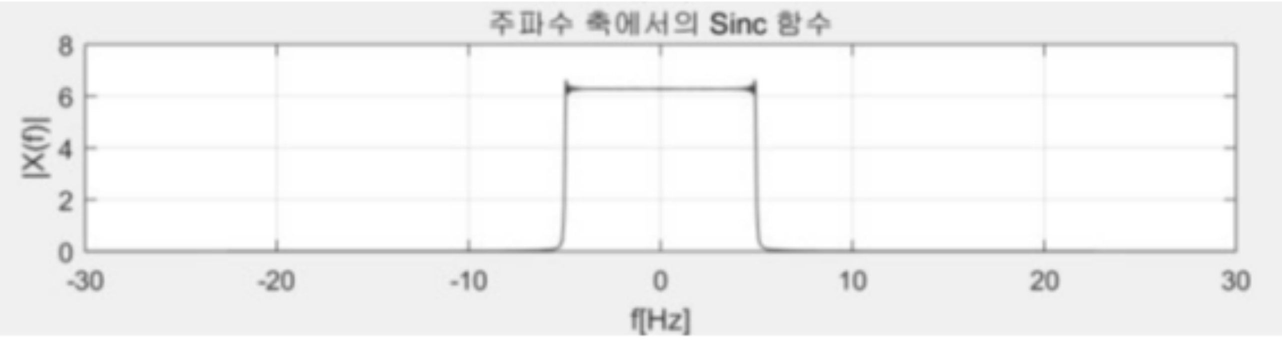
$P_\tau(t) \xrightarrow{F.T} \tau \operatorname{sinc}\left(\frac{\tau w}{2\pi}\right)$

$\tau \operatorname{sinc}\left(\frac{\tau w}{2\pi}\right) \xrightarrow{F.T} 2\pi P_\tau(-w)$

$$\therefore \mathcal{F}\left\{\tau \operatorname{sinc}\left(\frac{\tau t}{2\pi}\right)\right\} = 2\pi P_\tau(w)$$

$$= 2\pi \Pi\left(\frac{w}{\tau}\right)$$

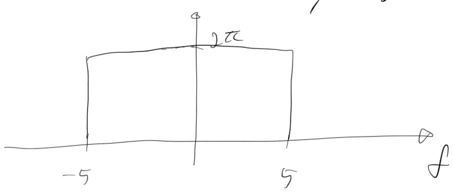
2.3 신호 의 크기 스펙트럼을 (MATLAB으로) 구하여 그래프에 표시하고 손으로 계산한 Fourier Transfom과 비교하라.



위 그래프는 Matlab을 활용하여 그린 그래프이며 주파수 축으로의 이동은 강의안에 기재되어 있는 SA 함수를 사용하였다. 손으로 계산한 값은 다음과 같다.

$$\mathcal{F}\{x(t)\} = 2\pi\Pi\left(\frac{w}{\tau}\right), \quad w = 2\pi f_0, \quad \tau = 20\pi$$

$$= 2\pi\Pi\left(\frac{2\pi f_0}{20\pi}\right) = 2\pi\Pi\left(\frac{f_0}{10}\right) = X(f)$$



Matlab을 통해서 구한 값과 큰 차이가 없어 보인다. 다만 Matlab에서는 이상적인 모습이 아닌 약간의 오차는 있어보인다. 이는 프로그램이 연속 값을 처리 하지 못해서 생기는 것으로 추정된다.

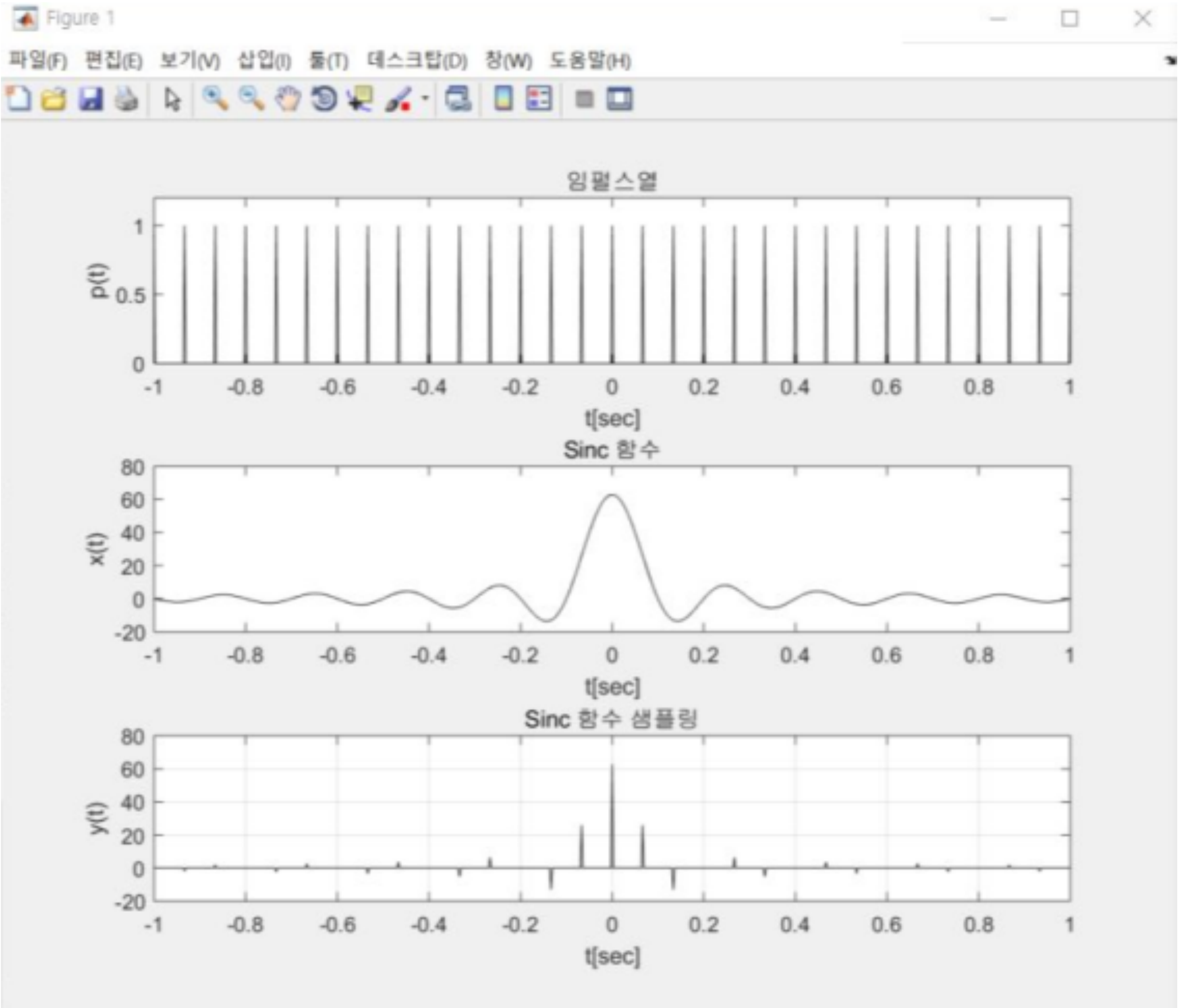
2.4 이 신호를 구성하는 가장 높은 주파수 B Hz는 얼마인가?

이 신호를 구성하는 가장 높은 주파수는 위의 그래프를 보면 알 수 있듯이 5Hz이다.

3.1 앞서 발생한 임펄스열을 이용해 sinc 함수를 표본화하고 표본화된 신호를 그래프에 표시하라. 표본화된 신호 y(t)는 다음과 같다.

$$y(t) = x(t)p(t)$$

위 신호를 구하는 과정에는 큰 어려움이 없었다. 위에 있는 식대로 두 신호를 곱해주면 된다.
다음 그래프는 임펄스열과 sinc함수 그리고 이를 곱해서 나온 신호들의 그래프이다.



3.2 표본화된 신호 y(t)의 Fourier transform 을 손으로 계산하라.

$y(t) = x(t)\,p(t)$

$\Rightarrow \dfrac{\tau}{T_s} \sum_{k=-\infty}^{\infty} \mathrm{Sinc}\left(\dfrac{\tau t}{2\pi}\right) e^{jk\omega_s t}$

$p(t) = \sum_{k=-\infty}^{\infty} \dfrac{1}{T_s} e^{jk\omega_s t}$

$x(t)\,p(t)$는 $\dfrac{\tau}{T_s} \sum_{k=-\infty}^{\infty} \mathrm{Sinc}\left(\dfrac{\tau t}{2\pi}\right) e^{jk\omega_s t}$ 이므로

$x(t) = \tau\,\mathrm{Sinc}\left(\dfrac{\tau t}{2\pi}\right)$

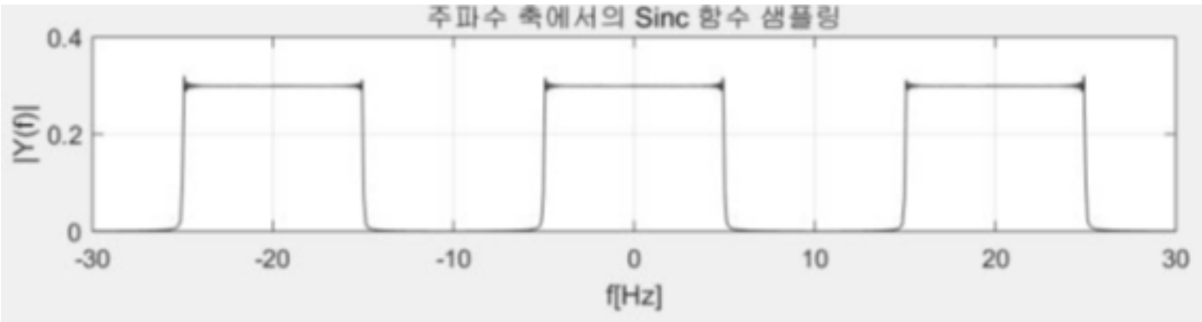
$\mathcal{F}(x(t)) = \mathcal{F}\left(\tau\,\mathrm{Sinc}\left(\dfrac{\tau t}{2\pi}\right)\right) = P_\tau(\omega)$

$\mathcal{F}(y(t)) = \mathcal{F}\left(\dfrac{\tau}{T_s} \sum_{k=-\infty}^{\infty} \mathrm{sinc}\left(\dfrac{\tau t}{2\pi}\right) e^{jk\omega_s t}\right.$

$Y(\omega) = \dfrac{1}{T_s} \sum_{k=-\infty}^{\infty} \mathcal{F}\left(x(t)\, e^{jk\omega_s t}\right)$

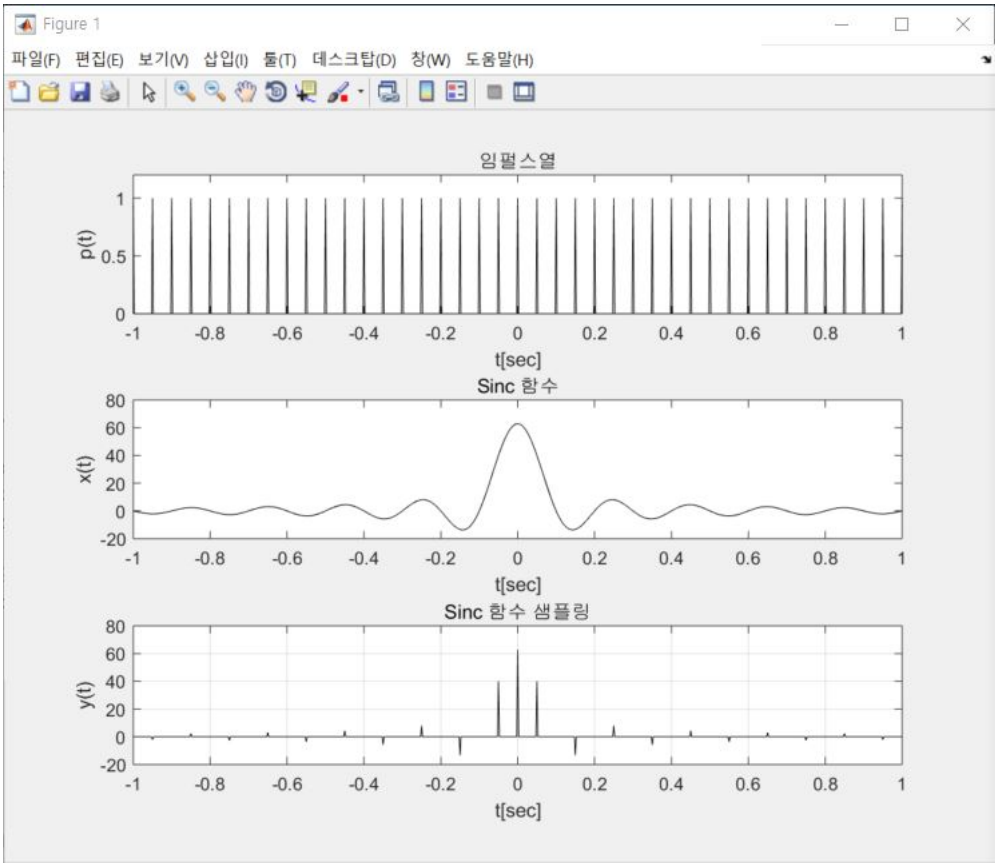
$= \dfrac{1}{T_s} \sum_{k=-\infty}^{\infty} X(\omega - k\omega_s)$

3.3 표본화된 신호 y(t)의 크기 스펙트럼을 구하여 그래프에 표시하고 이와 같은 스펙트럼이 나오는 이유를 설명하라.

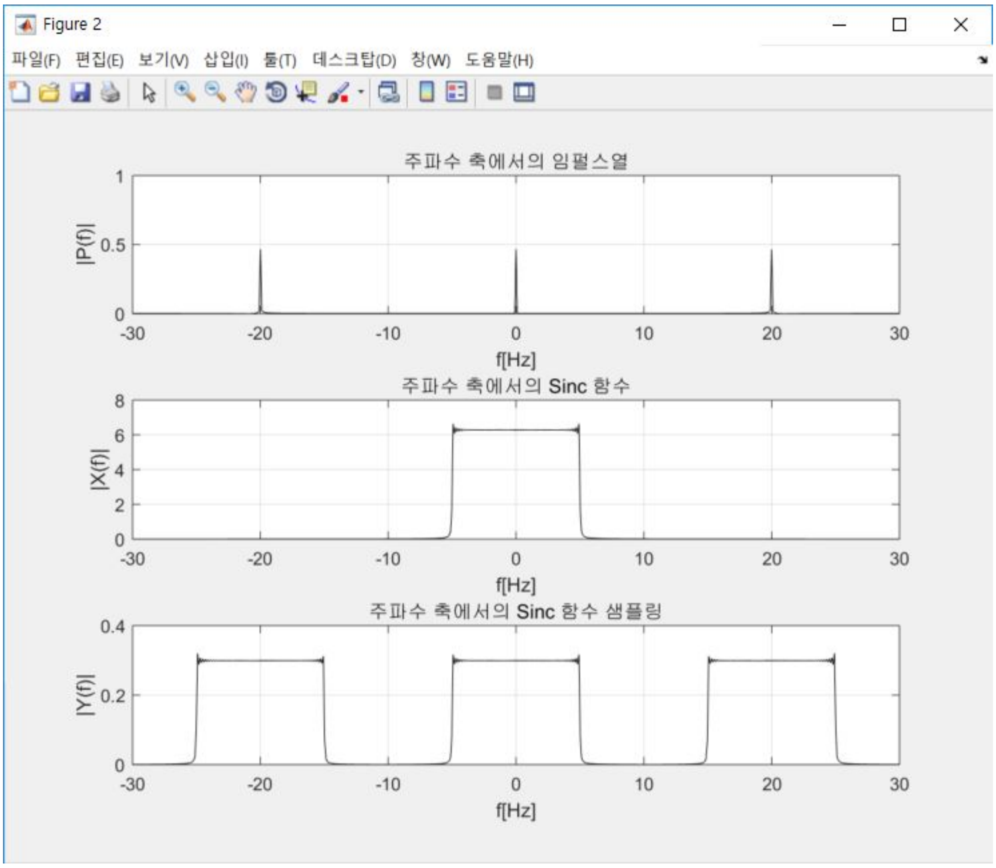


다음 그래프는 위에서 구한 신호 y(t)를 주파수축에서 관찰한 것이다. 주파수 축에서의 임펄스열과 주파수 축의 sinc 함수 즉 이산신호와 주기신호의 곱으로 델타 신호가 없는 값을 중심으로 sinc 함수를 Fourier transform 한 구형파가 나오는 것을 볼 수 있다.

4. 시간영역과 주파수 영역에서의 표본화 비교



[fs = 20hz 시간영역]



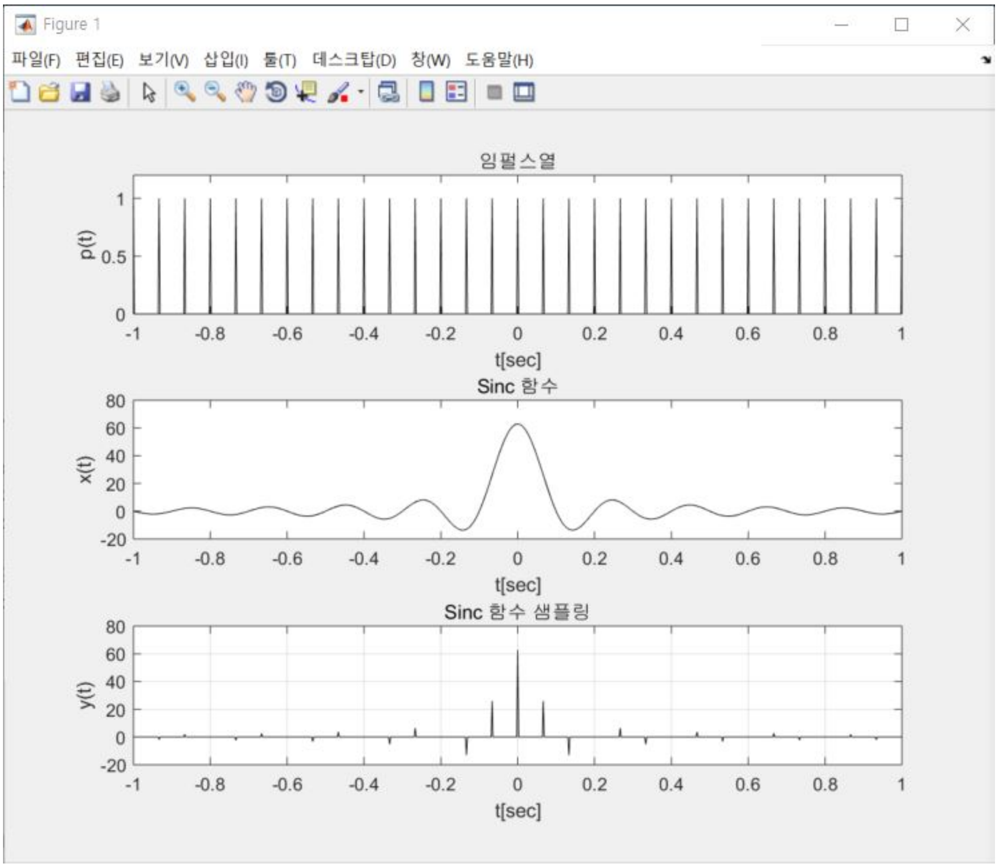
[fs = 20hz 주파수 영역]

- 시간영역과 주파수 영역에서의 표본화 과정 비교

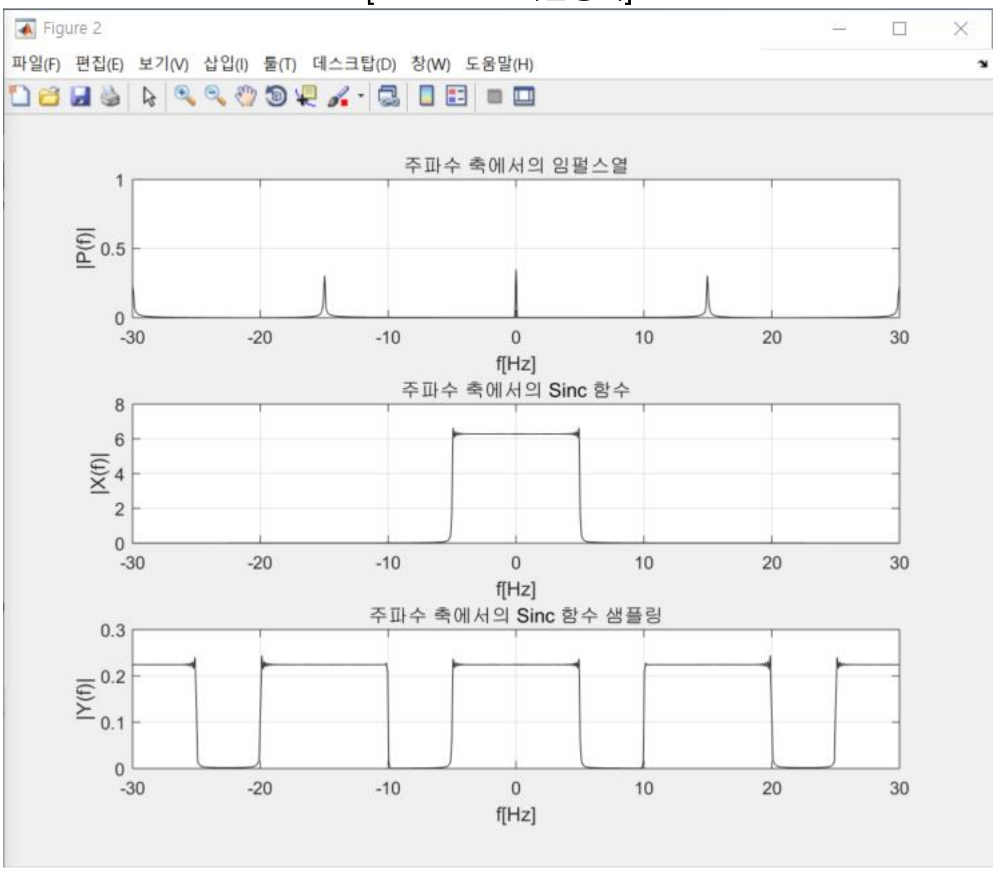
위 그래프는 Sampling 주파수를 20hz 로 설정한 후 임펄스열, sinc 함수 ,표본화된 sinc 함수이다. 이 세가지 신호들을 시간영역과 주파수 영역에서 비교를 차근차근 진행해 보도록 하겠다. 먼저 임펄스열을 시간축에서 살펴보고자 한다. 표본화 주파수가 20hz 이므로 주기는 1/20[sec]가 된다. 즉 0.05초마다 1의 값을 가지게 된다. 이를 주파수 영역으로 가져오게 되면 20hz 마다 1의 값이 반복되는 것을 볼 수 있다. Sinc 함수를 시간축에서 바라보면 그냥 Sinc 함수이다. 이를 주파수 영역에서 바라보게 된다면 구형파의 모습을 띄게 된다. 이는 Fourier Transform을 통해서 알수 있다. Sinc 함수와 임펄스열을 곱하게 되면 sinc 함수를 표본화한 것이 되는 신호를 볼 수 있다. 이를 주파수 영역에서 바라보게 되면 20hz를 주기로 반복하게 된다. 이는 표본화된 sinc 함수는 이산 비주기 신호이며 이를 푸리에 변화 시키면 반복 스펙트럼이 되기 때문이다. 이에 대한 자세한 설명은 다음 실습때 할 DFT에서 자세하게 나올 것 같다.

- Nyquist Sampling Rate에 대해서 설명하라.

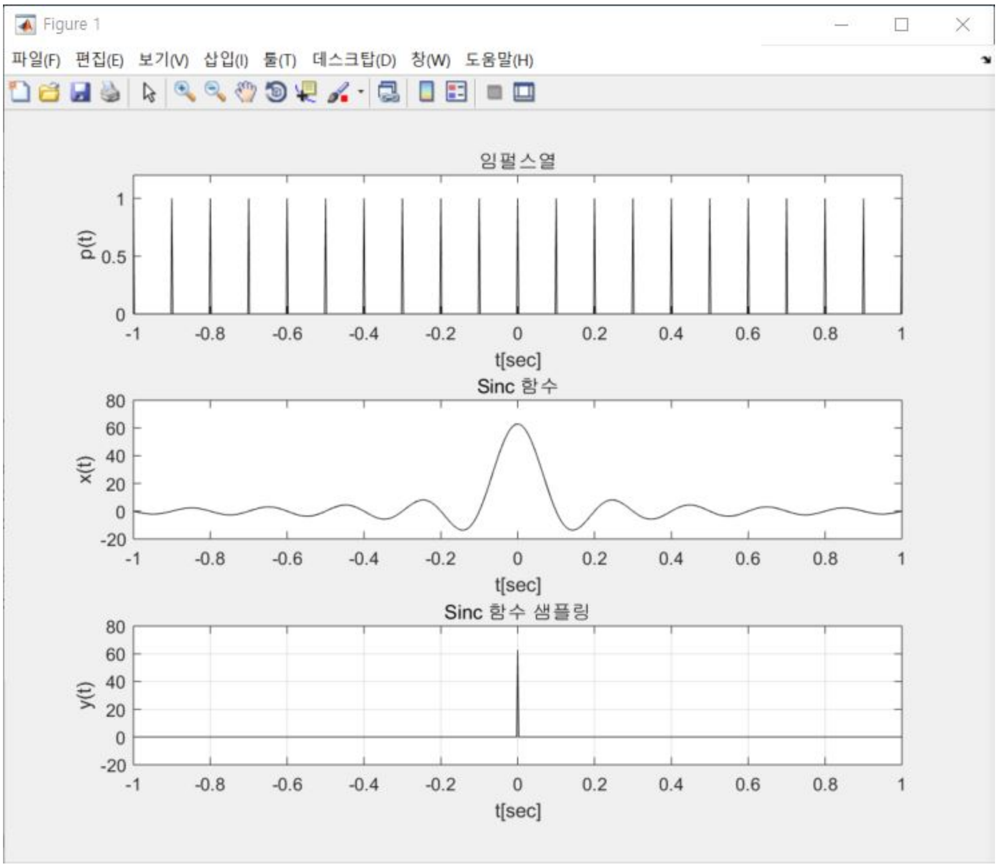
나이키스트 샘플링 속도라 함은 디지털 전송에서 부호 간 간섭을 없애기 위해, 입력 신호의 최고 주파수의 두배 이상의 주파수에서 표본화 하여 원신호를 충실하게 재현함으로써 디지털 부호 1과 0을 전달하는 속도를 의미한다. 아래의 그래프는 표본화 주파수를 15,10hz 로 설정한 것이다. 15hz 에서는 어느정도 구형파가 제대로 나오는 것을 알 수 있지만 10 hz 부터는 표본화된 sinc 함수부터 제대로 보여지지 않으며 여기서 나오는 구형파도 제대로된 모습이 아닌 것을 확인 할 수 있다.



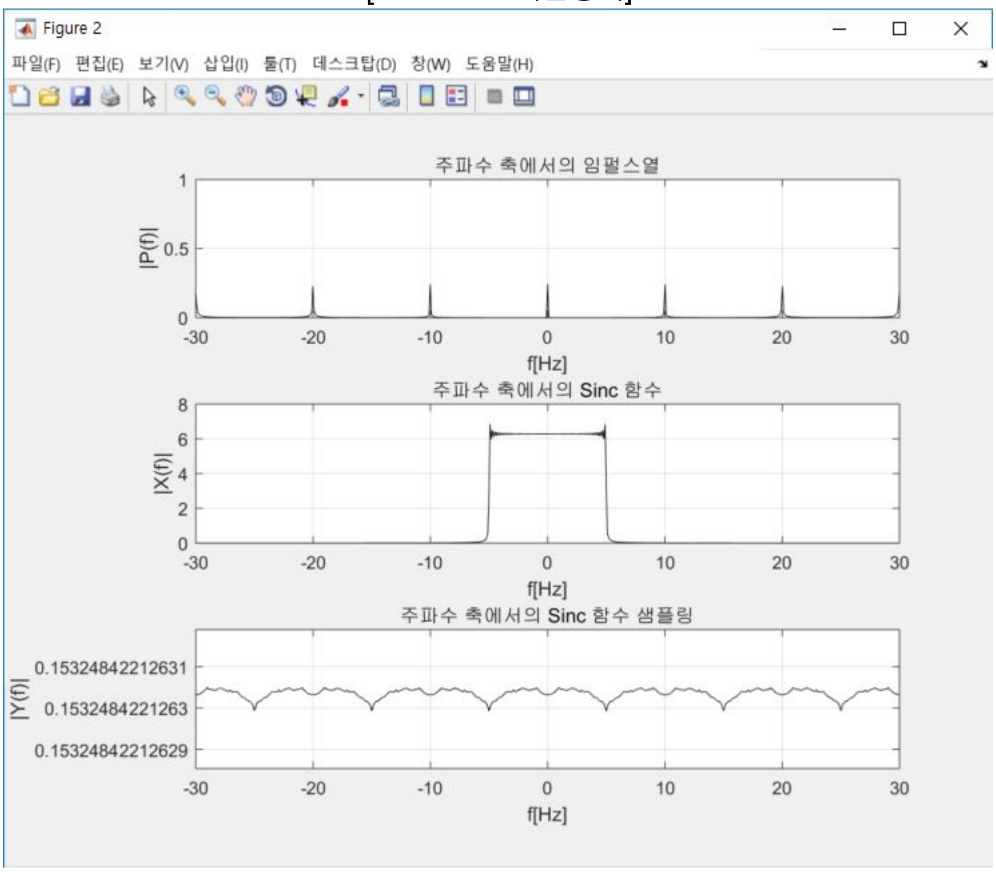
[fs =15hz 시간영역]



[fs =15hz 주파수 영역]



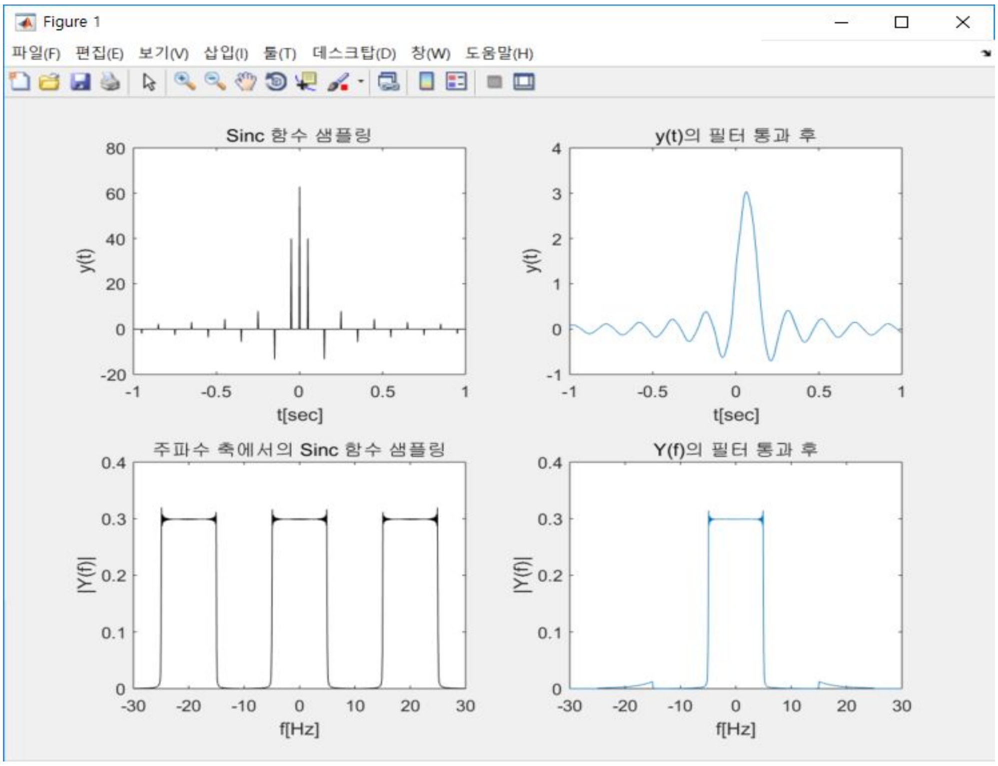
[fs =10hz 시간영역]



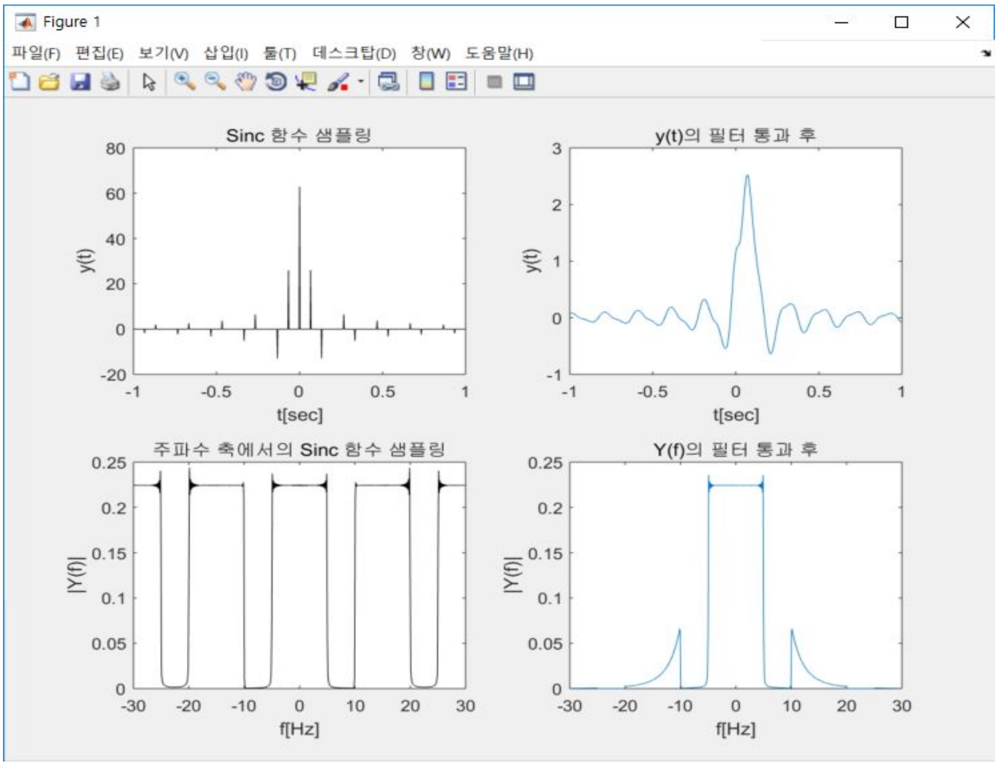
[fs =10hz 주파수 영역]

4. LPF를 이용한 복원

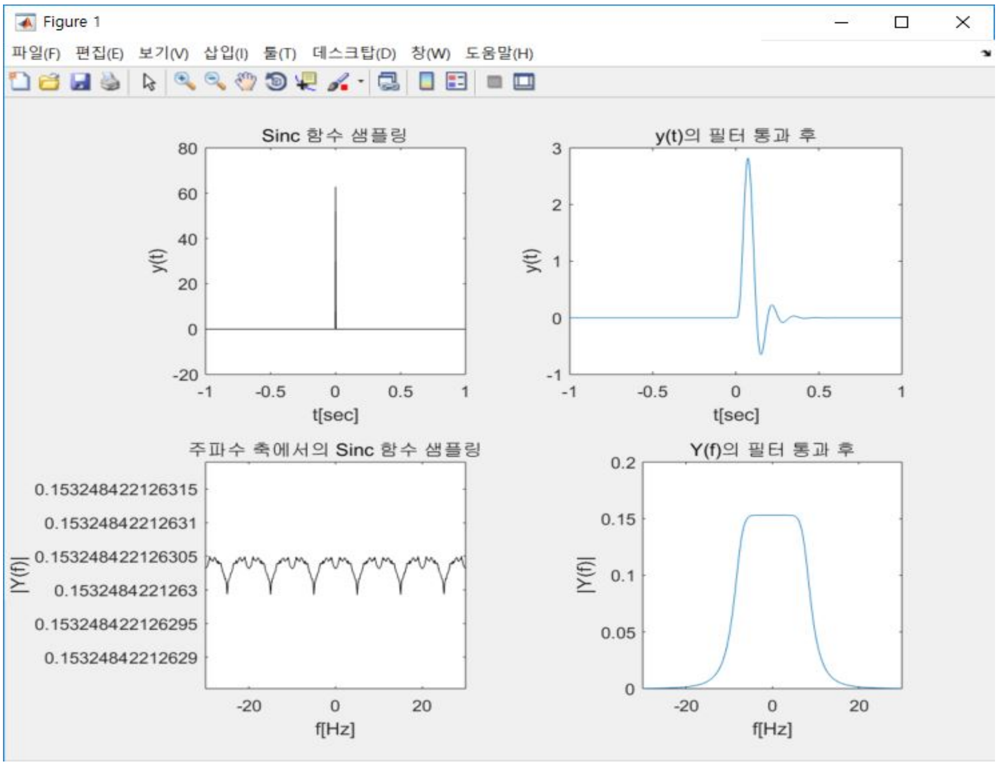
본 실습은 표본화된 신호를 원신호로 복원하는 과정이다. 필터의 코드는 강의안에 있는 것으로 사용하였다. Cutoff 주파수는 8Hz이며 5차 버터워스 LPF를 이용하였다.



[fs = 20Hz]



[fs = 15Hz]



[fs = 10Hz]

복원했을때도 알수 있듯이 20, 15 Hz에서는 복원이 어느정도 잘 이루어져 있다고 말할 수 있지만 10Hz에서는 제대로된 복원이 이루어지지 않는 것을 볼 수 있다.