HomeWork (MATLAB Signal & System) - Part #1

2019102136 최성준

Q1 : A triangular pulse $x\,t$ is defined by

$$x(t) = 1 - 1/5\,|t|\ ,\ t \le 5$$
$$0,\ \text{otherwise}\ .$$

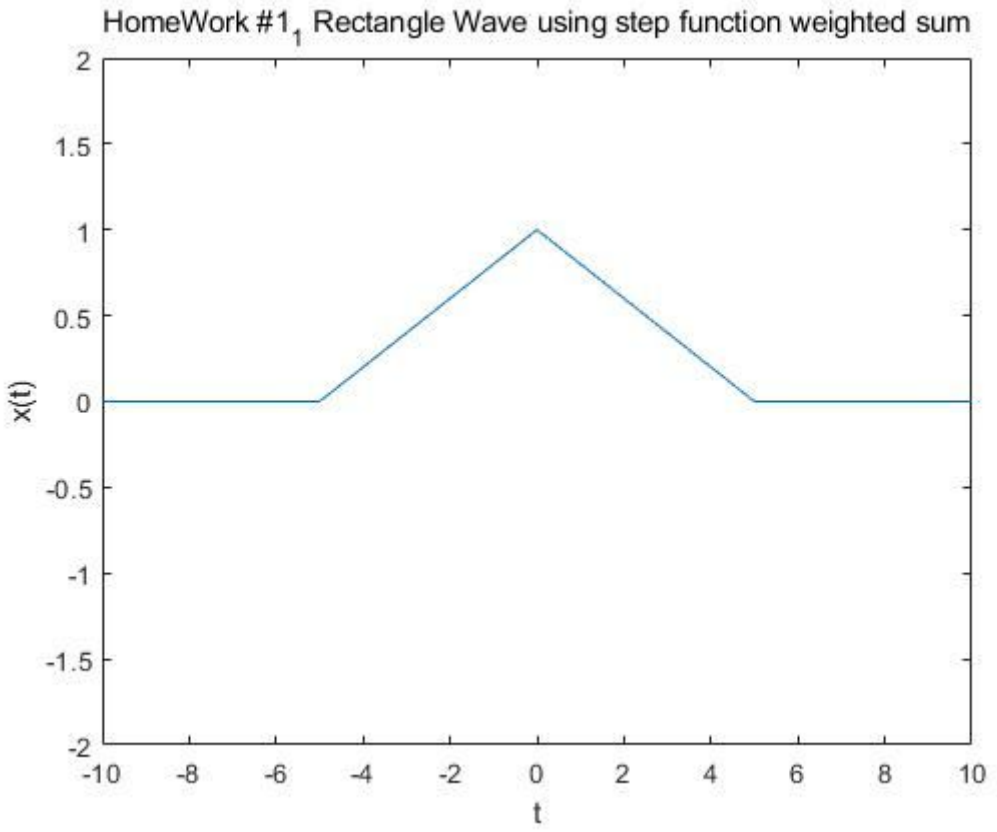
Write a MATLAB program to generate $x(t)$ for $|t| < 10$ and plot it.

---------- < CODE > ----------

```
clear all
t = -10:.01:10;

for n = -1000:1000
   t0 = n*0.01;
   if abs(t0) <= 5
      x(n+1001) = 1 - 1/5*abs(t0);
   else
      x(n+1001) = 0;
   end
end

plot(t,x)
xlabel("t")
ylabel("x(t)")
title(["HomeWork #1_1 Rectangle Wave using step function weighted sum"])
axis([-10 10 -2 2])
```



분석 : 위의 조건을 만족하는 함수를 만들기 위해서 t를 -10~10까지 0.01간격의 수로 설정해 준 후 for문을 이용하여 2000번 반복하여 함수 x를 구성하였다. x를 구성함에 있어 절댓값 함수를 사용하여 **abs(t0)가 5보다 작은 범위**에 한해서 **1 - 1/5*abs(t0)**를 갖도록 구성하였으며, 그 외는 0으로 처리하였다. 처음에는 단순 t의 범위만을 지정해주어 그래프를 그려주었으나 사다리꼴의 모양이 나타나서 **미세하게 sampling**을 해주어 원하는 그래프를 도출할 수 있었다.

Q2 : A raised cosine sequence is defined by

$x[n] = \cos 0.05\pi n + 1, \quad -20 \le n \le 20$

$\quad\quad 0, \text{ otherwise .}$

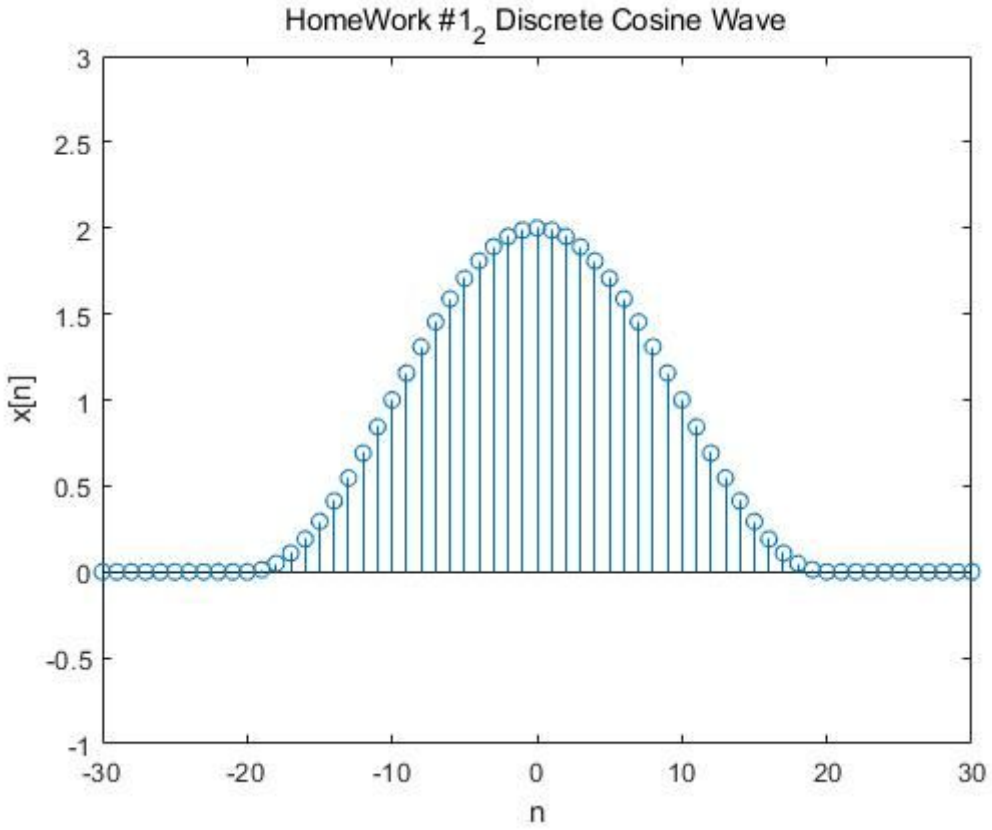
Write a MATLAB program to generate $x[n]$ for $-30 \le n \le 30$, and plot it.

---------- < CODE > ----------

```
n = -30:30;
y = cos(0.05*pi*n) + 1;

u1 = heaviside(n-20);
u2 = heaviside(n+20);
u = u2-u1;

stem(n, u.*y) % y에 u(step function weighted sum)를 곱해줌으로써 -10:10이외의 값은 0처리.
xlabel("n")
ylabel("x[n]")
title(["HomeWork #1_2 Discrete Cosine Wave"])
axis([-30 30 -1 3])
```



분석 : 함수는 단순 cos인이라는 내장함수를 활용하여 그릴 수 있었으나 문제는 otherwise를 구현하는 과정이었다. 이 문제를 어떻게 해결할까 생각하던 중 0에서 1로 변화하는 지점을 parameter로 받는 **heaviside라는 내장함수를 활용**하여 **step function을 구현한 후 weighted sum**을 통해 t값을 대입했을때 heaviside(t+20)에서 heaviside(t-20)를 뺀 값을 반환하는 step function을 구현하였다. 이후 n값의 범위가 **-30~30까지인 주어진 함수를 step function과 행렬 곱(.*)**해주었더니 원하는 함수를 그릴 수 있었다. 이외에도 몇가지 문제가 더 있었다.

1. n의 범위를 -20:20로 설정하니 other을 구현했어도 그래프가 표현되지 않았다. 이는 조금 투박하기는 하나 단순하게 **범위(-30:30)를 넓혀주어** 해결하였다.
2. 문제를 자세히 읽어보니 continuous한 함수를 구현하는 것이 아닌 discrete한 함수를 구현하는 것이었다. 이는 plot을 사용하는 것이 아닌 parameter는 동일한 **stem이라는 함수를 통해 구현**하였다.

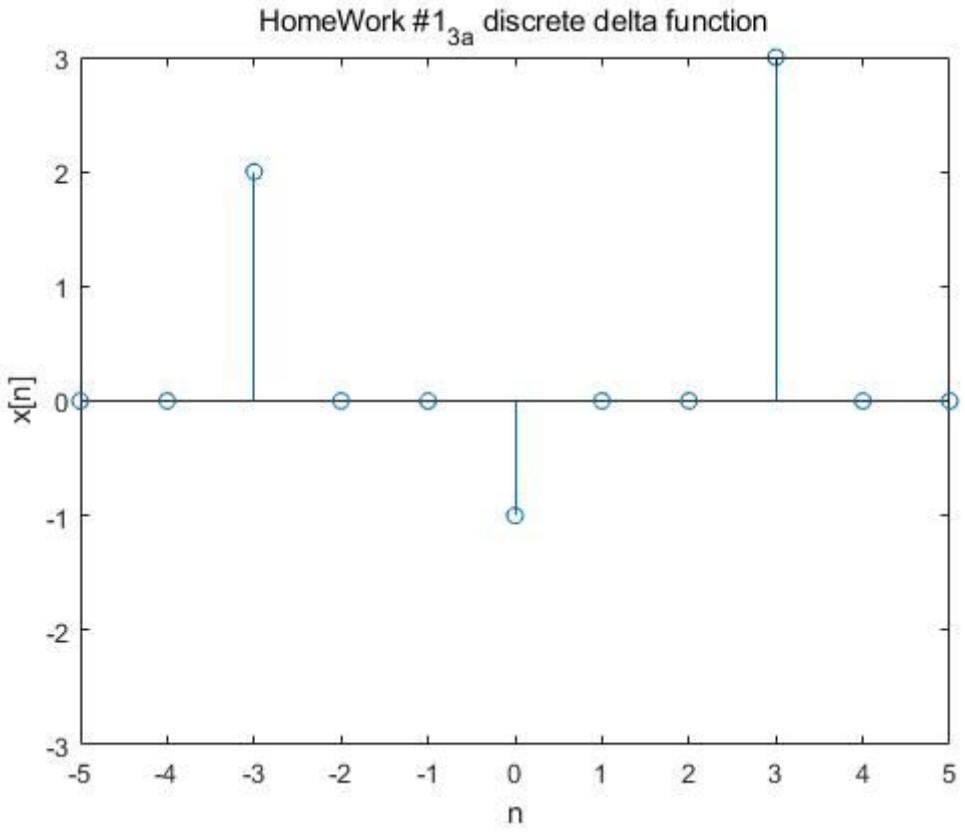
Q3 : Plot the following sequences using MATLAB.
a. $x[n] = 2\delta[n] + 3 - \delta[n] + 3\delta[n] - 3$, $-5 \le n \le 5$

---------- < CODE > ----------

```
clear all
n = -5:5;

impulse1 = n==3;
impulse2 = n==0;
impulse3 = n==-3;
u = 3*impulse1-impulse2+2*impulse3;

stem(n, u)
ylabel('x[n]')
title('HomeWork #1_3_a discrete delta function')
axis([-5 5 -3 3])
```



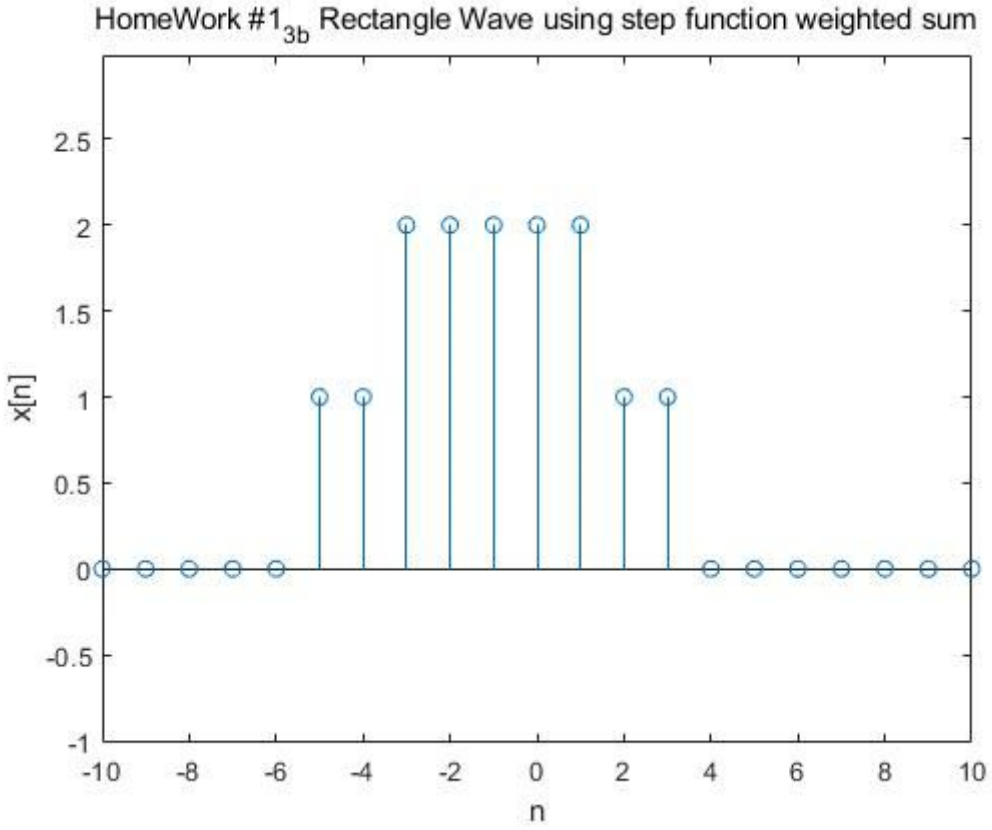
분석 : delta function을 구하는 방법은 impulse를 따로 구현을 하는 방법이 있어 참고하여 구현한 후 함수를 원하는 범위로 구성하고 **stem함수**를 활용하여 discrete하게 바꿔주었다. 그러고나서 소스코드를 검토하다가 위의 impulse함수를 구현하는 방법이 단순 대입이었다는 생각이 들었다. **impulse = n==0;**이렇게 impulse라고 선언해주어 따로 구현하는 방법인 줄 알았으나 다른 분이 임시로 지정한 변수가 impulse였던 것이었다. 또한 impulse로 지정한 값 외의 값은 어떻게 0으로 처리할까를 생각해봤는데, **y에 u(step function weighted sum)를 곱해줌으로써 처리**한다는 것을 알았다.

Q3 : Plot the following sequences using MATLAB.
b. $x[n] = u[n+5] + u[n+3] - u[n-2] - u[n-4]$, $-10 \le n \le 10$

---------- < CODE > ----------

```
clear all
t = -10:10;
u1 = heaviside(t+5);
u2 = heaviside(t+3);
u3 = heaviside(t-2);
u4 = heaviside(t-4);
u = u1+u2-u3-u4;

stem(t,u)
xlabel("n")
ylabel("x[n]")
title(["HomeWork #1_3_b Rectangle Wave using step function weighted sum"])
axis([-10 10 -1 3])
```



분석 : 이 문제는 **step function이 weighted sum되어있는 것**이다. **'Q2'**에서 이용한 함수인 **heaviside라는 내장함수를 활용**하여 각각의 step function을 구현한 후 앞에서 언급했듯이 step function의 weighted sum을 통해 주어진 함수를 구했다.

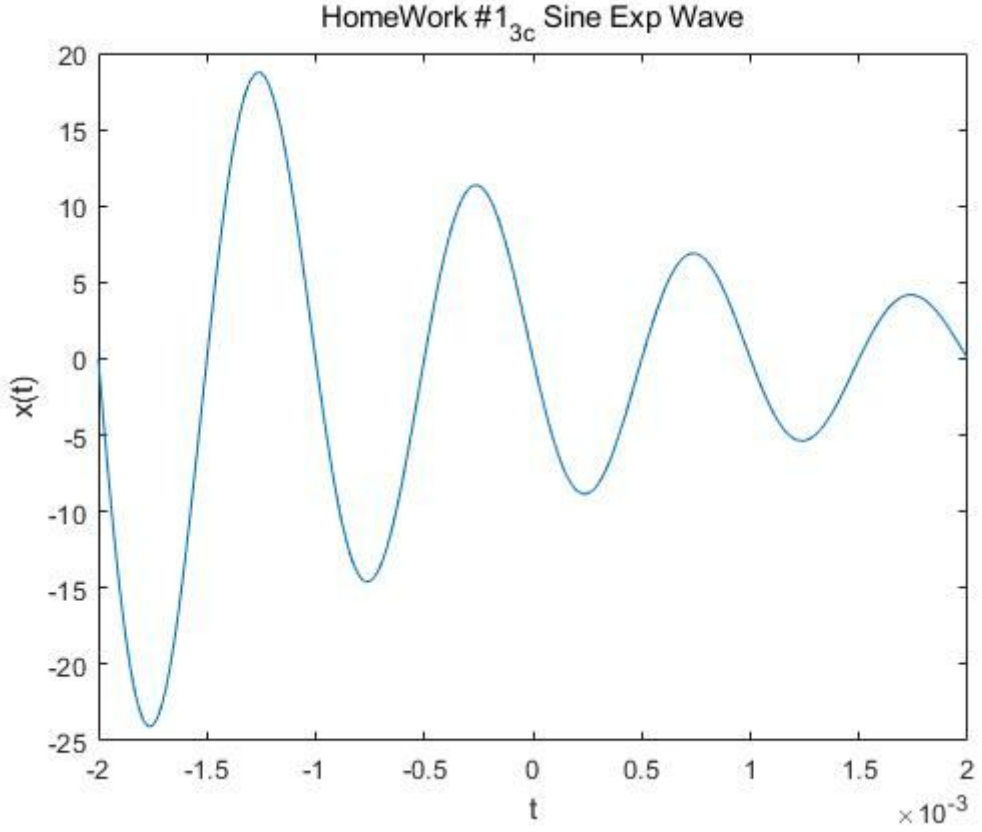
Q3 : Plot the following sequences using MATLAB.
c. x(t) = 10*sin (2000πt − π/3)*e −500t , −2 ≤ t ≤ 2 [msec.]

---------- < CODE > ----------

```
clear all
t = [-2:.01:2]/1000;

x1 = 10*sin(2000*pi*t - pi*3);
x2 = exp(-500*t);

x = x1.*x2;
plot(t,x)
xlabel("t")
ylabel("x(t)")
title(["HomeWork #1_3_c Sine Exp Wave"])
```



분석 : 간단하게 생각했으면 쉬웠을 문제를 '**Q1**'번과 같이 for문을 이용해보려다가 적절한 변수를 설정하지 않아서 생각보다 고전한 문제였다.
먼저 t라는 변수의 단위가 msec이므로 sec단위의 벡터를 1000으로 나눠주었으며, 이를 대입한 sin함수와 지수함수를 각각 구현하여 행렬곱을 하는 방식으로 문제를 해결했다.
지수함수를 구현하기 위해서는 **exp()라는 내장함수**를 활용하였고, cos함수를 구현하기 위해서는 **cos()라는 내장함수**를 활용하였다. 이후 각각을 곱하는 과정에서 위의 문제에서도 사용했었지만 **행렬 곱셈이 아닌 요소별 곱셈을 수행하기 위해 .* 연산자를 활용**하여 최종적인 x[n]을 구현하였다.