HomeWork_2020f (MATLAB Signal & System) - Part #2

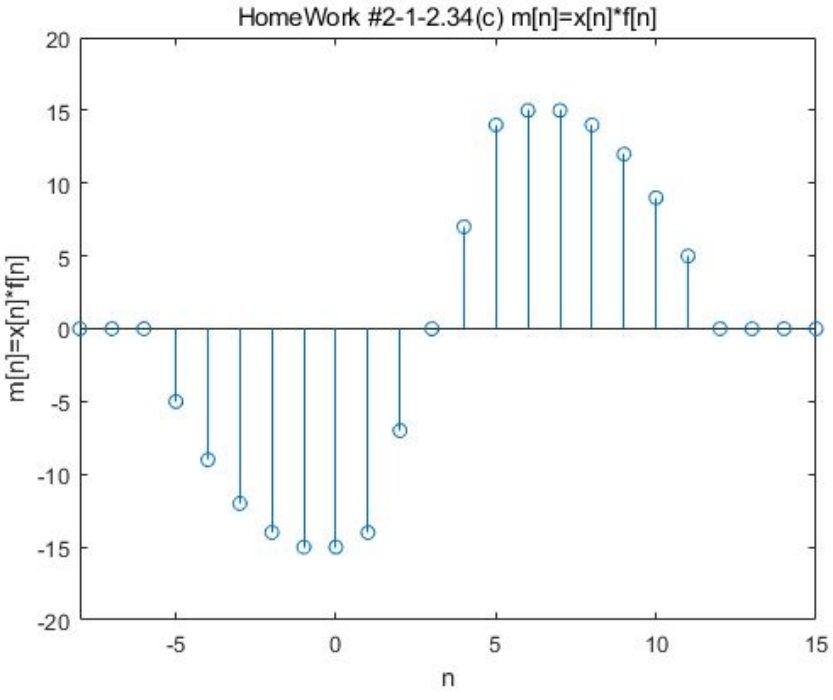
2019102136 최성준

Q 1-1 : Solve Problem 2.34 (c), using the MATLAB conv command.

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

```
clear all
x = [zeros(1,4), ones(1,7), zeros(1,4)];
t = -5:1:5;
for n = -5:1:5
   if abs(n) <= 5
      f_(n+6) = n;
   else
      f_(n+6) = 0;
   end
end
f__ = ones(1,11);
f = f_.*f__;
m = conv(x, f);

t = -9:1:15;
stem(t, m)
xlabel('n')
ylabel('m[n]=x[n]*f[n]')
title('HomeWork #2-1-2.34(c) m[n]=x[n]*f[n]')
axis([-8 15 -20 20])
```



Q 1-2 : Solve Problem 2.34 (g), using the MATLAB conv command.

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

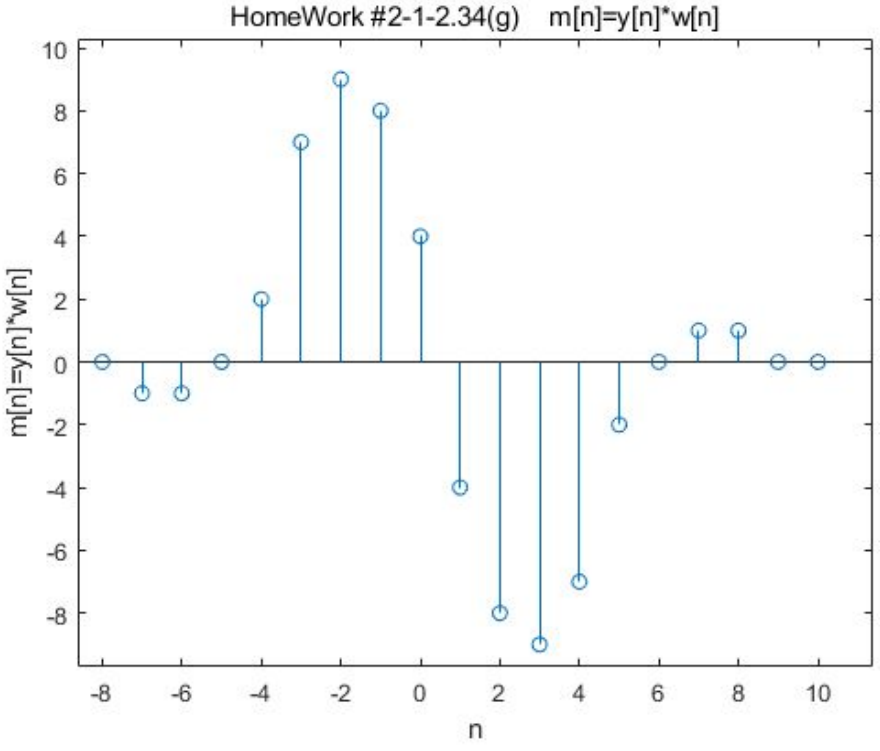
```
%% HomeWork #2_1_2.34(g)
% m[n]=y[n]*w[n]

clear all

y = [zeros(1,2), ones(1,4), -ones(1,4), zeros(1,2)];
t = -10:1:10;

for n = -4:1:4
   if abs(n) <= 4
      w(n+5) = 3 - abs(n);
   else
      w(n+5) = 0;
   end
end
m = conv(y, w);

t = -9:1:10;
stem(t, m)
xlabel('n')
ylabel('m[n]=y[n]*w[n]')
title('HomeWork #2-1-2.34(g)    m[n]=y[n]*w[n]')
axis([-10 10 -10 10])
```



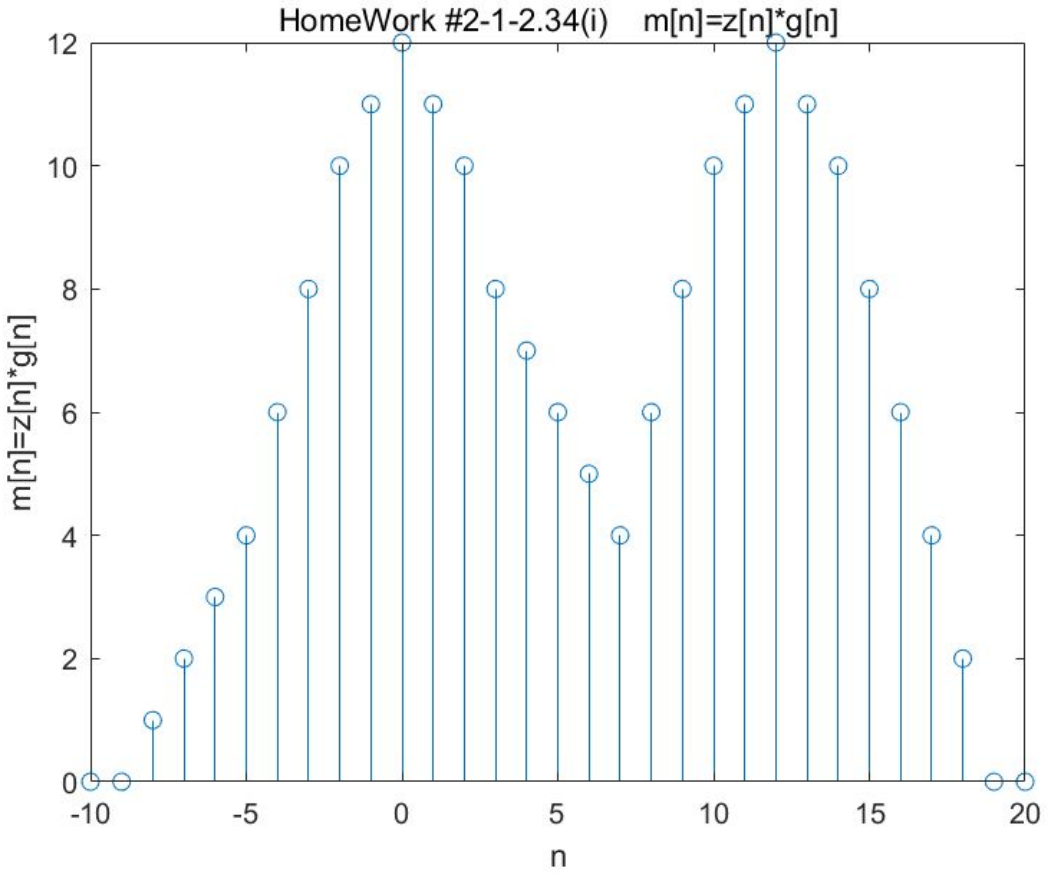
Q 1-3 : Solve Problem 2.34 (i), using the MATLAB conv command.

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

```
%% HomeWork #2_1_2.34(i)
% m[n]=z[n]*g[n]
clear all

z = [zeros(1,8), ones(1,4), 2*ones(1,5), zeros(1,2)];
g = [ones(1,7), zeros(1,5), ones(1,7)];
m = conv(z, g);

t = -16:1:20;
stem(t, m)
xlabel('n')
ylabel('m[n]=z[n]*g[n]')
title('HomeWork #2-1-2.34(i)    m[n]=z[n]*g[n]')
axis([-10 20 0 12])
```



분석 : 특정 신호들을 **conv함수**를 이용하여 **convolution sum**과정을 진행했다. discrete signal이기에 **stem**을 이용하여 그래프로 나타냈다. convolution sum한 값을 그래프로 나타내기 위해서 x축의 범위를 찾고, 일일이 지정해주는 과정에서 조금 비효율적이라는 생각이 들었다. 하지만 일일이 conv함수가 해주는 일을 일일이 계산한다면 더욱 오레걸리며 시각적으로 깔끔하게 보이기에 MATLAB이라는 tool을 이용하는 것이 더욱 효율적이다.

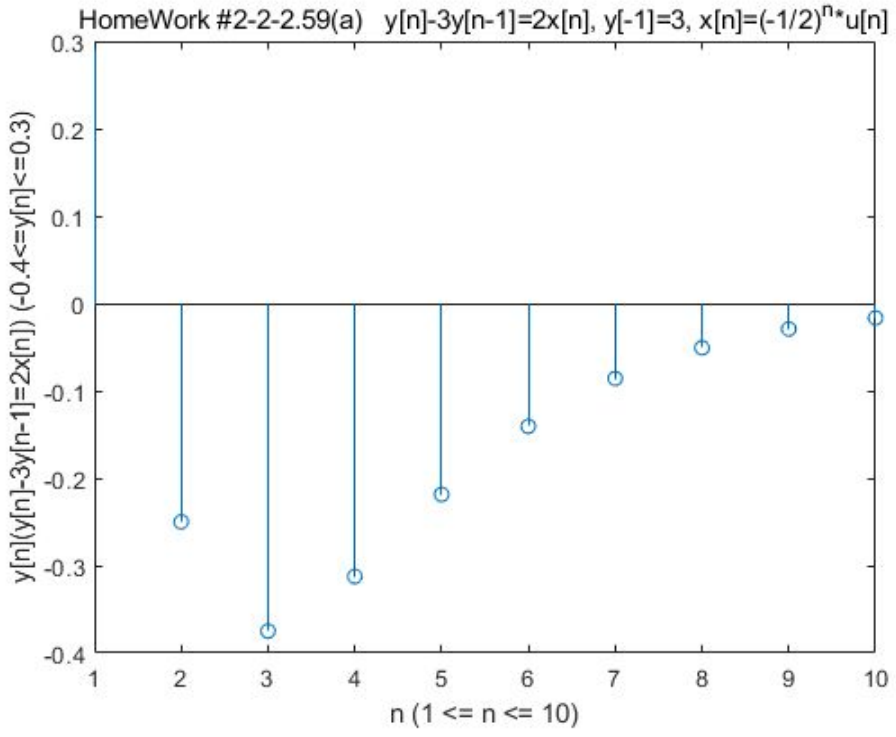
Q 2-1 : Use the MATLAB commands filter and filtic to determine the first 50 output values in Problem 2.59 (a)

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

```
%% HomeWork #2_2_2.59(a)
% y[n]-3y[n-1]=2x[n], y[-1]=3, x[n]=(-1/2)^n*u[n]
clear all

input = 10; % 입력의 개수(임의지정가능)
b = [2 0];
a = [1 -1/2]; % a의 첫 번째 요소가 1이 아니면 filter는 차분 방정식을 구현하기 전에 모든 계수를 a(1)로 나눔.
t = 1:10;
for n = 1:1:10
   f(n) = -1/2.^n;
end
x = ones(1, input).*f;
i = filtic(b,a,3); % filter initial condition 설정(b(행벡터) - forward loop 계수(x계수), a(행벡터) - feedback loop 계수(y계수))
y = filter(b,a,x,i); % filter 함수를 이용한 시스템 응답

n = 1:input; % 입력의 개수
stem(n, y)
xlabel('n (1 <= n <= 10)')
ylabel('y[n](y[n]-3y[n-1]=2x[n]) (-0.4<=y[n]<=0.3)')
title('HomeWork #2-2-2.59(a)   y[n]-3y[n-1]=2x[n], y[-1]=3, x[n]=(-1/2)^n*u[n]')
axis([1 input -0.4 0.3])
```



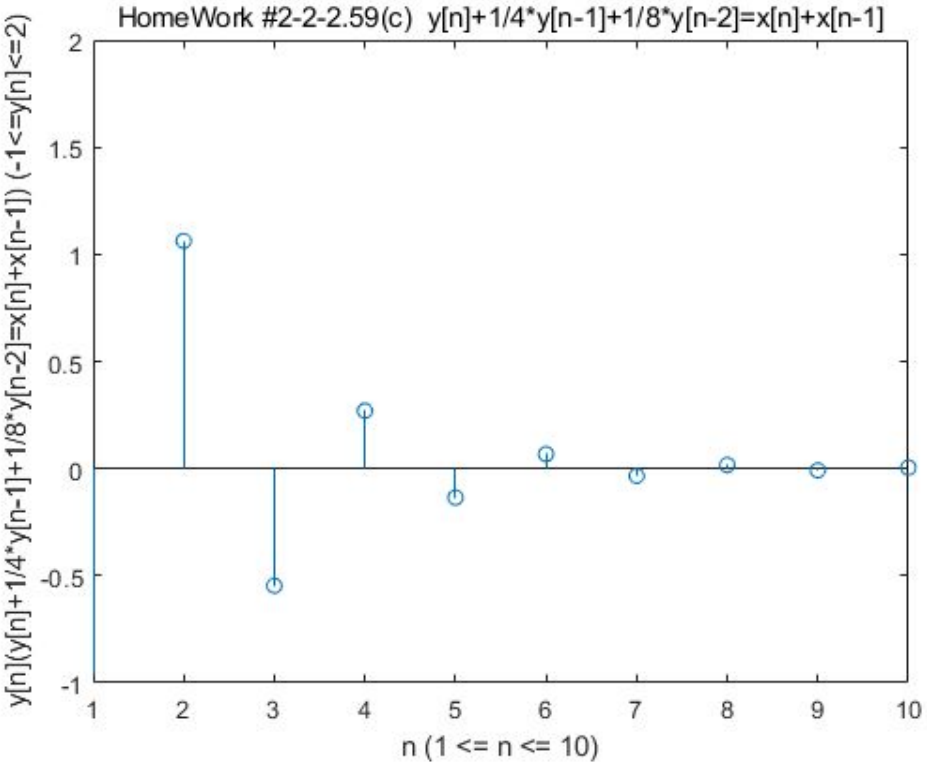
Q 2-2 : Use the MATLAB commands filter and filtic to determine the first 50 output values in Problem 2.59 (c)

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

```
%% HomeWork #2_2_2.59(d)
% y[n]+1/4*y[n-1]+1/8*y[n-2]=x[n]+x[n-1], y[-1]=4, y[-2]=--2, x[n]=(-1).*u[n]
clear all

input = 10; % 입력의 개수(임의지정가능)
b = [1 1 0];
a = [1 1/4 -1/8];
t = 1:10;
for n = 1:1:10
   f(n) = (-1).^n;
end
x = ones(1, input).*f;
i = filtic(b,a,[4 -2]);
y = filter(b,a,x,i);

n = 1:input; % 입력의 개수
stem(n, y)
xlabel('n (1 <= n <= 10)')
ylabel('y[n](y[n]+1/4*y[n-1]+1/8*y[n-2]=x[n]+x[n-1]) (-1<=y[n]<=2)')
title('HomeWork #2-2-2.59(c)  y[n]+1/4*y[n-1]+1/8*y[n-2]=x[n]+x[n-1], y[-1]=4, y[-2]=--2,
x[n]=(-1).*u[n]')
axis([1 input -1 2])
```



분석 : 특정 신호들을 **filtic**를 이용하여 **initial condition을 지정**한 후 **filter**를 이용하여 **difference equation을 구현**했다. discrete signal이기에 **stem**을 이용하여 그래프로 나타냈다. **filtic에서 두개의 행벡터(입력계수벡터, 출력계수베터)와 초기값(y[-1],y[-2]순)**을 인수로 받아 **filter의 인수에 넘겨줌**으로써 difference equation을 구성한다. 입력개수를 의미하는 상수를 바로 넣은 것이 아닌 **변수 input에 상수를 지정**해두어 **변수값을 변경하면 input에 따른 결과로 바뀌도록 구성**하였다. 또한 matlab은 vector index를 구성할때 zero-based가 아닌 **시작수를 1에 저장**한다.