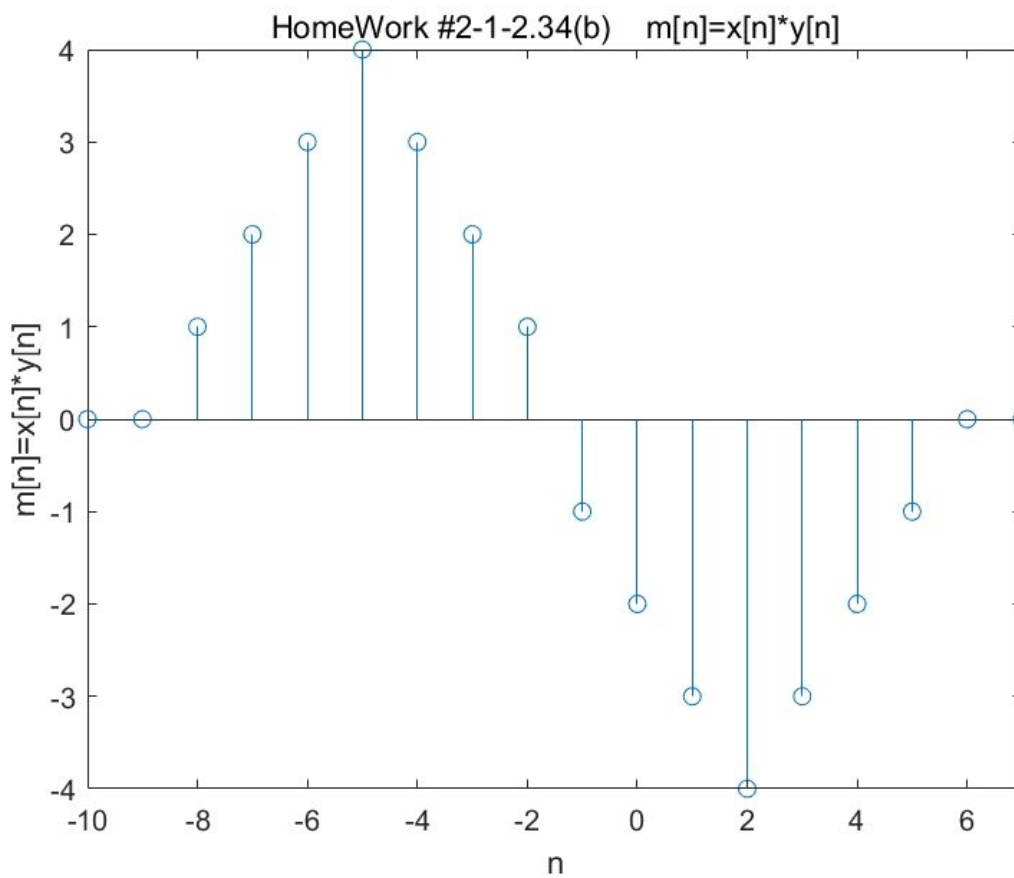


HomeWork (MATLAB Signal & System) - Part #2

2019102136 최성준

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

```
----- < CODE > -----  
%% HomeWork #2_1_2.34(b)  
%  $m[n]=x[n]*y[n]$   
clear all  
  
x = [ones(1,7), zeros(1,3)];  
y = [zeros(1,2), ones(1,4), -ones(1,4)];  
m = conv(x, y);  
  
t = -10:1:8;  
stem(t, m)  
xlabel('n')  
ylabel('m[n]=x[n]*y[n]')  
title('HomeWork #2-1-2.34(b)   $m[n]=x[n]*y[n]$ ')  
axis([-10 7 -4 4])  
-----
```



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

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

```
%% HomeWork #2_1_2.34(e)
```

```
% m[n]=y[n]*z[n]
```

```
clear all
```

```
y = [ones(1,4), -ones(1,4), zeros(1,5)];
```

```
z = [zeros(1,3), ones(1,4), 2*ones(1,5)];
```

```
m = conv(y, z);
```

```
t = -6:1:17;
```

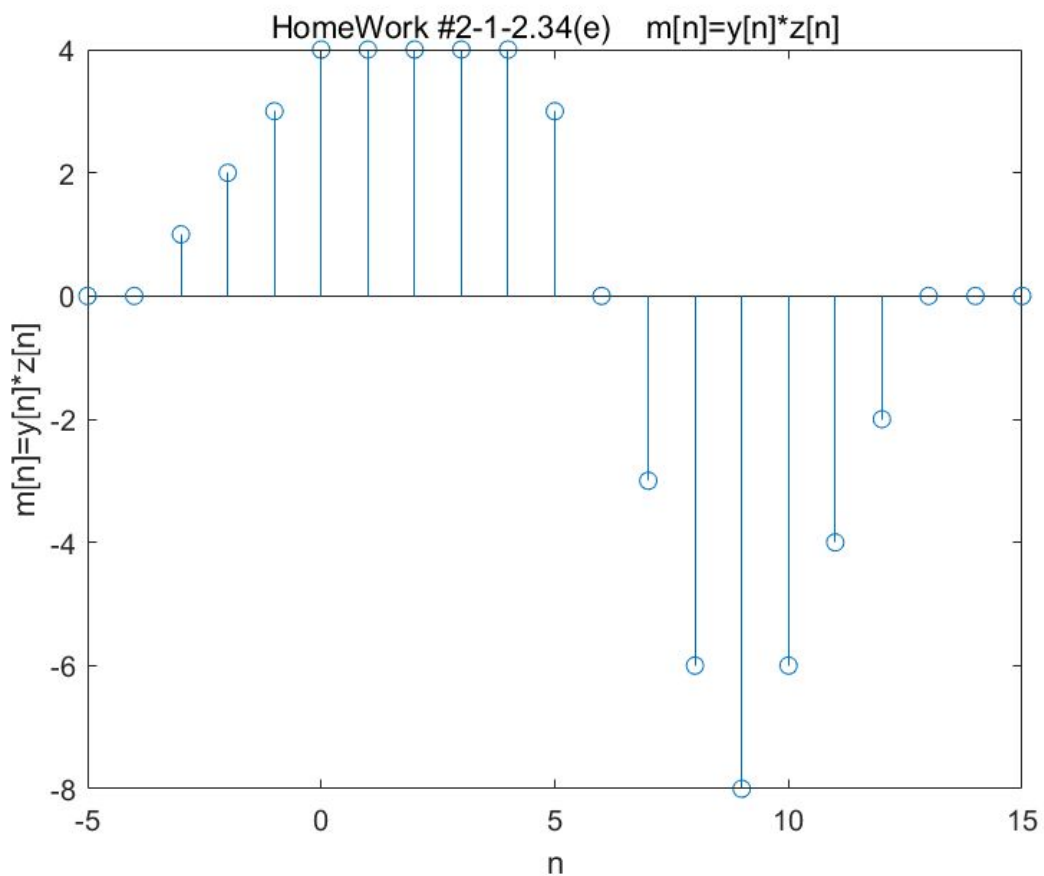
```
stem(t, m)
```

```
xlabel('n')
```

```
ylabel('m[n]=y[n]*z[n]')
```

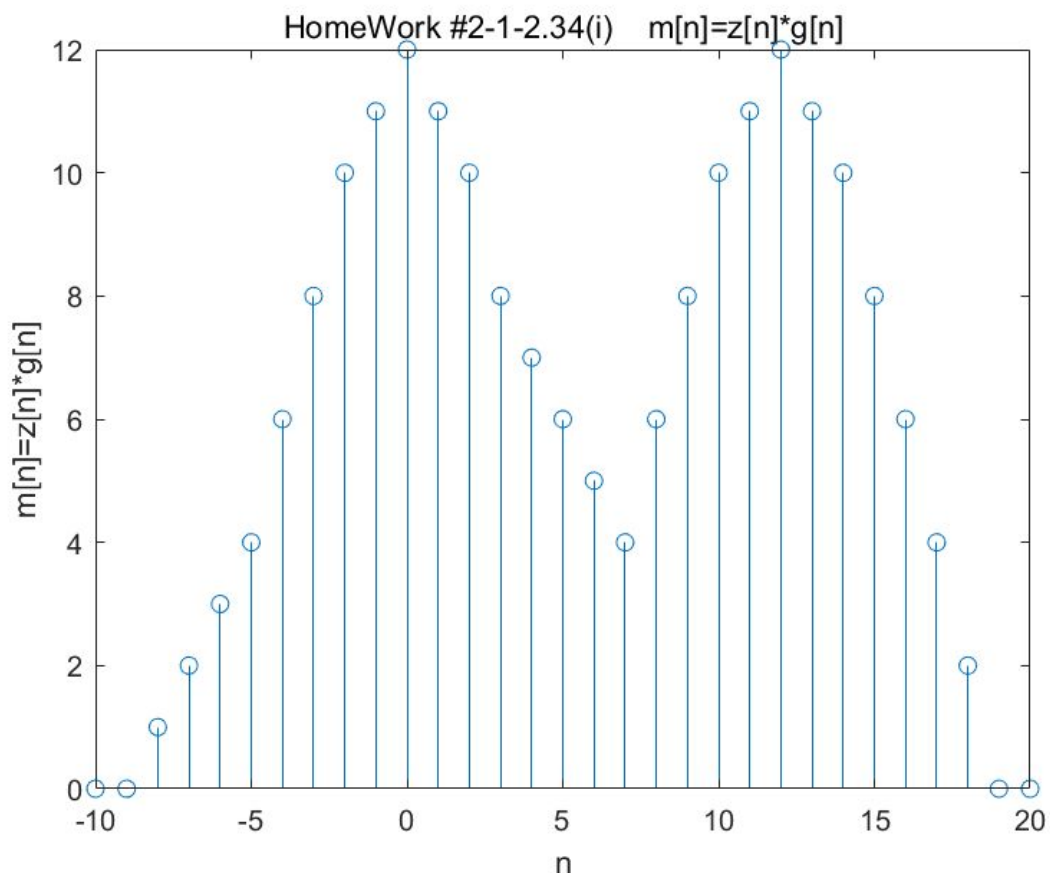
```
title('HomeWork #2-1-2.34(e) m[n]=y[n]*z[n]')
```

```
axis([-5 15 -8 4])
```



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

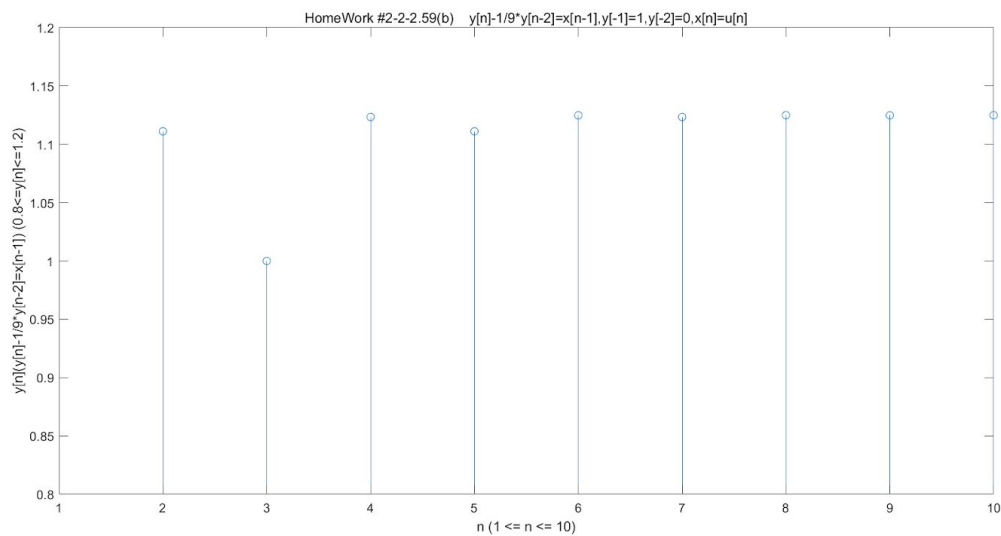
```
----- < CODE > -----  
%% HomeWork #2_1_2.34(i)  
% m[n]=z[n]*g[n]  
clear all  
  
z = [zeros(1,8), ones(1,4), 2*ones(1,5), zeros(1,2)];  
g = [ones(1,7), zeros(1,5), ones(1,7)];  
m = conv(z, g);  
  
t = -16:1:20;  
stem(t, m)  
xlabel('n')  
ylabel('m[n]=z[n]*g[n]')  
title('HomeWork #2-1-2.34(i) 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 오래걸리기에 tool을 이용하는 것이 더욱 효율적이다.

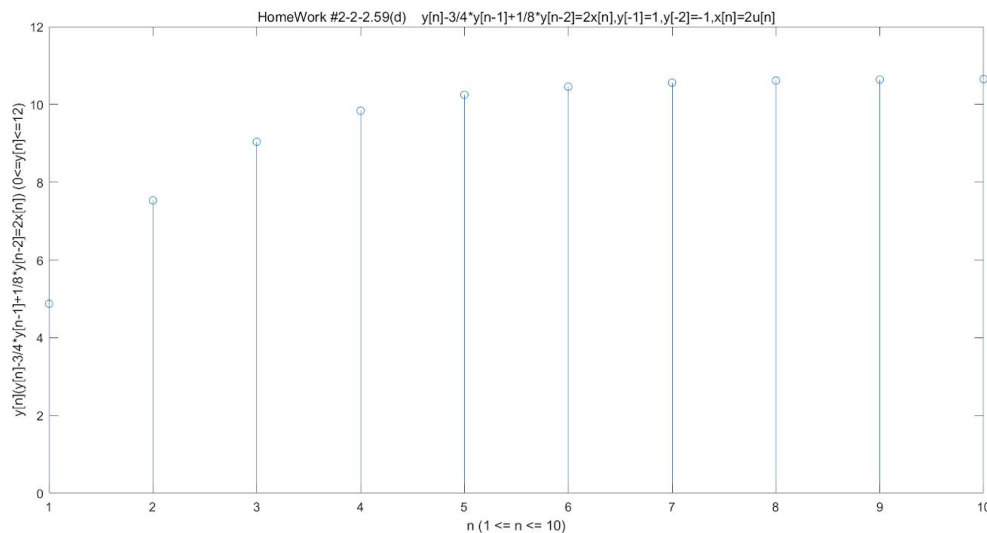
Q 2-1 : Use the MATLAB commands filter and filtic to determine the first 50 output values in Problem 2.59 (b)

```
----- < CODE > -----  
%% HomeWork #2_2_2.59(b)  
%  $y[n] - 1/9*y[n-2] = x[n-1]$ ,  $y[-1]=1$ ,  $y[-2]=0$ ,  $x[n]=u[n]$   
clear all  
  
input = 10; % 입력의 개수(입의 지정 가능)  
b = [0 1 0];  
a = [1 0 -1/9];  
x = ones(1, input);  
i = filtic(b,a,[1 0]);  
y = filter(b,a,x,i);  
  
n = 1:input;  
stem(n, y)  
xlabel('n (1 <= n <= 10)')  
ylabel('y[n](y[n] - 1/9*y[n-2] = x[n-1]) (0.8 <= y[n] <= 1.2)')  
title('HomeWork #2-2-2.59(b) y[n]-1/9*y[n-2]=x[n-1],y[-1]=1,y[-2]=0,x[n]=u[n]')  
axis([1 input 0.8 1.2])  
-----
```



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

```
----- < CODE > -----  
%% HomeWork #2_2_2.59(d)  
%  $y[n] - 3/4*y[n-1] + 1/8*y[n-2] = 2x[n]$ ,  $y[-1]=1$ ,  $y[-2]=-1$ ,  $x[n]=2u[n]$   
clear all  
  
input = 10; % 입력의 개수(임의 지정 가능)  
b = [2 0 0];  
a = [1 -3/4 +1/8];  
x = 2*ones(1, input);  
i = filtic(b,a,[1 -1]);  
y = filter(b,a,x,i);  
  
n = 1:input;  
stem(n, y)  
xlabel('n (1 <= n <= 10)')  
ylabel('y[n](y[n]-3/4*y[n-1]+1/8*y[n-2]=2x[n]) (0<=y[n]<=12)')  
title('HomeWork #2-2-2.59(d)  
y[n]-3/4*y[n-1]+1/8*y[n-2]=2x[n],y[-1]=1,y[-2]=-1,x[n]=2u[n]')  
axis([1 input 0 12])  
-----
```



분석 : 특정 신호들을 **filtic**를 이용하여 **initial condition**을 지정한 후 **filter**를 이용하여 **difference equation**을 구현했다. discrete signal이기에 **stem**을 이용하여 그래프로 나타냈다. **filtic**에서 두개의 행벡터(입력계수벡터, 출력계수벡터)와 초기값($y[-1], y[-2]$ 순)을 인수로 받아 **filter**의 인수에 넘겨줌으로써 difference equation을 구성한다. 입력개수를 의미하는 상수를 바로 넣은 것이 아닌 변수 **input**에 상수를 지정해두어 변수값을 변경하면 **input**에 따른 결과로 바뀌도록 구성하였다. 또한 matlab은 vector index를 구성할때 zero-based가 아닌 시작수를 1에 저장한다.