EE202 MATLAB Exercise 1

20160042 구인용

본 보고서는 EE202 MATLAB 과제 1에 대한 나의 해답을 서술하는 보고서이다. 각 문제에 대한 답의 소스코드, 출력되는 결과값, 그리고 그에 대한 토의를 제시할 것이다.

문제 1 Plotting Graphs

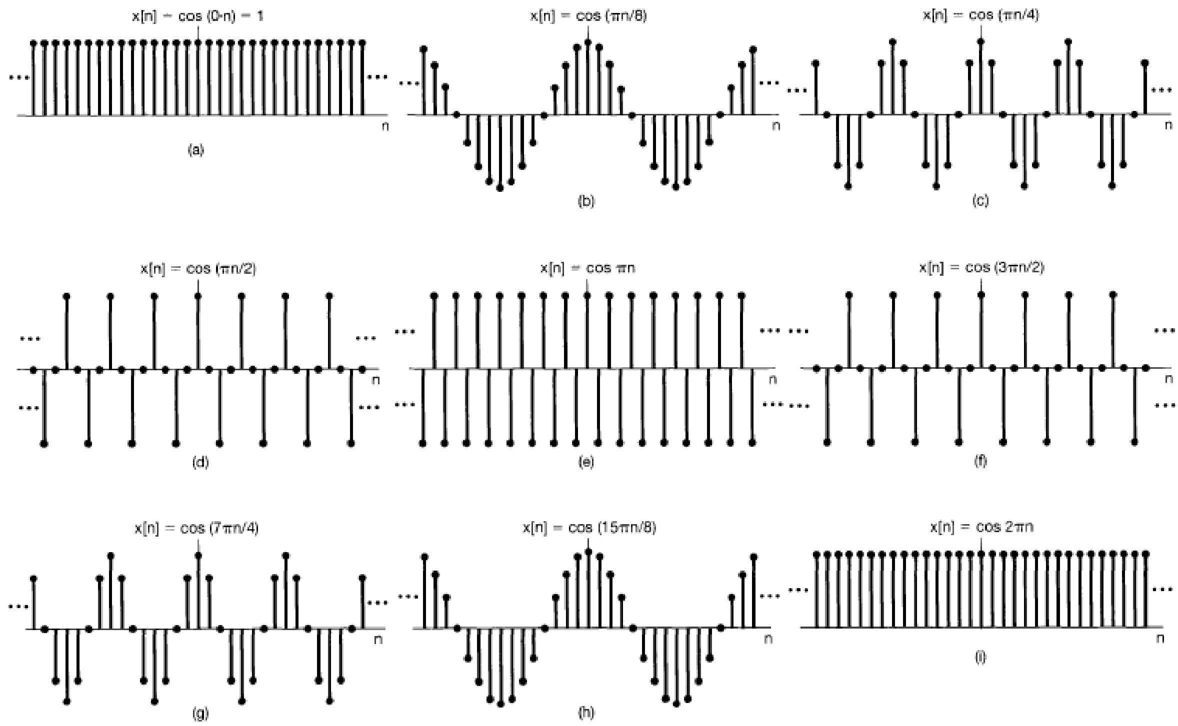

Figure 1.27 Discrete-time sinusoidal sequences for several different frequencies.

그림 1 O&W textbook에 제시된 Figure 1.27

Replicate the Figure 1.27 in O&W textbook (p.27). Plot nine graphs in the figure with the same range of n in ONE figure window. You don't have to replicate the ellipses and order captions e.g. (a), in the figure. (Hint: use *stem* and *subplot* function)

Note that in every figures and graphs, proper title and labels should be displayed. Use *title*, *xlabel*, and *ylabel* function.

제시된 문제에 대한 소스 코드는 다음과 같다.

```
1  %%%%%%%%%%%%%%%%% prob_2_1.m file for MATLAB Exercise #1 %%%%%%%%%%%%%%%%%
2  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3  %%%%%%%%%%%%%%%%%%%%%% FILL THE CODE AND TEST HERE %%%%%%%%%%%%%%%%%%%%%%
4  % (1) Define the signal x[n]
5  
6  clear
7  close all
8  
9  %Set Domain (-D~D)
10 D = 20;
11 n = -D:D;
12 
13 %x[n] is represented in forms x[n]=cos(k*n)
14 %determine 'k's, shown in the textbook
15 k = [0 pi/8 pi/4 pi/2 pi 3*pi/2 7*pi/4 15*pi/8 2*pi];
16 
17 %Now, we will define x[n] to each k
18 %each rows represent the x[n] of k
19 x = zeros(9,length(n)); %Initialize
20 for j=1:9
21     x(j,:) = cos(k(j)*n);
22 end
23 
24 % (2) Plot a graph of x[n]
25 f = figure;
26 f.Position = [100 100 900 600];
27 f.Name = 'Answer for Problem 1';
28 
29 %Title for each graphs
   tvar= str2mat('(0*n)', '(pi*n/8)', '(pi*n/4)', '(pi*n/2)', 'pi*n', '(3*pi*n/2)',
30 '(7*pi*n/4)', '(15*pi*n/8)', '2*pi*n');
31 for j=1:9
32     subplot(3,3,j);
33     h= stem(n,x(j,:));
34     t= sprintf('x[n] = cos %s',tvar(j,:));
35     title(t);
36     xlabel('n');
37     ylabel('x[n]');
38 
39     %adjust graphics
40     h.Color = 'k';
41     h.Marker = '.';
42     xlim([-15 15]);
43     ylim([-1.2,1.2]);
44 end
```

x[n]이 cos(k*n)꼴을 꼴을 갖는다는 것을 확인하여, 9개의 x[n]을 따로 하나씩 정의하기보다 k를 9개의 값을 갖는 벡터로 정의하고 (line 15), k를 변수로 갖는 $x_k[n]$를 for문을 통해 생성하여 (9*n) 행렬의 각 행에 저장하였다 (line 19~22).

마찬가지로, 각 그래프의 제목에서도 'x[n] = cos ~~'이 반복된다는 것을 확인하여, 변하는 부분에 대해서만 tvar이라는 행렬에 저장하였다 (line 30). **str2mat* 함수를 사용하는 것을 유의한다.

Title, x축과 y축을 모두 요구한 함수들을 사용하여 출력하였다.

그 외의 사소한 조작으로는 그래프가 너무 작게 출력되지 않도록 창의 크기를 설정해주었고 (line 26), 책과 유사한 모양이 나올 수 있도록 Color와 Marker, x축의 범위를 조작하였으며, 그래프의 경계에 점들이 놓이지 않도록 y축의 범위를 조정하였다 (line 40~43).

출력 결과는 다음과 같다.

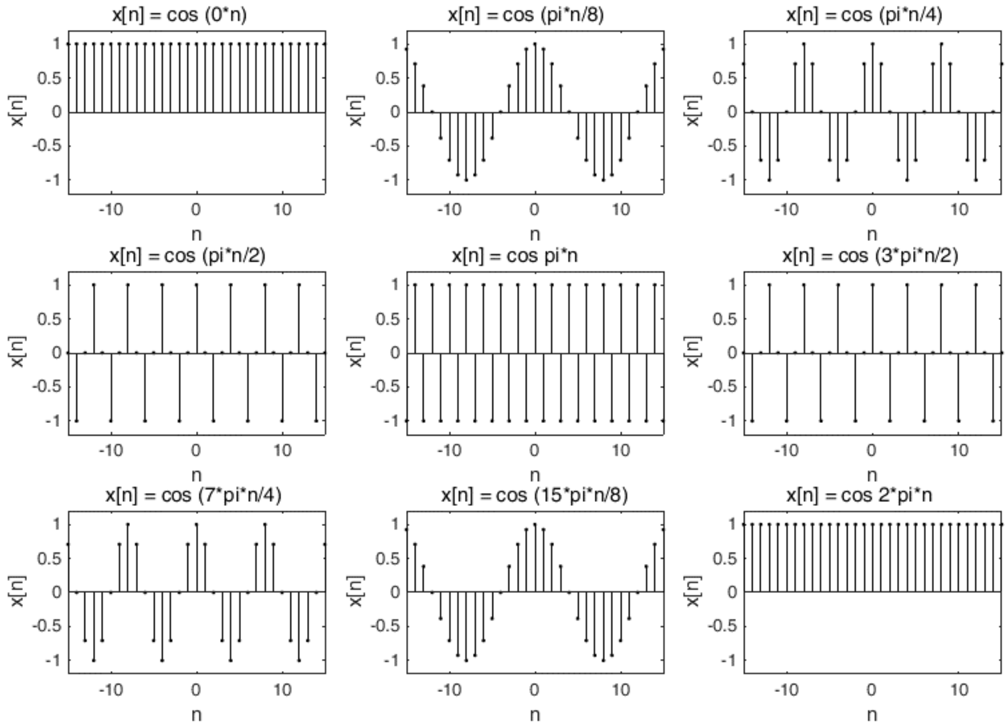

그림 2 Answer for Problem 1

문제에서 요구한 바와 같이, 그림 1을 모사한 그래프를 출력하였다.

문제 2 Convolution of LTI System

Consider a convolution of $x[n]$ and $h[n]$ where

$$x[n] = \left(\frac{1}{2}\right)^n u[n] + 2^n u[-n]$$

$$h[n] = u[n]$$

(a) Plot a graphs of $x[n]$ and $h[n]$ for n = −10, ..., 10.

(b) Compute and plot the graphs of $h[n-k]$ versus k = −10, ..., 10 where n = 0, 1, 2, 3.

(c) Compute the convolution $y[n] = x[n] * h[n]$ by implementing the direct computation of the formula, not the MATLAB-built in *conv* function.

(d) Compute the convolution $y[n]$ using *conv* function in MATLAB, plot the graphs of $y[n]$ computed in (c) and (d), and compare the results in the report.

제시된 문제에 대한 소스 코드는 다음과 같다.

```
%%%%%%%%%%%%%%%%% prob_2_2.m file for MATLAB Exercise #1 %%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%% FILL THE CODE AND TEST HERE %%%%%%%%%%%%%%%%%%%%%%
% Define the signal x[n] and h[n]

clear
close all

%Set domain (-D~D)
D = 10;
n = -D:D;

x = 2.^(-n).*(n>=0) + 2.^n.*(-n>=0);
h = 1*(n>=0);

% (a) Plot a graph of x[n] and h[n]
f1= figure;
f1.Position = [100 100 500 400];
f1.Name= 'Answer for Problem 2-(a)';

% Plot x[n]
subplot(2,1,1);
gx = stem(n,x);
xlabel('n');
ylabel('x[n]');
xlim([-10 10]);
ylim([0 2.2]);
title('x[n] = (1/2)^n * u[n] + 2^n * u[-n]');

%Plot h[n]
subplot(2,1,2);
gh = stem(n,h);
xlabel('n');
ylabel('h[n]');
xlim([-10 10]);
ylim([0 1.2]);
title('h[n] = u[n]');

%Adjust graphics
gx.Color = 'k';
gx.Marker = '.';
gh.Color = 'k';
gh.Marker = '.';

% (b) Compute and plot the graphs of h[n-k] where n=0,1,2,3.
f2= figure;
f2.Position = [600 100 500 600];
f2.Name = 'Answer for Problem 2-(b)';

knum = 10;
k= -knum:knum;

%Plot h[n-k] for n=t
h2=zeros(4,length(k));
for t=[0 1 2 3]
    h2(t+1,:) = (t-k>=0);
    subplot(4,1,t+1);
    gh2 = stem(k,h2(t+1,:));
    xlabel('n');
    ylabel('h[n]');
    xlim([-10 10]);
    ylim([0 1.2]);
    titles = sprintf('h[n-k] for n = %d', t);
    title(titles);
    gh2.Color = 'k';
    gh2.Marker = '.';
end

% (c) Compute the convolution y[n]=x[n]*h[n] by implementing the direct
```

```
70  % computation of the formula
71  y = zeros(1,length(n));
72  for j=1:length(n)
73      hn = (n(j)-k>=0);
74      y(j) = sum(x .* hn);
75  end
76
77  % (d) Compute the convolution y[n]=x[n]*h[n] by conv function and plot the
78  % graphs of both convolution and compare them.
79  f3 = figure;
80  f3.Position = [1100 100 500 400];
81  f3.Name = 'Answer for Problem 2-(d)';
82
83  y2=conv(x,h);
84
85  %Plot y[n] by implementing direct computation
86  subplot(2,1,1);
87  gy = stem(n,y);
88  xlabel('n');
89  ylabel('y[n]');
90  xlim([-10 10]);
91  ylim([0 4.2]);
92  title('y[n] by implementing direct computation');
93
94  %Plot y[n] by conv function
95  subplot(2,1,2);
96  gy2 = stem(n,y2(knum+1:knum+length(n)));
97  xlabel('n');
98  ylabel('y[n]');
99  xlim([-10 10]);
100 ylim([0 4.2]);
101 title('y[n] by conv function');
102
103 %adjust gaphics
104 gy.Color = 'k';
105 gy.Marker = '.';
106 gy2.Color = 'k';
107 gy2.Marker = '.';
```

모든 그래프를 출력할 때 가시성이 좋도록 문제 1과 마찬가지로 창의 크기, 위치, 그래프의 범위를 조절하였고 Color와 Marker를 각각 'k', '.' 으로 설정하였다. 모든 그래프의 제목과 축 이름을 설정하였다.

Unit step function을 MATLAB에서 $(n >= 0)$으로 표현하면 된다는 것을 알 수 있다. Boolean function으로 n이 0 이상일 때 1, 아닐 때 0을 만족하기 때문이다. 이를 이용하여 x, h를 정의해줄 수 있다 (line 13~14).

2-(a)에서 요구한 대로 그래프를 그려주었고, 그 창의 이름을 'Answer for Problem 2-(a)'로 지정하였다.

2-(b) 문제에 대해서는 문제 1에서 사용한 방법과 같이, n=t일 때의 h[n-k] 값을 for문을 통해 그려주었다. 창의 이름은 'Answer for Problem 2-(b)'로 지정하였다.

2-(c)에서 직접 convolution을 계산하였는데, j를 변화시켜주며 y(j)를 계산하였다. 변수 $\mathrm{hn} = u[j-k]$ 을 대응하는 $x[k]$에 곱해서 더해주는 방식이다 i.e, $y[j] = \sum x[k]u[j-k]$ (line 73~74).

2-(d)에서는 *conv* 함수를 이용하여 y2를 계산하였다 (line 83). 길이가 각각 a, b인 벡터를 convolution하면 a+b-1 길이의 값이 나오므로, 우리가 사용한 x, h는 각각 -10~10 범위, 즉 length(x)=length(h)= 21이어서 y2는 41 길이의 벡터로 계산된다. 우리가 원하는 range에서의 y2값은 11번째 값부터 시작하므로, plotting할 때 y2의 범위를 고려해주어야 한다 (line 96).

2-(c)에서 구한 y에 대한 그래프와 함께 y2를 한 창에 출력하였고 창의 이름은 'Answer for Problem 2-(d)'로 지정하였다.

출력 결과는 다음과 같다.

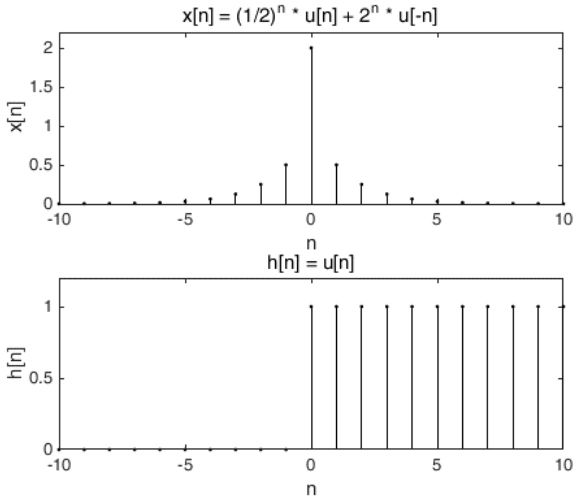

그림 3 Answer for Problem 2-(a)

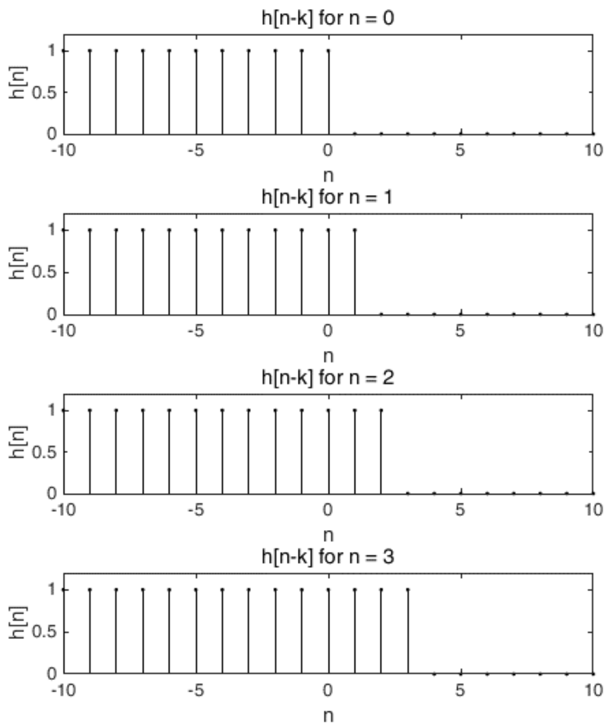

그림 4 Answer for Problem 2-(b)

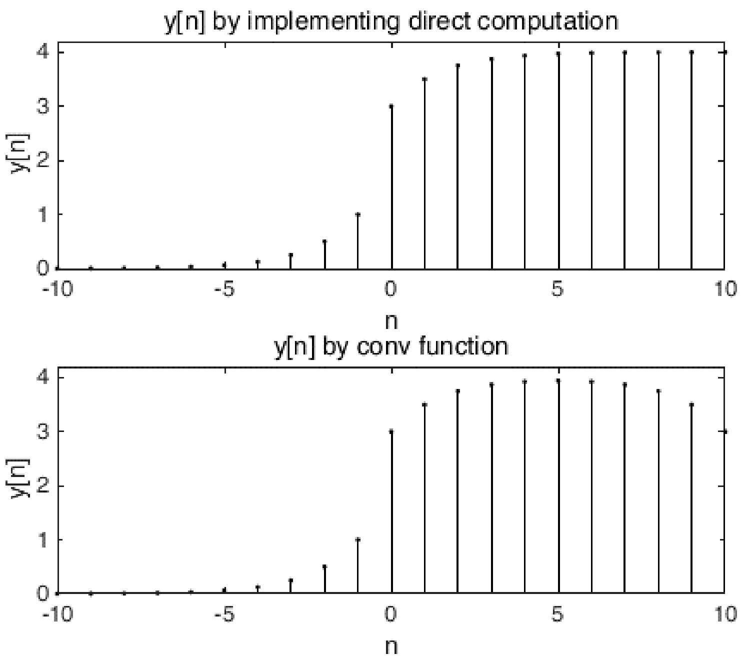

그림 5 Answer for Problem 2-(d)

그림 5를 보면 direct computation을 이용하여 구한 y[n]과 *conv* 함수를 이용한 y[n] (편의상 y2[n])의 모양이 다름을 알 수 있다. 그 이유는 x[k]에 곱해지는 h[n-k]의 범위에 의해 결정된다.

그림 2에서 볼 수 있듯이, direct computation에서 우리가 지정해주는 h[n-k]의 값은 n=t 일 때 t 이하의 값에서 모두 1이 된다. 그렇기 때문에 $y[n] = \sum_{k=-10}^{10} x[k]u[n-k] = \sum_{k=-10}^{n} x[k]$ 가 될 수 있다. (n이 -10~10 범위이므로).

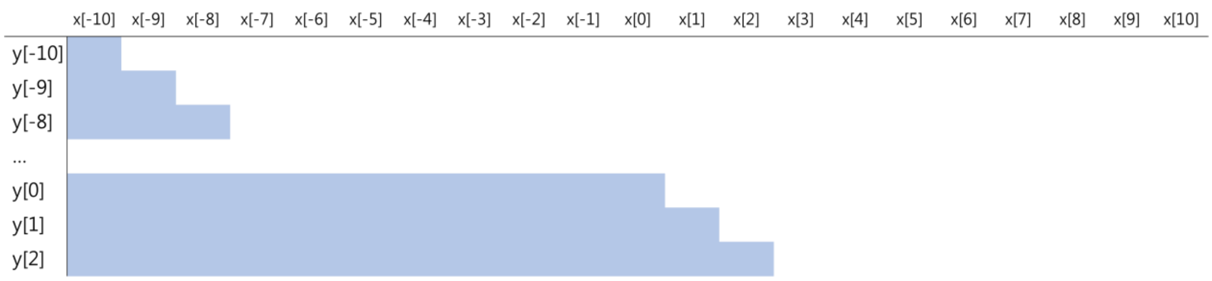

그림 6 direct computation에서 실제로 더해지는 x[k]의 범위

그러나 *conv* 함수를 이용할 경우 h[n]의 벡터가 초기에 설정해준 0~10의 범위에서 1을 갖는 크기 21의 벡터로 고정되어버린다. 즉, h[n−k]의 값은 n=t일 때 오직 t−10~t만 1이 된다. 따라서 $y[n] = \sum_{k=-10}^{10} x[k]h[n-k] = \sum_{k=n-10}^{n} x[k]$ 을 만족하게 된다.

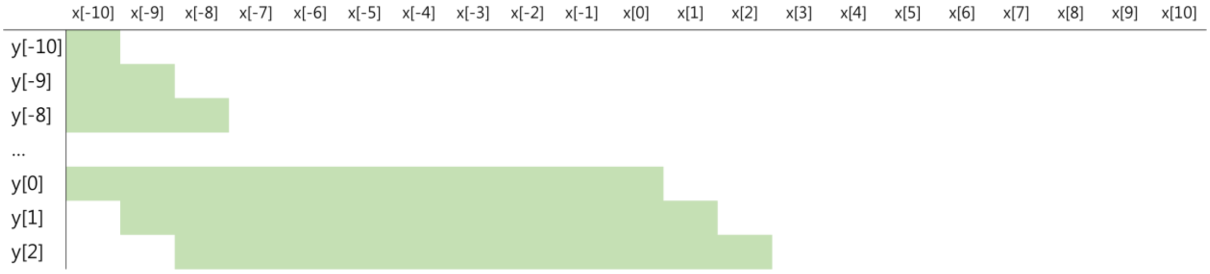

그림 7 *conv* 함수에서 실제로 더해지는 x[k]의 범위

따라서 n=0일 때까지 y[n]과 y2[n]은 동일하지만, 그 이후로 값이 차이가 나며, 그 차이는 $\sum_{-10}^{n-11} x[k]$, 즉 y[n−11]의 값과 같게 된다.

실제로 y[n]과 y2[n]의 값을 비교해보면 다음과 같다.

```
>> y

y =

  열 1 ~ 10

    0.0010    0.0029    0.0068    0.0146    0.0303    0.0615    0.1240    0.2490    0.4990    0.9990

  열 11 ~ 20

    2.9990    3.4990    3.7490    3.8740    3.9365    3.9678    3.9834    3.9912    3.9951    3.9971

  열 21

    3.9980

>> y2(11:31)

ans =

  열 1 ~ 10

    0.0010    0.0029    0.0068    0.0146    0.0303    0.0615    0.1240    0.2490    0.4990    0.9990

  열 11 ~ 20

    2.9990    3.4980    3.7461    3.8672    3.9219    3.9375    3.9219    3.8672    3.7461    3.4980

  열 21

    2.9990
```

그림 8 y[n]과 y2[n]의 비교

```
>> y-y2(11:31)

ans =

  열 1 ~ 10

        0         0         0         0         0         0         0         0         0         0

  열 11 ~ 20

        0    0.0010    0.0029    0.0068    0.0146    0.0303    0.0615    0.1240    0.2490    0.4990

  열 21

    0.9990
```

그림 9 y[n]과 y2[n]의 비교 2

처음 11개의 값은 같다가, 그 후로는 y[n-11]만큼의 차이를 갖는다는 것을 실제로 확인할 수 있었다.

비록 x[n]이 -10~10 영역에서 밖에 정의되지 않아 둘 다 정확한 값이라고 볼 수는 없지만, 실제 convolution에 더 근접하는 것은 h[n-k]를 더 근접하게 표현한 (c), direct computation임을 생각해 볼 수 있다