

BLM2041 - Bilgisayar Mühendisleri için Sinyaller ve Sistemler

Ödev 1

AYRIK ZAMANLI SİSTEMLERDE KONVOLÜSYON İŞLEMİ

Emir OĞUZ

20011059

```

close all
clearvars

% getting inputs from user
x = input('\nEnter x: ');
Lx = input('Lower limit of x: ');
Ux = input('Upper limit of x: ');
part_x = Lx:Ux;

y = input('\nEnter y: ');
Ly = input('Lower limit of y: ');
Uy = input('Upper limit of y: ');
part_y = Ly:Uy;

part_res = Lx + Ly : Ux + Uy;

n = length(x);
m = length(y);

result1 = myConv(x, n, y, m);
result2 = conv(x, y);

% vector results (myConv & conv)
fprintf('\nVector Representations:\n');
fprintf('x[n]: ');
disp(x);
fprintf('y[m]: ');
disp(y);
fprintf('myConv: ');
disp(result1);
fprintf('conv(): ');
disp(result2);

input('Press ''Enter'' to continue...\n', 's');

% graphical results (x[n], y[m], myConv, conv)
fprintf('Graphical Representations:\n')

figure;

subplot(4,1,1); stem(part_x, x, '-b^');
xlabel('n'); ylabel('x[n]'); grid on;
title('x[n]');

subplot(4,1,2); stem(part_y, y, '-ms');
xlabel('m'); ylabel('y[m]'); grid on;
title('y[m]');

subplot(4,1,3); stem(part_res, result1, '-ro');
ylabel('myResult[k]'); xlabel('k'); grid on;
title('Convolution Result with myConv() Function');

subplot(4,1,4); stem(part_res, result2, '-ro');
ylabel('convResult[k]'); xlabel('k'); grid on;
title('Convolution Result with conv() Function');

pause(3);
input('\nPlease press ''Enter'' for the first audio recording...\n', 's');

% audio recording (5 seconds)
recObj = audiorecorder; %% kayit baslatma nesnesi
fprintf('RECORD 1 (5 Seconds)\n');
fprintf('Start speaking...\n'); %% ekrana mesaj
recordblocking(recObj, 5); %% kayit islemi
fprintf('End of recording...\n'); %% ekrana mesaj
x1 = getaudiodata(recObj); %% kaydedilen sesi 'x' degiskenine saklama

sound(x1);
fprintf('1. The first recording is currently being played... (x1)\n');
pause(3);
input('\nPlease press ''Enter'' for the second audio recording...\n', 's');

% audio recording (10 seconds)
recObj = audiorecorder; %% kayit baslatma nesnesi
fprintf('RECORD 2 (10 Seconds)\n');
fprintf('Start speaking...\n'); %% ekrana mesaj
recordblocking(recObj, 10); %% kayit islemi
fprintf('End of recording...\n'); %% ekrana mesaj
x2 = getaudiodata(recObj); %% kaydedilen sesi 'x' degiskenine saklama

sound(x2);
fprintf('2. The second recording is currently being played... (x2)\n');
pause(3);
input('\nPress ''Enter'' to continue...\n', 's');

exit = 0;

while exit == 0

```

```

M = input('Value of M: ');

y1_system = system(M);
myConv_y1 = myConv(x1, length(x1), y1_system, length(y1_system));
conv_y1 = conv(x1, y1_system);

y2_system = system(M);
myConv_y2 = myConv(x2, length(x2), y2_system, length(y2_system));
conv_y2 = conv(x2, y2_system);

sound(myConv_y1);
fprintf('\n1.1. The first output is currently being played... (myConv_y1)');
pause(3);
input('\nPress ''Enter'' to continue...', 's');

sound(conv_y1);
fprintf('1.2. The first output is currently being played... (conv_y1)');
pause(3);
input('\nPress ''Enter'' to continue...', 's');

sound(myConv_y2);
fprintf('\n2.1. The second output is currently being played... (myConv_y2)');
pause(3);
input('\nPress ''Enter'' to continue...', 's');

sound(conv_y2);
fprintf('2.2. The second output is currently being played... (conv_y2)');
pause(3);
input('\nPress ''Enter'' to continue...\n', 's');

% graphical results of audio recordings
figure;

subplot(4,1,1); plot(myConv_y1);
xlabel('n'); ylabel('my Y1[n]'); grid on;
title('y1[n] Graph with myConv() Function');

subplot(4,1,2); plot(conv_y1);
xlabel('n'); ylabel('Y1[n]'); grid on;
title('y1[n] Graph with conv() Function');

subplot(4,1,3); plot(myConv_y2);
xlabel('n'); ylabel('my Y2[n]'); grid on;
title('y2[n] Graph with myConv() Function');

subplot(4,1,4); plot(conv_y2);
xlabel('n'); ylabel('Y2[n]'); grid on;
title('y2[n] Graph with conv() Function');

answer = input('Continue: 1\nExit: 0\nChoice: ');

if answer == 0
    exit = 1;
    fprintf('\nExiting the program...\n\n');
elseif answer == 1
    fprintf('\nThe program will be restarted...\n');
    input('Press ''Enter'' to continue...\n', 's');
else
    exit = 1;
    fprintf("\nYou entered an incorrect input value...");
    fprintf('\nExiting the program...\n\n');
end
end

% convolution function
function result = myConv(x, n, y, m)
    result(n + m - 1) = 0;
    for i = 1:n
        for j = 1:m
            result(i + j - 1) = result(i + j - 1) + x(i) * y(j);
        end
    end
end

% system function
function Y = system(M)
    Y = zeros(1, M * 400);
    Y(1) = 1;
    for i = 1:M
        Y(1 + (400 * i)) = 0.8 * i;
    end
end
end

```

1. myConv() ve conv() Fonksiyonları ile Hesaplanan Konvolüsyon Toplamının Vektörel ve Grafikselle Gösterimleri

- Kullanıcıdan girdi değeri olarak $x[n]$ sinyalinin değeri ve sinyalin başladığı alt limit ile üst limit değeri alınır. Kullanıcı, sinyalin değeri için girdiyi vektör halinde girmelidir. (Ödev belgesindeki değeri kullanılmıştır)

```
Enter x: [1 2 3]
Lower limit of x: -1
Upper limit of x: 1
```

- Kullanıcıdan girdi değeri olarak $y[m]$ sinyalinin değeri ve sinyalin başladığı alt limit ile üst limit değeri alınır. Kullanıcı, sinyalin değeri için girdiyi vektör halinde girmelidir. (Ödev belgesindeki değeri kullanılmıştır.)

```
Enter y: [1 2 3]
Lower limit of y: 0
Upper limit of y: 2
```

- Hazırladığım myConv() fonksiyonu ve hazır bulunan conv() fonksiyonu, kullanıcıdan alınan sinyal değeri için konvolüsyon toplamını hesaplar. İki fonksiyonun konvolüsyon toplamı sonuçlarının ilk olarak vektörel gösterimi, ardından grafikselle gösterimi çıktı olarak yazdırılır.

Vector Representations:

x[n]: 1 2 3

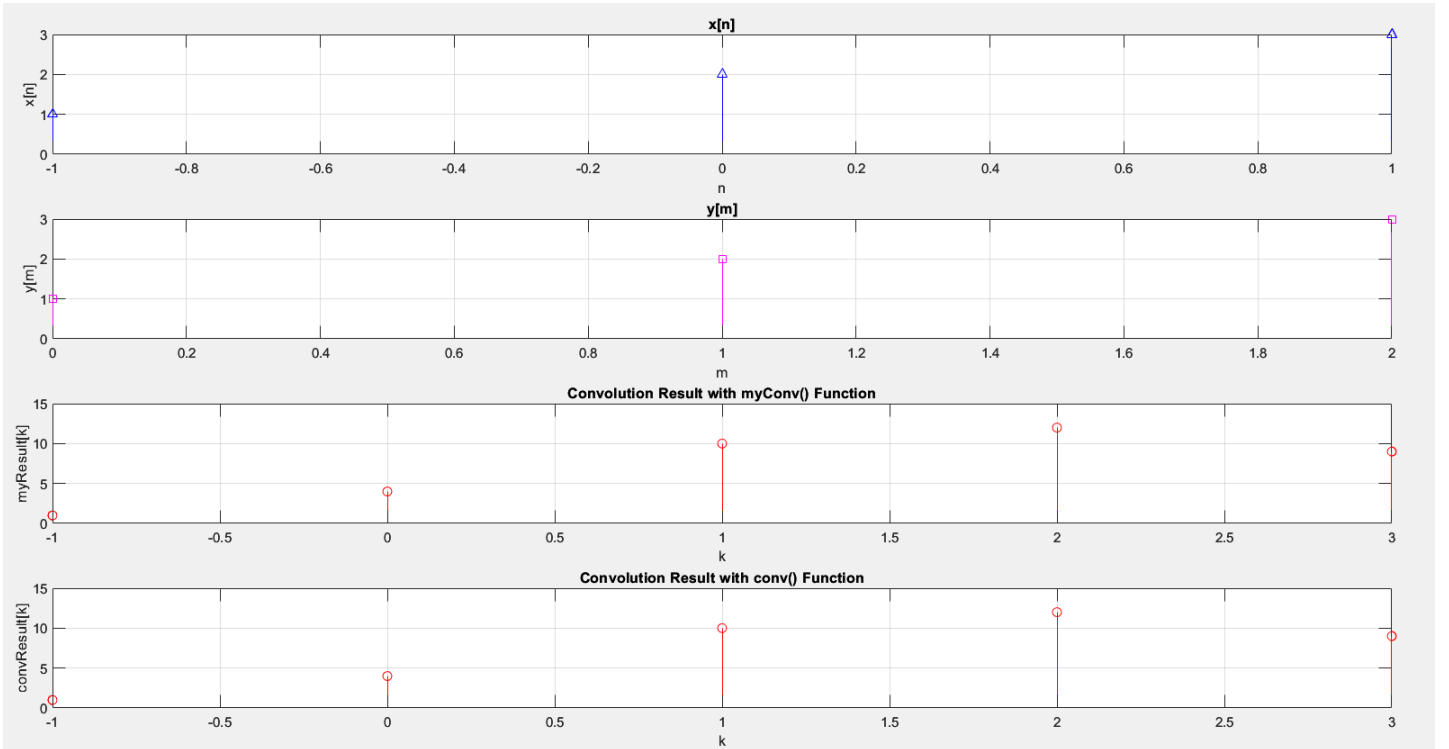
y[m]: 1 2 3

myConv: 1 4 10 12 9

conv(): 1 4 10 12 9

Press 'Enter' to continue...

Graphic Representations:



- Vektörel ve grafikselle gösterimlerden görüldüğü üzere myConv() ve conv() fonksiyonlarının ürettiği sonuçlar aynıdır.

2. Ses Kaydetme, Bu Sesleri Verilen Sisteme Girdi Olarak Ayrı Ayrı Uygulama ve Sonuçları Yazdırma İşlemleri

- Beş saniyelik ses kaydetme işlemi yapılır ve kaydedilen ses, x1 değişkeninde saklanır. Test için x1 değişkeninde saklanan ses, sound() fonksiyonu kullanılarak dinlenir.

```
Please click 'Enter' for the first audio recording...  
RECORD 1 (5 Seconds)  
Start speaking...  
End of recording...  
1. Listening the first record... (x1)
```
- On saniyelik ses kaydetme işlemi yapılır ve kaydedilen ses x2, değişkeninde saklanır. Test için x2 değişkeninde saklanan ses, sound() fonksiyonu kullanılarak dinlenir.

```
Please click 'Enter' for the second audio recording...  
RECORD 2 (10 Seconds)  
Start speaking.  
End of recording.  
2. Listening the second record... (x2)  
  
Press 'Enter' to continue...
```
- Ödevde verilen sistem için yazılmış fonksiyon içerisindeki M parametresi için [2, 3, 4] değerleri kullanılır. İlk girilen ses dosyaları (x1 ve x2) ve sistem fonksiyonunun ürettiği sonuçlar (y1_system ve y2_system) için myConv() ve conv() fonksiyonları ile konvolüsyon toplamı yapılır.

$$y[n] = x[n] + \sum_{k=1}^M A.k.x[n - 400.k]$$

- Farklı M değerleri kullanılarak gerçekleştirilen myConv() ve conv() fonksiyonları ile oluşturulan yeni sinyaller, sound() fonksiyonu kullanılarak dinlenir ve grafikleri çıktı olarak gösterilir.
- Farklı M değerleri için oluşturulan sinyallerin değer grafiklerinde farklılıklar gözükür.
- M değeri arttıkça konvolüsyon toplamı ile oluşan yeni sinyaldeki genlik seviyesi, eski sinyale göre daha yüksek olur. Ses düzeyi artar ve değerler daha sık tekrarlanır. Ses daha çok yankılanıyormuş gibi algılanabilir, sesteki kopukluk düzeyi artar, kalite bozulabilir.

➤ M = 2

Value of M: 2

1.1. Listening the first output (myConv_y1)...

Press 'Enter' to continue...

1.2. Listening the first output (conv_y1)...

Press 'Enter' to continue...

2.1. Listening the second output (myConv_y2)...

Press 'Enter' to continue...

2.2. Listening the second output (conv_y2)...

Press 'Enter' to continue...

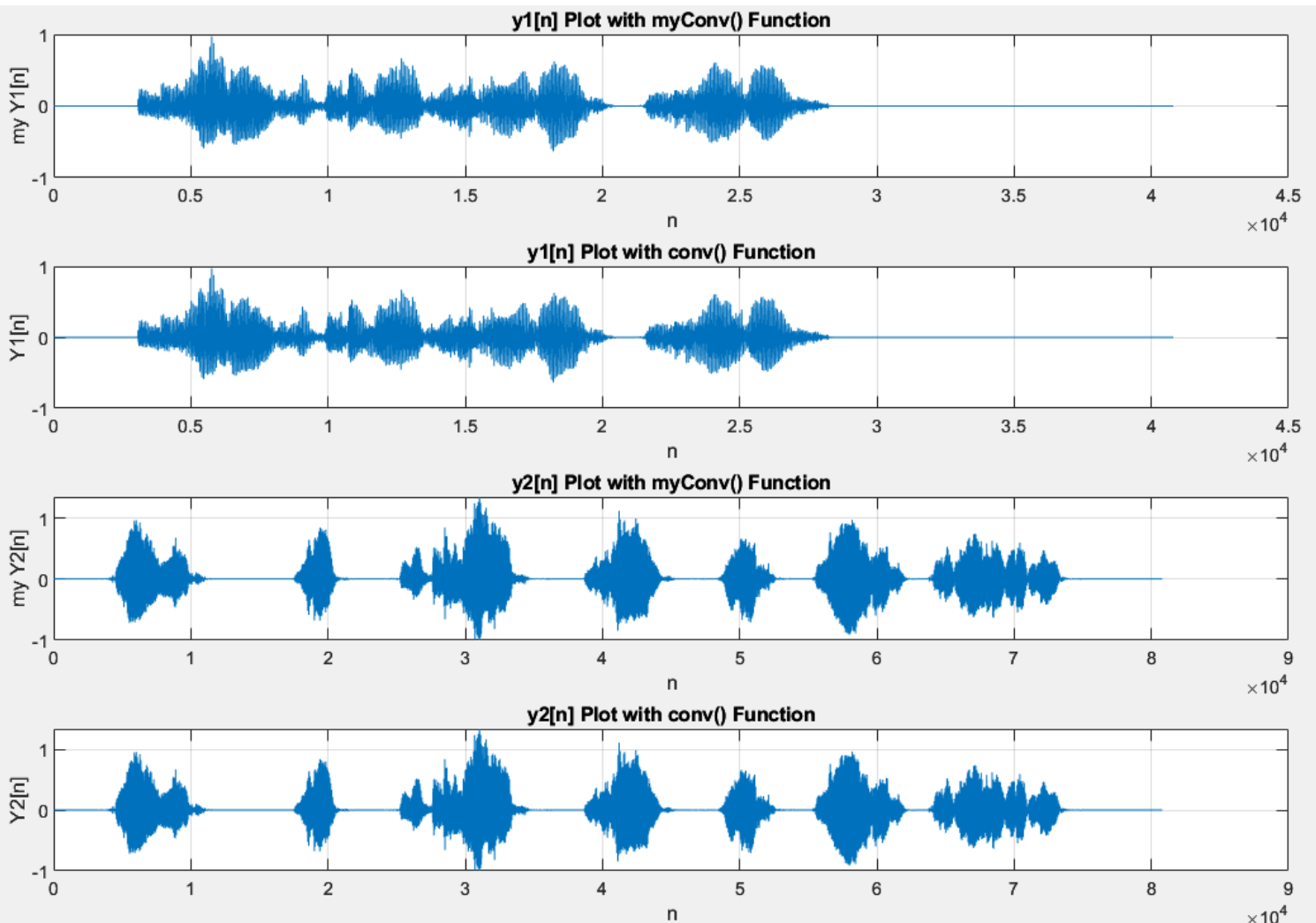
Continue: 1

Exit: 0

Choice: 1

The program will restart...

Press 'Enter' to continue...



➤ M = 3

Value of M: 3

1.1. Listening the first output (myConv_y1)...

Press 'Enter' to continue...

1.2. Listening the first output (conv_y1)...

Press 'Enter' to continue...

2.1. Listening the second output (myConv_y2)...

Press 'Enter' to continue...

2.2. Listening the second output (conv_y2)...

Press 'Enter' to continue...

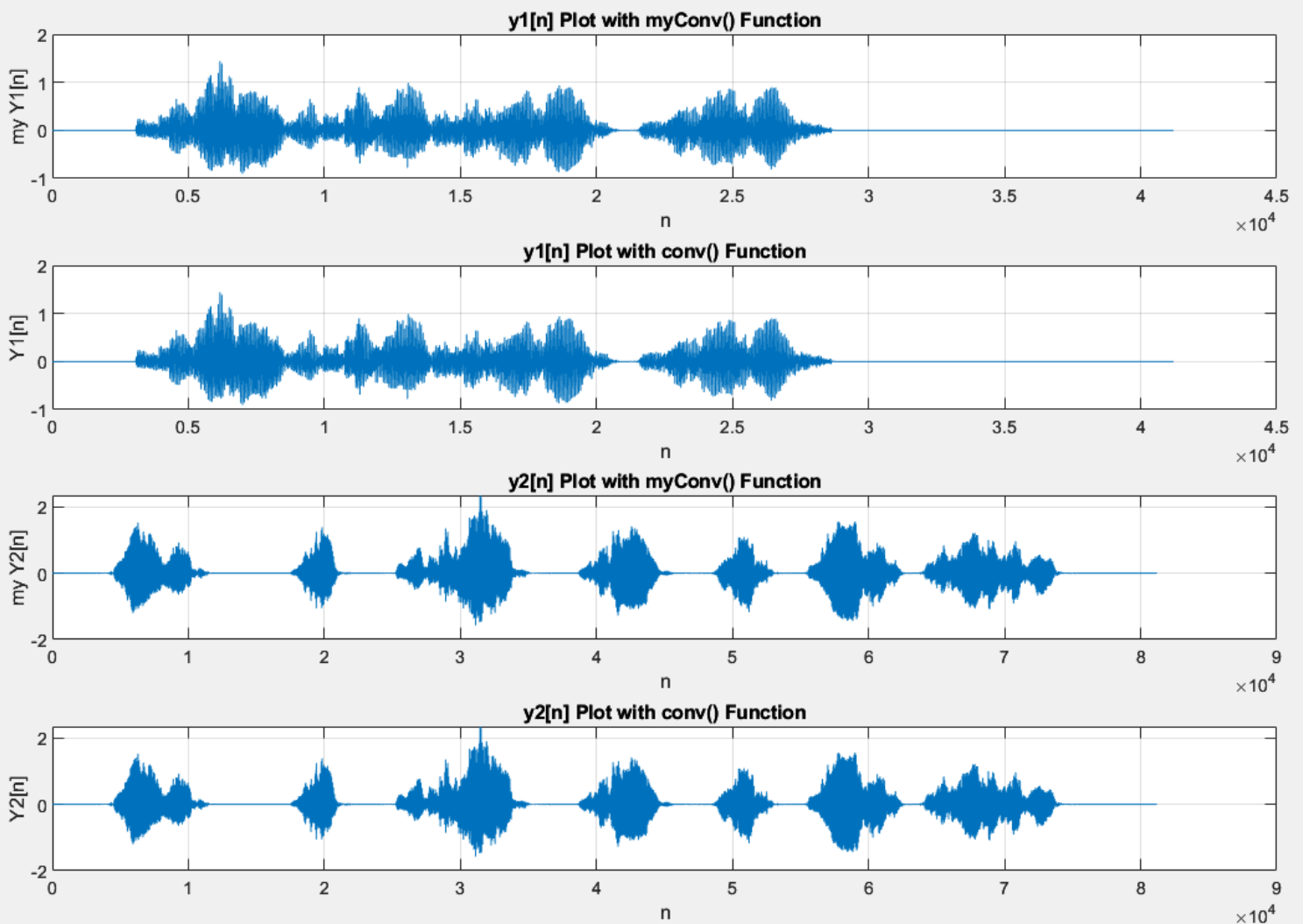
Continue: 1

Exit: 0

Choice: 1

The program will restart...

Press 'Enter' to continue...



➤ M = 4

Value of M: 4

1.1. Listening the first output (myConv_y1)...

Press 'Enter' to continue...

1.2. Listening the first output (conv_y1)...

Press 'Enter' to continue...

2.1. Listening the second output (myConv_y2)...

Press 'Enter' to continue...

2.2. Listening the second output (conv_y2)...

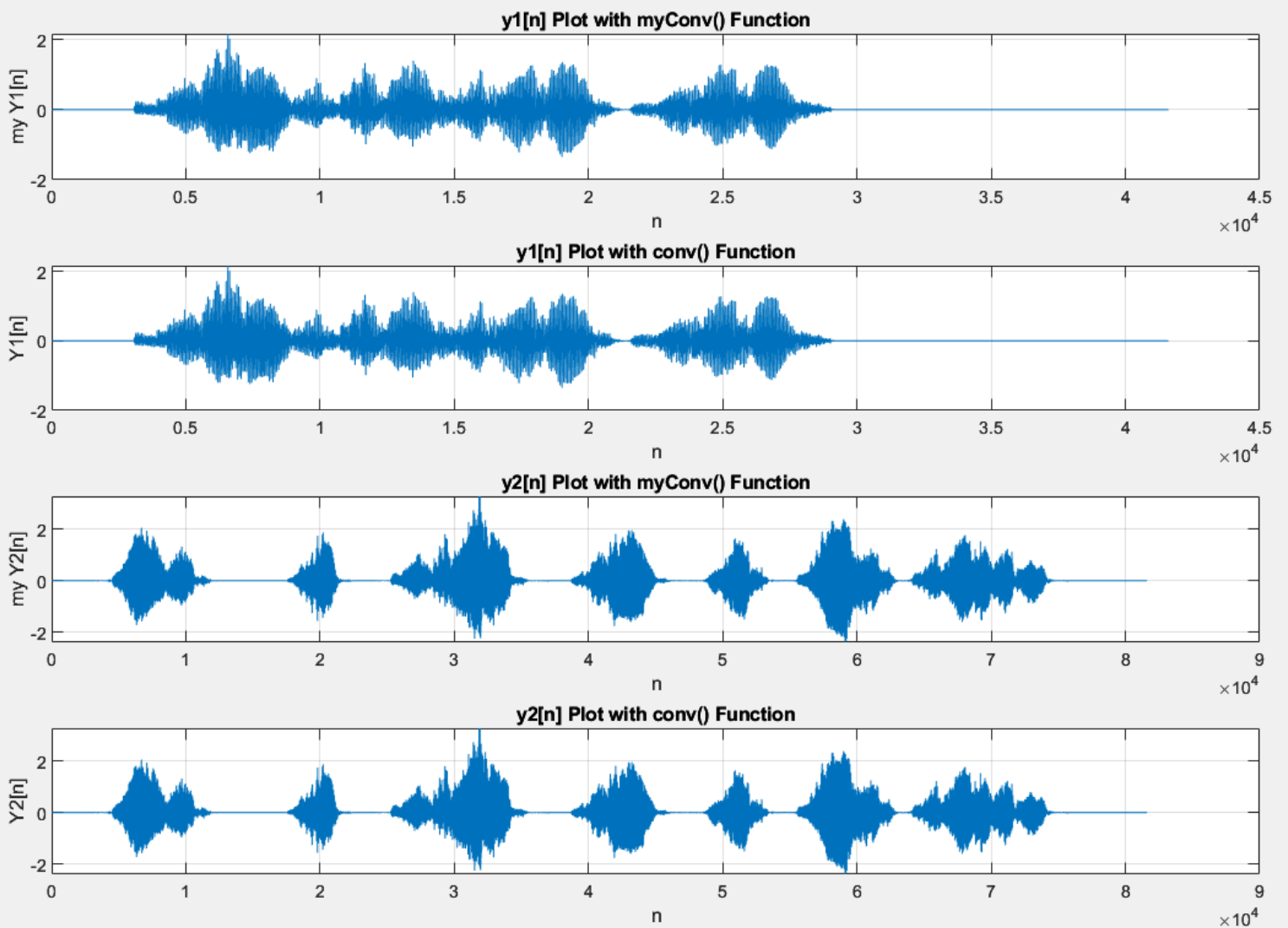
Press 'Enter' to continue...

Continue: 1

Exit: 0

Choice: 0

Exiting the program...



RAPOR SONU