Yıldız Teknik Üniversitesi

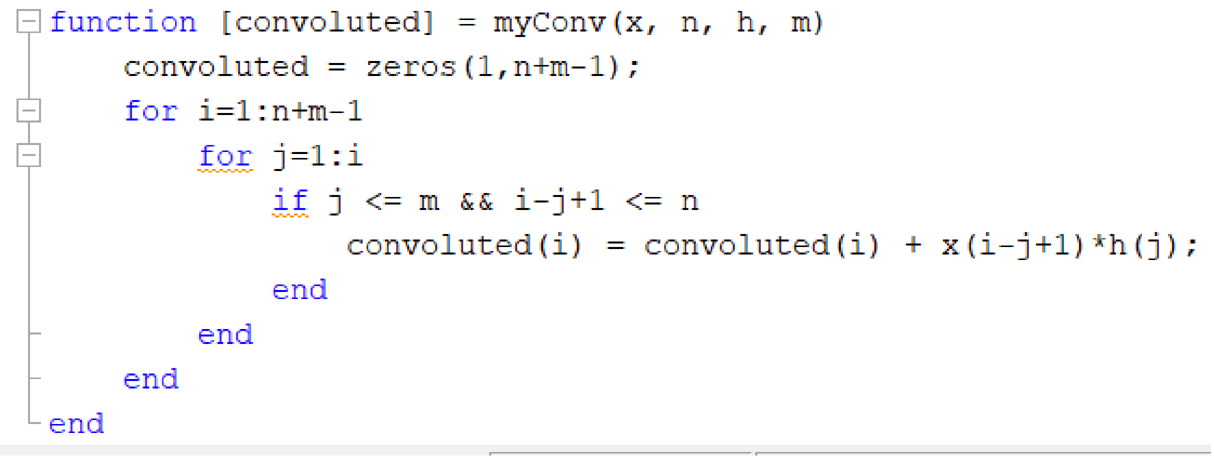
Bilgisayar Mühendisliği Bölümü

BLM2041 – Sinyaller ve Sistemler 1. Ödev

İbrahim Alperen Kürüm – 20011052 – Grup 1

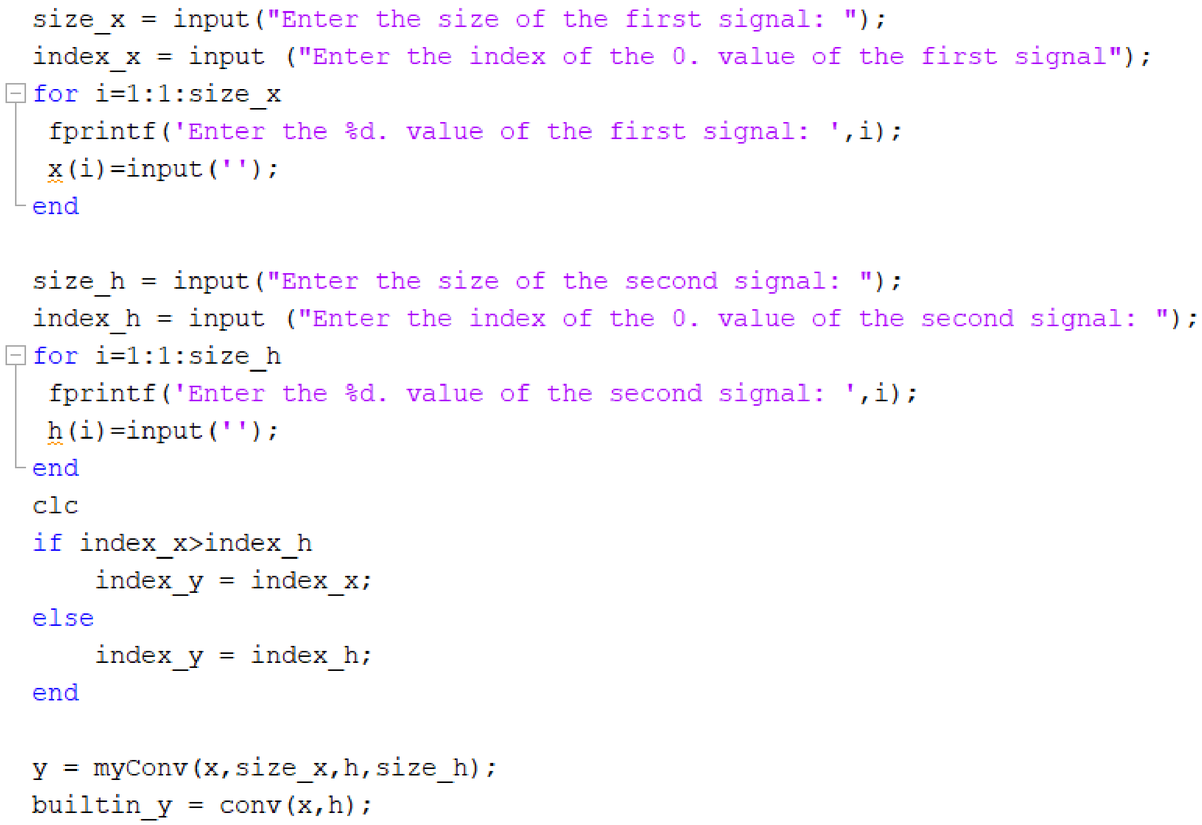
23.11.2022

Konvolüsyon Fonksiyonum



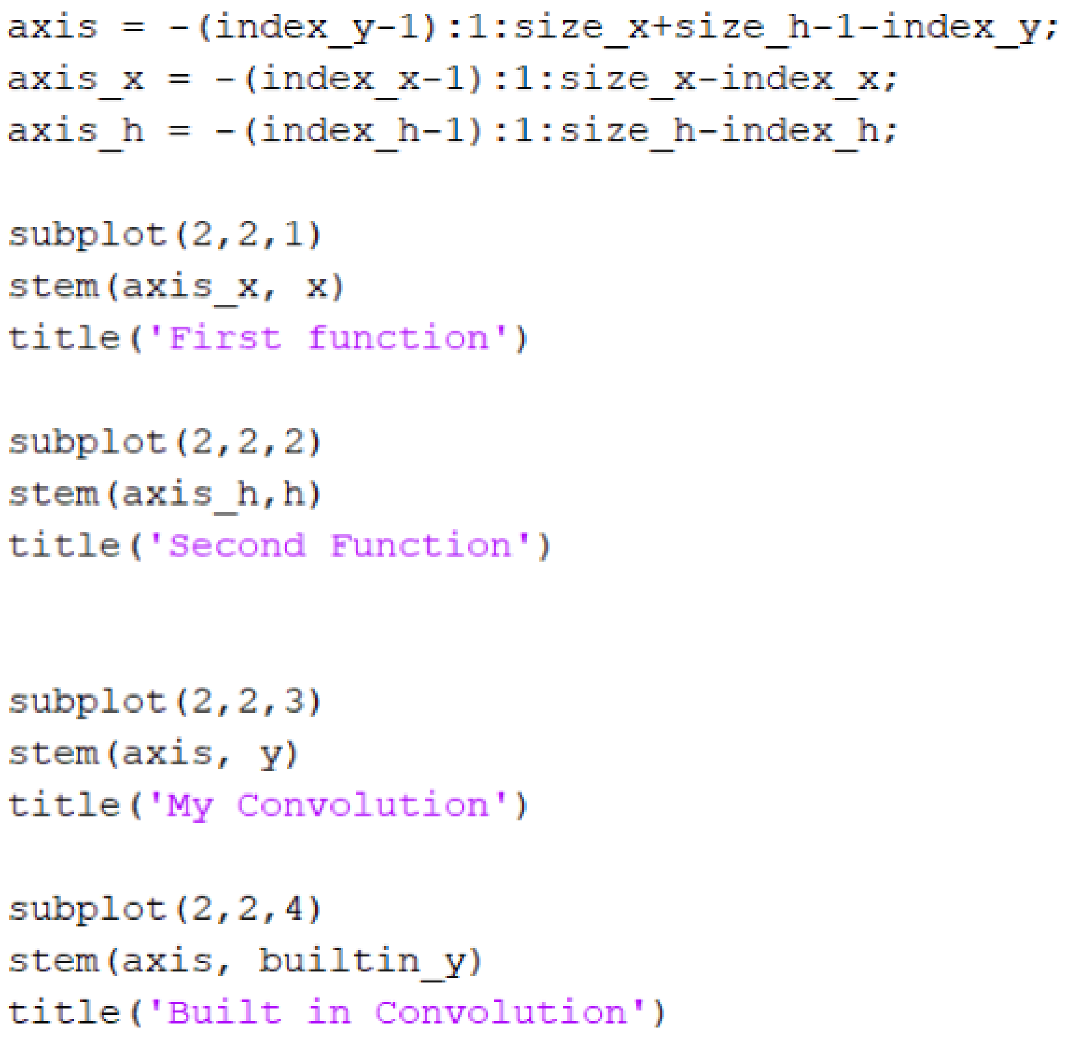
```
function [convoluted] = myConv(x, n, h, m)
    convoluted = zeros(1,n+m-1);
    for i=1:n+m-1
        for j=1:i
            if j <= m && i-j+1 <= n
                convoluted(i) = convoluted(i) + x(i-j+1)*h(j);
            end
        end
    end
end
```

Sinyallere Konvolüsyon Yapılaması

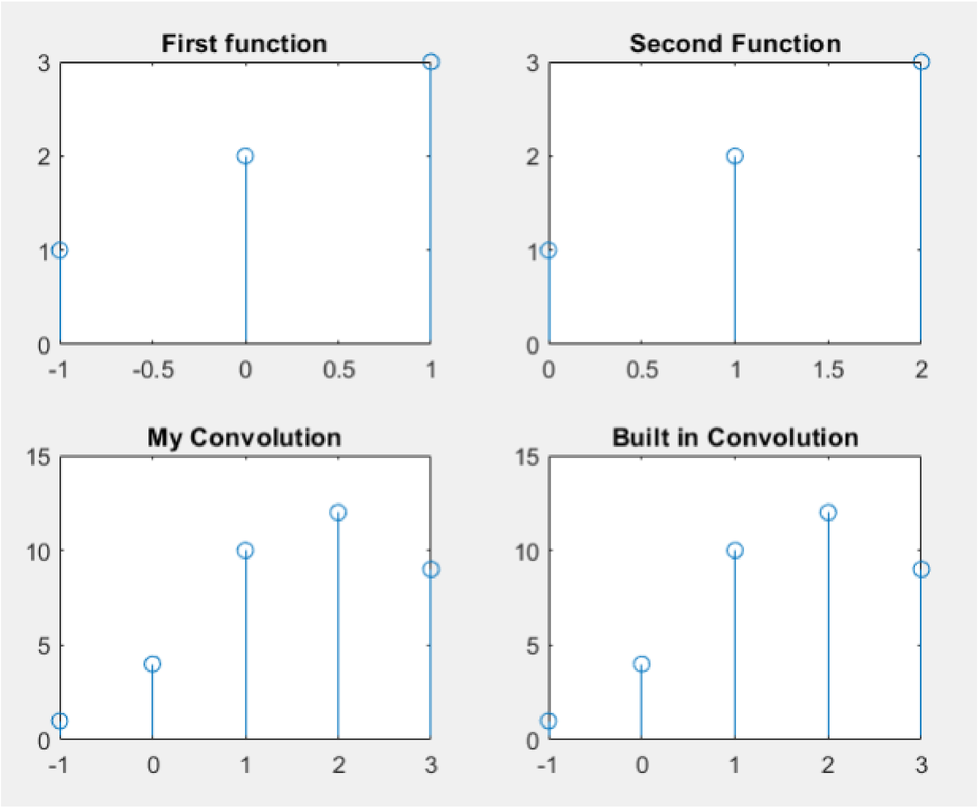


```
size_x = input("Enter the size of the first signal: ");
index_x = input ("Enter the index of the 0. value of the first signal");
for i=1:1:size_x
 fprintf('Enter the %d. value of the first signal: ',i);
 x(i)=input('');
end

size_h = input("Enter the size of the second signal: ");
index_h = input ("Enter the index of the 0. value of the second signal: ");
for i=1:1:size_h
 fprintf('Enter the %d. value of the second signal: ',i);
 h(i)=input('');
end
clc
if index_x>index_h
    index_y = index_x;
else
    index_y = index_h;
end

y = myConv(x,size_x,h,size_h);
builtin_y = conv(x,h);
```

Grafik Gösteriminin oluşturulması



```
axis = -(index_y-1):1:size_x+size_h-1-index_y;
axis_x = -(index_x-1):1:size_x-index_x;
axis_h = -(index_h-1):1:size_h-index_h;

subplot(2,2,1)
stem(axis_x, x)
title('First function')

subplot(2,2,2)
stem(axis_h,h)
title('Second Function')


subplot(2,2,3)
stem(axis, y)
title('My Convolution')

subplot(2,2,4)
stem(axis, builtin_y)
title('Built in Convolution')
```



First function

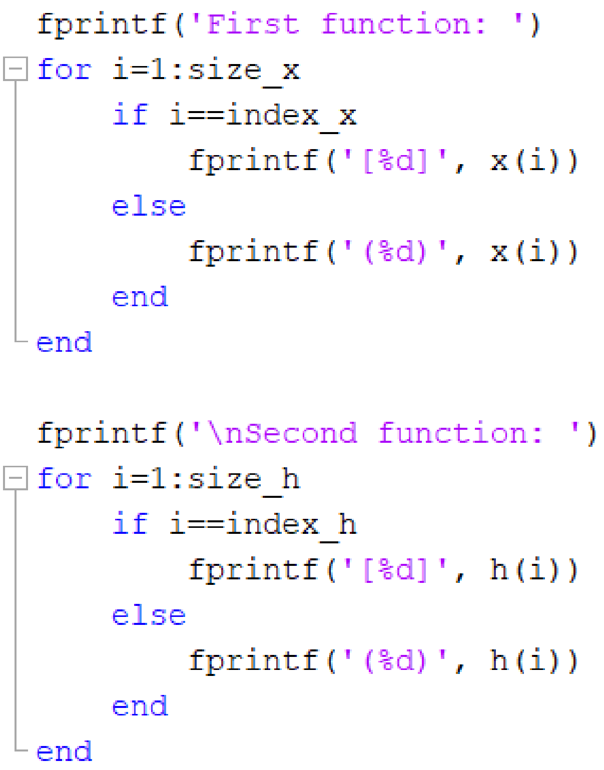
Second Function

My Convolution

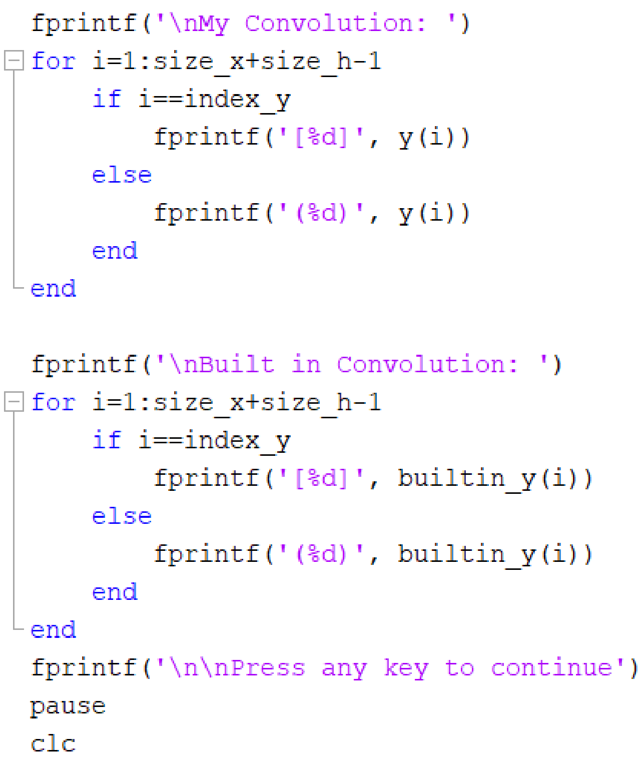
Built in Convolution

Vektör Gösteriminin Oluşturulması

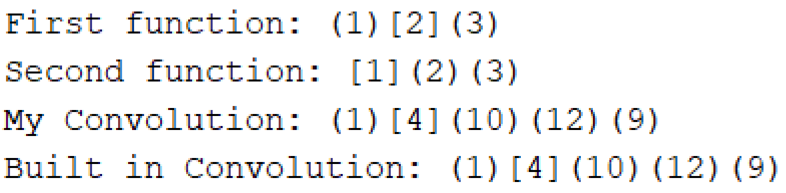
Bu kısımda sinyallerin 0. Elemanını belirtmek istedim. Bu yüzden disp() fonksiyonu yerine for döngüleri kullandım.



```
fprintf('First function: ')
for i=1:size_x
    if i==index_x
        fprintf('[%d]', x(i))
    else
        fprintf('(%d)', x(i))
    end
end

fprintf('\nSecond function: ')
for i=1:size_h
    if i==index_h
        fprintf('[%d]', h(i))
    else
        fprintf('(%d)', h(i))
    end
end
```



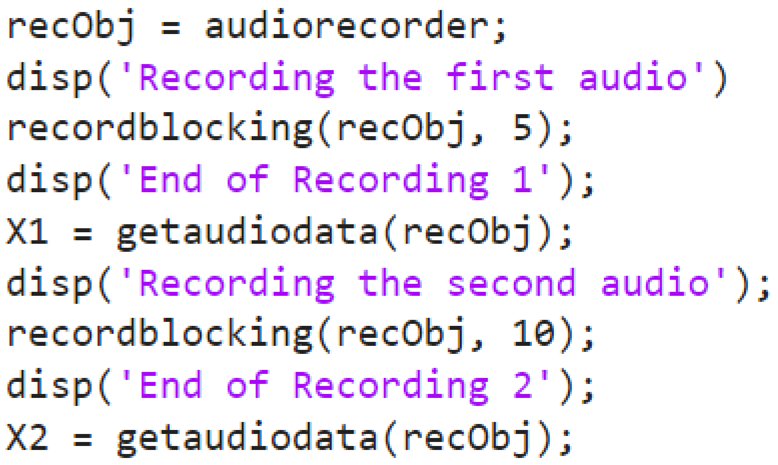
```
fprintf('\nMy Convolution: ')
for i=1:size_x+size_h-1
    if i==index_y
        fprintf('[%d]', y(i))
    else
        fprintf('(%d)', y(i))
    end
end

fprintf('\nBuilt in Convolution: ')
for i=1:size_x+size_h-1
    if i==index_y
        fprintf('[%d]', builtin_y(i))
    else
        fprintf('(%d)', builtin_y(i))
    end
end
fprintf('\n\nPress any key to continue')
pause
clc
```



```
First function: (1)[2](3)
Second function: [1](2)(3)
My Convolution: (1)[4](10)(12)(9)
Built in Convolution: (1)[4](10)(12)(9)
```

Çıkarım -->> Benim yaptığım konvolüsyon fonksiyonum ile hazır konvolüsyon fonskiyonu aynı sonuçları verdi.

Ses Kayıtlarının Alınması



```
recObj = audiorecorder;
disp('Recording the first audio')
recordblocking(recObj, 5);
disp('End of Recording 1');
X1 = getaudiodata(recObj);
disp('Recording the second audio');
recordblocking(recObj, 10);
disp('End of Recording 2');
X2 = getaudiodata(recObj);
```

Ses Kayıtlarına Konvolüsyon Yapılması

(A=0.8)

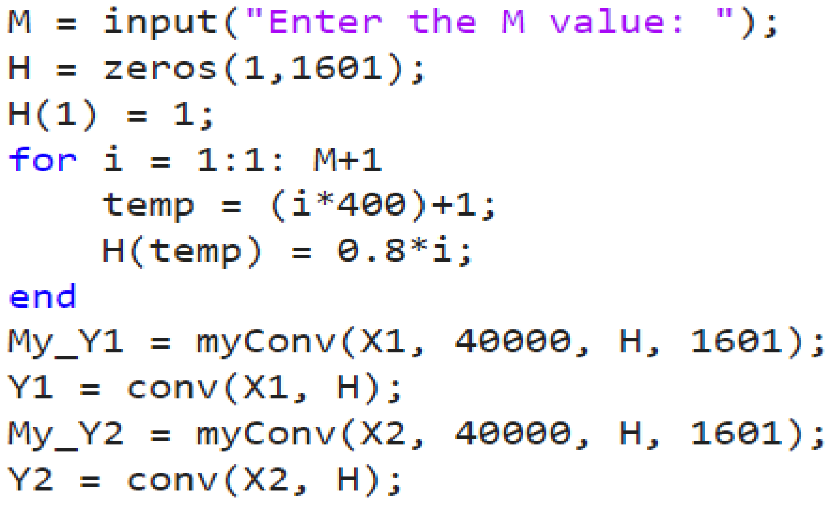
$y[n] = x[n] + \sum_{k=1}^{M} A.k.x[n - 400.k]$ fonksiyonunu sağlamak için x sinyalini:

- 1. değeri 1

- M = 2;
  - 401. değeri 0.8
  - 801. değeri 1.6
- M = 3;
  - 401. değeri 0.8
  - 801. değeri 1.6
  - 1201. değeri 2.4
- M = 4;
  - 401. değeri 0.8
  - 801. değeri 1.6
  - 1201. değeri 2.4
  - 1601. değeri 3.6
- Kalan değerlerin hepsi 0,

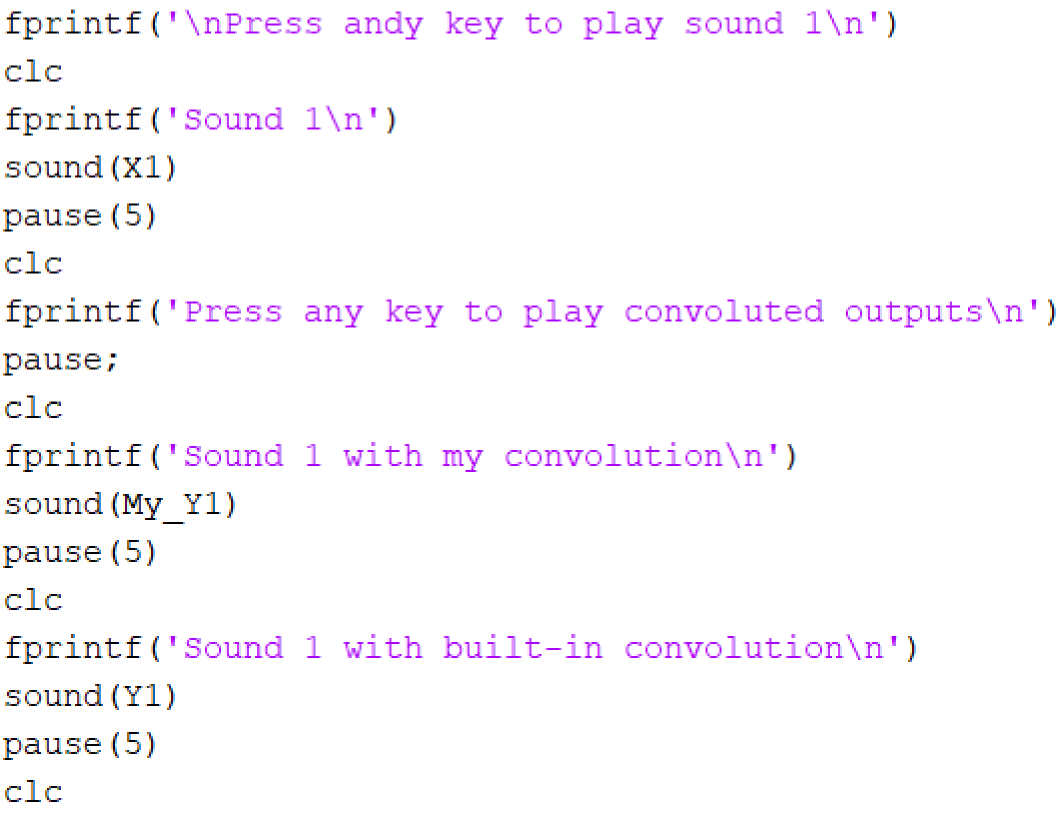
olan bir sinyal ile konvolüsyona sokmalıyız

NOT: Matlab'de dizi indeksleri 1'den başladığından her bir indis 1 arttırılmıştır.

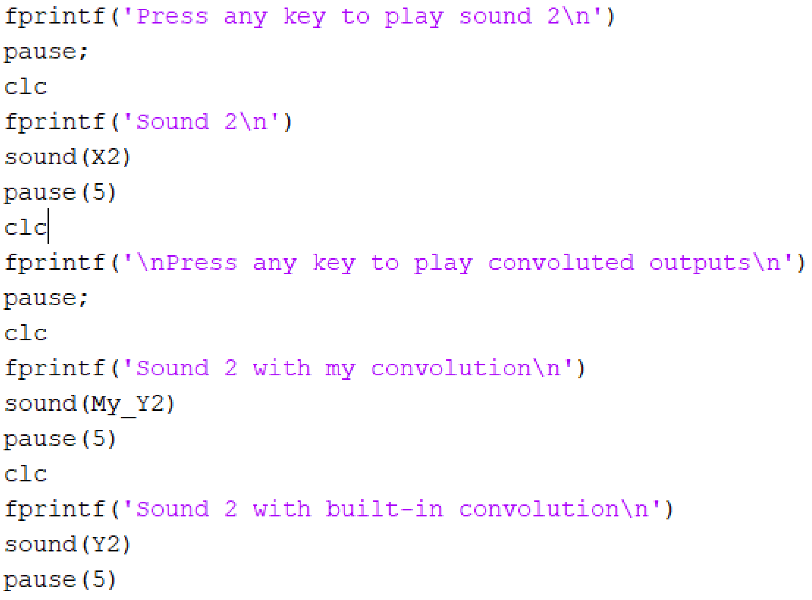


```
M = input("Enter the M value: ");
H = zeros(1,1601);
H(1) = 1;
for i = 1:1: M+1
    temp = (i*400)+1;
    H(temp) = 0.8*i;
end
My_Y1 = myConv(X1, 40000, H, 1601);
Y1 = conv(X1, H);
My_Y2 = myConv(X2, 40000, H, 1601);
Y2 = conv(X2, H);
```

Yeni ve Eski Kayıtların Çalınması



```
fprintf('\nPress andy key to play sound 1\n')
clc
fprintf('Sound 1\n')
sound(X1)
pause(5)
clc
fprintf('Press any key to play convoluted outputs\n')
pause;
clc
fprintf('Sound 1 with my convolution\n')
sound(My_Y1)
pause(5)
clc
fprintf('Sound 1 with built-in convolution\n')
sound(Y1)
pause(5)
clc
```



```
fprintf('Press any key to play sound 2\n')
pause;
clc
fprintf('Sound 2\n')
sound(X2)
pause(5)
clc
fprintf('\nPress any key to play convoluted outputs\n')
pause;
clc
fprintf('Sound 2 with my convolution\n')
sound(My_Y2)
pause(5)
clc
fprintf('Sound 2 with built-in convolution\n')
sound(Y2)
pause(5)
```

- Konvolüsyon sonucunda elde edilen seslerde yankı mevcut.
- M değerleri arttıkça;
    - Sesdeki yankı artıyor.
    - Robotikleşiyor.
    - Ses kirlileşiyor.
- Ayrıca örnekleme miktarı yüksek olmadığından ses robotik geliyor.

Kaynak Kodu

```
clear all
clc

size_x = input("Enter the size of the first signal: ");
index_x = input ("Enter the index of the 0. value of the first signal");
for i=1:1:size_x
 fprintf('Enter the %d. value of the first signal: ',i);
 x(i)=input('');
end

size_h = input("Enter the size of the second signal: ");
index_h = input ("Enter the index of the 0. value of the second signal: ");
for i=1:1:size_h
 fprintf('Enter the %d. value of the second signal: ',i);
 h(i)=input('');
end
clc
if index_x>index_h
    index_y = index_x;
else
    index_y = index_h;
end

y = myConv(x,size_x,h,size_h);
builtin_y = conv(x,h);

axis = -(index_y-1):1:size_x+size_h-1-index_y;
axis_x = -(index_x-1):1:size_x-index_x;
axis_h = -(index_h-1):1:size_h-index_h;

subplot(2,2,1)
stem(axis_x, x)
title('First function')

subplot(2,2,2)
stem(axis_h,h)
title('Second Function')


subplot(2,2,3)
stem(axis, y)
title('My Convolution')

subplot(2,2,4)
stem(axis, builtin_y)
title('Built in Convolution')

fprintf('First function: ')
for i=1:size_x
    if i==index_x
        fprintf('[%d]', x(i))
    else
        fprintf('(%d)', x(i))
    end
end

fprintf('\nSecond function: ')
for i=1:size_h
    if i==index_h
        fprintf('[%d]', h(i))
    else
        fprintf('(%d)', h(i))
    end
end

fprintf('\nMy Convolution: ')
for i=1:size_x+size_h-1
    if i==index_y
        fprintf('[%d]', y(i))
    else
        fprintf('(%d)', y(i))
    end
```

```
end

fprintf('\nBuilt in Convolution: ')
for i=1:size_x+size_h-1
    if i==index_y
        fprintf('[%d]', builtin_y(i))
    else
        fprintf('(%d)', builtin_y(i))
    end
end
fprintf('\n\nPress any key to continue')
pause
clc

recObj = audiorecorder;
disp('Recording the first audio')
recordblocking(recObj, 5);
disp('End of Recording 1');
X1 = getaudiodata(recObj);
disp('Recording the second audio');
recordblocking(recObj, 10);
disp('End of Recording 2');
X2 = getaudiodata(recObj);

M = input('Enter the M value');
H = zeros(1,1601);
H(1) = 1;
for i = 1:1: M+1
    temp = (i*400)+1;
    H(temp) = 0.8*i;
end
My_Y1 = myConv(X1, 40000, H, 1601);
Y1 = conv(X1, H);
My_Y2 = myConv(X2, 40000, H, 1601);
Y2 = conv(X2, H);

fprintf('\nPress andy key to play sound 1\n')
clc
fprintf('Sound 1\n')
sound(X1)
pause(5)
clc
fprintf('Press any key to play convoluted outputs\n')
pause;
clc
fprintf('Sound 1 with my convolution\n')
sound(My_Y1)
pause(5)
clc
fprintf('Sound 1 with built-in convolution\n')
sound(Y1)
pause(5)
clc
fprintf('Press any key to play sound 2\n')
pause;
clc
fprintf('Sound 2\n')
sound(X2)
pause(5)
clc
fprintf('\nPress any key to play convoluted outputs\n')
pause;
clc
fprintf('Sound 2 with my convolution\n')
sound(My_Y2)
pause(5)
clc
fprintf('Sound 2 with built-in convolution\n')
sound(Y2)
pause(5)

function [convoluted] = myConv(x, n, h, m)
    convoluted = zeros(1,n+m-1);
    for i=1:n+m-1
        for j=1:i
            if j <= m && i-j+1 <= n
                convoluted(i) = convoluted(i) + x(i-j+1)*h(j);
            end
        end
    end
end
```