

```

1 import sounddevice as sd
2 from scipy.io.wavfile import write
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 def MyConv(x: list, lenx: int, y:list, leny: int):
7     #x(t) fonksiyonu
8     #h(t) fonksiyonu
9     #h(t) fonksiyonunun uzunluğu = leny
10    result = np.array([0]*(leny+lenx-1))
11    j = -1
12    for i in x: #x(t) fonksiyonun elemanlarını alıyorum. Formüle baktım
13        j = j + 1
14        tmp = j
15        flag = 0
16        while(tmp < (leny + j) ): # result dizisinin içine çarpıp o indexteki değerle topluyorum.
17            result[tmp] = result[tmp] + i * y[flag]
18            tmp = tmp + 1
19            flag = flag + 1
20    return result

```

Kodumuzu incelemeye başlayacak olursak ilk 4 satırda gerekli kütüphaneleri ekledim. Kendi konvüle fonksiyonumu MyConv diye tanımladım. Sırasıyla da x dizisi, x uzunluğu, y dizisi, y uzunluğunu aldım.

Konvole sonuçlarımı result dizisinin içinde tuttum. Aslında biz önceden result dizisinin uzunluğunu biliyoruz. 10. satırda yaptığım işlem gibi dizilerin toplam uzunluğundan bir çıkartırsak bulmuş oluyoruz.

Konvolüsyon denklemine baktığımızda;

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

x dizisinden 1 tane değer alıp h dizisinin elemanlarıyla çarpıyoruz. h dizisinin içindeki k ifadesinden dolayı da x ekseninde kaydırmış

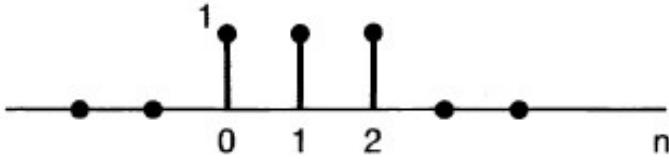
```

21 def start(sx, lenx): # grafikte x düzleminde x noktalarını doğru yazdırmak için dizinin içine atıyorum
22     tmp = sx
23     x = []
24     for i in range(lenx):
25         x.append(tmp)
26         tmp = tmp + 1
27     return x
28 def startn(sx, lenx): #fonksiyonun hangi index 0 noktasına gelecek onu tespit ediyorum.
29     tmp = sx
30     n = sx
31     for i in range(lenx):
32         if( (tmp + i) == 0 ):
33             n = i
34     return n

```

start fonksiyonum ile dizilerin x eksenindeki noktalarını dizide saklıyorum. Startn fonksiyonum ile de n = 0 noktası hangi indexte onu belirliyorum

Örnek verecek olursak:



Örnek verecek olursak:

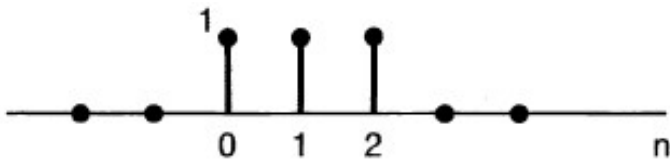
$x = [0, 1, 2]$

$n = 0$

fonksiyonlardan bu değerler üretilir.

```
36 def main():
37     x = []
38     y = []
39     lenx = int(input("x dizisinin uzunluğu: "))
40     sx = int(input("x dizisinin eksi sonsuzdan (x eksen düzleminde) gelirken 0'dan farklı değer geleceği x noktasını giriniz. "))
41     leny = int(input("y dizisinin uzunluğu: "))
42     sy = int(input("y dizisinin (x eksen düzleminde) eksi sonsuzdan gelirken 0'dan farklı değer geleceği y noktasını giriniz. "))
43     dizinx = []
44     diziny = []
45     diziresult = []
46     for i in range(lenx): # x ve y fonksiyonlarının değerlerini alıyorum
47         temp = float(input("X[{0}] degeri: ".format(i)))
48         x.append(temp)
49     for i in range(leny):
50         temp = float(input("Y[{0}] degeri: ".format(i)))
51         y.append(temp)
52     result = MyConv(x, lenx, y, leny)
53
```

x ve y dizilerinin uzunluğunu aldım. Sonra for döngüleri ile de dizinin değerlerini alıyorum. sx ve sy dediğim kavramlara değinmek istiyorum. Eksi sonsuzdan gelirken fonksiyonum hangi noktada 0'dan farklı değer alıyor kullanıcıdan onu istiyorum.



Örnek verecek olursak:

0 noktasından itibaren 1 değerini almaya başladığı için $sx = 0$ ya da $sy = 0$ olur.

```
54 dizinx = start(sx, lenx) #fonksiyonun grafiği için x noktalarını belirledim. Mesela sx = -1 ise dizinx = -1, 0 1 gibi
55 print(dizinx)
56 nx = startn(sx, lenx) #fonksiyonun hangi index 0 noktasına gelecek onu tespit ediyorum. Mesela
57 diziny = start(sy, leny) #fonksiyonun grafiği için x noktalarını belirledim
58 print(diziny)
59 ny = startn(sy, leny) #fonksiyonun hangi index 0 noktasına gelecek onu tespit ediyorum.
60 sr = sy + sx #konvule edeceğim fonksiyonun hangi degerden itibaren 0'dan farklı sayı gelecek onu hesaplıyorum
61 n = startn(sr, len(result)) #fonksiyonun hangi index 0 noktasına gelecek onu tespit ediyorum.
62 diziresult = start(sr, len(result)) #fonksiyonun grafiği için x noktalarını belirledim
```

```
63 print("x dizisi: ", x)
64 print("x: n = 0 noktasındaki index degeri: ", nx)
65 print("y dizisi: ", y)
66 print("y: n = 0 noktasındaki index degeri: ", ny)
67 print("result dizisi: ", result)
68 print("result: n = 0 noktasındaki index degeri: ", n)
69 result2 = np.convolve(x, y)
70 print("hazır fonksiyon result: ", result2)
```

Burada da x, y, result (kendi konvule ettiğim dizi), result2(hazır konvolüsyon)dizilerini yazdırdım.

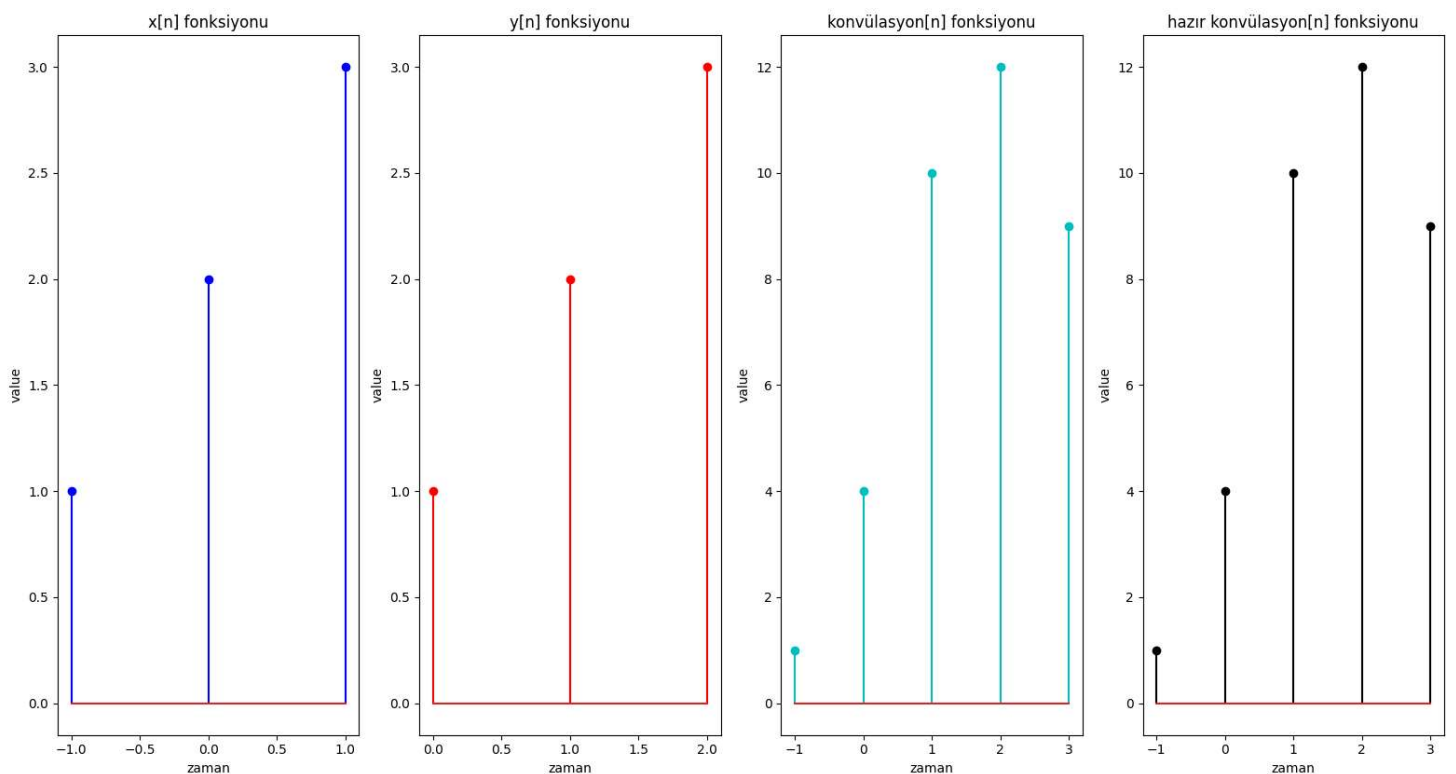
```

71     #x grafiği
72     t = dizinx
73     a = x
74     plt.subplot(1, 4, 1)
75     plt.stem(t, a, 'b')
76     plt.xlabel("zaman")
77     plt.ylabel("value")
78     plt.title("x[n] fonksiyonu")
79     #y grafiği
80     t = diziny #
81     a = y
82     plt.subplot(1, 4, 2)
83     plt.stem(t, a, 'b')
84     plt.xlabel("zaman")
85     plt.ylabel("value")
86     plt.title("y[n] fonksiyonu")
87     #sonucun grafiği
88     t = diziresult
89     a = result
90     plt.subplot(1, 4, 3)
91     plt.stem(t, a, 'b')
92     plt.xlabel("zaman")
93     plt.ylabel("value")
94     plt.title("konvülyasyon[n] fonksiyonu")
95     #hazır fonksiyon kullanıldı
96     t = diziresult
97     a = result2
98     plt.subplot(1, 4, 4)
99     plt.stem(t, a, 'b')
100    plt.xlabel("zaman")
101    plt.ylabel("value")
102    plt.title("hazır konvülyasyon[n] fonksiyonu")
103    plt.show()

```

$x = [1, 2, 3]$, $s_x = -1$, $n_x = 1$
 $y = [1, 2, 3]$, $s_y = 0$, $n_y = 0$
 $result = [1, 4, 10, 12]$, $n = 1$

$dizinx = [-1, 0, 1]$
 $diziny = [0, 1, 2]$
 $diziresult = [-1, 0, 1, 2, 3]$



```

105 #5 saniyelik ses kaydı
106 fs = 10000 # Saniye başına örnekleme
107 seconds = 5
108 print("5 saniyelik ses kaydı başladı\n")
109 myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2) #kaydediyor
110 sd.wait()
111 print("kayıt bitti")
112 write('output12.wav', fs, myrecording) # WAV dosyası şeklinde kaydediyor
113 result = np.array(myrecording).flatten() #Sesi matrix şeklinde depoladığı için dizi haline dönüştürmemiz gerek.
114 devam = 1
115 while(devam): #hazır konvolüsyon kullandım
116     M = int(input("-Hazır konvolüsyon fonksiyonu-Formüldeki M sayısını giriniz: "))
117     A = 0.8
118     h = [0]*M * 400 #y[n] fonksiyonunu M değerine bağlı olarak bir dizi oluşturuyorum.
119     h.append(0) #0 eklemem gerekiyordu çünkü 400. noktada dürtümün olmasını istiyorsam 401 boyutluk dizi oluşturmam gerek
120     h[0] = 1
121     for i in range(M):
122         h[(i + 1) * 400] = A * (i + 1)
123     convolved = np.convolve(result,h)
124     sd.play(convolved, blocking = True)
125     devam = int(input("İşleme devam etmek istiyorsanız 1'e basınız."))
126     if(devam != 1):
127         devam = 0
128     devam = 1
129 while(devam): #benim konvolüsyon fonksiyonum kullandım
130     M = int(input("-Benim konvolüsyon fonksiyonum-Formüldeki M sayısını giriniz: "))
131     A = 0.8
132     h = [0]*M * 400 #y[n] fonksiyonunu M değerine bağlı olarak bir dizi oluşturuyorum.
133     h.append(0) #0 eklemem gerekiyordu çünkü 400. noktada dürtümün olmasını istiyorsam 401 boyutluk dizi oluşturmam gerek
134     h[0] = 1
135     for i in range(M):
136         h[(i + 1) * 400] = A * (i + 1)
137     convolved = MyConv(result, len(result), h, len(h))
138     sd.play(convolved, blocking = True)
139     devam = int(input("İşleme devam etmek istiyorsanız 1'e basınız."))
140     if(devam != 1):
141         devam = 0
142

```

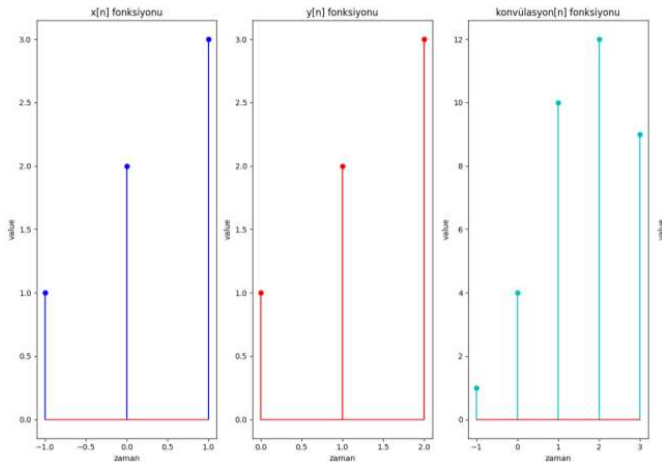
Örnekleme frekansımı 10.000 değerini aldım. 5 saniyelik ses kaydı için konuşuyoruz. Ses dosyasını matrix şeklinde depoladığı için önce onu diziye çeviriyoruz. 113. satırda o işlemi yaptım. Sesi kaydettikten sonra while döngüsüne aldım. Aynı ses dosyası üzerinde istediğiniz M değerlerini girerek çıktığı almamızı sağlamak için. Eğer M değerlerinin hepsini denediyseniz while döngüsünden çıkabilirsiniz. Öyle tasarladım. İlk baştaki while döngüsünde hazır fonksiyonu kullandım. İkinci while döngüsünde benim yazdığım fonksiyonu kullandım. Tekrardan değinmek istiyorum while döngüsü -115 ve 129. satır- olayına. İsteddiğiniz M değerini girerek istediğiniz çıktığı elde edebilirsiniz. Çıktı elde etmek istemiyorsanız döngüden çıkabilirsiniz.


```

143 #10 saniyelik ses kaydı
144 fs = 10000 # Saniye başına örnekleme |
145 seconds = 10
146 print("10 saniyelik ses kaydı başladı\n")
147 myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2) #kaydediyor
148 sd.wait()
149 print("kayıt bitti")
150 write('output123.wav', fs, myrecording) # WAV dosyası şeklinde kaydediyor
151 result = np.array(myrecording).flatten() #Sesi matrix şeklinde depoladığı için dizi haline dönüştürmemiz gerek.
152 devam = 1
153 while(devam): #hazır konvolüsyon kullandım
154     M = int(input("-Hazır konvolüsyon fonksiyonu-Formüldeki M sayisini giriniz: "))
155     A = 0.8
156     h = [0]* M * 400 #y[n] fonksiyonunu M değerine bağlı olarak bir dizi oluştuyorum.
157     h.append(0) #0 eklemem gerekiyordu çünkü 400. noktada dürtümün olmasını istiyorsam 401 boyutluk dizi oluşturmam gerek
158     h[0] = 1
159     for i in range(M):
160         h[(i + 1) * 400] = A * (i + 1)
161     convolved = np.convolve(result,h)
162     sd.play(convolved, blocking = True)
163     devam = int(input("İşleme devam etmek istiyorsanız 1'e basınız."))
164     if(devam != 1):
165         devam = 0
166     devam = 1
167 while(devam): #benim konvolüsyon fonksiyonumu kullandım
168     M = int(input("-Benim konvolüsyon fonksiyonum-Formüldeki M sayisini giriniz: "))
169     A = 0.8
170     h = [0]* M * 400 #y[n] fonksiyonunu M değerine bağlı olarak bir dizi oluştuyorum.
171     h.append(0) #0 eklemem gerekiyordu. Misal 400. noktada dürtümün olmasını istiyorsam 401 boyutluk dizi oluşturmam gerek
172     h[0] = 1
173     for i in range(M):
174         h[(i + 1) * 400] = A * (i + 1)
175     convolved = MyConv(result, len(result), h, len(h))
176     sd.play(convolved, blocking = True)
177     devam = int(input("İşleme devam etmek istiyorsanız 1'e basınız."))
178     if(devam != 1):
179         devam = 0
180     devam = 1
181 if __name__ == "__main__":
182     main()

```

10 saniyelik ses kaydı, 5 saniyelik ses kaydıyla tamamen aynıdır. Tek fark kaydedilen sesin süresidir.



4. soruyu yorumlamaya gelirse M değerini artırdığımızda ses kaydının genel olarak genliği arttığı için ses daha yüksek duyulmaya başlanıyor. Yandaki grafiğe bakınca da doğru yorumladığımızı gözlemleyebiliriz. Mesela $x = 2$ noktalarında değerler 2 ama konvüle ettikten sonra $x = 2$ noktasında 12 olmuş. Buradan da doğrulamış olduk.

Buna ek olarak seste eko oluyor. Üst üste binmeler olduğu için. M değeri arttıkça da eko olayını çok iyi şekilde gözlemliyoruz.

Not: Sonradan eklemeler yaptım ama kodun mantığını değiştirmedığı için sonradan düzenleme yapmadım.

Kaynakça:

- https://matplotlib.org/stable/api/as_gen/matplotlib.pyplot.plot.html
- Dersin slaytları