

Sinyal ve Sistemler Ödev-1

Şahin Doğruca

22011049

Soru-1

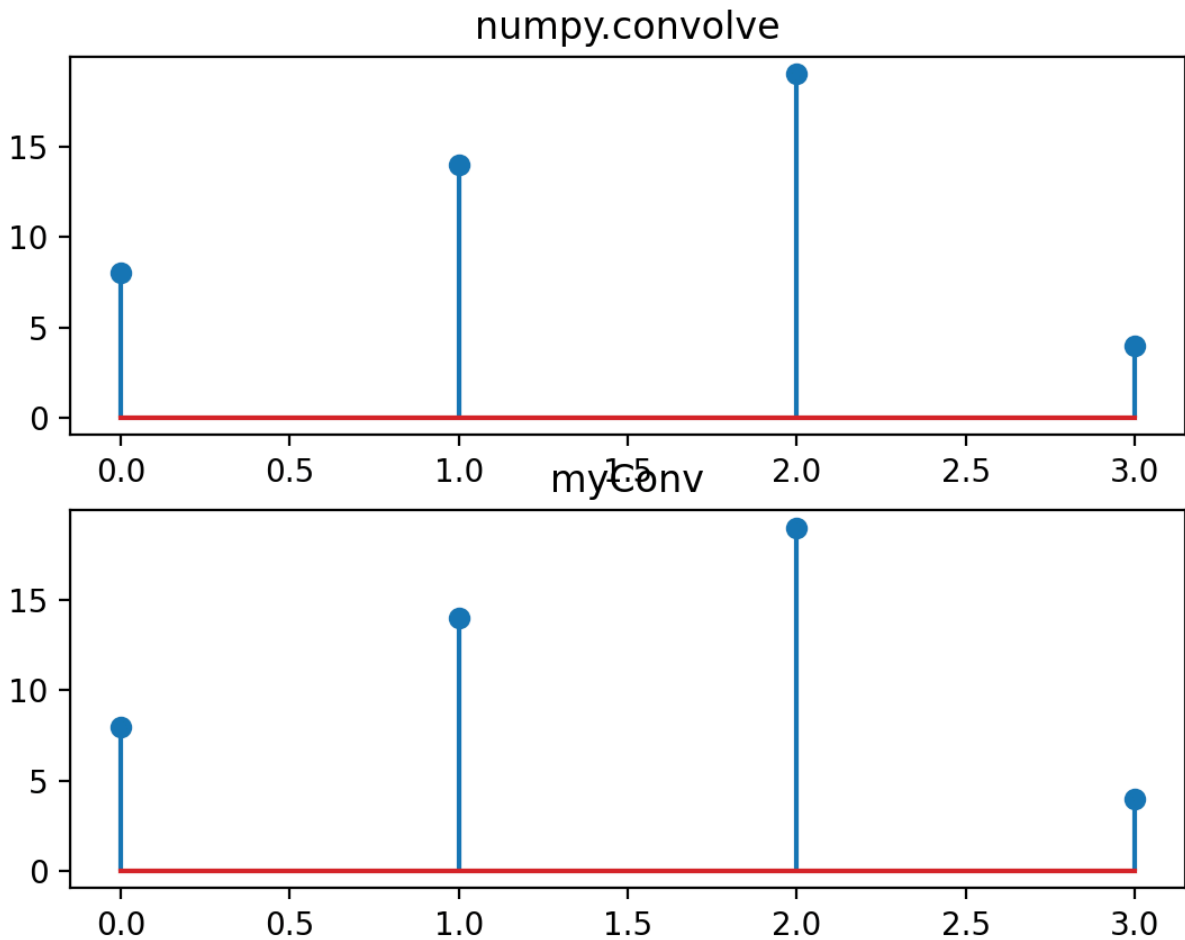
Bu Soruda bizden kendi ayrık konvülyasyon fonksiyonumuzu yazmamız isteniyor. Ayrık konvülyasyonda yapmamız gereken impulse responsu x fonksiyonu üzerinde kaydırıp üst üste gelen vektörlerde skaler çarpım yapmak.

```
27 | # myConv function that performs convolution of two discrete signals x and h are given as dictionaries
    | Codiumate: Options | Test this function
28 | def myConv(x, h):
29 |     y = {}
30 |     for n in range(min(x.keys()) + min(h.keys()), max(x.keys()) + max(h.keys()) + 1):
31 |         y[n] = 0
32 |         for k in x.keys():
33 |             if n - k in h.keys():
34 |                 y[n] += x[k] * h[n - k]
35 |     return y
36 |
```

Minumum indexlerin toplamından başlayıp, maximumların toplamına kadar devam ediyor çünkü 0 ve 1 den başlayan 2 dizi hayal edelim bunların örtüşmeye başlaması 1. İndexten itibaren olacaktır. Daha sonra tüm x değerleri üzerinde geziyor eğer üst üste binen var ise çarpıp topluyor.

Burda kendi konvölasyon fonksiyonumuz ile numpy'ın bize sunduğu konvölasyon fonksiyonlarını yazdırıp grafiklerine bakıyoruz.

```
Enter a valid number of samples: 2
Enter the index of sample 1: 1
Enter the value of sample 1: 4
Enter the index of sample 2: 2
Enter the value of sample 2: 1
x[n]: {1: 2.0, 2: 3.0, 3: 4.0}
h[n]: {1: 4.0, 2: 1.0}
numpy.convolve: [ 8. 14. 19.  4.]
myConv: {2: 8.0, 3: 14.0, 4: 19.0, 5: 4.0}
```



Grafikte de aynı değerleri yakaladığını görüyoruz.

sound_conv fonksiyonunda bize verilen impulse_response ile sesleri konvüle ediyoruz. Karşılaştırma bakımından np.convolve ile myConv'uda ekliyoruz.

M değerinin etkisine baktığımız zaman M=2 olduğu zaman ses orijinal sese yakın ama kalitesiz gibi oluyor. M = 5 olduğu zaman ise sesin kalitesi dahada düşüp boğukmuş gibi geliyor sanki gürültü ekliyor gibi.

```
84 def sound_conv(audio, fs, M=3, A=2):
85     # Create the impulse response signal
86     impulse_response = np.zeros(3000 * M + 1)
87     for k in range(1, M+1):
88         impulse_response[3000 * k] = A ** (-k) * k
89
90     # Perform convolution
91     output_array = np.convolve(audio, impulse_response, mode='same')
92     output_array2 = myConv2(audio, impulse_response)
93
94     # Normalize the outputs
95     output_array = output_array / np.max(np.abs(output_array))
96     output_array2 = output_array2 / np.max(np.abs(output_array2))
97
98     # Play the output audios
99     sd.play(output_array, fs)
100    sd.wait()
101    sd.play(output_array2, fs)
102    sd.wait()
103
104
105 # Main menu
```

