



YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM2041 – Bilg. Müh. İçin Sinyal ve Sistemler
Gr: 1

Doç. Dr. Ali Can Karaca

22.11.2022 - Ödev

Öğrenci: Selahattin Yasin ÇAYCI

No: 20011099

E-posta: yasin.cayci@std.yildiz.edu.tr

Soru 1

Kullanıcıdan alınan n ve m uzunluklu iki farklı ayrık zamanlı işaretin ($x[n]$ ve $y[m]$) konvolüsyon toplamını hesaplayan algoritmayı Python ya da Matlab dilinde parametrik olarak kodlayınız. **Bu aşamada hazır konvolüsyon fonksiyonu kullanılmayacaktır. Özetle;** $\text{myConv}(x, n, y, m)$: x ve y ayrık zamanlı işaretlerinin boyutları n ve m ile bu dizilerin değerleri ve indisleri kullanıcıdan alınacak ve konvolüsyon toplamı sonucu hesaplatılacaktır.

Örnek: $x[n] = [1 \ 2 \ 3]$, $y[n] = [1 \ 2 \ 3]$, Sonuç $(x * y) = [1 \ 4 \ 10 \ 12 \ 9]$ (Altı çizili ve kalın değerler $n=0$ noktasındaki örneklerdir)

Cevap:

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import sounddevice as sd
4
5  def myConvolve(signal1, signal2, m, n):
6      if m == 0:
7          raise ValueError('signal1 boş olamaz')
8      if n == 0:
9          raise ValueError('signal2 boş olamaz')
10     convLen = m + n - 1
11     y = np.zeros(convLen, dtype=float)
12     for i in range(convLen):
13         signal2_start = max(0, i - m + 1)
14         signal2_end = min(i + 1, n)
15         signal1_start = min(i, m - 1)
16         for j in range(signal2_start, signal2_end):
17             y[i] = y[i] + signal1[signal1_start] * signal2[j]
18             signal1_start = signal1_start - 1
19     return y
20
```

signal1: Birinci sinyal

signal2: İkinci sinyal

m: Birinci sinyalin boyutu

n: İkinci sinyalin boyutu

Soru 2

Birinci maddede kendi yazdığınız ve kullandığınız dilde hazır bulunan konvolüsyon toplamı fonksiyonlarının sonuçlarını iki farklı veri seti üzerinde **grafiksel ve vektörel** olarak karşılaştırınız. Bu karşılaştırma için en fazla 5 farklı sayı içeren veri girişi yapmanız gerekmektedir. **Birinci soruda altı çizili olarak verilen işaretin sıfır noktasına dikkat ediniz.**

**** **Grafiksel gösterim:** $X[n]$, $Y[m]$, MyConv sonucu ve Hazır fonksiyon sonucu elde edilen ayrık değerlerin dördünü içeren grafiklerin çizilmesidir. ****

Vektörel gösterim: $X[n]$, $Y[m]$, MyConv sonucu ve Hazır fonksiyon sonucu elde edilen ayrık değerlerin dördünü ekrana yazdırılmasıdır.

Cevap:

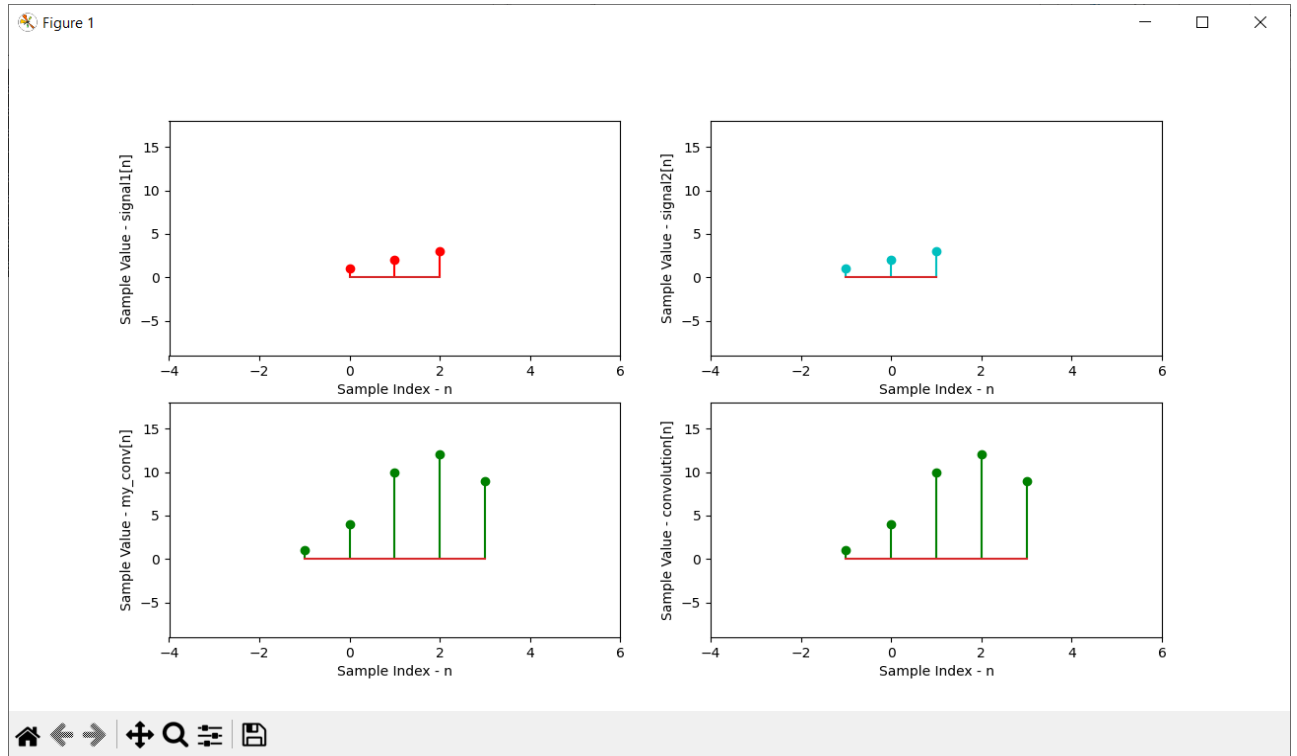
Öncelikle, sırasıyla kendi oluşturduğum ayrık zamanlı sinyallerin değerleri input olarak alındı:

```
113 len1 = int(input("1.sinyal dizisinin eleman uzunluğunu giriniz:"))
114 signalX = np.zeros(len1)
115 for i in range(0,len1):
116     signalX[i] = input("Eleman:")
117 len2 = int(input("2.sinyal dizisinin eleman uzunluğunu giriniz:"))
118 signalY = np.zeros(len2)
119 for i in range(0,len2):
120     signalY[i] = input("Eleman:")
121 zeroIndexX = int(input("1.Dizinin sıfır noktasını giriniz:"))
122 zeroIndexY = int(input("2.Dizinin sıfır noktasını giriniz:"))
123
124
125 graph_versus(signalX, signalY, zeroIndexX, zeroIndexY)
126 vector_versus(signalX, signalY)
```

```
1.sinyal dizisinin eleman uzunluğunu giriniz:3
Eleman:1
Eleman:2
Eleman:3
2.sinyal dizisinin eleman uzunluğunu giriniz:3
Eleman:1
Eleman:2
Eleman:3
1.Dizinin sıfır noktasını giriniz:0
2.Dizinin sıfır noktasını giriniz:1
```

Daha sonra girilen inputlar kendi fonksiyonum(myConvolve) ve diğer fonksiyon(np.convolve) da işleme tutuldu. Toplamda 4 sinyalin grafiksel ve vektörel karşılaştırmaları için gereken fonksiyonlar oluşturuldu:

```
21 def initialize(index, length):
22     x = []
23     for i in range(length):
24         x.append(index)
25         index = index + 1
26     return x
27
28 def graph_versus(signalX, signalY, zeroIndexX, zeroIndexY):
29
30     conv_ary = myConvolve(signalX, signalY, len(signalX), len(signalY))
31     convolution = np.convolve(signalX, signalY)
32
33     maxY = max(signalX) * max(signalY) * 2
34     minY = maxY * -1 / 2
35     maxX = max(len(signalX), len(signalY)) * 2
36     minX = min(-zeroIndexX, -zeroIndexY) - max(len(signalX), len(signalY))
37     startX = initialize(-zeroIndexX, len(signalX))
38     startY = initialize(-zeroIndexY, len(signalY))
39     startConv = initialize(-(zeroIndexX + zeroIndexY), len(signalX) + len(signalY) - 1)
40
41     fig = plt.figure(figsize=(13, 7), dpi=80, facecolor='w', edgecolor='k')
42     plt.subplot(2, 2, 1)
43     plt.stem(startX, signalX, "ro")
44     plt.xlabel('Sample Index - n')
45     plt.ylabel('Sample Value - signal1[n]')
46     plt.ylim([minY, maxY])
47     plt.xlim([minX, maxX])
48
49     plt.subplot(2, 2, 2)
50     plt.stem(startY, signalY, 'c')
51     plt.xlabel('Sample Index - n')
52     plt.ylabel('Sample Value - signal2[n]')
53     plt.ylim([minY, maxY])
54     plt.xlim([minX, maxX])
55
56     plt.subplot(2, 2, 3)
57     plt.stem(startConv, conv_ary, 'g')
58     plt.xlabel('Sample Index - n')
59     plt.ylabel('Sample Value - my_conv[n]')
60     plt.ylim([minY, maxY])
61     plt.xlim([minX, maxX])
62
63     plt.subplot(2, 2, 4)
64     plt.stem(startConv, convolution, 'g')
65     plt.xlabel('Sample Index - n')
66     plt.ylabel('Sample Value - convolution[n]')
67     plt.ylim([minY, maxY])
68     plt.xlim([minX, maxX])
69
70     plt.show()
71
72 def vector_versus(signalX, signalY):
73     conv_ary = myConvolve(signalX, signalY, len(signalX), len(signalY))
74     convolution = np.convolve(signalX, signalY)
75
76     print(signalX)
77     print(signalY)
78     print(conv_ary)
79     print(convolution)
80
```



```
Eleman:2
Eleman:3
1.Dizinin sıfır noktasını giriniz:0
2.Dizinin sıfır noktasını giriniz:1
[1. 2. 3.]
[1. 2. 3.]
[ 1.  4. 10. 12.  9.]
[ 1.  4. 10. 12.  9.]
```

signalX: İşleme sokulacak birinci sinyal

signalY: İşleme sokulacak ikinci sinyal

conv_ary: Kendi fonksiyonumun döndürdüğü konvolüsyon işlemi sonucu oluşturulan sinyal

convolution: Hazır fonksiyonun döndürdüğü konvolüsyon işlemi sonucu oluşturulan sinyal olarak alınmıştır.

Soru 3

Kendi sesinizi 5 ve 10 saniyelik süreler için ayrı ayrı kaydediniz. (Örneğin X1 ve X2 değişkenleri olsun)

Cevap:

Öncelikle ses kaydını alan, aldığı sesi oynatan, daha sonra ise aldığı sesi bir değişkene kaydedip kaydettiği değişkeni döndüren bir fonksiyon (audiorecord) oluşturuldu:

```
81 def audiorecord(duration):
82     freq = 44100
83     recording = sd.rec(int(duration * freq), samplerate=freq, channels=1)
84     print("Recording...")
85     sd.wait(duration)
86     print("Recording is over")
87     rec1 = np.array(recording).flatten()
88     sd.play(rec1, blocking=True)
89     return rec1
90
```

Daha sonra bu fonksiyonlarla iki defa kayıt alınarak elde edilen veriler sırasıyla record 5 ve record10 değişkenlerine kaydedildi:

```
127 record5 = audiorecord(5)
128 record10 = audiorecord(10)
```

```
Recording...
Recording is over
Recording...
Recording is over
```

Soru 4

4- (50 puan) Madde-3'te kaydettiğiniz ses dosyalarını (**X1 ve X2**) aşağıdaki bağıntısı verilen sisteme girdi olarak ayrı ayrı uygulayınız ve sonuçlarını kaydediniz. **Bu işlemi 4.1- 4.6 maddelerine göre yapınız. A=0.8 olarak alınız. Çarpma işlemi olarak denklemden operatörü kullanılmıştır.**

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

1. **M** değeri 2, 3 ve 4 için, 4.2 – 4.5'i bağımsız ve ayrı ayrı tekrar ediniz.
2. Kaydettiğiniz her bir ses dosyası (**X1 ve X2**) için 4.2 – 4.5 maddelerini bağımsız ve ayrı ayrı tekrar ediniz.
3. Kendi yazdığınız **myConv** fonksiyonunu kullanınız. Örneğin, $X[n] = X1[n]$ için **myY1[n]** çıktı değişkeni kaydediniz.
4. **Python veya MATLAB hazır Conv** fonksiyonunu kullanınız. Örneğin, $X[n] = X1[n]$ için **Y1[n]** çıktı değişkeni kaydediniz.
5. 4.3 ve 4.4'te elde ettiğiniz tüm sonuçları (örn: **myY1[n]** ve **Y1[n]**) ve kaydettiğiniz giriş verilerini (**X1[n]** ve **X2[n]**) seslendiriniz.
6. Önceki maddelerdeki tüm deneylerde elde ettiğiniz çıktılar ve girdiler arasında nasıl bir fark olduğunu M değerinin değişiminin (2, 3, ve 4) etkisini de ekleyerek yorumlayınız.

Cevap:

Öncelikle verilen formüle göre dürtü yanıtı olan h dizisi oluşturuldu. Daha sonra ise bu dürtü yanıtı kaydedilen ses ile konvolüsyona sokuldu. Oluşan sonuçlar Y1,Y2, my_Y1,my_Y2 değişkenlerine atanıp oynatıldı:

```
91 def newAudioWithMyConvolve(record, M):
92     h = np.zeros(M*400+1, dtype=float)
93     h[0] = 1
94     k = 1
95     for i in range(k,M+1):
96         j = i * 400
97         h[j] = 0.8
98     print("Processing...")
99     myAudioConv = myConvolve(record,h,len(record),len(h))
100     return myAudioConv
101
102 def newAudioWithNumpyConvolve(record, M):
103     h = np.zeros(M*400+1, dtype=float)
104     h[0] = 1
105     k = 1
106     for i in range(k,M+1):
107         j = i * 400
108         h[j] = 0.8
109     print("Processing...")
110     myAudioConv = np.convolve(record,h)
111     return myAudioConv
112
```

```

130 Y1 = newAudioWithNumpyConvelve(record5, 2)
131 sd.play(Y1, blocking=True)
132 Y2 = newAudioWithNumpyConvelve(record10, 2)
133 sd.play(Y2, blocking=True)
134 my_Y1 = newAudioWithMyConvolve(record5, 2)
135 sd.play(my_Y1, blocking=True)
136 my_Y2 = newAudioWithMyConvolve(record10, 2)
137 sd.play(my_Y2, blocking=True)

```

Formülde toplam sembolündeki $M = 2$ iken kendi yazdığım konvolüsyon fonksiyonu ile output 5 saniyelik kayıta 2 dk 8 saniye, 10 saniyelik kayıta ise 4 dakika 20 saniyede oluşturuldu. $M = 3$ iken orantılı olarak 3 dakika ve 6 dakikada output elde edildi. $M = 4$ iken de aynı oran orantı devam etti. Numpy kütüphanesindeki convolve fonksiyonu ise her değerde hemen sonuç verdi.

```

Eleman:2
Eleman:3
1.Dizinin sıfır noktasını giriniz:0
2.Dizinin sıfır noktasını giriniz:1
[1. 2. 3.]
[1. 2. 3.]
[ 1.  4. 10. 12.  9.]
[ 1.  4. 10. 12.  9.]
Recording...
Recording is over
Recording...
Recording is over
Processing...
Processing...

```

Sonuç olarak giriş sesleri olduğu gibi yeniden seslendirilirken, çıktılar elde edilen seslerin daha buğulu ve yankılı hali olarak geri döndü.