Sinyaller ve Sistemler Dersi 1. Ödev - Konvolüsyon

Öğrenci Adı: Mehmet Ali Duran

Öğrenci Numarası: 21011090

Dersin Eğitmeni: Ali Can Karaca

Ödev için kullanımı daha kolay olduğundan dolayı Python dili tercih edilmiştir. Programda kullanılan her bir fonksiyonun parametreleri ve ne iş yaptığı kısaca yorum satırları ile açıklanmıştır. Her soru için gerekli kod parçaları üzerinde ilgili olduğu soru belirtilerek tek bir Python dosyasında alt alta yazılmıştır.

1. Soruda verilen iki sayı dizisi için konvolüsyon hesabı yapan bir fonksiyon yazılması istenmiş. Kullanıcıdan önce girilecek dizilerin boyutları daha sonra kendileri alınmıştır. Oluşan çıktıları grafik olarak gösterirken değerlerin zaman ekseninde doğru yerde olmasını sağlamak adına x_zero ve y_zero değerleri de kullanıcıdan parametre olarak alınmıştır. Örnek olarak [1 **2** 3] değerleri girilirse x_zero değeri 2 değerinin bulunduğu 1 indisi olmalıdır.

x = [1 **2** 3]

x_zero = 1

```python
# Kullanicidan alinan x ve y dizileri ve bu dizilerin 0 noktalarinin hangi indiste oldugunu belirten x_zero ve y_zero
# degerleri alinir.
def myConv(x, x_zero, y, y_zero):

    # kisa olan diziyi kaydirmak daha kolay oldugu icin kisa olan y dizisine atanir
    if len(y) > len(x):
        x, y = y, x
        x_zero, y_zero = y_zero, x_zero

    # fonksiyon zaman ekseninde terslenir
    y = y[::-1]

    y_zero = len(y) - y_zero - 1
    # sonucun sıfır noktasi nerede hesaplanir
    result_zero = 0

    result_length = len(x) + len(y) - 1
    result = [0] * result_length
    # konvolusyon islemi yapilir
    for shift in range(-len(y) + 1, len(x)):
        for i in range(len(y)):
            if 0 <= shift + i < len(x):
                result[shift + len(y) - 1] += x[shift + i] * y[i]
            if shift + i == x_zero and i == y_zero:
                result_zero = shift + len(y) - 1
    # sonuc olarak sonuc dizisi ve sifir degerinin oldugu indis geri dondurulur
    return result, result_zero
```

1. Örnek

X = [1 **2** 3] Y = [**1** 2 3] x_zero = 1 y_zero = 0

Verilen değerler için çıktı aşağıdaki gibi olmalıdır:

```
PS C:\Users\mehme> python -u "c:\Users\mehme\Masaüstü\21011090.py"
x dizisinin eleman sayısını girin (n): 3
3 elemanlı x dizisini girin (örnek: 1 2 3 ... 3): 1 2 3
y dizisinin eleman sayısını girin (m): 3
3 elemanlı y dizisini girin (örnek: 1 2 3 ... 3): 1 2 3
x_zero: 1
y_zero: 0
[1, 4, 10, 12, 9]
PS C:\Users\mehme>
```

Diğer bir örnek için girdiler;

X = [0 **3** -2 4 6] Y = [2 **3** 1] x_zero = 1 y_zero = 1

Çıktılar:

```
PS C:\Users\mehme> python -u "c:\Users\mehme\Masaüstü\21011090.py"
x dizisinin eleman sayısını girin (n): 5
5 elemanlı x dizisini girin (örnek: 1 2 3 ... 5): 0 3 -2 4 6
y dizisinin eleman sayısını girin (m): 3
3 elemanlı y dizisini girin (örnek: 1 2 3 ... 3): 2 3 1
x_zero: 1
y_zero: 1
[0, 6, 5, 5, 22, 22, 6]
PS C:\Users\mehme>
```

2. Soruda 1 soruda yazılan konvolüsyon fonksiyonu ile hazır konvolüsyon fonksiyonunun çıktılarının grafiksel ve vektörel olarak karşılaştırılması istenmiş. Parametre olarak ise girilen dizilerin boyutları toplamı 5'i geçmeyecek şekilde bir sınırlandırma koyulmuş. Girdileri bu şekilde almak için ayrıca bir get_input_maks5() adında bir fonksiyon yazılmıştır. Girilen ilk dizinin boyutu ne ise diğeri de bunu 5'e tamamlamalıdır. Sonuç olarak ise boyutu 4 olan bir sonuç dizisi

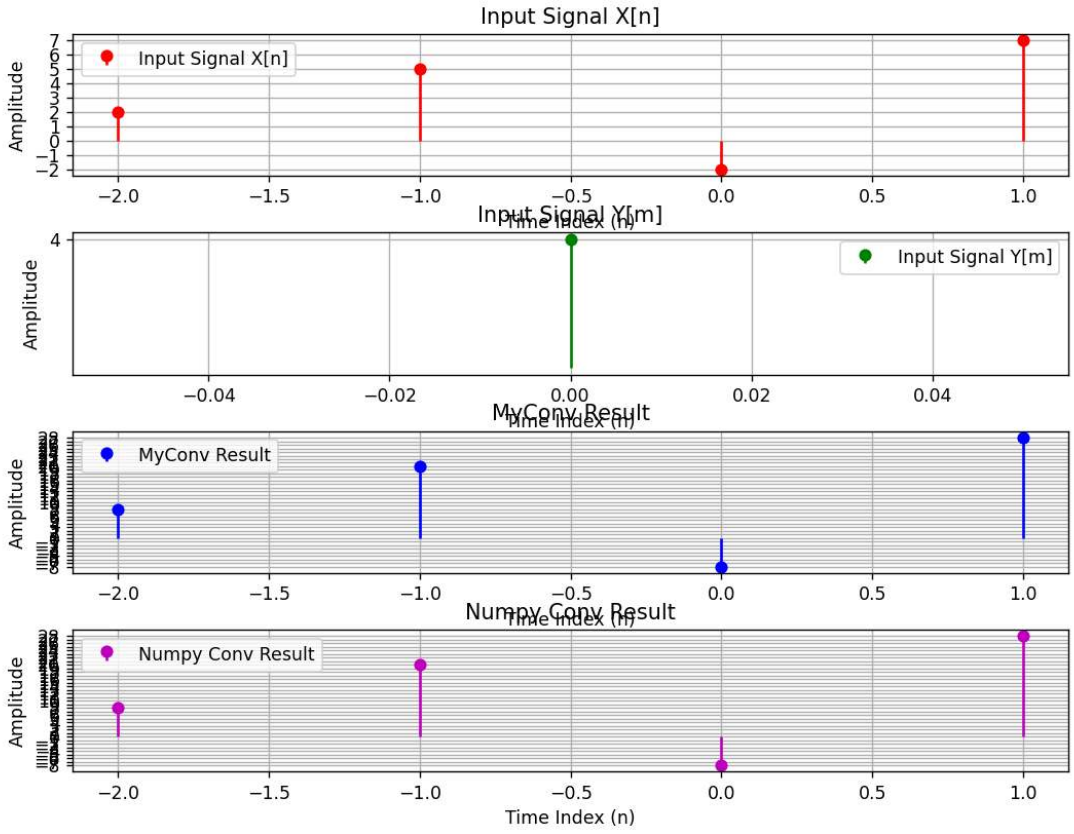
oluşmaktadır. Hazır konvolüsyon fonksiyonunu kullanmak için x ve y dizileri numpy dizilerine dönüştürülmüştür.

1. Örnek:

X = [2 5 **-2** 7] Y = [**4**] x_zero = 2 y_zero = 0

```
PS C:\Users\mehme> python -u "c:\Users\mehme\Masaüstü\21011090.py"
İlk diziyi girin (örn: 1 2 3): 2 5 -2 7
İkinci diziyi girin (maksimum 1 eleman): 4
x_zero: 2
y_zero: 0
------------------
X--> [ 2  5 -2  7]
Y--> [4]
MyConv Sonuc:  [8, 20, -8, 28]
Hazır Konvolusyon Sonuc [ 8 20 -8 28]
PS C:\Users\mehme>
```

Verilen girdiler sonucunda myConv ve hazır konvolüsyon sonucunda değerler hesaplanmış ve 4 sinyal de vektörel olarak ekrana basılmıştır. Çıktı sonucunda hazır fonksiyon ve myConv' un aynı çıktıyı verdikleri görülmüştür.



Grafiksel gösterimde x ve y vektörleri üst iki grafikte doğru bir şekilde gösterilmiştir. X vektörünün 0 noktasında -2 değeri bulunmakta ve diğer değerlerde sağında ve solunda doğru yerlerde gösterilmiştir.

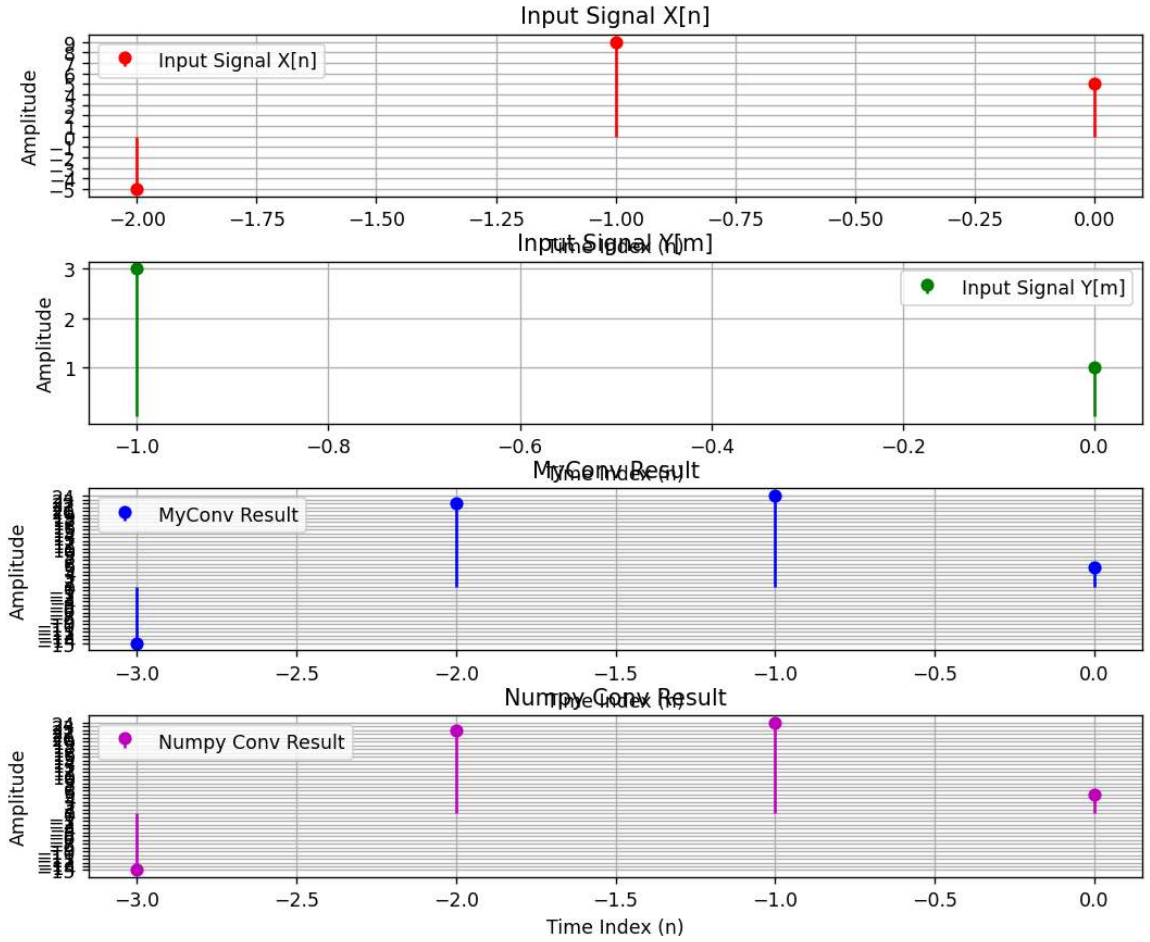
Alttaki iki grafikte de iki vektörün sıfırındaki elemanlar çarpılıp toplanınca oluşan değer sıfır değerine denk gelmektedir. Bu örnekte -8 değeridir, ve o da grafikte olması gereken yerde gösterilmiştir.

2. Örnek:

X = [-5 -9 **5**] Y = [3 **1**] x_zero = 2 y_zero = 1

Bu örnekte de girdiler sonucunda myConv ve hazır konvolüsyon sonucunda değerler hesaplanmış ve 4 sinyal de vektörel olarak ekrana basılmıştır. Çıktı sonucunda hazır fonksiyon ve myConv' un aynı çıktıyı verdikleri görülmüştür.

```
PS C:\Users\mehme> python -u "c:\Users\mehme\Masaüstü\21011090.py"
İlk diziyi girin (örn: 1 2 3): -5 9 5
İkinci diziyi girin (maksimum 2 eleman): 3 1
x_zero: 2
y_zero: 1
------------------
X--> [-5  9  5]
Y--> [3 1]
MyConv Sonuc:  [-15, 22, 24, 5]
Hazır Konvolusyon Sonuc [-15  22  24   5]
PS C:\Users\mehme>
```



Benzer şekilde bu örnekte de gösterimde x ve y vektörleri üst iki grafikte doğru bir şekilde gösterilmiştir. Y vektörünün 0 noktasında 1 değeri bulunmakta ve diğer değerlerde sağında ve solunda doğru yerlerde gösterilmiştir. Alttaki iki grafikte de iki vektörün sıfırındaki elemanlar çarpılıp toplanınca oluşan değer sıfır değerine denk gelmektedir. Bu örnekte 5 değeridir, ve o da grafikte olması gereken yerde gösterilmiştir.

3.Soruda istenen süre boyunca ses kaydı yapılması istenmiştir.

```
def record_audio(fs, duration):
    print("Kayıt başlıyor...")
    recording = sd.rec(int(duration * fs), samplerate=fs,
channels=1, dtype="float64")
```

```
    sd.wait()
    print("Kayıt bitti.")
    return recording
```

fs: Örnekleme frekansı

duration: Kaydedilmek istenen süre

4.Soru

Ödevin son kısmında kaydedilen 5 ve 10 saniyelik sesler üzerinde değişen M değerlerine (3, 4, 5) göre myConv ve hazır konvolüsyon fonksiyonu ile çıktıların elde edilmesi ve aralarındaki farkların yorumlanması istenmiş.

$y[n] = x[n] + \sum A\text{^-}k \; . \; k. \; x[n - 3000. \; k]$ A=2, Sigma aralık 1…M

Verilen denkleme ve alınan çıktılara göre bu fonksiyonun bir yankı (echo) fonksiyonu olduğu görülmektedir. A değeri zayıflatma faktörü, M değeri ise toplam yankı sayısını belirtmektedir. 3000k her yankının gecikme süresini verir. M değeri arttıkça ses kaydındaki yankı sayısının arttığı ve seslerin zayıfladığı görülmüştür. M=3, M=4 ve M=5 için sonuçlar karşılaştırıldığında, M arttıkça ses dosyasının süresi ve yankılı yapısı artmaktadır. myConv ve hazır fonksiyon arasında ise çıktı olarak fark bulunmamakla beraber çalışma süreleri arasında ciddi fark vardır. Hazır fonksiyon 2 saniye içinde sonuca ulaşırken myConv 5 saniyelik ses kaydını yaklaşık 5 dakikada sonuçlanmaktadır.

NOT: Yazılan kodda bütün soruların çıktıları aynı anda oluşmaması için sadece ilk soru açık bırakılıp diğer sorular yorum satırı olarak bırakılmıştır. Her sorunun numarası yorum satırında belirtilmiş ve kodu diğer soruya kadar olan kısma koyulmuştur.