



**Bilgisayar Mühendisleri için Sinyaller ve Sistemler**  
**BLM2041 Grup 2 - 1. Ödevi**

**Ayrık Zamanlı Sistemlerde Konvolüsyon İşlemi**

**Öğrenci Adı:** Sinem Sarak

**Öğrenci Numarası:** 22011647

**Dersin Öğretmeni:** Ahmet Elbir

## İçindekiler

<b>1. Ödevin ve Çözümün Genel Tanıtı.....</b>	<b>3</b>
<b>1.1. Ödevin Genel Tanıtı .....</b>	<b>3</b>
<b>1.2. Çözümün Genel Tanıtımı.....</b>	<b>3</b>
<b>2. Soruların Ayrıntılı Olarak İncelenmesi ve Çözüm Fonksiyonları .....</b>	<b>4</b>
<b>2.1 Soru 1 ve 2.....</b>	<b>4</b>
<b>2.2 Soru 3 ve 4.....</b>	<b>6</b>
<b>3. Sonuçların İncelenmesi ve Yorumlanması.....</b>	<b>7</b>

# 1. Ödevin ve Çözümün Genel Tanıtı

## 1.1. Ödevin Genel Tanıtı

Verilen ödevin amacı, ayırık zamanlı işaretlerin konvolüsyon işlemini anlamak ve uygulamak üzerine odaklanmaktadır. Ödev sonucunda ayırık zamanlı işaretlerin konvolüsyonunu hesaplayabilen bir algoritma geliştirilmesi ve bu algoritmanın belirli veri setleri üzerinde test edilmesi beklenmektedir.

Ödevin ilk aşamasında kullanıcıdan alınan iki ayırık zamanlı işaretin boyutları ve değerleri alınarak konvolüsyon toplamını hesaplamak için bir algoritma geliştirilmelidir (MyConv fonksiyonu). İkinci kısımda ise yazılan bu algoritmanın kullanılan dildeki hazır fonksiyonlarla karşılaştırılması ve bu karşılaştırmanın grafiksel olarak gösterilmesi gerekmektedir. Bu karşılaştırma, grafiksel ve vektörel olarak gerçekleştirilmelidir.

Son adımda ise 5 ve 10 saniyelik ses kayıtları oluşturularak verilmiş olan denklemlerle işlenmelidir. Bu işlem  $M$  değeri 3,4 ve 5 alınarak üçer kez tekrarlanmalıdır. En sonunda elde edilmiş olan işlenmiş sesler ve kaydedilmiş sesler bir önceki soruda hazırlanmış olan MyConv ve dilin hazır konvolüsyon fonksiyonlarına ayrı ayrı gönderilir ve elde edilen çıktılar analiz edilmelidir.

## 1.2. Çözümün Genel Tanıtımı

Ödevin çözümü olarak sunulan kodda İlk olarak, kullanıcıdan alınan ayırık zamanlı işaretlerin boyutları ve değerleri, ardından da işaretlerin sıfır noktaları bilgilerini alarak bu işaretlerin konvolüsyonunu hesaplamak için bir fonksiyon tanımlanmıştır. (Fonksiyonlar raporun ilerleyen kısımlarında ayrıntılı bir şekilde tanıtılacaktır.)

Ardından, kullanıcıya önceden tanımlanmış test işaretleri üzerinde MyConv fonksiyonunu denemesi için bir test fonksiyonu sunulmuştur. Bu fonksiyon, örnek veri setleri üzerinde hem MyConv fonksiyonun sonucunu hem de numpy kütüphanesinin sağladığı hazır konvolüsyon fonksiyonunun sonucunu karşılaştırarak doğruluğunu test etmektedir. Kullanıcı aynı zamanda bir alternatif olarak kendi değerlerini kullanarak da test yapabilmektedir.

Sonrasında, kullanıcıya ses kaydetme ve kaydedilen ses dosyasını işleme seçenekleri sunulmuştur. Ses kaydı yapmak için sounddevice kütüphanesi kullanılmış ve kaydedilen ses .wav uzantılı bir olarak bilgisayara kaydedilmiştir. Daha sonrasında kullanıcıdan  $M$  değeri alınarak, ses dosyası verilen formülle ve ilgili  $M$  değeriyle işlenerek yeni bir ses sinyali elde edilmiştir. Elde edilen bu ses sinyali önce numpy kütüphanesine ait convolve fonksiyonuna daha sonrasında da MyConv fonksiyonuna alınarak konvole edilmiş ve elde edilen ses sinyalleri seslendirilerek ses uzunluğu ve  $M$  değeri gibi farklı değişkenlerin konvolüsyon fonksiyonlarının çalışması üzerine etkileri gözlemlenmiştir. Buna ek olarak girilen  $M$  değerinin sese etkisi gözlemlenmiştir. Gözlem sonuçlarına raporun ileriki kısımlarında yer verilmiştir.

Son olarak, kullanıcının seçtiği işlemi gerçekleştirmek için bir menü oluşturulmuştur. Kullanıcıya önceden tanımlanmış seçenekler arasından seçim yapma imkanı sunulmuş ve seçilen işleme göre ilgili fonksiyonlar çağırılmıştır.

## 2. Soruların Ayrıntılı Olarak İncelenmesi ve Çözüm Fonksiyonları

### 2.1 Soru 1 ve 2

Bu soruya çözüm olarak iki ayrı zamanlı işaretin konvolüsyon toplamını hesaplamak için MyConv fonksiyonu tasarlanmıştır. İşleyişine ait detaylar kısaca şu şekildedir:

1. MyConv fonksiyonu, x ve y adında iki ayrı zamanlı işaret ve bunların boyutlarını (m ve n) ve her iki işaretin sıfır noktalarını (x\_zero ve y\_zero) parametre olarak alır.
2. Fonksiyon, konvolüsyon toplamını depolamak için bir dizi olan convolution'ı, x ve y işaretlerinin boyutlarına göre sıfırlarla başlatır. Bu dizi, konvolüsyon sonucunu tutacak şekilde boyutlandırılır.
3. İki for döngüsü iç içe kullanılarak, her bir x örneği ile her bir y örneği arasında tüm kombinasyonlar üzerinde dolaşılır. İç içe döngüler, x ve y işaretlerinin tüm elemanlarının birbirleriyle çarpılarak konvolüsyon toplamının hesaplanmasını sağlar.
4. Her bir çarpım sonucu, convolution dizisinin uygun indisine eklenir. Örneğin,  $x[i] * y[j]$  çarpımı,  $convolution[i+j]$  indisine eklenir. Bu, konvolüsyonun doğru şekilde hesaplanmasını sağlar.
5. Son olarak, konvolüsyon toplamının sıfır noktası hesaplanır ve bu bilgi, hesaplanan konvolüsyon toplamı ile birlikte dönüş değeri olarak verilir.

Bu fonksiyondan elde edilen sonuçlar “drawConvs” fonksiyonu ile görselleştirilir. Fonksiyonun işlevi kısaca şu şekildedir:

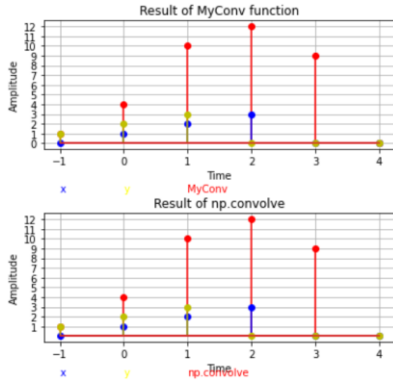
1. drawConvs fonksiyonu, altı adet parametre alır: MyConvResult, zero\_point, BuiltinFuncResult, x, y, x\_zero, ve y\_zero. Bu parametreler sırasıyla, MyConv konvolüsyon fonksiyonunun çıktısı, konvolüsyon sonucu elde edilen sıfır noktası, numpy'nin hazır konvolüsyon fonksiyonunun çıktısı, girdi olarak verilen x ve y işaretleri, x işaretinin sıfır noktası ve y işaretinin sıfır noktasıdır.
2. axis adında boş bir liste oluşturulur ve bu liste, x eksenindeki zaman değerlerini içerecektir. Sıfır noktasından başlayarak, MyConvResult'in uzunluğu kadar değer eklenir.
3. Daha sonra, x ve y işaretlerinin sıfır noktalarına göre dizilimini ayarlamak için for döngüleri kullanılır. x\_zero kadar 0 değeri x işaretinin başına eklenir ve kalan boşluklar 0 ile doldurulur. Aynı işlem y işareti için de yapılır. Bu işlemin amacı noncasual sinyallerin grafik üzerinde yanlış gösterimini önlemek ve sıfırdan küçük değerlere hizalayabilmektir.
4. plt.subplot(2, 1, 1) ile bir alt grafik alanı oluşturulur ve bu alana MyConvResult, x ve y işaretleri çizilir. plt.stem fonksiyonu kullanılarak işaretlerin değerleri belirli zaman dilimlerinde işaretlenir. plt.text fonksiyonu ile x, y ve MyConv işaretlerinin etiketleri belirli konumlara yerleştirilir.
5. plt.subplot(2, 1, 2) ile bir üst grafik alanı oluşturulur ve bu alana BuiltinFuncResult, x ve y işaretleri çizilir. Benzer şekilde, işaretlerin değerleri belirli zaman dilimlerinde işaretlenir ve etiketler konumlandırılır.
6. plt.subplots\_adjust fonksiyonu ile grafiklerin konumları ve aralarındaki boşluk ayarlanır.
7. Son olarak, plt.show() fonksiyonu ile grafikler ekranda gösterilir.

Bu fonksiyonları test etmek için kullanıcıya kendi data setini kullanma veya hazır örnek setleri kullanma imkanı sunulur. Kullanıcı kendi data setini girmek isterse “takeSignal” fonksiyonu çağırılır. Bu fonksiyonun detayları şu şekildedir:

1. takeSignal fonksiyonu, her biri boş bir liste olan x ve y değişkenlerini tanımlar. Bu değişkenler, kullanıcı tarafından girilecek x ve y işaretlerini depolamak için kullanılacaktır.
2. Kullanıcıdan n ve m uzunlukları alınır. Bu değerler, x ve y işaretlerinin uzunluklarını temsil eder.
3. for döngüsü kullanarak, kullanıcıdan x işaretinin elemanlarını girmesi istenir ve bu elemanlar x listesine eklenir.
4. Benzer şekilde, for döngüsü kullanarak, kullanıcıdan y işaretinin elemanlarını girmesi istenir ve bu elemanlar y listesine eklenir.
5. Kullanıcıdan x ve y işaretlerinin sıfır noktalarını girmesi istenir ve bu değerler x\_zero ve y\_zero değişkenlerine atanır.
6. MyConv fonksiyonu çağrılarak, x, y, m, n, x\_zero ve y\_zero parametreleri ile konvolüsyon işlemi gerçekleştirilir. MyConv fonksiyonunun çıktısı, MyConvResult ve zero\_point değişkenlerine atanır.
7. Kullanıcının girdiği x ve y işaretleri ile konvolüsyon sonucu ve sıfır noktası ekrana yazdırılır.
8. Ayrıca, numpy kütüphanesinin hazır konvolüsyon fonksiyonu olan np.convolve fonksiyonu da aynı x ve y işaretlerine uygulanarak elde edilen sonuçlar ekrana yazdırılır.
9. Son olarak drawConvs fonksiyonu ile oluşan sonuçlar grafiğe aktarılarak kullanıcıya gösterilir.

Kullanıcı aynı zamanda hazır data seti kullanarak da fonksiyonu test edebilir. Bu durumda “testSignal” fonksiyonu kullanılır. Bu fonksiyonun detayları şu şekildedir:

1. Fonksiyon içinde birinci test için kullanılacak x ve y işaretlerinin uzunlukları belirlenir.
2. Daha sonra, örnek x ve y işaretleri belirlenir.
3. x ve y işaretlerinin sıfır noktaları x\_zero ve y\_zero olarak belirlenir.
4. MyConvResult değişkeni, konvolüsyon sonucunu depolamak için kullanılan bir liste oluşturulur. Bu liste, MyConv fonksiyonunun çıktısı olarak elde edilecek konvolüsyon sonucunu tutar. Başlangıçta, bu liste sıfırlarla doldurulur.
5. np.convolve fonksiyonu kullanılarak, hazır konvolüsyon işlemi gerçekleştirilir ve sonuç BuiltinFuncResult değişkenine atanır.
6. Kullanıcının belirlediği x ve y işaretleri, MyConv fonksiyonuna parametre olarak verilerek konvolüsyon işlemi gerçekleştirilir. Bu işlem sonucunda, MyConvResult ve sıfır noktası zero\_point değişkenlerine atanır.
7. Sonuçlar ekrana yazdırılır. Kullanıcıya işaretler ve konvolüsyon sonuçları hakkında bilgi verilir.
8. MyConv sonucu ve sıfır noktası, BuiltinFunc sonucu eşliğinde görüntülenir.
9. drawConvs fonksiyonu çağrılarak, çizim işlemi gerçekleştirilir. Bu fonksiyon, MyConv sonucu, BuiltinFunc sonucu ve işaretlerin grafiğini çizer.
10. Daha sonrasında aynı adımlar farklı değerler kullanılarak tekrar edilir ve ikinci test gerçekleştirilir. Bu fonksiyonun çıktısında oluşan grafikler ve ekran çıktıları aşağıda verilmiştir:



Grafik 1: Birinci data setinin grafik çıktıları

```

Signals:
x: [1, 2, 3]
y: [1, 2, 3]

MyConv result:
[ 1.  4. 10. 12.  9.  0.]
Zero point is: convolution[0] = 4.0

Built in function result:
[ 1  4 10 12  9]

```

Resim 1: Birinci data setinin vektör çıktıları

```

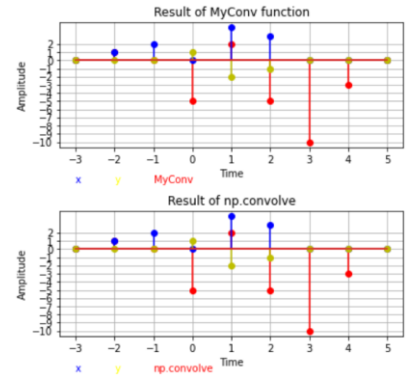
Signals:
x: [1, 2, 0, 4, 3]
y: [0, 1, -2, -1]

MyConv result:
[ 0.  1.  0. -5.  2. -5. -10. -3.  0.]
Zero point is: convolution[0] = -5.0

Built in function result:
[ 0  1  0 -5  2 -5 -10 -3]

```

Resim 2: İkinci data setinin vektör çıktıları



Grafik 2: İkinci data setinin grafik çıktıları

## 2.2 Soru 3 ve 4

Bu kısımda kullanıcının kendi sesini kaydetmesine veya daha önce kaydedilmiş bir ses dosyasını girmesine imkan verilmiştir. Kullanıcı kendi sesini kaydetmek isterse “recordVoice” fonksiyonu çağırılır. Bu fonksiyonun detayları şu şekildedir:

1. Fonksiyon seconds adında bir parametre alır. Bu parametre, kaydedilecek sesin süresini belirtir ve kullanıcıdan alınmaktadır.
2. Kayıt işlemi için örnekleme frekansı (fs) 8000 Hz olarak belirlenir. Bu, saniyede 8000 örneğin alınacağı anlamına gelir.
3. Kullanıcıya kaydın başladığı bildirilir ve "Start speaking..." mesajı ekrana yazdırılır, kayıt başlatılır.
4. sd.rec fonksiyonu kullanılarak ses kaydı yapılır. Bu fonksiyonun parametreleri şunlardır:
5. int(seconds \* fs): Kaydedilecek örneğin toplam sayısı hesaplanır. Saniye cinsinden verilen süre, örnekleme frekansı ile çarpılarak örnek sayısına dönüştürülür. samplerate=fs: Örnekleme frekansı belirtilir. channels=1: Tek bir kanal kullanılacağı belirtilir. dtype='int16': Kaydedilecek verinin veri tipi belirlenir. sd.wait() fonksiyonu kullanılarak kayıt işleminin tamamlanması beklenir.
6. Kayıt tamamlandığında "End of Recording." mesajı ekrana yazdırılır.
7. Kaydedilen ses verisi, "recording.wav" dosyasına yazılır. Bu işlem için write fonksiyonu kullanılır. Dosya formatı ve örnekleme frekansı belirtilir.

Kullanıcı ayrıca daha önce kaydedilmiş bir ses dosyası da yükleyebilmektedir. Bu kaydı yüklemek ve işlenebilir bir dizi haline getirmek için “openRecord” fonksiyonu kullanılır. Bu fonksiyonun detayları şu şekildedir:

1. Fonksiyonun tek bir parametresi vardır: name. Bu parametre, okunacak ses dosyasının adını belirtir ve ana menü içerisinde kullanıcıdan alınarak gönderilir.
2. Fonksiyonun içinde bir try-except bloğu kullanılmıştır. Kullanıcının var olmayan bir dosya ile işlem yapmaya çalışması gibi olası hataları kontrol etmek için kullanılır. Eğer dosya bulunamazsa, FileNotFoundError yakalanır ve kullanıcıya uygun bir hata mesajı gösterilir ve program -1 ile çıkış yapar (sys.exit(-1)).
3. with ifadesi kullanılarak dosya açılır. wave.open(name, 'rb') ile belirtilen dosya adı ve dosya modu ('rb' - binary okuma) ile dosya açılır.

4. Dosya açıldıktan sonra, getsampwidth metodu kullanılarak ses örnekleme genişliği (sample\_width) elde edilir.
5. Ses dosyasındaki tüm çerçeveler (frames) readframes metodu ile okunur. Bu çerçeveler, ses verisini içerir.
6. Ardından, ses dosyasının örnekleme genişliği kontrol edilir. Eğer sample\_width 1 ise, ses verileri np.uint8 türünde okunur; eğer 2 ise np.int16 türünde okunur. Diğer genişlikler desteklenmezse, ValueError ile bir hata fırlatılır.
7. Ses verileri frombuffer fonksiyonu kullanılarak bir numpy dizisine dönüştürülür. Bu işlem, ses verilerinin işlenmesini kolaylaştırır.
8. Dönüştürülen ses verileri (audio\_data), kullanıcıya dosyadaki sesin bir özetini dinletmek için playRecord fonksiyonuna gönderilir.
9. Son olarak, fonksiyon dönüş değeri olarak ses verilerini içeren audio\_data dizisini döndürür.

Ses dosyası alındıktan ve diziye döndürüldükten sonra ana menü içerisinde kullanıcıdan M değeri istenir. Alınan M değeri ve elde edilen dizi soruda verilen denklemde yerine konularak result\_of\_given\_formula dizisine aktarılır. result\_of\_given\_formula ve kullanıcıdan alınan ses sırasıyla np.convolve ve MyConv fonksiyonlarına gönderilerek sonuçları elde edilir. Konvolüsyon işlemleri sırasında süre de ölçülerek fonksiyonlar süre bakımından da kıyaslanmıştır.

### 3. Sonuçların İncelenmesi ve Yorumlanması

Bu program öncelikle 5 saniyelik bir ses kaydı alınarak M değerleri sırasıyla 3, 4 ve 5 verilerek çalıştırılmıştır. Ortaya çıkan sesler incelenmiş ve farkları analiz edilmiştir. Yapılan değerlendirmeler sonucunda soruda verilen fonksiyonun, kaydedilen sese yankı eklediği tespit edilmiştir. Girilen M değerinin artmasına bağlı olarak seste meydana gelen yankı miktarı da artmıştır. İlgili ekran çıktıları aşağıda verilmiştir:

```
Welcome to my homework, please choose an option:
Enter 1 question 1 (Try MyConv with your data)
Enter 2 question 2 (Try MyConv with my sample data)
Enter 3 for question 3 and 4
Enter 0 to exit
Please enter your desicion: 3

Please press 1 to record a voice
Please press 2 to enter a audio file name (File format must be WAV)
Please enter your desicion: 2
Please enter the name of the file (File format must be WAV): record5.wav
Here is the record in your file:
Please enter the value of M: 3
Here is the result of given formula with your file:
Here is the result of the np.convolve function:
This operation took 0.5800039768218994 seconds

Here is the result of the MyConv function:
This operation took 1506.0026156902313 seconds
```

```
Welcome to my homework, please choose an option:
Enter 1 question 1 (Try MyConv with your data)
Enter 2 question 2 (Try MyConv with my sample data)
Enter 3 for question 3 and 4
Enter 0 to exit
Please enter your desicion: 3

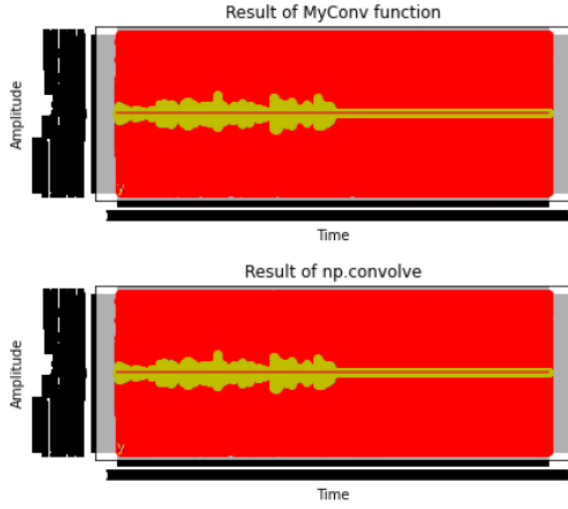
Please press 1 to record a voice
Please press 2 to enter a audio file name (File format must be WAV)
Please enter your desicion: 2
Please enter the name of the file (File format must be WAV): record5.wav
Here is the record in your file:
Please enter the value of M: 4
Here is the result of given formula with your file:
Here is the result of the np.convolve function:
This operation took 0.6281116008758545 seconds

Here is the result of the MyConv function:
This operation took 1323.0830385684967 seconds
```

```
Enter 1 question 1 (Try MyConv with your data)
Enter 2 question 2 (Try MyConv with my sample data)
Enter 3 for question 3 and 4
Enter 0 to exit
Please enter your desicion: 3

Please press 1 to record a voice
Please press 2 to enter a audio file name (File format must be WAV)
Please enter your desicion: 2
Please enter the name of the file (File format must be WAV): record5.wav
Here is the record in your file:
Please enter the value of M: 5
Here is the result of given formula with your file:
Here is the result of the np.convolve function:
This operation took 0.5979843139648438 seconds
Here is the result of the MyConv function:
This operation took 1882.366173028946 seconds
```

Süreyle ilgili yapılan analiz sonrasında M değerinin süreye kayda değer bir etkisi olmadığı saptanmıştır. Ayrıca MyConv fonksiyonu  $N^2$  gibi bir karmaşıklıkla çalıştığından ses gibi büyük bir veri setinde fazlasıyla yavaş çalışmaktadır. np.convolve ile arasındaki süre farkının önemli bir miktarda fazla olduğu fark edilmiştir.



np.convolve ve MyConv fonksiyonlarının çıktıları hem dinlenerek hem de bir sefere mahsus grafiğe dökülerek analiz edilmiştir. Ses dosyası dizisindeki eleman sayısının fazlalığı nedeniyle grafiğe çizme işlemi çok uzun sürmüştü bu durum çizim işleminin sadece bir defa yapılabilmesine neden olmuştur. Elde edilen grafik yan tarafta verilmiştir:

İki grafiğin neredeyse aynı sonuçlar içerdiği gözlenmiştir. Aynı şekilde ses kayıtlarının da birbirinden ayrılamaz şekilde benzer olduğu gözlenmiştir. Bu gözlemler dikkate alınarak MyConv fonksiyonunun doğruluğunun gerçeğe oldukça yakın olduğu söylenebilir.

Daha sonrasında aynı işlemler 10 saniyelik ses kaydı için denenmiştir. np.convolve fonksiyonunun süresinin yaklaşık 10 kat arttığı gözlenmiştir. MyConv fonksiyonu için ise 2 saate yakın beklendikten sonra bir çıktı elde edilememiştir. Sürenin uzamasından kaynaklı büyük bir gecikme yaşandığı tahmin edilmekle birlikte fonksiyonun bitiş süresi bilinmemektedir. Benzer şekilde M değerinin süreye kayda değer bir etkisi olmadığı gözlenmiştir. Süre uzunluğunun işleme süresini radikal bir şekilde etkilediği ortaya çıkmıştır.