



BİLGİSAYAR MÜHENDİSLERİ İÇİN
SİNYALLER VE SİSTEMLER

[BLM2041](#)

22011079

NESLİ ZİŞAN ÖZÇELİK

zisan.ozcelik@std.yildiz.edu.tr

İçerik

1.Kodun amacı

2.Fonksiyonlar

- myConv
- grafiksel_karsilastirma
- record_audio
- play_audio
- sistem
- impulse_response
- question_one
- question_two
- question_three
- question_four
- main

3.Ek yorum istenilen sorular için açıklama

Kodun amacı

Bu Python kodu, sinyal işleme ödevindeki 1–4. soruları tek bir çatı altında toplayarak hem elle yazılmış bir konvolüsyon algoritması (myConv) hem de NumPy’ın optimize np.convolve() fonksiyonunu karşılaştırmanıza, grafiksel sonuçları grafiksel_karsilastirma() ile JPEG’e kaydetmenize, mikrofonla 5 s ve 10 s’lik WAV kayıtları alıp (record_audio) bunları sisteme beslemenize, tanımlı fark denkleminin dürtü yanıtını (impulse_response) ve çıkış sinyalini (sistem) üretmenize ve tüm bu adımları question_one–question_four fonksiyonları üzerinden interaktif olarak seçilerek çalıştırmanıza imkân veren, tamamıyla Python ile yazılmış bir uygulamadır.

Fonksiyonlar

- **myConv**

Hazır kütüphane fonksiyonu kullanmadan

$$[n]=(x*h)[n]=\sum x[k]h[n-k]$$

bağıntısı ile iki sinyalin konvolüsyonunu hesaplar.

Parametreler:

x : Birinci sinyal (liste veya dizi)

x_zero_index : x sinyalinde fiziksel n=0'a denk gelen indeks

h : İkinci sinyal (liste veya dizi)

h_zero_index : h sinyalinde fiziksel n=0'a denk gelen indeks

Dönüş:

y : x ve h'nin konvolüsyonu sonucu elde edilen sinyal (liste)

y_zero_index : y sinyalinde fiziksel n=0'a karşılık gelen indeks

- **grafiksel_karsilastirma**

Dört sinyali ayrı alt grafikte (subplot) karşılaştırır.

Her sinyalin başlangıç indeksi dışarıdan verilir ve bu indekse göre zaman eksenini hesaplanır.

Her alt grafik üzerinde sinyale ait üst başlık (title) dışarıdan parametre olarak alınır.

Parametreler:

signal1, signal2, signal3, signal4 : Karşılaştırılacak sinyaller (liste veya numpy array)

start1, start2, start3, start4 : Her sinyalin başlangıç indeksi
(örn. sinyalin fiziksel tanım aralığında n=0'ın hangi indekste olduğunu belirtir)

title1, title2, title3, title4 : Her sinyalin başlığı

- **record_audio**

belirtilen süre boyunca ses kaydı yapar ve belirtilen dosya adına kaydeder.

parametreler :

duration: ses kaydının süresi (saniye)

filename: kaydedilecek dosya ad

- **play_audio**

Verilen dosya adındaki WAV ses dosyasını çalar.

Parametreler:

filename : Çalınacak ses dosyasının adı (string)

- **sistem**

Bu fonksiyon, giriş sinyali $x[n]$ 'ye aşağıdaki sistemi uygular:

$$y[n] = x[n] + \sum_{k=1}^M (A \cdot k) \cdot x[n - \text{delay} \cdot k]$$

Parametreler:

x: Giriş sinyali, numpy dizisi. Boyut: (örnek sayısı, kanal sayısı)

M: Toplam gecikme katmanı sayısı.

A: Katsayı parametresi.

delay: Gecikmenin birim adım değeri (varsayılan 400).

Dönüş:

y: Sistem çıkışı sinyali, $x[n]$ ile aynı boyutta, int16 formatında numpy dizisi.

- **impulse_response**

Bu fonksiyon, sistemimize impuls (delta) sinyali gönderip sistemin dürtü yanıtı $h[n]$ 'yi hesaplar.

Parametreler:

M: Gecikme katmanı sayısı.

A: Katsayı parametresi.

impulse_length: Impuls sinyalinin örnek sayısı; sistemin tüm gecikmeleri kapsayacak şekilde belirlenmelidir.

channels: Kanal sayısı (mono için 1, stereo için 2).

delay: Gecikme adımı (varsayılan 400).

Dönüş:

h: Hesaplanan dürtü yanıtı, numpy dizisi.

SORU FONKSİYONLARI

- **question_one**

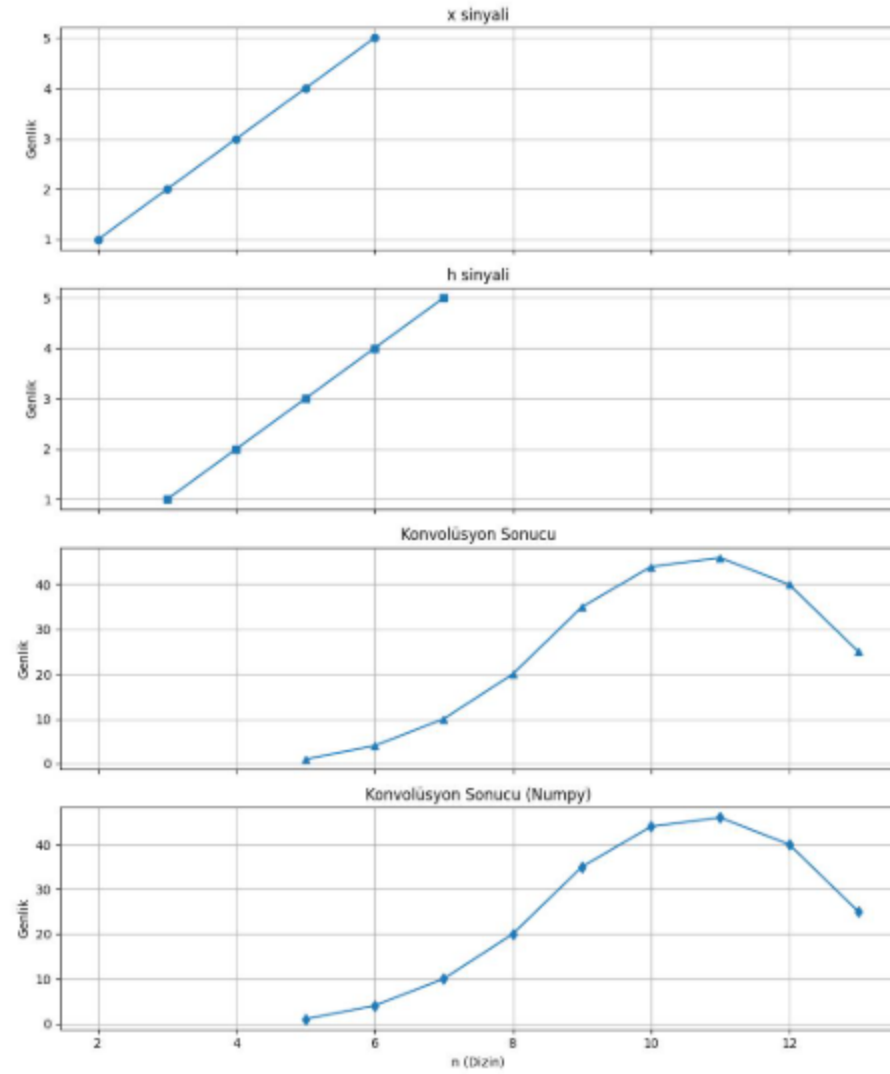
Ödev dosyasındaki 1. Soruya cevap veren sorudur. Kullanıcıdan alınan iki sinyalin konvülsiyonu hesaplanır. Ve konvülsiyon sonucu ekrana yazdırılır.

```
Seçiminizi yapın (1/2/3/4) çıkmak için q ya basın: 1
Birinci sinyali girin (boşlukla ayırın): 1 2 3
İkinci sinyali girin (boşlukla ayırın): 1 -1
Birinci sinyal için fiziksel n=0'a karşılık gelen indeksi girin: 0
İkinci sinyal için fiziksel n=0'a karşılık gelen indeksi girin: 0
Konvölüsyon işlemi başlatılıyor...
Konvölüsyon işlemi tamamlandı.
Konvölüsyon sonucu: [1.0, 1.0, 1.0, -3.0]
Konvölüsyon sonucu sinyalinin fiziksel n=0'a karşılık gelen indeksi: 0
```

- question_two

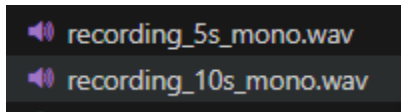
Ödev dosyasındaki 2. Soruya cevap veren sorudur. kendi yazdığım myConv fonksiyonu ile Python numpy kütüphanesinde hazır bulunan konvolüsyon toplamı fonksiyonu np.convolve() fonksiyonunun sonuçlarını karşılaştırır, vektörel olarak ekrana yazdırır ve grafiksel_karsilastirma() fonksiyonu ile grafiksel olarak karşılaştırıp ayrik_konvolusyon_karsilastirma.jpeg dosyasına kaydeder.

```
Seçiminizi yapın (1/2/3/4) çıkmak için q ya basın: 2
Birinci sinyali girin (boşlukla ayırın): 1 2 3 4 5
İkinci sinyali girin (boşlukla ayırın): 1 2 3 4 5
Birinci sinyal için fiziksel n=0'a karşılık gelen indeksi girin: 2
İkinci sinyal için fiziksel n=0'a karşılık gelen indeksi girin: 3
Konvolüsyon işlemi başlatılıyor...
Konvolüsyon işlemi tamamlandı.
Grafik 'ayrik_konvolusyon_karsilastirma.jpeg' dosyasına kaydedildi.
x sinyali: [1.0, 2.0, 3.0, 4.0, 5.0] , fiziksel n=0'a karşılık gelen indeks: 2
y sinyali: [1.0, 2.0, 3.0, 4.0, 5.0], fiziksel n=0'a karşılık gelen indeks: 3
Konvolüsyon sonucu: [1.0, 4.0, 10.0, 20.0, 35.0, 44.0, 46.0, 40.0, 25.0], fiziksel n=0'a karşılık gelen indeks: 5
Konvolüsyon sonucu (Numpy): [ 1.  4. 10. 20. 35. 44. 46. 40. 25.], fiziksel n=0'a karşılık gelen indeks: 5
```



• question_three

Ödev dosyasındaki 3. Soruya cevap veren sorudur. `record_audio()` fonksiyonu ile 5 ve 10 saniyelik ses kaydedip bunları `recording_5s_mono.wav` ve `recording_10s_mono.wav` isimleri ile kaydeder.



```
Seçiminizi yapın (1/2/3/4) çıkmak için q ya basın: 3
yeni ses kaydı yaparsanız eski kaydı kaybedeceksiniz. Devam etmek için 'Enter' tuşuna basın.

5 saniyelik ses kaydını başlatmak için 'Enter' tuşuna basın.

Recording for 5 seconds...
5 saniyelik ses kaydı tamamlandı ve 'recording_5s_mono.wav' dosyasına kaydedildi.
5 saniyelik ses kaydı yapıldı ve 'recording_5s_mono.wav' dosyasına kaydedildi.
10 saniyelik ses kaydını başlatmak için 'Enter' tuşuna basın.

Recording for 10 seconds...
10 saniyelik ses kaydı tamamlandı ve 'recording_10s_mono.wav' dosyasına kaydedildi.
10 saniyelik ses kaydı yapıldı ve 'recording_10s_mono.wav' dosyasına kaydedildi.
```

UYARI: Bu fonksiyon her çalıştığında bir önceki ses dosyalarını silip yenisini kaydeder. 4. Soru için işlenecek yeni ses kaydı son kaydedilen ses kaydı olur. Ödev dosyasında öğrencinin kendi sesini kaydetmesi gerektiği belirtilmiştir. Ödev bu şekilde teslim edilecektir ama bu fonksiyonun tekrar çalıştırılması durumunda bu kayıt silinecektir.

• question_four

Ödev dosyasındaki 4. Soru'ya cevap veren fonksiyondur. Burada amaç bir sistemin davranışını hem manuel hem de hazır araçlarla incelemek ve sistemin girdi-çıkı ilişkisindeki gecikme/echo etkisini ortaya koymaktır. Farklı M değerleri için ödev dosyasında verilen $y[n] = x[n] + \sum_{k=1}^M (A \cdot k) \cdot x[n - \text{delay} \cdot k]$ sistemini kopyalayan sistem() fonksiyonunun impulse_response ile dürtü yanıtı alınır ve 5 ve 10 saniyelik ses sinyallerimiz ile hazır ve bu ödev için yazılmış her iki konvulsiyon fonksiyonunu ile konvulsiyon işlemleri yapılır.

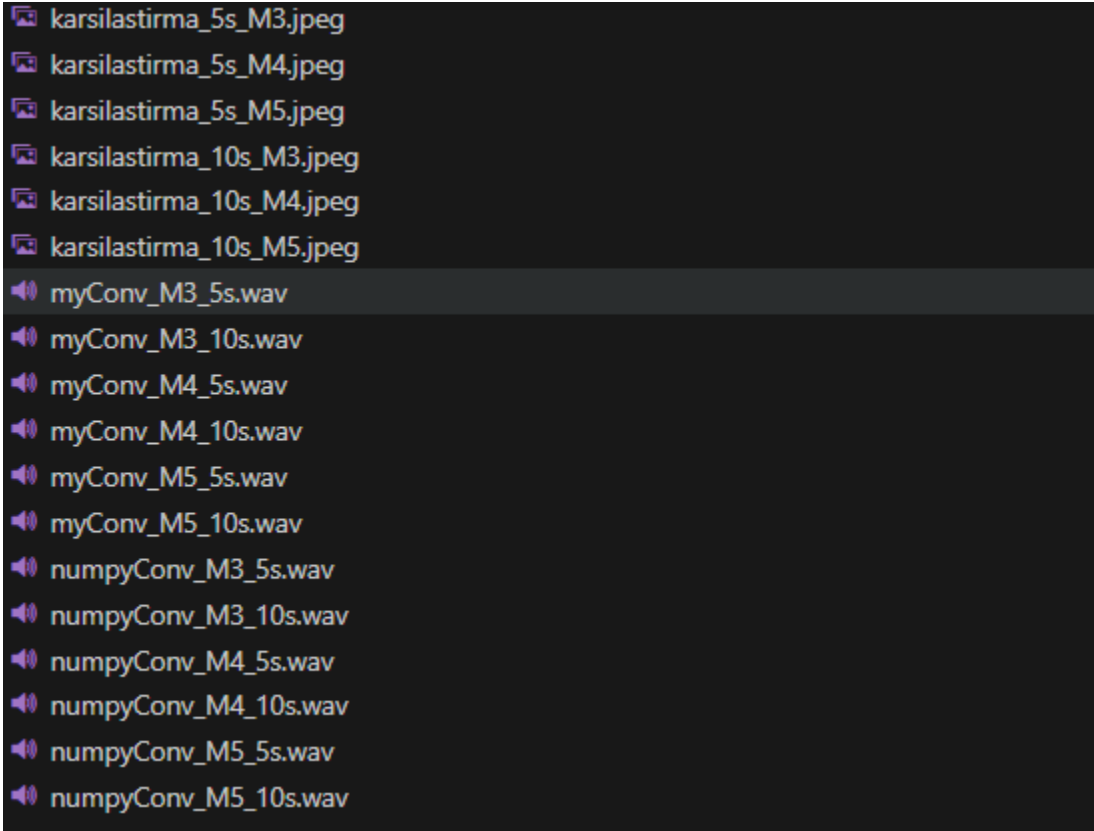
Bu işlemler neticesinde oluşan gecikmeli sinyaller her M değeri için kaydedilir ve farklı konvulsiyon fonksiyonları için oluşan sonuç sinyalleri her aşamada grafiksel_karsilastirma() fonksiyonu kullanılarak karşılaştırılır ve kaydedilir.

İşlemler bittikten sonra yeni kayıtlar play_audio() fonksiyonu kullanılarak oynatılır.

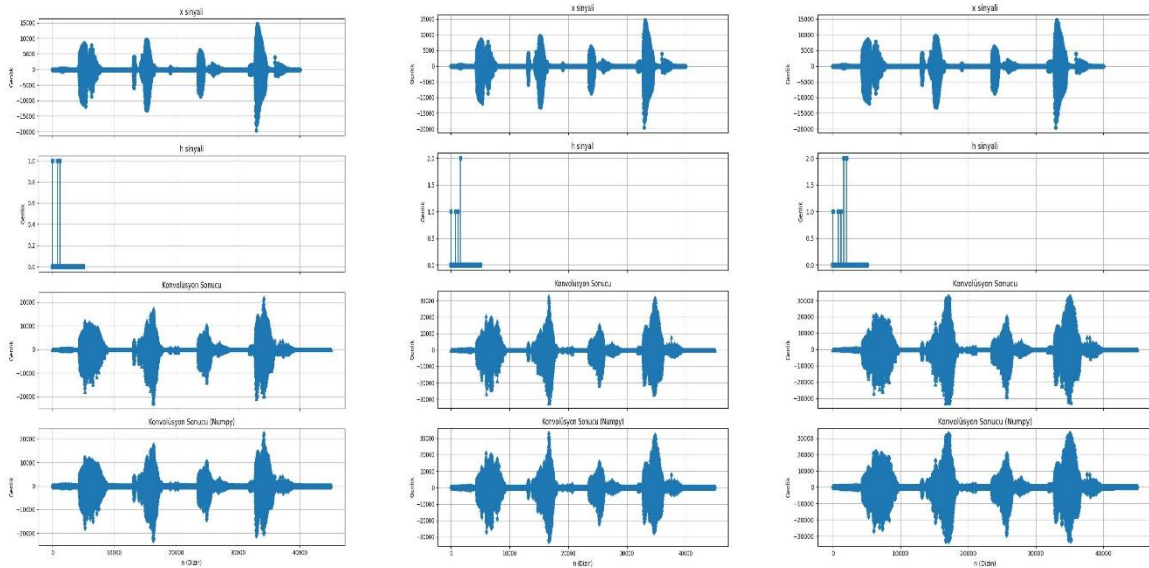
Seçiminizi yapın (1/2/3/4) çıkmak için q ya basın: 4
Konvolüsyon işlemi başlatılıyor...
Konvolüsyon işlemi tamamlandı.
5 saniyelik ses kaydı için myConv ile M=3 ile işleme yapıldı ve 'myConv_M3_5s.wav' dosyasına kaydedildi.
5 saniyelik ses kaydı için Numpy Conv ile M=3 ile işleme yapıldı ve 'numpyConv_M3_5s.wav' dosyasına kaydedildi.
Grafik 'karsilastirma_5s_M3.jpeg' dosyasına kaydedildi.
Konvolüsyon işlemi başlatılıyor...
Konvolüsyon işlemi tamamlandı.
10 saniyelik ses kaydı için myConv ile M=3 ile işleme yapıldı ve 'myConv_M3_10s.wav' dosyasına kaydedildi.
10 saniyelik ses kaydı için Numpy Conv ile M=3 ile işleme yapıldı ve 'numpyConv_M3_10s.wav' dosyasına kaydedildi.
Grafik 'karsilastirma_10s_M3.jpeg' dosyasına kaydedildi.
Konvolüsyon işlemi başlatılıyor...
Konvolüsyon işlemi tamamlandı.
5 saniyelik ses kaydı için myConv ile M=4 ile işleme yapıldı ve 'myConv_M4_5s.wav' dosyasına kaydedildi.
5 saniyelik ses kaydı için Numpy Conv ile M=4 ile işleme yapıldı ve 'numpyConv_M4_5s.wav' dosyasına kaydedildi.
Grafik 'karsilastirma_5s_M4.jpeg' dosyasına kaydedildi.
Konvolüsyon işlemi başlatılıyor...
Konvolüsyon işlemi tamamlandı.
10 saniyelik ses kaydı için myConv ile M=4 ile işleme yapıldı ve 'myConv_M4_10s.wav' dosyasına kaydedildi.
10 saniyelik ses kaydı için Numpy Conv ile M=4 ile işleme yapıldı ve 'numpyConv_M4_10s.wav' dosyasına kaydedildi.
Grafik 'karsilastirma_10s_M4.jpeg' dosyasına kaydedildi.
Konvolüsyon işlemi başlatılıyor...
Konvolüsyon işlemi tamamlandı.
5 saniyelik ses kaydı için myConv ile M=5 ile işleme yapıldı ve 'myConv_M5_5s.wav' dosyasına kaydedildi.
5 saniyelik ses kaydı için Numpy Conv ile M=5 ile işleme yapıldı ve 'numpyConv_M5_5s.wav' dosyasına kaydedildi.
Grafik 'karsilastirma_5s_M5.jpeg' dosyasına kaydedildi.
Konvolüsyon işlemi başlatılıyor...
Konvolüsyon işlemi tamamlandı.
10 saniyelik ses kaydı için myConv ile M=5 ile işleme yapıldı ve 'myConv_M5_10s.wav' dosyasına kaydedildi.
10 saniyelik ses kaydı için Numpy Conv ile M=5 ile işleme yapıldı ve 'numpyConv_M5_10s.wav' dosyasına kaydedildi.
Grafik 'karsilastirma_10s_M5.jpeg' dosyasına kaydedildi.
5 saniyelik ses kaydının tüm versiyonlarını (her biri için önce myConv sonra numpy sonuçlarını) çalmak için 'Enter' tuşuna basın.
p
'myConv_M3_5s.wav' dosyası çalınıyor...
'myConv_M3_5s.wav' dosyası çalma işlemi tamamlandı.
'numpyConv_M3_5s.wav' dosyası çalınıyor...
'numpyConv_M3_5s.wav' dosyası çalma işlemi tamamlandı.
'myConv_M4_5s.wav' dosyası çalınıyor...
'myConv_M4_5s.wav' dosyası çalma işlemi tamamlandı.
'numpyConv_M4_5s.wav' dosyası çalınıyor...
'numpyConv_M4_5s.wav' dosyası çalma işlemi tamamlandı.
'myConv_M5_5s.wav' dosyası çalınıyor...
'myConv_M5_5s.wav' dosyası çalma işlemi tamamlandı.
'numpyConv_M5_5s.wav' dosyası çalınıyor...
'numpyConv_M5_5s.wav' dosyası çalma işlemi tamamlandı.
10 saniyelik ses kaydının tüm versiyonlarını (her biri için önce myConv sonra numpy sonuçlarını) çalmak için 'Enter' tuşuna basın.
10 saniyelik ses kaydının tüm versiyonlarını (her biri için önce myConv sonra numpy sonuçlarını) çalmak için 'Enter' tuşuna basın.

'myConv_M3_10s.wav' dosyası çalınıyor...
'myConv_M3_10s.wav' dosyası çalma işlemi tamamlandı.
'numpyConv_M3_10s.wav' dosyası çalınıyor...
'numpyConv_M3_10s.wav' dosyası çalma işlemi tamamlandı.
'myConv_M4_10s.wav' dosyası çalınıyor...
'myConv_M4_10s.wav' dosyası çalma işlemi tamamlandı.
'numpyConv_M4_10s.wav' dosyası çalınıyor...
'numpyConv_M4_10s.wav' dosyası çalma işlemi tamamlandı.
'myConv_M5_10s.wav' dosyası çalınıyor...
'myConv_M5_10s.wav' dosyası çalma işlemi tamamlandı.
'numpyConv_M5_10s.wav' dosyası çalınıyor...
'numpyConv_M5_10s.wav' dosyası çalma işlemi tamamlandı.

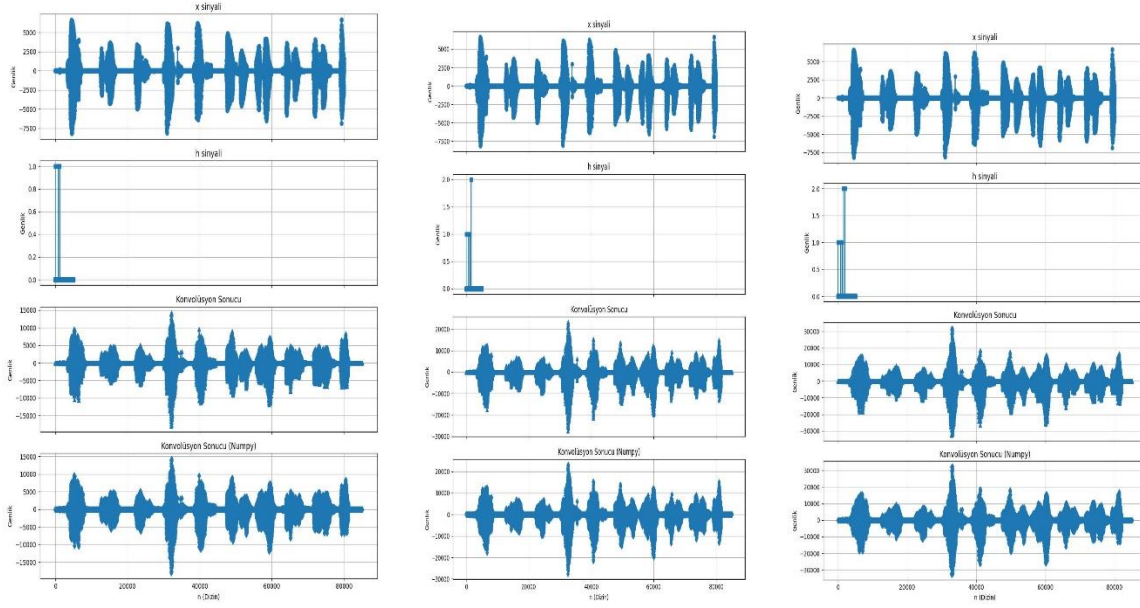
01



5 saniyelik sinyal için m[3,4,5] için degerler



10 saniyelik sinyal için $m[3,4,5]$ için degerler



- **main()**

sorular için yazılan fonksiyonların çalıştırıldığı fonksiyondur.

Sorularda istenilen yorumlar

Soru 1)

Ek yorum istenmemiştir. Bu soru için verilen cevabını görebilmek için kodu çalıştırdıktan sonra 1 seçeneğini seçebilirsiniz veya raporun fonksiyonlar kısmından question_one de verilen fonksiyonun ayrıntılı açıklaması ve çıktılarını inceleyebilirsiniz.

Soru 2)

Ek yorum istenmemiştir. Bu soru için verilen cevabını görebilmek için kodu çalıştırdıktan sonra 2 seçeneğini seçebilirsiniz veya raporun fonksiyonlar kısmından question_two da verilen fonksiyonun ayrıntılı açıklaması ve çıktılarını inceleyebilirsiniz.

Soru 3)

Ek yorum istenmemiştir. Bu soru için verilen cevabını görebilmek için kodu çalıştırdıktan sonra 3 seçeneğini seçebilirsiniz veya raporun fonksiyonlar kısmından question_three de verilen fonksiyonun ayrıntılı açıklaması ve çıktılarını inceleyebilirsiniz.

Soru 4)

Grafiklere baktığımızda önce “x sinyali”nin kaydedilmiş konuşmanın dalga formunu, yani ses patlamalarının yüksek genlikle, sessizlik anlarının ise neredeyse sıfır genlikle görüldüğünü fark ediyoruz. İkinci grafikteki “h sinyali” ise yalnızca birkaç sivri uçlu impulse’ten ibaret: her biri 400 örnek arayla ve artan ağırlık değerleriyle (0.5, 1.0, 1.5...) geliyor. Alt iki grafikte ise konvolüsyon sonrası ortaya çıkan yankı efektini izliyoruz—orijinal sinyale her impulse’in gecikmeli kopyaları eklenmiş. $M = 3$ ’te üç yankı, $M = 4$ ’te dört yankı, $M = 5$ ’te beş yankı çıkıyor; bu da sinyalin hem zaman ekseninde uzamasına hem de genlikte katmanlı bir yapı oluşmasına yol açıyor. Manuel myConv ile np.convolve sonuçlarının birebir örtüşmesi, uygulamamızın doğru çalıştığını teyit ediyor.

Ayrıca bu soru için verilen cevabını görebilmek için kodu çalıştırdıktan sonra 4 seçeneğini seçebilirsiniz veya daha fazla örnek grafik için raporun fonksiyonlar kısmından question_four de verilen fonksiyonun ayrıntılı açıklaması ve çıktılarını inceleyebilirsiniz.

Örnek grafik

