T.C.

İSTANBUL SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ

MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BIL206 – İŞARETLER VE SİSTEMLER DERSİ

ÖDEV1- SINUSOIDAL İŞARET ÜRETİMİ

İbrahim Serhat Aktaş - 210601020
Kutay Can Batur - 210601009
Mert Tosun - 210601027

ÖĞRETİM GÖREVLİSİ: Prof, Dr. Halis Altun

```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.widgets import Button
from scipy.io.wavfile import write
import pyaudio
```

Ödev yukarıdaki kütüphaneler kullanılarak Python dilinde hazırlandı.

```python
# Öğrenci numaralarına göre düzenlenmiş frekans değerleri
fr1 = 10 * 20  # Ibrahim Serhat Aktas --- 210601020
fr2 = 20 * 9   # Kutay Can Batur -------- 210601009
fr3 = 50 * 27  # Mert Tosun ------------- 210601027
```

Öğrenci numaralarına göre oluşturulmuş, ödev dosyasında ID olarak belirtilen frekans değerleri `'fr1', 'fr2' ve 'fr3'` değişkenlerine atandı.

```python
# Sinusoidal signyali oluşturma.
time = np.arange(0, 3/fr1, 1/44100)
```

'`time`'değişkeni, bir sinusoidal sinyal oluşturmak için kullanılan zaman dizisini temsil eder. Bu dizide, sinyalin örneklem noktalarının zamanları yer alır.

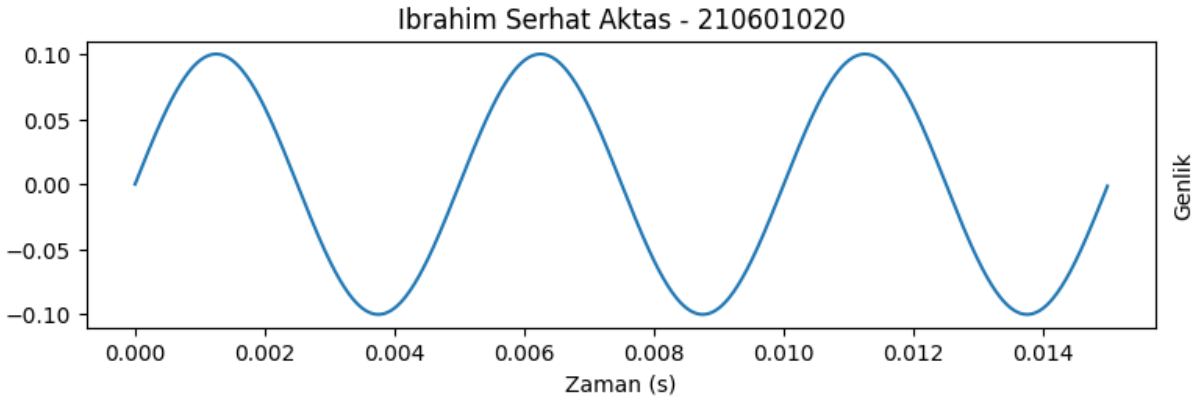
```python
# Sinusoidal signyali oluşturma.
signal1 = 0.1 * np.sin(2*np.pi*fr1*time)
signal2 = 0.1 * np.sin(2*np.pi*fr2*time)
signal3 = 0.1 * np.sin(2*np.pi*fr3*time)
signal_sum = signal1 + signal2 + signal3
```

Sinusodial sinyal değişkenleri '`signal1`','`signal2`' ve '`signal3`' şekline atandı.

```python
# Subplotları oluşturma ve sinyalleri grafiğe çizme
# 3 satır ve 2 sütuna sahip bir dizi oluşturuldu (array), ilk açılan ekranın boyutu 12,10 olarak ayarlandı
fig, axs = plt.subplots(3, 2, figsize=(12, 10))
# bu satır sayesinde nesneler arasında mesafe oluşturuldu
fig.tight_layout(h_pad=5)
# görsel düzenleme için konulmuş bir satırdır nesneleri biraz kaydırır
plt.subplots_adjust(top=0.95)
```

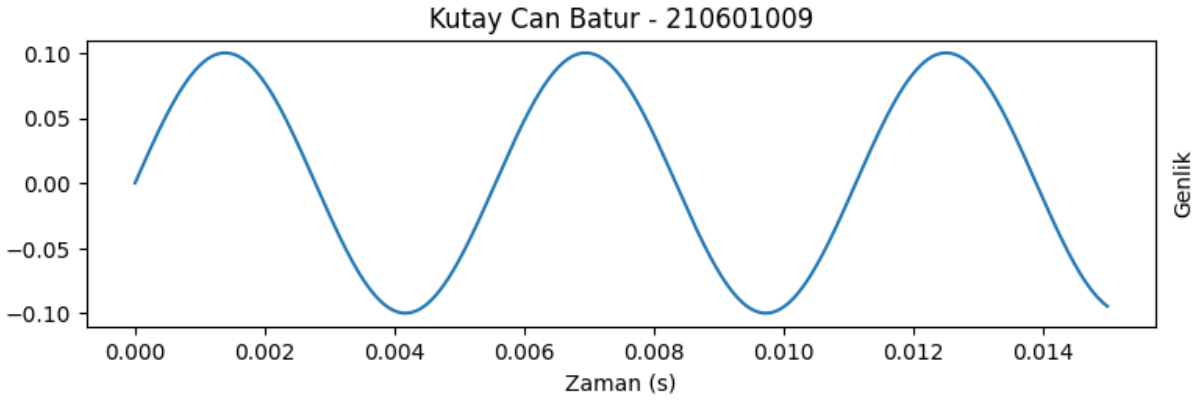
Grafik nesneleri oluşturuldu ve çeşitli düzenlemeler yapıldı.

```
#1. tablo
#210601020 numarasına uygun grafiği çizer
axs[0, 0].plot(time, signal1)
axs[0, 0].set_title('Ibrahim Serhat Aktas - 210601020')
axs[0, 0].set_xlabel('Zaman (s)')
axs[0, 0].set_ylabel('Genlik')
```



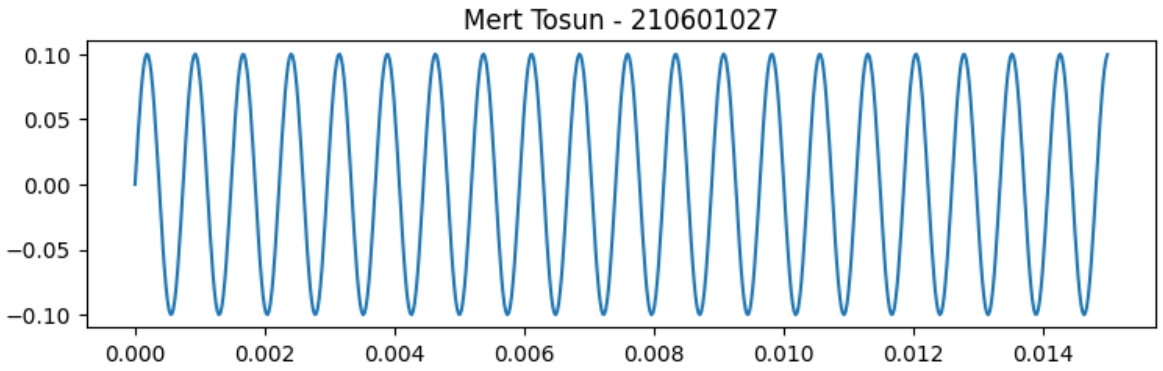
210601020 numaralı ilk tablonun oluşturma kodu ve grafik görseli.

```
#2. tablo
#210601009 numarasına uygun grafiği çizer
axs[1, 0].plot(time, signal2)
axs[1, 0].set_title('Kutay Can Batur - 210601009')
axs[1, 0].set_xlabel('Zaman (s)')
axs[1, 0].set_ylabel('Genlik')
```



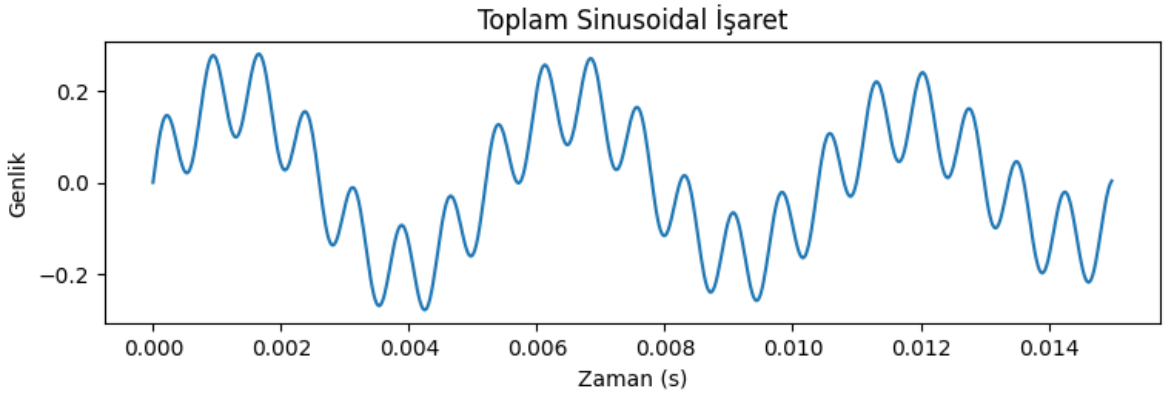
210601009 numaralı ikinci tablonun oluşturma kodu ve grafik görseli.

```python
#3. tablo
#210601027 numarasına uygun grafiği çizer
axs[2, 0].plot(time, signal3)
axs[2, 0].set_title('Mert Tosun - 210601027')
axs[2, 0].set_xlabel('Zaman (s)')
axs[2, 0].set_ylabel('Genlik')
```



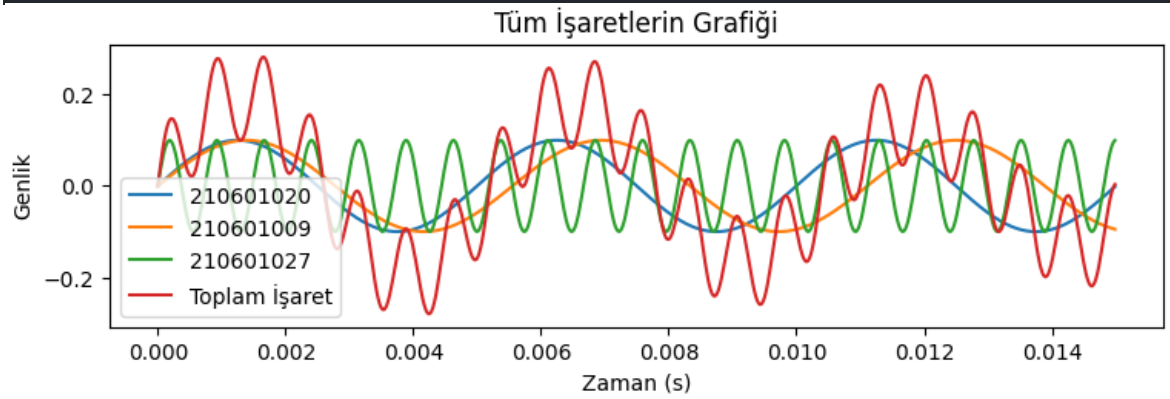
210601027 numaralı üçüncü tablonun oluşturma kodu ve grafik görseli.

```python
#4. tablo
#Toplam sinisodel işaret grafiğini çizer
axs[0, 1].plot(time, signal_sum)
axs[0, 1].set_title('Toplam Sinusoidal İşaret')
axs[0, 1].set_xlabel('Zaman (s)')
axs[0, 1].set_ylabel('Genlik')
```



Üç işaretin toplamının oluşturma kodu grafik görseli.

```
#5. 1ablo
#Tüm İşaretlerin Grafiğini çizer
axs[1,1].plot(time, signal1, label="210601020")
axs[1,1].plot(time, signal2, label="210601009")
axs[1,1].plot(time, signal3, label="210601027")
axs[1,1].plot(time, signal_sum, label="Toplam İşaret")
axs[1,1].set_title("Tüm İşaretlerin Grafiği")
axs[1,1].set_xlabel("Zaman (s)")
axs[1,1].set_ylabel("Genlik")
axs[1,1].legend()
```



Tüm sinyallerin aynı anda görülebildiği bir grafik görseli ve oluşturma kodu.

```
# Grafikleri çizerken işaretler 3 periyot uzunluğunda ayarlanmıştı.
# fakat sesleri 3 periyot uzunuğunda kullandığımızda duyması çok zorlaşıyor.
# o yüzden 3 yerine 300 periyot uzunuğunda olan yeni ses sinyalleri oluşturacağız.

sestime = np.arange(0, 300/fr1, 1/44100)
sessignal1 = 0.1 * np.sin(2*np.pi*fr1*sestime)
sessignal2 = 0.1 * np.sin(2*np.pi*fr2*sestime)
sessignal3 = 0.1 * np.sin(2*np.pi*fr3*sestime)
sessignal_sum = sessignal1 + sessignal2 + sessignal3
```

Sinyallerin seslerini çalabilmek için oluşturulan ses sinyali değişkenleri oluşturma kodu. Bu ifade, her bir zaman noktası için bir faz değeri hesaplar. Daha sonra, 0.1 ile bu sinüs dalgasının genliği ölçeklenir ve `sessignal1` adlı bir `Numpy` dizisinde depolanır.

```python
def play_sound(signal, framerate=44100):
    """Sinyalleri ses'e çeviren fonksyon"""
    p = pyaudio.PyAudio()
    stream = p.open(format=pyaudio.paFloat32,
                    channels=1,
                    rate=framerate,
                    output=True)
    stream.write(signal.astype(np.float32).tobytes())
    stream.stop_stream()
    stream.close()
    p.terminate()
```

Sinyalleri sese dönüştüren bir fonksiyon. Bu fonksiyon `PyAudio` kütüphanesi kullanarak sinyal verilerini alır ve belirtilen `framerate`'e göre oynatır. `signal` parametresi sinyal verilerini içeren bir `numpy` dizisi, `framerate` parametresi ise ses örnekleme hızını belirler ve varsayılan olarak 44100 Hz olarak ayarlanır.

```python
def button_callback(event, signal):
    """Butona tıklandığında tetiklenecek olan fonksyon"""
    play_sound(signal)
```

İstenilen grafiğin ses sinyalinin dinlenebilmesi için oluşturduğumuz butonların görev fonksiyonu. Bir sinyal verisi alır ve `play_sound`() fonksiyonunu kullanarak bu sinyal verilerini ses sinyallerine çevirir.

```python
# Buton ekleyerek ses çalma işlevselliği ekleme
button_ax = fig.add_axes([0.637, 0.135, 0.09, 0.05]) # buton konumu ve boyutu
play_button = Button(button_ax, 'Toplam İşaret', hovercolor='0.975') # buton oluşturma
play_button.on_clicked(lambda event: button_callback(event, sessignal_sum)) # butona tıklanınca tetiklenen olay

# Buton ekleyerek ses çalma işlevselliği ekleme
button_ax1 = fig.add_axes([0.54, 0.195, 0.09, 0.05]) # buton konumu ve boyutu
play_button1 = Button(button_ax1, '210601020', hovercolor='0.975') # buton oluşturma
play_button1.on_clicked(lambda event: button_callback(event, sessignal1)) # butona tıklanınca tetiklenen olay

# Buton ekleyerek ses çalma işlevselliği ekleme
button_ax2 = fig.add_axes([0.54, 0.135, 0.09, 0.05]) # buton konumu ve boyutu
play_button2 = Button(button_ax2, '210601009', hovercolor='0.975') # buton oluşturma
play_button2.on_clicked(lambda event: button_callback(event, sessignal2)) # butona tıklanınca tetiklenen olay

# Buton ekleyerek ses çalma işlevselliği ekleme
button_ax3 = fig.add_axes([0.54, 0.075, 0.09, 0.05]) # buton konumu ve boyutu
play_button3 = Button(button_ax3, '210601027', hovercolor='0.975') # buton oluşturma
play_button3.on_clicked(lambda event: button_callback(event, sessignal3)) # butona tıklanınca tetiklenen olay
```

Butonların özelliklerinin belirlendiği kod satırları.