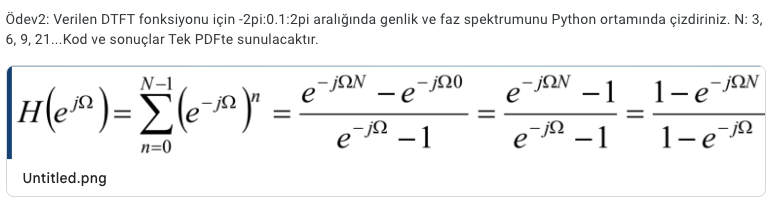
## BLM6105-2 Uzaktan Algılama Dersi Ödev 2

## 24501100 – Aleyna Nil Uzunoğlu

**Giriş**

Bu çalışmanın amacı, belirli bir Ayrık Zamanlı Fourier Dönüşümü (DTFT) ile tanımlanan bir sistemin frekans yanıtını incelemektir.



**Yöntem**

Genlik ve faz spektrumlarının hesaplanması ve görselleştirilmesi için Python programlama dili ve NumPy ile Matplotlib kütüphaneleri kullanılmıştır.

1. **Frekans Vektörü:** Verilen aralıklarda (-2π ile 2π aralığında) 0.1 radyanlık adımlarla ayrık frekans değerleri (Ω) içeren bir NumPy dizisi oluşturulmuştur
   * (np.arange(-2\*np.pi, 2\*np.pi, 0.1)) satırında bu durum görülebilir.
2. **DTFT Hesaplanması:** Belirlenen her bir N değeri (3, 6, 9, 21) için, yukarıda verilen kapalı formül kullanılarak H(e^jΩ)'nın kompleks değerleri hesaplanmıştır.
3. **Kararlılık:** Verilen formülün paydasında bulunan terim, Ω'nın 2π'nin katları olduğu durumlarda sıfır olacağından dolayı, doğrudan hesaplamada sıfıra bölme hatasına yol açabilir. Bu sorunu aşmak için, paydanın mutlak değeri çok küçük (belirlenen bir epsilon 1e-12 değerinden küçük) olduğunda, payda yerine bu küçük epsilon değeri kullanılarak kararlılık sağlanmıştır.
4. **Genlik ve Faz Hesabı:** Hesaplanan kompleks H(e^jΩ) değerlerinin her biri için:
   * Genlik spektrumu np.abs(H) fonksiyonu ile hesaplanmıştır.
   * Faz spektrumu np.angle(H) fonksiyonu ile (radyan cinsinden) hesaplanmıştır.
5. **Görselleştirme:** Her bir N değeri için genlik ve faz spektrumları ayrı Matplotlib alt grafikleri (subplots) kullanılarak çizdirilmiştir. Grafikler, frekansa (Ω) karşı genlik ve fazı göstermektedir. Grafikler ilerleyen sayfalarda paylaşılmıştır.

metin, diyagram, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Grafik 1: N = [3,6,9,21] durumları için genlik ve faz spektrumları

**Ek: main.py**

import numpy as np

import matplotlib.pyplot as plt

Omega = np.arange(-2\*np.pi, 2\*np.pi, 0.1)

N\_values = [3, 6, 9, 21]

fig, axs = plt.subplots(len(N\_values), 2, figsize=(10, 12))

fig.suptitle('DTFT H(e^{jΩ}) için Genlik ve Faz Spektrumu')

for idx, N in enumerate(N\_values):

# H(e^{jΩ}) = (1 - e^{-jΩN}) / (1 - e^{-jΩ})

numerator = 1 - np.exp(-1j \* Omega \* N)

denominator = 1 - np.exp(-1j \* Omega)

epsilon = 1e-12

denominator\_safe = np.where(np.abs(denominator) < epsilon, epsilon, denominator)

H = numerator / denominator\_safe

magnitude = np.abs(H)

phase = np.angle(H)

axs[idx, 0].plot(Omega, magnitude, label=f'N={N}')

axs[idx, 0].set\_title(f'N={N} - Genlik')

axs[idx, 0].set\_xlabel('Ω (radyan)')

axs[idx, 0].set\_ylabel('|H(e^{jΩ})|')

axs[idx, 0].grid(True)

axs[idx, 1].plot(Omega, phase, label=f'N={N}')

axs[idx, 1].set\_title(f'N={N} - Faz')

axs[idx, 1].set\_xlabel('Ω (radyan)')

axs[idx, 1].set\_ylabel('∠H(e^{jΩ}) (radyan)')

axs[idx, 1].grid(True)

plt.tight\_layout()

plt.savefig('DTFT\_H\_spectra.png')

plt.show()