ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

ELM 368 SAYISAL İŞARET İŞLEME LABORATUVARI

ÖN HAZIRLIK ÇALIŞMASI

VE

ÖDEV-3

1 AMAÇ

- Ayrık-zamanlı işaretlerin Ayrık-zamanlı Fourier Dönüşü'münün (DTFT) ve Ayrık Fourier Dönüşümü'nün (DFT) hesaplanması, genlik ve faz grafiklerinin çizdirilmesi ve yorumlanması.
- DFT hesaplanırken dikkat edilmesi gerekenlerin öğrenilmesi.
- Periyodik işaretlerin DFT ile sentezi

2 KODLAR

2.1 Ayrık-zamanlı Fourier dönüşümü (DTFT)

Ayrık zamanlı periyodik işaretlerde Fourier serisi analiz ve sentez denklemleri sırasıyla aşağıdaki gibidir.

$$a_k = \frac{1}{N} \sum_{n=<N>} \tilde{x}[n] e^{-j\frac{2\pi}{N}kn} \ (Analiz)$$

$$\tilde{x}[n] = \sum_{k=<N>} a_k e^{+j\frac{2\pi}{N}kn} \ (Sentez)$$

Eğer işaret periyodik değilse ayrık-zamanlı Fourier serisi (DFS) denklemleri aşağıda verilen ayrık-zamanlı Fourier dönüşümü (DTFT) denklemlerine dönüşür. Bu denklemlerin derivasyonlarını görmek için Oppenheim-Signals and Systems 2th Ed. Sayfa:358-361'e bakınız.

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n] e^{-j\omega n} \ (Analiz)$$

$$x[n] = \frac{1}{2\pi} \int_{<2\pi>} X(e^{j\omega}) e^{+j\omega n} d\omega \ (Sentez)$$

Ayrık zamanlı olan işaretin DTFT'si $X(e^{j\omega})$, kompleks değerli bir fonksiyondur. Bu fonksiyonun bağımsız değişkeni ω'dır ve bu değişken ayrık değil, süreklidir. $X(e^{j\omega})$ periyodiktir ve periyodu $\boldsymbol{2\pi}$'dir. Bu sebeple, $X(e^{j\omega})$'nın grafiği $[\boldsymbol{-\pi, +\pi}]$ aralığında veya $[\boldsymbol{0, 2\pi}]$ aralığında çizdirilir. Kompleks değerli bir fonksiyon olduğu için $X(e^{j\omega})$'nın ya genlik-faz grafiklerini yada gerçek-sanal kısımlarını çizdirmek gerekir.

Aşağıda verilen $x[n]$ işaretini düşünün.

$$x[n] = \delta[n] + \delta[n-1]$$

Bu işaretin DTFT'si:

$$\begin{aligned}
X(e^{j\omega}) &= \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n} \\
&= \sum_{n=-\infty}^{+\infty} (\delta[n] + \delta[n-1])e^{-j\omega n} \\
&= \sum_{n=-\infty}^{+\infty} \delta[n]e^{-j\omega} + \sum_{n=-\infty}^{+\infty} \delta[n-1]e^{-j\omega n} \\
&= 1 + e^{-j\omega} = 2e^{-j\frac{\omega}{2}} \frac{\left(e^{+j\frac{\omega}{2}} + e^{-j\frac{\omega}{2}}\right)}{2} \\
&= 2e^{-j\frac{\omega}{2}} \cos\left(\frac{\omega}{2}\right)
\end{aligned}$$

Burada, $X(e^{j\omega})$'nın genliği $|X(e^{j\omega})|$ ve fazı $\measuredangle X(e^{j\omega})$ aşağıdaki gibidir;

$$|X(e^{j\omega})| = \left|2\cos\left(\frac{\omega}{2}\right)\right|$$

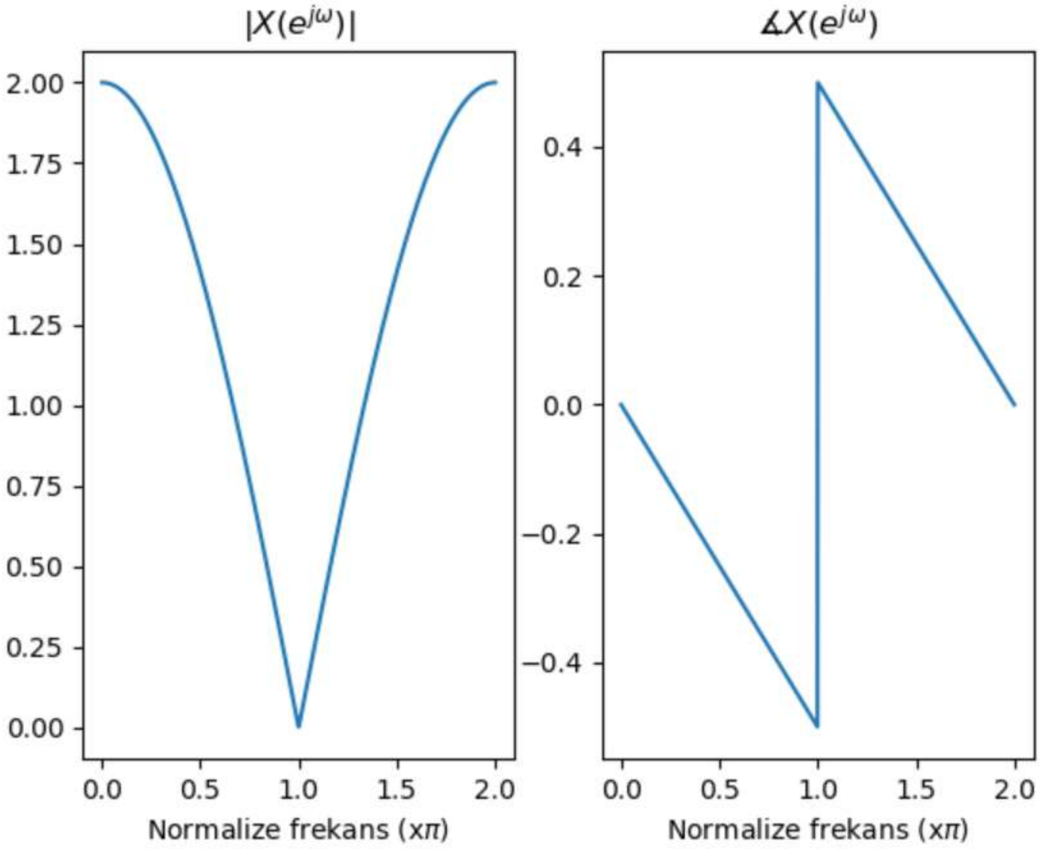
$$\measuredangle X(e^{j\omega}) = \begin{cases} -\dfrac{\omega}{2} \; eğer \; 0 \le \omega < \pi \\ -\dfrac{\omega}{2} + \pi \; eğer \; \pi \le \omega < 2\pi \end{cases}$$

Yukarıda $\measuredangle X(e^{j\omega})$'nın parçalı fonksiyon olmasının sebebi $\pi \leq \omega < 2\pi$ aralığında " $2\cos\left(\frac{\omega}{2}\right)$" fonksiyonu negatif değerler alır. Ancak, genlik hesaplarken $\left|2\cos\left(\frac{\omega}{2}\right)\right|$ hesapladığımız için daima pozitif sayı elde ederiz. Buradaki negatif değer, $e^{j\pi} = -1$ olduğundan, $\pi < \omega < 2\pi$ aralığında $+\pi$ kadar bir faz ilavesine sebep olur; diğer bir ifade ile $e^{j\pi} = -1$ ile çarpılması gerekir.

Aşağıda verilen kod parçası, yukarıda bulunan $|X(e^{j\omega})|$ ve $\measuredangle X(e^{j\omega})$ çizdirir. Bu fonksiyonların bağımsız değişeni ω sürekli olduğu için $[\mathbf{0}, +\mathbf{2}\boldsymbol{\pi}]$ aralığında örnekleyerek çizdirilebilir. ω bu aralıkta 1000 nokta gibi yüksek bir örnekleme oranı ile oluşturulup plot ile çizdirildiğinde sürekli bir fonksiyon gibi gözükür.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig

%matplotlib notebook
w_cont=np.linspace(0,2*np.pi,1000)
X_abs=np.abs(2*np.cos(w/2))
X_phase=np.array([-w/2 if w<np.pi else -w/2+np.pi for w in w_cont])
plt.subplot(121)
plt.plot(w/np.pi,X_abs)
plt.title('$|X(e^{j\omega})|$')
plt.xlabel('Normalize frekans (x$\pi$)')
plt.subplot(122)
plt.plot(w/np.pi,X_phase/np.pi)
plt.title('$ \measuredangle X(e^{j\omega})$')
plt.xlabel('Normalize frekans (x$\pi$)')
```



Yukarıdaki grafiklerde $X\left(e^{j\omega}\right)$ işaretinin genliğini ve fazını her ne kadar ω'yı ayrık noktalarla ifade edip çizdirmiş olsakta aslında $X\left(e^{j\omega}\right)$ ω'ya göre sürekli bir fonksiyondur. Bu sebeple **DTFT, dijital ortamda kullanmaya elverişli değildir**. Bilgisayar ortamında DTFT yerine, aşağıda anlatılan Ayrık Fourier Dönüşümü kullanılır.

2.2 Ayrık Fourier dönüşümü (DFT)

Ayrık Fourier dönüşümünü (DFT), ayrık zamanlı Fourier dönüşümünün örneklenmiş hali olarak düşünebilirsiniz. DFT'nin analiz ve sentez denklemleri aşağıda verildiği gibidir. $\tilde{X}[k]$, $\tilde{x}[n]$ üzerindeki tilda sembolleri periyodik olduklarını belirtmek içindir. $\boldsymbol{\tilde{X}[k]}$ **ve** $\boldsymbol{\tilde{x}[n]}$ **işaretleri** $\boldsymbol{N}$ **ile periyodiktir**.

$$\tilde{X}[k] = \sum_{n=<N>} \tilde{x}[n] e^{-j\frac{2\pi}{N}kn} \;\; (Analiz)$$

$$\tilde{x}[n] = \frac{1}{N} \sum_{k=<N>} \tilde{X}[k] e^{+j\frac{2\pi}{N}kn} \ (Sentez)$$

Bölüm 2.1 ‘de verilen $x[n] = \delta[n] + \delta[n-1]$ işaretinin $N = 8$ için DFT’sini hesaplayıp DTFT grafiğiyle aynı grafik üzerinde çizdirip aralarındaki farkı gözlemleyelim.

Aşağıda $\delta[n]$ fonksiyonu gerçeklenmiştir. Bu fonksiyonu analiz denkleminde, $\tilde{x}[n]$ kısmında, kullanacağız.

```
def dirac(n):
    if n==0:
        return 1
    else:
        return 0
```

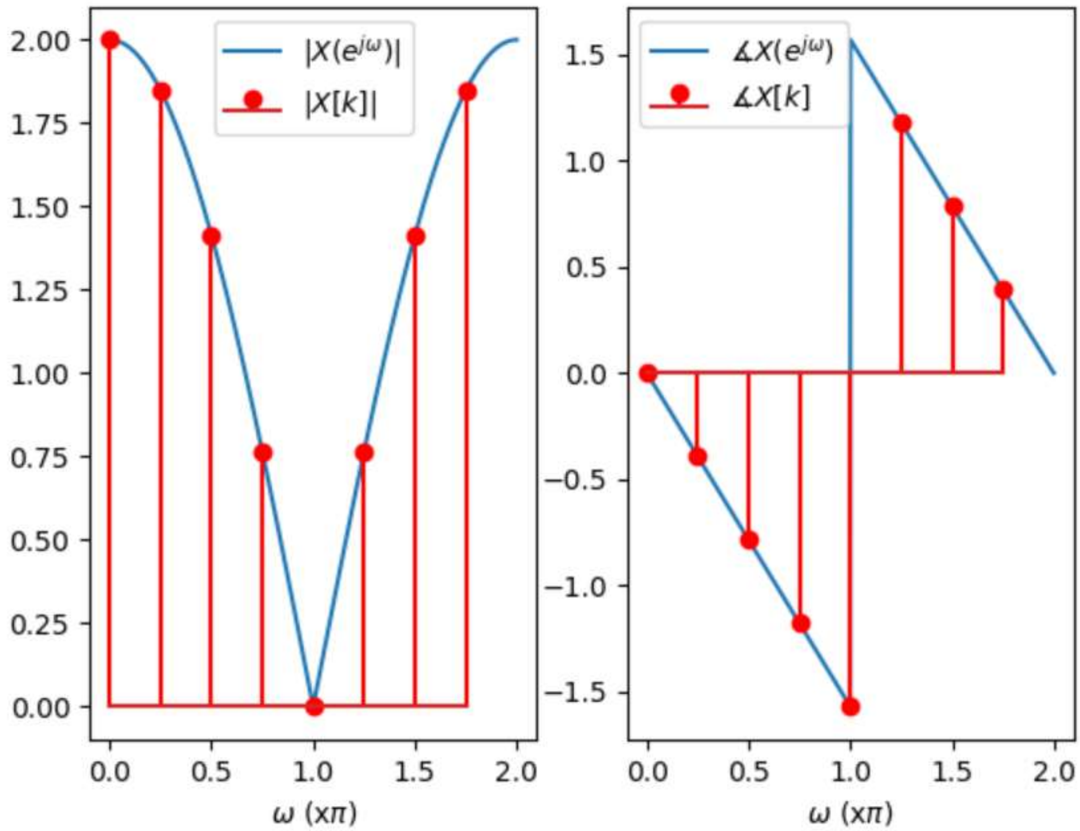
Bir kütüphane kullanmadan DFT analiz denklemini kullanarak $\tilde{X}[k]$ katsayılarını hesaplayan bir kod örneği aşağıda verilmektedir. Herhangi bir k değeri için $n = 0, \dots, N-1$ olmak üzere toplamda N defa $\tilde{x}[n]e^{-j\frac{2\pi}{N}kn}$ çarpımı yapılıp sonuçların toplanması gerekmektedir. İçteki *for* döngüsü bu işlemi yapmaktadır. $k = 0, \dots, N-1$ aralığında olduğundan toplamda N tane $\tilde{X}[k]$ hesaplayacağız. Bu sebeple, dışta da k için bir *for* döngüsü mevcuttur. Buradan N noktalı DFT hesaplamanın $O(N^2)$’lik bir zaman kompleksliği olduğunu görebiliriz.

```
N=8
X_k=np.zeros([N],dtype=complex)
for k in range(N):
    for n in range(N):
        X_k[k]=X_k[k]+(dirac(n)+dirac(n-1))*np.exp(-1j*(2*np.pi/N)*k*n)
X_k_abs=np.abs(X_k)
X_k_phase=np.angle(X_k)

plt.figure()
plt.subplot(121)
plt.plot(w_cont/np.pi,X_abs,label='$|X(e^{j\omega})|$')
w_discrete=np.arange(0,N)*(2*np.pi/N)
plt.stem(w_discrete/np.pi,X_k_abs,'r-',label='$|X[k]|$',markerfmt='ro')
```

```
plt.xlabel('$\omega$ (x$\pi$)')
plt.legend()

plt.subplot(122)
plt.plot(w_cont/np.pi,X_phase,label='$ \measuredangle X(e^{j\omega})$')
plt.stem(w_discrete/np.pi,X_k_phase,'r-',label='$ \measuredangle
X[k]$',markerfmt='ro')
plt.xlabel('$\omega$ (x$\pi$)')
plt.legend()
```



Yukarıdaki figürde solda $x[n]$ işaretinin DTFT'sinin (mavi) ve 8-noktalı DFT'sinin (kırmızı) genlik grafiği, sağda ise $x[n]$ işaretinin DTFT'sinin (mavi) ve 8-noktalı DFT'sinin (kırmızı) faz grafikleri verilmiştir. Açık bir şekilde görülüyor ki N noktalı DFT hesapladığımızda aslında kompleks DTFT işaretini $\frac{2\pi}{N}$ aralıklarla toplamda N noktadan oluşacak şekilde örnekliyoruz. $N = 4$, $N = 16$ ve $N = 32$ için aynı grafikleri tekrar çizdirip DFT ve DTFT arasındaki farkı yorumlayınız. N nokta sayısının artmasının etkilerini gözlemleyiniz. N noktalı DFT hesapladığınızda birinci, ikinci, üçüncü örneklerin, $k = 1,2,3$, hangi frekans değerlerine, ω'ya, karşı geldiğine dikkat ediniz. Bunun için bir formül önerebilir misiniz? Önereceğiniz formül ω ve k arasındaki ilişkiyi vermelidir.

Hızlı Fourier Dönüşümü (FFT)

Bölüm 2.2'de N noktalı DFT hesaplamak için N iterasyonlu olan iki tane iç içe for döngüsü içeren bir algoritma verilmişti. DFT'yi bu şekilde hesaplamanın maliyeti N^2 ile orantılıdır. N büyüdükçe hesaplama süresi hızla artar. DFT'yi hızlı hesaplamak için kullanılan Hızlı Fourier dönüşümü (FFT) algoritması ise N-noktalı DFT'yi, N eğer 2'nin bir kuvveti şeklinde ise, $NlogN$ ile orantılı bir işlem karmaşıklığına sahiptir. Bu sebeple, DFT'yi daha hızlı hesaplamak için genellikle FFT algoritması kullanılır. FFT ile ilgili daha detaylı bilgi almak için ders kitabınızın 8. Bölümüne bakınız.

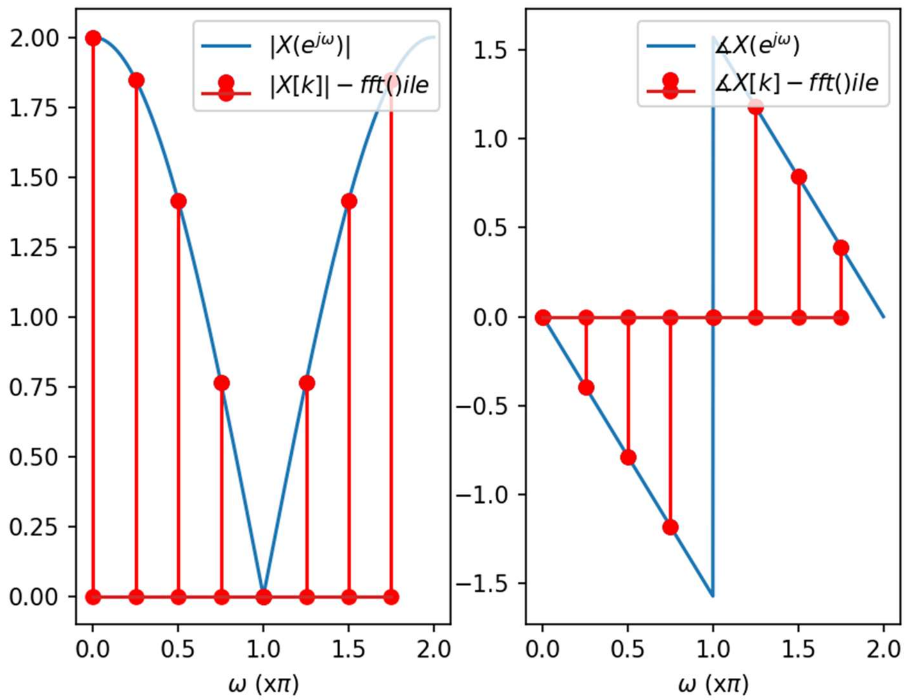
Bu derste DFT hesaplarken hep FFT algoritması kullanılacaktır. Aşağıda scipy kütüphanesinin fft modülündeki fft() komutunun nasıl kullanılacağı aşağıdaki kod parçasında gösterilmektedir. DFT'sini hesapladığımız diziyi yine $x[n] = \delta[n] + \delta[n-1]$ seçelim.

```
from scipy.fftpack import fft , ifft
x=np.array([1,1])
fft_X=fft(x,8)
abs_fft_X=np.abs(fft_X)
phase_fft_X=np.angle(fft_X)

## Grafik çizimi
plt.figure()
plt.subplot(121)
plt.plot(w_cont/np.pi,X_abs,label='$|X(e^{j\omega})|$')
w_discrete=np.arange(0,N)*(2*np.pi/N)
plt.stem(w_discrete/np.pi,abs_fft_X,'ro-',label='$|X[k]|-fft() ile$')
plt.xlabel('$\omega$ (x$\pi$)')
plt.legend(loc='upper right')

plt.subplot(122)
plt.plot(w_cont/np.pi,X_phase,label='$ \measuredangle X(e^{j\omega})$')
plt.stem(w_discrete/np.pi,phase_fft_X,'ro-',label='$ \measuredangle
X[k]-fft() ile$')
plt.xlabel('$\omega$ (x$\pi$)')
plt.legend(loc='upper right')
```

Aşağıdaki grafikten de anlaşılacağı gibi, N noktalı *fft* hesaplamak DFT analiz denklemini doğrudan kullanarak hesapladığımız noktaların aynısını üretmektedir. $fft(x, N)$ fonksiyonunda ilk parametre işareti, ikinci parametre N ise nokta sayısına karşılık gelmektedir. Burada eğer işaretin boyu N'den küçük ise algoritma işaretinizin boyunu N'e tamamlayacak şekilde sonuna sıfırlar ekler. Eğer tam tersi bir durum var ise yani işaretinizin uzunluğu N'den büyük ise, algoritma işaretinizin ilk N noktasını alacak şekilde işareti kırpar.

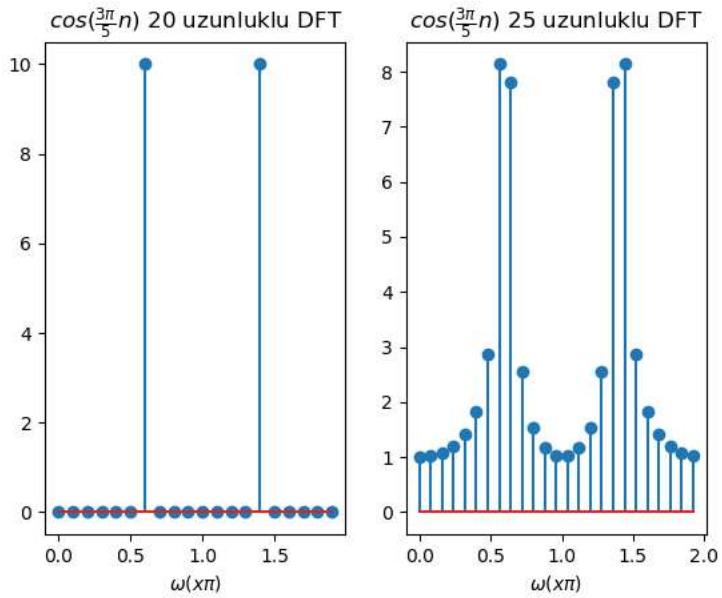


2.2.1 Frekansta örnekleme →Zamanda periyodiklik ilişkisi

Bir önceki bölümde N noktalı DFT hesaplarken hem $X[k]$'nın hemde $x[n]$'in N'ile periyodik olduklarından bahsetmiştik bu periyodikliği belirtmek için analiz ve sentez denklemlerinde $\tilde{X}[k]$, $\tilde{x}[n]$ olarak yazmıştık. Bu periyodikliğin nedeni frekans uzayında $\frac{2\pi}{N}$ aralıklı dürtü katarının ters Fourier dönüşümünün ayrık zamanda N aralıklı dürtü katarına eşit olmasıdır. Frekansta çarpım, zamanda konvolüsyona karşılık geldiği için de $x[n]$ dizisinin N ile periyodik versiyonu $\tilde{x}[n]$'i elde ederiz. Dolayısıyla, N noktalı DFT hesaplarken işaretinizin N ile periyodik olduğunu varsayarak DFT hesaplamanız gerekir. Aşağıda $\cos\left(\frac{3\pi}{5}n\right)$ işaretini 20 noktalı ve 25 noktalı iki ayrı versiyonunu oluşturup Python'da DFT'lerini fft() ile hesaplayıp sadece genliklerini çizdirelim;

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.fftpack import fft , ifft
#20 uzunluklu versiyon
n1=np.arange(0,20)
x1=np.cos(3*np.pi/5*n1)
X1_abs=np.abs(fft(x1))
w_disc_1=n1*2*np.pi/len(n1)   #0-2pi arası 2pi/20 adımlı vektör (2pi
noktası dahil değil)
plt.figure()
plt.subplot(121)
plt.stem(w_disc_1/np.pi,X1_abs)
plt.xlabel('$\omega (x\pi)$')
plt.title('$cos(\\frac{3\\pi}{5}n)$ 20 uzunluklu DFT')

#25 uzunluklu versiyon
n2=np.arange(0,25)
x2=np.cos(3*np.pi/5*n2)
X2_abs=np.abs(fft(x2))
w_disc_2=n2*2*np.pi/len(n2)   #0-2pi arası 2pi/25 adımlı vektör (2pi
noktası dahil değil)
plt.subplot(122)
plt.stem(w_disc_2/np.pi,X2_abs)
plt.xlabel('$\omega (x\pi)$')
plt.title('$cos(\\frac{3\\pi}{5}n)$ 25 uzunluklu DFT')
```



$fft(x)$ komutu ikinci bir parametre olmadığında DFT hesaplarken nokta sayısını x 'in boyuna eşit seçer. $\cos(\frac{3\pi}{5}n)$ işaretinin DFT'sini hesapladığımızda yukarıda soldaki grafikteki gibi $\frac{3\pi}{5}$ ve $-\frac{3\pi}{5}$ noktalarında dürtüler görmeyi bekleriz. ($[0-2\pi]$ arasını incelediğimiz için $-\frac{3\pi}{5} \rightarrow -\frac{3\pi}{5}+2\pi=\frac{7\pi}{5}$'e denk gelir.). Yukarıda sağdaki grafikte ikiden fazla dürtü elde etme sebebimizin nedeni indis vektörü $n2$'nin, **$\cos(\frac{3\pi}{5}n)$ işaretinin periyodunun tam katı olmayacak şekilde oluşturulmasından kaynaklanmaktadır.** Özetle, $\cos(\frac{3\pi}{5}n)$ işaretinin periyodu 10 örnek olduğu için indis vektörünü 10'un bir tam katı olacak şekilde ayarladığınız sürece daima DFT genlik grafiğinde 2 tane dürtü görürsünüz.

2.2.2 DFT genlik ve faz grafiklerinden işaret sentezi

Zamandaki kapalı formunu bilmediğiniz bir diziyi farklı frekanslardaki sinüs ve kosinüslerin doğrusal kombinasyonu şeklinde, dizinin DFT'sine bakarak yazabilirsiniz. Bunun için aşağıda verilen iki denklemi bilmeniz gerekmektedir.

$$\tilde{x}\,[n]=\sum_{k=0}^{N-1} a_k e^{+j\frac{2\pi}{N}kn} \quad (DFS\ sentez\ denklemi)$$

$$\tilde{x}\,[n]=\frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] e^{+j\frac{2\pi}{N}kn} \quad (DFT\ sentez\ denklemi)$$

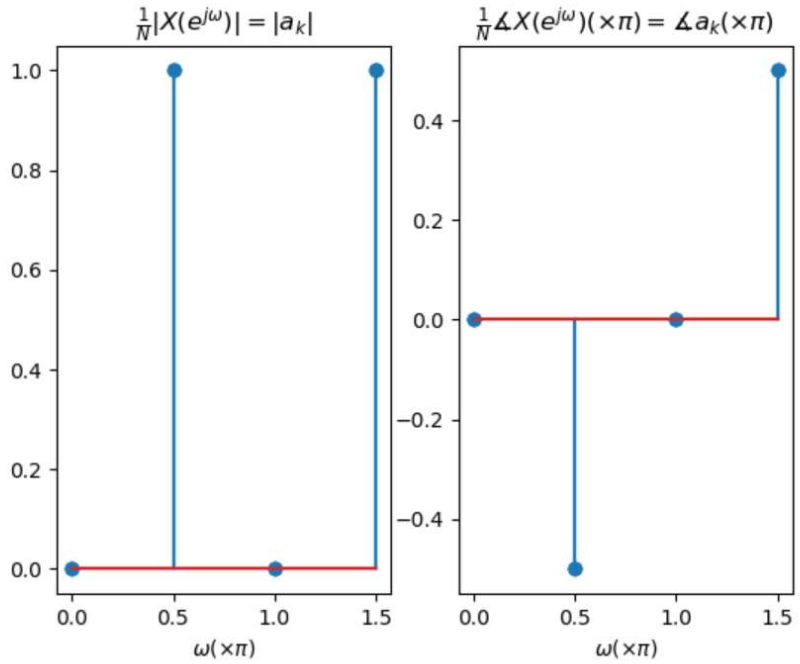
Yukarıdaki iki denkleme baktığınızda periyodik olan ayrık zamanlı işaretin Fourier serisi katsayıları a_k ile DFT katsayıları $\tilde{X}[k]$ arasında $a_k=\frac{\tilde{X}[k]}{N}$ ilişkisi olduğunu görüyoruz. Diğer bir değişle, **DFT hesapladıktan sonra elde ettiğiniz diziyi dizinin boyuna bölerseniz elde ettiğiniz yeni dizi DFS katsayıları olur**.

Örnek: Aşağıda verilen dizi $A\cos(\omega_0 n+\phi)$ formatında bir sinüzoidal işaretin bir periyotta aldığı değerlere karşılık gelmektedir.

$$x = [0,2,0,-2]$$

Aşağıdaki kod, yukarıdaki dizinin fft() komutu ile DFT'sinin alınıp daha sonra işaretin uzunluğu olan 4'e bölünüp elde ettiğimiz yeni kompleks dizinin genlik ve faz grafiklerini çizmektedir. Grafikleri çizdirirken okuma kolaylığı açısından hem yatay eksen hemde DFT dizisinin fazı π sayısına bölünmüştür. Bu nedenle, yatay eksendeki bir değeri okurken o sayıyı π ile çarpmayı unutmayınız; ilgili eksen etiketlerine "$(\times \pi)$" ifadesi bu sebeple eklenmektedir.

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.fftpack import fft , ifft
x=np.array([0,2,0,-2])
N=len(x)
n=np.arange(0,4)
w_disc=n*2*np.pi/N # 0-2pi arasında(2pi noktası dahil değil) 2pi/4
adımlı vektör
X_abs=np.abs(fft(x)/N)
X_phase=np.angle(fft(x)/N)
plt.subplot(121)
plt.stem(w_disc/np.pi,X_abs)
plt.xlabel('$\omega (\\times \pi)$')
plt.title('$\\frac{1}{N}|X(e^{j\omega})|=|a_{k}|$')
plt.subplot(122)
plt.stem(w_disc/np.pi,X_phase/np.pi)
plt.xlabel('$\omega (\\times \pi)$')
plt.title('$\\frac{1}{N}\measuredangle X(e^{j\omega})(\\times
\pi)=\measuredangle a_k(\\times \pi)$')
```



Yukarıda, solda fft() fonksiyonu ile işaretin DFT'sini hesaplayıp işaretin uzunluğuna bölünmüş yeni dizinin genlik grafiği, sağda ise yine aynı dizinin faz grafiği verilmiştir. Genlik grafiğinden sadece birinci indisteki 0.5π ve üçüncü indisteki 1.5π frekanslarında DFS katsayılarının sıfırdan farklı değere eşit olduğunu görüyoruz. Bu nedenle DFS sentez denklemini kullanarak $\tilde{x}[n]$'i sentezlemek istersek sadece $a_1 e^{j0.5\pi n}$ ve $a_3 e^{j1.5\pi}$ kompleks üstel işaretlerini toplamamız gerekir.

a_1 katsayısınu bulmak için;

$$a_1 = |a_1| e^{j\angle a_1} = 1e^{-j\frac{\pi}{2}}$$

Yukarıda $|a_1|$ değerini genlik grafiğinde birinci indisteki değerden, $\angle a_1$ değerini ise faz grafiğindeki birinci indisteki değerden okuyoruz. Faz grafiğini çizdirirken okuma kolaylığı olması için vektör π 'ye bölünmüştü. Bu nedenle, dik eksendeki değeri alırken π ile çarpılması gerekir. Benzer şekilde a_3'ü hesaplarsak;

$$a_3 = |a_3| e^{j\angle a_3} = 1e^{+j\frac{\pi}{2}}$$

Sonuç olarak $\tilde{x}[n]$ aşağıdaki ifadeye eşit olur;

$$\tilde{x}\,[n] = \sum_{k=0}^{N-1} a_k e^{+j\frac{2\pi}{N}kn} = a_1 e^{j0.5\pi n} + a_3 e^{j1.5\pi n}$$

$$= 1e^{-j\frac{\pi}{2}} e^{j0.5\pi n} + 1e^{+j\frac{\pi}{2}} e^{j1.5\pi n}$$

$e^{j1.5\pi n} = e^{j(2\pi-0.5\pi)n} = e^{j2\pi n} e^{-j0.5\pi n} = e^{-j0.5\pi n}$ olduğu için;

$$= 1e^{-j\frac{\pi}{2}} e^{j0.5\pi n} + 1e^{+j\frac{\pi}{2}} e^{-j0.5\pi n}$$

$$= e^{j\left(0.5\pi n-\frac{\pi}{2}\right)} + e^{-j\left(0.5\pi n-\frac{\pi}{2}\right)}$$

$$= 2\cos\left(0.5\pi n - \frac{\pi}{2}\right)$$

Buradan $A = 2$, $\omega_0 = \frac{\pi}{2}$ ve $\phi = -\frac{\pi}{2}$ olduğunu görürüz.

Eğer $n = [0,1,2,3]$ vektörü için $2\cos\left(0.5\pi n - \frac{\pi}{2}\right)$ işaretini hesaplarsanız en başta verilen $[0,2,0,-2]$ dizisini elde edersiniz.

3 BASAMAK, DÜRTÜ İŞARETLERİNİN KOLAYCA OLUŞTURULMASI

İndis vektörü $n = -20, -19, \dots, 0, \dots, 20$ olduğunu varsayalım. Bu indis vektörü için aşağıda $u[n]$, $u[n-3]$,$\delta[n]$, $\delta[n+2]$ ve $u[n] - u[n-10]$ işaretlerini üreten kodlar verilmektedir.

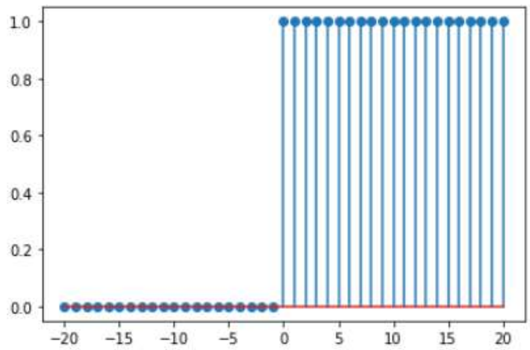
Öncelikle $u[n]$'i oluşturalım:

```
n = np.arange(-20,21)
#yol-1
u_n = np.array([0 if i < 0 else 1 for i in n])
#yol-2
u_n = []
for i in n:
    if i<0:
        u_n.append(0)
    else:
        u_n.append(1)
```

```
#yol-3(bu yol önerilmemektedir)
u_n = np.concatenate((np.zeros(20),np.ones(21)))

#yol-4
u_n=np.ones(len(n))
u_n[n<0]=0
```

Yukarıda verilen örneklerden farklı yaklaşımlarla da verilen indis vektörü için basamak fonksiyonu üretilebilir. Yukarıdaki 'u_n' değişkenini n'e göre çizdirirseniz aşağıdaki grafiği elde edersiniz.



Şimdi de $u[n-3], \delta\,[n], \delta[n+2]$ ve $u[n] - u[n-10]$ işaretlerini yukarıdaki "1. Yol" ile aynı indis vektörü için oluşturalım. Diğer yollar veya kendinizin bulduğu yollar ile bu üç işareti oluşturmayı ayrıca deneyiniz.

$u[n-3]$:

```
n= np.arange(-20,21)
u_n_3 = np.array([0 if i<3 else 1 for i in n])
```

$\delta\,[n]$:

```
n= np.arange(-20,21)
dirac_n = np.array([0 if i!=0 else 1 for i in n])
```

$\delta[n+2]$:

```
n= np.arange(-20,21)
```

```
dirac_n_arti_2 = np.array([0 if i!=-2 else 1 for i in n])
```

$u[n] - u[n-10]$:

```
n= np.arange(-20,21)
u_n_eksi_u_n_10 =np.array([1 if i>=0 and i<10 else 0 for i in n])
```

4 KODLAR İLE ALAKALI SIK KARŞILAŞILAN HATALAR/SORULAR VE ÇÖZÜMLERİ/CEVAPLARI

1) Grafik çizdirmek istiyorum ama aşağıdaki gibi bir hata alıyorum:

x and y must have same first dimension, but have shapes…

Çözüm: İster $\mathtt{stem}(x, y)$ *ister* $\mathtt{plot}(x, y)$ *olsun grafik çizdirirken tek boyutlu* x *ve* y *dizilerinin boyları aynı olmak zorundadır. Bu hatayı alıyorsanız boyları aynı değildir. Düzeltip tekrar deneyin.*

2) fft() komutu ile DFT hesaplayıp çizdirmek istiyorum şu şekide bir uyarı alıyorum:

…ComplexWarning: Casting complex values to real discards the imaginary part

Çözüm: Böyle bir uyarıyı almanızın nedeni kompleks bir diziyi doğrudan plot() veya stem() ile çizdirmeye çalışmanızdır. fft() ile elde edeceğiniz dizi kompleks olacağıdan ya genlik-faz ikilisini yada gerçek-sanal ikilisini çizdirirsiniz (Bu derste genelde genlik-faz ikilisini çizdiririz). Bunun için Numpy'ın abs() fonsiyonuna giriş olarak fft() sonucu elde ettiğiniz diziyi uygularsanız çıkışta genliği, yine Numpy'ın angle() fonsiyonuna giriş olarak fft() sonucu elde ettiğiniz diziyi uygularsanız fazı elde etmiş olursunuz.

3) “fft() komutu ile DFT genlik ve faz grafiklerini çizdirirken yatay ekseni oluşturmakta zorlanıyorum”:

Çözüm: N uzunluklu $x[n]$ *dizisinin fft() ile DFT’si,* $[0-2\pi]$ *aralığında sürekli olan* $x[n]$ *‘in DTFT’si* $X(e^{j\omega})$*’nın* 0*’dan* 2π*’ye (*2π *noktası dahil değil) N noktalı olacak şekilde(noktaların arasındaki mesafe* $\frac{2\pi}{N}$*)örneklenmesiyle elde edilen diziye karşılık geliyor. Bu örnekleme noktaları da aşağıda verdiğim* ω_{DFT} *vektörüne karşılık gelir:*

$$\omega_{DFT} = \left[0, \frac{2\pi}{N}, \frac{4\pi}{N}, \ldots, 2\pi - \frac{2\pi}{N}\right]$$

Bu vektörü oluşturmanın en basit yollarından biri 0*’dan* $N-1$*’e giden bir indis vektörü oluşturup daha sonra bu vektörü* $\frac{2\pi}{N}$*’ile çarpmaktır. Örneğin,* $N=8$ *için:*

```
w=np.arange(0,8)*(2*np.pi/8)
```

Aşağıdaki kodlar da aynı vektörü üretir:

```
w=np.arange(0,2*np.pi,2*np.pi/8)
w=np.linspace(0,2*np.pi-2*np.pi/8,8)
```

4) “Verilen kodun aynısını kendim yazıyorum ancak aşağıdaki gibi bir hata alıyorum”:

NameError: name '……' is not defined

Çözüm: Yukarıda ..... yerinde yazan kütüphane import edilmemiştir. İlgili kütüphaneleri kodunuzun en başında import ediniz.

5) "Grafik çizdirirken grafiği bir önceki figürün üzerine çizdiriyor"

Çözüm:Yeni figür çizdirmeden hemen oncesine aşağıdaki kodu yazınız:

```
plt.figure()
```

6) "Grafik çizdirirken frekans eksenini π'ye (veya 2π'ye) bölmek zorunlu mu? Bunu neden yapıyoruz?"

Cevap: Grafik çizdirirken frekans eksenini π*'ye bölmek sadece grafiği okuma kolaylığı açısından yapılır. Mesela* $cos\left(\frac{3\pi}{5}n\right)$ *işaretinin DFT'sini hesaplayıp genlik grafiğini çizdirdiğinizde dürtüleri* 0.6π *ve* 1.4π*'de görürüz. Eğer yatay ekseni* π*'ye bölerseniz dürtüler* 0.6 *ve* 1.4 *noklarında olur ve bu değerleri kolayca okuyabiliriz (bu değerlerin* π *ile çarparak okunması için eksen etiketinde* $(\times \pi)$ *koymayı unutmayınız). Eğer* π*'ye bölmezseniz, dürtüler* $0.6\pi =$*1.88495.... ve* $1.4\pi = 4.3982297$*.... şeklinde kolay okunup yorumlanacak değerlere sahip olmayacaktır. Bu işlemin yapılması zorunlu değildir fakat yapmanız halinde hem grafikleri okurken hem de soruları cevaplarken size kolaylık sağlayacaktır.*

7) Kodlarda eksen etiketleri veya başlıkları eklerken '$$' sembolü arasında yazılanlar ne anlama geliyor? Bizim yapmamız zorunlu mu?

Cevap: Dolar sembolü arasına yazılanlar Latex kodları, tek amaçları eksen etiketlerinde veya başlıklarda matematiksel ifadelerin gösterimi içindir. Örneğin,

$plt.xlabel$('ω') *yazdığınızda x-ekseninde ω sembolünü görürsünüz. Bu tür Latex kodlarını bilmeniz sizden beklenmemektedir. Eksen veya başlıkları isimlendirirken herkesin anlayacağı şekilde isimlendirmeniz, örneğin "omage" yazmanız, yeterli olacaktır.*

5 ÖDEV-3:

1) **(50 PUAN)** DZD olan bir sistemin dürtü cevabı $h[n] = \delta[n] - \delta[n-1]$ olarak verilmektedir.

   a) $H(e^{j\omega})$'yı elinizle hesaplayın ve yorum olarak ekleyin. Hesapladığınız bu fonksiyonunun genliği $|H(e^{j\omega})|$ ve fazı $\angle H(e^{j\omega})$ ‘yı ω'yı $0 - 2\pi$ aralığında1000 noktadan oluşacak şekilde oluşturup < ile çizdirin.

   b) $|H(e^{j\omega})|$'ya bakarak bu filtrenin nasıl bi karakteristiğe sahip olduğunu yorumlayın.

   c) $h[n]$ işaretini $n = 0, \ldots, 15$ indislerinde tanımlı 16 noktalı olacak şekilde oluşturun. Oluşturduğunuz bu dizinin fft() fonksiyonu ile 16 noktalı DFT'sini hesaplayıp genlik ve faz grafiklerini çizdirin.

2) **(25 PUAN)** Aşağıdaki dizi $Acos(\omega_0 n + \phi)$ formatında olan sinüzoidal işaretin tam bir periyot $n = 0,1, \ldots 7$'ye kadar aldığı değerlere karşılık gelmektedir.

   $$[0, 0.707106, -1, 0.707106, 0, -0.707106, 1, -0.707106]$$

   Bu dizinin DFT'sinin genlik ve faz grafiklerine bakarak A, ω_0 ve ϕ değerlerini bulunuz.

3) **(25 PUAN)** $x[n] = cos(\frac{\pi}{3}n)$ işareti dürtü cevabı $h[n] = \delta[n] - \delta[n-1]$ olan bir sisteme giriş olarak verildiğinde sistemin çıkışında elde edilen işareti $n = 0,1,2, \ldots, 10$ aralığında çizdiriniz.

   Not: Bu soruda, giriş işaretinin özfonksiyon olduğuna ve çıkış işaretinin $y[n] = \left|H\left(e^{j\frac{\pi}{3}}\right)\right| cos\left(\frac{\pi}{3}n + \angle H\left(e^{j\frac{\pi}{3}}\right)\right)$ ile bulunabileceğine dikkat ediniz. Dolayısıyla, birinci sorudan yararlanarak önce $\left|H\left(e^{j\frac{\pi}{3}}\right)\right|$ ve $\angle H\left(e^{j\frac{\pi}{3}}\right)$ değerlerini hesaplayınız ve bu değerleri print() komutu ile yazdırınız ve yorum olarak bu değerleri belirtiniz.

6 TESLİM ŞEKLİ ve ZAMANI

Kodlar bölümünde yazılan kodları kendiniz bir Jupyter Notebook'ta yazarak sonuçları gözlemleyin. ÖDEV-3 başlığı altında verilen sorularda istenenleri Python'da (Jupyter Notebook kullanarak) kodlayınız ve yaptığınız çalışmayı **ÖDEV3_AnonimKimlikNumarası.ipynb** ismiyle kaydedip Ders Web Sayfası'na yükleyiniz. Laboratuvar ön çalışmaları (ev ödevi), haftaya Perşembe günü sabah 05:00'e kadar sisteme yüklenmelidir. Sisteme geç yüklenen dosyalar kabul edilmeyecektir. Ekte, örnek bir ödev çözümü şablonu verilmektedir (bknz: **ODEV3_c98765.ipynb**). Jupyter Notebook'ta yapacağınız çözümler **bu şablona göre hazırlanmalıdır.**

AnonimKimlikNumarası: Öğrencinin TC Kimlik Numarası'nı girerek ürettiği bir kod numarasıdır. Kod numarası üretmek için ders web sayfasındaki "**AnonimKimlikNumarasiUreteci.ipynb**" isimli kodu çalıştırınız. Bu kod ilk harfi a, b veya c olan ve devamında 5 haneli sayı içeren bir koddur. Bu kod kişiye özeldir ve ödevlerde anonim değerlendirmeyi sağlamak üzere kullanılmaktadır; başkaları ile paylaşmayınız.