Örnek1.ipynb

In [ ]:

```
import matplotlib.pyplot as plt  # grafik çizimi için gerekli kütüphanenin aktifleştirilmesi
import numpy as np               # dizi işlemleri için kütüphane aktif hale getirilir.
f=100  # x(t) işaretinin temel frekansı
T=1/f # x(t) işaretinin temel periyodu
t=np.arange(0.,2*T,0.0001)     # t zaman indisinin tanımlanması(2 periyot boyunca)
x=5*np.cos(200*np.pi*t)  # x(t) işaretinin tanımlanması

plt.figure()
plt.subplot(3,1,1)
plt.plot(t,x, 'black')     # x(t)) işaretinin grafiğinin çizdirilmesi
plt.xlabel("t (sn)")    # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("x(t)")     # grafiğin y ekseninin isimlendirilmesi

# x(t)=5*cos(200*pi*t) sinyalinin Fs=1200 Hz ile örneklenmesi
Fs=1200     # örnekleme frekansının tanımlanması
Ts=1/Fs   # örnekleme periyodunun tanımlanması
N=12    # bir periyottaki örnek sayısının tanımlanması
n=np.arange(0.,2*N)  # örnekleme indisinin 0'dan iki periyot olacak şekilde array olarak tanımlanması
xn=5*np.cos(200*np.pi*n*Ts)  #örneklenmiş x[n] işaretinin tanımlanması

plt.subplot(3,1,2)
plt.stem(n,xn)      # x[n] işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("x[n]")     # grafiğin y ekseninin isimlendirilmesi


M = 2   # Seyrek örnekleme (down sampling) oranı

xn_d = xn[np.arange(0, np.size(xn, 0), M)] # x[n] işaretinden sadece M katlarındaki örneklerin alınması
Nn_d = len(xn_d)
n_d = np.arange(0,Nn_d)  # x_d[n] işaretinin indis dizisi

plt.subplot(3,1,3)
plt.stem(n_d, xn_d) # x_d[n] işaretinin grafiğinin çizdirilmesi
plt.ylabel('$x_d[n]$')  # grafiğin x ekseninin isimlendirilmesi
plt.xlabel('n (örnek)')  # grafiğin y ekseninin isimlendirilmesi
plt.show() # grafiklerin gösterilmesi
```

In [ ]:

```
# x(t) işaretinin Fs=150 Hz ile örneklenmesi
Fs=150     # örnekleme frekansının tanımlanması
Ts=1/Fs   # örnekleme periyodunun tanımlanması
N=3    # bir periyottaki örnek sayısının tanımlanması
n=np.arange(0.,2*N)  # örnekleme indisinin 0'dan iki periyot olacak şekilde array olarak tanımlanması
xn=5*np.cos(200*np.pi*n*Ts)  #örneklenmiş x[n] işaretinin tanımlanması

plt.figure()
plt.stem(n,xn)      # x[n] işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)") # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$x_2[n]$") # grafiğin y ekseninin isimlendirilmesi
plt.show()      # grafiklerin gösterilmesi
```

In [ ]:

Örnek2.ipynb

Öncelikle $x[n]$ işaretini ve bu işaretin Fourier transformu olan $X(\omega)$ işaretini tanımlayalım

In [ ]:
```
import matplotlib.pyplot as plt  # grafik çizimi için gerekli kütüphanenin aktifleştirilmesi
import numpy as np               # dizi işlemleri için kütüphane aktif hale getirilir.

N = 80    # çizdirilmek istenen toplam örnek sayısının tanımlanması
nTs = np.arange(-10 , 10, 20/N)     # nTs indislerinin tanımlanması
xn = np.sinc(nTs)**2  # x(nTs) işaretinin tanımlanması
n = np.arange(-40, 40) # x[n] işaretinin indis ekseninin tanımlanması

# x[n] işaretinin fourier transformu
w = np.arange(-np.pi, np.pi, 2*np.pi/N)  # omega ekseninin -pi ile +pi arasında tanımlanması
xw = np.fft.fftshift(np.fft.fft(xn,N)/N)  # ayrık zamanlı işaretin Fourier transformu
```

Şimdi $x[n]$ işaretinin ve $X(\omega)$ işaretinin grafiklerini çizdirelim

In [ ]:
```
plt.figure()
plt.subplot(2,1,1)
plt.stem(n,xn)      # x[n] işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("x[n]")     # grafiğin y ekseninin isimlendirilmesi

plt.subplot(2,1,2)
plt.stem(w/np.pi,abs(xw))      # X(w) işaretinin grafiğinin çizdirilmesi
plt.xlabel("$\omega$ / $\pi$")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$X(\omega)$")     # grafiğin y ekseninin isimlendirilmesi
plt.show()
```

$M = 2$ ile seyrek örnekleme yaparak $x_d[n]$ işaretinin ve bu işaretin Fourier dönüşümü olan $X_d(\omega)$ işaretinin elde edilmesi

In [ ]:
```
M = 2   # Seyrek örnekleme (down sampling) oranı

xn_d = xn[np.arange(0, np.size(xn, 0), M)] # x[n] işaretinden sadece M katlarındaki örneklerin alınması
N_d = (round)(N/M)
n_d = np.arange(-N_d/2,N_d/2)  # x_d[n] işaretinin indis dizisi


# x_d[n] işaretinin fourier transformu
w_d = np.arange(-np.pi, np.pi, 2*np.pi/N_d)  # omega ekseninin -pi ile +pi arasında tanımlanması
xw_d = np.fft.fftshift(np.fft.fft(xn_d,N_d)/N_d)  # ayrık zamanlı işaretin Fourier transformu
```

Şimdi $x_d[n]$ işaretinin ve $X_d(\omega)$ işaretinin grafiklerini çizdirelim

In [ ]:
```
plt.figure()
plt.subplot(2,1,1)
plt.title('M = 2 için')
plt.stem(n_d, xn_d) # x_d[n] işaretinin grafiğinin çizdirilmesi
plt.ylabel('$x_d[n]$')  # grafiğin x ekseninin isimlendirilmesi
plt.xlabel('n (örnek)')  # grafiğin y ekseninin isimlendirilmesi
plt.show() # grafiklerin gösterilmesi

plt.subplot(2,1,2)
plt.stem(w_d/np.pi,abs(xw_d))      # X_d(w) işaretinin grafiğinin çizdirilmesi
plt.xlabel('$\omega$ / $\pi$')  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel('$X_d(\omega)$')     # grafiğin y ekseninin isimlendirilmesi
```

$M = 3$ ile seyrek örnekleme yaparak $x_d[n]$ işaretinin ve bu işaretin Fourier dönüşümü olan $X_d(\omega)$ işaretinin elde edilmesi

In [ ]:
```
M = 3   # Seyrek örnekleme (down sampling) oranı

xn_d = xn[np.arange(0, np.size(xn, 0), M)] # x[n] işaretinden sadece M katlarındaki örneklerin alınması
N_d = (round)(N/M)
n_d = np.arange(-N_d/2,N_d/2)  # x_d[n] işaretinin indis dizisi


# x_d[n] işaretinin fourier transformu
w_d = np.arange(-np.pi, np.pi, 2*np.pi/N_d)  # omega ekseninin -pi ile +pi arasında tanımlanması
xw_d = np.fft.fftshift(np.fft.fft(xn_d,N_d)/N_d)  # ayrık zamanlı işaretin Fourier transformu
```

Şimdi $x_d[n]$ işaretinin ve $X_d(\omega)$ işaretinin grafiklerini çizdirelim

In [ ]:

```
plt.figure()
plt.subplot(2,1,1)
plt.title('M = 3 için')
plt.stem(n_d, xn_d) # x_d[n] işaretinin grafiğinin çizdirilmesi
plt.ylabel('$x_d[n]$')  # grafiğin x ekseninin isimlendirilmesi
plt.xlabel('n (örnek)')  # grafiğin y ekseninin isimlendirilmesi
plt.show() # grafiklerin gösterilmesi

plt.subplot(2,1,2)
plt.stem(w_d/np.pi,abs(xw_d))      # X_d(w) işaretinin grafiğinin çizdirilmesi
plt.xlabel('$\omega$ / $\pi$')  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel('$X_d(\omega)$')     # grafiğin y ekseninin isimlendirilmesi
```

In [ ]:

```

```

Örnek 3.ipynb

In [ ]:
```
import matplotlib.pyplot as plt  # grafik çizimi için gerekli kütüphanenin aktifleştirilmesi
import numpy as np               # dizi işlemleri için kütüphane aktif hale getirilir.


# x(t)=5*cos(200*pi*t) sinyalinin Fs ile örneklenmesi örneği sonucunda elde edilen x[n] işareti

Fs=1200     # örnekleme frekansının tanımlanması
Ts=1/Fs   # örnekleme periyodunun tanımlanması
N=12    # örnek sayısının tanımlanması
n=np.arange(0,N)  # Bir periyot için örnekleme indisinin array olarak tanımlanması
xn=5*np.cos(200*np.pi*n*Ts)  #örneklenmiş x[n] işaretinin tanımlanması

plt.figure()
plt.subplot(3,1,1)
plt.stem(n,xn)      # x[n] işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)")
plt.ylabel("x[n]")

L = 2              # Sık örnekleme(Up Samling)Katsayısı
Nn_u = N*L        # up sample yapılmış işaret için indis array inin oluşturulması
xn_u = np.zeros(Nn_u)       # 0'lar ile dolu bir dizi oluşturulması
xn_u[np.arange(0,len(xn_u),L)] = xn     # 0 ile dolu dizinin üzerine L aralıklar ile x[n] işaretinin
                                        # değerlerinin atanması
n_u = np.arange(0, Nn_u)   # indis dizisi

plt.subplot(3,1,2)
plt.stem(n_u, xn_u)   # x_u[n] işaretinin grafiğinin çizdirilmesi
plt.ylabel('$x_u[n]$')
plt.xlabel('n (örnek)')


# interpolasyon işlemi
hn = np.array([0,1/2,1,1/2,0])  # lineer interpolasyonda L=2 için h[n] işareti
xn_i = np.convolve(xn_u,hn,'full')  # konvolüsyon işlemi
n_i = np.arange(0, len(xn_i))   # indis dizisi
plt.subplot(3,1,3)
plt.stem(n_i,xn_i)      # x_i[n] çıkış işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)")
plt.ylabel("$x_i[n]$")
plt.show()      # grafiklerin gösterilmesi
```

In [ ]:
```
```

Örnek4.ipynb

Öncelikle $x[n]$ işaretini ve bu işaretin Fourier transformu olan $X(\omega)$ işaretini tanımlayalım

In [ ]:
```
import matplotlib.pyplot as plt  # grafik çizimi için gerekli kütüphanenin aktifleştirilmesi
import numpy as np               # dizi işlemleri için kütüphane aktif hale getirilir.

N = 40    # çizdirilmek istenen toplam örnek sayısının tanımlanması
nTs = np.arange(-10,10,20/40)     # nTs indislerinin tanımlanması
xn = np.sinc(nTs)**2  # x(nTs) işaretinin tanımlanması
n = np.arange(-20, 20) # x[n] işaretinin indis ekseninin tanımlanması

# x[n] işaretinin fourier transformu
w = np.arange(-np.pi, np.pi, 2*np.pi/N)  # omega ekseninin -pi ile +pi arasında tanımlanması
xw = np.fft.fftshift(np.fft.fft(xn,N)/N)  # ayrık zamanlı işaretin Fourier transformu
```

Şimdi $x[n]$ işaretinin ve $X(\omega)$ işaretinin grafiklerini çizdirelim

In [ ]:
```
plt.subplot(2,1,1)
plt.stem(n,xn)      # x[n] işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$x[n]$")     # grafiğin y ekseninin isimlendirilmesi

plt.subplot(2,1,2)
plt.stem(w/np.pi,abs(xw))      # X(w) işaretinin grafiğinin çizdirilmesi
plt.xlabel("$\omega$ / $\pi$")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$X(\omega)$ (Genlik)")     # grafiğin y ekseninin isimlendirilmesi
```

$L = 2$ ile sık örnekleme yaparak $x_u[n]$ işaretinin ve bu işaretin Fourier dönüşümü olan $X_u(\omega)$ işaretinin elde edilmesi

In [ ]:
```
L = 2     # Sık örnekleme (up sampling) oranı

N_u = N*L    # sık örnekleme sonucu elde edilecek dizinin toplam örnek miktarı
xn_u = np.zeros(N_u)      # Başlangıçta N*L elemanlı 0 dizisinin oluşturulması
xn_u[np.arange(0,len(xn_u),L)] = xn  # oluşturulan 0 elemanlı dizide her L katı elemana x[n] işaretinin
                                     #  elemanlarının sırasıyla atanması
n_u = np.arange(-N_u/2, N_u/2)  # sık örneklenmiş işaretin indis dizisi


# x_u[n] işaretinin fourier transformu
w_u = np.arange(-(np.pi), (np.pi), 2*(np.pi)/N_u)  # omega ekseninin -pi ile +pi arasında tanımlanması
xw_u = np.fft.fftshift(np.fft.fft(xn_u, N_u)/N_u)   # ayrık zamanlı x_u[n] işaretin Fourier transformu
```

Şimdi $x_u[n]$ işaretinin ve $X_u(\omega)$ işaretinin grafiklerini çizdirelim

In [ ]:
```
plt.subplot(2,1,1)
plt.stem(n_u,xn_u)      # x_u[n] işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$x_u[n]$")     # grafiğin y ekseninin isimlendirilmesi

plt.subplot(2,1,2)
plt.stem(w_u/np.pi,abs(xw_u))      # X_u(w) işaretinin grafiğinin çizdirilmesi
plt.xlabel("$\omega$ / $\pi$")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$X_u(\omega)$ (Genlik)")     # grafiğin y ekseninin isimlendirilmesi
```

Elde edilen grafikte $x_u[n]$ işaretinde $L = 2$ için her iki örnek arasına bir adet 0 değerli örnek eklenmektedir. $X_u(\omega)$ işaretine bakıldığında ise $-\pi$ ile $-\pi/2$ ve $\pi/2$ ile π bölgeleri arasında kalan yarım üçgen darbelerin olmaması gerekmektedir. Bu nedenle işaret frekans domaininde bir ideal alçak geçiren filtreden (AGF) geçirilecektir. İşaretin filtreleme sonrasında enerjisinin aynı kalması için AGF'nin kazancı L kadardır. Kesim frekansı ise π/L 'dir. Buradaki AGF işlemi zaman domaininde işaretin ara değerlemesinin yani interpolasyon işleminin yapılmasına karşılık gelmektedir.

NOT: Frekans domaininde ideal AGF'den geçirmek, zaman domaininde sinc(.) işareti ile konvolüsyon yapmaya karşı gelir.

In [ ]:
```
# AGF'nin frekans cevabının oluşturulması

hw = np.zeros(N_u)      # H(w) ile X_u(w) noktasal çarpılacağı için uygun boyutta başlangıçta sıfır
                        # vektörü oluşturulur
w_i = np.arange(-(np.pi), (np.pi), 2*(np.pi)/N_u)  # açısal frekans olan w değerlerinin oluşturulması
indis = np.where(abs(w_i)<=np.pi/2)    # Kesim frekansı (W_c) pi/2 olan filtrenin pi/2'den küçük
                                       # frekanslarının indis değerlerinin bulunması
hw[indis] = L   # bulunan indis değerlerine Filtre kazancına göre değer atanması
```

$X_i(\omega)$ =$X_u(\omega)$ *$H(\omega)$ işlemi

In [ ]:
```
xw_i = xw_u*hw     # X_i(w) işaretinin elde edilmesi
N_i =len(xw_i)
```

Grafiklerin çizidirilmesi:

In [ ]:
```
plt.subplot(2,1,1)
plt.stem(w_i/np.pi, abs(hw))      # H(w) işaretinin (AGF) grafiğinin çizdirilmesi
plt.xlabel("$\omega$ / $\pi$")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$H(\omega)$")     # grafiğin y ekseninin isimlendirilmesi

plt.subplot(2,1,2)
plt.stem(w_i/np.pi,abs(xw_i))      # X_u(w) işaretinin grafiğinin çizdirilmesi
plt.xlabel("$\omega$ / $\pi$")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$X_i(\omega) (Genlik)$")     # grafiğin y ekseninin isimlendirilmesi
```

Şimdi işaretin zaman domainindeki haline bakıp sık örneklenip örneklenmediğini kontrol edelim

In [ ]:
```
xn_i = np.fft.ifft(np.fft.ifftshift(xw_i), N_i)*N_i  # X_i(w)'dan ters Fourier dönüşümü ile x_i[n]
                                                      # işaretinin elde edilmesi
ni =np.arange(-N_i/2, N_i/2)   # indis arrayi
```

In [ ]:
```
plt.stem(ni, abs(xn_i))
plt.xlabel("n (örnek)")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$x_i[n]$")     # grafiğin y ekseninin isimlendirilmesi
plt.show()
```

Dolayısıyla $x[n]$ işareti düzgün bir şekilde sık örneklenerek $x_i[n]$ işareti elde edilmiştir.

NOT: Konunun daha iyi anlaşılabilmesi için $x[n]$, $x_u[n]$ ve $x_i[n]$ işaretlerinin grafiklerini ve $X(\omega)$, $X_u(\omega)$ ve $X_i(\omega)$ işaretlerinin grafiklerini ayrı ayrı inceleyiniz

In [ ]:

Örnek6.ipynb

İlgili ses dosyasının yüklenerek işarete ait bilgilerin alınması grafiğinin çizdirilmesi

Öncelikle "sound.wav" isimli ses işaretini yükleyelim.

In [ ]:
```
import matplotlib.pyplot as plt  # grafik çizimi için gerekli kütüphanenin aktifleştirilmesi
import numpy as np
from scipy.io.wavfile import read  # Ses dosyasının okunması için gerekli kütüphane

Fs, x = read("sound.wav")
```

Ses işaretimiz hakkında yorum yapabilmek için işaretin boyut ve örnekleme frekans değerlerine bakalım.

In [ ]:
```
print(x.shape)  # x'in boyut bilgisi
print(Fs)
```

Ses işaretimizin boyut ve örnekleme frekans değerlerine bakacak olursak şu yorumları yapabiliriz: Ses işaretimiz iki kanaldan oluşmaktadır ve her kanalında 106150 adet örnek mevcuttur. Ayrıca bu ses işareti $Fs = 22050Hz$ ile örneklenerek elde edilmiştir.

Ele aldığımız örnek ses işareti iki kanallı olduğu ve bu örnekte ilgili konunun işleyişini göstermek açısından tek kanalın yeterli olması nedeniyle aşağıdaki komut satırı aracılığıyla işaretin sadece tek kanalı alınarak örnekleme frekansı 22050 Hz olan ve toplamda 106150 adet örnek içeren $x[n]$ işaretini elde etmiş oluruz. Örnekte sık ve seyrek örnekleme işlemleri bu işaret üzerinden yapılacaktır.

In [ ]:
```
xn = x[:,1]  # x'in sadece ilk sütunundaki örnekler alınmıştır (1. kanalın örnekleri)
```

In [ ]:
```
N = len(x)
n=np.arange(0,N)  # örnekleme indisinin 0'dan iki periyot olacak şekilde array olarak tanımlanması
plt.figure()
plt.stem(n,xn)      # x[n] işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)")  # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("x[n]")     # grafiğin y ekseninin isimlendirilmesi
```

$x[n]$ işaretinin $M = 4$ ile seyrek örneklenerek $x_d[n]$ işaretinin elde edilmesi

In [ ]:
```
M = 4   # Seyrek örnekleme (down sampling) oranı
xn_d = xn[np.arange(0, np.size(xn, 0), M)] # x[n] işaretinden sadece M katlarındaki örneklerin alınması
N_d = len(xn_d)
n_d = np.arange(0,N_d)  # x_d[n] işaretinin indis dizisi

plt.figure()
plt.stem(n_d, xn_d) # x_d[n] işaretinin grafiğinin çizdirilmesi
plt.ylabel('$x_d[n]$')  # grafiğin x ekseninin isimlendirilmesi
plt.xlabel('n (örnek)')  # grafiğin y ekseninin isimlendirilmesi
plt.show() # grafiklerin gösterilmesi
```

Şimdi seyrek örnekleme yapılmadan önceki ses işaretimiz olan $x[n]$ ve $M = 4$ ile seyrek örnekleme sonucu elde edilen $x_d[n]$ işaretlerini sırasıyla dinleyelim.

NOT: Ses işaretini dinlerken işareti uygun Fs ile dinlediyip dinlemediğinize dikkat ediniz.

In [ ]:
```
from IPython.lib.display import Audio
print('x[n] işareti için:')
display(Audio(xn, rate=Fs))
print('x_d[n] işareti için:')
display(Audio(xn_d, rate=Fs/M))
```

$x[n]$ işaretinin $L = 4$ ile sık örneklenerek $x_u[n]$ işaretinin elde edilmesi

In [ ]:
```
L = 4     # Sık örnekleme (up sampling) oranı
N_u = N*L    # sık örnekleme sonucu elde edilecek dizinin toplam örnek miktarı
xn_u = np.zeros(N_u)     # Başlangıçta N*L elemanlı 0 dizisinin oluşturulması
xn_u[np.arange(0,len(xn_u),L)] = xn  # oluşturulan 0 elemanlı dizide her L katı elemana x[n] işaretinin
                                     #  elemanlarının sırasıyla atanması
n_u = np.arange(0,N_u)  # x_u[n] işaretinin indis dizisi

# interpolasyon işlemi
hn = np.array([1/4,2/4,3/4,3/4,2/4,1/4])  # lineer interpolasyonda L=2 için h[n] işareti
xn_u = np.convolve(xn_u,hn,'full')  # konvolüsyon işlemi
n_u = np.arange(0, len(xn_u))   # indis dizisi

plt.figure()
plt.stem(n_u,xn_u)      # x_u[n] işaretinin grafiğinin çizdirilmesi
plt.xlabel("n (örnek)") # grafiğin x ekseninin isimlendirilmesi
plt.ylabel("$x_u[n]$")     # grafiğin y ekseninin isimlendirilmesi
```

Şimdi sık örnekleme yapılmadan önceki ses işaretimiz olan $x[n]$ ve $L = 4$ ile sık örnekleme sonucu elde edilen $x_u[n]$ işaretlerini sırasıyla dinleyelim.

NOT: Ses işaretini dinlerken işareti uygun Fs ile dinlediyip dinlemediğinize dikkat ediniz.

In [ ]:
```
from IPython.lib.display import Audio
print('x[n] işareti için:')
display(Audio(xn, rate=Fs))
print('x_u[n] işareti için:')
display(Audio(xn_u, rate=Fs*L))
```

In [ ]: