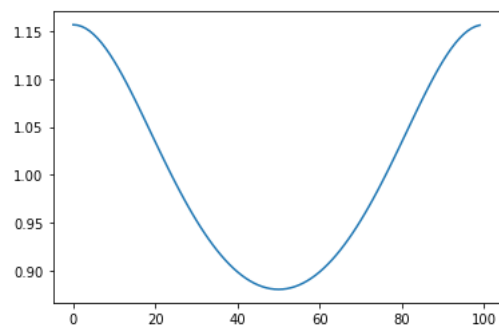
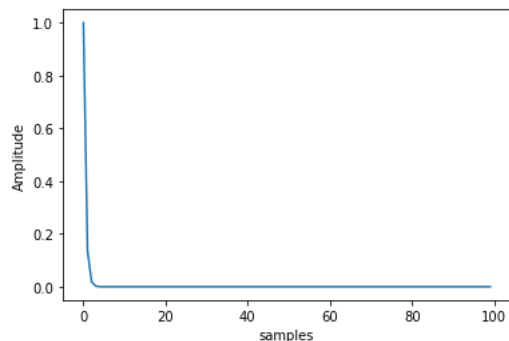


Afshar - Q1

سوال اول در مورد استفاده از سری فوریه است. ابتدا یک سیگنال دلخواه تولید شده و سری فوریه و عکس سری فوریه برای سیگنال بدست آورده شده. سپس سیگنال صوتی فایل صوتی گفته شده خوانده شده است و سری فوریه آن بدست آورده شده و عکس سری فوریه و مقدار خطای بین آنها بدست آورده و نمایش داده شده است. در بخش آخر سوال، دو سیگنالی که یکی آورده شده در صورت سوال هست و دیگری فایل صوتی معرفی شده در صورت سوال، خوانده شده است و هر دو سیگنال فریم بندی شده اند و ضرایب سری فوریه برای هر کدام بدست آورده شده است و در نهایت نمودار و اترفال سه بعدی برای هر کدام رسم شده است

```
1 #Part 1
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 N = 100
6 n = np.arange(N)
7 x = np.exp(-2*n)
8
9 plt.figure()
10 plt.plot(n, x)
11 plt.xlabel('samples')
12 plt.ylabel('Amplitude')
13
14 Xk = np.fft.fft(x)
15
16 plt.figure()
17 plt.plot(n, np.abs(Xk))
18
```

[<matplotlib.lines.Line2D at 0x21a9a7b79d0>]

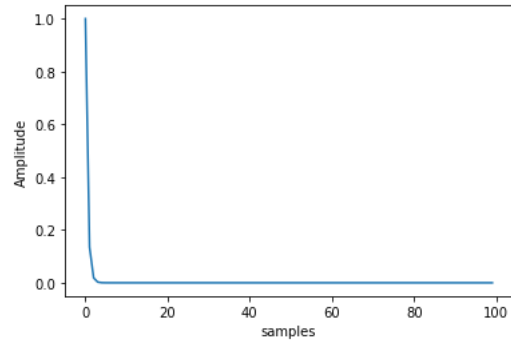


```
1 #part2
2
3 x_hat = np.fft.ifft(Xk)
4
5 plt.figure()
6 plt.plot(n, x_hat.real)
7 plt.xlabel('samples')
```

```
8 plt.ylabel('Amplitude')
```

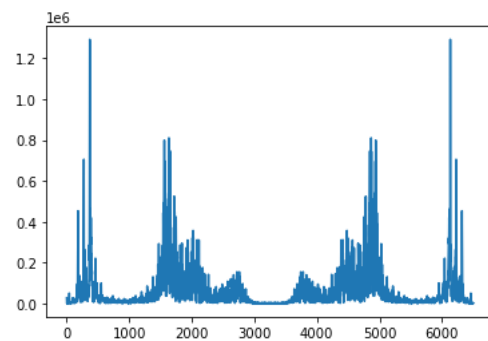
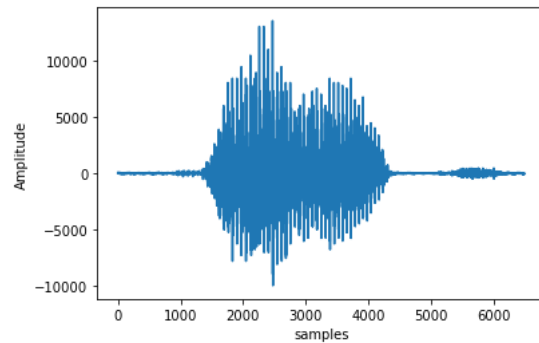
```
9
```

```
Text(0, 0.5, 'Amplitude')
```



```
1 from scipy.io.wavfile import read
2 import matplotlib.pyplot as plt
3
4 (rate, x1) = read("1.wav")
5
6 plt.figure()
7 plt.plot(x1)
8 plt.xlabel('samples')
9 plt.ylabel('Amplitude')
10
11 N1 = len(x1)
12 Xk1 = np.fft.fft(x1)
13
14 plt.figure()
15 plt.plot(np.abs(Xk1))
16
```

```
[<matplotlib.lines.Line2D at 0x21a9ab4bbb0>]
```



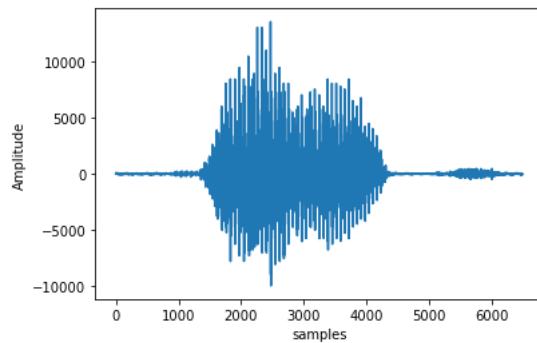
```
1 #part4
2 x1_hat = np.fft.ifft(Xk1)
```

```

3
4 plt.figure()
5 plt.plot(x1_hat.real)
6 plt.xlabel('samples')
7 plt.ylabel('Amplitude')
8
9 MSE = np.mean((x1 - x1_hat.real) ** 2)
10
11 print('Error of reconstruction is :')
12 print(MSE)
13

```

Error of reconstruction is :
2.1240496449281226e-25



```

1  #a functions that we need in part5
2  def enframe(x, win):
3      nx = len(x)
4      if isinstance(win,int):
5          len_ = win
6      else:
7          len_ = len(win)
8      nf = nx // len_
9      f = np.zeros((nf, len_))
10     inds = np.arange(len_)
11     for i in range(nf):
12         f[i,:] = x[inds]
13     return f

```

```

1 #part5
2 import numpy as np
3 from scipy.io import wavfile
4 import matplotlib.pyplot as plt
5 from mpl_toolkits.mplot3d import Axes3D
6
7 fs = 10000
8 t = np.linspace(0, 1, int(fs))
9 alpha = 2
10 beta = 0.7
11
12 x2 = alpha * np.sin(beta * t ** 2)
13 fs, x3 = wavfile.read('variable_pitch.wav')
14
15 enfx2 = enframe(x2, 200)
16 enfx3 = enframe(x3, 200)
17
18 nf2 = enfx2.shape[0]
19 nf3 = enfx3.shape[0]
20

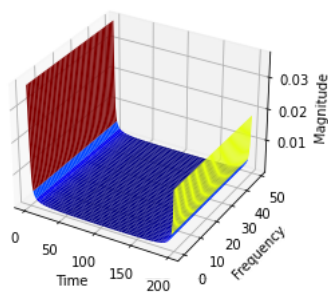
```

```

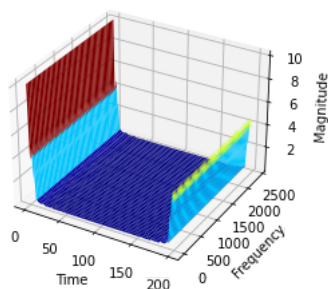
21 X2 = np.zeros((nf2, 200), dtype=complex)
22 for i in range(nf2):
23     X2[i, :] = np.fft.fft(enfx2[i, :], 200)
24
25 X3 = np.zeros((nf3, 200), dtype=complex)
26 for i in range(nf3):
27     X3[i, :] = np.fft.fft(enfx3[i, :], 200)
28
29 T3 = np.arange(200)
30 F3 = np.arange(nf3)
31
32 T2 = np.arange(200)
33 F2 = np.arange(nf2)
34
35 fig = plt.figure()
36 ax = fig.add_subplot(111, projection='3d')
37 X, Y = np.meshgrid(T2, F2)
38 ax.plot_surface(X, Y, np.abs(X2), cmap='jet')
39 ax.set_xlabel('Time')
40 ax.set_ylabel('Frequency')
41 ax.set_zlabel('Magnitude')
42 ax.set_title('Magnitude of DFS coefficients of x2')
43 plt.show()
44
45 fig = plt.figure()
46 ax = fig.add_subplot(111, projection='3d')
47 X, Y = np.meshgrid(T3, F3)
48 ax.plot_surface(X, Y, np.abs(X3), cmap='jet')
49 ax.set_xlabel('Time')
50 ax.set_ylabel('Frequency')
51 ax.set_zlabel('Magnitude')
52 ax.set_title('Magnitude of DFS coefficients of x2')
53 plt.show()
54
55 plt.show()
56

```

Magnitude of DFS coefficients of x2



Magnitude of DFS coefficients of x2



*part6

شکل های بالا به ترتیب برای سیگنال صورت پروژ و سیگنال صوتی ذکر شده است. همان طور که دیده می شود در حالت استفاده از فریم بندی و زمان کوتاه، سیگنال زمانی به درستی قابل تشخیص است و همچنین دقت فرکانسی در ضرایب سری فوریه نیز بیشتر خواهد بود.

1