Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Signals and Systems II
Course Number:	ELE632
Semester/Year (e.g.F2016)	W2023

Instructor:	Dr. Androutsos

Assignment/Lab Number:	Lab 3
Assignment/Lab Title:	Discrete-Time Fourier Series

Submission Date:	2023-03-04
Due Date:	2023-03-12

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Zelenovic	Danilo	501032542	08	DZ

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf

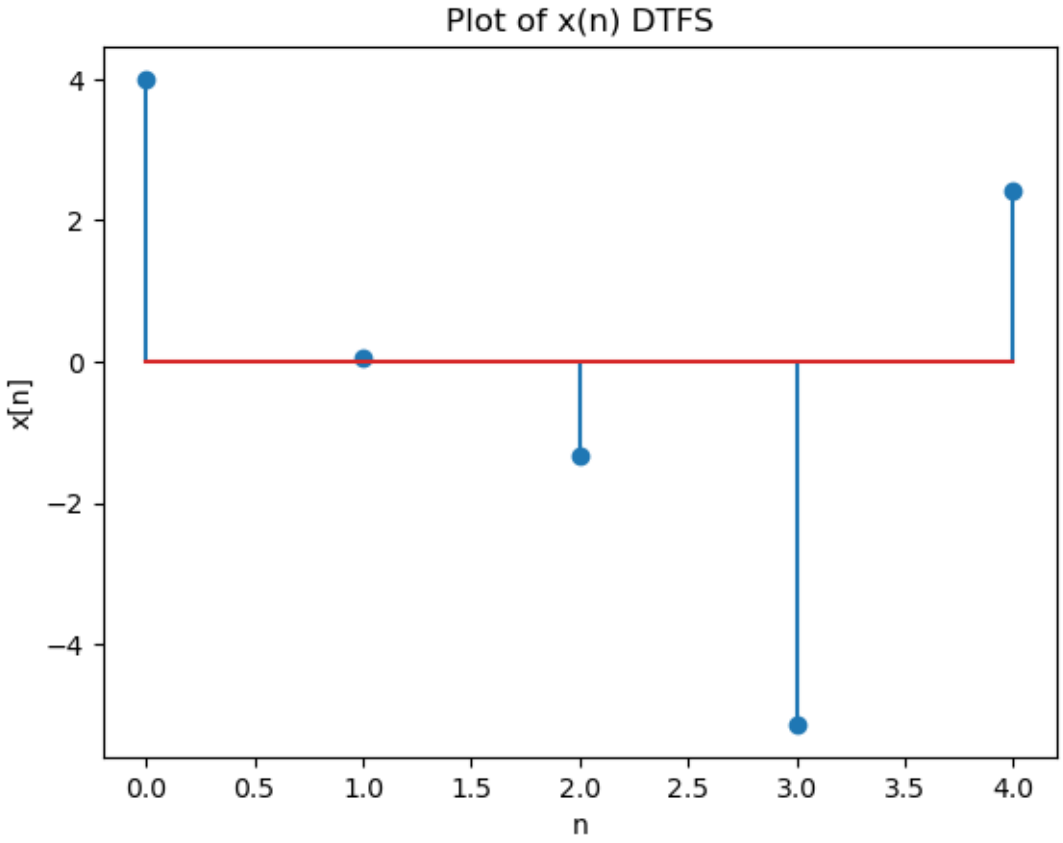
ELE632_Lab3_DaniloZelenovic_501032542_Section08

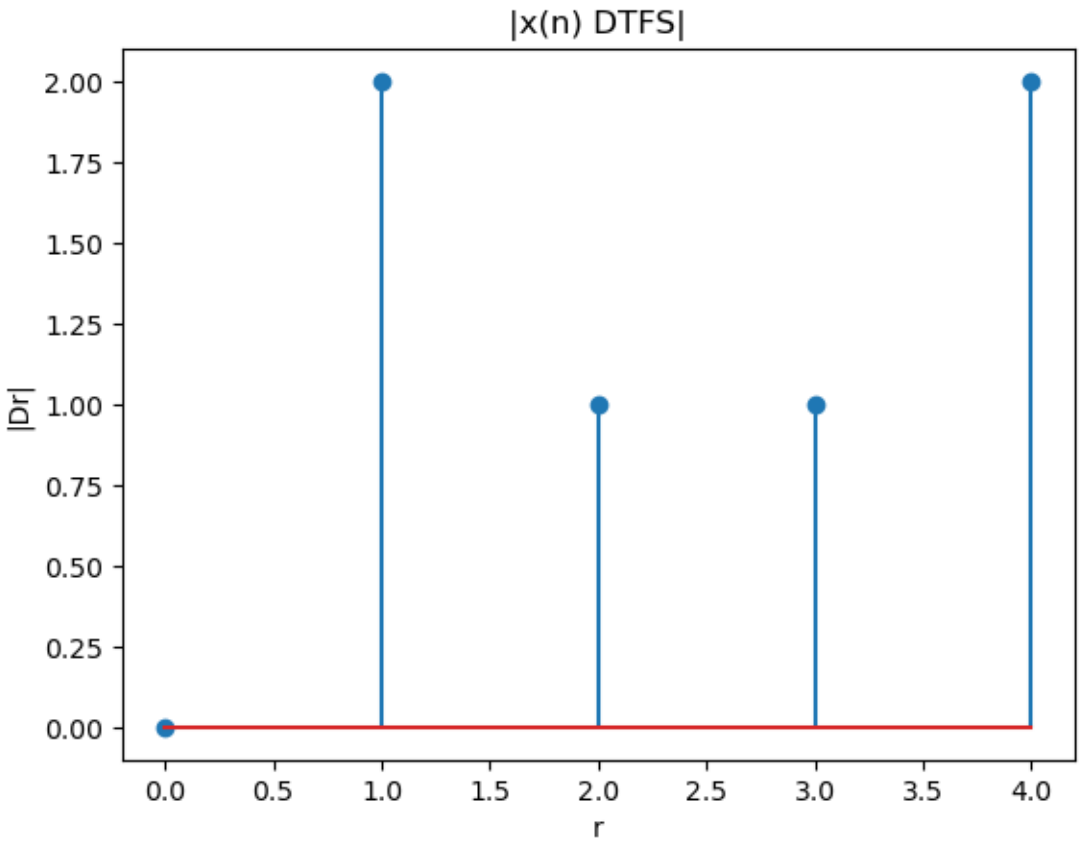
March 4, 2023

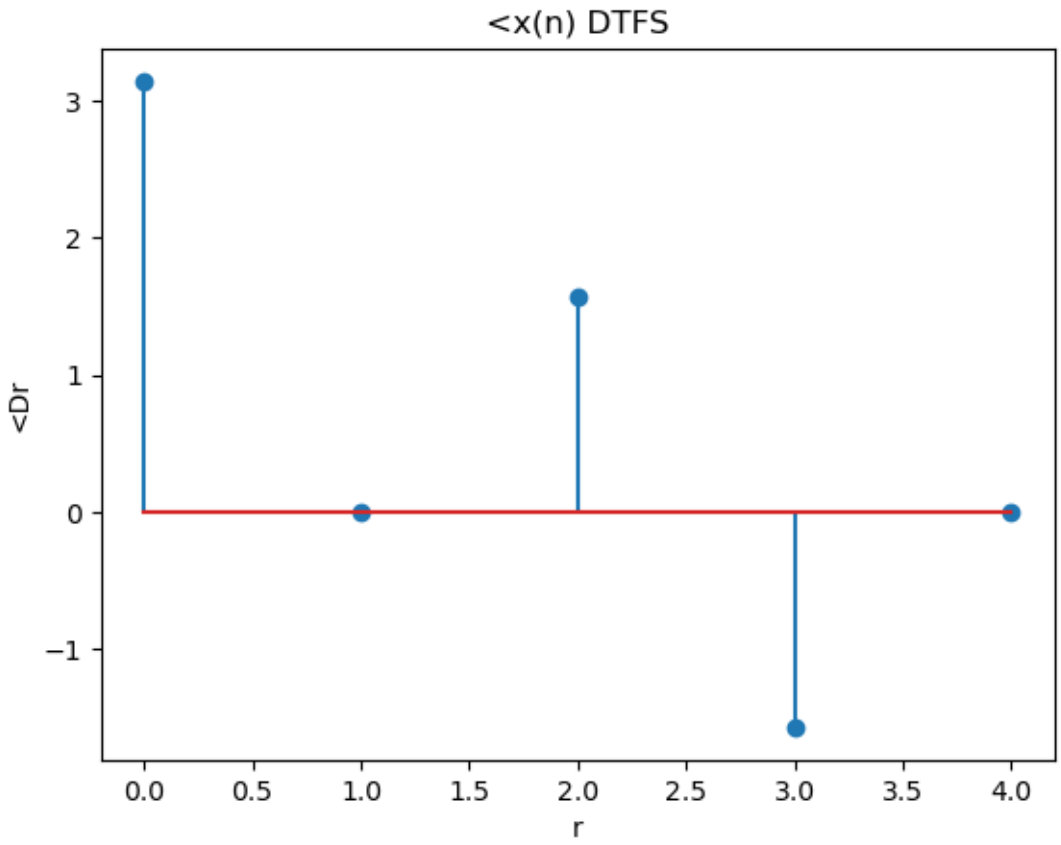
[19]:
```
#Part A1

import numpy as np
import matplotlib.pyplot as plt
n = np.arange(0, 21)
x = 4*np.cos(2.4*np.pi*n) + 2*np.sin(3.2*np.pi*n)
mN1 = (2.4*np.pi)/(2*np.pi)
mN2 = (3.2*np.pi)/(2*np.pi)
#m/N1 = 6/5
#m/N2 = 8/5
N0 = 5
Om0 = 2*np.pi/N0
```

[20]:
```
#Part A2

import numpy as np
import matplotlib.pyplot as plt
n = np.arange(0, N0)
x = 4*np.cos(2.4*np.pi*n) + 2*np.sin(3.2*np.pi*n)
Dr = np.zeros(N0, dtype=complex)
for r in range(N0):
    Dr[r] = np.sum(x * np.exp(-1j * r * n * Om0)) / N0

plt.stem(n, x)
plt.title('Plot of x(n) DTFS')
plt.xlabel('n')
plt.ylabel('x[n]')
plt.show()

plt.stem(np.abs(Dr))
plt.title('|x(n) DTFS|')
plt.xlabel('r')
plt.ylabel('|Dr|')
plt.show()

plt.stem(np.angle(Dr))
plt.title('<x(n) DTFS')
plt.xlabel('r')
```

```
plt.ylabel('<Dr')
plt.show()
```

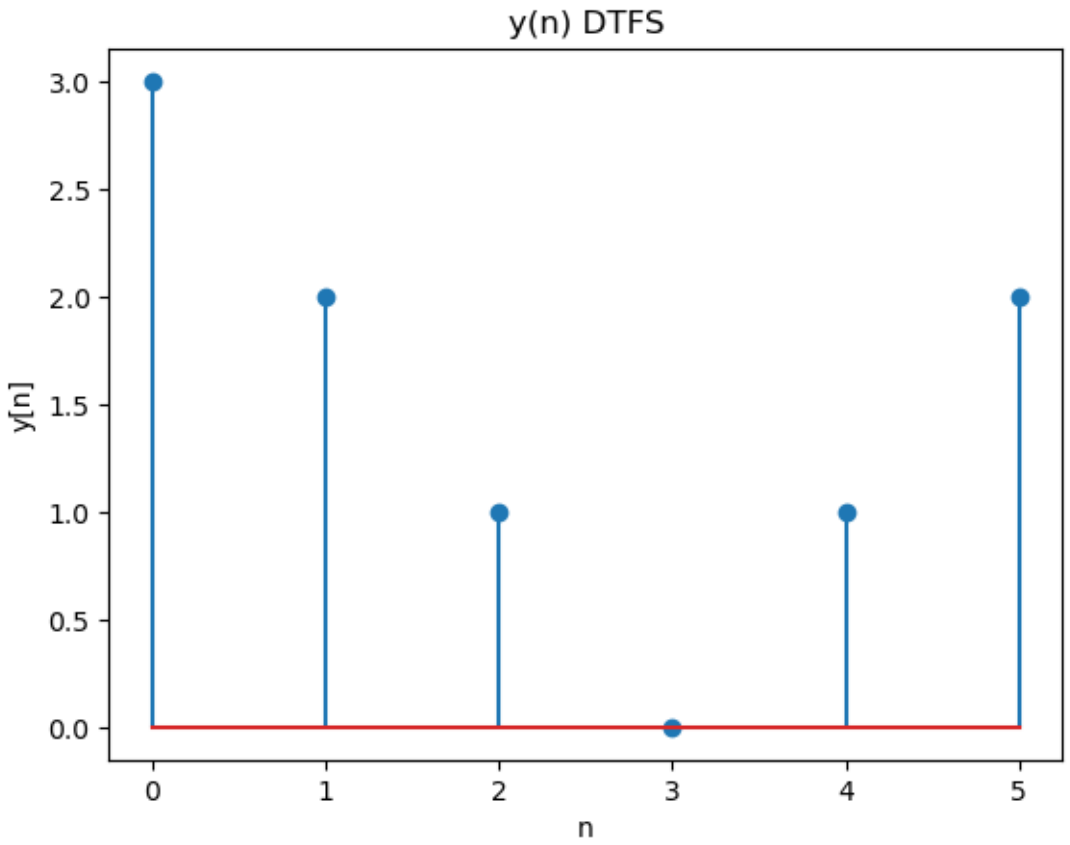


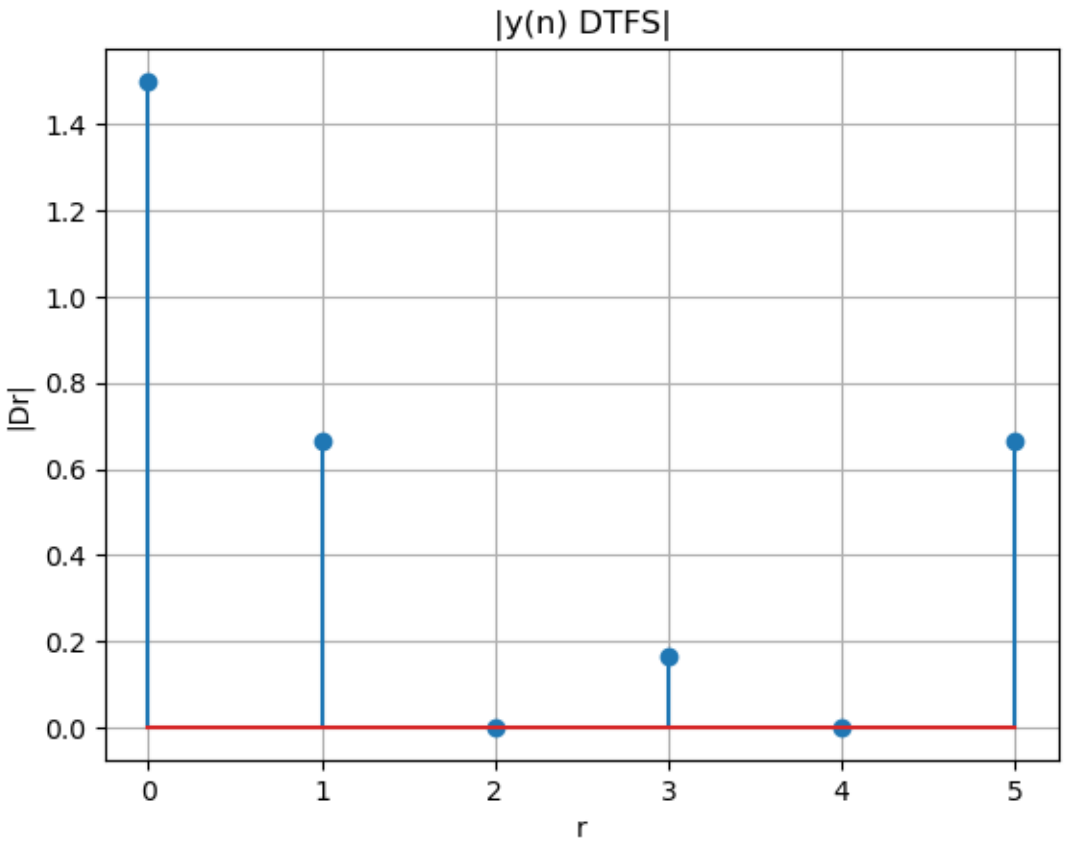

|x(n) DTFS|
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
|Dr|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
r

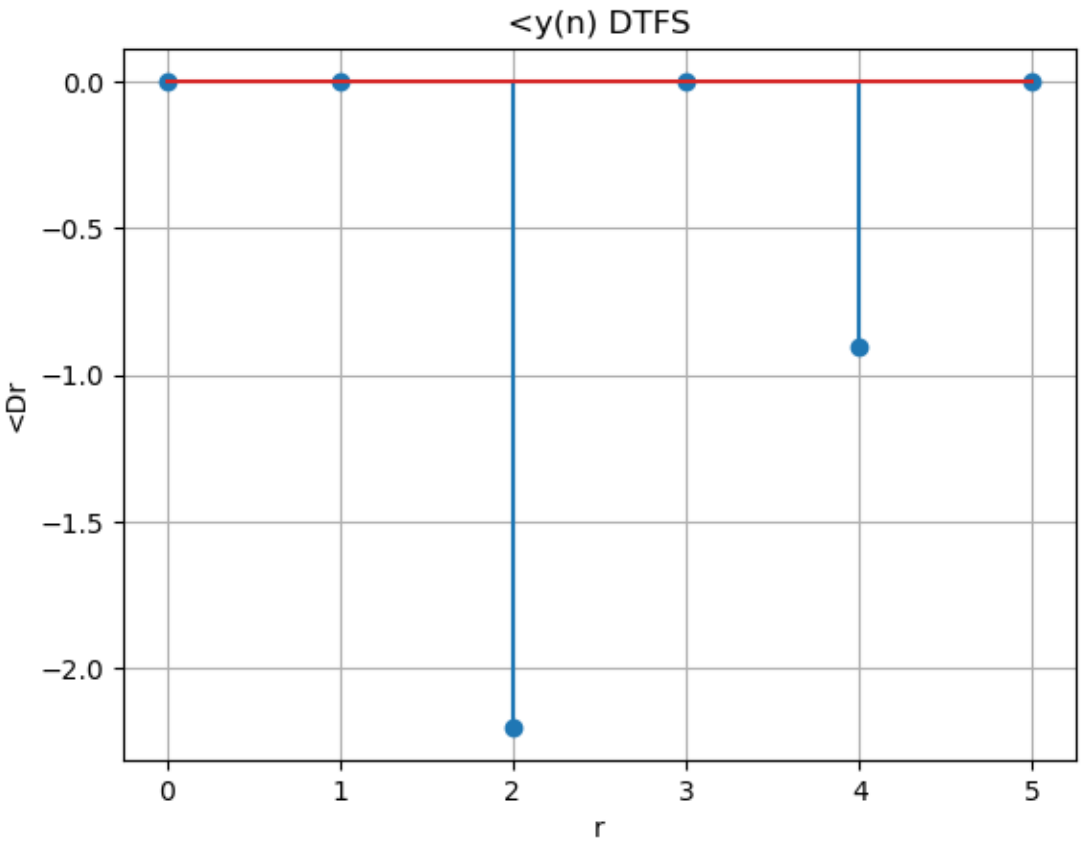


[21]:
```
#Part A3

import numpy as np
import matplotlib.pyplot as plt

N0 = 6
Om0 = 2*np.pi/N0

n = np.arange(0, N0)
y = np.array([3, 2, 1, 0, 1, 2])
Dr = np.zeros(N0, dtype=complex)
for r in range(N0):
    Dr[r] = np.sum(y * np.exp(-1j * r * n * Om0)) / N0

plt.stem(n, y)
plt.title('y(n) DTFS')
plt.xlabel('n')
plt.ylabel('y[n]')
plt.show()
```

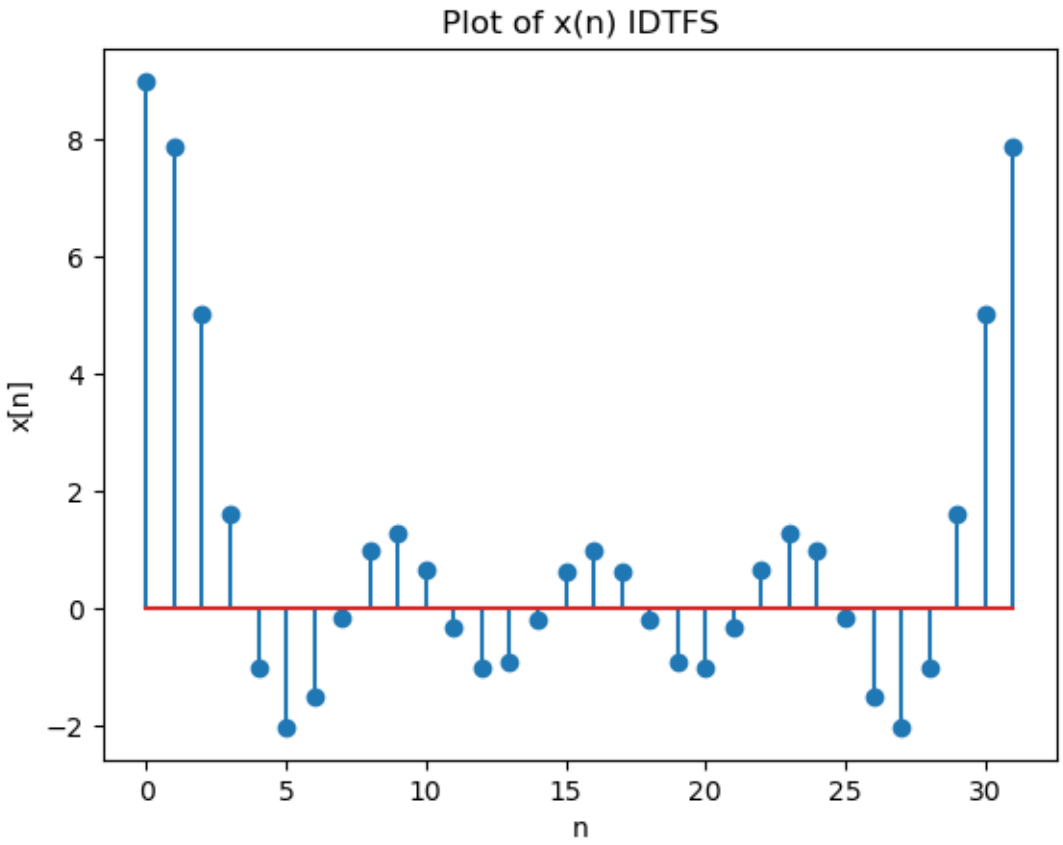
```
plt.stem(np.abs(Dr))
plt.title('|y(n) DTFS|')
plt.xlabel('r')
plt.ylabel('|Dr|')
plt.grid(True)
plt.show()

plt.stem(np.angle(Dr))
plt.title('<y(n) DTFS')
plt.xlabel('r')
plt.ylabel('<Dr')
plt.grid(True)
plt.show()
```




|y(n) DTFS|
|Dr|
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
r

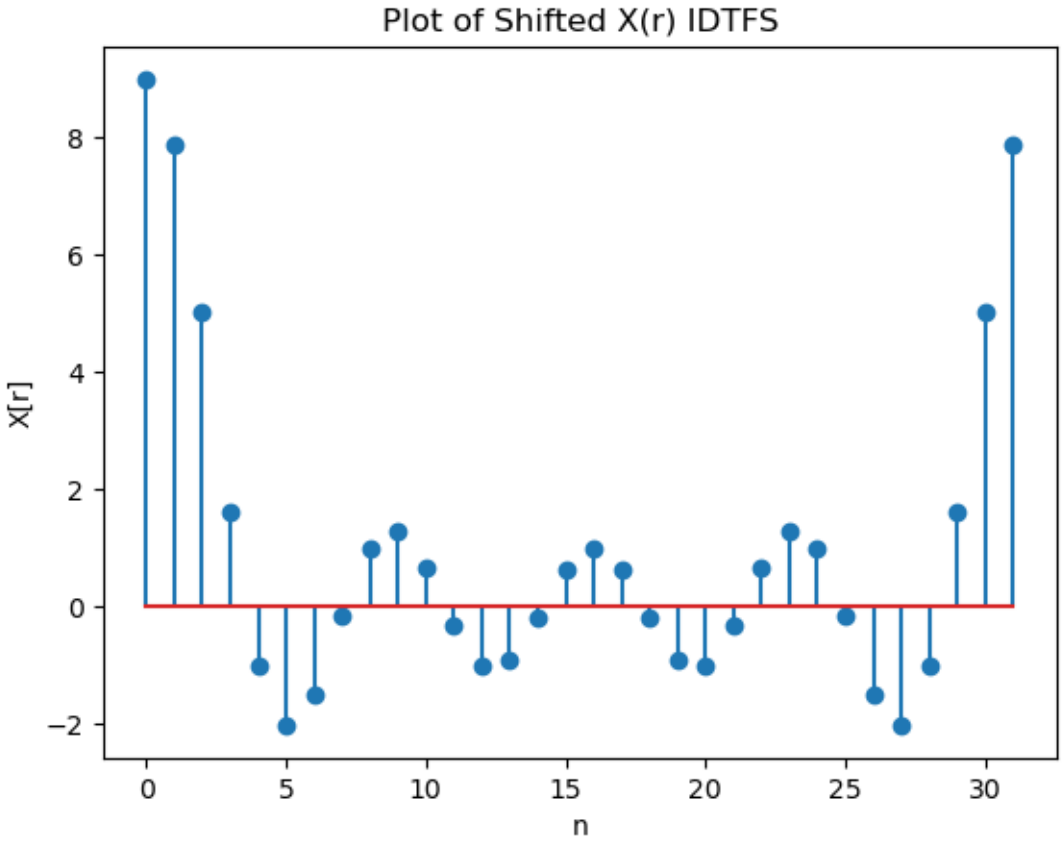


[22]:
```
#Part B1

import numpy as np
import matplotlib.pyplot as plt

N0 = 32
Om0 = 2*np.pi/N0
n = np.arange(N0)

xr = np.concatenate((np.ones(5), np.zeros(23), np.ones(4)))
x = np.fft.ifft(xr)*N0

plt.stem(n, x)
plt.title('Plot of x(n) IDTFS')
plt.xlabel('n')
plt.ylabel('x[n]')
plt.show()
```

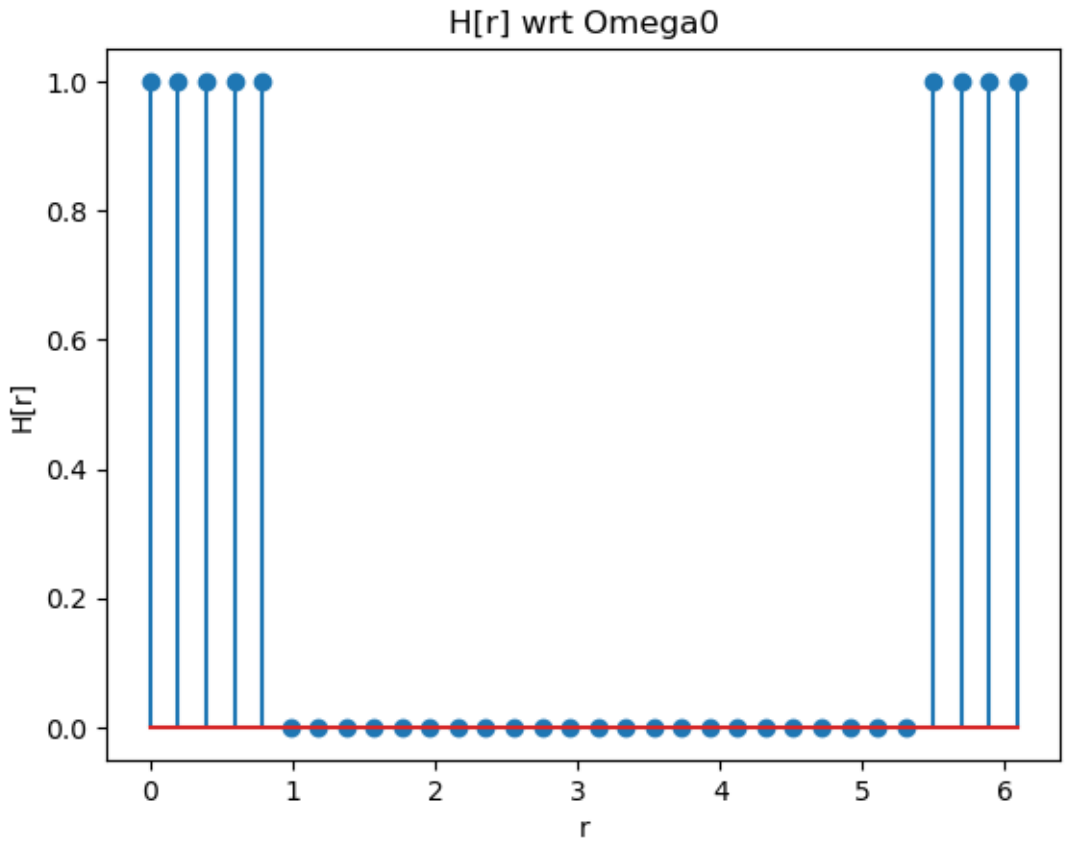


[23]:
```
#Part B2 - NOTE: My D value is 0

import numpy as np
import matplotlib.pyplot as plt

N0 = 32
Om0 = 2*np.pi/N0
n = np.arange(N0)

xr = np.concatenate((np.ones(5), np.zeros(23), np.ones(4)))
X = xr*np.exp(-1j*0*n*Om0)
X = np.fft.ifft(X)*N0

plt.stem(n, X)
plt.title('Plot of Shifted X(r) IDTFS')
plt.xlabel('n')
plt.ylabel('X[r]')
plt.show()
```

```
# In my case, the plot does not differ, as e^0 = 1, but if the D values were␣
 ↪else, the exponential term shifts the spectrum
# left or right depending on the D value (or n in general).
```

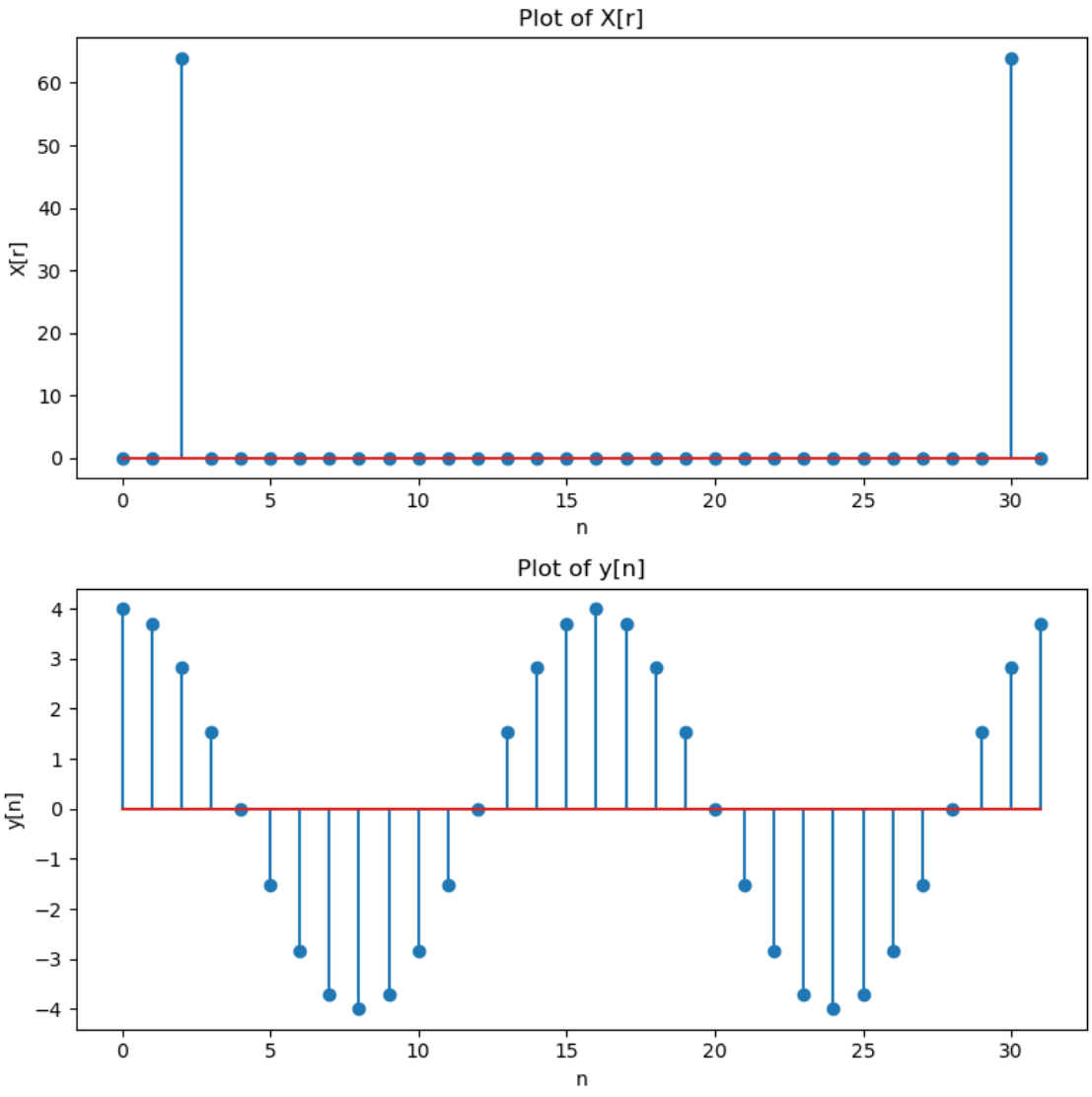


[24]:
```
#Part C1

import numpy as np
import matplotlib.pyplot as plt

N0 = 32
Om0 = 2*np.pi/N0
r = np.arange(N0)
Hr = np.concatenate((np.ones(5), np.zeros(23), np.ones(4)))

plt.stem(Om0*r, Hr)
plt.title('H[r] wrt Omega0')
plt.xlabel('r')
plt.ylabel('H[r]')
plt.show()
```



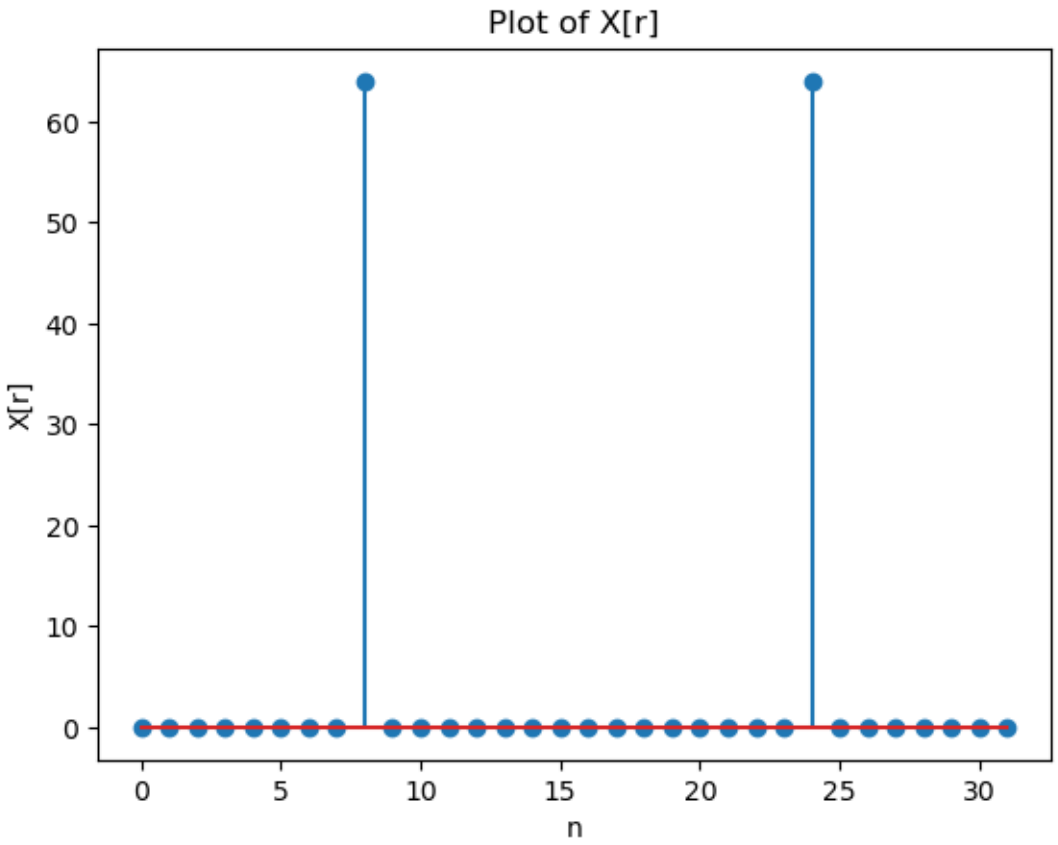
[25]:
```
#Part C2

import numpy as np
import matplotlib.pyplot as plt

N0 = 32
Om0 = 2 * np.pi / N0
r = np.arange(N0)

Hr = np.concatenate((np.ones(5), np.zeros(23), np.ones(4)))
x = 4 * np.cos(np.pi * r / 8)
X = np.fft.fft(x)

fig, axs = plt.subplots(2, 1, figsize=(8, 8))

axs[0].stem(r, X.real)
axs[0].set_title('Plot of X[r]')
axs[0].set_xlabel('n')
axs[0].set_ylabel('X[r]')
```

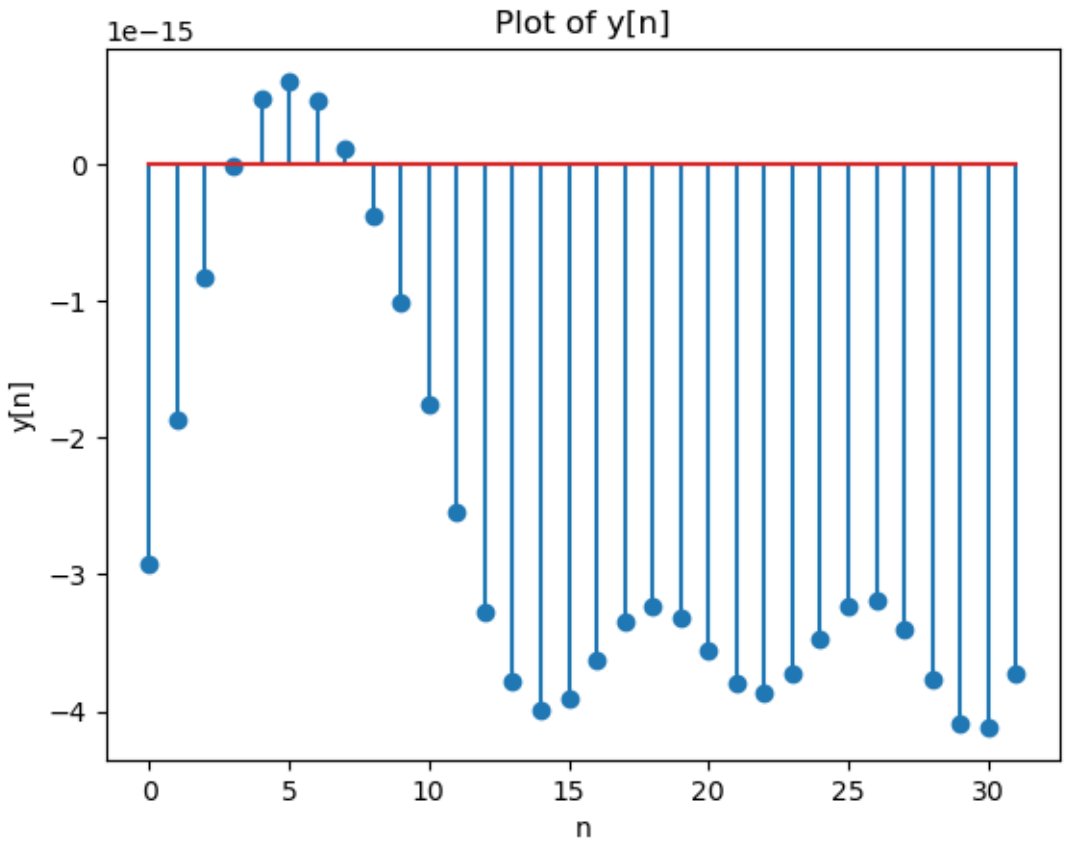
```
Y = X * Hr
y = np.fft.ifft(Y)

axs[1].stem(r, y.real)
axs[1].set_title('Plot of y[n]')
axs[1].set_xlabel('n')
axs[1].set_ylabel('y[n]')

plt.tight_layout()
plt.show()
```



[26]:
```
#Part C3

import numpy as np
import matplotlib.pyplot as plt

N0 = 32
Om0 = 2*np.pi/N0
r = np.arange(0, N0)

Hr = np.concatenate((np.ones(5), np.zeros(23), np.ones(4)))
x = 4*np.cos(np.pi*r/2)
X = np.fft.fft(x)

fig, ax = plt.subplots()
ax.stem(r, X)
ax.set_title('Plot of X[r]')
ax.set_xlabel('n')
ax.set_ylabel('X[r]')

Y = X*Hr
y = np.fft.ifft(Y)

fig, ax = plt.subplots()
ax.stem(r, y.real)
ax.set_title('Plot of y[n]')
ax.set_xlabel('n')
ax.set_ylabel('y[n]')
```

[26]:
```
Text(0, 0.5, 'y[n]')
```


Plot of X[r]
X[r]
n
0
10
20
30
40
50
60
0
5
10
15
20
25
30



[27]:
```
#Part C4

# The graphs are different because the frequencies are different. As they␣
 ↪differ, the cosine portion
# within the Figure 3 is of varying frequencies.
```

[ ]:
```

```