Experiment No. 02

2.1 Experiment Name

Study of auto correlation and cross correlation of the given signals using Python Code

2.2 Objectives

- To get a better understanding of auto correlation and cross correlation of the given signals
- To gain a deeper understanding of signal properties, relationships, and characteristics

2.3 Apparatus

- Jupyter Notebook

2.4 Theory

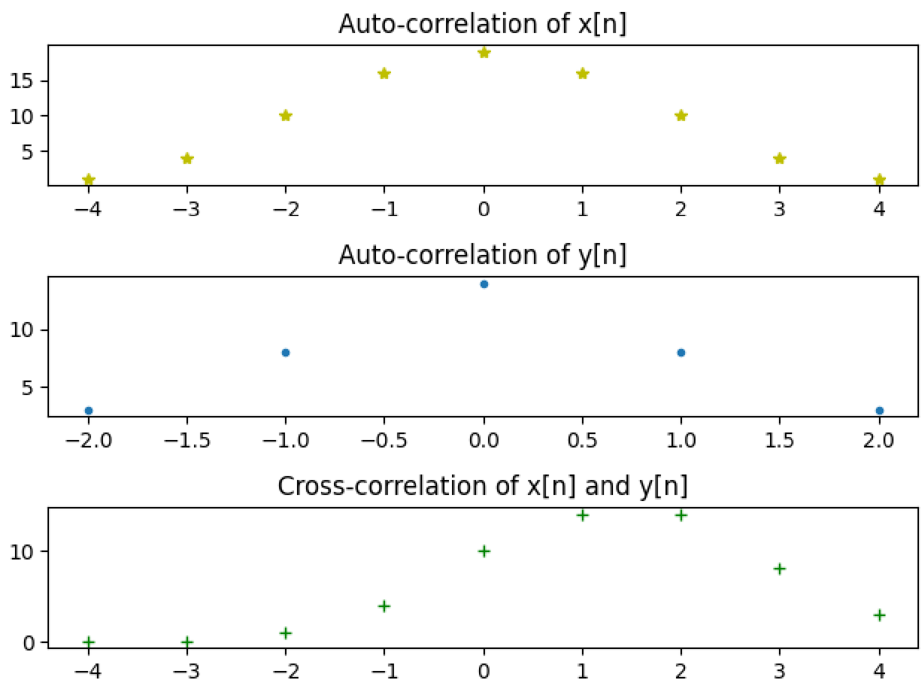
Auto-correlation measures the similarity between a signal and a time-delayed version of itself. It is a mathematical operation that calculates the correlation coefficient at different time lags. The similarity or correlation between two separate signals is measured via cross-correlation. The correlation coefficient between two signals with various time delays is calculated. Both auto-correlation and cross-correlation offer insightful information about the characteristics, connections, and patterns of a signal.

2.5 Python code & graph

```python
import numpy as np
import matplotlib.pyplot as plt

# Define the signals
x = np.array([1, 2, 3, 2, 1])
y = np.array([3, 2, 1])

# Compute the lengths of the signals
A = len(x)
B = len(y)

# Compute the auto-correlation of x[n]
rxx = np.zeros(2*A-1)
for i in range(A):
    for j in range(A):
        rxx[A-1+i-j] += x[i]*x[j]

# Compute the cross-correlation of x[n] and y[n]
rxy = np.zeros(2*A-1)
for i in range(A):
    for j in range(B):
        rxy[A-1+i-j] += x[i]*y[j]
print('cross-correlation of x[n] and y[n]:',rxy)
# Compute the auto-correlation of y[n]
ryy = np.zeros(2*B-1)
for i in range(B):
    for j in range(B):
        ryy[B-1+i-j] += y[i]*y[j]
```

```
cross-correlation of x[n] and y[n]: [ 0.  0.  1.  4. 10. 14. 14.  8.  3.]
```

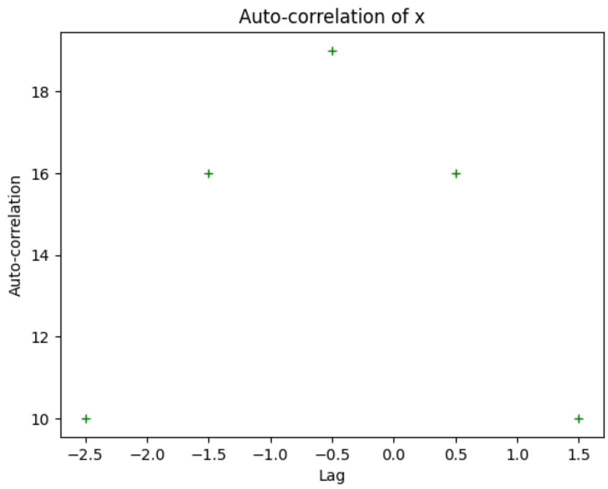
```python
# Plot the results
nxx = np.arange(-(A-1), A)
nxy = np.arange(-(A-1), A)
nyy = np.arange(-(B-1), B)
plt.subplot(311)
plt.plot(nxx, rxx,'y*')
plt.title('Auto-correlation of x[n]')

plt.subplot(312)
plt.plot(nyy, ryy,'.')
plt.title('Auto-correlation of y[n]')

plt.subplot(313)
plt.plot(nxy, rxy,'g+')
plt.title('Cross-correlation of x[n] and y[n]')
plt.tight_layout()
plt.show()
```

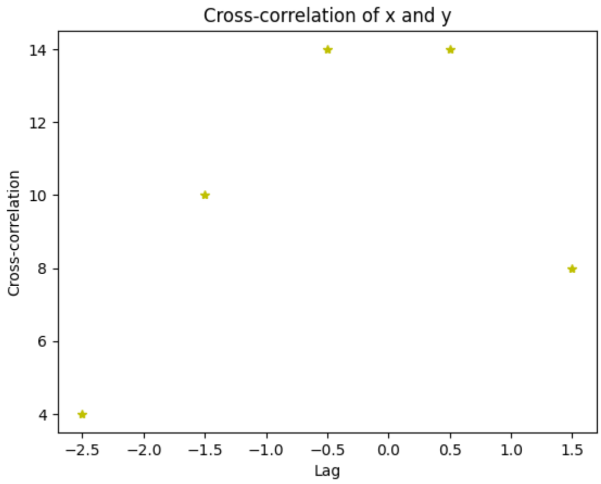


Using correlate function

```
import numpy as np
import matplotlib.pyplot as plt
# import plotly.express as px

# Generate two signals
N = 5  # Number of samples
x  = [1, 2, 3, 2, 1]
y = [3, 2, 1]

# Calculate auto-correlation of x
corr_x = np.correlate(x, x, mode='same')
lags_x = np.arange(-N/2, N/2)

# Plot auto-correlation of x
fig, ax = plt.subplots()
plt.plot(lags_x,corr_x,'g+')
plt.xlabel('Lag')
plt.ylabel('Auto-correlation')
plt.title('Auto-correlation of x')

# Calculate cross-correlation of x and y
corr_xy= np.correlate(x, y,mode='same')
lags_xy= np.arange(-N/2, N/2)
print('Cross-correlation of x and y',corr_xy)
# Plot cross-correlation of x and y
fig, ax = plt.subplots()
plt.plot(lags_xy,corr_xy,'y*')
plt.xlabel('Lag')
plt.ylabel('Cross-correlation')
plt.title('Cross-correlation of x and y')

plt.show()
```

```
Cross-correlation of x and y [ 4 10 14 14  8]
```





2.6 Discussion & Conclusion

In this experiment, we used the python to analyze auto correlation and cross correlation. Both auto-correlation and cross-correlation offer insightful information about the characteristics, connections, and patterns of a signal. They are extensively utilized in many different industries, including as data analysis, picture processing, telecommunications, and audio processing. We may better grasp the behavior and properties of signals and make wise decisions in signal processing applications by studying auto-correlation and cross-correlation.