EXPT NO 3: Discrete Correlation

Aim: To study mathematical operation correlation and measure degree of similarity between two signals

Input Specifications:

- Length of first Signal L and signal values.
- Length of second Signal M and signal values.

Objective:

- Write a function to find Correlation Operation
- Calculate correlation of a DT signals and verify the results using mathematical formula
- Measure the degree of similarity using Carl Pearson Correlation formula

Problem Definition:

1. Find auto correlation of input signal and find the significance of value of output signal at n=0.

Let y[n] = x[n] ○ x[n]

Classify the resultant signal (Even / Odd). Calculate the energy of the signal.

Q. What is the significance of value of y [0]?

2. Find auto correlation of delayed input signal.

Let p[n] = x [n-1] ○ x [n-1].

Compare the resultant signal p[n] with y[n]. Give your conclusion.

3. Find cross correlation of input signal and delayed input signal

Let q[n] = x[n] ○ x [n-1].

Compare the resultant signal q[n] with p[n] and y[n]

Give your conclusion.

4. Find cross correlation of input signal and scaled input signal.

Let s[n] = x[n] ○ ax [n-2].

Compare the resultant signals.

Give your conclusion.

Experimentation and Result Analysis

Sample Example: {Value at n=0 is highlighted}

1. To find y[n] = x[n] ○ x[n]

   Input x[n] = {1, 2, 3, 4}

   Output y[n] = {4, 11, 20, **30**, 20, 11, 4}

Conclusion:

Here, y(n) = y(-n)
Which means, auto correlation output signal is an even signal.
At n = 0, y(0) is maximum. Energy of signal x(n) is $E = \Sigma |x(n)|^2$
That means, y(0) = Energy of signal = 30

2. To find p[n] = x[n-1] ○ x[n-1]

   Input x [n-1] = {0, 1, 2, 3, 4}

   Output p[n] = {0, 4, 11, 20, **30**, 20, 11, 4, 0}

Conclusion:

Now p(n) = y(n), same signal
This means auto-correlation of x(n-1) is same as auto-correlation of x(n)

3. To find q[n] = x[n] ○ x[n-1]

   Input x[n] = {1, 2, 3, 4}

   x[n-1] = {0, 1, 2, 3, 4}

   Output q[n] = {4, 11, 20, 30, **20**, 11, 4, 0}

Conclusion:

Here q(n) = y(n+1)
The resultant is an advanced signal

4. To find r[n] = x[n] ○ x[n-2]

   Input x[n] = {1, 2, 3, 4}

   x[n-2] = {0, 0, 1, 2, 3, 4}

   Output r[n] = {4, 11, 20, 30, 20, **<u>11</u>**, 4, 0, 0}

Conclusion:

Here r(n) = y(n+2)
The resultant is an advanced signal

5. To find s[n] = x[n] ○ a x[n-2] Let a = 0.5

   Input x[n] = {1, 2, 3, 4}

   0.5x [n-2] = {0, 0, 0.5, 1, 1.5, 2}

Output s[n] = {2, 5.5, 10, 15, 10, **<u>5.5</u>**, 2, 0 0}

Conclusion:

Here s(n) = 0.5 y(n+2)
The resultant is scaled and advanced signal

Karl Pearson's Correlation Coefficient

Use of Correlation Coefficient in Recognition

Coefficient of correlation is given by

$$r_{xy} = \frac{\sigma_{xy}}{\sigma_x \ \sigma_y}$$

Where σ_{xy} is covariance and σ_x & σ_x are standard deviation.

$$\sigma_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i - m_x)(y_i - m_y) \qquad \sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - m_x)^2 \qquad \sigma_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - m_y)^2$$

Sample example:

Case 1:
X= (1 2 3 4) and Y= (1 2 3 4)
Here $|r_{xy}| = 1$, **Degree of Similarity is 100%,** *Therefore two signals x[n] and y[n] are exactly same*

Case2:
X= (1, 2, 3, 4) and Y= (5, 6, 7, 8)
Here $|r_{xy}| = 0.58$, **Degree of Similarity is 58 %**
Therefore two signals x[n] and y[n] are NOT Same.

Case 3:
X= (1 2 3 4) and Y= (1.2, 2.6, 3.1, 4.3)
Here $|r_{xy}| = 0.98$, Degree of Similarity is 98 %
Therefore two signals x[n] and y[n] are almost similar.

Conclusion:

1. Auto correlation signal is an even signal
2. If input signal are delayed, then the auto-correlation of delayed signals is same as original signal
3. Cross-correlation of input signal with delayed signal is same as advanced auto-correlation input
4. Application of auto correlation is to find the degree of similarity of two signals

Name: Flavion D'sa   Roll No.: 7371

Program: Discrete Correlation

Code:

```
def cross_correlate(x, y, n1, n2):
    N1 = len(x)
    N2 = len(y)
    mi = n1-(n2+N2-1)
    mf = mi+(N1+N2-2)
    diff = N1-N2+1
    result = []
    for _ in range(N1-N2+1):
        y.append(0)
    for arg in range(mi, mf+1):
        corr_sum = 0
        if arg < 0:
            negative = True
            limit = N1 + arg
        else:
            negative = False
            limit = N1 - arg
        for n in range(1, limit+1):
            if not negative:
                corr_sum += x[n+arg-1]*y[n-diff]
            else:
                corr_sum += x[n-1]*y[n-arg-diff]
        result.append(corr_sum)
    print(result)

if __name__ == '__main__':
    print("=========== CORRELATION ===========")
    print("Enter samples for x(n) :")
    x = [float(temp) for temp in input().split(' ')]
    print("Enter start index for x(n) :")
    n1 = int(input())
    print("Enter samples for y(n) :")
    y = [float(temp) for temp in input().split(' ')]
    print("Enter start index for y(n) :")
    n2 = int(input())
    cross_correlate(x, y, n1, n2)
```

1. Input: x(n) = {1, 2, 3, 4}

```
=========== CORRELATION ===========
Enter samples for x(n) :
1 2 3 4
Enter start index for x(n) :
0
Enter samples for y(n) :
1 2 3 4
Enter start index for y(n) :
0
[4.0, 11.0, 20.0, 30.0, 20.0, 11.0, 4.0]
```

2. Input: x(n) = {1, 2, 3, 4}   x(n-1) = {0, 1, 2, 3, 4}

```
=========== CORRELATION ===========
Enter samples for x(n) :
1 2 3 4
Enter start index for x(n) :
0
Enter samples for y(n) :
0 1 2 3 4
Enter start index for y(n) :
0
[0, 4.0, 11.0, 20.0, 30.0, 20.0, 11.0, 4.0]
```

3. Input: x(n) = {1, 2, 3, 4}   x(n-2) = {0, 0, 1, 2, 3, 4}

```
=========== CORRELATION ===========
Enter samples for x(n) :
1 2 3 4
Enter start index for x(n) :
0
Enter samples for y(n) :
0 0 1 2 3 4
Enter start index for y(n) :
0
[0, 0, 4.0, 11.0, 20.0, 30.0, 20.0, 11.0, 4.0]
```

4. Input: x(n) = {1, 2, 3, 4}   x(n-2) = {0, 0, 0.5, 1, 1.5, 2}

```
=========== CORRELATION ===========
Enter samples for x(n) :
1 2 3 4
Enter start index for x(n) :
0
Enter samples for y(n) :
0 0 0.5 1 1.5 2
Enter start index for y(n) :
0
[0, 0, 2.0, 5.5, 10.0, 15.0, 10.0, 5.5, 2.0]
```