# Experiment 2 : Discrete Correlation

| | |
|---|---|
| **Name** | Prathamesh Mane |
| **UID no. & Branch** | **2022200078 (B1)** |
| **Experiment No.** | 2 |

| | |
|---|---|
| **AIM:** | The aim of this experiment is To study mathematical operation correlation and measure degree of similarity between two signals. |
| **OBJECTIVE:** | 1. Write a function to find Correlation Operation<br>2. Calculate correlation of a DT signals and verify the results using mathematical formula |
| **INPUT SPECIFICATION:** | 1. Length of first Signal L and signal values.<br>2. Length of second Signal M and signal values. |
| **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] O x[n] Classify the resultant signal( Even / Odd ). Calculate the energy of the signal . Q. What is the significance of value of y[0].<br><br>2. Find auto correlation of delayed input signal. Let p[n]= x[n-1] O x[n-1]. Compare the resultant signal p[n] with y[n]. Give your conclusion.<br><br>3. Find cross correlation of input signal and delayed input signal q[n] = x[n] 0 x[n-1]. Compare the resultant signal q[n] with p[n] and y[n] Give your conclusion.<br><br>4. Find cross correlation of input signal and scaled input signal. Let s[n] = x[n] O a x[n-2] where “a” is any constant.Compare the resultant signals.Give your conclusion. |
| **THEORY:** | 1. **Auto-Correlation of a Signal**<br><br>2. Definition: Auto-correlation is a mathematical tool used to measure the similarity between a signal and a time-shifted version of itself. It is often used to detect repeating patterns or periodic signals hidden within noise.<br>3. Formula: The auto-correlation ( Rx[n] ) of a signal ( x[n] ) is given by: |

$$R_x[n] = \sum_{m=-\infty}^{\infty} x[m] \cdot x[m-n]$$

4. Significance of ( Rx[0] ): The value of the auto-correlation function at ( n = 0 ), ( Rx[0] ), represents the total energy of the signal. It indicates how much the signal correlates with itself at zero lag. A higher value indicates a stronger similarity and higher energy.
5. Even/Odd Classification: The auto-correlation function of a real-valued signal is always even, meaning ( Rx[n] = Rx[-n] ). This is because the correlation of a signal with a time-shifted version of itself is symmetric.

**2. Auto-Correlation of a Delayed Signal**

1. Delayed Signal: If the signal is delayed by ( d ) units, the delayed signal is ( x[n-d] ).
2. Auto-Correlation of Delayed Signal: The auto-correlation function for a delayed signal ( p[n] = x[n-1] . x[n-1] ) can be computed similarly as:

$$R_p[n] = \sum_{m=-\infty}^{\infty} x[m-1] \cdot x[m-1-n]$$

-

3. Comparison with Original Signal: The auto-correlation of a delayed signal will have the same shape and energy as the original signal but shifted. The delay doesn't change the energy or the even/odd nature of the signal.

**3. Cross-Correlation of Signals**

1. Definition: Cross-correlation measures the similarity between two different signals as one is shifted in time relative to the other.
2. Formula: The cross-correlation ( R{xy}[n] ) between two signals ( x[n] ) and ( y[n] ) is:

$$R_{xy}[n] = \sum_{m=-\infty}^{\infty} x[m] \cdot y[m-n]$$

3. Cross-Correlation of Input and Delayed Input: The cross-correlation of the input signal and its delayed version, ( q[n] = x[n] * x[n-1] ),

| | |
|---|---|
| | will typically resemble the auto-correlation function but will not be symmetric unless the signals are identical.<br>4. Comparison with Auto-Correlation: Cross-correlation can be asymmetric and provides information on how two signals are related as one is shifted relative to the other. The result can indicate the delay or phase difference between the signals. |

**Theoretical solution using formula**

**X[n]={ 6 , 12 , 7 , 14 }**

**Solution :**
**Case 1: y[n] = x[n] o x[n]**

Hence by using the formula

x = [6, 12, 7, 14]
y = [6, 12, 7, 14]

1. Correlation for ( k = -3 ):
   Shift ( y ) by 3 to the left (introduce 3 leading zeros):

   rxy(-3) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 6)
   rxy(-3) = 0 + 0 + 0 + 84 = 84

2. Correlation for ( k = -2 ):
   Shift ( y ) by 2 to the left (introduce 2 leading zeros):

   rxy(-2) = (6 * 0) + (12 * 0) + (7 * 6) + (14 * 12)
   rxy(-2) = 0 + 0 + 42 + 168 = 210

3. Correlation for ( k = -1 ):
   Shift ( y ) by 1 to the left (introduce 1 leading zero):

   rxy(-1) = (6 * 0) + (12 * 6) + (7 * 12) + (14 * 7)
   rxy(-1) = 0 + 72 + 84 + 98 = 254

4. Correlation for ( k = 0 ):
   No shift (original sequences):

   rxy(0) = (6 * 6) + (12 * 12) + (7 * 7) + (14 * 14)
   rxy(0) = 36 + 144 + 49 + 196 = 425

5. Correlation for ( k = 1 ):
   Shift ( y ) by 1 to the right:

   rxy(1) = (6 * 12) + (12 * 7) + (7 * 14) + (14 * 0)
   rxy(1) = 72 + 84 + 98 + 0 = 254

6. Correlation for ( k = 2 ):
   Shift ( y ) by 2 to the right:

   rxy(2) = (6 * 7) + (12 * 14) + (7 * 0) + (14 * 0)
   rxy(2) = 42 + 168 + 0 + 0 = 210

7. Correlation for ( k = 3 ):
   Shift ( y ) by 3 to the right:

   rxy(3) = (6 * 14) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(3) = 84 + 0 + 0 + 0 = 84

Final Correlation Sequence for ( k = -3 ) to ( k = 3 ):
**rxy = [84, 210, 254, 425, 254, 210, 84]**

**Case 2 : p[n] = x[n-1] o x[n-1]**

x = [0, 6, 12, 7, 14]
y = [0, 6, 12, 7, 14]

1. Correlation for( k = -4 ):
   Shift ( y ) by 4 to the left (introduce 4 leading zeros):

   rxy(-4) = (0 * 0) + (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(-4) = 0 + 0 + 0 + 0 + 0 = 0

2. Correlation for ( k = -3 ):
   Shift ( y\) by 3 to the left (introduce 3 leading zeros):

   rxy(-3) = (0 * 0) + (6 * 0) + (12 * 0) + (7 * 0) + (14 * 6)
   rxy(-3) = 0 + 0 + 0 + 0 + 84 = 84

3. Correlation for ( k = -2 ):
   Shift ( y ) by 2 to the left (introduce 2 leading zeros):

   rxy(-2) = (0 * 0) + (6 * 0) + (12 * 0) + (7 * 6) + (14 * 12)
   rxy(-2) = 0 + 0 + 0 + 42 + 168 = 210

4. Correlation for ( k = -1 ):
   Shift ( y ) by 1 to the left (introduce 1 leading zero):

   rxy(-1) = (0 * 0) + (6 * 0) + (12 * 6) + (7 * 12) + (14 * 7)
   rxy(-1) = 0 + 0 + 72 + 84 + 98 = 254

5. Correlation for ( k = 0 ):
   No shift (original sequences):

   rxy(0) = (0 * 0) + (6 * 6) + (12 * 12) + (7 * 7) + (14 * 14)
   rxy(0) = 0 + 36 + 144 + 49 + 196 = 425

6. Correlation for ( k = 1 ):
   Shift ( y ) by 1 to the right:

   rxy(1) = (0 * 6) + (6 * 12) + (12 * 7) + (7 * 14) + (14 * 0)
   rxy(1) = 0 + 72 + 84 + 98 + 0 = 254

7. Correlation for ( k = 2 ):
   Shift ( y ) by 2 to the right:

   rxy(2) = (0 * 12) + (6 * 7) + (12 * 14) + (7 * 0) + (14 * 0)
   rxy(2) = 0 + 42 + 168 + 0 + 0 = 210

8. Correlation for( k = 3 ):
   Shift ( y ) by 3 to the right

   rxy(3) = (0 * 7) + (6 * 14) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(3) = 0 + 84 + 0 + 0 + 0 = 84

9. Correlation for ( k = 4 ):
   Shift ( y) by 4 to the right:

   rxy(4) = (0 * 14) + (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(4) = 0 + 0 + 0 + 0 + 0 = 0

Final Correlation Sequence for( k = -4 ) to ( k = 4):
**rxy = [0, 84, 210, 254, 425, 254, 210, 84, 0]**

**Case 3 : q[n] = x[n] o x[n-1]**

x = [6, 12, 7, 14]
y = [0, 6, 12, 7, 14]

1. Correlation for k = -4:
   Shift y by 4 to the left (introduce 4 leading zeros):

   rxy(-4) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(-4) = 0 + 0 + 0 + 0 = 0

2. Correlation for k = -3:
   Shift y by 3 to the left (introduce 3 leading zeros):

   rxy(-3) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 6)
   rxy(-3) = 0 + 0 + 0 + 84 = 84

3. Correlation for k = -2:
   Shift y by 2 to the left (introduce 2 leading zeros):

   rxy(-2) = (6 * 0) + (12 * 0) + (7 * 6) + (14 * 12)
   rxy(-2) = 0 + 0 + 42 + 168 = 210

4. Correlation for k = -1:
   Shift y by 1 to the left (introduce 1 leading zero):

   rxy(-1) = (6 * 0) + (12 * 6) + (7 * 12) + (14 * 7)
   rxy(-1) = 0 + 72 + 84 + 98 = 254

5. Correlation for k = 0:
   No shift (original sequences):

   rxy(0) = (6 * 0) + (12 * 6) + (7 * 12) + (14 * 7)
   rxy(0) = 0 + 72 + 84 + 98 + 171 = 425

6. Correlation for k = 1:
   Shift y by 1 to the right:

   rxy(1) = (6 * 6) + (12 * 12) + (7 * 7) + (14 * 0)
   rxy(1) = 36 + 144 + 49 + 0 = 254

7. Correlation for k = 2:
   Shift y by 2 to the right:

   rxy(2) = (6 * 12) + (12 * 7) + (7 * 14) + (14 * 0)
   rxy(2) = 72 + 84 + 98 + 0 = 254

8. Correlation for k = 3:
   Shift y by 3 to the right:

   rxy(3) = (6 * 7) + (12 * 14) + (7 * 0) + (14 * 0)
   rxy(3) = 42 + 168 + 0 + 0 = 210

9. Correlation for k = 4:
   Shift y by 4 to the right:

   rxy(4) = (6 * 14) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(4) = 84 + 0 + 0 + 0 = 84

Final Correlation Sequence for k = -4 to k = 4:

**rxy = [ 84, 210, 254, 425, 254, 210, 84, 0]**

**Case 4 : r[n] = x[n] o x[n-2]**

x = [6, 12, 7, 14]
y = [0 ,0 , 6, 12, 7, 14]

1. Correlation for k = -5:
   Shift y by 5 to the left (introduce 5 leading zeros):

   rxy(-5) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(-5) = 0 + 0 + 0 + 0 = 0

2. Correlation for k = -4:
   Shift y by 4 to the left (introduce 4 leading zeros):

   rxy(-4) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 6)
   rxy(-4)= 0 + 0 + 0 + 84 = 84

3. Correlation for k = -3:
   Shift y by 3 to the left (introduce 3 leading zeros):

   rxy(-3) = (6 * 0) + (12 * 0) + (7 * 6) + (14 * 12)
   rxy(-3) = 0 + 0 + 42 + 168 = 210

4. Correlation for k = -2:
   Shift y by 2 to the left (introduce 2 leading zeros):

   rxy(-2) = (6 * 0) + (12 * 6) + (7 * 12) + (14 * 7)
   rxy(-2) = 0 + 72 + 84 + 98 = 254

5. Correlation for k = -1:
   Shift y by 1 to the left (introduce 1 leading zero):

   rxy(-1) = (6 * 6) + (12 * 12) + (7 * 7) + (14 * 0)
   rxy(-1) = 36 + 144 + 49 + 0 = 254

6. Correlation for k = 0:
   No shift (original sequences):

   rxy(0) = (6 * 12) + (12 * 7) + (7 * 14) + (14 * 0)
   rxy(0) = 72 + 84 + 98 + 0 = 425

7. Correlation for k = 1:
   Shift y by 1 to the right:

   rxy(1) = (6 * 7) + (12 * 14) + (7 * 0) + (14 * 0)
   rxy(1) = 42 + 168 + 0 + 0 = 210

8. Correlation for k = 2:
   Shift y by 2 to the right:

   rxy(2) = (6 * 14) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(2) = 84 + 0 + 0 + 0 = 84

9. Correlation for k = 3:
   Shift y by 3 to the right (introduce 3 trailing zeros):

   rxy(3) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(3) = 0 + 0 + 0 + 0 = 0

10. Correlation for k = 4:
    Shift y by 4 to the right (introduce 4 trailing zeros):

    rxy(4) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
    rxy(4) = 0 + 0 + 0 + 0 = 0

11. Correlation for k = 5:
    Shift y by 5 to the right (introduce 5 trailing zeros):

    rxy(5) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
    rxy(5) = 0 + 0 + 0 + 0 = 0

Final Correlation Sequence for k = -5 to k = 5:

rxy = [0, 0, 84, 210, 254, 425, 254, 210, 84, 0, 0]

Case 5 : s[n] = x[n] o x[n-2]

x = [6, 12, 7, 14]
y = [0, 0, 12, 24, 14, 28]

1. Correlation for k = -5:
   Shift y by 5 to the left (introduce 5 leading zeros):

   rxy(-5) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(-5) = 0 + 0 + 0 + 0 = 0

2. Correlation for k = -4:
   Shift y by 4 to the left (introduce 4 leading zeros):

   rxy(-4) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 12)
   rxy(-4) = 0 + 0 + 0 + 168 = 168

3. Correlation for k = -3:
   Shift y by 3 to the left (introduce 3 leading zeros):

   rxy(-3) = (6 * 0) + (12 * 0) + (7 * 12) + (14 * 24)
   rxy(-3) = 0 + 0 + 84 + 336 = 420

4. Correlation for k = -2:
   Shift y by 2 to the left (introduce 2 leading zeros):

   rxy(-2) = (6 * 0) + (12 * 12) + (7 * 24) + (14 * 14)
   rxy(-2) = 0 + 144 + 168 + 196 = 508

5. Correlation for k = -1:
   Shift y by 1 to the left (introduce 1 leading zero):

   rxy(-1) = (6 * 12) + (12 * 24) + (7 * 14) + (14 * 28)
   rxy(-1) = 72 + 288 + 98 + 392 = 850

6. Correlation for k = 0:
   No shift (original sequences):

   rxy(0) = (6 * 24) + (12 * 14) + (7 * 28) + (14 * 0)
   rxy(0) = 144 + 168 + 196 + 0 = 508

7. Correlation for k = 1:
   Shift y by 1 to the right:

   rxy(1) = (6 * 14) + (12 * 28) + (7 * 0) + (14 * 0)
   rxy(1) = 84 + 336 + 0 + 0 = 420

8. Correlation for k = 2:
   Shift y by 2 to the right:

   rxy(2) = (6 * 28) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(2) = 168 + 0 + 0 + 0 = 168

9. Correlation for k = 3:
   Shift y by 3 to the right (introduce 3 trailing zeros):

   rxy(3) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
   rxy(3) = 0 + 0 + 0 + 0 = 0

10. Correlation for k = 4:
    Shift y by 4 to the right (introduce 4 trailing zeros):

    rxy(4) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
    rxy(4) = 0 + 0 + 0 + 0 = 0

11. Correlation for k = 5:
    Shift y by 5 to the right (introduce 5 trailing zeros):

    rxy(5) = (6 * 0) + (12 * 0) + (7 * 0) + (14 * 0)
    rxy(5) = 0 + 0 + 0 + 0 = 0

Final Correlation Sequence for k = -5 to k = 5:

**rxy = [0, 168, 420, 508, 850, 508, 420, 168, 0, 0, 0]**

| Developing algorithm using programming language | |
|---|---|
| **PROBLEM STATEMENT:** | Write a program in any programming language to perform correlation operation of two signals with length L and M respectively. Also showcase the different cases as mentioned in the problem statement. |
| **PROGRAM:** | (see code below) |

```
#include <stdio.h>

int main() {
    const int MAX_SIZE = 100;

    // Input the size of the first signal
    int n1;
    printf("Enter the number of discrete points in the first signal (n1): ");
    scanf("%d", &n1);

    // Declare array to hold the first signal
    double x[MAX_SIZE];

    // Input the first signal
    printf("Enter the first signal (x): ");
    for (int i = 0; i < n1; ++i) {
        scanf("%lf", &x[i]);
    }

    // Input the size of the second signal
    int n2;
    printf("Enter the number of discrete points in the second signal (n2): ");
    scanf("%d", &n2);

    // Declare array to hold the second signal
    double y[MAX_SIZE];

    // Input the second signal
    printf("Enter the second signal (y): ");
    for (int i = 0; i < n2; ++i) {
        scanf("%lf", &y[i]);
    }

    // Compute the correlation
    int result_length = n1 + n2 - 1;
    double correlation[MAX_SIZE * 2 - 1] = {0};

    for (int k = -(n2 - 1); k < n1; ++k) {
        double sum = 0;
        for (int i = 0; i < n1; ++i) {
```

```
            int j = i - k;
            if (j >= 0 && j < n2) {
                sum += x[i] * y[j];
            }
        }
        correlation[k + (n2 - 1)] = sum;
    }

    // Find the maximum value in the correlation array
    double max_correlation = correlation[0];
    for (int i = 1; i < result_length; ++i) {
        if (correlation[i] > max_correlation) {
            max_correlation = correlation[i];
        }
    }

    // Output the correlation and the maximum value
    printf("\nCorrelation result y[n]: ");
    for (int i = 0; i < result_length; ++i) {
        printf("%.2f   ", correlation[i]);
    }
    printf("\n");

    printf("\nMaximum correlation value: %.2f\n", max_correlation);
    printf("Hence n=0 is at : %.2f\n", max_correlation);

    return 0;
}
```

**RESULT:**

**CASE 1 : AUTO CORRELATION [x[n]]**

```
/tmp/wpd6z48hmU.o
Enter the number of discrete points in the first signal (n1): 4
Enter the first signal (x): 6 12 7 14
Enter the number of discrete points in the second signal (n2): 4
Enter the second signal (y): 6 12 7 14

Correlation result y[n]:
84.00   210.00   254.00   425.00   254.00   210.00   84.00

Maximum correlation value: 425.00
Hence n=0 is at : 425.00
```

**CASE 2 : AUTO CORRELATION OF SHIFTED SIGNAL [x[n-1]]**

```
Enter the number of discrete points in the first signal (n1): 5
Enter the first signal (x): 0 6 12 7 14
Enter the number of discrete points in the second signal (n2): 5
Enter the second signal (y): 0 6 12 7 14

Correlation result y[n]:
0.00   84.00   210.00   254.00   425.00   254.00   210.00   84.00   0.00

Maximum correlation value: 425.00
Hence n=0 is at : 425.00
```

**CASE 3 : CORRELATION OF ORIGINAL SIGNAL AND SHIFTED SIGNAL BY (1)**

```
Enter the number of discrete points in the first signal (n1): 4
Enter the first signal (x): 6 12 7 14
Enter the number of discrete points in the second signal (n2): 5
Enter the second signal (y): 0 6 12 7 14

Correlation result y[n]:
84.00   210.00   254.00   425.00   254.00   210.00   84.00   0.00

Maximum correlation value: 425.00
```

**CASE 4 : CORRELATION OF ORIGINAL SIGNAL AND SHIFTED SIGNAL BY (2)**

```
Enter the number of discrete points in the first signal (n1): 4
Enter the first signal (x): 6 12 7 14
Enter the number of discrete points in the second signal (n2): 6
Enter the second signal (y): 0 0 6 12 7 14

Correlation result y[n]:
84.00   210.00   254.00   425.00   254.00   210.00   84.00   0.00   0.00

Maximum correlation value: 425.00
```

**CASE 5 : CORRELATION OF ORIGNAL SIGNAL AND SCALED SHIFTED SIGNAL**

```
Enter the number of discrete points in the first signal (n1): 4
Enter the first signal (x): 6 12 7 14
Enter the number of discrete points in the second signal (n2): 6
Enter the second signal (y): 0 0 12 24 14 28

Correlation result y[n]:
168.00   420.00   508.00   850.00   508.00   420.00   168.00   0.00   0.00


Maximum correlation value: 850.00
```

| | |
|---|---|
| **CONCLUSION:** | • The value at n=0 represents the energy of the original signal. This is because it is the sum of the squares of the signal's values.Therefore, y[0] gives us the total energy of the signal.<br>• The auto correlation of x[n-1] is same as x[n]<br>• The result obtained by correlation of x[n] and x[n-1] is advanced by 1 unit . and the zero is shifted to right side.<br>• The result obtained by correlation of x[n] and x[n-m] is advanced by m unit .<br>• The result obtained by correlation of x[n] and a*x[n-1] is scaled by a units . |

| | |
|---|---|
| | • Delay and scaling impact the correlation functions but don’t change the fundamental nature of the signals, like energy or symmetry. The analysis of these factors is crucial in many practical applications such as communication systems, signal processing, and time series analysis.<br><br>• Cross-correlation of input signal with delayed signal is same as advanced autocorrelated input signal. |