

EXPERIMENT-1

Theme: Convolution and Correlation

Name: Anupama Kulshreshtha

Roll no.: EE22BTECH11009

Simulate convolution and correlation between two signals in Matlab and C. Take input as

x= {0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631}
h= {0.7147 -0.2050 -0.1241 1.4897 1.4090}

(a)MATLAB:

Code:

```
clc
clear
close all

% Define two signals
x = [0.3426      3.5784      2.7694      -1.3499      3.0349      0.7254      -0.0631];
h = [0.7147      -0.2050      -0.1241      1.4897      1.4090];

% Call the convolution function
convolutionResult = myConvolution(x, h);
% Call the correlation function
correlationResult = myCorrelation(x, h);

% Display the input and output signals
disp('Input Signal x:')
disp(x)
disp('Input Signal h:')
disp(h)
disp('Convolution Result:')
disp(convolutionResult)
disp('Correlation Result:')
disp(correlationResult)

function result = myConvolution(x,h)
    % Lengths of the signals
    M = length(x);
    N = length(h);

    % Length of the result signal
    L = M + N - 1;

    % Initialize the result signal
    result = zeros(1, L);
```

```

% Perform convolution
for n = 1:L
    for k = max(1, n-N+1):min(n, M)
        result(n) = result(n) + x(k) * h(n-k+1);
    end
end
end

function result = myCorrelation(x,h)
% Lengths of the signals
M = length(x);
N = length(h);

% Length of the result signal
L = M + N - 1;

% Initialize the result array
result = zeros(1, L);
result1 = zeros(1,L);

% Compute cross-correlation
for n = (1-M):(N-1)
    for k = 1:M
        if (n+k>=1 && n+k<=N)
            result1(n+M) = result1(n+M) + x(k)*h(n+k);
        end
    end
end
% Reverse the array
for i = 1:L
    result(i) = result1(L-i+1);
end
end

```

Output:

```

Command Window

Input Signal x:
    0.3426    3.5784    2.7694   -1.3499    3.0349    0.7254   -0.0631

Input Signal h:
    0.7147   -0.2050   -0.1241    1.4897    1.4090

Convolution Result:
Columns 1 through 10
    0.2449    2.4872    1.2032   -1.4662    7.9156    9.2314    1.3207    2.5420    5.3646    0.9281

Column 11
   -0.0889

Correlation Result:
Columns 1 through 10
    0.4827    5.5523    9.1903    1.7093    1.4328    7.7005    2.8711   -1.7710    2.0282    0.5314

Column 11
   -0.0451

fx >>

```

(b) C:

Code:

File: main.c

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    double signal1[] = {0.3426, 3.5784, 2.7694, -1.3499, 3.0349, 0.7254, -0.0631};
    double signal2[] = {0.7147, -0.2050, -0.1241, 1.4897, 1.4090};

    int M = sizeof(signal1)/sizeof(signal1[0]);
    int N = sizeof(signal2)/sizeof(signal2[0]);
    int L = M+N-1;

    double convolution_result[L];
    double correlation_result[L];

    convolution(signal1, M, signal2, N, convolution_result);
    correlation(signal1, M, signal2, N, correlation_result);

    printf("Signal 1: ");
    printArray(signal1, M);
    printf("Signal 2: ");
    printArray(signal2, N);

    printf("Convolution Result: ");
    printArray(convolution_result, L);

    printf("Correlation Result: ");
    printArray(correlation_result, L);

    return 0;
}
```

File: functions.c

```
#include <stdio.h>
```

```
void convolution(double *x, int M, double *h, int N, double *result){  
    int i, j;  
    int L = M+N-1;  
    for(i=0; i < L; i++){  
        result[i] = 0;  
        for(j=0; j< N; j++){  
            if(i - j >= 0 && i - j < M){  
                result[i] += x[i - j] * h[j];  
            }  
        }  
    }  
}
```

```
void correlation(double *x, int M, double *h, int N, double *result) {  
    // Length of the result signal  
    int L = M + N - 1;  
    double result1[L];  
    for(int i=0; i<L; i++){  
        result[i] = 0.0;  
        result1[i] = 0.0;  
    }  
    // Compute cross-correlation  
    for (int n = 1 - M; n < N; n++) {  
        for (int k = 0; k < M; k++) {  
            if (n + k >= 0 && n+k<N) {  
                result1[n + M - 1] += x[k] * h[n + k];  
            }  
        }  
    }  
    //Reversing the array  
    for(int j=0; j<L; j++){  
        result[j] = result1[L-j-1];  
    }
```

```

    }
}

void printArray(double array[], int size){
    for(int i=0; i<size; i++){
        printf("%f ", array[i]);
    }
    printf("\n");
}

```

Output:

```

Signal 1: 0.342600 3.578400 2.769400 -1.349900 3.034900 0.725400 -0.063100
Signal 2: 0.714700 -0.205000 -0.124100 1.489700 1.409000
Convolution Result: 0.244856 2.487249 1.203202 -1.466209 7.915556 9.231352 1.320703 2.541995 5.364633 0.928089 -0.088908
Correlation Result: 0.482723 5.552337 9.190310 1.709254 1.432830 7.700457 2.871109 -1.770950 2.028167 0.531379 -0.045098

Process returned 0 (0x0)   execution time : 0.052 s
Press any key to continue.

```