EXPERIMENT-4

Theme: Fixed point Convolution and Correlation

Name: Anupama Kulshreshtha

Roll no.: EE22BTECH11009

Simulate fixed point 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}

Find out the mean square error for fixed point outputs and plot them.

Aim: To simulate fixed point convolution and correlation between the given signals and find out the mean square errors.

Implementation Steps:

1) Functions were made for convolution, correlation, fixed point addition, fixed point multiplication, fixed-point convolution and fixed-point correlation.
2) First, convolution and correlation were performed on the given signals using the myConvolution and myCorrelation functions.
3) For fixed point convolution and correlation, the fixed-point addition and fixed-point multiplication functions were called instead of the usual + and * operators, whenever the numbers were supposed to be added or multiplied.
4) The fixed-point addition and fixed-point multiplication functions would first convert the given floating point to fixed point, perform the required operation, then give the result in the form of floating point.
5) Finally, the mean square error for both convolution and correlation was found out using the formula $\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{e}[\mathrm{i}])^2$, where e[i] is the difference between value obtained by convolution/correlation and fixed-point convolution/correlation for separate indices.

(a) MATLAB:

Code:

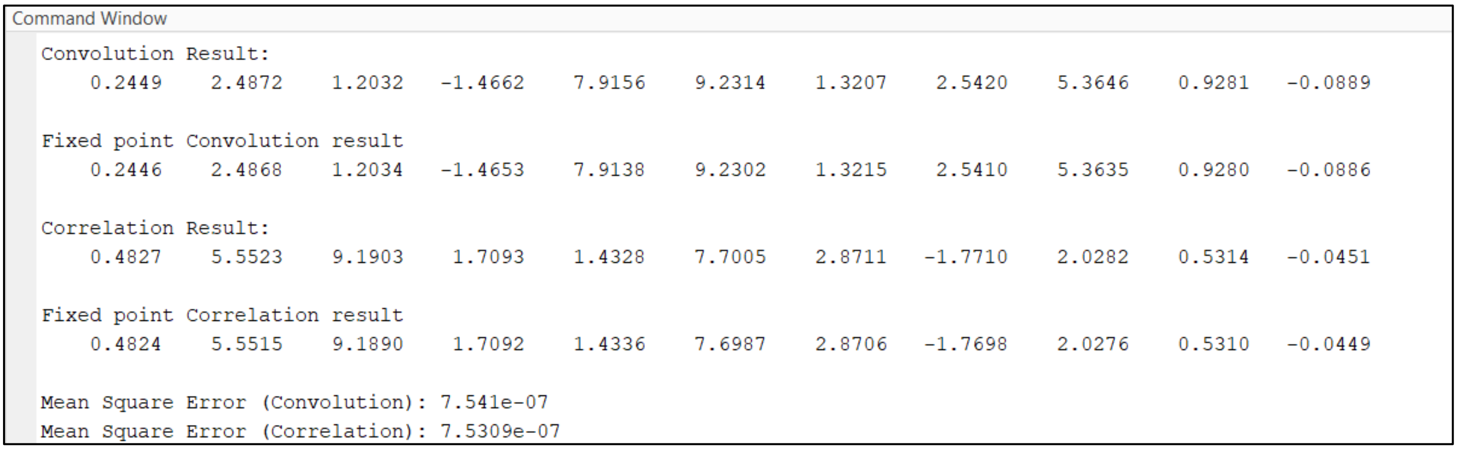
```
clc
clear
```

```
close all

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];
q = 12;

conv = myConvolution(x,h);
corr = myCorrelation(x,h);
fixed_convolve = myConvolution_fixed(x, h);
fixed_correlation = myCorrelation_fixed(x, h);

mse_conv = mean_square_error(conv, fixed_convolve);
mse_corr = mean_square_error(corr, fixed_correlation);

disp("Convolution Result:")
disp(conv)
disp('Fixed point Convolution result')
disp(fixed_convolve)
disp("Correlation Result:")
disp(corr)
disp('Fixed point Correlation result')
disp(fixed_correlation)
disp(['Mean Square Error (Convolution): ', num2str(mse_conv)]);
disp(['Mean Square Error (Correlation): ', num2str(mse_corr)]);

% Compute absolute error
abs_error_convolve = abs(conv - fixed_convolve);
abs_error_correlation = abs(corr - fixed_correlation);

% Plot
figure;
subplot(2,1,1);
stem(abs_error_convolve);
xlabel('Index');
ylabel('Absolute Error');
title('Absolute Error Between Convolution and Fixed-point Convolution');

subplot(2,1,2);
stem(abs_error_correlation);
xlabel('Index');
ylabel('Absolute Error');
title('Absolute Error Between Correlation and Fixed-point Correlation');


function result = fixed_add(x1,x2,q)
% Converting floating to fixed
x1_fixed = fix(x1*2^q);
x2_fixed = fix(x2*2^q);
%disp(x1_fixed)
%disp(x2_fixed)
sum = x1_fixed + x2_fixed;
% Converting fixed to floating
result = sum/2^q;
end

function result = fixed_multiply(x1, x2, q)
% Converting floating to fixed
x1_fixed = fix(x1*2^q);
x2_fixed = fix(x2*2^q);

product = (x1_fixed*x2_fixed)/2^q;
% Converting fixed to floating
result = product/2^q;
end
```

```matlab
% For convolution of 2 signals
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

% For correlation of 2 signals
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
    for i = 1:L
        result(i) = result1(L-i+1);
    end
end

% For fixed point convolution of 2 signals
function result = myConvolution_fixed(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) = fixed_add(result(n),fixed_multiply(x(k),h(n-k+1),12),12);
        end
    end
end

% For fixed point correlation of 2 signals
```

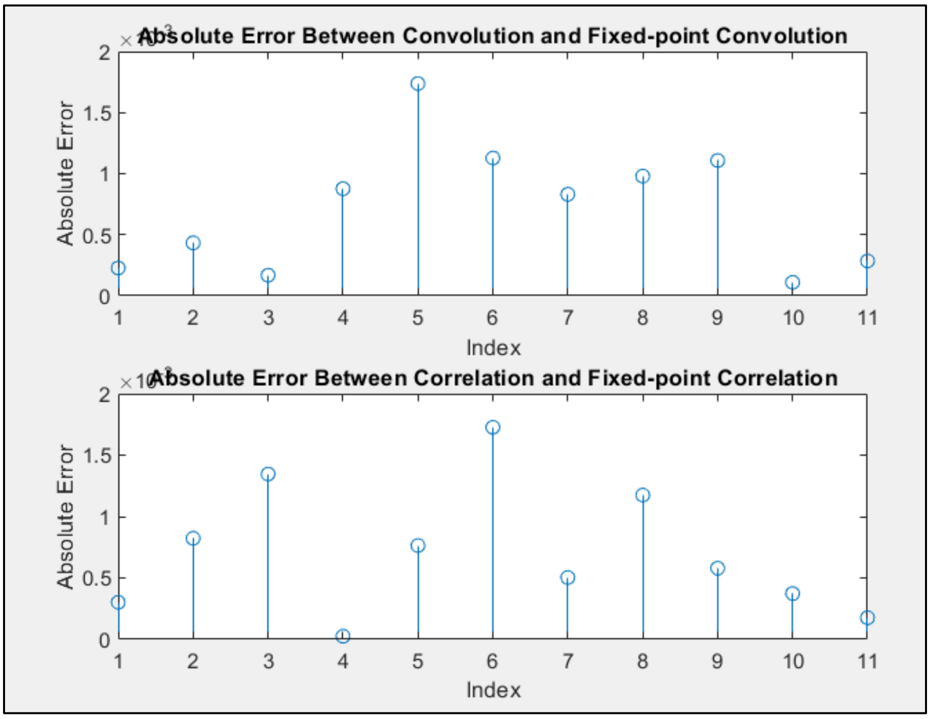
```
function result = myCorrelation_fixed(x,h)
    % Lengths of the signals
    M = length(x);
    N = length(h);

    % 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) = fixed_add(result1(n+M), fixed_multiply(x(k),h(n+k),12),12);
            end
        end
    end
    for i = 1:L
        result(i) = result1(L-i+1);
    end
end

% To  compute mean square error of 2 signals
function result = mean_square_error(a,b)
    n = length(a);
    k = zeros(1,n);
    for i = 1:n
        k(i) = (a(i)-b(i))^2;
    end

    sum = 0;
    for j = 1:n
        sum = sum + k(j);
    end
    result = sum/n;
end
```

Output:



```
Convolution Result:
    0.2449    2.4872    1.2032   -1.4662    7.9156    9.2314    1.3207    2.5420    5.3646    0.9281   -0.0889

Fixed point Convolution result
    0.2446    2.4868    1.2034   -1.4653    7.9138    9.2302    1.3215    2.5410    5.3635    0.9280   -0.0886

Correlation Result:
    0.4827    5.5523    9.1903    1.7093    1.4328    7.7005    2.8711   -1.7710    2.0282    0.5314   -0.0451

Fixed point Correlation result
    0.4824    5.5515    9.1890    1.7092    1.4336    7.6987    2.8706   -1.7698    2.0276    0.5310   -0.0449

Mean Square Error (Convolution): 7.541e-07
Mean Square Error (Correlation): 7.5309e-07
```

Plot:



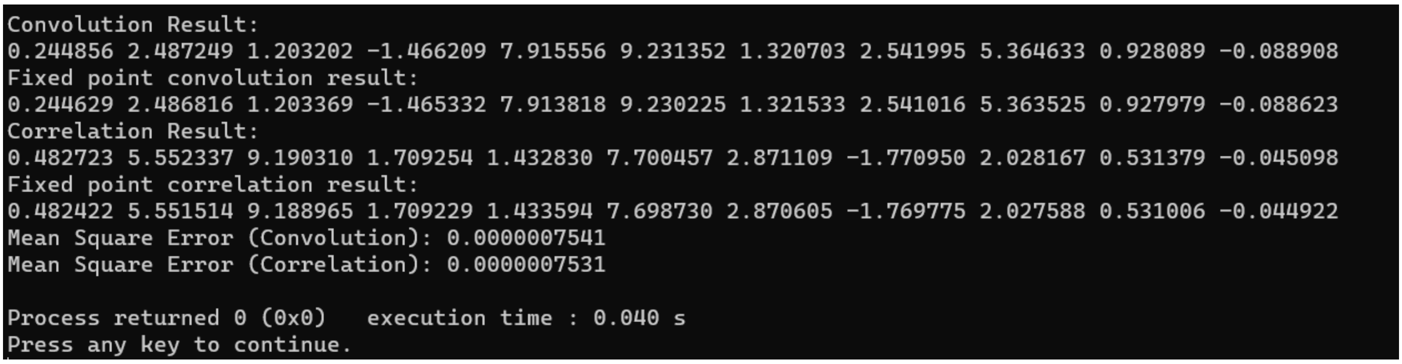
(b) C:

Code:

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

#define q 12

double fixed_add(double x1, double x2){
    int x1_fixed = x1*pow(2,q);
    int x2_fixed = x2*pow(2,q);
    int sum = x1_fixed + x2_fixed;
    return (double)sum/pow(2,q);
}

double fixed_multiply(double x1, double x2){
    int x1_fixed = x1*pow(2,q);
    int x2_fixed = x2*pow(2,q);

    double product = (x1_fixed*x2_fixed)/pow(2,q);
    return (double)product/pow(2,q);
}

void myConvolution(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 myCorrelation(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 myConvolution_fixed(const double *x, int M, const 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] = fixed_add(result[i], fixed_multiply(x[i-j],h[j]));
            }
        }
    }
}

void myCorrelation_fixed(const double *x, int M, const 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] = fixed_add(result1[n+M-1], fixed_multiply(x[k],
h[n+k]));
            }
        }
    }
    //Reversing the array
    for(int j=0; j<L; j++){
        result[j] = result1[L-j-1];
    }
}

double mean_square_error(const double *a, const double *b, int n) {
    double k[n];
    for (int i = 0; i < n; i++) {
        k[i] = (a[i] - b[i])*(a[i] - b[i]);
    }
    double sum = 0;
    for(int j=0; j<n;j++){
        sum += k[j];
    }
    return sum / n;
}

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

int main() {
    double x[] = {0.3426, 3.5784, 2.7694, -1.3499, 3.0349, 0.7254, -0.0631};
    double h[] = {0.7147, -0.2050, -0.1241, 1.4897, 1.4090};
    int M = sizeof(x) / sizeof(x[0]);
    int N = sizeof(h) / sizeof(h[0]);

    double conv[M + N - 1];
    double corr[M + N - 1];
    double fixed_convolve[M + N - 1];
    double fixed_correlation[M + N - 1];

    myConvolution(x, M, h, N, conv);
    myCorrelation(x, M, h, N, corr);
    myConvolution_fixed(x, M, h, N, fixed_convolve);
    myCorrelation_fixed(x, M, h, N, fixed_correlation);

    double mse_conv = mean_square_error(conv, fixed_convolve, M + N - 1);
    double mse_corr = mean_square_error(corr, fixed_correlation, M + N - 1);

    printf("Convolution Result:\n");
    printArray(conv, M+N-1);
    printf("Fixed point convolution result:\n");
    printArray(fixed_convolve, M+N-1);
    printf("Correlation Result:\n");
    printArray(corr, M+N-1);
    printf("Fixed point correlation result:\n");
    printArray(fixed_correlation, M+N-1);
    printf("Mean Square Error (Convolution): %.10f\n", mse_conv);
    printf("Mean Square Error (Correlation): %.10f\n", mse_corr);

    return 0;
}
```

Output:



```
Convolution Result:
0.244856 2.487249 1.203202 -1.466209 7.915556 9.231352 1.320703 2.541995 5.364633 0.928089 -0.088908
Fixed point convolution result:
0.244629 2.486816 1.203369 -1.465332 7.913818 9.230225 1.321533 2.541016 5.363525 0.927979 -0.088623
Correlation Result:
0.482723 5.552337 9.190310 1.709254 1.432830 7.700457 2.871109 -1.770950 2.028167 0.531379 -0.045098
Fixed point correlation result:
0.482422 5.551514 9.188965 1.709229 1.433594 7.698730 2.870605 -1.769775 2.027588 0.531006 -0.044922
Mean Square Error (Convolution): 0.0000007541
Mean Square Error (Correlation): 0.0000007531

Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.
```

Observations and Understanding:

The mean square error for both convolution and correlation is of the order 10^{-7}. Using the same method of implementation for both matlab and C, we observe that the mse values for both convolution and correlation in the output of both the codes are equal and close to 7.5 * 10^{-7}. Also, we have used q = 12, as the number of fractional bits for the fixed_to_floating and floating_to_fixed conversion. As we take a larger value of q, the error will further decrease to an even smaller value.