# DSP- LAB

# EXPERIMENT-4

**NAME – AMAN KUMAR**

**ROLL NO – EE22BTECH11006**

## AIM –

1. To implement fixed-point convolution and correlation in MATLAB and C
2. Evaluating the Mean square Error between the fixed-point outputs and the without fixed-point convolution and correlation.
3. Plotting the absolute error.

## Procedure:

- Convert the given x[ ] and h[ ] signals into the fixed point numbers
- We first define the Q value which represents the number of fractional bits in the fixed-point.
- Then we multiply the whole array by 2^Q to get the fixed-point numbers
- Then feeding the fixed-point array for the convolution and correlation calculations
- Then converting the output back to floating point by scale it down by dividing it by (2^(2Q))
- Then finding the absolute error between each element of fixed point convolution/correlation and the float point convolution correlation
- The for the Mean square error we will sum all the errors divide it by the length of the array.

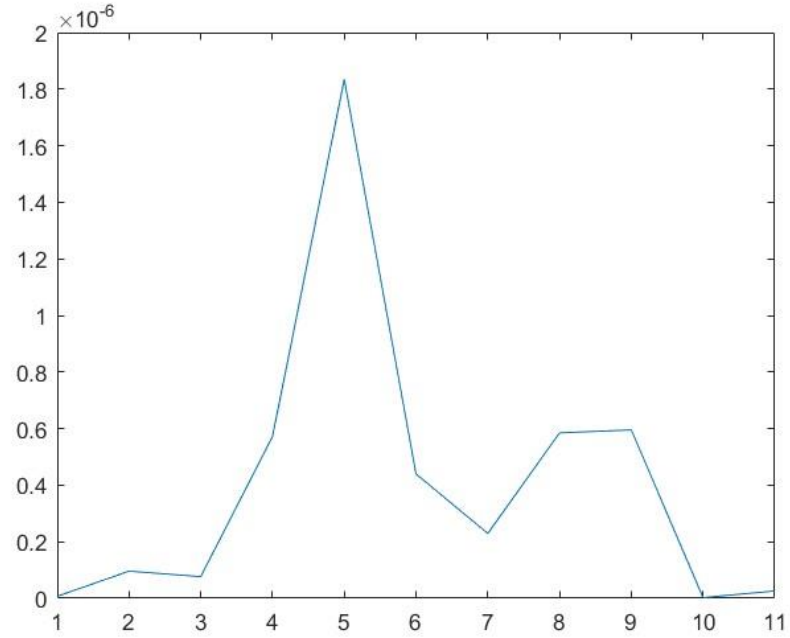
## MATLAB code for convolution:

```
x1 = [0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631];
x2 = [0.7147 -0.2050 -0.1241 1.4897 1.4090];
m = length(x1);
n = length(x2);
scale_factor = 2^12;

fixed_x1 = fixed_point(x1, scale_factor);%converting x1,x2 into fixed points
fixed_x2 = fixed_point(x2, scale_factor);

ans_conv = conv_fix(fixed_x1, fixed_x2, m, n);%convolution of fixed x1,x2
ans_conv_1 = conv_fix(x1, x2, m, n);%convolution of given x1,x2
float_conv = fix_float(ans_conv, scale_factor);%converting the convolution of
fixed x1,x2 back to float
disp("The convolution of numbers x1 and x2:");
disp(ans_conv_1);
disp('The convolution of fixed numbers is:');
disp(float_conv);

error_2 = abs(float_conv - ans_conv_1).^2;%finding the absolute error
disp('Error:');
disp(error_2);
plot(error_2);

MSE = sum(error_2) / numel(error_2);%MSE value of the error
disp("The MSE value of convolution:");
disp(MSE);
%function for calculating convolution
function output = conv_fix(fixed_x1, fixed_x2, m, n)
    Y = zeros(1, m + n - 1);
    N = [fixed_x1, zeros(1, m - 1)];
    M = [fixed_x2, zeros(1, n - 1)];

    for i = 1:m + n - 1
        for j = 1:m
            if (i - j + 1 > 0 && i - j + 1 <= m)
                Y(i) = Y(i) + N(j) * M(i - j + 1);
            end
        end
    end
    output = Y;
end
%function for converting into fixed points
function x_fixed = fixed_point(x, scale_factor)
    x_fixed = zeros(size(x));
    for i = 1:length(x)
        x_fixed(i) = fix(x(i) * scale_factor);
    end
end
%function for converting it into float point
function x_float = fix_float(x, scale_factor)
    x_float = x / scale_factor^2;
end
```

## RESULT:

```
The convolution of numbers x1 and x2:

  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

The convolution of fixed numbers is:

  Columns 1 through 10

    0.2448    2.4869    1.2035   -1.4655    7.9142    9.2307    1.3212    2.5412    5.3639    0.9281

  Column 11

   -0.0887

Error:

   1.0e-05 *

  Columns 1 through 10

    0.0007    0.0096    0.0077    0.0572    0.1834    0.0439    0.0229    0.0585    0.0595    0.0003

  Column 11

    0.0026

The MSE value of convolution:

   4.0578e-07
```

## Plot for absolute error in convolution:



## C code:

```c
#include <stdio.h>
#define MAX_SIZE 100

//function to calculate convolution
void conv_fixed(int unscale_x1[], int unscale_x2[], int m, int n, int output[]) {
    for(int i = 0; i < m; i++) {
        for(int j = 0; j < n; j++) {
            output[i+j] += unscale_x1[i] * unscale_x2[j];
        }
    }
}

void conv_double(double unscale_x1[], double unscale_x2[], int m, int n, double output[]) {
    for(int i = 0; i < m; i++) {
        for(int j = 0; j < n; j++) {
            output[i+j] += unscale_x1[i] * unscale_x2[j];
        }
    }
}

//function for converting into fixed points
void fixed_point(double x[], int fix[], int scale_factor, int size) {
    for(int i = 0; i < size; i++) {
        fix[i] = (int)(x[i] * scale_factor);
    }
}
//function for converting back to float points
void unscale(int fixed[], double conv_float[], int scale_factor, int size) {
    for(int i = 0; i < size; i++) {
        conv_float[i] = (double)(fixed[i]) / ((scale_factor)*(scale_factor));
    }
}
//function for calculating MSE value
float MSE(double output[], double output_fixed[], int length) {
    float mse = 0.0;
    for(int i = 0; i < length; i++) {
        double err = output[i] - output_fixed[i];
        mse += err * err;
    }
    mse /= length;
    return mse;
}

int main() {
    int m;
    int n;
    double x1[] = {0.3426, 3.5784, 2.7694, -1.3499, 3.0349, 0.7254, -0.0631};
    double x2[] = {0.7147, -0.2050, -0.1241, 1.4897, 1.4090};
    m = sizeof(x1) / sizeof(x1[0]);
    n = sizeof(x2) / sizeof(x2[0]);
    int Q = 12;
    int factor = 1 << Q;
    int output[MAX_SIZE + MAX_SIZE - 1] = {0};
    double output_float[MAX_SIZE + MAX_SIZE - 1] = {0};
    double unscaled_output[MAX_SIZE + MAX_SIZE - 1] = {0};
    int fixed_x1[MAX_SIZE];
    int fixed_x2[MAX_SIZE];

    fixed_point(x1, fixed_x1, factor, m);
```

```
    fixed_point(x2, fixed_x2, factor, n);
    conv_fixed(fixed_x1, fixed_x2, m, n, output);//conv of the fixed numbers
    unscale(output, unscaled_output, factor, m + n - 1);//converting the fixed numbers of conv
to float value
    conv_double(x1, x2, m, n, output_float);// conv of original x1,x2

    printf("The convolved_fixed output is:\n");
    for(int i = 0; i < m + n - 1; i++) {
        printf("%.4f ", unscaled_output[i]);
    }
    printf("\n");

    printf("The convolved_float output is:\n");
    for(int i = 0; i < m + n - 1; i++) {
        printf("%.4f ", output_float[i]);
    }
    printf("\n");

    printf("Absolute Error:\n");
    for(int i = 0; i < m + n - 1; i++) {
        double abs_err = ((unscaled_output[i] - output_float[i])*(unscaled_output[i] -
output_float[i]));
        printf("%E ", abs_err);
    }
    printf("\n");

    float mse = MSE(unscaled_output, output_float, m + n - 1);
    printf("Mean Squared Error: %E\n", mse);

    return 0;
}
```

# Result:

```
The convolved_fixed output is:
0.2448 2.4869 1.2035 -1.4655 7.9142 9.2307 1.3212 2.5412 5.3639 0.9281 -0.0887

The convolved_float output is:
0.2449 2.4872 1.2032 -1.4662 7.9156 9.2314 1.3207 2.5420 5.3646 0.9281 -0.0889

Absolute Error:
7.211110E-009 9.605695E-008 7.656130E-008 5.719258E-007 1.834485E-006 4.394407E-007 2.291396E-
007 5.848687E-007 5.953196E-007 2.514351E-009 2.607305E-008

Mean Squared Error: 4.057814E-007
```

## MATLAB code for correlation:

```
x1 = [0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631];
x2 = [0.7147 -0.2050 -0.1241 1.4897 1.4090];
m = length(x1);
n = length(x2);
scale_factor = 2^12;

fixed_x1 = fixed_point(x1, scale_factor);%converting x1,x2 into fixed point
numbers
fixed_x2 = fixed_point(x2, scale_factor);

ans_corr = corr_fix(fixed_x1, fixed_x2, m, n);%finding correlation of the fixed
point numbers
```

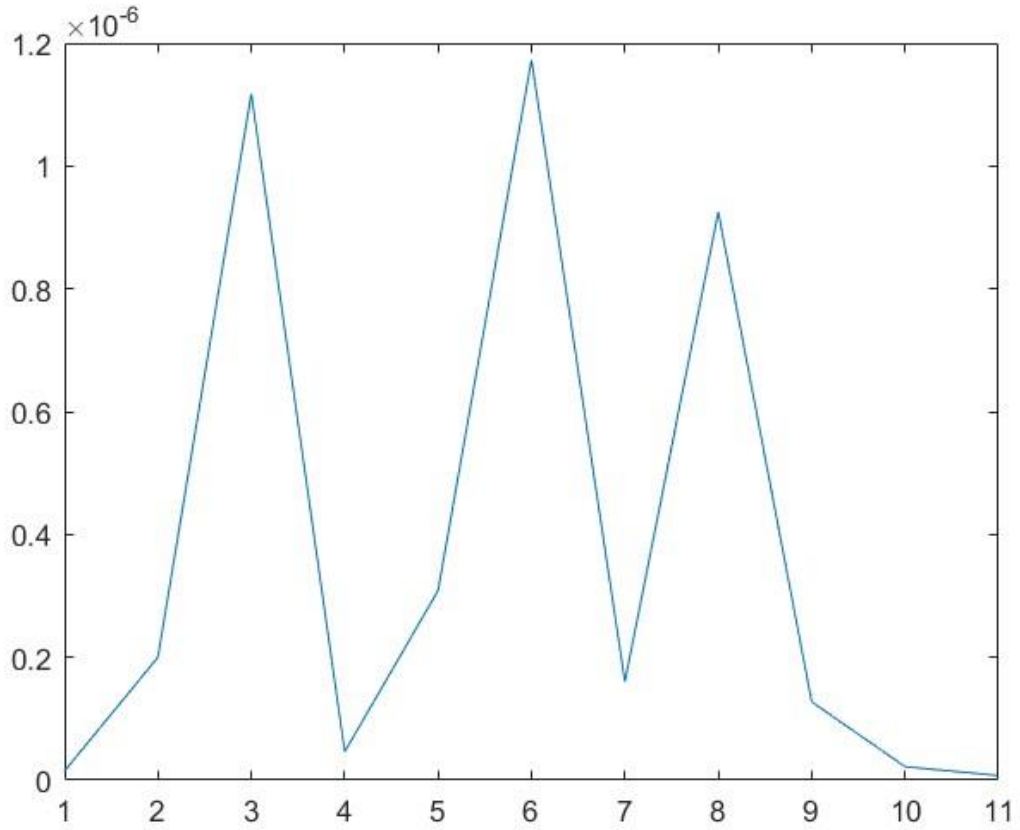
```
ans_corr_1 = corr_fix(x1, x2, m, n);
ans_corr_float=fix_float(ans_corr,scale_factor);%converting the correlation of
fixed point back to float
disp("The correlation of numbers x1 and x2:");
disp(ans_corr_1);
disp('The correlation of fixed numbers is:');
disp(ans_corr_float);

error_corr = abs(ans_corr_float - ans_corr_1).^2;%finding absolute error
disp('Error:');
disp(error_corr);
plot(error_corr);

MSE_corr = sum(error_corr) / numel(error_corr);%finding the mse value
disp("The MSE value of correlation:");
disp(MSE_corr);
%function for correlation
function output = corr_fix(unscale_x1, unscale_x2, m, n)
    Y = zeros(1, m + n - 1);
    N = [unscale_x1, zeros(1, m - 1)];
    M = [unscale_x2(end:-1:1), zeros(1, n - 1)];

    for i = 1:m + n - 1
        for j = 1:m
            if (i - j + 1 > 0 && i - j + 1 <= m)
                Y(i) = Y(i) + N(j) * M(i - j + 1);
            end
        end
    end
    output = Y;
end
%function for converting them to fixed point
function x_fixed = fixed_point(x, scale_factor)
    x_fixed = zeros(size(x));
    for i = 1:length(x)
        x_fixed(i) = fix(x(i) * scale_factor);
    end
end
%function for converting into float
function x_float = fix_float(x, scale_factor)
    x_float = x / scale_factor^2;
end
```

## RESULT:

```
The correlation of numbers x1 and x2:

  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

The correlation of fixed numbers is:

  Columns 1 through 10
```

```
   0.4826    5.5519    9.1893    1.7090    1.4334    7.6994    2.8707   -1.7700    2.0278    0.5312

  Column 11

   -0.0450

Error:

   1.0e-05 *

  Columns 1 through 10

   0.0015    0.0200    0.1118    0.0046    0.0309    0.1173    0.0160    0.0925    0.0127    0.0022

  Column 11

   0.0007

The MSE value of correlation:

   3.7289e-07
```

## Plot for Absolute error in correlation:




## C code for correlation:

```c
#include <stdio.h>
#define MAX_SIZE 100
 //function for calculaing correlation
void corr_fixed(int unscale_x1[], int unscale_x2[], int m, int n, int output[]) {
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
```

```
            output[i + j] += unscale_x1[i] * unscale_x2[n - j - 1];
        }
    }
}

void corr_double(double unscale_x1[], double unscale_x2[], int m, int n, double output_float[])
{
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            output_float[i + j] += unscale_x1[i] * unscale_x2[n - j - 1];
        }
    }
}
//function for converting into fixed points
void fixed_point(double x[], int fix[], int scale_factor, int size) {
    for (int i = 0; i < size; i++) {
        fix[i] = (int)(x[i] * scale_factor);
    }
}
//function for converting into float
void unscale(int fixed[], double conv_float[], int scale_factor, int size) {
    for (int i = 0; i < size; i++) {
        conv_float[i] = (double)(fixed[i]) / ((scale_factor) * (scale_factor));
    }
}
//function for calculating MSE value
float MSE(double output[], double output_fixed[], int length) {
    float mse = 0.0;
    for (int i = 0; i < length; i++) {
        double err = output[i] - output_fixed[i];
        mse += err * err;
    }
    mse /= length;
    return mse;
}

int main() {
    int m;
    int n;
    double x1[] = {0.3426, 3.5784, 2.7694, -1.3499, 3.0349, 0.7254, -0.0631};
    double x2[] = {0.7147, -0.2050, -0.1241, 1.4897, 1.4090};
    m = sizeof(x1) / sizeof(x1[0]);
    n = sizeof(x2) / sizeof(x2[0]);
    int Q = 12;
    int factor = 1 << Q;
    int output_fixed[MAX_SIZE + MAX_SIZE - 1] = {0};
    double output_float[MAX_SIZE + MAX_SIZE - 1] = {0};
    double unscaled_output[MAX_SIZE + MAX_SIZE - 1] = {0};
    int fixed_x1[MAX_SIZE];
    int fixed_x2[MAX_SIZE];

    fixed_point(x1, fixed_x1, factor, m);
    fixed_point(x2, fixed_x2, factor, n);
    corr_double(x1, x2, m, n, output_float);//giving us the float value of correlation
    corr_fixed(fixed_x1,fixed_x2,m,n,output_fixed);//it is giving fixed value of correlation
    unscale(output_fixed, unscaled_output, factor, m + n - 1);//converting the fixed value corr
to float value
//printing the corr of fixed numbers after unscaling
    printf("The correlated_fixed output is:\n");
    for (int i = 0; i < m + n - 1; i++) {
        printf("%.4f ", unscaled_output[i]);
    }
```

```
    printf("\n");
//printing the original x1,x2 correlation
    printf("The correlated_float output is:\n");
    for (int i = 0; i < m + n - 1; i++) {
        printf("%.4f ", output_float[i]);
    }
    printf("\n");
//calculating absolute error
    printf("Absolute Error:\n");
    for (int i = 0; i < m + n - 1; i++) {
        double abs_err = ((unscaled_output[i] - output_float[i]) * (unscaled_output[i] -
output_float[i]));
        printf("%E ", abs_err);
    }
    printf("\n");
//calling the mse function
    float mse = MSE(unscaled_output, output_float, m + n - 1);
    printf("Mean Squared Error: %E\n", mse);

    return 0;
}
```

## Result:

```
The correlated_fixed output is:
0.4826 5.5519 9.1893 1.7090 1.4334 7.6994 2.8707 -1.7700 2.0278 0.5312 -0.0450

The correlated_float output is:
0.4827 5.5523 9.1903 1.7093 1.4328 7.7005 2.8711 -1.7710 2.0282 0.5314 -0.0451

Absolute Error:
1.481035E-008 2.000605E-007 1.117691E-006 4.621002E-008 3.086498E-007 1.172762E-006 1.600544E-
007 9.249876E-007 1.273220E-007 2.183735E-008 7.425066E-009

Mean Squared Error: 3.728918E-007
```

## Observations:

- The error caused in the fixed-point correlation and convolution are very small in as we have calculated the absolute error where it is in the order of 10^(-8) and thus there is very slight error between the fixed number convolution/correlation and float number convolution/correlation.
- The use of fixed point is useful as the computation time and memory usage is lesser than the float point as it only deals with integer values.
- We found the MSE values of Convolution and correlation which are 4.0578e-07 and 3.7289e-07 respectively.