# Experiment-2

EE:2801 DSP-Lab

Indian Institute of Technology, Hyderabad

Jay Vikrant
EE22BTECH11025

## I. Question

Downsample and upsample the signal given below with a factor 2 and 3 in Matlab and C.

x[n]=0.5377 1.8339 -2.2588 0.8622 0.3188 -1.3077 -0.4336 0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631 0.7147 -0.2050 -0.1241 1.4897 1.4090 1.4172

## II. solution

Here is the C code,

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

#define N 20 // Assuming a maximum signal length of 20

double *upsampling(double x[], int f, int length) {
    // Pre-allocate the output array with zeros
    double *y1 = (double *)calloc(length * f, sizeof(double));

    // Upsample the signal using indexing
    for (int i = 0, j = 0; i < length * f; i += f, j++) {
        y1[i] = x[j];
    }

    return y1;
}

double *downsampling(double x[], int f, int length) {
    // Pre-allocate the output array
    double *y2 = (double *)calloc((length + f - 1) / f, sizeof(double));

    // Downsample the signal using indexing
    for (int i = 0, j = 0; i < length; i += f, j++) {
        y2[j] = x[i];
    }

    return y2;
```

```
}

int main() {
    // Example usage:
    double x[N] = {0.5377, 1.8339, −2.2588, 0.8622, 0.3188, −1.3077, −0.4336, 0.3426, 3.5784,
        2.7694, −1.3499, 3.0349, 0.7254, −0.0631, 0.7147, −0.2050, −0.1241, 1.4897, 1.4090, 1.4172
        }; // Fill in your signal values
    int f = 2;
    int length = sizeof(x) / sizeof(x[0]); // Determine actual signal length

    // Upsample and downsample the signal
    double *upsampled_sequence = upsampling(x, f, length);
    double *downsampled_sequence = downsampling(x, f, length);

    // Display the results
    printf("Upsampled sequence:\n");
    for (int i = 0; i < length * f; i++) {
        printf("%f ", upsampled_sequence[i]);
    }
    printf("\n");

    printf("Downsampled sequence:\n");
    for (int i = 0; i < (length + f - 1) / f; i++) {
        printf("%f ", downsampled_sequence[i]);
    }
    printf("\n");

    // Free allocated memory
    free(upsampled_sequence);
    free(downsampled_sequence);

    return 0;
}
```

The following got computed in C,
for factor = 2

```
Upsampled sequence:
0.537700 0.000000 1.833900 0.000000 -2.258800 0.000000 0.862200 0.000000 0.318800 0.000000 -1.307700 0.000000 -0.433600 0.000000 0.342600 0.000000 3.578400 0.000000 2.769400 0.000000 -1.349900 0.000000 3.034900 0.000000 0.725400 0.000000 -0.063100 0.000000 0.714700 0.000000 -0.205000 0.000000 -0.124100 0.000000 1.489700 0.000000 1.409000 0.000000 1.417200 0.000000
Downsampled sequence:
0.537700 -2.258800 0.318800 -0.433600 3.578400 -1.349900 0.725400 0.714700 -0.124100 1.409000
```

for factor = 3

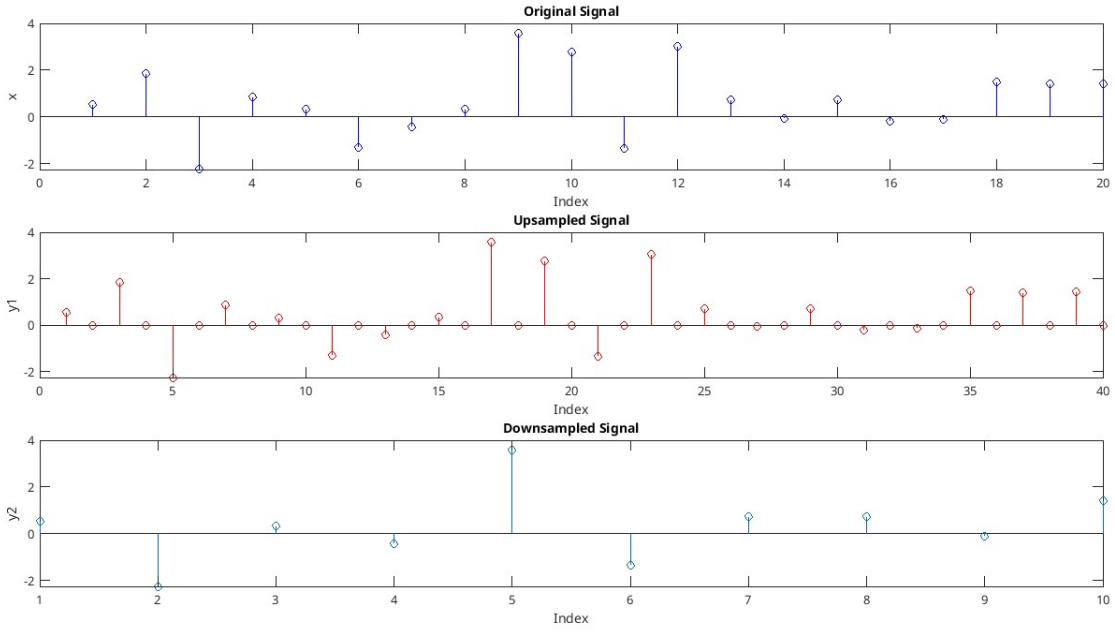
```
Upsampled sequence:
0.537700 0.000000 0.000000 1.833900 0.000000 0.000000 -2.258800 0.000000 0.000000 0.862200 0.000000 0.000000 0.318800 0.000000 0.000000 -1.307700 0.000000 0.000000 -0.433600 0.000000 0.000000 0.342600 0.000000 0.000000 3.578400 0.000000 0.000000 2.769400 0.000000 0.000000 -1.349900 0.000000 0.000000 3.034900 0.000000 0.000000 0.725400 0.000000 0.000000 -0.063100 0.000000 0.000000 0.714700 0.000000 0.000000 -0.205000 0.000000 0.000000 -0.124100 0.000000 0.000000 1.489700 0.000000 0.000000 1.409000 0.000000 0.000000 1.417200 0.000000 0.000000
Downsampled sequence:
0.537700 0.862200 -0.433600 2.769400 0.725400 -0.205000 1.409000
```

Now, for the Matlab simulation,

```
function main()
  % Example usage:
    x = [0.5377 1.8339 −2.2588 0.8622 0.3188 −1.3077 −0.4336 0.3426 3.5784 2.7694 −1.3499
        3.0349 0.7254 −0.0631 0.7147 −0.2050 −0.1241 1.4897 1.4090 1.4172];
    f = 3;

  % Upsample and downsample the signal
    upsampled_sequence = upsampling(x, f);
    downsampled_sequence = downsampling(x, f);
  % Display the results
    disp('upsample ');
    disp(upsampled_sequence);
    disp('downsample ');
    disp(downsampled_sequence);

  % Plot the signals
    subplot(3,1,1); % Arrange plots vertically in 3 rows, 1 column
    stem(1:length(x), x, 'b', 'Marker', 'o', 'DisplayName', 'Original Signal');
    title('Original Signal');
    xlabel('Index');
    ylabel('x');

    subplot(3,1,2);
    stem(1:length(upsampled_sequence), upsampled_sequence, 'r', 'Marker', 'o', 'DisplayName', '
        Upsampled Signal'); % Corrected marker to 'x'
    title('Upsampled Signal');
    xlabel('Index');
    ylabel('y1');

    subplot(3,1,3);
    stem(1:length(downsampled_sequence), downsampled_sequence, 'p', 'Marker', 'o', 'DisplayName
        ', 'Downsampled Signal'); % Corrected marker to 'x'
    title('Downsampled Signal');
    xlabel('Index');
    ylabel('y2');

end

function y1 = upsampling(x, f)
  % Pre−allocate the output array with zeros
    y1 = zeros(1, length(x) * f);

  % Upsample the signal using vectorized operations
    y1(1:f:end) = x;
end

function y2 = downsampling(x, f)
```

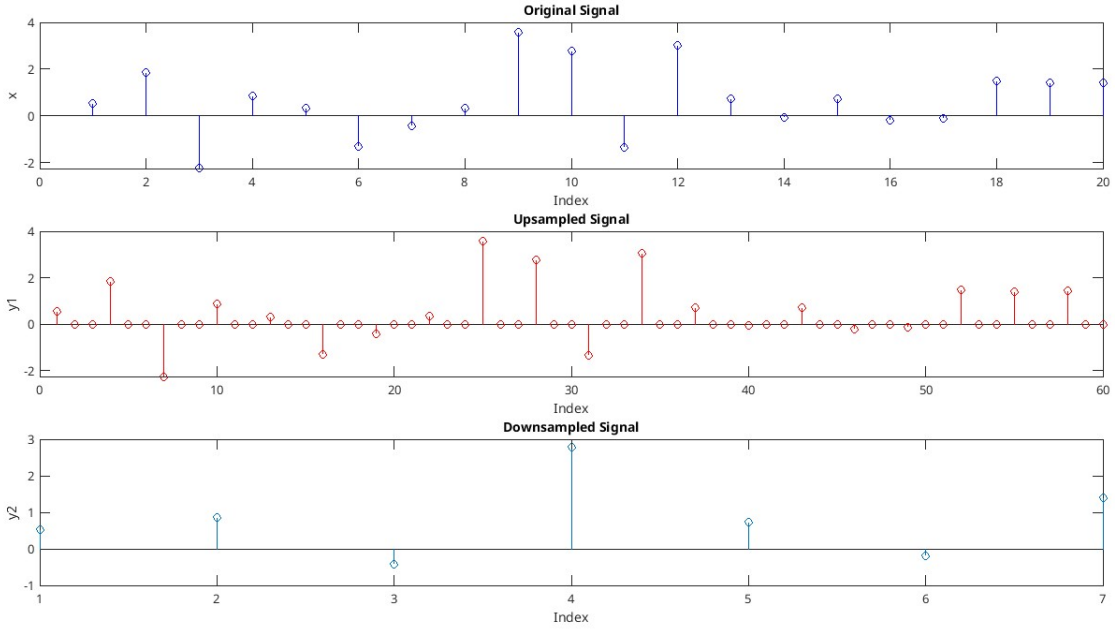
```
    % Pre-allocate the output array with zeros
    y2 = zeros(1, floor(length(x) / f));

    % Downsample the signal using vectorized operations
    y2 = x(1:f:end);
end
```

The following got computed in Matlab,
for factor = 2




```
>> main1
upsample
  Columns 1 through 20

    0.5377         0    1.8339         0   -2.2588         0    0.8622         0    0.3188         0   -1.3077         0   -0.4336         0    0.3426         0    3.5784         0    2.7694         0

  Columns 21 through 40

   -1.3499         0    3.0349         0    0.7254         0   -0.0631         0    0.7147         0   -0.2050         0   -0.1241         0    1.4897         0    1.4090         0    1.4172         0

downsample
    0.5377   -2.2588    0.3188   -0.4336    3.5784   -1.3499    0.7254    0.7147   -0.1241    1.4090
```

for factor = 3




```
>> main1
upsample
  Columns 1 through 22

    0.5377         0         0    1.8339         0         0   -2.2588         0         0    0.8622         0         0    0.3188         0         0   -1.3077         0         0   -0.4336         0         0    0.3426

  Columns 23 through 44

         0         0    3.5784         0         0    2.7694         0         0   -1.3499         0         0    3.0349         0         0    0.7254         0         0   -0.0631         0         0    0.7147         0

  Columns 45 through 60

         0   -0.2050         0         0   -0.1241         0         0    1.4897         0         0    1.4090         0         0    1.4172         0         0

downsample
    0.5377    0.8622   -0.4336    2.7694    0.7254   -0.2050    1.4090
```