

# DSP LAB - LAB 2

Ajay Krishnan K  
EE22BTECH11003

January 31, 2024

## 1 Upsampling

### 1.1 MATLAB

#### Code

```
1      % Upsampling using zero insertion
2
3      upsamplingFactor2 = 2;
4      upsamplingFactor3 = 3;
5
6      x = [0.5377,1.8339,-2.2588,0.8622,0.3188,-1.3077,-0.4336,
7           0.3426,3.5784,2.7694,-1.3499,3.0349,0.7254,-0.0631,
8           0.7147,-0.2050,-0.1241,1.4897,1.4090,1.4172];
9
10     upsampledBy2 = upSample(x, upsamplingFactor2);
11     upsampledBy3 = upSample(x, upsamplingFactor3);
12
13     disp('Original Signal');
14     disp(x);
15     disp('Upsampled Signal by 2');
16     disp(upsampledBy2);
17     disp('Upsampled Signal by 3');
18     disp(upsampledBy3);
19
20     % Plot the original and upsampled signals
21     subplot(3,1,1);
22     stem(x, 'b', 'DisplayName', 'Original Signal');
23     title('Original Signal');
24     xlabel('Sample Index');
25     ylabel('Amplitude');
26     legend('Original Signal');
27
28     subplot(3,1,2);
```

```

29     stem(upsampledBy2, 'r', 'DisplayName', 'Upsampled Signal by 2
30         ');
31     title('Upsampled Signal by 2');
32     xlabel('Sample Index');
33     ylabel('Amplitude');
34     legend('Upsampled Signal by 2');
35
36     subplot(3,1,3);
37     stem(upsampledBy3, 'r', 'DisplayName', 'Upsampled Signal by 3
38         ');
39     title('Upsampled Signal by 3');
40     xlabel('Sample Index');
41     ylabel('Amplitude');
42     legend('Upsampled Signal by 3');
43
44     function y = upSample(x, n)
45     N = length(x);
46     y = zeros(1, N * n);
47     y(1:n:end) = x;
48     end

```

## Output

```

1     Original Signal
2     Columns 1 through 17
3
4     0.5377    1.8339   -2.2588    0.8622    0.3188   -1.3077
5     -0.4336    0.3426    3.5784    2.7694   -1.3499
6     3.0349    0.7254   -0.0631    0.7147   -0.2050
7     -0.1241
8
9     Columns 18 through 20
10
11     1.4897    1.4090    1.4172
12
13     Upsampled Signal by 2
14     Columns 1 through 17
15
16     0.5377         0    1.8339         0   -2.2588         0
17     0.8622         0    0.3188         0         0   -1.3077
18     0   -0.4336         0    0.3426         0
19     3.5784
20
21     Columns 18 through 34
22
23     0    2.7694         0   -1.3499         0    3.0349
24         0    0.7254         0   -0.0631         0
25     0.7147         0   -0.2050         0
26     -0.1241         0

```

18	
19	Columns 35 through 40
20	
21	1.4897                      0            1.4090                      0            1.4172                      0
22	
23	Upsampled Signal by 3
24	Columns 1 through 17
25	
26	0.5377                      0                      0            1.8339                      0                      0
	-2.2588                      0                      0            0            0.8622                      0
	0            0.3188                      0                      0            -1.3077
	0
27	
28	Columns 18 through 34
29	
30	0            -0.4336                      0                      0            0.3426                      0
	0                      3.5784                      0                      0            2.7694
	0                      0            -1.3499                      0
	0            3.0349
31	
32	Columns 35 through 51
33	
34	0                      0            0.7254                      0                      0            -0.0631
	0                      0            0.7147                      0                      0
	-0.2050                      0                      0            -0.1241
	0                      0
35	
36	Columns 52 through 60
37	
38	1.4897                      0                      0            1.4090                      0                      0
	1.4172                      0                      0

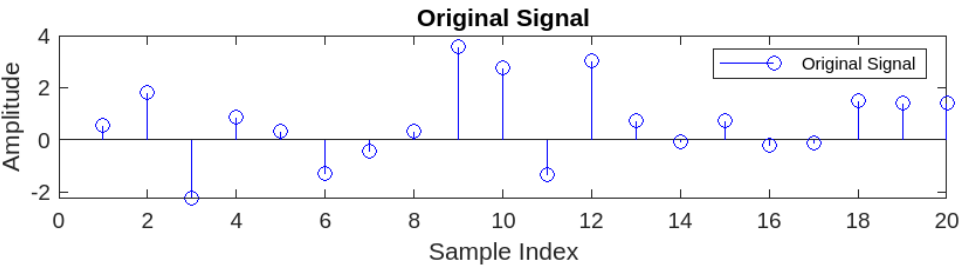


Figure 1: Upsampling Sample Signal

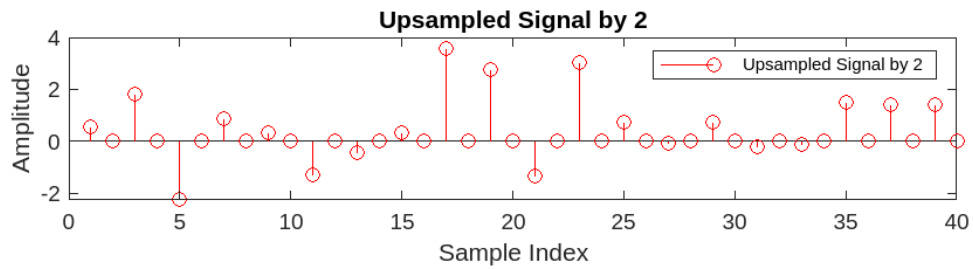


Figure 2: Upsampling by Factor of 2

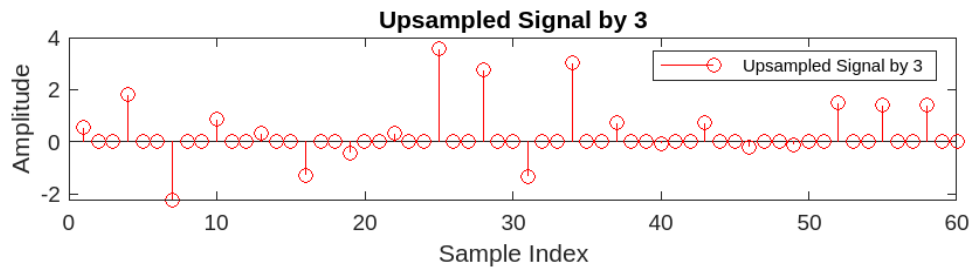


Figure 3: Upsampling by Factor of 3

## 1.2 C

### Code

```

1  #include <stdio.h>
2
3  void upsampleSignal(float input[], int inputSize, float
4  output[], int n)
5  {
6      int outputSize = inputSize * n;
7      int index = 0;
8
9      for (int i = 0; i < inputSize; i++)
10     {
11         output[index++] = input[i];
12         for (int j = 1; j < n; j++)
13         {
14             output[index++] = 0.0;
15         }
16     }
17 }
18
19 int main()
20 {
21     int inputSize = 20;
22     float input[20] = {0.5377, 1.8339, -2.2588, 0.8622,
23                        0.3188, -1.3077, -0.4336, 0.3426, 3.5784, 2.7694,
24                        -1.3499, 3.0349, 0.7254, -0.0631, 0.7147, -0.2050,
25                        -0.1241, 1.4897, 1.4090, 1.4172};
26     int upsampleSignal2 = 2;

```

```

23     int upsampleSignal3 = 3;
24     int outputSize2 = inputSize * upsampleSignal2;
25     int outputSize3 = inputSize * upsampleSignal3;
26     float output2[outputSize2];
27     float output3[outputSize3];
28
29     upsampleSignal(input, inputSize, output2, upsampleSignal2
30                   );
31     upsampleSignal(input, inputSize, output3, upsampleSignal3
32                   );
33
34     printf("Input Signal: ");
35     for (int i = 0; i < inputSize; i++)
36     {
37         printf("%f ", input[i]);
38     }
39
40     printf("\nUpsampled Signal by 2: ");
41     for (int i = 0; i < outputSize2; i++)
42     {
43         printf("%f ", output2[i]);
44     }
45
46     printf("\nUpsampled Signal by 3: ");
47     for (int i = 0; i < outputSize3; i++)
48     {
49         printf("%f ", output3[i]);
50     }
51
52     return 0;
53 }

```

## Output

```

1     Input Signal: 0.537700 1.833900 -2.258800 0.862200 0.318800
2     -1.307700 -0.433600 0.342600 3.578400 2.769400 -1.349900
3     3.034900 0.725400 -0.063100 0.714700 -0.205000 -0.124100
4     1.489700 1.409000 1.417200
5
6     Upsampled Signal by 2: 0.537700 0.000000 1.833900 0.000000
7     -2.258800 0.000000 0.862200 0.000000 0.318800 0.000000
8     -1.307700 0.000000 -0.433600 0.000000 0.342600 0.000000
9     3.578400 0.000000 2.769400 0.000000 -1.349900 0.000000
10    3.034900 0.000000 0.725400 0.000000 -0.063100 0.000000
11    0.714700 0.000000 -0.205000 0.000000 -0.124100 0.000000
12    1.489700 0.000000 1.409000 0.000000 1.417200 0.000000
13
14    Upsampled Signal by 3: 0.537700 0.000000 0.000000 1.833900
15    0.000000 0.000000 -2.258800 0.000000 0.000000 0.862200
16    0.000000 0.000000 0.318800 0.000000 0.000000 -1.307700
17    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

```

## 2 Downsampling

### 2.1 MATLAB

#### Code

```

1      % Downsampling
2
3      downsamplingFactor2 = 2;
4      downsamplingFactor3 = 3;
5
6      x = [0.5377,1.8339,-2.2588,0.8622,0.3188,-1.3077,-0.4336,
7           0.3426,3.5784,2.7694,-1.3499,3.0349,0.7254,-0.0631,
8           0.7147,-0.2050,-0.1241,1.4897,1.4090,1.4172];
9
10     downsampled_2 = downSample(x, downsamplingFactor2);
11     downsampled_3 = downSample(x, downsamplingFactor3);
12
13     % Plot the original and downsampled signals
14     subplot(3,1,1);
15     stem(x, 'b', 'DisplayName', 'Original Signal');
16     title('Original Signal');
17     xlabel('Sample Index');
18     ylabel('Amplitude');
19     legend('Original Signal');
20
21     subplot(3,1,2);
22     stem(downsamped_2, 'r', 'DisplayName', 'Downsampled by 2');
23     title('Downsampled Signal');
24     xlabel('Sample Index');
25     ylabel('Amplitude');
26     legend('Downsampled by 2 Signal');
27
28     subplot(3,1,3);
29     stem(downsamped_3, 'r', 'DisplayName', 'Downsampled by 3');
30     title('Downsampled Signal');
31     xlabel('Sample Index');
32     ylabel('Amplitude');
33     legend('Downsampled by 3 Signal');

```

```

34
35 function y = downSample(x, n)
36 y = x(1:n:end);
37 end

```

## Output

```

1 Original Signal
2 Columns 1 through 17
3
4      0.5377      1.8339     -2.2588      0.8622      0.3188     -1.3077
5      -0.4336      0.3426      3.5784      2.7694     -1.3499
6      3.0349      0.7254     -0.0631      0.7147     -0.2050
7      -0.1241
8
9 Columns 18 through 20
10
11      1.4897      1.4090      1.4172
12
13 Downsampled by 2 Signal
14      0.5377     -2.2588      0.3188     -0.4336      3.5784     -1.3499
15      0.7254      0.7147     -0.1241      1.4090
16
17 Downsampled by 3 Signal
18      0.5377      0.8622     -0.4336      2.7694      0.7254     -0.2050
19      1.4090

```

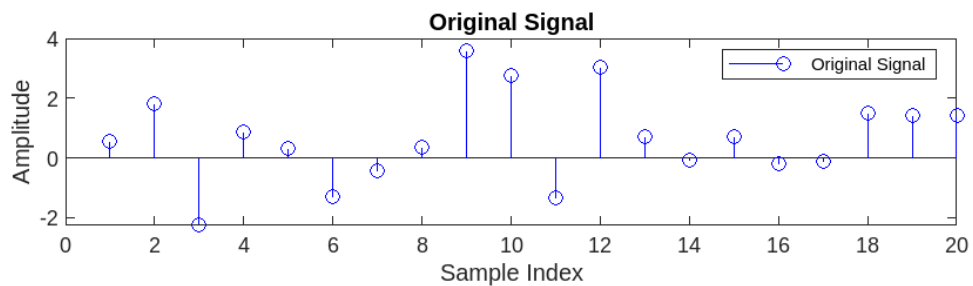


Figure 4: Downsampling Sample Signal

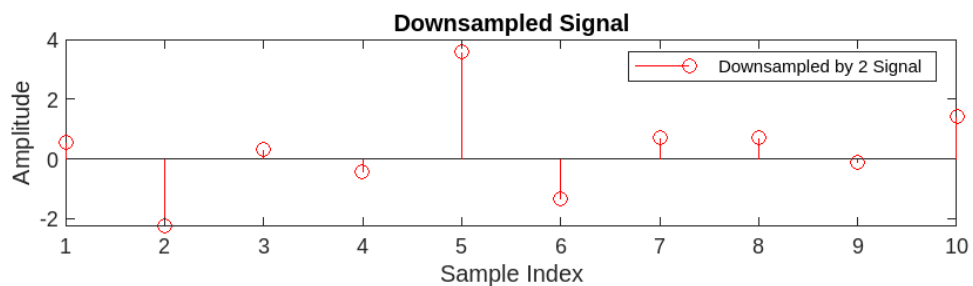


Figure 5: Downsampling Signal by Factor of 2

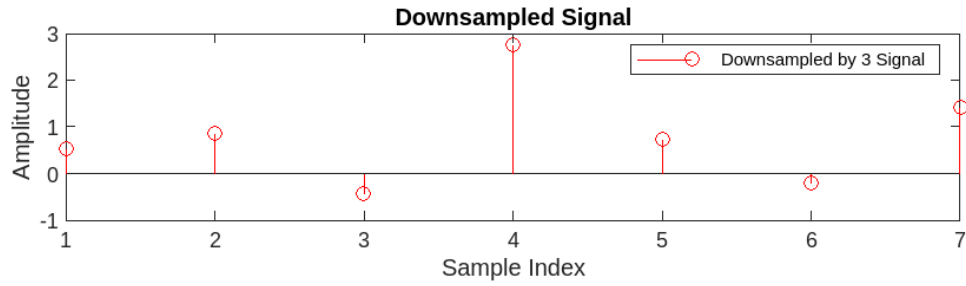


Figure 6: Downsampling Signal by Factor of 3

## 2.2 C

### Code

```

1  #include <stdio.h>
2
3  void downsampleSignal(float input[], int inputSize, float
4      output[], int n)
5  {
6      int outputSize = inputSize / n;
7      int index = 0;
8
9      for (int i = 0; i < inputSize; i += n)
10     {
11         output[index++] = input[i];
12     }
13 }
14
15 int main()
16 {
17     int inputSize = 20;
18     float input[20] = {0.5377, 1.8339, -2.2588, 0.8622,
19         0.3188, -1.3077, -0.4336, 0.3426, 3.5784, 2.7694,
20         -1.3499, 3.0349, 0.7254, -0.0631, 0.7147, -0.2050,
21         -0.1241, 1.4897, 1.4090, 1.4172};
22     int downsampleFactor2 = 2;
23     int downsampleFactor3 = 3;
24     int outputSize2 = inputSize / downsampleFactor2;
25     int outputSize3 = inputSize / downsampleFactor3;
26     float output2[outputSize2];
27     float output3[outputSize3];
28
29     downsampleSignal(input, inputSize, output2,
30         downsampleFactor2);
31     downsampleSignal(input, inputSize, output3,
32         downsampleFactor3);
33
34     printf("Input Signal: ");
35     for (int i = 0; i < inputSize; i++)
36     {

```



```

31         printf("%f ", input[i]);
32     }
33
34     printf("\nDownsampled Signal by 2: ");
35     for (int i = 0; i < outputSize2; i++)
36     {
37         printf("%f ", output2[i]);
38     }
39
40     printf("\nDownsampled Signal by 3: ");
41     for (int i = 0; i < outputSize3; i++)
42     {
43         printf("%f ", output3[i]);
44     }
45
46     return 0;
47 }

```

## Output

```

1      Input Signal: 0.537700 1.833900 -2.258800 0.862200 0.318800
      -1.307700 -0.433600 0.342600 3.578400 2.769400 -1.349900
      3.034900 0.725400 -0.063100 0.714700 -0.205000 -0.124100
      1.489700 1.409000 1.417200
2      Downsampled Signal by 2: 0.537700 -2.258800 0.318800
      -0.433600 3.578400 -1.349900 0.725400 0.714700 -0.124100
      1.409000
3      Downsampled Signal by 3: 0.537700 0.862200 -0.433600 2.769400
      0.725400 -0.205000

```