DSP LAB - LAB 1

Ajay Krishnan K - EE22BTECH11003

January 22, 2024

1 Convolution

1.1 MATLAB

1.1.1 Code

```
% CONVOLUTION

clc
clear
close all

a = input('Enter␣the␣signal:␣');
b = input('Enter␣the␣impulse␣response:␣');

% a = [1 2 3];
% b = [4 5 6];
conv_val = conv_taker(a,b);
disp(conv_val)

function [conv] = conv_taker(signal, impulse)
L = length(signal) + length(impulse) - 1;
conv = zeros(1, L);

for n = 1:L
    for k = 1:length(impulse)
        m = n - k + 1;
        if 1 <= m && m <= length(signal)
            conv(n) = conv(n) + signal(m) *
            impulse(k);
        end
    end
end
end
```

1.1.2 Output

```
Enter the signal: [0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631]
Enter the impulse response: [0.7147 -0.2050 -0.1241 1.4897 1.4090]
    0.2449    2.4872    1.2032   -1.4662    7.9156
        9.2314    1.3207    2.5420    5.3646    0.9281   -0.0889
```

1.2 C

1.2.1 Code

```
#include <stdio.h>

void convolve(int signal[], int impulse[],int signal_length,int impulse_length,
  int result[]) {
int i, j, k;

for (i = 0; i < signal_length + impulse_length - 1;
 i++) {
    result[i] = 0;
    for (j = 0; j <= i && j < signal_length; j++) {
        k = i - j;
        if (k < impulse_length) {
            result[i] += signal[j] * impulse[k];
            }
        }
    }
}

int main() {
int signal_length, impulse_length, i;
printf("Enter the size of the first array: ");
scanf("%d", &signal_length);
int signal[signal_length];
printf("Enter the elements of the first array: ");
for (i = 0; i < signal_length; i++) {
    scanf("%d", &signal[i]);
}

printf("Enter the size of the second array: ");
scanf("%d", &impulse_length);
int impulse[impulse_length];
printf("Enter the elements of the second array: ");
for (i = 0; i < impulse_length; i++) {
    scanf("%d", &impulse[i]);
}

int result[signal_length + impulse_length - 1];
convolve(signal, impulse, signal_length, impulse_length,result);

printf("The convolution of the two arrays is: ");
for (i = 0; i < signal_length + impulse_length - 1;i++) {
    printf("%d ", result[i]);
}
printf("\n");

return 0;
}
```

1.2.2 Output

```
Enter the size of the first array: 7
Enter the elements of the first array: 0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631
```

```
    Enter the size of the second array: 5
    Enter the elements of the second array: 0.7147 -0.2050 -0.1241 1.4897 1.4090
    The convolution of the two arrays is: 0.245 2.487 1.203
    -1.466 7.916 9.231 1.321 2.542 5.365 0.928 -0.089
```

2 Correlation

2.1 MATLAB

2.1.1 Code

```
    % CORRELATION

clc
clear
close all

% a = input('Enter the signal: ');
% b = input('Enter the impulse response: ');

a = [1 2 3];
b = [4 5 6];

correlation_val = correlation_taker(a,b);
disp(correlation_val)

function [correlation] = correlation_taker(signal, impulse)
L = length(signal) + length(impulse) - 1;
correlation = zeros(1, L);

for n = 1:L
    for k = 1:length(impulse)
        m = n - k + 1;
        if 1 <= m && m <= length(signal)
            correlation(n) = correlation(n) + signal(m) * impulse(length(impulse) - k + 1);
        end
    end
end
end
```

2.1.2 Output

```
    Enter the signal: [0.3426 3.5784 2.7694 -1.3499 3.0349 0.7254 -0.0631]
    Enter the seconfd signal: [0.7147 -0.2050 -0.1241 1.4897 1.4090]
        0.4827     5.5523     9.1903     1.7093     1.4328
            7.7005     2.8711    -1.7710     2.0282     0.5314    -0.0451
```

2.2 C

2.2.1 Code

```
    #include <stdio.h>

    void correlate(float signal[], float impulse[],
int signal_length, int impulse_length, float result[])
{
    int i, j, k;

    for (i = 0; i < signal_length + impulse_length − 1; i++)
    {
        result[i] = 0;
        for (j = 0; j <= i && j < signal_length; j++)
        {
            k = i − j;
            if (k < impulse_length)
            {
                result[i] += signal[j] ∗ impulse[impulse_length − 1 − k];
            }
        }
    }
}

int main()
{
    int signal_length, impulse_length, i;
    printf(”Enter the size of the first array: ”);
    scanf(”%d”, &signal_length);
    float signal[signal_length];
    printf(”Enter the elements of the first array: ”);
    for (i = 0; i < signal_length; i++)
    {
        scanf(”%f”, &signal[i]);
    }

    printf(”Enter the size of the second array: ”);
    scanf(”%d”, &impulse_length);
    float impulse[impulse_length];
    printf(”Enter the elements of the second array: ”);
    for (i = 0; i < impulse_length; i++)
    {
        scanf(”%f”, &impulse[i]);
    }

    float result[signal_length + impulse_length − 1];
    correlate(signal, impulse, signal_length, impulse_length, result);

    printf(”The correlation of the two arrays is: ”);
    for (i = 0; i < signal_length + impulse_length − 1; i++)
    {
        printf(”%0.3f ”, result[i]);
    }
    printf(”\n”);

    return 0;
}
```

2.2.2 Output

```
Enter the size of the first array: 7
Enter the elements of the first array: 0.3426 3.5784 2.7694 −1.3499 3.0349
```

```
0.7254 -0.0631
Enter the size of the second array: 5
Enter the elements of the second array: 0.7147 -0.2050 -0.1241 1.4897 1.4090
The correlation of the two arrays is: 0.483 5.552 9.190 1.709 1.433 7.700
 2.871 -1.771 2.028 0.531 -0.045
```