

# Communication and Signal Processing Lab

## Assignment No.-1

EE21MTECH14002

### 1. Convolution:

Convolution is defined as a process which relates the three signals i.e. input signal, output signal and impulse response of a system.

If we have a linear system with impulse response  $h[n]$ , and input  $x[n]$ . Then we can define its output  $y[n]$  using convolution.

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

### Convolution Function Using C-code:

```
void convolution(float x1[], int lenx1, float h1[], int lenh1, float y1[], int leny1)
{
    for(int n=0; n<leny1;n++)
    {
        y1[n]=0;
        for(int k=0; k<=n;k++)
        {
            if((k<=(lenx1-1)) && ((n-k)<=(lenh1-1)))
            {
                y1[n]+=x1[k]*h1[n-k];
            }
        }
    }
}
```

Above code depict the method to convolve two sequence of some length using two for loops in c-programming.

**x1[]** is an array contains our input sequence, **h1[]** is our impulse response array, **y1[]** contains output and **lenx1, lenh1, leny1** are length variable to hold lengths of our three sequences.

## 2. Correlation:-

Correlation is the measure of similarity between two signal as a function of displacement of one signal with respect to the other signal given. It is mathematically equal to convolution of one signal with the time inversed version of other signal.

$$R_{xy}[k] = \sum_{n=0}^{\infty} x(n) y(n - k)$$

### Cross-Correlation Function Using C-code:

```
void correlation(int x2[], int lenx2, int y2[], int leny2, int R_xy[], int lenR_xy)
{
    int p=leny2-1;
    int h2[leny2];

    for(int i=0; i<leny2; i++) // Loop to time reverse the sequence 2
    {
        h2[i]= y2[p];
        p=p-1;
    }

    for(int n=0; n<lenR_xy; n++) //Loop to convolve one signal with time reversed of other signal
    {
        R_xy[n]=0;
        for(int k=0; k<=n; k++)
        {
            if((k<=(lenx2-1)) && ((n-k)<=(leny2-1)))
            {
                R_xy[n]+=x2[k]*h2[n-k];
            }
        }
    }
}
```

Above code depict the method to convolve two sequence of some length using two for loops in c-programming.

**X2[]** is an array contains our input sequence, **y2[]** is our second sequence/signal, **h2[]** is a time reversed version of **y2[]** and **R\_xy[]** is our output. **lenx2**, **leny2** and **lenR\_xy** are length variable to hold lengths of our three sequences.

### 3. Down sampling:-

It is also known as **decimation**. It is a compression or reducing the number of samples in a signal .It is done with the help of downscaling variable/factor.

$$y[n]=x[D*n]$$

*'D' is a downsampling facto*

### Downsampling Function Using C-code:

```
void downsampling(float x3[], int lenx3, float y3[], int leny3, float y4[], int leny4){  
    for(int i=0; i<leny3;i++)  
    {   int D=2;  
        y3[i]=x3[D*i];  
    }  
  
    for(int i=0; i<leny4;i++)  
    {   int D=3;  
        y4[i]=x3[D*i];  
    }  
}
```

Above code depict the method to downsample a given sequence by factor of 2 and 3 using for loops in c-programming.

**X3[]** is an array contains our input sequence that we have to compress, **y3[]** is an array to hold our output that is compressed by factor of 2, **y4[]** is an array to hold our output that is compressed by factor of 3 and **lenx3**, **leny4**, **leny5** are length variable to hold lengths of our three sequences.

## 4. Up sampling:-

Also, known as **Interpolation**. It is totally opposite of downsampling. In place of compressing a signal it actually add more samples to it or we can say that stuff samples in it.

$$y[n]=x[n/U]$$

*'U' is Upsampling factor*

### Upsampling Function Using C-code:

```
void upsampling(float x4[], int lenx4, float y5[], int leny5, float y6[], int leny6){  
    for(int i=0; i<leny5;i++){  
        {  
            int U = 2;  
            if(i%2==0)  
            {  
                y5[i]=x4[i/U];  
            }  
            else  
            {  
                y5[i]=0;  
            }  
        }  
    }  
    for(int i=0; i<leny6;i++){  
        {  
            int U=3;  
            if(i%3==0)  
            {  
                y6[i]=x4[i/U];  
            }  
            else  
            {  
                y6[i]=0;  
            }  
        }  
    }  
}
```

**X4[]** is an array contains our input sequence that we have to upsample, **y5[]** is an array to hold our output that is upsampled by factor of 2, **y6[]** is an array to hold our output that is upsampled by factor of 3 and **lenx3,leny5,leny6** are length variable to hold lengths of our three sequences.

-----\*-----\*-----*END*-----\*-----