**Lab # 5: Interpolation & Decimation**

**Objective:**

The objective of this lab is to understand the concepts of interpolation and decimation in digital signal processing (DSP). Students will implement interpolation and decimation techniques using MATLAB and analyze their effects on signals.

**Description:**

**1. Interpolation**

Interpolation is the process of increasing the sampling rate of a signal. It involves inserting new samples between existing samples. The steps for interpolation are:

* **Upsampling**: Increase the sampling rate by a factor of *L* (insert L−1zeros between samples).
* **Low-pass filtering**: Remove imaging artifacts caused by upsampling.

The interpolation factor *L* determines how many new samples are added.

**Effects on Fourier Spectrum:**

 The spectrum is **compressed** in the frequency domain by a factor of L.

 The frequency axis gets **scaled** such that the spectrum now repeats L times in the range –π to π.

**Aliasing and Spectral Replication**

Since the frequency axis is compressed, the original spectrum now repeats periodically in the Fourier domain. This causes aliasing unless a low-pass filter (interpolation filter) is applied after upsampling to remove unwanted copies of the spectrum**.**

**2. Decimation**

Decimation is the process of reducing the sampling rate of a signal. It involves removing samples from the signal. The steps for decimation are:

* **Low-pass filtering**: Anti-aliasing filter to prevent aliasing.
* **Downsampling**: Reduce the sampling rate by a factor of *M* (keep every *M*-th sample).

The decimation factor *M* determines how many samples are removed.

**Effects on Fourier Spectrum:**

 The spectrum is **stretched** in the frequency domain by a factor of M.

 The frequency components at multiples of 2π/M **alias** into the baseband (i.e., the frequency range −π to π).

**Aliasing and Spectral Overlap**

Since the frequency axis is expanded by a factor of M, the replicated spectra can **overlap**. This overlapping causes **aliasing**, where different frequency components mix and become indistinguishable. To prevent aliasing before downsampling, a **low-pass filter** (anti-aliasing filter) is typically applied to remove frequency components above π/M.

**LAB TASKS**

**1.** Record an audio signal saying @ 8kHz: “*Welcome to Digital Signal Processing Course.*”

and plot the signal in time and frequency domain with all axes correctly labeled. You can play the audio by using the MATLAB command sound. Note: if you don’t have a working microphone, you can always use the recorded audio.

**2.** Now decimate the audio signal using **downsample**() and plot the resultant signal in both time and frequency domain. Listen and observe the change in the audio. Kindly note that the decimation in Time domain results in spread of the spectrum in Frequency domain by the same factor.

Keep on doing decimation and plot signals in time domains until aliasing starts to occur and you can feel the change in voice while playing it.

**3.** Upsample the audio signal repeatedly by a factor 2 and plot the resultant signal in both time and frequency domain using **upsample()** function. **Note the changes happening to the signal in time domain and corresponding spectrum whenever we add upsample by adding zeros in the time domain.** Listen and observe the change in the audioAW. Do follow the same steps and plot the input signal in time and frequency domain.

**4.** Resample the audio signal to a sampling rate of 3/5 by using the functions downsample() and upsample(). Repeat the task for sampling rate of 5/3.

Analyze the relation between original audio signal and its decimated and interpolated versions based on the magnitude spectrum and by listening the audio after each step.