Experiment-8

EE:2801 DSP-Lab

Indian Institute of Technology, Hyderabad

Jay Vikrant
EE22BTECH11025

I. Aim of the experiment

Take,

$$x[n] = x(t)_{|t=nT_s} \tag{1}$$

$$x(t) = sin(2 \times \pi \times 500 \times t) + 2 \times sin(2 \times \pi \times 700 \times t) + 1.5 \times sin(2 \times \pi \times 1000 \times t) \tag{2}$$

from t = 0 to 1 and do the following operation shown in the block diagram.
Also, find the mean of absolute error as $E[|\widetilde{x_d}(n) - x(n)|]$

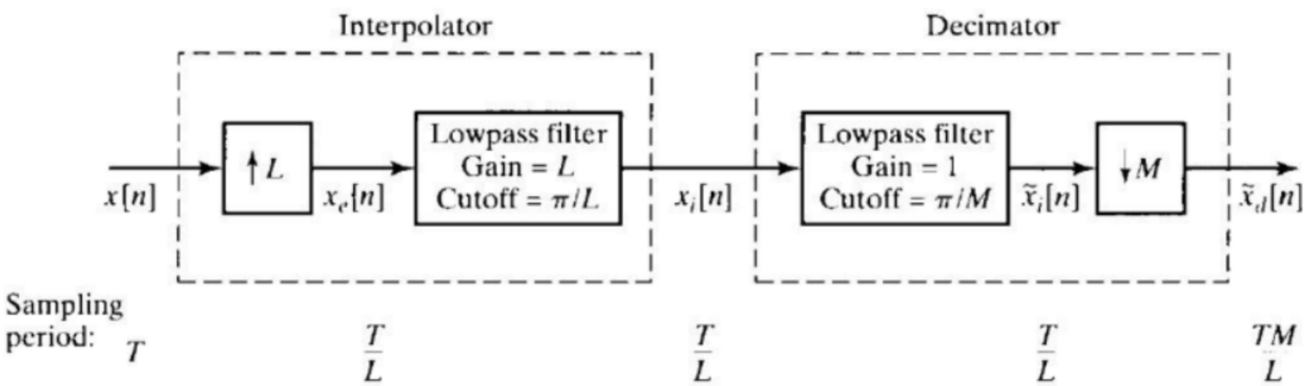


Fig. 0. Block Diagram

II. Theory

A. Interpolation

.

- Interpolation is a technique used to increase the sampling rate of a discrete-time signal. It involves inserting additional samples between existing samples of a signal to achieve a higher sampling rate.
- Interpolation is followed by filtering to remove unwanted frequency components that are introduced during the upsampling process.
- The main purpose of interpolation is to raise the sampling rate at the output of a system, enabling compatibility with another system that requires a higher sampling rate for input.(Example - Changing resolution of image.)
- Interpolation requires upscaling, allowing interpolation only by integer factors. Fractional factors are not feasible for interpolation alone. However, combining interpolation with decimation can achieve an overall rational factor for resampling.

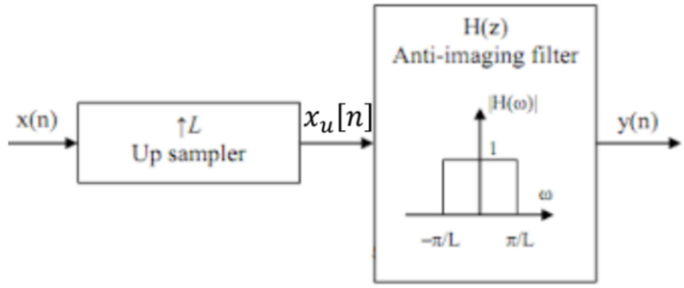


Fig. 0. Practical Interpolator

B. Decimation

- Decimation is a process used to reduce the sampling frequency of a discrete-time signal. It involves decreasing the number of samples in the signal to achieve a lower sampling rate.
- The main purpose of decimation is to conserve computational resources and reduce data storage requirements by lowering the sampling rate of the signal. (for example, data compression, and telecommunications by reducing the amount of data without significant loss of information.)
- **Filtering Before down-sampling**, a lowpass filter is applied to the signal to remove high-frequency components above the desired Nyquist frequency of the down-sampled signal.After filtering, the signal is down-sampled.The filter in decimation is crucial to prevent aliasing, ensuring that no frequency components above half the new sampling rate (Nyquist frequency) remain in the down-sampled signal.
- Decimation is complementary to interpolation, which is used to increase the sampling rate of a signal. Combining decimation with interpolation can achieve arbitrary resampling rates efficiently.

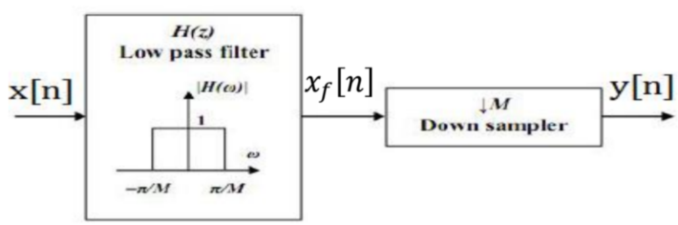


Fig. 0. Practical Decimator

III. How filter cutoff frequencies are decided in Interpolation and Decimation

A. Interpolation

- The pupose of Interpolation increases the sampling rate of a signal by inserting new samples between existing ones.
- When performing interpolation, a low-pass filter is typically used before upsampling to remove high-frequency components that could cause aliasing during the subsequent interpolation process.
- The cutoff frequency of the low-pass filter in interpolation is chosen based on the desired new sampling rate after interpolation. It should be set below half of the original sampling rate (Nyquist frequency) to avoid aliasing when new samples are inserted. Specifically, for interpolation by a factor of L (where L is an integer greater than 1), the cutoff frequency of the low-pass filter is often set at $f_c = \frac{f_s}{2L}$, where f_s is the original sampling frequency.

B. Decimation

- The objective of Decimation reduces the sampling rate of a signal by selecting a subset of existing samples.
- In decimation, a low-pass filter is applied after downsampling to remove high-frequency components introduced by the downsampling process.
- The cutoff frequency of the low-pass filter in decimation is crucial for preventing aliasing in the downsampled signal. The filter's cutoff frequency is typically set at or below $f_c = \frac{f_s}{2M}$, where M is the decimation factor (an integer greater than 1) and f_s is the original sampling frequency. This ensures that any high-frequency components introduced during downsampling (due to folding) are adequately filtered out.

In summary,the choice of filter cutoff frequency is guided by the Nyquist criterion to prevent aliasing. The specific values for these cutoff frequencies are determined by the desired resampling factors relative to the original sampling rate.

IV. MATLAB Simulation

A. Matlab Code

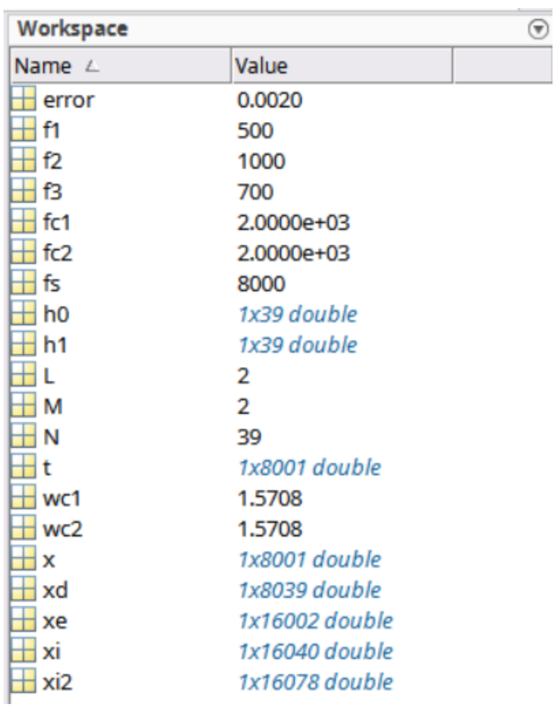
This is the main matlab code,

```
function main()
fs = 8000;
f1 = 500;
f2 = 1000;
f3 = 700;

t = 0:1/fs:1;
x = sin(2 * pi * f1 * t) + 2 * sin(2 * pi * f2 * t) + 1.5 * sin(2 * pi * f3 * t);

%(INTERPOLATION)
N = 39;
L = 2;

wc1 = pi/L;
fc1 = (wc1*fs)/(2*pi);

% Step-1 (upscaling)
xe = upsampler(x,L);

% Step-2 (Lowpass filtered)
h0 = lpf_ftr(fc1,fs,N,L);
xi = convl(xe,h0);


%(DECIMATION)
M = 2;
wc2 = pi/M;
fc2 = (wc2*fs)/(2*pi);

% Step-3 (Lowpass filtered)
h1 = lpf_ftr(fc2,fs,N,1);
```

```
xi2 = convl(xi,h1);

% Step-4 (downscaling)
xd = downsampler(xi2,M);

error = mean(abs(xd((N+1)/2:fs+(N+1)/2)-x));
disp(error)

end
function y1 = upsampler(x, f)
    y1 = zeros(1, length(x) * f);
    y1(1:f:end) = x;
end

function y2 = downsampler(x, f)
    y2 = zeros(1, floor(length(x) / f));
    y2 = x(1:f:end);
end

function y = lpf_ftr(fc,fs,N,G)

wc = (2*pi*fc)/(fs);
n = 1;
y = zeros(1,N-1);
for k = -(N-1)/2:(N-1)/2
    if k == 0
        y(n) =G*wc/pi;
    else
        y(n) = G*sin((wc*k))/(pi*k);
    end
    n = n + 1;
end

y = y.*hw(N);

end

function w = hw(N) %window function

n = 1;
w = zeros(1,N-1);
for k = 0 : (N-1)
    if k >= 0 && k <= (N-1)
        w(n) = 0.54 - 0.46*cos((2*pi*k)/(N-1));
    else
        w(n) = 0;
    end
    n = n + 1;
end
```

```
end

function y = convl(x,h) % y here is the output

  l = length(x) + length(h) − 1;

  y = zeros(1, l);

  for n = 1:l
    for k = 1:length(x)
      if (n − k + 1) >= 1 && (n − k + 1) <= length(h)
        y(n) = y(n) + x(k) ∗ h(n − k + 1);
      end
    end
  end
end
```

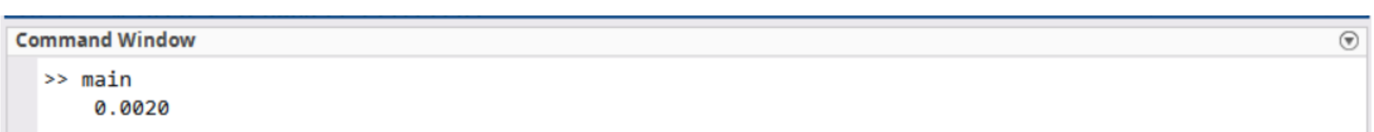
B. Outputs



Workspace

Name	Value
error	0.0020
f1	500
f2	1000
f3	700
fc1	2.0000e+03
fc2	2.0000e+03
fs	8000
h0	1x39 double
h1	1x39 double
L	2
M	2
N	39
t	1x8001 double
wc1	1.5708
wc2	1.5708
x	1x8001 double
xd	1x8039 double
xe	1x16002 double
xi	1x16040 double
xi2	1x16078 double

Fig. 0. No of samples x(n) is 8001, $x_e(n)$ = L × 8001, $x_i(n)$ L × 8001 , $\widetilde{x_i} \approx$ L × 8001, $\widetilde{x_d} \approx \frac{M}{L} \times$ 8001 samples, (L = M = 2)



Command Window

```
>> main
   0.0020
```

Fig. 0. Mean absolute error for Sampling frequency = 8000Hz

V. Observations

1) **Upsampling (Interpolation):**
   - After upsampling the signal by a factor of 2, the number of samples in the signal increased accordingly (from 8001 to 16002 samples in this case).
   - The interpolation process involved applying a low-pass filter with a gain of 2 and a cutoff frequency of $\frac{\pi}{2}$ to remove high-frequency components that could cause aliasing during upsampling.
2) **Downsampling (Decimation):**
   - Following interpolation, the signal was then downsampled by a factor of 2.
   - A low-pass filter with a gain of 1 and a cutoff frequency of $\frac{\pi}{2}$ was used after downsampling to eliminate any high-frequency components introduced by the downsampling process.
3) **Mean Absolute Error:**
   - The mean absolute error $E[|\widetilde{x_d}(n) - x(n)|]$ was calculated to quantify the deviation between the interpolated and original signals which came as 0.002 at fs = 8000Hz and N = 39. This make sense because we have take good enough value of sampling frequecny (i.e $> 2 \times f_{max}$).
   - This error metric helps assess the accuracy of the resampling process and the effectiveness of the filtering techniques employed.

VI. Conclusion

- **Interpolation** effectively increased the sampling rate of the input signal by inserting new samples between existing ones. The use of a low-pass filter with an appropriate cutoff frequency helped prevent aliasing during upsampling.
- **Decimation** reduced the sampling rate of the signal while maintaining essential signal characteristics. The application of a low-pass filter post-downsampling ensured that unwanted frequency components were adequately filtered out.
- The overall resampling process, combining interpolation and decimation, allowed for the transformation of the input signal to a different sampling rate, demonstrating the utility of these techniques in applications of telecommunication, image and audio processing.
- The mean absolute error analysis provided insights into the fidelity of the resampled signal compared to the original, highlighting the importance of proper filtering in maintaining signal integrity during sampling rate conversion.

In conclusion, the experiment successfully showcased the principles of interpolation and decimation in signal processing, emphasizing the critical role of filtering in ensuring accurate resampling operations.

END OF REPORT