EE5900 Programming Assignment 2

Gautam Singh
CS21BTECH11018

1) The code for both implementations is given in Source Code 1. We have implemented the resampler by performing upsampling first (Method 1) and also by performing downsampling first (Method 2). The results are shown in Figure 1 and Figure 2 for two signals of different frequencies. In the first case, both methods give almost the same results, since the frequency of the signal is within the cutoff. However, in the second case, the 9 kHz signal is filtered and aliased when Method 2 is employed, since it lies beyond the cutoff of 6 kHz.

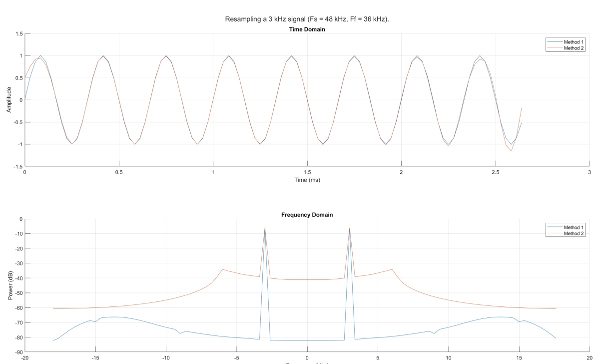

Fig. 1: Resampling a 3 kHz Signal Using Both Methods.

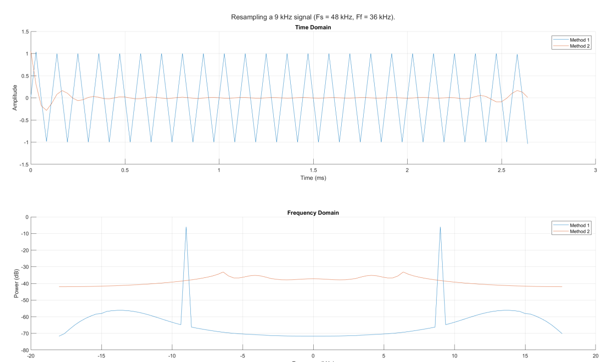

Fig. 2: Resampling a 9 kHz Signal Using Both Methods.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name        : Gautam Singh                               %
% Roll Number : CS21BTECH11018                             %
% Date        : 2023-11-11                                 %
% File        : ee5900_assign_2.m                          %
% Purpose     : Resample signals initially sampled at Fs  %
%               to Ff. Here, Fs = 48 kHz and Ff = 44 kHz. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear
close all

% List of constants
F = 3e3;    % Frequency of signal
N = 128;    % Number of samples
Fs = 48e3;  % Initial sampling frequency
Ff = 36e3;  % Final sampling frequency
L = 3;      % Upsampling factor
M = 4;      % Downsampling factor

% Sampling interval
Ts = 1/Fs;

% Timestamps
t = 0:Ts:(N-1)*Ts;

% Create samples of signal at rate Fs
x = sin(2*pi*F*t);

% Method 1: upsample, then downsample

% Upsampling
xu = upsample(x,L);

% Filtering (combined minimum cutoff)
yu = lowpass(xu,min(1/L,1/M));

% Downsampling
xud = L*downsample(yu,M);
```

```
% Method 2: downsample, then upsample

% Decimation
% Filtering
y = lowpass(x, 1/M);

% Downsampling
xd = downsample(y,M);

% Interpolation
% Upsampling
yd = upsample(xd,L);

% Filtering
xdu = L*lowpass(yd,1/L);

% Final timestamps (in ms)
tf = M*Ts/L*(0:1:length(xdu)-1)*1e3;

tlo = tiledlayout(2,1);
title(tlo, ['Resampling a ', num2str(F/1e3), ' kHz signal (Fs = ', ...
            num2str(Fs/1e3), ' kHz, Ff = ', num2str(Ff/1e3), ' kHz).']);
% Compare results (time domain)
nexttile
hold on
grid on
plot(tf, xud);
plot(tf, xdu);
legend('Method 1', 'Method 2');
xlabel('Time (ms)');
ylabel('Amplitude');
title('Time Domain')

% Compare results (frequency domain)
nexttile
hold on
grid on
yud = fftshift(fft(xud))/(L*N/M);
ydu = fftshift(fft(xdu))/(L*N/M);
n = length(xud);
f = (-n/2:n/2-1)*L*Fs/(M*1e3*n);
plot(f, 20*log10(abs(yud)), f, 20*log10(abs(ydu)));
legend('Method 1', 'Method 2');
xlabel('Frequency (kHz)');
ylabel('Magnitude (dB)');
title('Frequency Domain');
```

Source Code 1: MATLAB Code for Question 1.