Audio Sample-rate Converter Project

Ankita Sahoo (16EC10004), Sneha Vinjam (16EC10059)

***Abstract*—The project aims at realizing an audio-sample rate converter. This would require designing a fractional-rate (L/M) converter. First the sample rate conversion is performed by simply upsampling and downsampling to that particular ratio required. Then an ideal filter is used to pass the upsampled signal to observe a better audio signal obtained. Finally we have a polyphase implementation of filter with different passband ripples and stopband attenuation and the complexity of the filters is calculated to find the optimised one.**

I. Introduction

The project aims to build to L/M converter. The audio samples would be the audio sample that you will be recording yourself using the software "Free PC Audio Recorder". The software allows you to record at various sample-rates. It records in two-channel format (for stereo effect). While processing make sure that you are processing two-channels together. However, it compresses the audio signals to "mp3" files which results in a lossy compression. Firstly the audio sample recorded using the Audio Recorder software at 44.1kHz is preprocessed to '.wav' file and the characteristics of the audio is extracted in MATLAB. The audio sample up-sampled by a factor of 160, the upsampled signal is played. Then it is downsampled by 147 factor to make the sampling frequency 48kHz. The same process is repeated by passing the signal through an ideal filter. Then we design H1(z), H2(z), H3(z) with different specifications (passband ripple 0.1, 0.01, 0.001 dB, respectively and stopband attenuation of 30 dB, 55 dB, and 80 dB, respectively) to find the optimised filter design for the converter.

II. Procedure and Results

1. Audio sample up-sampled by a factor of 160 - The audio has a lot of hissing sounds and humming effects.

2. Upsampled audio decimated by a factory of 147 - The new audio now has a sampling rate of 48000. The stretching effect of the upsampled audio has reduced but not completely.

3. Upsampled audio passed through an ideal filter and then decimated by 147 factor - The audio obtained is almost similar to the original audio but the quality is reduced heavily, i.e. the audio has a lesser amplitude than the original signal.

4. Designing optimal systems using polyphase filters for realising H1(z), H2(z), H3(z).

The upsampling factor is broken into factorisation of 160 and the downsampling factor is broken into factorisation of 147 so as to convert the sampling rate of the audio sample of 44.1kHz to 48kHz. After every upsampling we pass the signal through a filter with cutoff at pi/L, and the passband and stopband frequency get scaled by fracction of L, where is the upsample rate also after every downsampling we pass the signal through a filter with bandwidth of pi/M and the passband and stopband frequency get multiplied by M, where M refers to the downsample rate.

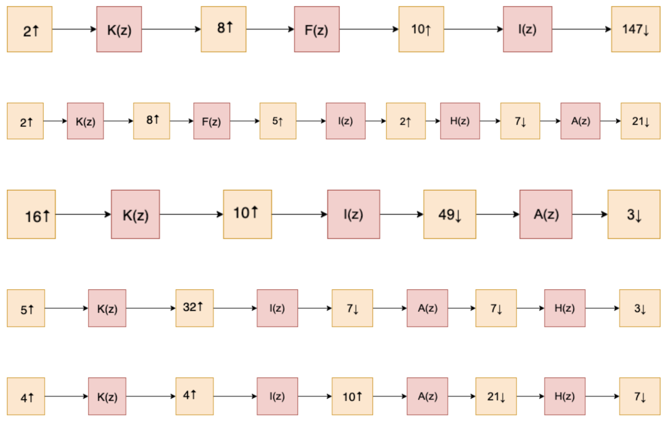
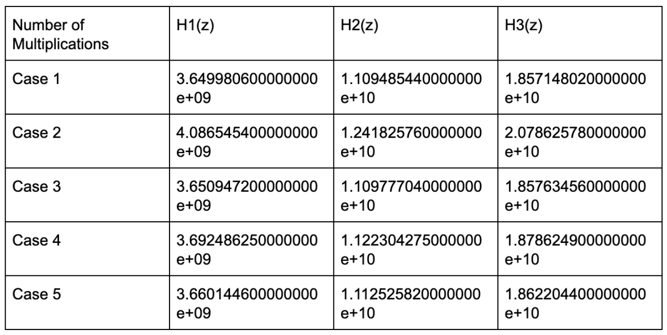


Fig. 1. Cases used for designing L/M fractional converter



Number of Multiplications	H1(z)	H2(z)	H3(z)
Case 1	3.649980600000000 e+09	1.109485440000000 e+10	1.857148020000000 e+10
Case 2	4.086545400000000 e+09	1.241825760000000 e+10	2.078625780000000 e+10
Case 3	3.650947200000000 e+09	1.109777040000000 e+10	1.857634560000000 e+10
Case 4	3.692486250000000 e+09	1.122304275000000 e+10	1.878624900000000 e+10
Case 5	3.660144600000000 e+09	1.112525820000000 e+10	1.862204400000000 e+10

Fig. 2. Number of multiplications in each case given in the above diagram

III. Discussions

1. Instead of direct upsampling and downsampling, passing the signals through filters retrives the original signal in a better way.
2. The case 1 configuration provides the most optimised implementation of the filter in H1(z), H2(z) and H3(z) followed the third one.
3. With increase in stopband attenuation and decrease in passband ripple the number of multiplications increase. This is due to increase in the order of the given filters.

IV. Appendix

1. **Step 1**
[y, Fs] = audioread('NewFile.wav');
y1 = upsample(y, 160);
*audiowrite('step1.wav', y1, Fs*160);*

2. **Step 2**
y2 = downsample(y1, 147);
$Fs_new = Fs * 160/147;$
$audiowrite('step2.wav', y1, Fs_new ew);$

*3. **Step 3***
[n,fo,ao,w] = firpmord([(44100/160)-(4100/160) (44100/160)],[1 0],[0.01, 0.1],44100);
b1 = firpm(n,fo,ao,w);
$y_filtered = filtfilt(b1, 1, y1);$
$audiowrite('step3.wav', y_filtered, Fs * 160);$

*4. **Step 4***
$y_final = downsample(y_filtered, 147);$
$audiowrite('step4.wav', y_final, 48000);$

Step 5 - Here only H1(z) code is written; all others can be obtained by simply changing the passband ripple and stopband attenuation.
*5. **Case 1***
rp = 10$^(-0.1/20);$
$rs = 10^(-30/20);$
$wp = 20000;$
$ws = 22000;$
$wp = wp/2;$
$ws = ws/2;$
$[n1, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$K = remez(n1, fo, ao, w);$
$wp = wp/8;$
$ws = ws/8;$
$[n2, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$F = remez(n2, fo, ao, w);$
$wp = wp/10;$
$ws = ws/10;$
$[n3, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$I = remez(n3, fo, ao, w);$
$mul = (n1 + 1) * 44100 * 2/2 + (n2 + 1) * 44100 * 2 * 8/2 + (n3 + 1) * 44100 * 2 * 8 * 10/2;$

*6. **Case 2***
rp = 10$^(-0.1/20);$
$rs = 10^(-30/20);$
$wp = 20000;$
$ws = 22000;$
$wp = wp/2;$
$ws = ws/2;$
$[n1, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$K = remez(n1, fo, ao, w);$
$wp = wp/8;$
$ws = ws/8;$
$[n2, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$F = remez(n2, fo, ao, w);$
$wp = wp/5;$
$ws = ws/5;$
$[n3, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$I = remez(n3, fo, ao, w);$
$wp = wp/2;$
$ws = ws/2;$
$[n4, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$H = remez(n4, fo, ao, w);$
$wp = wp * 7;$
$ws = ws * 7;$
$[n5, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$A = remez(n5, fo, ao, w);$
$mul = (n1 + 1) * 44100 * 2/2 + (n2 + 1) * 44100 * 2 * 8/2 + (n3 + 1) * 44100 * 2 * 8 * 2/2 + (n4 + 1) * 44100 * 2 * 8 * 2 * 5/2 + (n5 + 1) * 44100 * 2 * 8 * 2 * 5/(2 * 7);$

*7. **Case 3***
rp = 10$^(-0.1/20);$
$rs = 10^(-30/20);$
$wp = 20000;$
$ws = 22000;$
$wp = wp/16;$
$ws = ws/16;$
$[n1, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$K = remez(n1, fo, ao, w);$
$wp = wp/10;$
$ws = ws/10;$
$[n2, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$F = remez(n2, fo, ao, w);$
$wp = wp * 49;$
$ws = ws * 49;$
$[n3, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$I = remez(n3, fo, ao, w);$
$mul = (n1 + 1) * 44100 * 16/2 + (n2 + 1) * 44100 * 16 * 10/2 + (n3 + 1) * 44100 * 16 * 10/(2 * 49);$

*8. **Case 4***
rp = 10$^(-0.1/20);$
$rs = 10^(-30/20);$
$wp = 20000;$
$ws = 22000;$
$wp = wp/5;$
$ws = ws/5;$
$[n1, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$K = remez(n1, fo, ao, w);$
$wp = wp/32;$
$ws = ws/32;$
$[n2, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$F = remez(n2, fo, ao, w);$
$wp = wp * 7;$
$ws = ws * 7;$
$[n3, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$I = remez(n3, fo, ao, w);$

$wp = wp * 7;$
$ws = ws * 7;$
$[n4, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$H = remez(n4, fo, ao, w);$
$mul = (n1 + 1) * 44100 * 5/2 + (n2 + 1) * 44100 * 5 * 32/2 + (n3+1)*44100*5*32/(2*7)+(n4+1)*44100*5*32/(2*7*7);$

9. ***Case 5***
rp = 10$^(- 0.1/20);$
$rs = 10^(- 30/20);$
$wp = 20000;$
$ws = 22000;$
$wp = wp/4;$
$ws = ws/4;$
$[n1, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$K = remez(n1, fo, ao, w);$
$wp = wp/4;$
$ws = ws/4;$
$[n2, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$F = remez(n2, fo, ao, w);$
$wp = wp/10;$
$ws = ws/10;$
$[n3, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$I = remez(n3, fo, ao, w);$
$wp = wp * 21;$
$ws = ws * 21;$
$[n4, fo, ao, w] = remezord([wpws], [10], [rprs], 44100);$
$H = remez(n4, fo, ao, w);$
$mul = (n1 + 1) * 44100 * 4/2 + (n2 + 1) * 44100 * 4 * 4/2 + (n3+1)*44100*4*4*10/2+(n4+1)*44100*4*4*10/(2*21);$

10. ***Audio files required*** *-*
https://drive.google.com/open?id=141tCVZvnc26Q8GyfwfcvfGxIiFaNEVuJ