Lab 10

**Sound Echo Effect Using FIR and IIR Comb Filters**

**OBJECTIVES**

• Using FIR filter to build comb filter to generate sound echo effect.

• Using IIR filter to build comb filter to generate sound echo effect.

**INTRODUCTION**

Echo effect in sound is generated by playing the sound with a delayed attenuated version of the sound. This can be accomplished using comb filter. The filter can be of FIR or IIR type. For FIR comb filter the difference equation and system realization will be

*y*[*n*] = *x*[*n*] + α*x*[*n* - *m*] 🡪 1

Z-m

α

*x*[*n*]

*y*[*n*]

Fig.1

Where *m* is the delay in terms of samples. If the sampling rate 48000 samples/sec and the desired delay is 0.25 second, then *m* = 0.25 \* 48000 = 12000 samples. The gain α should be less than one for the system to be stable. It should be between 0.4 and 0.8. To derive the transfer function *H*(*z*) of the system we take the z-transform of the difference equation, Eq. 1.

*Y*(z) = *X*(z) + αz-m *X*(z) 🡪 2

🡪 3

The transfer function can be rewritten in a different way by multiplying and dividing the above equation by z*m*.

🡪 4

The frequency response *H*(Ω) of the system can be found by taking the discrete time Fourier transform (DTFT) of the difference equation Eq. 1 or replacing z in Eq. 3 by ejΩ.

*Y*(Ω) = *X*(Ω) + αz-m *X*(Ω) 🡪 5

🡪 6

The plot of the frequency response magnitude of the FIR system is shown in Fig. 2 below for a delay *m* = 8 and amplification α = 0.8.

Fig.2



To achieve multiple echoes then IIR comb filter is used that will have feedback of the output. The difference equation of the IIR filter and system realization are

*y*[*n*] = *x*[*n*] + α*y*[*n* - *m*] 🡪 7

Z-m

α

*x*[*n*]

*y*[*n*]

Fig.3

The transfer function *H*(*z*) and the frequency response *H*(Ω) of the system can be derived in a similar way as we did for the FIR system above.

🡪 8

🡪 9

The plot of the frequency response magnitude of the IIR system is shown in Fig. 4 below for a delay *m* = 8 and amplification α = 0.8.

Fig. 4



**lab Exercises**

**Task 1: Building the FIR comb filter in LCDK.**

Download the audio file “Sentence” from the course website to Matlab workspace. Listen to the audio file and write down what you heard. Let us call the downloaded audio *x*[*n*].

Write down the sentence: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Program the LCDK DSP board to be FIR comb filter to produce echo of the sound as described on the lab introduction. The difference equation of the echo FIR filter is

*y*[*n*] = *x*[*n*] + α*x*[*n* - *m*]

You need to choose a value for the sample delay *m* that will generate a delay of 100 to 200 millisecond. The value of *m* that generate a delay of specific millisecond will depend on the sampling rate. The gain α should be less than 1 for the system to be stable. The value of α should be between 0.4 to 0.8. Try different values of *m* and α to produce the best echo effect.

Paly the sentence or any audio file from YouTube to the audio input terminal and listen to the audio signal from the audio output terminal of the LCDK. Describe the output audio signal. You can experiment with changing the amplification α and the delay *m*. Write the values of α and *m* that give you best echo performance.

**Task 2: Building the IIR comb filter in LCDK.**

Program the LCDK DSP board to be IIR comb filter to produce echo of the sound as described on the lab introduction. The difference equation of the echo IIR filter is

*y*[*n*] = *x*[*n*] + α*y*[*n* - *m*]

Paly the sentence or any audio file from YouTube to the audio input terminal and listen to the audio signal from the audio output terminal of the LCDK. Describe the output audio signal. You can experiment with changing the amplification α and the delay *m*. Write the values of α and *m* that give you best echo performance. Compare the outputs of the FIR and IIR filters.

**Material that you may need**

// Filename: ISRs.c

// Synopsis: Echo Comp Filter

#include "DSP\_Config.h"

// Data is received as 2 16-bit words (left/right) packed into one

// 32-bit word. The union allows the data to be accessed as a single

// entity when transferring to and from the serial port, but still be

// able to manipulate the left and right channels independently.

#define LEFT 0

#define RIGHT 1

volatile union {

Uint32 UINT;

Int16 Channel[2];

} CodecDataIn, CodecDataOut;

/\* add any global variables here \*/

float xLeft, xRight, yLeft, yRight;

Uint32 oldest = 0; // index for buffer value

#define BUFFER\_LENGTH 6000 // buffer length in samples

#pragma DATA\_SECTION (buffer, "CE0"); // put "buffer" in external SDRAM

volatile float buffer[2][BUFFER\_LENGTH]; /\* need space for left and right \*/

volatile float gain = 0.5; /\* set gain value for echoed sample \*/

void ZeroBuffer() // Call this function from the StartUp.c file

// Purpose: Initial fill of all buffer locations with 0.0 call this function from StartUp.c

{

int i;

for(i=0; i < BUFFER\_LENGTH; i++) {

buffer[LEFT][i] = 0.0;

buffer[RIGHT][i] = 0.0;

}

}

interrupt void Codec\_ISR()

// Purpose: Codec interface interrupt service routine

// Calls: CheckForOverrun, ReadCodecData, WriteCodecData

{

/\* add any local variables here \*/

if(CheckForOverrun()) // overrun error occurred (i.e. halted DSP)

return; // so serial port is reset to recover

CodecDataIn.UINT = ReadCodecData(); // get input data samples

xLeft = CodecDataIn.Channel[LEFT]; // current LEFT input value to float

xRight = CodecDataIn.Channel[RIGHT]; // current RIGHT input value to float

/\* add your code starting here \*/

/\* end your code here \*/

CodecDataOut.Channel[LEFT] = yLeft; // setup the LEFT value

CodecDataOut.Channel[RIGHT] = yRight; // setup the RIGHT value

WriteCodecData(CodecDataOut.UINT); // send output data to port

}

**LAB REPORT FORMAT**

Please use D2L to submit the lab report electronically. In case there is a technical issue with D2L, please email the lab report to the course instructor before the cutoff time. The lab report should include: (1) Title Page; (2) Introduction; (3) Results (Include programs, answers, and figures to all steps in Lab Procedure); (4) Conclusion/Discussion.