Electrical Engineering 434

Computer Exercise 3, Fall 2004: VOICE REMOVAL SYSTEM

This assignment is a group exercise with 3 due dates:
Lesson 37 COB: Matlab Design
Lesson 40 COB: 6711 DSP Implementation
Lesson 42: Group Presentation

1. Introduction

For this exercise, your instructor will form you into teams of 3 to 4 students that will simulate a small DSP company that is competing for a design contract. Use your judgment and the knowledge you've gained so far in EE434 to design an audio Voice Removal system. You can employ any basic DPS techniques you wish, and use any FIR or IIR filter design tools available in Matlab to meet the project specifications. Any specifications you are unable to meet or any engineering trade-offs you must decide between must be fully justified. Your professor and a panel of reviewers will attend your company's final presentation. The panel will review your design, evaluate your system based on the stated objectives, and assess your response to any questions. The panel will determine the "winning" company. All members of a given company will share the same basic group score for this exercise, weighted by their individual contribution (discussed later). The winning company normally receives a grade in the "A" range. The other companies normally receive grades between a 0 and an 87.

2. Background

In this exercise you will design and implement a simple voice removal DSP application, removing the lead singer from an audio recording. The attached article by Terry Weeder discusses the basic concept of voice stripping and the details of his analog implementation. Your system will be a DSP implementation, replacing all Weeder's analog electronics shown in his Figure 2 with the Texas Instruments 6711 DSP chip and your software.

Voice removal works on the assumption that the lead singer's voice is equally recorded (magnitude and phase) in both the left and right channels (i.e., center stage), while the background instruments have a stronger amplitude and phase shift in either the left or right channel (i.e., stage left or stage right). This assumption usually does not hold for bass frequencies in which all signals tend to be equally recorded in both the left and right channels. Therefore, the voice removal application must first separate the bass from the rest of the signal. Then the remaining signals of the left and right channels are subtracted, resulting in a new signal where the background instruments and singers are present, but the lead singer is removed (or reduced). Finally the bass signal is added back to the instrument signal, resulting in a voice removed signal. Weeder's Figure 1 illustrates this for his analog system. Below is a similar block diagram.

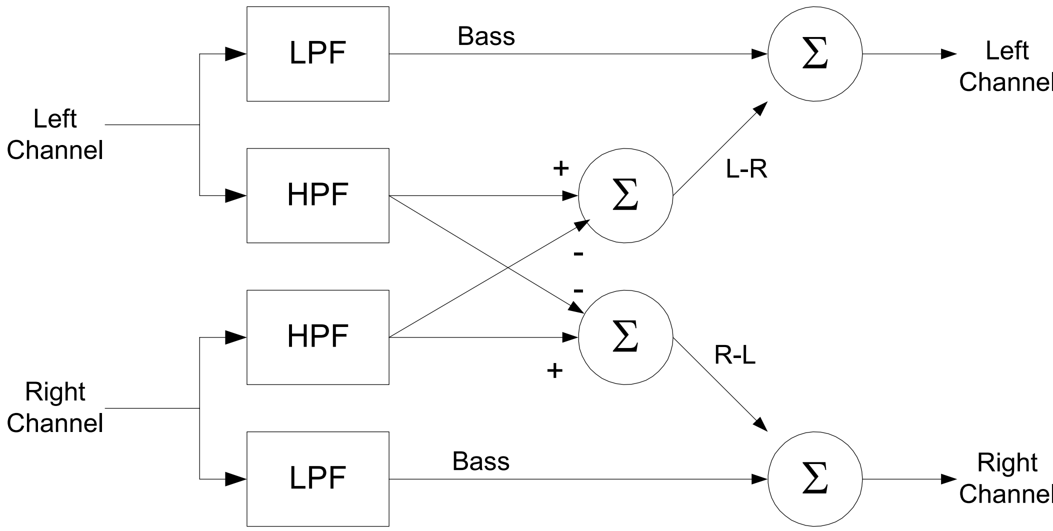

Figure 1. Block Diagram of a Voice Removal System

A side effect of this technique is that it results in a mono signal; stereo is lost. This technique also only works for CDs recorded following the prior assumptions. This will not work for CDs recorded with certain audio special effects, like reverb.

This application has an obvious commercial application for making cheap Karaoke machines. Does it have a military application? Yes, there are several intelligence applications that require separating the voice signal from the background signal, then processing them separately to gain some rather interesting intelligence.

3. Design Specifications

Your filters must meet the following specifications.

- Cutoff Frequency: 200 Hz (separates bass from the mids and highs)
- Transition Bandwidth: 50 Hz (from 175 to 225 Hz)
- Maximum Passband Ripple: 3 dB
- Minimum Stopband Attenuation: 20 dB
- Sample Frequency: 48000 Hz (speed of codecs on DSP board)
- The DSP boards A/D converters sample at 16 bits per sample per channel
- Are Linear Phase filters needed?
- Group delay of filters should be taken into account when adding signals together (i.e., must be in phase… not delayed with respect to each other)
- Filter Order? The processing capability of the 6711 DSP chip will dictate your maximum filter order. See following paragraph

As each new left and right sample enters this system, all this processing must be completed and output before the next left and right sample arrive (20.8 microseconds at 48000 samples/second). Assuming floating point math using the C language on the 6711, you have enough time to implement four filters of about order 135; or two filters of order 270; or one filter of order 540 [a hissing sound is a good indication you exceeding the computation time per sample]. You can experiment to see if these numbers can be pushed farther.

You may find it challenging to implement Figure 1 and meet all the specifications. This will require either making justifiable engineering trade-offs OR finding faster implementations which still meet the specifications.

Some ideas on faster implementations:

- It may be possible to reduce the system from 4 filters down to 2. With even more creativity, it may be possible to reduce to complete system down to one filter and one subtractor. Some alternative block diagrams may be found on the World-Wide-Web.
- Switch from Floating point to Fixed-point math. (only for advanced companies)
- Switch from C to assembly language (probably beyond expectations for your company)

After thoroughly researching voice removal, your company can recommend changes to the specifications or proposed implementation method. These changes must be justified both in your matlab design report, your 6711 DSP implementation report, and in your group presentation. This can be risky if the panel disagrees with your specification changes, however, can put you ahead of the other companies if the panel feels your changes will result in an improved voice removal system.

4. Testing/Analysis

In the **first part** of this exercise you will design your system using Matlab. Two test files are provided on the website for your testing and analysis.

- *gen_tones.m*: running this function generates two signals of about 48000 samples, sampled at Fs = 48000 Hz of simple tones illustrated in Figure 2 and Figure 3. This file is useful for a detail *quantitative* analysis of your system. To use:

  x = gen_tones;

  To clarify: your report's analysis should show the gen_tones spikes you want to keep are within 3 dB of their original value. The spikes you do not want to keep should be reduced at least 20 dB from their original value.

- *buffet.wav*: Contains two channels of 1102500 samples (25 seconds) sampled at Fs = 44100 Hz of a song by Jimmy Buffet. Voice removal works fairly well on this recording. This file is useful for *qualitative* analysis of your system. Note the slightly different sample rate than that of your system. To use:

load buffet.bin -mat;

% to hear

soundsc(x, Fs);

- The above files are meant to be a ***starting*** point for your testing/analysis. You are encouraged to find other methods/files for testing your design.

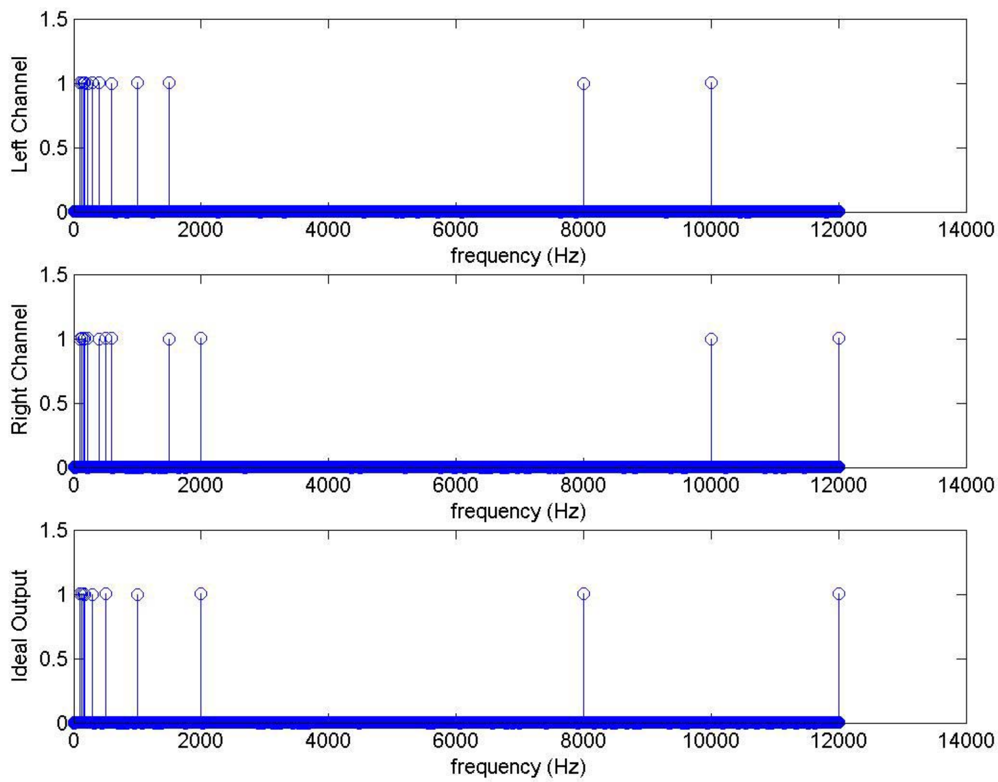

Figure 2. Spectrum of the left and right channel signal generated by gen_tones.m. The ideal output plot illustrates the results after voice removal. Note: Frequencies above 200 Hz shared by the left and right channel are eliminated. See Figure 3, zooming in to lower frequencies.

In the **second part** of this exercise, you will implement your Matlab design in C language on the 6711 DSP board. Your instructor will provide information at a latter time to your company on the details of setting up and programming the 6711 DSP, along with an example program for you to modify (see website). You will be left to your own ingenuity to find methods to test your system both *quantitatively* and *qualitatively*. How can you quantitatively measure the output of this real time system, with the data steaming by so fast? Hint: **See Figure 4**. You can use a second computer with a second DSK board. The second system can sample the output of the 1st system (which is performing voice removal). You could either use MATLAB on the second system to store the real time data, and plot the results as you wish either in real-time or off-line (see the *scope2.m* program on the website). Or you could use the WinDSK tool you used in lab to monitor the signal. Plots of this real time data will be useful in your report. To create a live signal, you can send (or modify) the *gen_tones.m* signal (or *buffet*) out the laptop audio port using matlab's *soundsc(x, Fs)* command. Or you can burn your own audio CDROM with test tones.

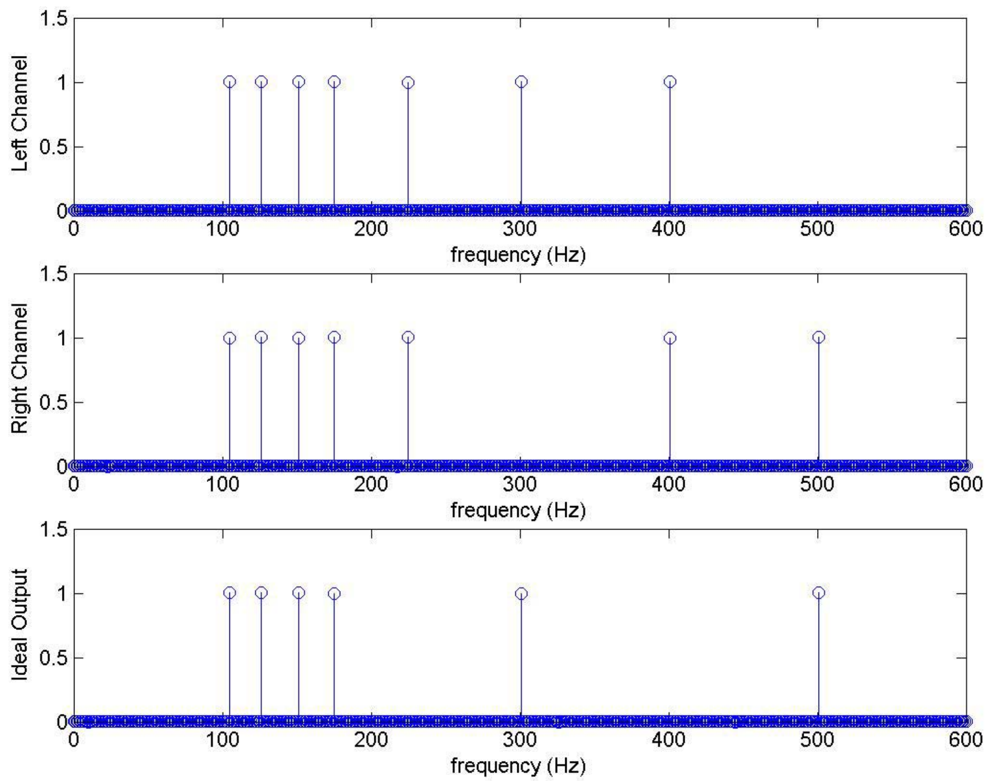

Figure 3. Zoomed plot of Figure 2. The ideal output plot illustrates the results after voice removal. Note: Frequencies above 200 Hz shared by the left and right channel are eliminated. All others are kept.

The third part of this exercise is your group presentation, attempting to convince the panel that your company has the best design and implementation. Your presentation of your testing and analysis results of the Matlab design and the 6711 implementation will be the key component for your company to win.

5. CompEx Requirements

For both your Matlab design and your 6711 DSP implementation, your filter design and overall system will be evaluated in terms of correctness of approach, achievement of the design objectives, and the quantitative and qualitative performance of your system. If your 6711 DSP implementation is different than your Matlab design your company will be *penalized* a significant amount of points.

By **COB Lesson M37**, email your professor your company's **Matlab design**. This should include a matlab file that will simulate your system using the given test files and any other files you've created. The matlab file should produce any relevant plots to quantitatively demonstrate the system's performance. In addition, a very concise report in Microsoft word should be included, discussing your system's design and the test/analysis results.

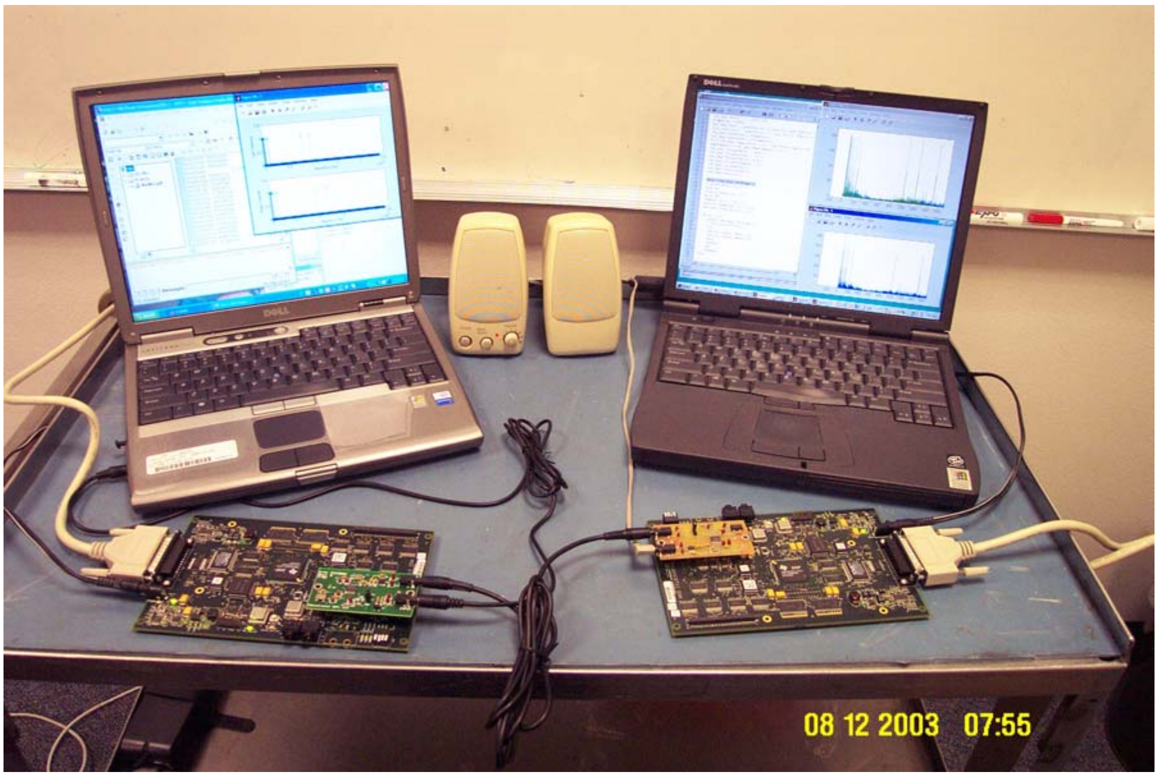

Figure 4. The laptop on the left loads and runs the voice removal C code on the left 6711 DSK and generates the source audio files from its CD player, while the right laptop running MATLAB serves as a spectrum analyzer along with the right 6711 DSK board that samples the output voice removed signal.

By **COB lesson M40**, email your professor your company's 6711 DSP implementation. The only file needed is the *"codec_low.c"*, along with a Microsoft word report, discussing your system's design, any differences with the Matlab design, and the test/analysis results (both qualitative and quantitative results).

Important note: When the 6711 DSP implementation is turned in lesson M40, your company's system design is then frozen. By COB lesson M40 each company must supply all the other competing companies in their EE434 section their matlab design via email (must include the filter coefficients and matlab code to simulate the system). You must CC your instructor on this email. This simulates a bit of industrial espionage. Each company has a chance to find any weak areas of flaws in their competitor's designs to exploit in their own presentation to the panel. Failure to provide this data on time in a usable manner will result in disqualification of the company from the competition and a very poor grade for all the members of the company.

Another important note: The individual's grade on this group work exercise is partially determined by a confidential "teamwork report." By **COB lesson M42**, *each student* must complete the on line "team report" (similar to a CPH) in which you evaluate yourself and your team mates efforts.

6. Presentation Requirements

The presentation should be very concise (no longer than 10 minutes), professional, and involve *every* member of the company. Your presentation will be evaluated based on correctness, completeness, organization, evidence of insight into DSP topics, communication skills, and how your presentation compares to the presentations of the other companies. As a minimum, your presentation should include the following

- Justification for your design approach
- Frequency domain plots showing the magnitude and phase of any filters used, along with frequency domain plots of the overall system output for the given test files for the Matlab design.
- A table showing required specifications and the measured results for each specification using the test files for the Matlab design.
- A qualitative demonstration of the operational system on the 6711 DSP board

You should supply a hard copy of your presentation slides to your professor and the panel members. The presentation grading criteria is posted on the website, under lesson 42.

Don't wait until the last minute to do this!   Have fun...