1. INTRODUCTION:

According to numbers that WHO provides, only 10 percent of the hearing device necessity can be required to people who have hearing problem. This devices are mostly expensive and not suitable for continuously, daily, usage. Our aim in this project is to give all patients a same chance of using a hearing device without charging any extra money to hardware. I basically used DSP Board for solution. In my previous year I due to design hearing aid for Android Mobile Device. I was successful but the quality of the output sound wasn't as good as an input sound.

TMS320C5505 is a 16 bit fixed point DSP, it is mostly used for low power applications. Since I will be working on battery operating hearing aid device, this board is one of the best options for me. I use Code Composer Studio Development Environment (IDE) for implementation of the Hearing Aid Device code.

2. HEARING TEST SOFTWARE:

Our first assignment was developing hearing test software measuring someone's frequency response. I created an 8 band test system covering 0 to 5KHz with 10 KHz sampling frequency. I recorded -30dB sample frequencies (375Hz, 500Hz, 750Hz, 1KHz, 1.5KHz, 2KHz, 3KHz and 4KHz) from the webpage of New South Wales University. We used JAVA software language in order to create our test. Java code is attached at the appendix. Following is the user interface of our test executing program.

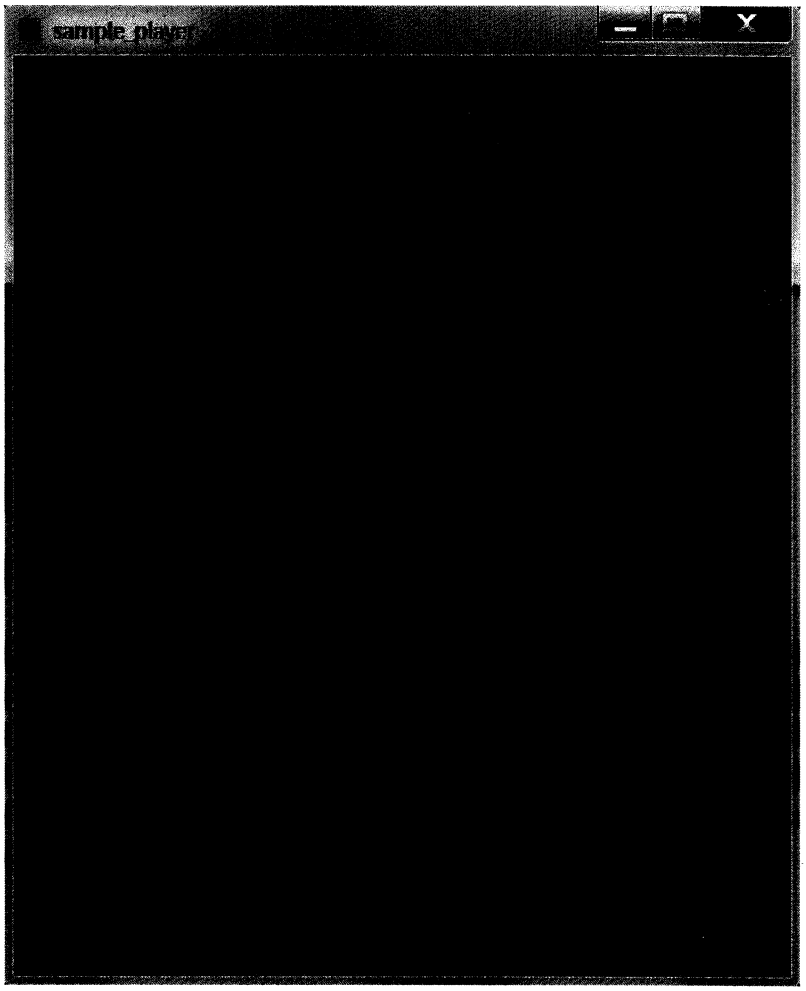


Figure 1: Sample_Player

Button 1 corresponds to 375Hz and it goes on up to button 8 which corresponds to 4KHz. When user presses one of the buttons from 1 to 8, program plays the corresponding frequency. If the person presses more than one time to the same button, volume of sound starts to increase.

3. FILTER DESIGN:

Our second assignment was recording a speech and put it into a bandpass filter. We used a 20^{th} order filter with sampling frequency 11025Hz. After filtering the signal, we divided it by 2 in order to prevent clipping.

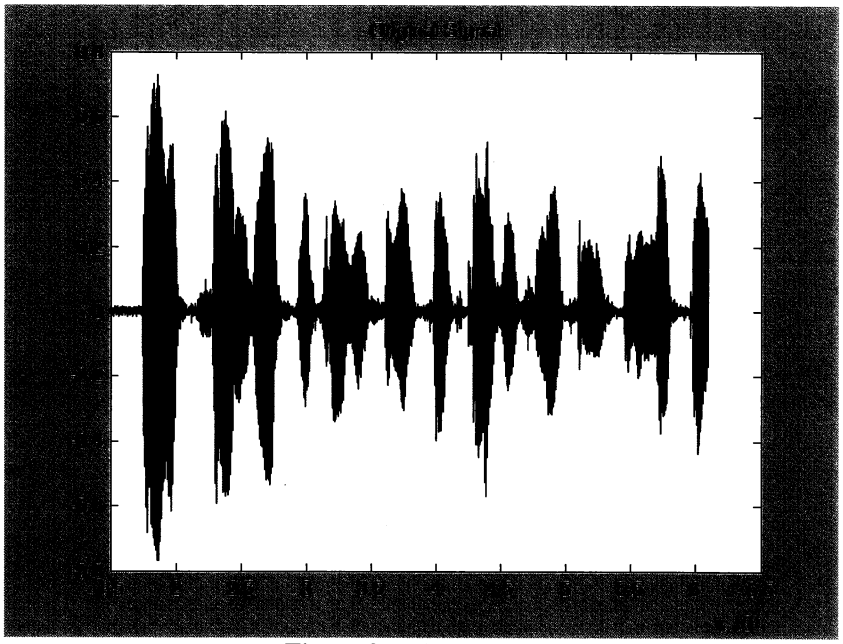


Figure 2: Original Signal

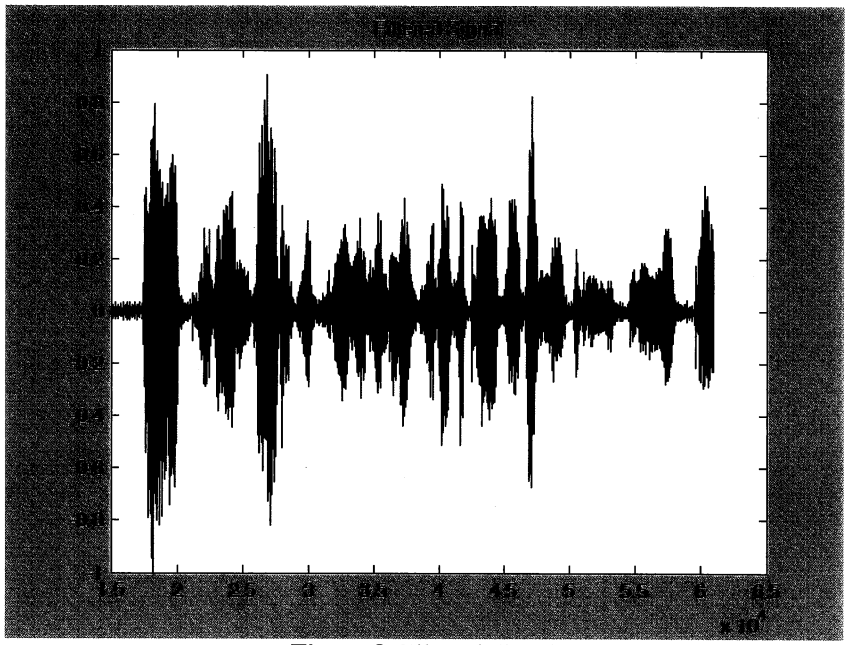


Figure 3: Filtered Signal

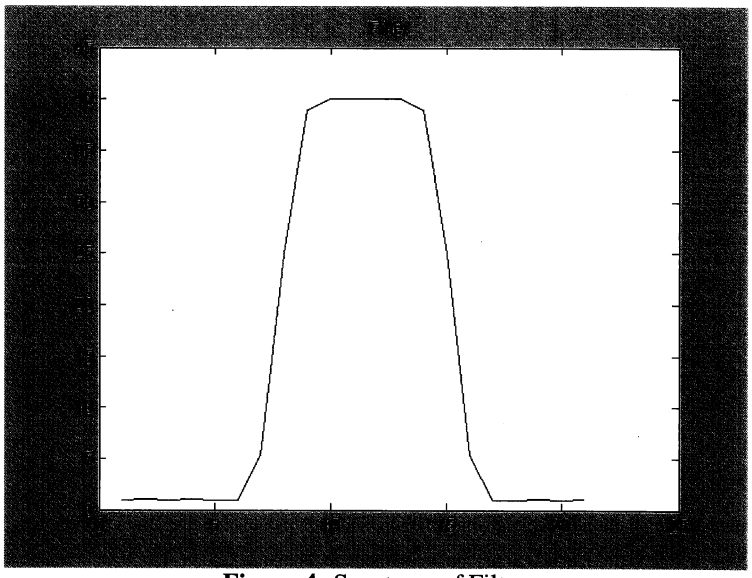


Figure 4: Spectrum of Filter

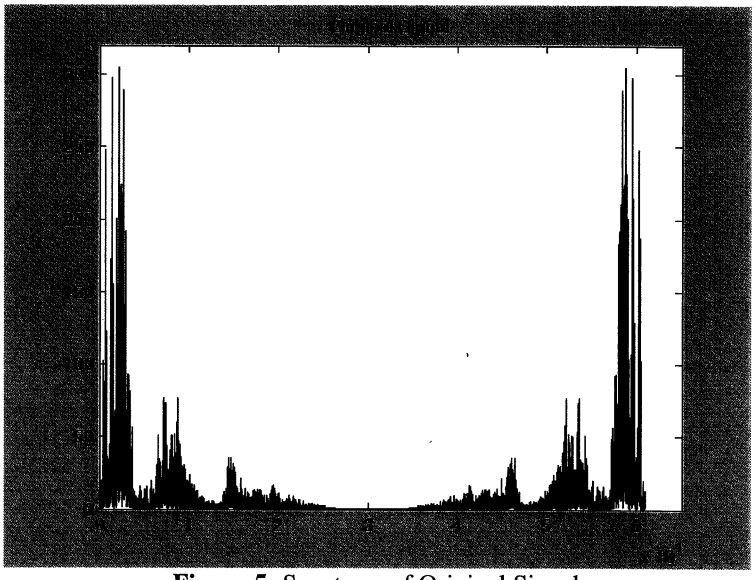


Figure 5: Spectrum of Original Signal

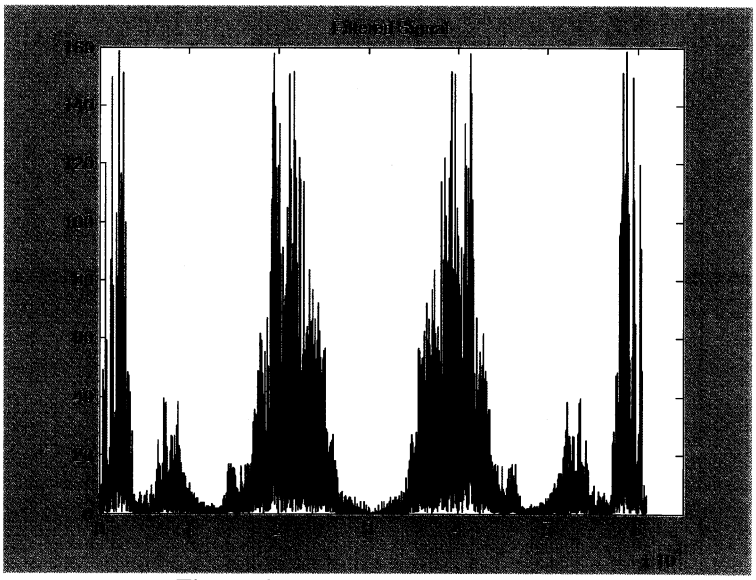


Figure 6: Spectrum of Filtered Signal

4. TMS320C5535 TI DSP BOARD IMPLEMENTATION

4.1. DSP Board vs. Android

In Android implementation I had encountered distortion and noises. Because of Automatic Gain Control, android mobile normalizes the output of the signal by boosting and lowering input from the microphone to match a preset level. So it doesn't allow me to capture a gain of audio in the level that I desired. But in DSP board there wasn't such a problem like that in FFT Class and Hamming class, I easily captured the desired output.

Second problem that I encounter is number of bands. I need to use eight bands to modify. However, I can use at most five bands. I did an internet based research about this issue and I couldn't find even a commercial application constructed on Android. There were lots applications for other operating systems. However, Android offers us only five bands to use.

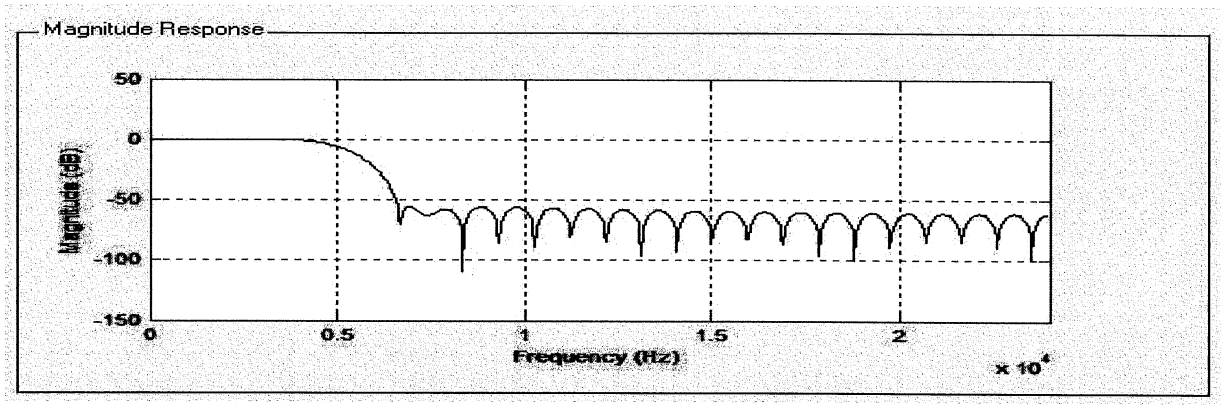
Third problem was choosing bands. Although I developed and used filters for any pre-recorded audio in DSP Board, I couldn't do the same process on cell phones for real time audio. We used all available five bands but, we cannot choose frequency intervals. According to our researches, in order to set the frequency values, we need to have the source of Android cell phone or we had to compile Android again. Both options are not valid for us. Therefore, we are forced to use pre-set frequency intervals.

4.2. Work Done:

In the main class used classes can be seen, I put the important ones to the appendix. By the help of a TI tutorial I found chance to look up lots of different DSP applications and try them. firstly, I removed the hamming low pass filter's coefficients and change them with the 21-order HPF that Prof. Dr. Enis Çetin has proposed. I preferred 21 order because as the order increases filter becomes sharper.

In main class I determined sample frequency as 12KHz since human can hear up to 25 KHz, it is in desired range. Also, I determined the input gain as 20dB.

Basically, C5505 takes convolution of the input voice, it passed through the filter. Convolution process is made in main class and the convolved signal is given to headset. Delay between input and output is very small and negligible.



Response of the 500Hz 50 Coeff. FIR filter. Source: Texas Instruments

5. CONCLUSION

We have estimated a person's spectrum in previous works. We did it with creating an exe. file from JAVA. After that we created a bandpass filter in order to increase amplitude of the speech.
In my Android Hearing Aid project I worked with my partner. We did this process for pre-recorded audio. However, we couldn't use this filter in real time audio files. According to our researches, in order to set the frequency values, we need to have the source of Android cell phone or we had to compile Android again. Both options are not valid for us. Therefore, we are forced to use pre-set frequency intervals. To conclude we can get the real time audio from microphone and increase the gain up to 30dB for five bands that are pre-specified. However, this process is a bit noisy in cell phones comparing to DSP or FPGA boards. I learned the implementation in Code Composer easily from the Tutorial series of TI and with internet research. As a result I reached to a good ending point for the Hearing Aid Project.

6. REFERENCES

1. http://www.ti.com/lsds/ti/dsp/tech_docs.page

2. http://dsptute.blogspot.com/

3. Digital Signal Processing, A Practical Approach by Emmanuel C. Ifeachor and Barrie W. Jervis. ISBN 0201-59619-9.