MULTIBAND FIR FILTER DESIGN AND IMPLEMENTATION TO RECOVER REAL- TIME CORRUPTED INPUT SPEECH USING DSK TMS320C6711

Sung Ho Van, Que Le Thanh

Faculty of Electronics and Telecommunication, College of Technology, VNU

Abstrac. In this paper, the five-band notch filter was designed using MATLAB and implemented by the floating-point digital signal processor TMS320C6711 based on TI's DSP starter kit (DSK) board connected to the parallel port of the PC through the DB25 cable included with the DSK package. The original segment of symphony orchestra corrupted two undesired sinusoids at frequencies of 750Hz and 1750Hz is conduced to input of DSK to produce the corrupted input signal which is saved in the wave file (Corrupthandel.wav) for the digital signal processor. The two noise sinusoidal signals are generated by the real-time signal generator. The output signal of DSK is plotted on an digital oscilloscope, on CCS window or on an earphones.

1. Introduction

Filtering is one of the most useful signal processing operations. In this paper, we employ the TI's DSK TMS320C6711 to implement Finite Impulse Response (FIR) digital filters for recovery a noise corrupted input speech-signal in real time. The TMS320C6x instruction set and architecture makes it well suited for such filtering operations. In the last few years, the cost of digital signal processors has been reduced significantly, therefore the using digital filters becomes popular because of it's high reliability, accuracy and less sensitivity to temperature and ageing. The characteristics of digital filters such as center frequency, bandwidth, attenuation and filter type can be modified flexibly. A number of tools are available to design and implement the different FIR filters in real time. The FIR filters such as lowpass, highpass; bandpass, bandstop and notch are now designed and implemented using DSK TMS 320C6711 in combination with MATLAB. These filters with high quality are employed to cancel a multi-tone noise for a input voice.

To perform the experiments, the following tools are used:

1. TI's DSP starter kit (DSK). The DSK package includes:

- Code Composer Studio (CCS), which provides the necessary software support tools and an integrated development environment (IDE) bringing together the C compiler, assembler, linker, debugger and so on.

- A board that contains the TMS320C6711 floating point digital processor as well as 16-bit codec for input and output support.

- A parallel cable (DB25) that connects the DSK board to PC.

- A power supply for the DSK board.

2. A Pentum compatible PC.

3. A digital oscilloscope, signal generator and earphones. Shareware utilities are available that utilize the PC and sound card to create a virtual instrument such as an oscilloscope, a function generator or a spectrum analyzer.

2. FIR filter structures

The input-output relationship of a N-order causal FIR digital filter is described the following difference equation

$$y[n] = \sum_{i=0}^{N} b_i x[n-i] \tag{1}$$

where b_i, i = 0, 1, 2, …, N are the constant coefficients and just as the impulse response $h[i]$ of this FIR filter. The filter design with the desired frequency response consists of the determining these coefficients b_i. In other words, N+1 values of impulse response should be determined. The difference equation (1) is a convolution between N+1 input samples $x[n]$ and N+1 samples of a impulse response $h[i]$ as

$$h[i] = \begin{cases} b_i, \ i = 0,1,\ldots,N \\ 0, \ other \quad values \end{cases} \tag{2}$$

The direct form structure of the difference equation (1) is plotted in the figure 1. This structure requires N+1 multipliers, N two-input adders and N unit-delays for implementation. The difference equation (1) shows that the value of one output sample at time n can be determined with knowledge of the N+1 values of input signal included the present input $x[n]$ and N values of the past inputs $x[n\text{-}i]$. Thus, the FIR filters are called transversal or tapped-delay filters.

If its impulse response $h[n]$ is either symmetric, i.e.

$$h[n] = h[N\text{-}1], \ 0 \le n \le N, \tag{3}$$

or is antisymmetric, i.e.,

$$h[n] = -\, h[N\text{-}1], \ 0 \le n \le N, \tag{3}$$

then the FIR filter has a linear phase. Sine the order of a filter can be either even or odd; it can define four types of symmetry for the impulse response. Figure2 and 3 is plot of linear- phase FIR structures with N even and odd. It should be noted that the linear- phase FIR structures reduces a half of multipliers in comparison with the direct form realization of the original FIR filter.

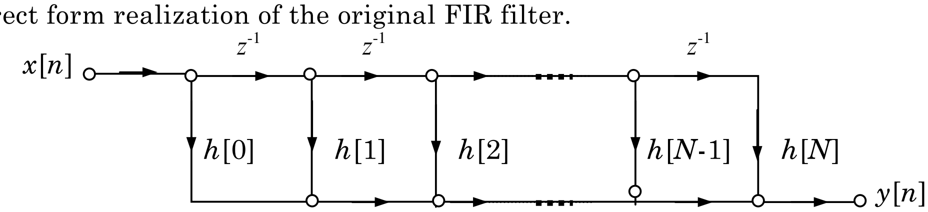

Figure1. Signal flow graph of direct form structure of a FIR filter.

If the order N is even and it's impulse response is symmetric, then the filter is called *type 1 linear phase FIR filter*. In the general case for the type 1 FIR filters, the frequency response is of the form

$$H_1(e^{j\omega}) = |H_1(e^{j\omega})|.e^{-jN\omega/2} \tag{5}$$

where the amplitude response is given by

$$|H_1(e^{j\omega})| = \left| h\left[\frac{N}{2}\right] + 2\sum_{n=1}^{N/2} h\left[\frac{N}{2} - n\right]\cos(n\omega) \right| \tag{6}$$

If the order N is odd and the impulse response is symmetric, then the filter is called *type 2 linear phase FIR filter*. In the general case for the type 2 FIR filters, the frequency response is of the form

$$H_2(e^{j\omega}) = |H_2(e^{j\omega})|.e^{-jN\omega/2} \tag{7}$$

where the amplitude response is given by

$$|H_2(e^{j\omega})| = \left| 2\sum_{n=1}^{(N+1)/2} h\left[\frac{N+1}{2} - n\right]\cos(n - \frac{1}{2})\omega \right| \tag{8}$$

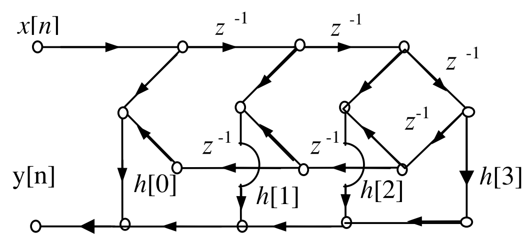

Figure2. Signal flow graph of the sixth-order linear-phase FIR filter.

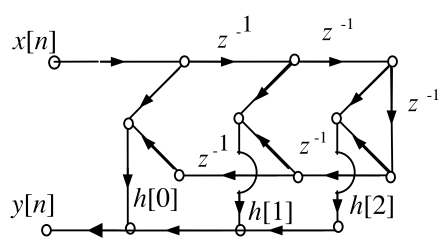

Figure3. Signal flow graph of the fifth-order linear-phase FIR filter.

If the order N is even and the impulse response is antisymmetric, then the filter is called *type 3 linear phase FIR filter*. In the general case for the type 3 FIR filters, the frequency response is of the form

$$H_3(e^{j\omega}) = |H_3(e^{j\omega})|.e^{-j(N\omega/2-\pi/2)}, \tag{9}$$

where the amplitude response is given by

$$|H_3(e^{j\omega})| = \left| 2\sum_{n=1}^{N/2} h[\frac{N}{2} - n]\sin(n\omega) \right| \tag{10}$$

$$h[N/2] = 0$$

If the order N is odd and the impulse response is antisymmetric, then the filter is called *type 4 linear phase FIR filter*. In the general case for the type 4 FIR filters, the frequency response is of the form

$$H_4(e^{j\omega}) = |H_4(e^{j\omega})|.e^{-j(N\omega/2-\pi/2)} \tag{11}$$

where the amplitude response is given by

$$|H_4(e^{j\omega})| = \left| 2\sum_{n=1}^{(N+1)/2} h[\frac{N+1}{2} - n]\sin(n - \frac{1}{2})\omega \right| \tag{12}$$

The FIR type 3 and 4 are the basis to design and implement the ideal digital differentiators or the Hilbert transforms.

An FIR filter with a frequency response that is a real function is often called a *zero-phase filter.* Such a filter must have a noncausal impulse response.

3. Multiband FIR filter design

The digital filters can be designed by using many MATLAB functions. Here, the MATLAB function *remez* uses the Parks-McClellan algorithm based on the Remez exchange algorithm and Chebyschev's approximation theory and the Kaiser - window function. The desired filter is the five-band notch filter with the center frequencies of 808 Hz and 1800Hz. In this design, the normalized frequency defined as 2f/F, where F is the sampling frequency (F = 8000Hz). This notch filter with two stopbands represented by a total of five bands: the first passband has normalized frequencies between 0 and 0.17 (0 and 680Hz) with corresponding magnitude of 1; the second band is the first stopband, which has normalized frequencies between 0.175 and 0.18 (700 to 720Hz), with a corresponding magnitude of 0; the second stopband has normalized frequencies between 0.395 and 0.4 (1580 to 1600Hz). The stopbands and passbands of this 88th order five-band notch filter are summarized on the following table 1.

Table 1. Specification of the five-band FIR notch filter

Band	Normalized frequency 2f/F	Frequency (Hz)	Magnitude
1	0 – 0.17	0 - 680	1
2	0.175 - 0.18	700 - 720	0
3	0.185 - 0.385	740 - 1540	1
4	0.395 - 0.4	1580 - 1600	0

	0.405 - 1	1620-4000	1

The figure 4 plotted the magnitude response and impulse response of the designed five-band notch filter. This program generated a set of 89 coefficients saved into the coefficient file *matnoch89.cof* in ASCII format. This coefficient file uses a float data format and is shown in the following table 2.

Table 2. Coefficients of the 88th-order five-band notch filter

Float h[n]=

{-0.1329 0.0859 0.0099 0.0082 0.0565 0.0844 0.0553 -0.0002 -0.0265 -0.0060 0.0276
0.0322 0.0048 -0.0205 -0.0146 0.0132 0.0296 0.0165 -0.0090 -0.0184 -0.0052 0.0099
0.0076 -0.0075 -0.0141 -0.0029 0.0131 0.0154 0.0035 -0.0070 -0.0050 0.0026 0.0029 -
0.0057 -0.0112 -0.0052 0.0059 0.0097 0.0042 -0.0011 0.0007 0.0051 0.0026 -0.0060
0.9888 -0.0060 0.0026 0.0051 0.0007 -0.0011 0.0042 0.0097 0.0059 -0.0052 -0.0112 -
0.0057 0.0029 0.0026 -0.0050 -0.0070 0.0035 0.0154 0.0131 -0.0029 -0.0141 -0.0075
0.0076 0.0099 -0.0052 -0.0184 -0.0090 0.0165 0.0296 0.0132 -0.0146 -0.0205 0.0048
0.0322 0.0276 -0.0060 -0.0265 -0.0002 0.0553 0.0844 0.0565 0.0082 0.0099 0.0859
-0.1329}.

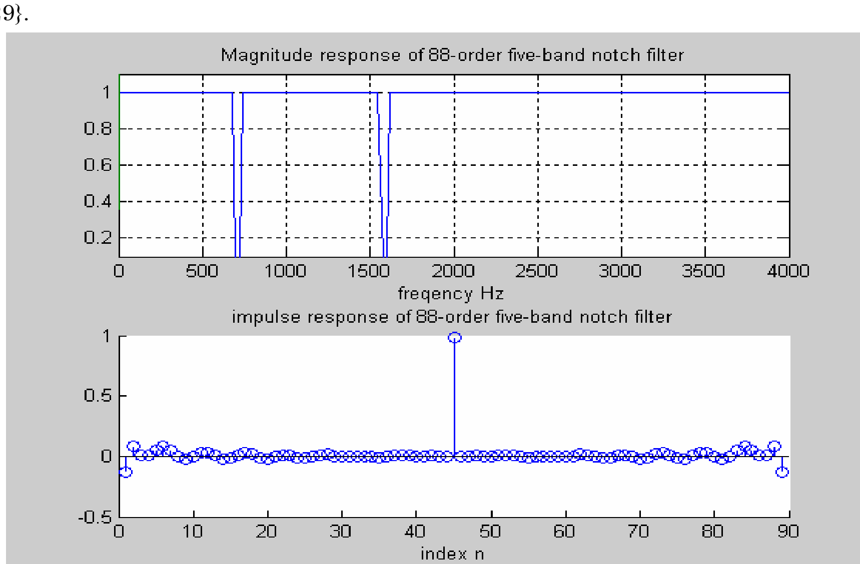

Figure 4.Magnitude response and impulse response of the 88-order five-band notch filter.

The multiband notch filters can be implemented by making in cascade many notch filters centered at different frequencies. In this method, a buffer is employed for the delay samples of each filter. The output of the first notch filter becomes the input to second notch filter, centered at adjoining frequency, and so on. The output of final notch filter is also the output of an overall multiband notch filter. Figure 5 shows the ideal impulse response, designed impulse response, the Kaiser-window and the magnitude response of the two FIR notch filters implemented by using the Kaiser-window function to truncate the impulse response of an ideal FIR filter. The β parameter of the Kaiser window is equal to 5.6533.

From the magnitude plot in figure5, we observe that the minimum stopband attenuation is smaller than 60 dB. The first notch filter centered at normalized frequency of 0.202 (808Hz) containing 89 coefficients. The output of this filter is the input to second notch filter centered at normalized frequency of 0.428 (1720Hz). The coefficients of these two notch filters are saved in two coefficients file, *notch1.cof* and *notch2.cof*, respectively.

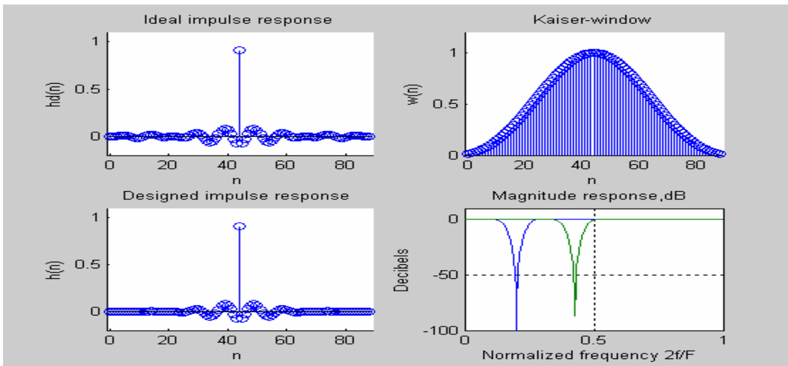

Figure 5. Ideal and designed impulse response and magnitude response of five-band notch filter designed using MATLAB with Kaiser window.

4. Multiband FIR implementation to recover corrupted input speech on DSK TMS320C6711

The five-band FIR filter is designed scarcely above linked to DSK TMS320C6711 to remove two undesired sinusoidal signals corrupting an input speech signal. In these studies, the input speech signal is a segment of symphony orchestra which is saved in the wave file (*Handel.wav*). This original segment of symphony orchestra was added with two undesired sinusoidal at frequencies of 750Hz and 1750Hz to produce the corrupted in put signal which is saved also in the wave file (*Corrupthandel.wav*) for the digital signal processor. The two noise sinusoidal signals are generated by the real-time low frequency generator.

To recover the original segment of symphony orchestra using TMS320C6711, the program MULTINOTCH.C is written, which implement the five-band notch filter with the coefficient file *fivenotch89.cof* listing in the table 2. This five- band notch filter is equivalent to two notch filters, each containing 89 coefficients and designed with MATLAB, centered at 750Hz and 1750Hz, respectively. The buffer dly[n] is used for the delay samples of notch filter.

The corrupted input signal, the recovered output signal and the amplitude response of five band notch filter can be obtained on the digital oscilloscope or on the CCS window of TMS320C6711. The size of input and output signals can be regulated by using a GEL file. Change the positions of the slider in program, we can observe the corrupted signal, the recovered signal at DSK output and it's spectrum as shown in the figure 6 and 7.

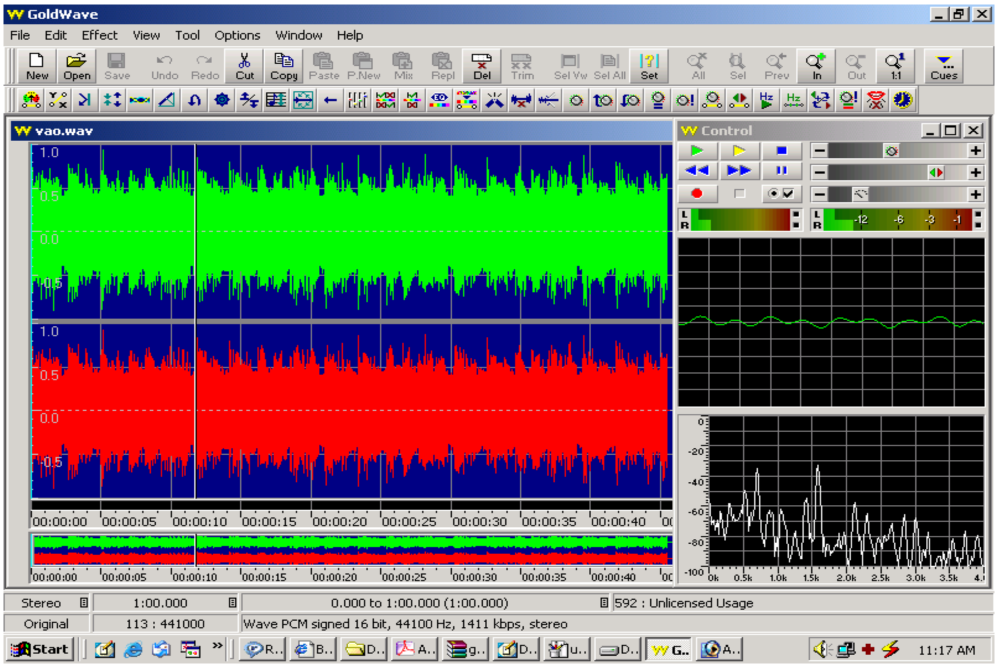

Figure 6. Input corrupted symphony orchestra and it's spectrum obtained on the virtual Instrument of Goldwave software.

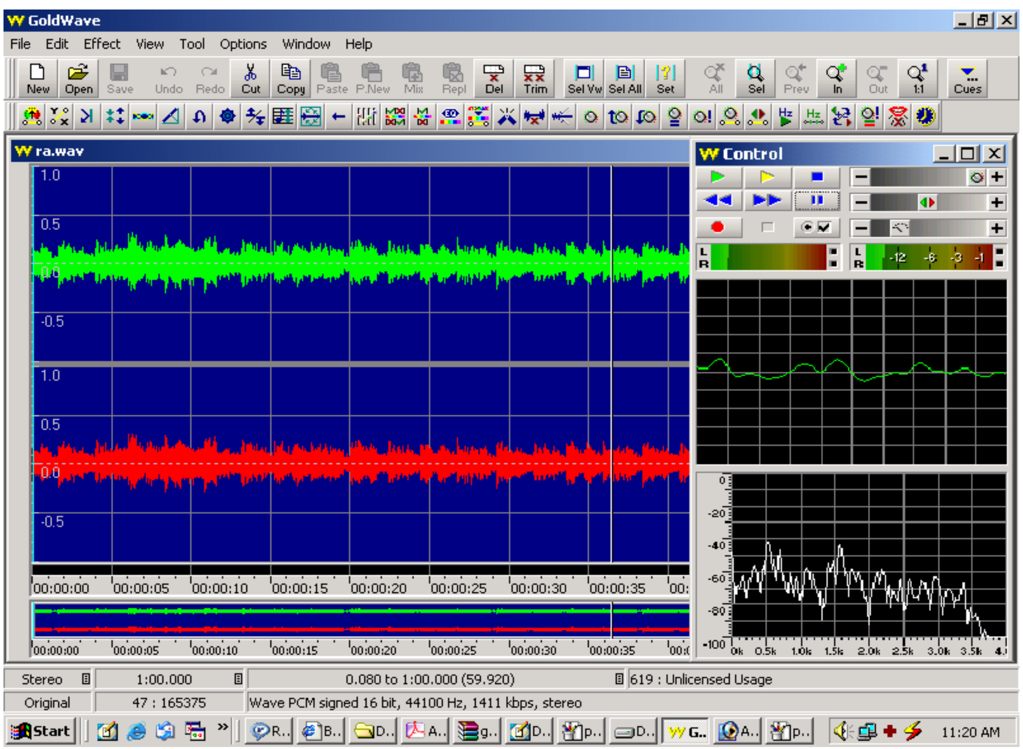

Figure 6. Output filtered symphony orchestra and it's spectrum obtained on the virtual instrument of Goldwave software.

5. Conclusion

Multiband FIR notch filters are designed and implemented to remove two or many undesired sinusoidal noise signals corrupting an arbitrary input voice signal. The stopband and the center cut-off frequencies of these filters can be adjusted by the access instructions in the operation C program of the TMS320C6711 floating - point digital signal processor. It is more convenient to implement multiband FIR notch filters by using C program calling ASM function. The resulted implementation would have faster operating speed. This problem will be reported in another paper.

References

1. Hồ Văn Sung, **Xử lý số tín hiệu, T1&2,** *Phương pháp truyền thống kết hợp với phần mềm MATLAB,* NXB Giáo Dục, Hà Nội 2003.
2. Hồ Văn Sung, *Thực hành xử lý số tín hiệu trên máy tính PC với MATLAB,* NXB Khoa học và Kỹ thuật, Hà Nội 2005.
3. *TMS320C6000 CPU and Instruction Set Reference Guide, SPRU189F,* Texas Instruments, Dallas, TX, 2000.
4. G. Pallot, *Processeurs de signaux et logique programmable,* cours et TP CNAM/MEDIAS 2002
5. S.K. Mitra, *Digital Signal Processing : A computer - Based Approach,* McGraw-Hill IRwin , 2001.
6. S. K. Oppenhaim and at.all: *Discrete-time Signal Processing,* Prentice Hall, 1999.
7. D.J.DeFatta, J.G.Lucas and W.S. Hodgkiss, *Digital Signal Processing: A System Design Approach,* Wiley, New York NY 1988.
8. R. Chassaing, *DSP Applications using C and the TMS320C6X DSK,* John Wiley & Sons , INC. 2002.
9. James H.McClellan, C. Sidney Burrus and at.all. *Computer - Based Exercises for Signal Processing Using MATLAB 5,* Prentice Hall, 1998.
10. *TMS320C6000 Code Composer Studio Tutorial, SPRU301C,* Texas Instruments, Dallas, TX, 2001.