Session:1432

Real-Time Noise Reduction Using the TMS320C31 Digital Signal Processing Starter Kit

Jianxin Tang
Division of Electrical Engineering
Alfred University
Alfred, NY 14802
Email: ftang@bigvax.alfred.edu

Abstract

This paper addresses real-time noise reduction using the low-cost TMS320C31 digital signal processing starter kit (DSK). Digital filters, including FIR low pass filter, FIR comb filter, and FIR averaging filter, are designed using MATLAB functions to generate coefficients associated with desired filter characteristics. The filter coefficients are then included in assembly language programs that implement these digital filters. A MATLAB program is used to activate a digital filter, and calculate and plot the frequency response of the filter. When the MATLAB program is run, it plots the spectrum of the filter on the PC screen, assembles the filter assembly language program, and loads/runs the resulting executable file on the TMS320C31 to achieve real-time filtering. Sinusoidal, square, and triangular waveform signals corrupted with white noise are used as input signals to test the filters. Results show the improvement of filter outputs as expected, with actual filter outputs matching theoretical calculations.

1. Introduction

One of the most common problems in signal processing is to extract a desired signal from a noisy measured signal[1]. The noise component of the signal depends on the application. For example, it could be (1) a white noise signal, which is typical of the background noise picked up during the measurement process; (2) a periodic interference signal such as the 60 Hz power-frequency pickup; (3) a low-frequency noise signal, such as radar clutter; (4) any other signal-not necessarily measurement noise-that must be separated. The use of digital signal processors (DSPs) has permitted the increasingly stringent performance requirements and fast, efficient, and accurate filtering of the desired signal from the noise. DSPs, such as the TMS320C31 (C31) from Texas Instruments, are currently used for a wide range of applications from controls and communications to speech processing. They continue to be more and more successful because of available low-cost support tools. DSP-based systems can be readily reprogrammed for a different application.

The C31-based $99 DSK includes Texas Instruments' C31 floating-point digital signal processor, and an analog interface circuit (AIC) chip with A/D and D/A converters, input (anti-aliasing) and output (reconstruction) filters, all on a single chip. It also includes an assembler, a debugger, and many application examples.

This paper addresses real-time noise reduction using the low-cost TMS320C31 (DSK). Digital filters, including FIR low pass filter, FIR comb filter, and FIR averaging filter, are designed using MATLAB functions to generate coefficients associated with desired filter characteristics. The filter coefficients are then included in assembly language programs that implement these digital filters. A MATLAB program is used to activate a digital filter, and calculate and plot the frequency response of the filter. When the MATLAB program is run, it plots the spectrum of the filter on the PC screen, assembles the filter assembly language program, and loads/runs the resulting executable file on the TMS320C31 to achieve real-time filtering. Sinusoidal, square, and triangular waveform signals corrupted with white noise are used as input signals to test the filters. Results show the improvement of filter outputs as expected, with actual filter outputs matching theoretical calculations.

In Section 2, design of the FIR low pass filter, the FIR comb filter, and the averaging filter is discussed. Implementation of the filters using the C31 is addressed in Section 3. Test results of real-time filtering and comparison with theoretical calculations are presented in Section 4. Finally conclusion and future work are given in Section 5.

2. Design of the Digital Filters for Noise Reduction

The discussion of digital filters can be found in many standard textbooks[1,2,3]. The design of FIR low pass filters can be realized using MATLAB function *remez*[4]. The FIR low pass filter is used to extract a sinusoidal signal from a noisy measurement signal. In this paper a 55th order FIR filter was designed with a passband of 2000 Hz. Its frequency response is displayed in Figure 1.

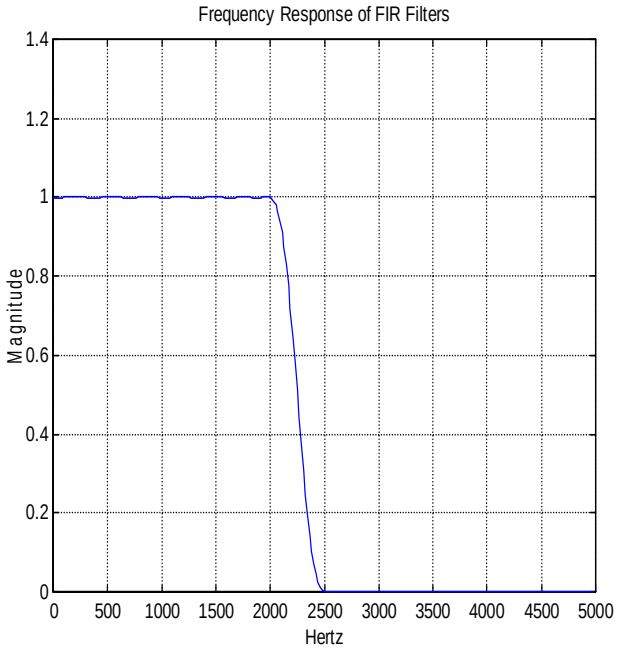

Fig.1. Frequency response of the FIR lowpass filter

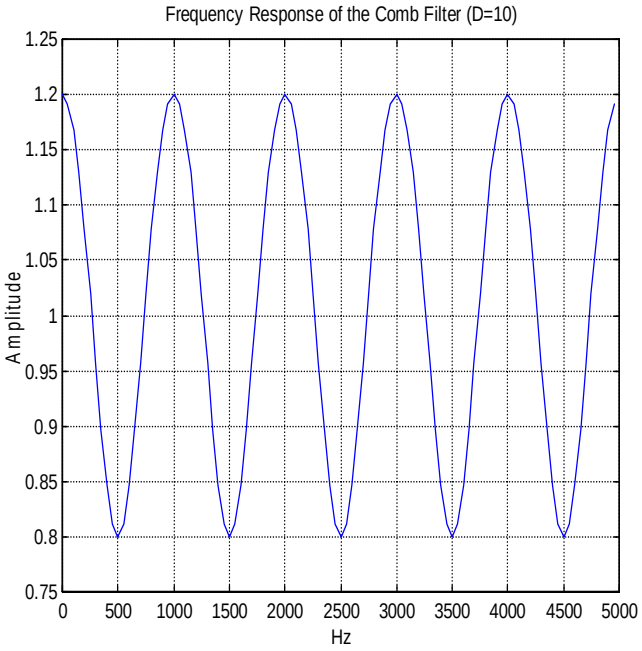

Fig.2. Frequency response of the FIR comb filter (D=10)

A FIR comb filter has the following transfer function[1]

$$H(z) = 1 + az^{-D} \qquad (1)$$

with zeros located at

$$z_k = \rho e^{\pi j(2k+1)/D}, \quad k=0,1,\ldots D\text{-}1 \qquad (2)$$

where $\rho = a^{1/D}$. At the dip frequencies $\omega_k = (2k+1)\pi/D$, we have $e^{j\omega_k D} = e^{j\pi}$=-1, giving $H(\omega_k) = 1-a$. Between the dip frequencies, that is, at $\omega_k = 2k\pi/D$, we have peaks with value $H(\omega_k) = 1+a$, because $e^{j\omega_k D} = 1$. In units of Hz, these peak frequencies are:

$$f_k = k\frac{f_s}{D} = kf_1, \quad k=0,1,\ldots D\text{-}1 \qquad (3)$$

where f_s is the sampling frequency. The magnitude response of the FIR comb filter with D=10 and a=0.2 is shown in Figure 2. The noise reduction ratio (*NRR)* of this filter is $NRR = (1-a)/2 = 0.4$, which corresponds to a $10\log_{10}(1/NRR) = 4$ dB improvement of the signal-to-noise ratio (SNR).

A widely used alternative method for noise reduction is *signal averaging*[1]. The transfer function has the following form:

$$H(z) = \frac{1}{N}(1 + z^{-D} + z^{-2D} + \cdots + z^{-(N-1)D}) = \frac{1}{N}\frac{1-z^{-ND}}{1-z^{-D}}. \qquad (4)$$

With D=10 and N=5, the frequency response is very similar to the one in Figure 2.

3. Implementation of the Noise Reduction Filters Using the TMS320C31 DSK

Figure 3 shows the MATLAB main program (NOISE1.M) for noise reduction using the TMS320C31 DSK. For the FIR lowpass filter, the MATLAB function *remez* is used to generate the filter coefficients (Line 9 in the program). For the FIR comb filter and the signal averaging filter, filter coefficients are obtained by inspection and are included in Line 11 and 12 of the program. MATLAB functions *freqz* and *plot* are used to calculate and plot the frequency response of the filter. The actual implementation of the filters on the DSK is through the assembly language program NOISE1.ASM (figure 4). This program is a modification of a FIR filter program in [2]. After defining starting addresses for the text and data, the AIC communication routine AICCOM31.ASM[2] is included for input to and output from the C31 (via the I/O routine AICIO_P). The input sample is then processed according to transfer functions of the filters. One of the features of the C31 is that it can perform multiplication and addition/subtraction in parallel. This is especially suitable for processing difference equations represented in transfer functions (1) and (4). An output signal is generated and the delays are updated. Continuous processing takes place within a loop starting at the label FILT in the program in figure 4. Filter parameters are in a

separate coefficient file fn.COF (Figure 5) and are “included” in the NOISE1.ASM through Line 5 of Figure 4. The sampling frequency is configured to be 10 KHz in the statement labeled AICSEC.

```
%NOISE1.M–Main MATLAB program for noise reduction
%
fs = 10000;
T = 1/fs;
%
N = 54;
%f=[0 0.4 0.5 1];
%m=[1 1 0 0];
%[b] = remez(N,f,m);
%
b=[1.0,0,0,0,0,0,0,0,0,0,0.2];
a=1;
[H, wT]=freqz(b,a,100);
herts=wT/(2*pi*T);
plot(herts, abs(H)), grid
title('Frequency Response of the Comb Filter (D=10)')
xlabel('Hz'), ylabel('Amplitude')
dos('dsk3a noise1');
dos('dsk3load noise1 BOOT');
```

Fig. 3. Main MATLAB Program NOISE1.M

```
;NOISE1.ASM – GENERIC FIR FILTER. INCLUDE COEFFICIENTS FILE
        .start   ".text",0x809900   ;starting address of text
        .start   ".data",0x809C00   ;starting address of data
        .include "AICCOM31.ASM"     ;AIC communication routines
        .include "NOISE1.COF"          ;coefficients file
        .data                       ;data section
XB_ADDR .word    XN+LENGTH-1        ;last (bottom) sample address
HN_ADDR .word    COEFF              ;starting addr of coefficients
AICSEC  .word    162ch,1h,3872h,67h ;AIC configuration data
        .brstart "XN_BUFF",64       ;align samples buffer
XN      .sect    "XN_BUFF"          ;buffer section for samples
        .loop    LENGTH             ;loop LENGTH times
        .float   0                  ;init samples buffer to zero
        .endloop                    ;end of loop
        .text                       ;text section
        .entry   BEGIN              ;start of code
BEGIN   LDP      AICSEC             ;init to data page 128
        CALL     AICSET             ;init AIC
        LDI      LENGTH,BK          ;BK=size of circular buffer
        LDI      @XB_ADDR,AR1       ;AR1=last sample address
FILT    LDI      LENGTH-1,AR4       ;AR4=length-1 as loop counter
LOOP    CALL     AICIO_P            ;AICIO routine,IN->R6 OUT->R7
        FLOAT    R6,R3              ;input new sample ->R3
        STF      R3,*AR1++%         ;store newest sample
        LDI      @HN_ADDR,AR0       ;AR0 points to H(N-1)
        LDF      0,R0               ;init R0
        LDF      0,R2               ;init R2
```

Fig.4. Assembly language program NOISE1.ASM

```
        RPTS     LENGTH-1           ;repeat LENGTH-1 times
        MPYF3    *AR0++,*AR1++%,R0  ;R0 = HN*XN
||      ADDF3    R0,R2,R2           ;accumulation in R2
        DBNZD    AR4,LOOP           ;delayed branch until AR4<0
        ADDF     R0,R2              ;last accumulation
        FIX      R2,R7              ;convert float R2 to integer R7
        NOP                         ;added due to delayed branch
        BR       FILT               ;branch to filter routine
        .end                        ;end
```

Fig.4. Assembly language program NOISE1.ASM(cont'd)

```
;NOISE1.COF - LOWPASS FIR FILTER COEFFICIENTS
       .data                        ;coefficients section
COEFF  .float   3.6353E-003,-1.1901E-003,-4.5219E-003, 2.6882E-003, 5.1775E-003
       .float  -4.7252E-003,-5.4097E-003, 7.2940E-003, 4.9986E-003,-1.0343E-002
       .float  -3.6979E-003, 1.3778E-002, 1.2276E-003,-1.7460E-002, 2.7529E-003
       .float   2.1222E-002,-8.7185E-003,-2.4870E-002, 1.7465E-002, 2.8205E-002
       .float  -3.0642E-002,-3.1035E-002, 5.2556E-002, 3.3190E-002,-9.9226E-002
       .float  -3.4540E-002, 3.1598E-001, 5.3500E-001, 3.1598E-001,-3.4540E-002
       .float  -9.9226E-002, 3.3190E-002, 5.2556E-002,-3.1035E-002,-3.0642E-002
       .float   2.8205E-002, 1.7465E-002,-2.4870E-002,-8.7185E-003, 2.1222E-002
       .float   2.7529E-003,-1.7460E-002, 1.2276E-003, 1.3778E-002,-3.6979E-003
       .float  -1.0343E-002, 4.9986E-003, 7.2940E-003,-5.4097E-003,-4.7252E-003
       .float   5.1775E-003, 2.6882E-003,-4.5219E-003,-1.1901E-003
H0     .float   3.6353E-003

LENGTH .set     55                  ;LENGTH = 55
```

5(a). Coefficient file for the FIR lowpass filter.

```
;NOISE2.COF - FIR COMB FILTER COEFFICIENTS
       .data                        ;coefficients section
COEFF  .float   0.2000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
       .float   0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
H0     .float   1.0000E+000

LENGTH .set     11                  ;LENGTH = 11
```

5(b). Coefficient file for the FIR comb filter.

```
;NOISE3.COF - FIR AVERAGING FILTER COEFFICIENTS
       .data                        ;coefficients section
COEFF  .float   0.2000E-001, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
       .float   0.0000E-000, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
       .float   0.2000E-001, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
```

Fig.5. Coefficient files for the three filters.

```
        .float   0.0000E-000, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
        .float   0.2000E-001, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
        .float   0.0000E-000, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
        .float   0.2000E-001, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
        .float   0.0000E-000, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
        .float   0.2000E-001, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
        .float   0.0000E-000, 0.0000E+000, 0.0000E+000, 0.0000E+000, 0.0000E+000
H0      .float   0.2000E-001

LENGTH .set      51                ;LENGTH = 51
```

5(c). Coefficient file for the FIR averaging filter.

Fig.5. Coefficient files for the three filters (cont'd).

From the MATLAB program NOISE1.M, the assembly language program NOISE1.ASM is assembled, loaded into and run on the C31 on board the DSK. While the filter is working, the MATLAB program NOISE1.M plots the frequency response of the filter and displays the response on the PC monitor.

4. Test Results

Tests were performed on the three filters using 1000 Hz sinusoidal, square, and triangular input signals corrupted with white noise. The source of the white noise is from a SR770 Spectrum Analyzer. The input and output waveforms are displayed using a HP54603 oscilloscope. The 55^{th} order FIR low pass filter is used for the sinusoidal noisy signal and the input/output waveforms are shown in Fig. 6. The comb filter and the averaging filter are used for the noisy square and triangular signals respectively and their input/output waveforms are shown in Fig. 7 and Fig. 8.It can be seen that actual system outputs agreed with theoretical results in terms of frequency responses. Rise time and overshoot are observed in the outputs due to the D/A process of the filter outputs.

5. CONCLUSION AND FUTURE WORK

Digital FIR lowpass, comb, and averaging filters are successfully implemented using the C31 DSK and tested on a sinusoidal, a square, and a triangular signal corrupted with white noise. Actual system outputs also agreed with theoretical results in terms of frequency responses. Future work includes adding the adaptation capability to the comb filter to achieve better noise reduction results.

Acknowledgement

This paper was supported in part by the National Science Foundation under Grant DUE-9950306.

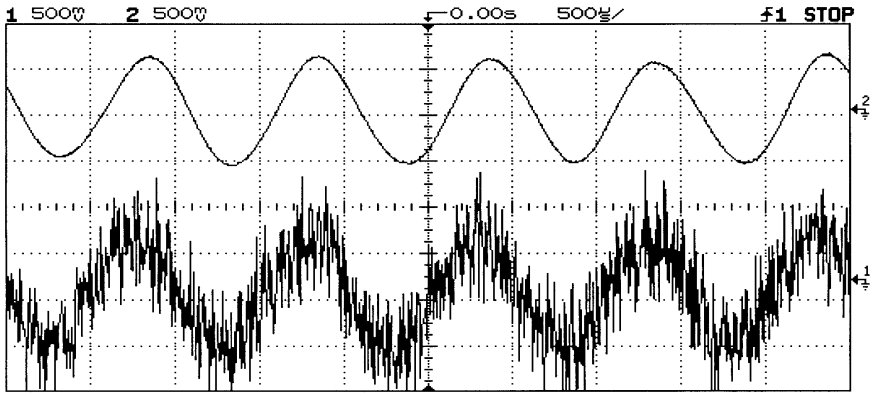

Fig.6. Input and output of the FIR lowpass filter

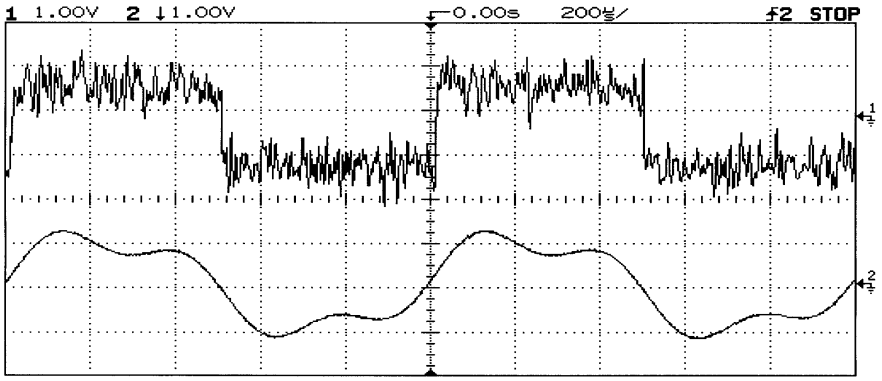

Fig.7. Input and output of the FIR comb filter

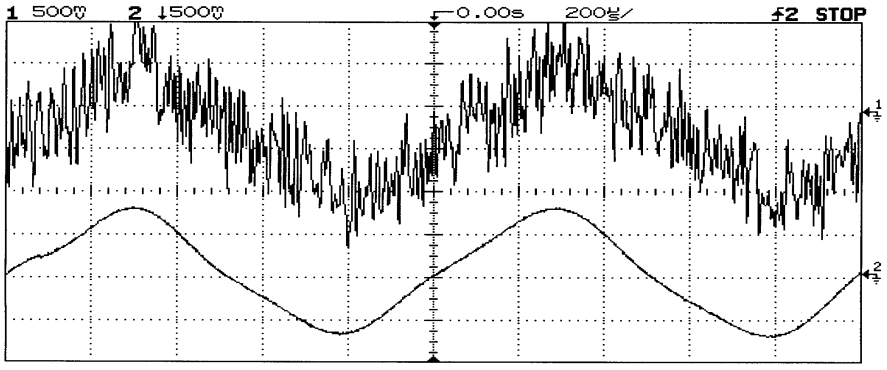

Fig.8. Input and output of the FIR averaging filter

Bibliography
1. Orfanidis, S. *Introduction to Signal Processing,* Prentice-Hall, 1996.
2.Chassaing, R. *Digital Signal Processing, Laboratory Experiments Using C and the TMS320C31 DSK,* Wiley, 1999.
3. Ziemer, R., Tranter, W., & Fannin, D., *Signals and Systems Continuous and Discrete,* Prentice-Hall, 1998.
4. Etter, D., *Engineering Problem Solving with MATLAB,* Prentice-Hall, 1997.

JIANXIN TANG
Jianxin Tang is an Associate Professor of Electrical Engineering at Alfred University. Dr. Tang received a B.S. degree in Electrical Engineering from Guangxi University, China, in 1976, a M.S. in Electrical Engineering from the University of Bridgeport, CT, in 1984, and a Ph.D. from the Department of Electrical and Systems Engineering at the University of Connecticut in 1989.