IIR Half-band Filter Design with TMS320VC33 DSP

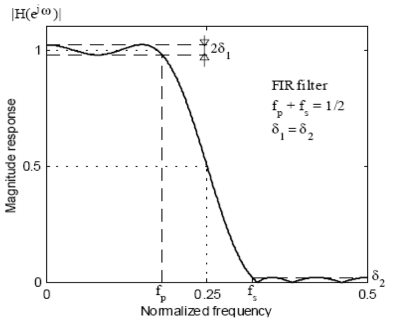
Ottó Nyári, Tibor Szakáll, Péter Odry

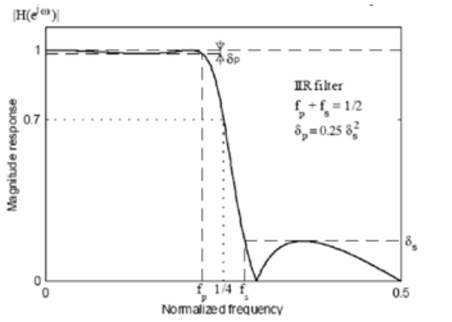
Polytechnical Engineering College, Marka Oreskovica 16, Subotica, Serbia and Montenegro
nyario@vts.su.ac.yu, tibi@vts.su.ac.yu, odry@vts.su.ac.yu

Abstract: A half-band IIR filter can have fewer multipliers than the FIR filter for the same sharp cut-off specification. Elliptic IIR filters are the most efficient. To overcome phase nonlinearity one can use optimization to design an IIR filter with an approximately linear phase response [1] or apply the double filtering technique with the Powell and Chau modification for real-time processing [2], [3] and [4].

1 Definition of Half-band Filter

A half-band filter is usually specified by the normalized passband edge frequency f_p, the stopband edge frequency f_s, the maximum passband magnitude ripple δ_1 and the maximum stopband magnitude ripple δ_2, as shown on figure 1.





Alternatively, the filter can be specified by the passband magnitude tolerance δ_p and the stopband magnitude tolerance δ_s (figure 2) or by the maximum passband attenuation a_p in dB and the minimum stopband attenuation a_s in dB.

A half-band filter must fulfil three conditions:

- the frequency symmetry condition,
- the magnitude condition,

- the filter order condition.

An elliptic half-band IIR filter is usually specified by the normalized passband edge frequency f_p, the stopband edge frequency f_s, the maximum passband magnitude tolerance δ_p and the maximum stopband magnitude tolerance δ_s, as shown on figure 2.

We can choose arbitrarily $n \geq n_{min}$ and only one filter parameter; say f_p, from the design space, that is, from the range $f_{p,min} \leq f_p \leq f_{p,max}$ [2]. We compute the other three parameters from the above three conditions for the elliptic half-band IIR filter. Figure 3 displays the minimum stopband attenuation versus stopband edge frequency for FIR and IIR half-band filters. The transfer function poles of a half-band IIR filter are placed on the imaginary axis of the z plane. If the filter order n is an odd number, the halfband filter can be implemented as a parallel connection of two allpass networks thus coinciding with the lattice wave digital filters [2], [3] and [4]. The allpass structure leads to the most efficient implementations that are possible.

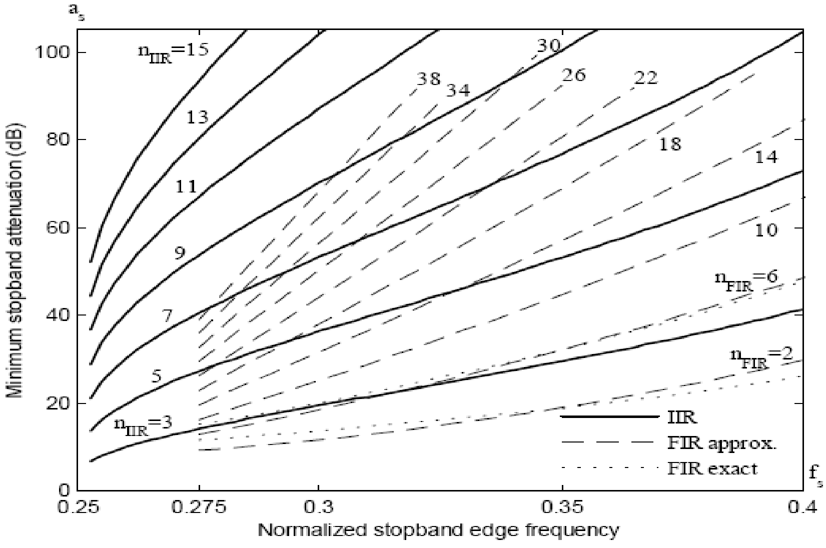


Figure 3
Design curves for IIR and FIR half-band filters

2 Half-band IIR Filter Design Using MATLAB

The application of MATLAB in IIR filter design requires four filter parameters to be specified. The design procedure which, starting from the specifications provides the exact adjustment of all half-band IIR filter parameters:

$$f_{sHB}=\min(f_s,\ 1/2-f_p) \tag{1}$$

$$f_{pHB}=1/2-\ f_{sHB} \tag{2}$$

$$a_{sHB}=\max\ (-10\ \log_{10}(1-\ 10^{-a_p/10},\ a_s) \tag{3}$$

$$a_{pHB}=10\ \log_{10}(1+\frac{1}{10^{a_{sHB}/10}-1}) \tag{4}$$

For the given filter order n and one of the parameters, the remaining parameters are computed using explicit expressions. The design starts from a specification S = {f_p, f_s, a_p, a_s}. We can find the corresponding half-band filter specification satisfying the first two conditions, S_{HB} = {f_{pHB}, f_{sHB}, a_{pHB}, a_{sHB}}. We can use MATLAB for determining the minimal filter order: n=ellipord (2*f_{pHB}, 2*f_{sHB}, a_{pHB}, a_{sHB}), but we have to use a half-band filter specification. The filter parameters (f_p, f_s, a_p, a_s) no longer meet exactly the calculated specification for the halfband IIR filter (f_{pHB}, f_{sHB}, a_{pHB}, a_{sHB}).

One of the filter parameters (f_p, f_s, a_p, a_s) needs to be chosen, and the remaining three parameters are computed from the three conditions for half-band IIR filters. Next, we can use the function "ellipin" MATLAB to determine the transfer function of the n-th-order half-band IIR filter. We can determine the minimum and the maximum passband edge frequencies and the range of all other three parameters for three chosen passband edge frequencies:

$$f_p = f_{p,\ min} \quad ; \quad f_p = f_{pHB} \quad ; \quad f_p = f_{p,max} \tag{5}$$

Using the function "ellip" in MATLAB we can compute poles p_k of the filter transfer function H(z). Notice that poles are placed on the imaginary axis of the z plane. The most efficient realization of H(z) is a parallel connection of two allpass branches. Thus we present H(z) as a sum of two allpass functions:

$$H(z)=\frac{\beta_2+z^2}{1+\beta_2 z^{-2}}\ \frac{\beta_4+z^{-2}}{1+\beta_4 z^{-2}}+z^{-1}\ \frac{\beta_3+z^{-2}}{1+\beta_3 z^{-2}}\ \frac{\beta_5+z^{-2}}{1+\beta_5 z^{-2}} \tag{6}$$

Where the β_i are the squared pole magnitudes.

We can use f_p to optimize some filter property, for example, to minimize error due to the coefficient quantization. If we choose f_p = $f_{p,min}$, we reach the widest permitted transition band and get the smallest pass-band ripple and the highest stopband attenuation that is possible. For the choice f_p = $f_{p,max}$, we get the narrowest possible transition band, and we reach the highest pass-band ripple and the smallest stopband attenuation. Illustrated on figure 5. We use the function “ellipj” in MATLAB to determine the Jacobi’s elliptic s_n function: x_i=ellipj (((2*i-1)/n+1)*ellipke (1/ε^2), 1/ε^2), and we use the function “ellipke” to determine the complete elliptic integral.

Figure 6 presents the frequency response of the half-band FIR and IIR filter. The half-band IIR filter has fewer multipliers than the FIR filter and the half-band IIR filter also has a better magnitude characteristic.

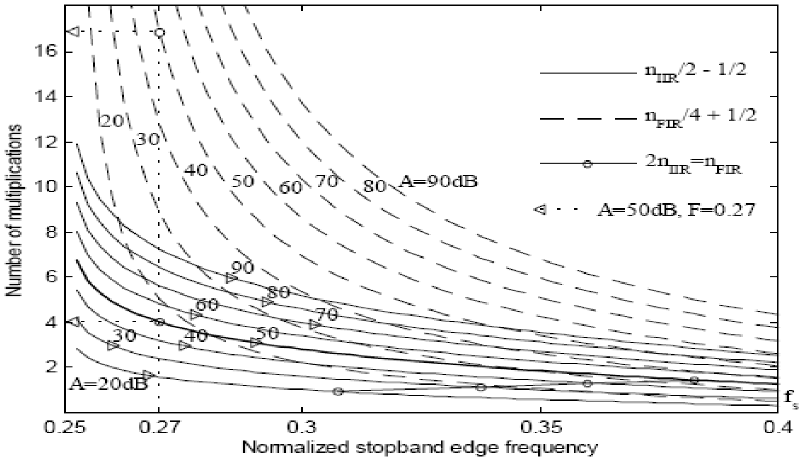


Figure 5
Plot of number of multiplications as a function of normalized stopband edge frequency and minimum stopband attenuation

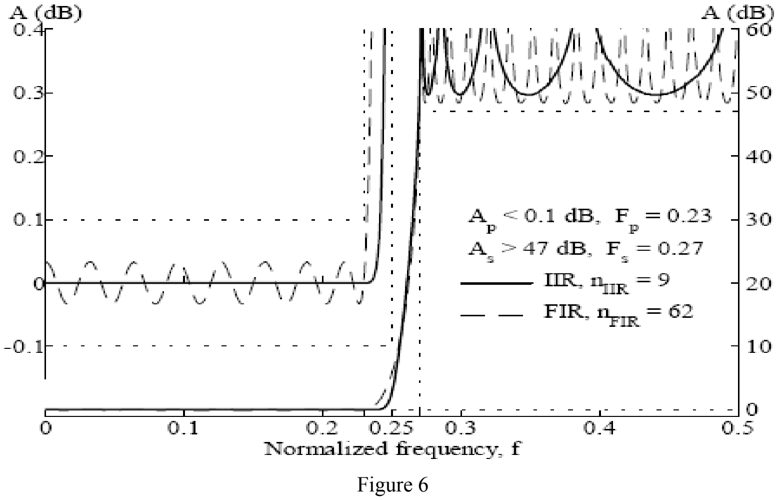


Figure 6
Attenuation of the example filters: 9th-order half-band IIR filter, sold line, and 62nd-order half-band FIR filters, dashed line

3 IIR Filter Realization with TMS320VC33 DSP

The TMS320VC33-150 is a high-performance floating-point digital signal processor (DSP), with 13-ns instruction cycle time, 150 million floating-point operations per second (MFLOPS) and 75 million instructions per second (MIPS).

General-purpose applications are greatly enhanced by the large address space, multiprocessor interface, internally and externally generated wait states, one external interface port, two timers, one serial port, and multiple-interrupt structure. The TMS320C3x supports a wide variety of system applications from host processor to dedicated coprocessor. High-level-language support is easily implemented through a register-based architecture, large address space, powerful addressing modes, flexible instruction set, and well-supported floating-point arithmetic.

In the more general case, the IIR filter contains N>1 biquads, as well IIR half-band filters. The equations for its implementation are given by the following pseudo-C language code:

```
y [0,n] = x [n]
for (i = 0; i < N; i ++){
   d [i,n]=a2 [i] d [i, n – 2] + a1 [i] d [i,n –1] + y [i – 1,n]
   y [i,n]=b2[i] d [i – 2] + b1 [i] d [i,n – 1] + b0 [i] d [i,n]
}
y [n] = y [N – 1,n]
```

First - in MATLAB – we have calculated the IIR half-band filter :

[b,a] = halfbandiir ('minorder',F_p,Dev) designs the minimum order IIR filter,

[b,a] = halfbandiir (N, F_p) designs a lowpass N-th order halfband IIR filter,

[b,a] = halfbandiir (N, F_p,’high’) designs a highpass N-th order halfband IIR filter.

Then we have generated biquad coefficients with MATLAB command “tf2sos”. After then we can use one or more “include” commands in the assembly-language code for embedding the section of IIR half-band coefficients. It is also possible to use multiple input channels (e.g. left, right) in the TMS320VC33 DSK. Each cannel we can separate to two output channels using a lowpass and a highpass filter for each channel.

3.1 Fragments of ’C3x Assembly-language Code

```
* TITLE IIR FILTERS (N > 1 BIQUADS)
* ARGUMENT ASSIGNMENT:
* R2  | INPUT SAMPLE x(n)
* AR0 | ADDRESS OF FILTER COEFFICIENTS (a2(0))
* AR1 | ADDRESS OF DELAY NODE VALUES (d(0,n+-2))
* RC  | NUMBER OF BIQUADS (N) +-2
* REGISTERS CONTAINING RESULT: R0
    ...
        .include "half band iir biquads coefficients.asm"
    ...

        NOP  *--AR0(1)                               ; prepare for looping
        NOP  *--AR1(IR0)                             ; predecrement by ovks
        RPTB LOOP                                    ; Loop for 1 <= i < n
        MPYF3 *++AR0(1),*++AR1(IR0), R0              ; a2(0) * d(0,n 2)  > R0
        MPYF3 *++AR0(1), *AR1--(1)%, R1              ; b2(0) * d(0,n 2)  > R1
        MPYF3 *++AR0(1),*AR1,R0                      ; a1(0) * d(0,n 1)  > R0
    || ADDF3 R0, R2, R2                              ; First sum term of d(0,n)
        MPYF3 *++AR0(1),*AR1--(1)%,R0                ; b1(0) * d(0,n 1)  > R0
    || ADDF3 R0, R2, R2                              ; Second sum term of d(0,n)
        MPYF3 *++AR0(1),R2,R2                        ; b0(0) * d(0,n)  > R2
    || STF R2, *AR1--(1)%                            ; store d(0,n)
        ADDF  R0,R2                            ; First sum term of y(n 1,n)
LOOP    ADDF3 R1,R2,R2                         ; Second sum term,  y(0) -> R2
        LDF  R2,R0                                   ; R2 == Y -> R0
        NOP *AR1--(IR1)                              ; Return to first biquad
        NOP *AR1--(1)%                               ; Point to d(0,n 1)
        STI  AR1,@IIR_RDELAY
        FIX   R0,R0                                  ; float -> int
        STI   R0,@ADCo_R                             ; store to memory, D/A
    ...
```

3.2 Demonstration of Practical Results

First example (figure 7) is for 7th order half-band IIR filter.

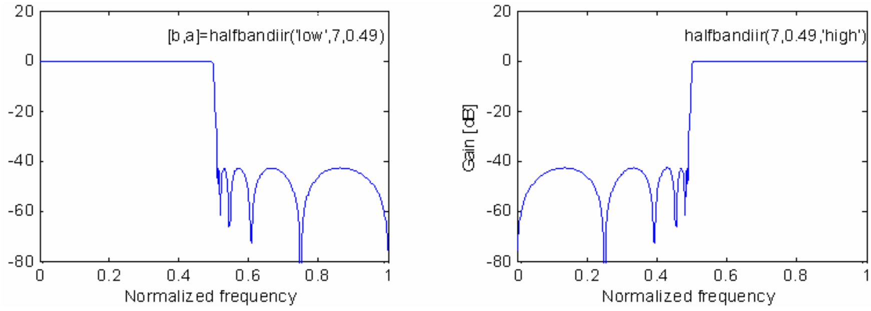


Figure 7

left: lowpass filter frequency response; right: highpass filter frequency response for 7th order IIR

When the filter order had raised from 7th order into 23rd order, the frequency response has become far better. Second example: figure 8.

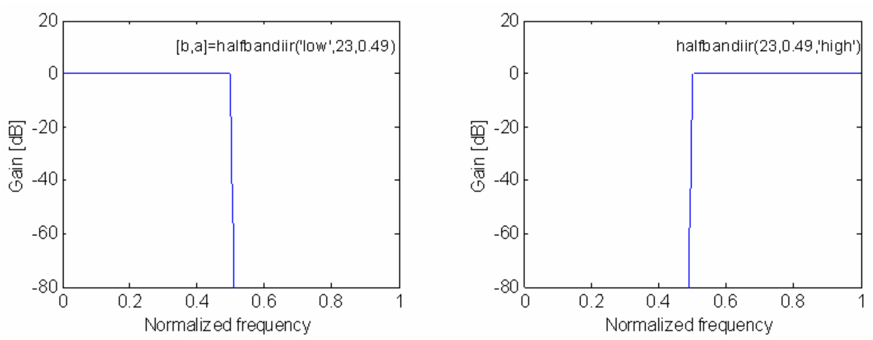


Figure 8

left: lowpass filter frequency response; right: highpass filter frequency response for 23rd order IIR

Conclusions

Multirate techniques are used to improve a digital filter design and implementation. The multirate approach increases the computation speed, decreases the overall filter order, and decreases power consumption. Various multirate techniques can be used to reach the filter requirements regarding the magnitude and phase responses, sensitivity to coefficient accuracy, robustness, noise and overflow, computation speed, miniaturization and modularity of the realization structure or power consumption. Half-band filters are basic building blocks in multirate systems.

References

[1] Surma-aho, K. and Saramäki, T. (1999, July). A systematic technique for designing approximately linear phase recursive digital filters. IEEE Transactions on Circuits and Systems–II, 46, 956-63

[2] Lutovac, M. D. and Milic, L. D. (2000, March). Approximate linear-phase multiplierless IIR halfband filter. IEEE Signal Processing Letters, 7, 52-3

[3] Lutovac, M. D. and Milic, L. D. (2000, November). Half-band IIR filter design using MATLAB. Conference TELFOR, Yugoslavia, 329-32

[4] Powell, S. and Chau, M. (1991, November). A technique for realizing linear phase IIR filters. IEEE Trans. Signal Process., 39, 2425-35

[5] TI-Books for TMS320C3x DSP