XVII IMEKO World Congress
Metrology in the 3rd Millenium
June 22–27, 2003, Dubrovnik, Croatia

IMPLEMENTATION OF 1/1 OCTAVE AND 1/3 OCTAVE FILTERS IN DIGITAL SIGNAL PROCESSOR

Janusz M. Mosakowski; Andrzej Podgórski

Svantek Ltd; Warsaw Technical University, Faculty of Electronics and Information Technology; Warsaw, Poland

Abstract – Set of filters is required for the analysis of sound and vibration signals usually composed of different elements. The designing process of so-called 1/1 octave and 1/3 octave digital filters and their implementation in the digital signal processor, working in the real time system, is shortly presented. The obtained exemplary frequency characteristics are given.

Keywords: analysis of sound signals, 1/1 & 1/3 octave digital filters, digital signal processing.

1. INTRODUCTION

Information about the spectral distribution of the acoustic or vibration signal is required in many different fields of science, technology, law, art, etc. The existing signal types are composed of elements with different shapes, amplitudes, periods, frequency characteristics etc.

The standards are required in order to analyse these signals with different instrumentation and to obtain the comparable and univocal results. The sets of middle passing 1/1 octave and 1/3 octave filters covering the whole required band are used for such analysis in practice. The parameters of these filters, i.e. their number, the centre frequency, the pass band, attenuation in the stop band, ripple in the pass band, slope of the frequency characteristics, etc. are defined in the ANSI standard S1.11-1986 [1].

The extended set of 1/1 octave filters starts from the centre frequency equal to 16 kHz. The centre frequency of each consecutive 1/1 octave filter is equal to half of the centre frequency of the previous one: 8 kHz, 4 kHz, 2 kHz etc. down to the single Hz. In the case of 1/3 octave filters three of them constitute one octave. For the upper octave the filters with the following centre frequencies are taken into account: 20 kHz, 16 kHz and 12,5 kHz. The centre frequencies of the consecutive filters are calculated in the same way as described above for 1/1 octave filters.

2. DESIGNING OF 1/1 & 1/3 OCTAVE FILTERS

The aim of the presented work was to design the set of 1/1 & 1/3 octave filters and to implement them in the digital signal processor DSP56002 of Motorola [2]. It is 24-bits, fixed point processor with up to 80 MHz clock. The additional requirement was the performance of the filtration in the real time. It means that all samples of the measured or analysed signal with 48 kHz of sampling frequency should be taken into account. To fulfil this requirement such structure of the filters was searched which could be effectively implemented in the digital processor. Finally, the designed filters are composed of the second order sections described by the equation:

$$H(z)=\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{a_0+a_1z^{-1}+a_2z^{-2}} \qquad (1)$$

Each descending slope (so-called High Pass) of 1/1 or 1/3 octave filter and growing slope (so-called Low Pass) of 1/1 octave filter of the frequency characteristics consists of three sections. In the case of 1/3 octave filters for each octave only one filter for the increasing (growing) slope is developed but composed of four sections. In order to fulfil the requirements of the real time processing the digital processor in which 1/1 octave filtering is implemented works with 66 MHz clock and with 80 MHz – in the case of 1/3 octave filtering. Almost all digital filters are performed using the same procedure. The reduction of the frequency of the input signal is performed by the decimation filter which selects the proper samples given on the input of the filtration procedure. The performed tests proved that 1/1 octave filters with the centre frequency equal to 8 kHz and 16 kHz have to be realised differently.

3. EXEMPLARY OF 1/1 OCTAVE FILTERS DESIGN

The coefficients of the filters were searched and normalised using the scripts written in the environment of Matlab. The exemplary function for increasing slopes for 1/1 octave filters is presented below. This function contains calls for few functions from Signal Processing Toolbox of Matlab as well as some auxiliary written by the authors.

```
    function oktsvlp(n,rg,r1,r2,f0,fp);
% Low Pass filter for the set of 1/1 octave filters
%
% format:
%   oktsvlp(n,r1,f0,fp)
% where:
%   n  - order of the filter
%   rg - attenuation on the limit frequency
%   r1 - ripple in the pass band (must be lower than rg)
%   r2 - attenuation in the stop band
%   f0 - centre frequency of 1/1 octave filter
%   fp - sampling frequency
```

```
% exemplary call:
%oktsvlp(6,3.0,0.001,100,8000,48000)                % type = 1
%---------------------------------------------------------
rt2=sqrt(2);                   % setting of the first limit frequency
fg2=f0*rt2;    fp2=fp/2;
% setting of the amplification on the limit frequency
a3=-rg;
% loop for searching the proper filter
e=1.000001;    % related accuracy of the searched limit frequency
type=1;          % 1-Tshebyshev I, 2-Tshebyshev II, 3-elliptic
%---------------------------------------------------------
f2=fg2;    koniec1=0;
dfg=-fp2/50;                   % beginning frequency step
koniec2=0;
while koniec2==0,            % loop for f2 frequency change
  f2=f2+dfg;
  if type==1
    [a,b]=cheby1(n,r1,f2/fp2);
  elseif type==2
    [a,b]=cheby2(n,r2,f2/fp2);
  else
    [a,b]=ellip(n,r1,r2,f2/fp2);
  end;
  [z,p,k]=tf2zp(a,b);  sos=zp2sos(z,p,k);  sos=testdcsos(sos);
  % searching for 3 dB frequency
  fx=findf(a3,0,fp2,fp,sos);
  if ((fx>fg2) & (dfg>0)) | ((fx<fg2) & (dfg<0))
    dfg=-dfg/2;
  end;
  dfxfg=fx/fg2;
  if ((1/e)<dfxfg) & (dfxfg<e)
    koniec2=1;
  end;
end;
if (n>1) & (1==1)
  for i=1:(n+1)/2,
    if sos(i,3)==0
      sos(i,2)=sos(i,1);
    else
      sos(i,3)=sos(i,1);   sos(i,2)=2*sos(i,1);
    end;
  end;
end;
sos=testsos(sos)
[a,b]=sos2tf(sos);
%---------------------------------------------------------
fo=f0;             %/1.26
f_G_1_2=hz(f2z(fo/rt2,fp),sos)
f_G1=hz(f2z(fo*1.0000,fp),sos)
f_G1_4=hz(f2z(fo*1.1892,fp),sos)
f_G3_8=hz(f2z(fo*1.2968,fp),sos)
f_T1_055=hz(f2z(fo*1.25992105*1.05594,fp),sos)
f_T1_087=hz(f2z(fo*1.25992105*1.08776,fp),sos)
f_G1_2=hz(f2z(fo*rt2,fp),sos)
f_T1_295=hz(f2z(fo*1.25992105*1.29565,fp),sos)
f_G1=hz(f2z(fo*2.0,fp),sos)
f_G2=hz(f2z(fo*4.0,fp),sos)
%f_G3=hz(f2z(fo*8.0,fp),sos)
%f_G4=hz(f2z(fo*16.0,fp),sos)
fc=(fp/2)-(f0/rt2)
f_c = hz(f2z(fc,fp),sos)
if 1==0
% searching for -3 dB frequency
f1_3=findf(-3.0,0,f1,fp,sos)/f0
f2_3=findf(-3.0,f2,fp/2,fp,sos)/f0
% searching for -18.0 dB frequency
f1_18=findf(-18.0,0,f1,fp,sos)/f0
f2_18=findf(-18.0,f2,fp/2,fp,sos)/f0
% searching for -42.5 dB frequency
f1_42=findf(-42.5,0,f1,fp,sos)/f0
f2_42=findf(-42.5,f2,fp/2,fp,sos)/f0
% searching for -62.0 dB frequency
f1_62=findf(-62.0,0,f1,fp,sos)/f0
f2_62=findf(-62.0,f2,fp/2,fp,sos)/f0
end;
%---------------------------------------------------------
if 1==1
   sos=mbsos(sos);               % saving the coefficients to a file
 if n==1
   save oktsvlp8.w sos -ascii -double
 else
   if fp==51200
     save oktsvlp.w3 sos -ascii -double
   else
     save oktsvlp.w sos -ascii -double
   end;
 end;
 sos=mbsos(sos);
end;
%---------------------------------------------------------
nfft=2048;
[h,w]=freqz(a,b,nfft,fp);
h1=20*log10(abs(h));
dm=-120;  hmin=0;  ihmin=nfft;
for i=nfft:-1:1,
  if h1(i)<dm
    h1(i)=dm;
  end;
  if h1(i)<=hmin
    hmin=h1(i);  ihmin=i;
  end;
end;
w1=w;           %/pi;
%wz1=1+f1*(N-1);
%wz2=1+f2*(N-1);

t1=1;           %t2=round(nfft*f80/fp2);
figure(2);      %plot(w1(t1:t2),h1(t1:t2));
plot(w1,h1);
grid;
return
%---------------------------------------------------------
```

The similar scripts were written for descending slopes for 1/1 octave filters. Using these scripts the coefficients for all sections of the digital 1/1 octave filters were calculated. Below, the coefficients of the increasing and descending slope of 1/1 octave filters are given (without the low pass filter for 8 kHz centre frequency and the high pass filter for 16 kHz centre frequency):

```
a0_olp1    equ    0.036147742233380460
a1_olp1    equ    0.072295480851986697
a2_olp1    equ    0.036147742233380460
b0_olp1    equ    1.000000000000000000
b1_olp1    equ    1.290972130587618000
b2_olp1    equ   -0.435409957138915600

a0_olp2    equ    0.069493900482308160
a1_olp2    equ    0.138987794015226272
a2_olp2    equ    0.069493900482308160
b0_olp2    equ    1.000000000000000000
b1_olp2    equ    1.286714166639202000
b2_olp2    equ   -0.567337245336483000
```

```
a0_olp3    equ   0.115479343062797900
a1_olp3    equ   0.230958674577661494
a2_olp3    equ   0.115479343062797900
b0_olp3    equ   1.000000000000000000
b1_olp3    equ   1.365051468638126000
b2_olp3    equ  -0.823096117319185200

a0_ohp1    equ   0.383769035090592800
a1_ohp1    equ  -0.767538070181185600
a2_ohp1    equ   0.383769035090592800
b0_ohp1    equ   1.000000000000000000
b1_ohp1    equ   0.438856991062667800
b2_ohp1    equ  -0.0962154133597fi68460

a0_ohp2    equ   0.588722005354193400
a1_ohp2    equ  -1.177444010708386800
a2_ohp2    equ   0.588722005354193400
b0_ohp2    equ   1.000000000000000000
b1_ohp2    equ   0.907594527642175800
b2_ohp2    equ  -0.452034053465067300

a0_ohp3    equ   0.785448656933730300
a1_ohp3    equ  -1.570897313867460600
a2_ohp3    equ   0.785448656933730300
b0_ohp3    equ   1.000000000000000000
b1_ohp3    equ   1.321709834519823000
b2_ohp3    equ  -0.817392416418538100
```

In the similar form the remaining coefficients for the mentioned above filters (the low pass filter for 8 kHz centre frequency and the high pass filter for 16 kHz centre frequency) could be presented.

4. REALISATION OF 1/1 OCTAVE FILTERS IN DSP

The procedures written in the assembler language of the DSP56002 processor which realise 1/1 octave filters are given below:

```
;-------------------------------------------------------------------------------
; Procedure of one section of 1/1 octave filter
; input:
;     A   - input sample
;     R1 - address of the coefficients in X memory:
;     R5 - address of the section in Y memory:
;     R3 - address of the result record in X memory:
; output:
;     R3 – incremented by 4
;-------------------------------------------------------------------------------
FiltrY
 ;1/1 Octave filter           A = input sample
 asr   a        x:(r1)+,x1  y:(r5)+,y0  ;+a1*w(t-1)   x1:=b2   w(t-2)
 mpy   x1,y0,b  x:(r1)+,x1  a,y1        ;b2*w(t-2)    x1:=a0   x(t)
 mac   x1,y1,b  x:(r1)+,x1  y:(r5)-,y1  ;+a0*x(t)     x1:=b1   w(t-1)
 macr  x1,y1,b  x:(r1)+,x1  y1,y:(r5)+  ;+b1*w(t-1)   x1:=a2   w(t-1)
 mac   x1,y0,b  x:(r1)+,x1  b,y:(r5)+   ;+a2*w(t-2)   x1:=a1   w(t)

 macr  x1,y1,b  x:(r1)+,x1  y:(r5)+,y0  ;+a1*w(t-1)   x1:=b2   w(t-2)
 mpy   x1,y0,b  x:(r1)+,x1  b,y1        ;b2*w(t-2)    x1:=a0   x(t)
 mac   x1,y1,b  x:(r1)+,x1  y:(r5)-,y1  ;+a0*x(t)     x1:=b1   w(t-1)
 macr  x1,y1,b  x:(r1)+,x1  y1,y:(r5)+  ;+b1*w(t-1)   x1:=a2   w(t-1)
 mac   x1,y0,b  x:(r1)+,x1  b,y:(r5)+   ;+a2*w(t-2)   x1:=a1   w(t)

 macr  x1,y1,b  x:(r1)+,x1  y:(r5)+,y0  ;+a1*w(t-1)   x1:=b2   w(t-2)
 mpy   x1,y0,b  x:(r1)+,x1  b,y1        ;b2*w(t-2)    x1:=a0   x(t)
 mac   x1,y1,b  x:(r1)+,x1  y:(r5)-,y1  ;+a0*x(t)     x1:=b1   w(t-1)
 macr  x1,y1,b  x:(r1)+,x1  y1,y:(r5)+  ;+b1*w(t-1)   x1:=a2   w(t-1)
 mac   x1,y0,b  x:(r1)+,x1  b,y:(r5)+   ;+a2*w(t-2)   x1:=a1   w(t)

 macr  x1,y1,b              a,y1        ;+a1*w(t-1),           sample
 ;B  = output sample
 ;RMS value calculation and increase of the sample's counter
 move           x:(r3)+,a   b,y0        ;A:=counter, Y0:=sample
 mpy   y0,y0,b  x:(r3)+,x0
 not   a        x:(r3)+,x1              ;counter's increase
 add   x,b      x:(r3)-,x0
 neg   a        (r3)-                   ;counter's increase
 move           (r3)-
 clr   a        a1,x:(r3)+              ;save increased counter
 adc   x,a      b0,x:(r3)+              ;save RMS_Low
 move           b1,x:(r3)+              ;save RMS_Mid
 move           a0,x:(r3)+              ;save RMS_High
 ;Low Pass filter             Y1 = input sample
 move           x:(r1)+,x1  y:(r5)+,y0  ;             x1:=b2   w(t-2)
 mpy   x1,y0,a  x:(r1)+,x1              ;b2*w(t-2)    x1:=a0
 mac   x1,y1,a  x:(r1)+,x1  y:(r5)-,y1  ;+a0*x(t)     x1:=b1   w(t-1)
 macr  x1,y1,a  x:(r1)+,x1  y1,y:(r5)+  ;+b1*w(t-1)   x1:=a2   w(t-1)
 mac   x1,y0,a  x:(r1)+,x1  a,y:(r5)+   ;+a2*w(t-2)   x1:=a1   w(t)

 macr  x1,y1,a  x:(r1)+,x1  y:(r5)+,y0  ;+a1*w(t-1)   x1:=b2   w(t-2)
 mpy   x1,y0,a  x:(r1)+,x1  a,y1        ;b2*w(t-2)    x1:=a0   x(t)
 mac   x1,y1,a  x:(r1)+,x1  y:(r5)-,y1  ;+a0*x(t)     x1:=b1   w(t-1)
 macr  x1,y1,a  x:(r1)+,x1  y1,y:(r5)+  ;+b1*w(t-1)   x1:=a2   w(t-1)
 mac   x1,y0,a  x:(r1)+,x1  a,y:(r5)+   ;+a2*w(t-2)   x1:=a1   w(t)

 macr  x1,y1,a  x:(r1)+,x1  y:(r5)+,y0  ;+a1*w(t-1)   x1:=b2   w(t-2)
 mpy   x1,y0,a  x:(r1)+,x1  a,y1        ;b2*w(t-2)    x1:=a0   x(t)
 mac   x1,y1,a  x:(r1)+,x1  y:(r5)-,y1  ;+a0*x(t)     x1:=b1   w(t-1)
 macr  x1,y1,a  x:(r1)+,x1  y1,y:(r5)+  ;+b1*w(t-1)   x1:=a2   w(t-1)
 mac   x1,y0,a  x:(r1)+,x1  a,y:(r5)+   ;+a2*w(t-2)   x1:=a1   w(t)

 macr  x1,y1,a                          ;+a1*w(t-1)
 ;A = output sample
 rts
;-------------------------------------------------------------------------------
```

The coefficients taken for the above calculations are presented below. The coefficients for the descending slope of 16 kHz filter and the increasing slope of 8 kHz filter are not included.

```
;-------------------------------------------------------------------------------
; Coefficients of 1/1octave filters
             XDEF      LiczbaWspOk
P_WspOk      equ       *
             ;High Pass Filter (1/1 octave)
             include   'ohp1.wsp'
             include   'ohp2.wsp'
             include   'ohp3.wsp'
             ;first section
             dc        0.5*b2_ohp1
             dc        0.5*a0_ohp1
             dc        0.5*b1_ohp1
             dc        0.5*a2_ohp1/a0_ohp1
             dc        0.5*a1_ohp1/a0_ohp1
             ;second section
             dc        0.5*b2_ohp2
             dc        0.5*a0_ohp2
             dc        0.5*b1_ohp2
             dc        0.5*a2_ohp2/a0_ohp2
             dc        0.5*a1_ohp2/a0_ohp2
             ;third section
             dc        0.5*b2_ohp3
             dc        0.5*a0_ohp3
             dc        0.5*b1_ohp3
             dc        0.5*a2_ohp3/a0_ohp3
             dc        0.5*a1_ohp3/a0_ohp3
```

```
            ;Low Pass Filter
            include     'olp1.wsp'
            include     'olp2.wsp'
            include     'olp3.wsp'
            ;first section
            dc          0.5*b2_olp1
            dc          0.5*a0_olp1
            dc          0.5*b1_olp1
            dc          0.5*a2_olp1/a0_olp1
            dc          0.5*a1_olp1/a0_olp1
            ;second section
            dc          0.5*b2_olp2
            dc          0.5*a0_olp2
            dc          0.5*b1_olp2
            dc          0.5*a2_olp2/a0_olp2
            dc          0.5*a1_olp2/a0_olp2
            ;third section
            dc          0.5*b2_olp3
            dc          0.5*a0_olp3
            dc          0.5*b1_olp3
            dc          0.5*a2_olp3/a0_olp3
            dc          0.5*a1_olp3/a0_olp3
LiczbaWspO              equ     @CVS(N,*-P_WspOk)
;----------------------------------------------------------------
```

5. CHARACTERISTICS OF THE SELECTED FILTERS

The exemplary amplitude - frequency characteristics of the realised 1/1 octave filter is given in Fig. 1. The indication F denotes the current frequency, Fo – the centre frequency of any 1/1 octave filter. In the lower line of the frequency axis description the values in dB are given, which correspond to the crossing points of the characteristics with the net lines.

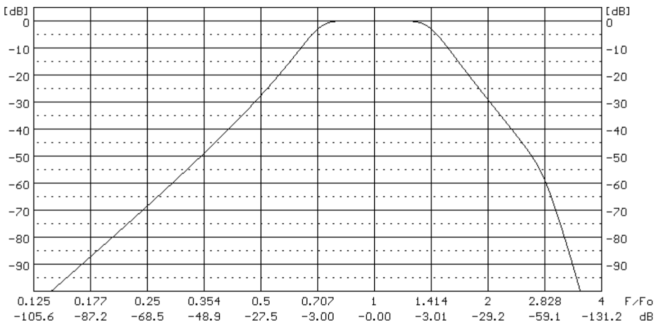

Fig. 1. Characteristics of the exemplary 1/1 octave filter

6. DESIGN, REALISATION AND CHARACTERISTICS OF 1/3 OCTAVE FILTERS

The coefficients of 1/3 octave filters were searched and normalised using the scripts written in the environment of Matlab similar to those presented in Chapter 3. The procedures written in the DSP assembler language realising 1/3 octave filters, were also implemented. In this case the DSP has to work with 80 MHz clock in order to fulfil the real time analysis requirement. The exemplary amplitude - frequency characteristics of the realised 1/3 octave filters, which constitute one octave, are given in Fig. 2 (the lower 1/3 octave filter), Fig. 3 (the middle 1/3 octave filter) and in Fig. 4 (the upper 1/3 octave filter). The description of the axis of these figures is the same as that presented for Fig. 1.

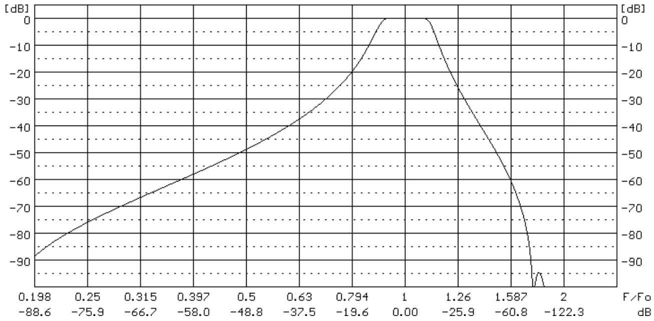

Fig. 2. Characteristics of the exemplary lower 1/3 octave filter

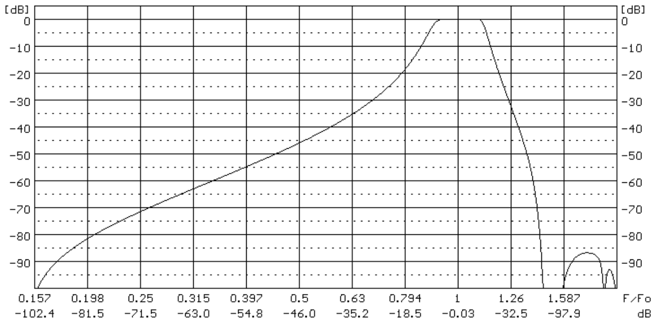

Fig. 3. Characteristics of the exemplary middle 1/3 octave filter

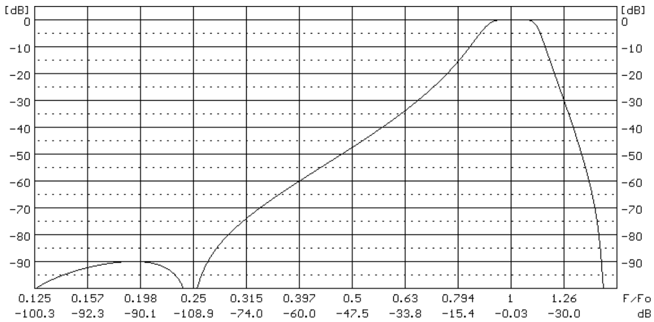

Fig. 4. Characteristics of the exemplary upper 1/3 octave filter

7. CONCLUSIONS

The aim of the work was achieved. 1/1 octave and 1/3 octave filters, conforming to ANSI S1.11-1986 standard and working in the real time system analysing the sound and vibration signals up to 22,4 kHz frequency band are implemented in the fixed-point DSP using the designing procedures described in the paper.

REFERENCES

[1] American Standard, "Specification for Octave-Band and Fractional-Octave-Band Analog and Digital Filters", ANSI S1.11-1986.

[2] Motorola "DSP56000/DSP56001 Digital Signal Processor User's Manual", Rev 1.

Authors: MSc., Janusz Maciej Mosakowski, Svantek Ltd, 01-410 Warsaw, ul. Sitnika 1/68; +4822 8710469 jmmos@svantek.com.pl
PhD., Andrzej Podgórski, Warsaw University of Technology, Faculty of Electronics and Information Technology, Institute of Radioelectronics, 00-665 Warsaw, ul. Nowowiejska 15/19, +48 501413586, A.Podgorski@ire.pw.edu.pl