

SECTION 11

FIR DIGITAL DESIGN SEGMENT

11.1 INTRODUCTION

Finite impulse response (FIR) digital filters can be designed by the S/FILSYN program through the use of either the well-known windowing method or the celebrated McClellan-Parks-Rabiner algorithm, which has been found unsurpassed in efficiency. This segment is accessible directly from the SMAIN (S) segment by selecting the digital option. If, in response to the next question as to whether we want FIR (F) or IIR (I) design, we select the I option, we get the recursive (infinite impulse response) design technique which is described in *Sections 9 and 10*. If, on the other hand we select the F option, we get the FIR design segment, which enables us to design, analyze and modify filters of up to 10 pass- and stop-bands, as well as differentiators and Hilbert transformers. If you are using the multi-executable version of S/FILSYN, the FIR program is a *completely stand-alone unit*.

The FIR segment of S/FILSYN contains two major sub-segments, which can be used individually or in combination. The first one has two parallel paths, one for the windowed design method and the other for the equal-ripple design of linear phase FIR filters. This latter is the McClellan-Parks-Rabiner program, converted to interactive form, with a few convenience features added.

The second sub-segment takes a set of FIR filter coefficients, which were obtained either from the design segment or directly entered into the program, and manipulates them in many ways. The major possibilities are:

- a) The transfer function may be factored into second order factors. This means that, if we are given the following impulse response coefficients \mathbf{h}_k , we can calculate the coefficients \mathbf{a}_i and \mathbf{b}_i in the identity:

$$H(z) = \sum_{k=0}^{N-1} h_k z^{-k} = h_0 \prod_{i=1}^{[N/2]} (1 + a_i z^{-1} + b_i z^{-2}) \quad (11.1)$$

As an option, the factored form if we know it, can also be entered into this segment directly.

- b) Any number of these factors can be remultiplied into up to five individual polynomial form impulse responses. This step is useful should we wish to break up a longer FIR filter into a cascade of two or more shorter ones. If we select the factors properly, the individual components can also be made into linear phase functions, if the original was of a linear phase design.

- c) Before remultiplying, we may flip all the zeros of the factored transfer function located outside the unit circle in the z-plane, to their image location inside it, thereby converting the linear phase function into a minimum phase one.

11.2

d) If we have started from the appropriate linear phase design and if the resulting minimum phase function has all double roots, we may take the 'square root' of the function. This will yield the optimal minimum phase FIR filter.

e) Some other operations available at any stage are:

- coefficient truncation or rounding,
- frequency domain analysis and plotting,
- scaling,
- temporary or permanent storage and recall of data.

11.2 LINEAR PHASE EQUAL RIPPLE DESIGN

The theory of the linear phase FIR filter design method has been described in the literature in detail. The process is an iterative one and converges to an equi-ripple optimal design which is shown schematically in Fig. 11.1, for a bandpass filter. Up to 10 frequency bands (passbands and stopbands) may be specified for filters. For differentiators and Hilbert transformers, a single band is automatically assumed. The bands are specified by their edge frequencies, the desired function value in the band, and a weight factor. If the desired function value is 1 (passband) or 0 (stopband), we may replace the weight factors by equivalent loss values. The relationship between weights and loss values will be discussed in *Section 11.3* below.

First we wish to establish the relationships between the δ_i ripple values of Fig. 11.1 and the passband loss ripple a_p or stopband minimum loss a_s . These are given in dB by the expressions:

Passband ripple:

$$a_p = 20 \log_{10} (1 + \delta_i)/(1 - \delta_i) \quad (11.2a)$$

Stopband suppression:

$$a_s = -20 \log_{10} \delta_i \quad (11.2b)$$

The filter length (number of taps N) is the most important design parameter and in most cases can be determined only by trial and error. For the special case of lowpass filters, one can estimate the length by an empirical expression, which is built into the program. It may be called by entering zero for the required filter length.

The maximum filter length in this program segment is 512 taps. Frequencies may be entered either in normalized form, in which case the highest frequency must be less than or equal to 0.5, or alternatively in Hz, in which case the last value entered will be considered to be half the sampling rate (the so-called *Nyquist* frequency).

As an additional feature, we may request nonconstant passband behavior. Here the passband(s) must be specified by a simple table of required loss vs. frequency and contained in a file with an extension of .REQ (for requirements). Any constant segment of the passband(s) needs only its end-points specified, everywhere else we may specify as dense a set of points as we wish, with an upper limit of 501 frequencies.

The file structure is similar to the one used in the OPT optimizing preprocessor (see *Section 14*) and we will demonstrate this feature in the next section in more detail.

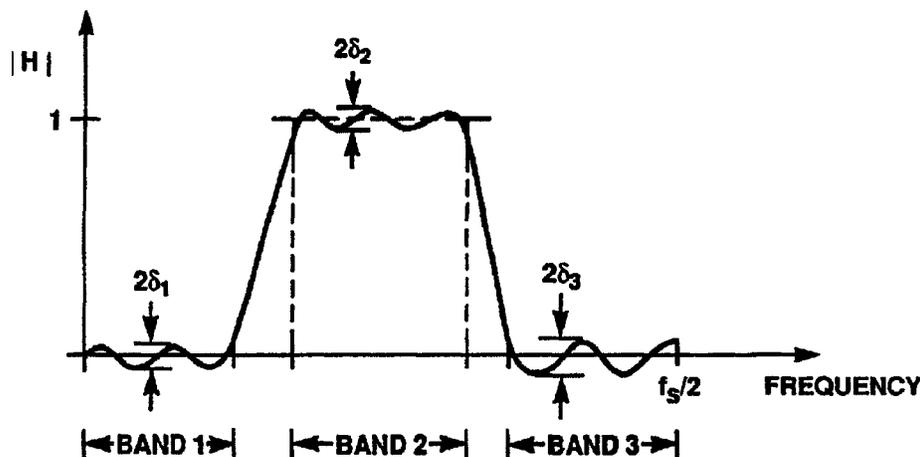


Fig. 11.1 Optimal bandpass design

11.3 DATA INPUT TO EQUAL RIPPLE DESIGN SEGMENT

Input is entered into this program segment in the usual conversational mode, i.e. a question is printed by the program and must be answered by the user. When possible, acceptable answers are also indicated in the question. This method allows the program to check the consistency of the input and prevent it from requesting irrelevant data. Input data can be interrupted by the usual backup symbol (BRK or its alias BAK) and program execution is terminated by entering the STOP command.

Title information is requested first, for labeling purposes only. It is followed by a request for the required filter length (number of taps). In the lowpass case, one may enter zero and the program will estimate the length. The program then asks for the passband ripple and stopband loss from which to estimate the length. In other than the lowpass case, we must do the estimating and, if results were unsatisfactory, rerun the program. Next we must tell the program whether we want a filter, a differentiator or a Hilbert transformer. If we request a filter, we must subsequently specify the number of bands (*both* passbands and stopbands; they need *not* alternate). This must be followed by the edge frequencies, two to a band, either all in Hz or all in normalized frequencies. If the program is requested to estimate the length, some of the questions are not asked, since such a request implies that the filter is a lowpass. For differentiators and Hilbert transformers, the number of bands is automatically set to one. For Hilbert transformers this

11.4

band may not start at 0. and for differentiators it may not extend to 0.5 (or the Nyquist frequency).

Next, to specify which band is which, we enter the required function values, one per band. We should specify 1.0 for passbands and 0.0 for stopbands, although other values are also allowed. For Hilbert transformers 1.0 should be entered and for differentiators, the required slope in normalized form, should be entered. Finally we must enter either a set of weights or a set of loss values, one per band. The loss values are the passband ripple in the passbands or the minimum stopband loss in the stopbands. The relationship between these weights and loss values is given by the equations:

$$w_p = (10^{(a_p/20)} + 1)/(10^{(a_p/20)} - 1) \quad (11.3a)$$

and

$$w_s = 10^{(a_s/20)} \quad (11.3b)$$

in the passbands and stopbands, respectively.

If the length of the filter is adequate, the weights will insure that the passband and stopband requirements are met simultaneously. Otherwise, both will deviate from the requirements by varying amounts. Also, it should be noted that only the relative weights are of importance. Multiplying all of them by the same factor will leave the final results unchanged.

As usual, recognizably incorrect entries are flagged down by the program and requested again. Entries that are accepted, though incorrect, may be corrected by the use of the backup symbol (BRK). The program may always be terminated by entering STOP.

11.4 EQUAL RIPPLE DESIGN OUTPUT

Most of the printout of this segment is self-explanatory. Heading and labeling are followed by a couple of explanatory items and then by the listing of coefficients (impulse response data) in both decimal and hexadecimal formats. Decimals are printed to eight places, hexadecimals to 10. If we so desire, we may also obtain a printer plot of the impulse response.

Next, we are given a band-by-band performance summary, containing the normalized band edge frequencies, desired function value and weight, maximum deviation from desired value (δ_i values in Fig. 11.1) and the corresponding passband ripple or stopband loss values, all in dB. At the conclusion of this tabulation we automatically enter the analysis and manipulation segment of the program. Before considering that segment, we will illustrate the design of equal ripple filters by the use of examples.

11.5 EQUAL RIPPLE DESIGN EXAMPLES

Example 11.5.1 Lowpass Filter

We will design a lowpass filter with passband up to 1500 Hz and passband ripple of 1.3 dB. The stopband starts at 2000 Hz and goes all the way to 5000 Hz, which is the Nyquist frequency, that is to say, the sampling frequency is 10 kHz. We need 62 dB minimum stopband loss at the Nyquist frequency. The data input is:

```
C:>fir
***** S/FILSYN *****
RELEASE 3.2  VERSION 1  4/1/94

** FIR  SEGMENT **

Copyright (C) 1983 - 1995 Dr. George Szentirmai
All Rights Reserved.

SYNTHESIS: S, ANALYSIS: A OR END: E
> s
ENTER TITLE
> example # 1
DESIGN - EQUAL RIPPLE: 1 OR WINDOWED: 2
> 1
ENTER FILTER LENGTH
> 0
ENTER PASSBAND RIPPLE IN DB
> 1.3
ENTER STOPBAND LOSS IN DB
> 62
THIS WILL BE A LOWPASS FILTER
ENTER BANDEDGE FREQUENCIES ( 4 VALUES)
> 0 1500 2000 5000
```

The single-executable version differs from the PC multi-executable version above, because there the FIR segment is an integral part of S/FILSYN. To access the FIR segment in a workstation, or mainframe version, one enters:

```
vax2 $ run sfilsyn
***** S/FILSYN *****
RELEASE 3.2  VERSION 1  4/1/94

Copyright (C) 1983 - 1995 Dr. George Szentirmai
All Rights Reserved.

READ DATA FROM FILE: Y/N
> n
PLACER: P, SMAIN: S, LADDER: L, DIGITAL: D, ACTIVE: A OR END: E
> s  ! Always use this for synthesis.
ENTER TITLE
> example # 1
FILTER KIND - LUMPED: 0, DIGITAL: 1 OR MICROWAVE: 2
> 1  ! Pick the digital synthesis.
DIGITAL - FIR: F OR BILINEAR IIR: I
> f  ! Finally select the FIR segment.
DESIGN - EQUAL RIPPLE: 1 OR WINDOWED: 2
> ....
```

The "vax2 \$" above is the system prompt. From this point the two versions are identical. Since we requested the program to estimate the filter length, it automatically assumed a lowpass and thus skipped some of the questions. The results indicate that the program estimated the need for a 39 tap filter:

```
ESTIMATED DEGREE = 39
*****
```

11.6

FINITE IMPULSE RESPONSE (FIR)
 LINEAR PHASE DIGITAL FILTER DESIGN

example # 1

REMEZ EXCHANGE ALGORITHM
 BANDPASS FILTER
 FILTER LENGTH = 39
 FILTER LENGTH DETERMINED BY APPROXIMATION
 ***** IMPULSE RESPONSE *****

FUNCTION NO. 0
 SAMPLING FREQUENCY = 1.0000000D+04

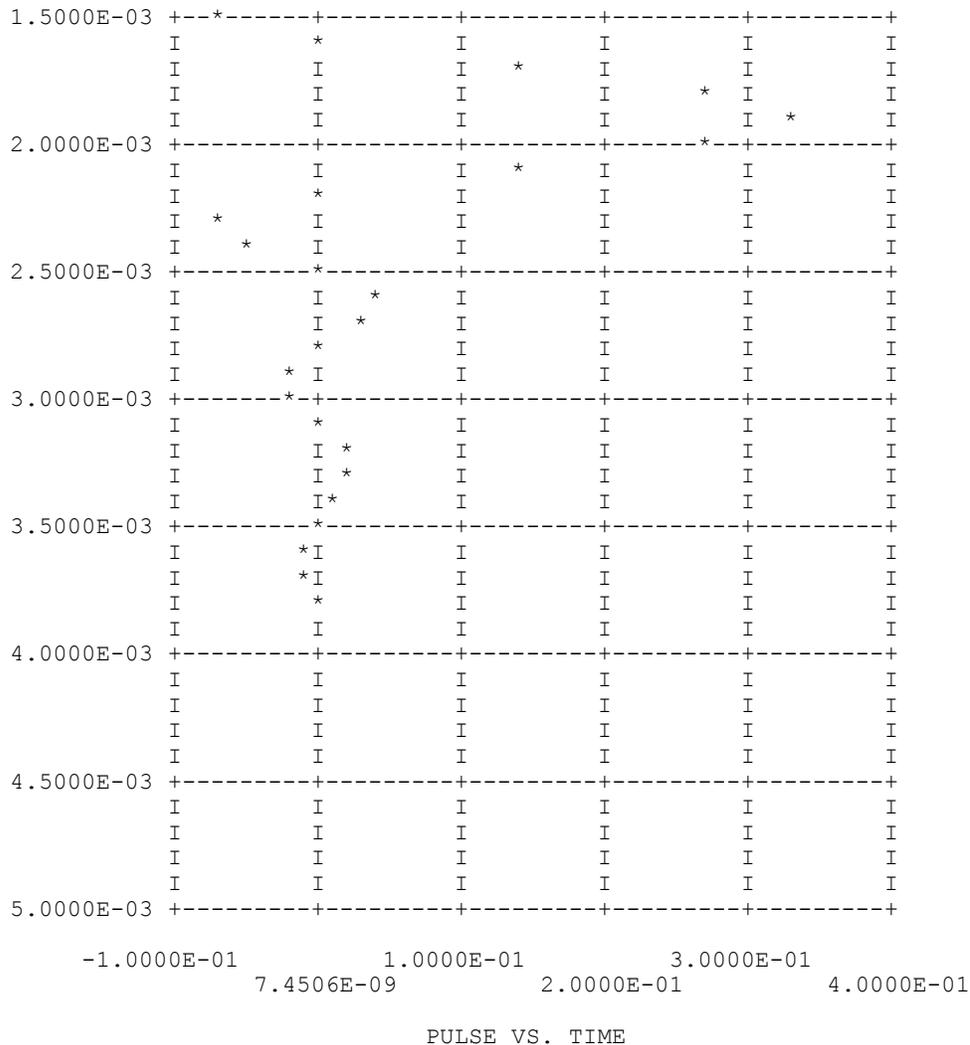
DECIMAL		HEXADECIMAL	
H(1) =	-2.5007369D-03	= H(39) =	-0.00A3E3670
H(2) =	-5.4111211D-03	= H(38) =	-0.01629F8C0
H(3) =	-5.4150233D-03	= H(37) =	-0.0162E1040
H(4) =	1.6054693D-03	= H(36) =	0.0069374E0
H(5) =	1.4557569D-02	= H(35) =	0.03BA0B7C0
H(6) =	2.4822060D-02	= H(34) =	0.065ABD100
H(7) =	2.1755816D-02	= H(33) =	0.0591CA080
H(8) =	3.6276549D-03	= H(32) =	0.00EDBDF30
H(9) =	-1.6521698D-02	= H(31) =	-0.043AC4180
H(10) =	-1.9262802D-02	= H(30) =	-0.04EE68300
H(11) =	2.3680239D-03	= H(29) =	0.009B30D90
H(12) =	3.0874243D-02	= H(28) =	0.07E75FD80
H(13) =	3.5031106D-02	= H(27) =	0.08F7CC700
H(14) =	-4.8954837D-04	= H(26) =	-0.002015424
H(15) =	-5.1926974D-02	= H(25) =	-0.0D4B16100
H(16) =	-6.5564498D-02	= H(24) =	-0.10C8D5C00
H(17) =	9.6633856D-04	= H(23) =	0.003F54788
H(18) =	1.3695237D-01	= H(22) =	0.230F4F800
H(19) =	2.7466232D-01	= H(21) =	0.465045000
H(20) =	3.3263367D-01	= H(20) =	0.55277B000

WISH TO WRITE RESPONSE DATA ON FILE? (Y/N)
 > n
 PLOT - WIDE: W, NARROW: N, GRAPHICS: G OR END: E
 > n
 PLOT - NO: N, PULSE: P, STEP: S
 > p
 ENTER STARTING AND ENDING TIME OF PLOT
 > 0 1

The 'FUNCTION NO. 0' legend, shown immediately above the listing, is printed to distinguish it from other functions which may be computed later. We now can have this impulse response plotted. The plot (generated by a workstation system) is shown on the next page. The personal computer form is slightly different.

```

0.0000E+00 +-----*-----+-----+-----+-----+
            I      *I      I      I      I      I
            I      *I      I      I      I      I
            I      *      I      I      I      I
            I      I*      I      I      I      I
5.0000E-04 +-----+*-----+-----+-----+-----+
            I      I *      I      I      I      I
            I      *      I      I      I      I
            I      * I      I      I      I      I
            I      * I      I      I      I      I
1.0000E-03 +-----*-----+-----+-----+-----+
            I      I *      I      I      I      I
            I      I *      I      I      I      I
            I      *      I      I      I      I
            I      * I      I      I      I      I
  
```



The plot is followed by the performance summary, which indicates that the estimated filter length does not quite make it. We would probably need a 41 tap design to meet the requirements:

	BAND 1	BAND 2	BAND
LOWER BAND EDGE	.000000000	.200000000	
UPPER BAND EDGE	.150000000	.500000000	
DESIRED VALUE	1.000000000	.000000000	
WEIGHTING	1.000000000	94.034970000	
DEVIATION	.092894810	.000987875	
DEVIATION IN DB	1.618414000	-60.105960000	

After the summary has been printed, the program automatically proceeds to the analysis segment with its own COMMAND mode of operation. However, we will postpone any discussion of this second segment and its functions until later in this manual. What we want to do now is to save the data on a permanent file for later use. We use the SAVE command to save the data on a file called EXNO1.QFF. The END command then terminates this phase of the operation to let us proceed with our examples.

11.8

Example 11.5.2 Band-Elimination Filter

Instead of repeating the previous design with 41 taps, we will design a length 25 band-elimination filter. We arbitrarily select the stopband weight of 20 with reference to a passband weight of unity.

```
C:>fir
***** S/FILSYN *****
RELEASE 3.2  VERSION 1  4/1/94

** FIR  SEGMENT **

Copyright (C) 1983 - 1995 Dr. George Szentirmai
All Rights Reserved.

SYNTHESIS: S, ANALYSIS: A OR END: E
> s
ENTER TITLE
> example # 2
DESIGN - EQUAL RIPPLE: 1 OR WINDOWED: 2
> 1
ENTER FILTER LENGTH
> 25
FILTER TYPE - FILTER: 1, DIFF.: 2, HILBERT TR.: 3
> 1
ENTER NO. OF BANDS
> 3
ENTER BANDEDGE FREQUENCIES ( 6 VALUES)
> 0 1500 2000 3000 3500 5000
ENTER 3 FUNCTION VALUES OR SLOPE
> 1,0,1
SPECIFY WEIGHTS: 0 OR LOSS VALUES: 1
> 0
ENTER 3 WEIGHTS
> 1, 20 1
IS PASSBAND SHAPED? (Y/N)
> n
```

This time we decided not to ask for the plot of the impulse response:

```
*****
FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN

example # 2
REMEZ EXCHANGE ALGORITHM
BANDPASS FILTER
FILTER LENGTH = 25
***** IMPULSE RESPONSE *****

FUNCTION NO. 0
SAMPLING FREQUENCY = 1.0000000D+04

DECIMAL                                HEXADECIMAL
H( 1) = -4.4832829D-02 = H( 25) = -0.0B7A2A100
H( 2) =  1.3489886D-04 = H( 24) =  0.0008D73A3
H( 3) = -6.1468448D-02 = H( 23) = -0.0FBC65700
H( 4) =  2.6490481D-04 = H( 22) =  0.00115C5D8
H( 5) =  5.8800016D-02 = H( 21) =  0.0F0D84900
H( 6) =  3.0257646D-04 = H( 20) =  0.0013D4640
H( 7) =  3.5541549D-03 = H( 19) =  0.00E8ECD30
H( 8) =  2.0278530D-04 = H( 18) =  0.000D4A2C4
H( 9) = -1.3293089D-01 = H( 17) = -0.2207C2400
H(10) =  1.7419107D-04 = H( 16) =  0.000B6A70F
```

```

H( 11) = 2.7070156D-01 = H( 15) = 0.454CB2800
H( 12) = 1.9221375D-04 = H( 14) = 0.000C98CFC
H( 13) = 6.7070913D-01 = H( 13) = 0.ABB398000
WISH TO WRITE RESPONSE DATA ON FILE? (Y/N)
> n
PLOT - WIDE: W, NARROW: N, GRAPHICS: G OR END: E
> e

```

```

          BAND 1          BAND 2          BAND 3          BAND
LOWER BAND EDGE    .000000000    .200000000    .350000000
UPPER BAND EDGE    .150000000    .300000000    .500000000
DESIRED VALUE      1.000000000    .000000000    1.000000000
WEIGHTING          1.000000000    20.000000000  1.000000000
DEVIATION          .144187200    .007209357    .144187200
DEVIATION IN DB    2.522366000   -42.842070000  2.522366000
*****

```

At the end of this printout we again find ourselves in the analysis segment. We will therefore analyze the filter in the frequency domain. This is done by the familiar `FREQ` command, which subsequently prompts us to specify the frequencies for the analysis. We specify a single range from 0 to 5000 Hz, with a 100 Hz step size.

```

COMMAND:
> freq
ENTER FREQ:
> 0 5000 100
ENTER FREQ:
>
TABULATE: Y/N
> y
example # 2

```

RESULTS OF THE ANALYSIS, 5X, 12HFUNCTION NO. 0 BITLENGTH = 48

FREQUENCY IN HZ	GAIN IN DB	PHASE IN DEGREES	DELAY IN SECONDS
0.000000E+00	-1.3010	.0000	1.2000E-03
1.000000E+02	-.9394	43.2000	1.2000E-03
2.000000E+02	-.1014	86.4000	1.2000E-03
3.000000E+02	.7223	129.6000	1.2000E-03
4.000000E+02	1.1497	172.8000	1.2000E-03
5.000000E+02	1.0165	216.0000	1.2000E-03
6.000000E+02	.3651	259.2000	1.2000E-03
7.000000E+02	-.5442	302.4000	1.2000E-03
8.000000E+02	-1.2424	345.6000	1.2000E-03
9.000000E+02	-1.2729	28.8000	1.2000E-03
1.000000E+03	-.5996	72.0000	1.2000E-03
1.100000E+03	.3440	115.2000	1.2000E-03
1.200000E+03	1.0371	158.4000	1.2000E-03
1.300000E+03	1.1179	201.6000	1.2000E-03
1.400000E+03	.3719	244.8000	1.2000E-03
1.500000E+03	-1.3524	288.0000	1.2000E-03
1.600000E+03	-4.2177	331.2000	1.2000E-03
1.700000E+03	-8.4731	14.4000	1.2000E-03
1.800000E+03	-14.5970	57.6000	1.2000E-03
1.900000E+03	-23.7997	100.8000	1.2000E-03
2.000000E+03	-42.8421	144.0001	1.2000E-03
2.100000E+03	-42.8028	7.1999	1.2000E-03
2.200000E+03	-50.3423	50.4002	1.2000E-03
2.300000E+03	-49.2786	273.6000	1.2000E-03
2.400000E+03	-43.6838	316.8000	1.2000E-03
2.500000E+03	-42.8447	.0000	1.2000E-03
2.600000E+03	-43.5219	43.2000	1.2000E-03
2.700000E+03	-48.8179	86.3999	1.2000E-03
2.800000E+03	-50.8327	309.6003	1.2000E-03

11.10

2.900000E+03	-42.9058	352.8000	1.2000E-03
3.000000E+03	-42.8421	216.0001	1.2000E-03
3.100000E+03	-23.8041	259.2000	1.2000E-03
3.200000E+03	-14.5968	302.4000	1.2000E-03
3.300000E+03	-8.4714	345.6000	1.2000E-03
3.400000E+03	-4.2163	28.8001	1.2000E-03
3.500000E+03	-1.3524	72.0001	1.2000E-03
3.600000E+03	.3699	115.2001	1.2000E-03
3.700000E+03	1.1137	158.4000	1.2000E-03
3.800000E+03	1.0309	201.6001	1.2000E-03
3.900000E+03	.3371	244.8000	1.2000E-03
4.000000E+03	-.6044	288.0001	1.2000E-03
4.100000E+03	-1.2718	331.2000	1.2000E-03
4.200000E+03	-1.2337	14.4000	1.2000E-03
4.300000E+03	-.5306	57.6001	1.2000E-03
4.400000E+03	.3785	100.8000	1.2000E-03
4.500000E+03	1.0249	144.0001	1.2000E-03
4.600000E+03	1.1490	187.2000	1.2000E-03
4.700000E+03	.7084	230.4001	1.2000E-03
4.800000E+03	-.1312	273.6001	1.2000E-03
4.900000E+03	-.9843	316.8000	1.2000E-03
5.000000E+03	-1.3524	.0001	1.2000E-03

These results indicate an acceptable filter with linear phase and constant delay. Incidentally, this delay calculation is exact, although an occasional 0./0. computation may cause some glitches. The last item in the header, i.e., BITLENGTH = 48 indicates that full machine precision was used for the tap weights in these calculations.

Example 1.5.3 Hilbert Transformer

This example is a 20 tap Hilbert transformer. The input is even briefer than usual since the number of bands and the corresponding function values are preset:

```
C:>fir
***** S/FILSYN *****
RELEASE 3.2  VERSION 1  4/1/94

** FIR  SEGMENT **

Copyright (C) 1983 - 1995 Dr. George Szentirmai
All Rights Reserved.

SYNTHESIS: S, ANALYSIS: A OR END: E
> s
ENTER TITLE
> example # 3
DESIGN - EQUAL RIPPLE: 1 OR WINDOWED: 2
> 1
ENTER FILTER LENGTH
> 3
INCORRECT, PLEASE REPEAT
ENTER FILTER LENGTH
> 20
FILTER TYPE - FILTER: 1, DIFF.: 2, HILBERT TR.: 3
> 3
ENTER BANDEDGE FREQUENCIES ( 2 VALUES)
> 500 5000
ENTER SAMPLING RATE IN HZ
> 10k
```

The results are:

```

FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN

example # 3

REMEZ EXCHANGE ALGORITHM
HILBERT TRANSFORMER
FILTER LENGTH = 20

***** IMPULSE RESPONSE *****

FUNCTION NO. 0
SAMPLING FREQUENCY = 1.0000000D+04

DECIMAL                                HEXADECIMAL
H( 1) = 1.6026309D-02 = -H( 20) = 0.041A4CD80
H( 2) = 1.4173329D-02 = -H( 19) = 0.03A0DD000
H( 3) = 2.0452505D-02 = -H( 18) = 0.053C60180
H( 4) = 2.8736923D-02 = -H( 17) = 0.075B4D900
H( 5) = 3.9852612D-02 = -H( 16) = 0.0A33C7E00
H( 6) = 5.5333309D-02 = -H( 15) = 0.0E2A52E00
H( 7) = 7.8542754D-02 = -H( 14) = 0.141B60C00
H( 8) = 1.1823758D-01 = -H( 13) = 0.1E44D1800
H( 9) = 2.0664129D-01 = -H( 12) = 0.34E671800
H(10) = 6.3475609D-01 = -H( 11) = 0.A27F60000
WISH TO WRITE RESPONSE DATA ON FILE? (Y/N)
> n
PLOT - WIDE: W, NARROW: N, GRAPHICS: G OR END: E
> e

BAND 1      BAND
LOWER BAND EDGE    .050000000
UPPER BAND EDGE    .500000000
DESIRED VALUE      1.000000000
WEIGHTING          1.000000000
DEVIATION          .020556010

```

Note that a filter length of 3 or shorter will not be accepted by the program. The analysis of this circuit does not show much, i.e., the magnitude is flat above 500 Hz with less than 0.36 dB peak-to-peak ripple.

Example 11.5.4 Lowpass Filter

In preparation for the examples of the analysis segment, our final example is a lowpass with the same bands as our example 11.5.1 above, except that we now request a 1 dB passband ripple and 15 dB stopband suppression. These relatively modest requirements are selected to keep the length of these examples and hence this manual within bounds.

```

C:>fir
***** S/FILSYN *****
RELEASE 3.2 VERSION 1 4/1/94

** FIR SEGMENT **

Copyright (C) 1983 - 1995 Dr. George Szentirmai
All Rights Reserved.

SYNTHESIS: S, ANALYSIS: A OR END: E
> s
ENTER TITLE
> example # 4

```

11.12

```
DESIGN - EQUAL RIPPLE: 1 OR WINDOWED: 2
> 1
ENTER FILTER LENGTH
> 0
ENTER PASSBAND RIPPLE IN DB
> .1
ENTER STOPBAND LOSS IN DB
> 15.
THIS WILL BE A LOWPASS FILTER
ENTER BANDEDGE FREQUENCIES ( 4 VALUES)
> 0. 1500 2k 5k
ESTIMATED DEGREE = 27
```

The resulting filter data are shown below. They indicate a computer estimated length of 27, which, as the summary below shows, is satisfactory. We will use these results later on for comparison, after discussing the features available in the analysis segment.

```
FINITE IMPULSE RESPONSE (FIR)
  LINEAR PHASE DIGITAL FILTER DESIGN

example # 4

  REMEZ EXCHANGE ALGORITHM
  BANDPASS FILTER
  FILTER LENGTH = 27
  FILTER LENGTH DETERMINED BY APPROXIMATION
  ***** IMPULSE RESPONSE *****

      FUNCTION NO. 0
  SAMPLING FREQUENCY = 1.0000000D+04

      DECIMAL                      HEXADECIMAL
H( 1) = -2.9046400D-02 = H( 27) = -0.076F95B80
H( 2) =  4.8885655D-02 = H( 26) =  0.0C83C5300
H( 3) =  3.6612619D-04 = H( 25) =  0.0017FE940
H( 4) = -2.6308274D-02 = H( 24) = -0.06BC23980
H( 5) = -2.3689516D-02 = H( 23) = -0.061084200
H( 6) =  8.6471243D-03 = H( 22) =  0.0236B2AC0
H( 7) =  3.9819255D-02 = H( 21) =  0.0A3198400
H( 8) =  2.7704213D-02 = H( 20) =  0.07179F900
H( 9) = -3.0453444D-02 = H( 19) = -0.07CBCC000
H(10) = -7.5544290D-02 = H( 18) = -0.1356DDE00
H(11) = -3.0934468D-02 = H( 17) = -0.07EB52400
H(12) =  1.1770092D-01 = H( 16) =  0.1E21A5C00
H(13) =  2.8970751D-01 = H( 15) =  0.4A2A45800
H(14) =  3.6577240D-01 = H( 14) =  0.5DA342800
WISH TO WRITE RESPONSE DATA ON FILE? (Y/N)
> n
PLOT - WIDE: W, NARROW: N, GRAPHICS: G OR END: E
> e

      BAND 1          BAND 2          BAND
LOWER BAND EDGE    .000000000    .200000000
UPPER BAND EDGE    .150000000    .500000000
DESIRED VALUE      1.000000000    .000000000
WEIGHTING          1.000000000    .032370610
DEVIATION          .004415507    .136404800
DEVIATION IN DB    .076705230   -17.303410000
```

11.6 WINDOWED DESIGN METHOD

As was mentioned in *Section 11.1*, we can also design linear phase digital FIR filters through the use of the familiar windowing technique. About 20 different windowing functions are built into the program. No discussion of the relative merits of these windows, or any comparison of the two techniques, will be offered here. However, the references in *Appendix A* are sufficient for that purpose.

We need only say that we may design lowpass, highpass, bandpass and bandreject filters with even fewer pieces of input needed, than for equal ripple design. We must enter the filter length (no estimate is available here), its type, and the bandedge frequencies, where the final entry is the *Nyquist* frequency (half the sampling frequency). Finally, we must select a window, and in a few cases, an associated parameter. The window is selected by a numerical key; the HELP response yields a list of the windows and their keys.

These entries are followed by the usual printout of the impulse response but with no performance summary. In order to observe the performance, we must analyze the resulting filter.

Example 11.6.1 Windowed Bandpass

We will design a bandpass filter from 1000 Hz to 2500 Hz, with a sampling frequency of 10 kHz (i.e., a Nyquist rate of 5 kHz). We will use a Kaiser type window with 35 dB loss. The following is the data input sequence:

```
C:>fir
***** S/FILSYN *****
RELEASE 3.2 VERSION 1 4/1/94

** FIR SEGMENT **

Copyright (C) 1983 - 1995 Dr. George Szentirmai
All Rights Reserved.

SYNTHESIS: S, ANALYSIS: A OR END: E
> s
ENTER TITLE
> windowed bandpass
DESIGN - EQUAL RIPPLE: 1 OR WINDOWED: 2
> 2
TYPE - LOWPASS: 1, HIGHPASS: 2, BANDPASS: 4 OR BAND-REJECT: 5
> 4
ENTER TWO PASSBAND EDGE FREQUENCIES AND THE NYQUIST FREQUENCY IN HZ
> 1k 2.5k 5k
ENTER FILTER LENGTH
> 20
ENTER WINDOW TYPE
> help
WINDOW TYPES -

0: RECTANGULAR
1: TRIANGULAR
2: HAMMING
3: HANN
4: COSINE
5: COS**3
6: COS**4
7: BLACKMAN
```

11.14

```
8: 'EXACT' BLACKMAN
9: 3-TERM BLACKMAN-HARRIS
10: 3-TERM 18 DB/OCT
11: OPTIMUM 3-TERM
12: 4-TERM BLACKMAN-HARRIS
13: 4-TERM 40 DB/OCT
14: 4-TERM 18 DB/OCT
15: OPTIMUM 4-TERM
16: KAISER
17: CHEBYSHEV
18: GAUSSIAN
19: TAYLOR
20: 4-TERM SAMPLED KAISER
21: BARCILON-TEMES
```

```
ENTER WINDOW TYPE
> 16
ENTER LOSS VALUE IN DB
> 35
```

Since we did not remember the codes for the window types, when we entered HELP in response to the ENTER WINDOW TYPE query, we were provided with the list shown. This list is the mainframe-workstation version, the personal computer version help messages are substantially more elaborate, containing detailed descriptions of the window functions, even the equations where feasible. The output is:

```
WINDOWED LINEAR-PHASE FILTER DESIGN
windowed bandpass

FILTER TYPE : BANDPASS
WINDOW TYPE : KAISER

PASSBAND EDGE = 1.0000000D+03 HZ.
PASSBAND EDGE = 2.5000000D+03 HZ.
SAMPLING RATE = 1.0000000D+04 HZ.
***** IMPULSE RESPONSE *****
FUNCTION NO. 0
SAMPLING FREQUENCY = 1.0000000D+04

          DECIMAL                      HEXADECIMAL
H( 1) = 8.2687390D-03 = H( 20) = 0.021DE66BC
H( 2) = 1.9673276D-02 = H( 19) = 0.05094ECEB
H( 3) = 5.6678388D-03 = H( 18) = 0.0173728E8
H( 4) = 2.8278938D-03 = H( 17) = 0.00B9542F6
H( 5) = 3.9618460D-02 = H( 16) = 0.0A246F741
H( 6) = 2.1744355D-02 = H( 15) = 0.059109BD8
H( 7) = -1.1823414D-01 = H( 14) = -0.1E4497C2A
H( 8) = -2.0114795D-01 = H( 13) = -0.337E6E9B1
H( 9) = -2.1035649D-02 = H( 12) = -0.056297A17
H( 10) = 2.5265657D-01 = H( 11) = 0.40AE19EA7
```

Due to a quirk in the program, upon completion of the design we do not get the expected prompt for saving or plotting the data. However, if we again print it, these options will be offered.

Since we are again in the analysis segment, we will request a frequency domain analysis which we will not tabulate but only plot. The results are shown below and illustrate about a 6 dB loss at the band edges and a minimum of 40 dB stopband loss below about 400 Hz and above 3150 Hz.

```

COMMAND:
> freq
ENTER FREQ:
> 0 5k max
ENTER FREQ:
>
TABULATE: Y/N
> n
WISH TO WRITE ANALYSIS DATA ON FILE: Y/N
> y
ENTER FILENAME
> test
** DONE **

PLOT - NO: N, LOSS: L, PHASE: P, DELAY: D
> stop
ARE YOU SURE? (Y/N)
> y
*** PROGRAM TERMINATED ***

```

The results of this analysis are shown on the next page, in high-resolution graphics format, obtained using the GRAPH utility.

11.7 SHAPED PASSBAND

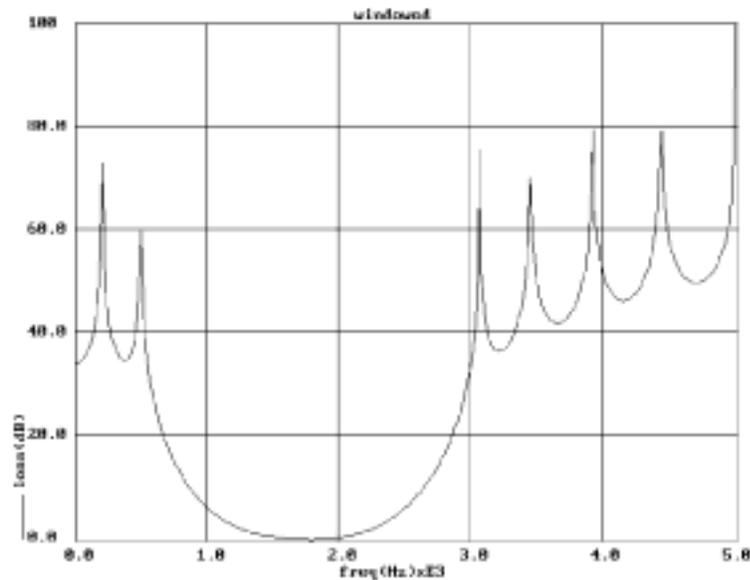
The optimizing preprocessor can now provide us with a shaped passband in the case of infinite impulse response (IIR) digital filters. The same capability was built into the MacClellan-Parks-Rabiner algorithm as well. However, very few versions of that program make this feature available to the user. We have enabled this capability by offering to read a simple tabular data file containing the required shape. This file must have an .REQ extension, and is an ASCII file with the following structure:

```

Line 1: Title           (arbitrary)
Line 2: Blank
Line 3:  FREQ  LOSS    (keywords in that sequence)
Line 4:  kHz   dB      (the first of these is significant and indicates the units used below
                        for the frequencies in the table)
Subsequent lines:      pairs of frequency and loss values in free format.

```

Blank lines and comments (following the ! comment separator) may appear anywhere in the file.



The table should contain a reasonably dense set of frequencies and cover *all* passbands (i.e. ranges where the function value is specified as nonzero), except that a range where the function is to remain constant, needs only their endpoints specified.

Example 11.7.1 Lowpass Filter with Shaped Passband

We will design a lowpass filter with zero loss from zero up to 2kHz and a loss rising linearly from zero to 3dB in the range from 2kHz to 3kHz. The stopband starts at 3.5 kHz and ends at 5kHz which is the Nyquist frequency. The passband shape is stored in the file called FIRTEST.REQ, shown below:

Test for shaped FIR filter

FREQ kHz	LOSS dB
0.0	0.0
2.0	0.0
2.1	0.3
2.2	0.6
2.3	0.9
2.4	1.2
2.5	1.5
2.6	1.8
2.7	2.1
2.8	2.4
2.9	2.7
3.0	3.0

Note that we only need to have the first two lines of data for the first, constant, segment of the passband, while the linear range is specified by 10 additional points.

Next, we use this file in the FIR data entry segment, where we shall meet a new prompt about "IS PASSBAND SHAPED? (Y/N)". If we answer N<o>, we get the standard behavior, with

constant passbands; on the other hand answering Y<es> will enable us to read this file and thus enter the required shape into the program:

```
C:>fir
***** S/FILSYN *****
RELEASE 3.2 VERSION 1 4/1/94

** FIR SEGMENT **

Copyright (C) 1983-1995 Dr. George Szentirmai.
All Rights Reserved.

SYNTHESIS: S, ANALYSIS: A OR END: E
> s
ENTER TITLE
>
DESIGN - EQUAL RIPPLE: 1 OR WINDOWED: 2
> 1
ENTER FILTER LENGTH
> 45
FILTER TYPE - FILTER: 1, DIFF.: 2, HILBERT TR.: 3
> 1
ENTER NO. OF BANDS
> 2
ENTER BANDEDGE FREQUENCIES ( 4 VALUES)
> 0 3k 3.5k 5k
ENTER 2 FUNCTION VALUES OR SLOPE
> 1 0
SPECIFY WEIGHTS: 0 OR LOSS VALUES: 1
> 0
ENTER 2 WEIGHTS
> 1 1
IS PASSBAND SHAPED? (Y/N)
> y
WE NEED THE FILE CONTAINING TABULATED LOSS SHAPE NEXT
ENTER FILE NAME
> firtest
*****
FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN

Test for shaped FIR filter

REMEZ EXCHANGE ALGORITHM
BANDPASS FILTER
FILTER LENGTH = 45
***** IMPULSE RESPONSE *****
FUNCTION NO. 0
SAMPLING FREQUENCY = 1.0000000D+04

DECIMAL                                HEXADECIMAL
H( 1) = 2.4900071D-03 = H( 45) = 0.00A32F630
H( 2) = -1.3969804D-03 = H( 44) = -0.005B8D710
H( 3) = 8.0974423D-04 = H( 43) = 0.003511410
H( 4) = 3.7469489D-03 = H( 42) = 0.00F58F5F0
H( 5) = -4.2136549D-03 = H( 41) = -0.011425660
H( 6) = -1.7016631D-03 = H( 40) = -0.006F852B8
H( 7) = 5.9267581D-03 = H( 39) = 0.01846A800
H( 8) = -3.6784550D-03 = H( 38) = -0.00F1123C0
H( 9) = -2.5486040D-03 = H( 37) = -0.00A7067B0
H(10) = 8.7929228D-03 = H( 36) = 0.024040C40
H(11) = -7.8789787D-03 = H( 35) = -0.02045B540
H(12) = -5.3317696D-03 = H( 34) = -0.015D6C400
H(13) = 1.7315023D-02 = H( 33) = 0.046EC1E00
H(14) = -7.9695247D-03 = H( 32) = -0.020A4A700
H(15) = -1.4539094D-02 = H( 31) = -0.03B8D5840
```

11.18

```

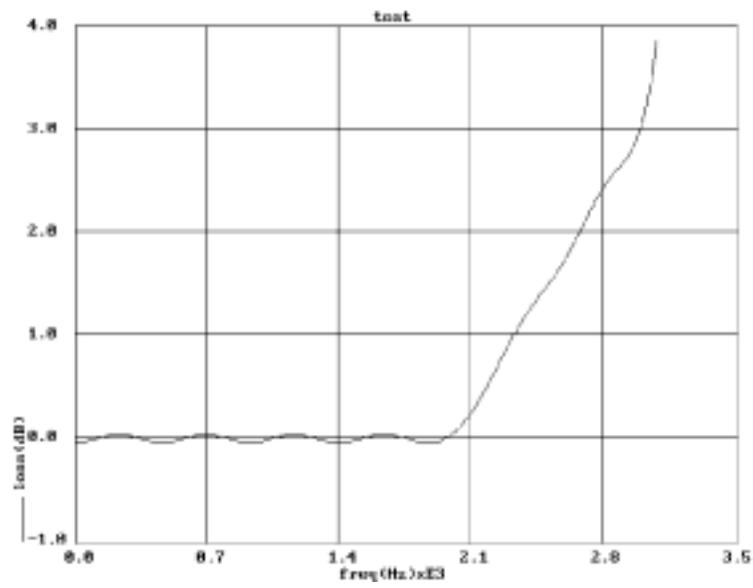
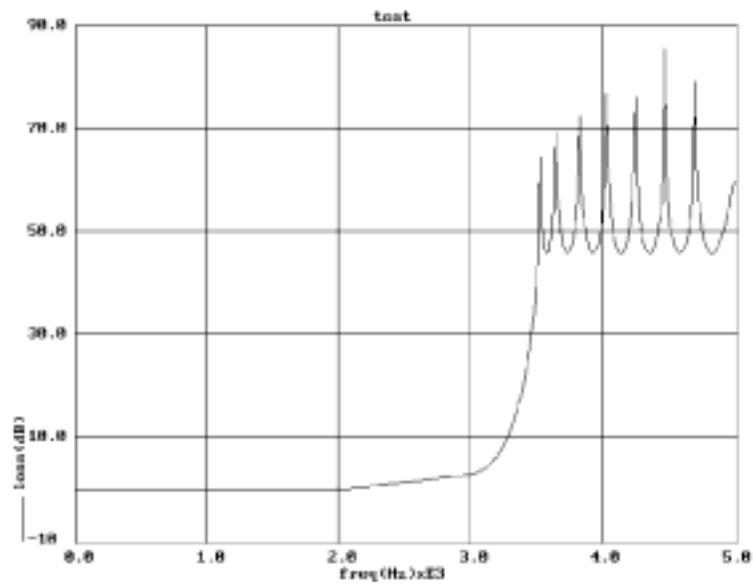
H( 16) = 2.2847325D-02 = H( 30) = 0.05D952800
H( 17) = -9.7352602D-03 = H( 29) = -0.027E02900
H( 18) = -1.7420342D-02 = H( 28) = -0.0475A8D80
H( 19) = 4.9801186D-02 = H( 27) = 0.0CBFC5400
H( 20) = -3.8338307D-02 = H( 26) = -0.09D08A100
H( 21) = -8.8494413D-02 = H( 25) = -0.16A791E00
H( 22) = 2.9195949D-01 = H( 24) = 0.4ABDDDB800
H( 23) = 6.0412008D-01 = H( 23) = 0.9AA79D000
WISH TO WRITE RESPONSE DATA ON FILE? (Y/N)
> n
PLOT - WIDE: W, NARROW: N, GRAPHICS: G OR END: E
> e
          BAND 1          BAND 2          BAND
LOWER BAND EDGE    .000000000    .350000000
UPPER BAND EDGE    .300000000    .500000000
DESIRED VALUE      1.000000000    .000000000
WEIGHTING          1.000000000    1.000000000
DEVIATION          .005004857    .005004857
DEVIATION IN DB    .086944070   -46.012170000
*****

COMMAND:
> freq
ENTER FREQ:
> 0 5k max
ENTER FREQ:
>
TABULATE? (Y/N)
> n
WISH TO WRITE ANALYSIS DATA ON FILE? (Y/N)
> y
ENTER FILE NAME
> firtest
** DONE **

PLOT - WIDE: W, NARROW: N, GRAPHICS: G OR END: E
> stop
ARE YOU SURE? (Y/N)
> y
*** PROGRAM TERMINATED ***

```

The run is otherwise uneventful and the analysis results, shown below, including an enlargement of the passband, indicate perfect agreement with requirements.



11.20

Notes: