

10/
11-19-90 JSC

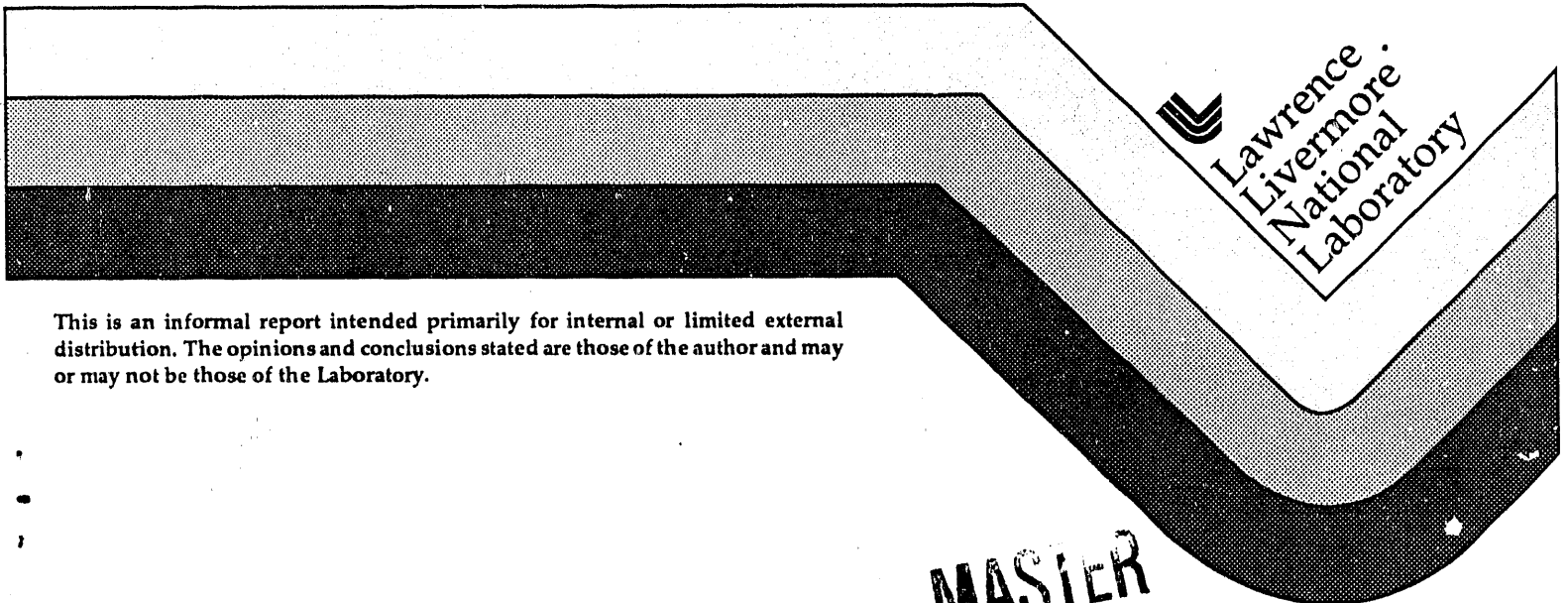
UCRL-ID--106005

DE91 002569

XAPiir: A Recursive Digital Filtering Package

David Harris

September 21, 1990



This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

Price Code

Page Range

A01

Microfiche

Papercopy Prices

A02	1- 10
A03	11- 50
A04	51- 75
A05	76-100
A06	101-125
A07	126-150
A08	151-175
A09	176-200
A10	201-225
A11	226-250
A12	251-275
A13	276-300
A14	301-325
A15	326-350
A16	351-375
A17	376-400
A18	401-425
A19	426-450
A20	451-475
A21	476-500
A22	501-525
A23	526-550
A24	551-575
A25	576-600
A99	601 & UP

Introduction

XAPiir is a basic recursive digital filtering package, containing both design and implementation subroutines. XAPiir was developed for the experimental array processor (XAP) software package, and is written in FORTRAN. However, it is intended to be incorporated into any general- or special-purpose signal analysis program. It replaces the older package RECFIL [1], offering several enhancements. RECFIL is used in several large analysis programs developed at LLNL, including the seismic analysis package SAC [2], several expert systems (NORSEA and NETSEA), and two general purpose signal analysis packages (SIG [3] and VIEW [4]).

The two principal enhancements offered in XAPiir are an increase in speed (a factor of 4 to 5 on Sun Sparc stations) and an increase in numerical stability. The increase in numerical stability is achieved by replacing the direct form implementation of RECFIL with a cascade of second-order sections filter representation. This change is fundamental in the sense that the routines of RECFIL and XAPiir are incompatible. The increase in speed has been achieved through careful optimization of loops in the main implementation subroutine.

XAPiir implements the standard recursive digital filter design through bilinear transformation of prototype analog filters [5]. Lowpass Butterworth, Bessel, and Chebyshev types I and II are the available analog prototypes. These prototype filters are specified in terms of their poles and zeros, then are transformed to highpass, bandpass and band reject filters using analog spectral transformations. The subroutine package is organized in such a manner that other prototype filters can be added easily.

A cascade of second-order sections implementation is provided for filtering data, with a

forward-and-reverse filtering option for zero-phase filtering. Options are provided for in-place filtering to minimize storage requirements.

Several subroutines are provided for supporting the design of digital filters. These include subroutines for computing the frequency response of the designed analog and digital filters (amplitude, phase and group delay), and a routine for calculating the Chebyshev design parameters.

This report is divided into two sections: the first describes the use of the subroutine package, and the second, its internal organization. In the first section, the filter design problem is briefly reviewed, along with the definitions of the filter design parameters and their relationship to the subroutine input parameters. In the second section, the internal organization is documented to simplify maintenance and extensions to the package.

I. Filter Characteristics and Specifications

The recursive filters implemented in XAPiir are obtained through transformation of familiar analog lowpass prototypes: Butterworth, Bessel, and Chebyshev types I and II. These are transformed to one of four passband types: lowpass, highpass, bandpass and band reject. The frequency domain specifications of ideal passbands are illustrated in Figure 1, showing that lowpass filters are described in terms of a high cutoff frequency (f_h), highpass filters by a low frequency cutoff (f_l), and bandpass and band reject filters by both types of cutoffs.

Butterworth Filters

In addition to frequency cutoffs, Butterworth

MASTER

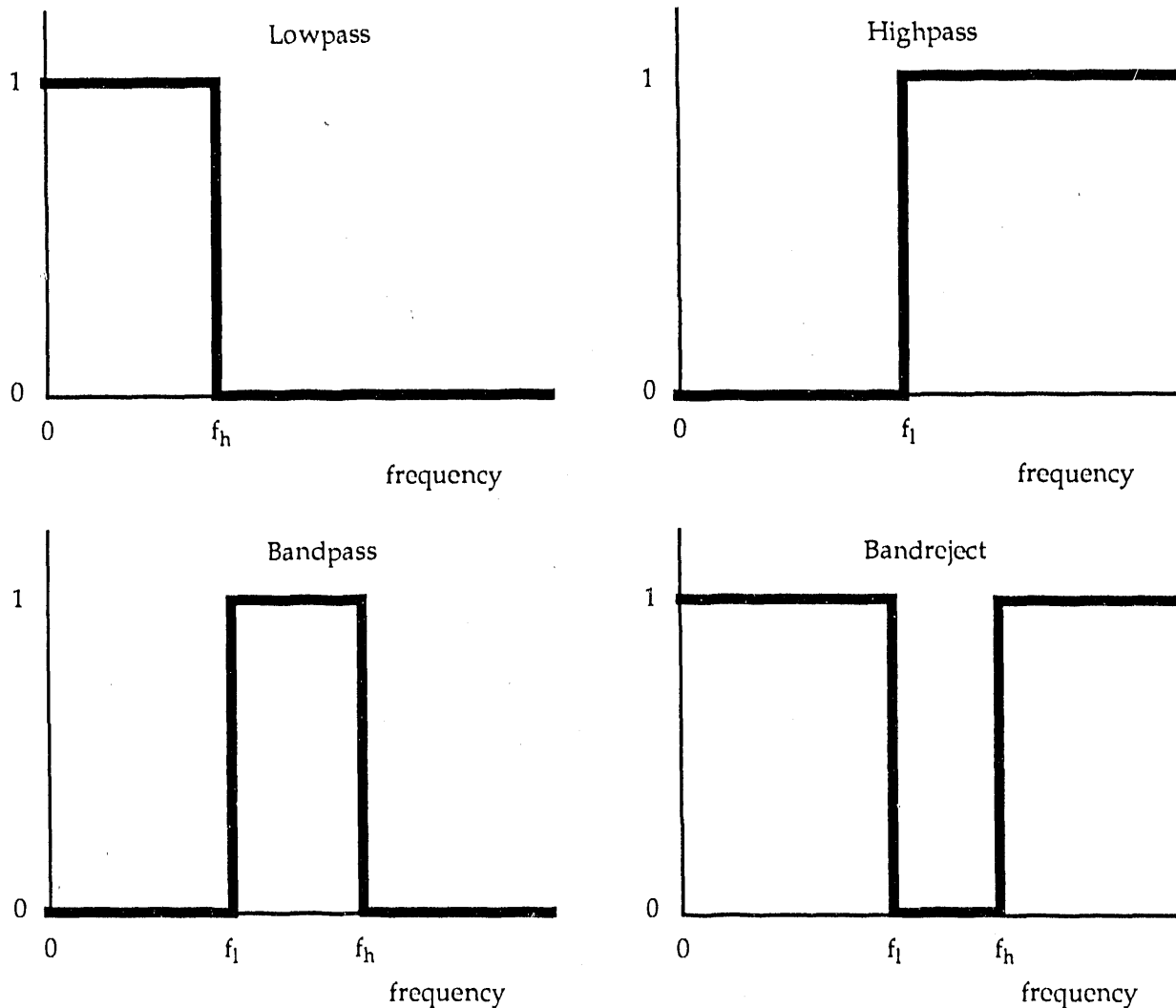


Figure 1 The four passband types implemented by XAPiir. They are specified by the low and high cutoff frequencies, f_l and f_h .

filters are specified by the number of poles used in the design. Butterworth filters are characterized by half-power points at the specified band edges (see Figure 2). From this figure it is apparent that at least 3 or 4 poles are required for sufficient out-of-band signal rejection.

The amplitude, phase, group delay and impulse responses of typical Butterworth filters are shown in Figure 4 for comparison with the responses of the other filter types (shown in subsequent figures). In each section of Figure 4, the responses of

the digital Butterworth filter and its associated analog prototype are overlaid for comparison. Since the digital filters are obtained by bilinear transformation of the analog filters, some distortion of the ideal analog response occurs.

The Butterworth design is obtained using the subroutine DESIGN. The calling sequence for DESIGN is:

```
DESIGN( IORD, TYPE, APROTO,
        A, TRBNDW, FL, FH,
        TS, SN, SD, NSECTS )
```

IORD is the number of poles

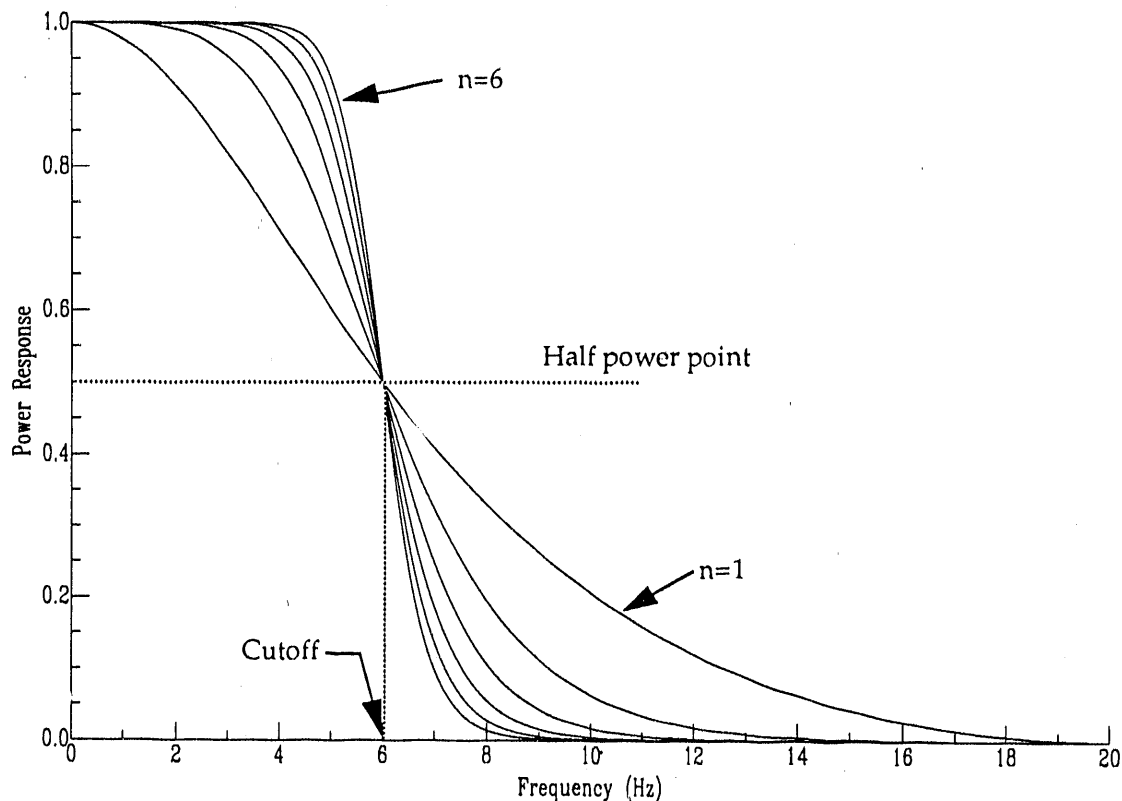


Figure 2 Power response curves for a suite of 6 Hz lowpass Butterworth digital filters. The number of poles (n) ranges from 1 to 6. Note that all curves pass through 0.5 (half-power) at the cutoff frequency. The assumed sampling rate is 40 samples/second.

TYPE is a character variable with values

- 'LP' for lowpass
- 'HP' for highpass
- 'BP' for bandpass
- 'BR' for band reject

APROTO is a character variable which is set to 'BU' for a Butterworth design.

A, and TRBNDW are not used.

FL and FH are the low and high cutoff frequencies in Hertz.

TS is the sampling interval in seconds.

SN, SD and NSECTS are the data structure containing the resulting cascade of second-order sections

representation of the filter.

Bessel Filters

Analog lowpass Bessel filters are designed to have a minimum of phase distortion in the filter passband, specified as a flat group delay at zero frequency. The design parameters for this filter are the same as for the Butterworth: cutoff frequencies and the number of poles. As shown in Figure 5, the bilinear analog-to-digital transformation spoils the flat group delay response of the filter in the passband. Nonetheless, the digital Bessel filter has the least frequency dispersion among all the designs implemented in XAPiir.

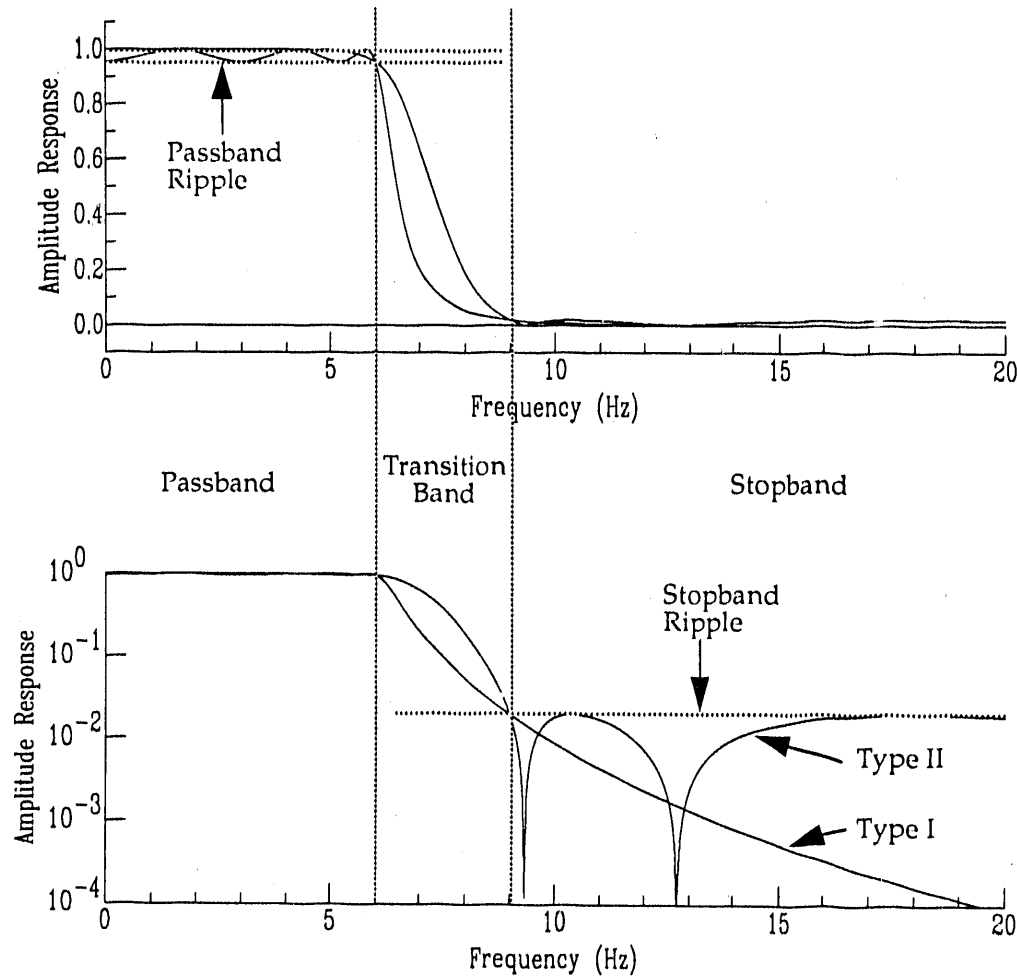


Figure 3 Amplitude response characteristics of the analog lowpass Chebyshev types I and II filters. The filters are characterized by a passband (nominal amplitude: 1), a stopband (nominal amplitude: $1/\text{attenuation}$), and a transition band. The design parameters are the stopband attenuation, the width of the transition band (as a fraction of the passband width), and the number of poles. The passband ripple is a function of these parameters.

Chebyshev Filters

There is little reason to use the Bessel filter for bandpass, bandreject or highpass designs, since its desirable lowpass phase characteristics are not preserved by the analog spectral transformations used to derive the other passband types.

The call to DESIGN for the Bessel filter is identical to that for the Butterworth filter, with the exception that the analog prototype argument APROTO is set to 'BE'.

The Chebyshev filters offer the greatest frequency selectivity of all the analog prototypes implemented in XAPiir, but are more complicated to specify and design. The lowpass prototypes are characterized by a passband where the filter amplitude is nominally one, a stopband where the amplitude is specified not to exceed some small value (specified as the reciprocal of an attenuation factor), and a transition band between the other two bands. The design parameters may be chosen among the following: the cutoff frequency, the number of poles, the transition bandwidth, the stopband

Butterworth Filter Responses

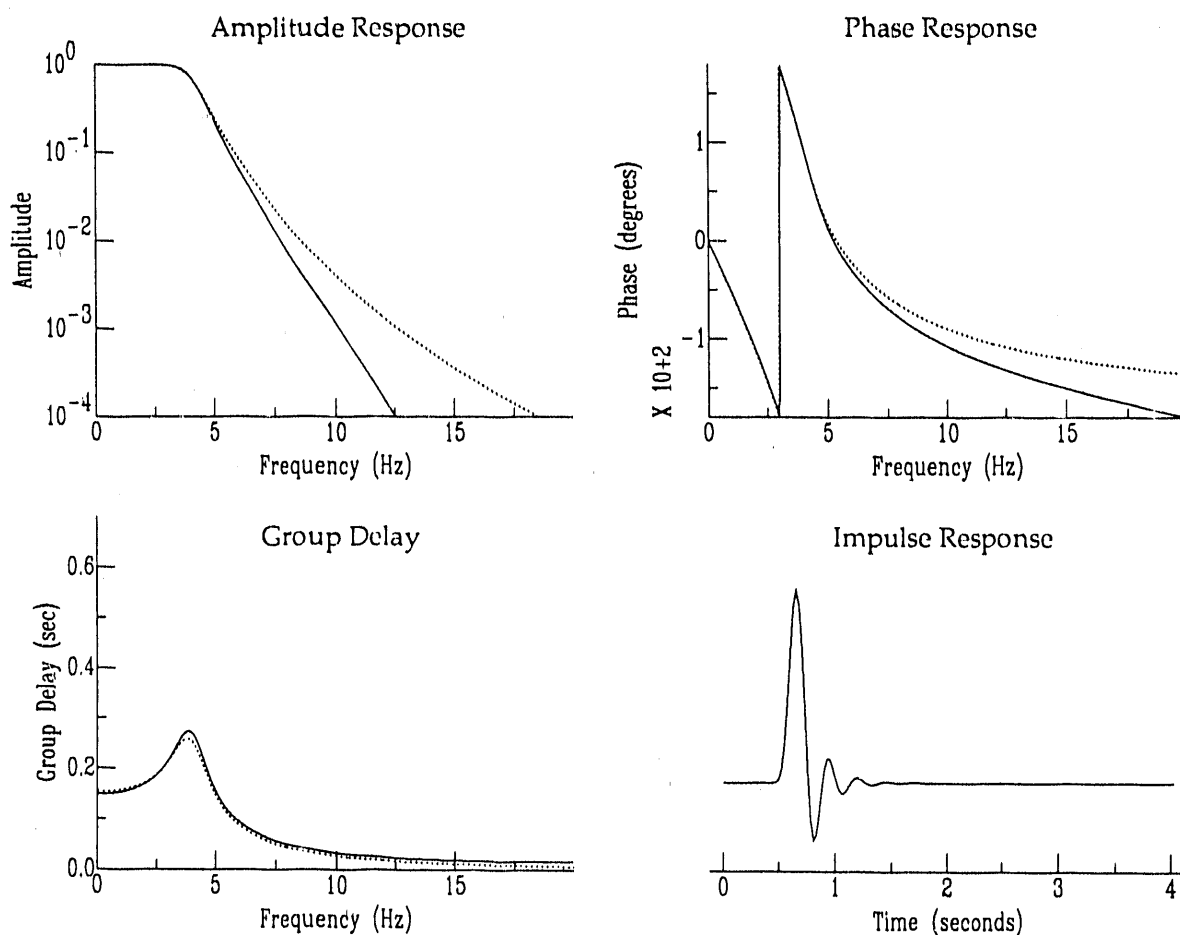


Figure 4 Suite of response curves for 4 Hz, 6-pole Butterworth lowpass filters. Digital filter responses are indicated with solid lines, and prototype analog responses by dashed lines. Differences between the responses are caused by the bilinear transformation. Note the moderately sharp cutoff and moderately peaked group delay of this filter.

attenuation and the passband ripple. The transition bandwidth is usually specified as a fraction of the passband width (the cutoff frequency in a lowpass design). Among the last four parameters, only three may be chosen independently. See [5] for a more detailed explanation of Chebyshev designs.

In XAPIir, the three independent parameters are the number of poles, the transition bandwidth and the stopband attenuation. The parameter specifying the passband ripple is computed from these by the subroutine CHEB-

PARM. It should be emphasized that these parameters specify the passband and stopband edges for the analog prototype Chebyshev filters only. The stopband edge (and, consequently, the transition bandwidth) is altered by the bilinear transformation. If desired, a digital filter with correctly specified stopband edge can be designed by calculating a 'prewarped' analog prototype stopband edge, which would map to the correct digital stopband edge under the bilinear transformation. The necessary transition bandwidth would be determined by the difference between the prewarped value and the passband edge. The function WARP

Bessel Filter Responses

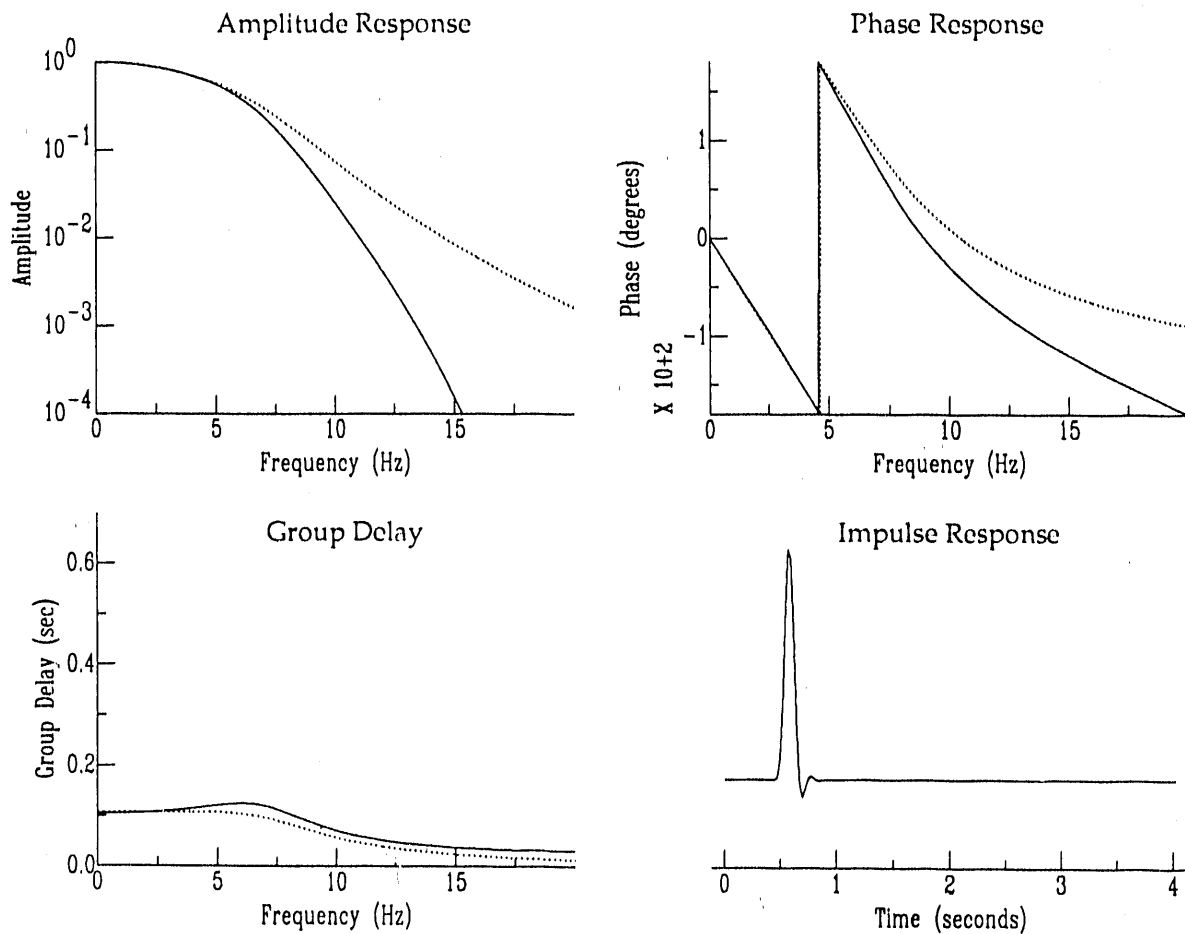


Figure 5 Frequency responses of 6-pole, 4 Hz lowpass digital (solid) and analog (dashed) Bessel filters. Bessel filters are chosen for their flat group delay within the passband. Note the relatively poor amplitude response and the pulse-like impulse response. The bilinear transform slightly spoils the group delay response of the digital filter.

can be used to calculate the prewarped value for the stopband edge.

In the call to `DESIGN` for the Chebyshev filters, the arguments are the same as in calls for `BUTTERWORTH` designs, except that `APPROX` is specified as 'C1' for Chebyshev type I filters and as 'C2' for Chebyshev type II filters, and the stopband attenuation argument `A` and transition bandwidth argument `TRBNDW` must be specified. The attenuation parameter is defined to be the reciprocal of the maximum stopband ripple amplitude. For example, if `A` were specified as 50 (as it is in the examples of

Figures 6 and 7), then the Type I and II stopband amplitudes would not exceed 0.02.

A large stopband attenuation and a narrow transition band are desirable. However, if the first is greatly increased and the second is decreased, the penalty is a large deviation from the desired (flat) response in the passband. Reference [5] has detailed design charts and formulas for determining the necessary order (number of poles) to achieve a desired attenuation, passband ripple and transition bandwidth.

Chebyshev Type I Filter Responses

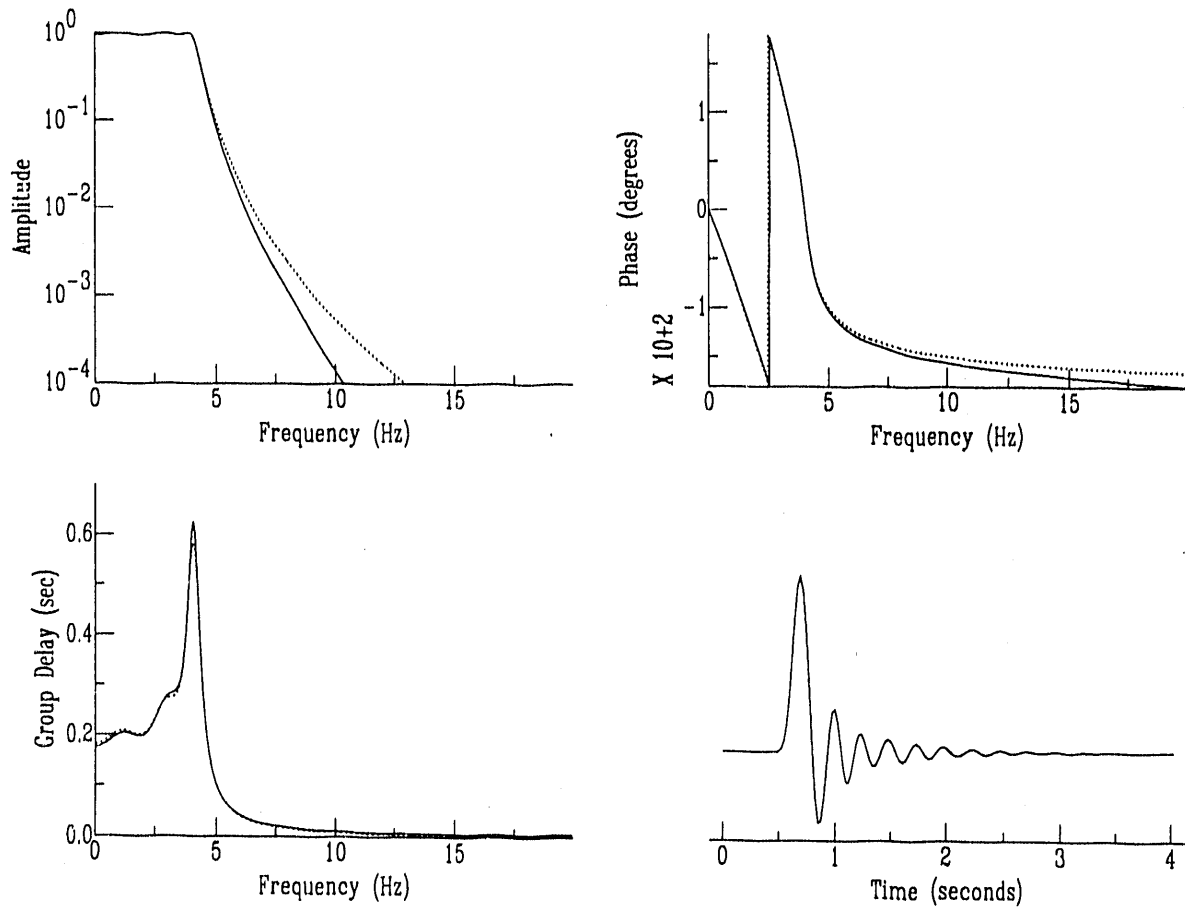


Figure 6 Frequency responses of 6-pole, 4 Hz digital (solid) and analog (dotted) Chebyshev Type I filters. The Chebyshev Type I has the most selective response of all filters implemented in XAPIir, but it also has the most non-linear phase. The group delay exhibits a sharp peak at the passband edge. Consequently, its impulse response rings more than those of the other types of filters. Note that Chebyshev Type I filters has passband ripples.

Chebyshev Type II Filter Responses

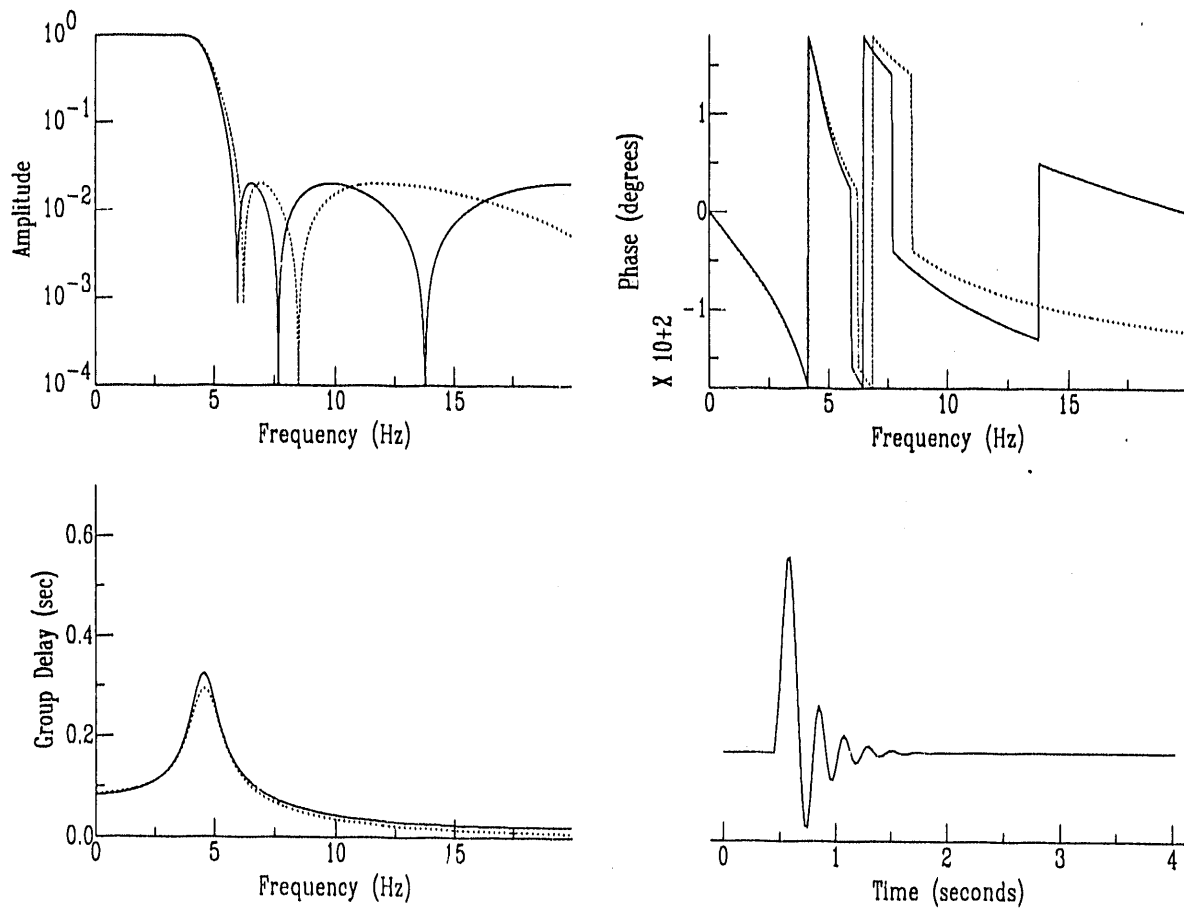


Figure 7 Frequency response curves for 6-pole, 4 Hz lowpass digital (solid) and analog (dotted) Chebyshev Type II filters. This filter exhibits ripples in the stopband. Group delay variations are less extreme than those of the Type I filter. Consequently, the impulse response exhibits less ringing. Chebyshev Type II filters have zeroes as well as poles, which accounts for the notches in the stopband amplitude response.

II. Software Package Organization

XAPIIR is a collection of 18 FORTRAN subroutines and functions. It is organized into 5 subsections as shown in Figure 8. The first subsection is an interface to user calling routines. Three options are supplied for three common filtering applications. The first option is the simplest, satisfied by the single interface routine XAPIIR. This option satisfies the need to design a filter and apply it to an individual waveform. The second option is to design a filter and apply it to a collection of waveforms. In this case, it is faster to design the filter once with a call to the DESIGN subroutine, then apply it to each signal in the collection in the sequence with the APPLY subroutine. Finally, an interface for design purposes is supplied through the INSPECT subroutine, which allows the user to compute digital and analog frequency responses of the specified filters.

The second set of subroutines compute low-pass analog prototype design parameters: poles and zeros. XAPIIR generates lists of poles and zeros for lowpass prototype filters, which are converted to the coefficients of the numerator and denominator of a second-order section representation by the third set of routines. It is a fairly simple matter to add new filter types to the package. It is necessary to create a new subroutine specifying the poles and zeros of the filter in complex FORTRAN arrays, to specify the number of second-order sections in the lowpass prototype, to specify if each second order section consists of a single real pole, a complex-conjugate pair of poles, or complex-conjugate pairs of poles and zeros. The simplest method to add a new filter subroutine is to use the subroutines C1ROOTS and C2ROOTS as templates, as these exercise all pole-zero combinations. The pole-zero lists contain only one complex pole or zero for each

of the complex conjugate pairs.

The pole-zero lists developed by the analog prototype routines are for unit radian cutoff frequency lowpass filters. These are transformed to lowpass, bandpass, bandreject and highpass analog filters with arbitrary frequency cutoffs by the analog spectral transformations [5] of the third set of subroutines (analog-analog transformation in Figure 8). The cutoff frequencies in the arguments to these routines are specified in Hz.

A single analog-digital transformation routine is supplied (BILIN2), which implements the bilinear transformation on the second-order section representation.

The fifth set of routines calculates frequency response curves for digital (DFR) and analog (AFR) filters. These are accessed through the interface routine INSPECT. Magnitude, phase and group delay responses are available. Analog frequency responses may be sampled on a linearly-spaced frequency grid or a logarithmically-spaced frequency grid, and they may be computed over a specified (positive) frequency interval. Digital responses are calculated on a linearly-spaced frequency grid, and only on the interval ranging from d.c. to the folding frequency.

Appendix I contains the calling sequences for all of the routines in XAPIIR, and a description of each of the arguments used in the routines. Output arguments are indicated in *italics*. Arguments that are both input and output arguments are underlined and italicized.

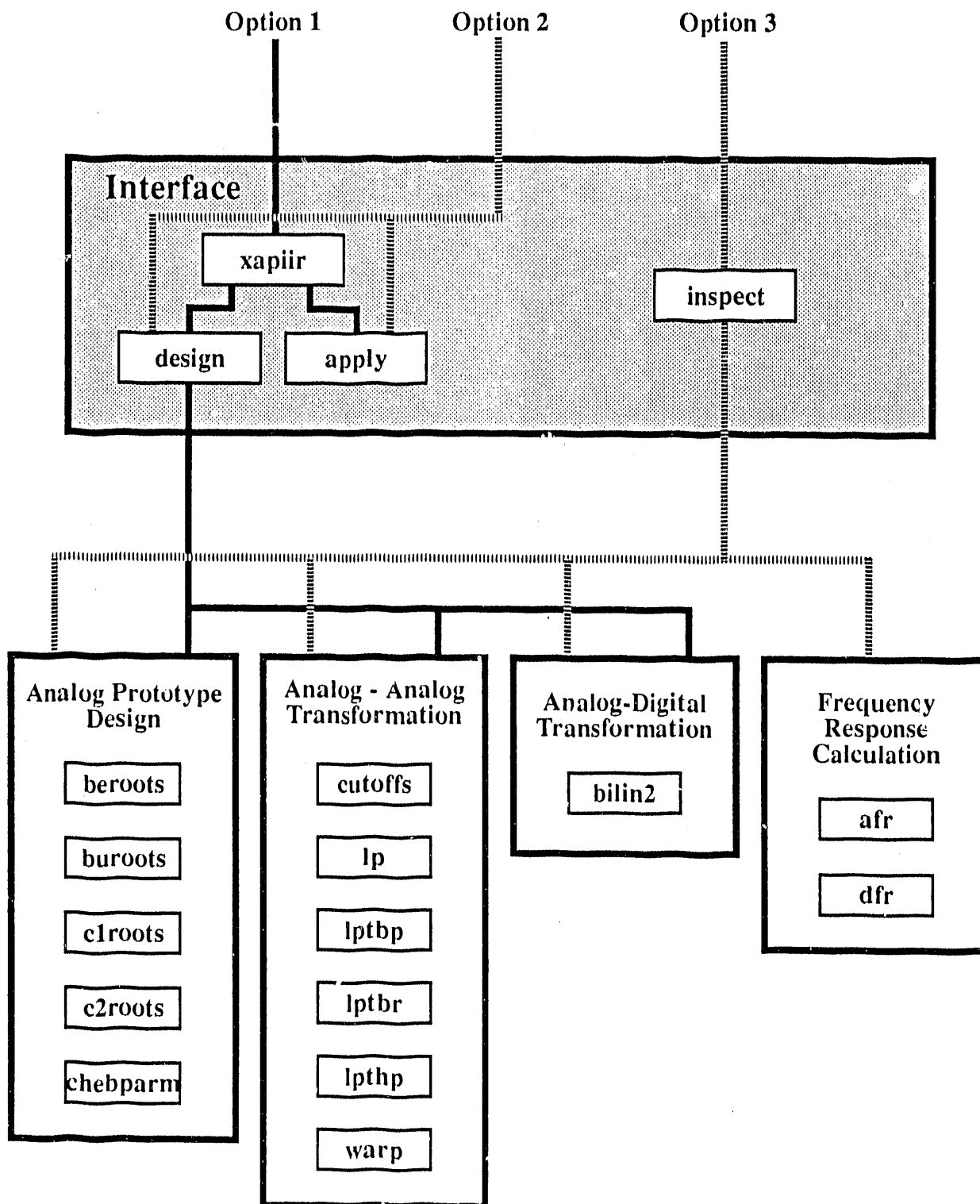


Figure 8 Overall organization of the software package. Three calling options exist. Option 1 is for situations where a single signal is to be filtered. Filter design and application are combined in the single call to subroutine XAPIIR. Option 2 is for situations where the filter is to be designed once and applied to many signals. Direct calls to subroutines DESIGN and APPLY are appropriate in this case. Option 3 is used when only frequency response curves are needed. Subroutine INSPECT is called in this instance.

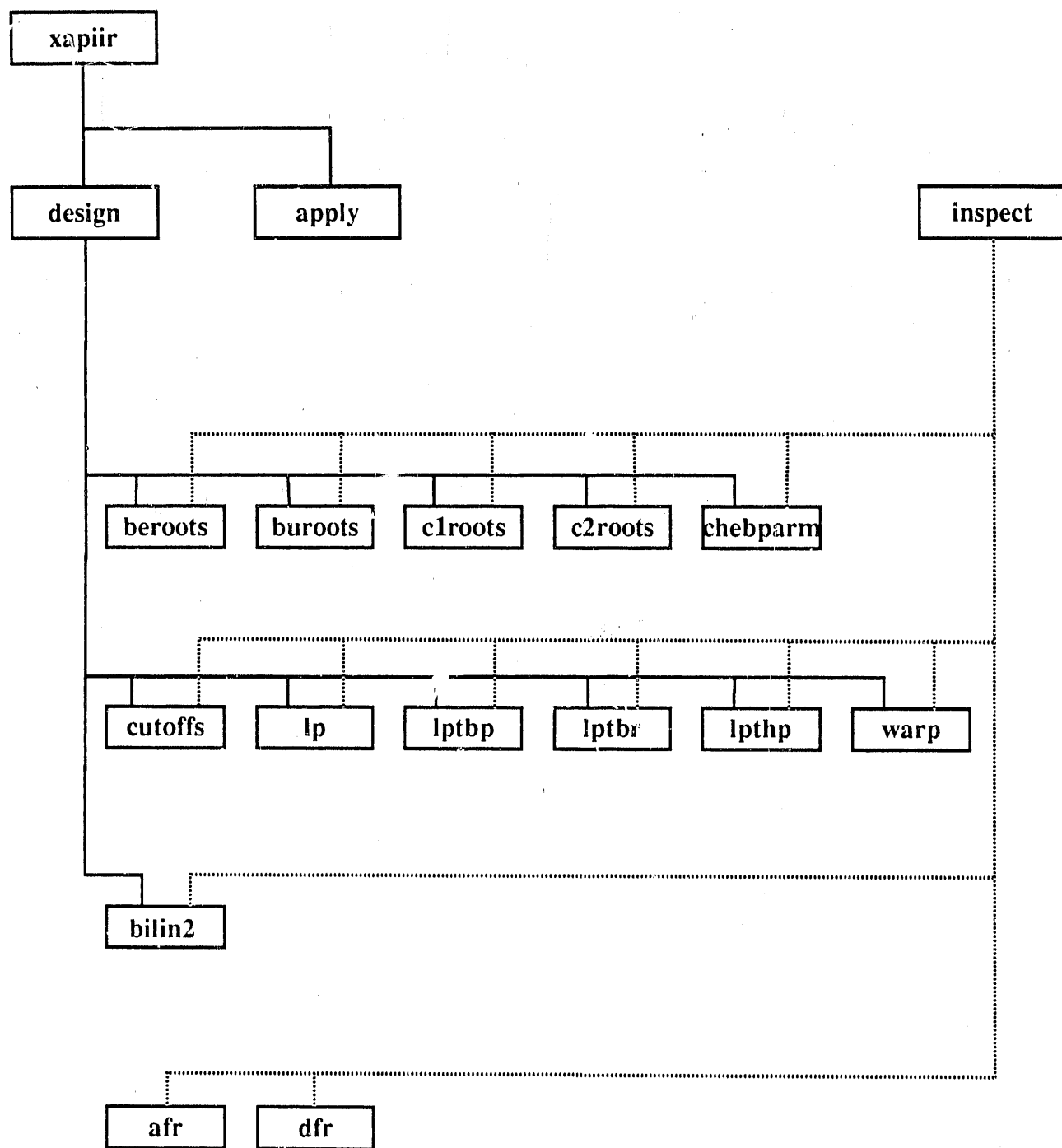


Figure 9 XAPIir structure chart

Appendix I. Calling Sequences and Variable Definitions

Output arguments are italicized. Those arguments that are both input and output arguments are underlined and boldface.

Interface Routines

```
SUBROUTINE APPLY( DATA, NSAMPS, ZP, SN, SD, NSECTS )
SUBROUTINE DESIGN( IORD, TYPE, APROTO, A, TRBNDW,
                   FL, FH, TS, SN, SD, NSECTS )
SUBROUTINE INSPECT( IORD, TYPE, APROTO, A, TRBNDW,
                   FL, FH, TS, RTYPE, NREQS, RFL, RFH,
                   SAMPLING, RESPONSE, FREQS )
SUBROUTINE XAPIIR( DATA, NSAMPS, APROTO, TRBNDW, A, IORD,
                  TYPE, FLO, FHI, TS, PASSES )
```

Arguments:

DATA	REAL*4 array: data sequence to be filtered. Filtering is in-place, destroying original data.
NSAMPS	INTEGER variable: the number of data samples.
ZP	LOGICAL variable: TRUE. for zero-phase (2-pass) filtering, .FALSE. otherwise.
SN	REAL*4 array: numerator coefficients of second-order sections.
SD	REAL*4 array: denominator coefficients of second-order sections.
NSECTS	INTEGER variable: number of second-order sections.
IORD	INTEGER variable: desired filter order (number of poles).
TYPE	CHARACTER*2 variable indicating passband type: 'LP' lowpass 'HP' highpass 'BP' bandpass 'BR' bandreject
APROTO	CHARACTER*2 variable indicating analog prototype: 'BU' Butterworth 'BE' Bessel 'C1' Chebyshev I 'C2' Chebyshev II
A	REAL*4 variable: Chebyshev stopband attenuation factor.
TRBNDW	REAL*4 variable: Chebyshev transition band width as a fraction of the passband width.
FL	REAL*4 variable: low cutoff frequency (Hz).
FH	REAL*4 variable: high cutoff frequency (Hz).
TS	REAL*4 variable: data sampling interval (seconds).
RTYPE	CHARACTER*3 variable indicating type of frequency response desired to be calculated by INSPECT: 'AM' analog amplitude response. 'PH' analog phase response. 'GD' analog group delay response. 'DAM' digital amplitude response. 'DPH' digital response response.

	'DGD' digital group delay response.
NFREQS	INTEGER variable: number of sampling frequencies for frequency response calculation.
RFL	REAL*4 variable: low sampling frequency (Hz).
RFH	REAL*4 variable: high sampling frequency (Hz).
SAMPLING	CHARACTER*8 variable: 'LINEAR' linear frequency spacing 'LOG' logarithmic frequency spacing
RESPONSE	REAL*4 array containing frequency response values.
FREQS	REAL*4 array containing sampling frequencies used only for analog responses. Contains linearly space frequency samples if SAMPLING is 'LINEAR' and logarithmically spaced frequency samples if SAMPLING is 'LOG'
PASSES	REAL*4 variable indicating number of filter passes in XAPIIR. Included to maintain some compatibility with older RECFIL call: 1 single forward pass 2 forward and reverse pass

Analog Prototype Routines

SUBROUTINE BERROOTS(*P*, *RTYPE*, *DCVALUE*, *NSECTS*, *IORD*)
SUBROUTINE BUROOTS(*P*, *RTYPE*, *DCVALUE*, *NSECTS*, *IORD*)
SUBROUTINE C1ROOTS(*P*, *RTYPE*, *DCVALUE*, *NSECTS*, *IORD*, *EPS*)
SUBROUTINE C2ROOTS(*P*, *Z*, *RTYPE*, *DCVALUE*, *NSECTS*, *IORD*, *A*, *OMEGAR*)
SUBROUTINE CHEBPARM(*A*, *TRBNDW*, *IORD*, *EPS*, *RIPPLE*)

Additional arguments:

<i>P</i>	COMPLEX array containing poles of analog prototype filters.
<i>RTYPE</i>	CHARACTER*3 array indicating type of roots in each second-order section: 'SP' single real pole 'CP' complex conjugate pole pair 'CPZ' complex conjugate pole/zero pairs
<i>DCVALUE</i>	REAL*4 variable: zero-frequency amplitude of analog prototype filter used to scale filters properly.
<i>OMEGAR</i>	REAL*4 variable: equal to 1.0 + <i>TRBNDW</i>
<i>EPS</i>	REAL*4 variable: the Chebyshev design parameter related to passband ripple
<i>RIPPLE</i>	REAL*4 variable: Chebyshev passband ripple

Analog-Analog Transformation Routines

SUBROUTINE CUTOFFS(*SN*, *SD*, *NSECTS*, *F*)
SUBROUTINE LP(*P*, *Z*, *TYPE*, *DCVALUE*, *NSECTS*, *SN*, *SD*)
SUBROUTINE LPTBP(*P*, *Z*, *TYPE*, *DCVALUE*, *NSECTS*, *FL*, *FH*, *SN*, *SD*)
SUBROUTINE LPTBR(*P*, *Z*, *TYPE*, *DCVALUE*, *NSECTS*, *FL*, *FH*, *SN*, *SD*)
SUBROUTINE LPTHP(*P*, *Z*, *TYPE*, *DCVALUE*, *NSECTS*, *SN*, *SD*)
REAL FUNCTION WARP(*F*, *TS*)

Additional arguments:

F Analog cutoff frequency for transformation

Analog-Digital Transformation

SUBROUTINE BILIN2(*SN*, *SD*, *NSECTS*)

Frequency Response Calculations

SUBROUTINE AFR(*SN*, *SD*, *NSECTS*, *TYPE*, *SAMPLING*, *FL*, *FH*, *NSAMPS*, *RESPONSE*, *FREQS*)
SUBROUTINE DFR(*SN*, *SD*, *NSECTS*, *TYPE*, *NSAMPS*, *TS*, *RESPONSE*)

Appendix II. Sample Code Calling XAPiir

This FORTRAN program calls XAPiir to create a filter, apply the filter to an impulse sequence to generate a digital impulse response, and then to generate one analog or digital frequency response curve. The response curves are written out in SAC format. This program implements both options 2 and 3 of figure 8.

```
C
C Copyright 1990 Regents of the University of California
C
C
C Author: Dave Harris
C
C Lawrence Livermore National Laboratory
C L-205
C P.O. Box 808
C Livermore, CA 94550
C USA
C
C (415) 423-0617
C
C
C      PROGRAM TEST
C
C      REAL*4 SN(30), SD(30)
C      CHARACTER*2 APROTO, PASSBAND
C      LOGICAL ZP
C      CHARACTER*3 RTYPE
C
C      REAL*4 DATA(1000), FREQS( 501 ), RESPONSE( 501 )
C
C      WRITE(*,*) 'Enter prototype analog filter: (b)utterworth'
C      WRITE(*,*) '(b)essel'
C      WRITE(*,*) '(c1) chebyshev type I'
C      WRITE(*,*) '(c2) chebyshev type II'
C      READ(*,*) APROTO
C      IF ( APROTO .EQ. 'BU' .OR. APROTO .EQ. 'bu' ) THEN
C          APROTO = 'BU'
C      ELSE IF ( APROTO .EQ. 'BE' .OR. APROTO .EQ. 'be' ) THEN
C          APROTO = 'BE'
C      ELSE IF ( APROTO .EQ. 'C1' .OR. APROTO .EQ. 'c1' ) THEN
C          APROTO = 'C1'
C          WRITE(*,*) 'Enter stop band attenuation: '
C          READ(*,*) A
C          WRITE(*,*) 'Enter transition ratio: '
C          READ(*,*) TRBNDW
C      ELSE IF ( APROTO .EQ. 'C2' .OR. APROTO .EQ. 'c2' ) THEN
C          APROTO = 'C2'
C          WRITE(*,*) 'Enter stop band attenuation: '
C          READ(*,*) A
C          WRITE(*,*) 'Enter transition ratio: '
C          READ(*,*) TRBNDW
C      ELSE
C          CALL ERROR('Analog prototype not implemented')
```

END IF

C

```
WRITE(*,*) 'Enter passband type: (lp) lowpass'
WRITE(*,*) ' (hp) highpass'
WRITE(*,*) ' (bp) bandpass'
WRITE(*,*) ' (br) bandreject'
READ(*,*) PASSBAND
IF ( PASSBAND .EQ. 'BP' .OR. PASSBAND .EQ. 'bp' ) THEN
    PASSBAND = 'BP'
    WRITE(*,*) 'Enter low cutoff frequency: '
    READ(*,*) FL
    WRITE(*,*) 'Enter high cutoff frequency: '
    READ(*,*) FH
ELSE IF ( PASSBAND .EQ. 'LP' .OR. PASSBAND .EQ. 'lp' ) THEN
    PASSBAND = 'LP'
    WRITE(*,*) 'Enter high cutoff frequency: '
    READ(*,*) FH
ELSE IF ( PASSBAND .EQ. 'BR' .OR. PASSBAND .EQ. 'br' ) THEN
    PASSBAND = 'BR'
    WRITE(*,*) 'Enter low cutoff frequency: '
    READ(*,*) FL
    WRITE(*,*) 'Enter high cutoff frequency: '
    READ(*,*) FH
ELSE IF ( PASSBAND .EQ. 'HP' .OR. PASSBAND .EQ. 'hp' ) THEN
    PASSBAND = 'HP'
    WRITE(*,*) 'Enter low cutoff frequency: '
    READ(*,*) FL
ELSE
    CALL ERROR('Passband type not implemented')
END IF
```

C

```
WRITE(*,*) 'Enter filter order: '
READ(*,*) IORD
```

C

```
TS = 0.025
CALL DESIGN( IORD, PASSBAND, APROTO, A, TRBNDW,
&           FL, FH, TS, SN, SD, NSECTS )
```

C

```
NSAMPS = 1000
CALL ZERO( DATA, NSAMPS )
DATA(500) = 1.
ZP = .FALSE.
CALL APPLY( DATA, NSAMPS, ZP, SN, SD, NSECTS )
```

C

```
CALL WSAC1( 'impulse_response', DATA, 1000, 0., TS, NERR )
WRITE(*,*) 'Enter response type:'
READ(*,*) RTYPE
CALL INSPECT( IORD, PASSBAND, APROTO, A, TRENDW,
&           FL, FH, TS, RTYPE,
&           501, 0.0, 1./(2.*TS), 'LINEAR ', RESPONSE,
&           FREQS )
```

C

```
IF ( RTYPE(1:1) .EQ. 'D' ) THEN
    CALL WSAC1( RTYPE(1:3), RESPONSE, 501, 0.0, 1./(1000.*TS), NERR)
```

```

        ELSE
            CALL WSAC2( RTYPE(1:2), RESPONSE, 501, FREQS, NERR )
        END IF
C
        CALL EXIT(0)
END
C
C
SUBROUTINE ERROR( MSG )
C
    CHARACTER*(*) MSG
C
    WRITE(*,*) MSG
    CALL EXIT(0)
RETURN
END

```

References

- [1] Harris, D., 1981, RECFIL: Software Package For Recursive Digital Filtering, UCID-19021, Lawrence Livermore National Laboratory, Livermore, CA.
- [2] Tull, J., SAC, 1986, A Signal Processing System For Research Seismology, Proceedings of the 1986 IEEE Symposium on Circuits and Systems, San Jose, CA, May 5-7.
- [3] Lager, D. and S. Azevedo, 1987, SIG: A General Purpose Signal Processing Program, UCRL-95312 revision 1, Lawrence Livermore National Laboratory, Livermore, CA.
- [4] Brase, J., V. Miller and M. Wieting, (1988), The View Signal and Image Processing System, UCID-21368, Lawrence Livermore National Laboratory, Livermore CA.
- [5] Rabiner, L. R., and B. Gold, 1975, **Theory and Application of Digital Signal Processing**, Prentice-Hall, Inc., Chapter 4.

Acknowledgements

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

The author would like to thank Keith Nakanishi of the Treaty Verification Program for supporting this work, and the DOE Office Of Arms Control for funding the work. The author is also indebted to Bill Tapley for testing the software, Elaine Price for typesetting assistance and Amanda Goldner for reviewing the manuscript.

END

DATE FILMED

12 / 13 / 90

