

AD-A103 112

TECHNICAL LIBRARY

AD *A103 112*

TECHNICAL REPORT ARBRL-TR-02346

COMPUTER ALGORITHMS FOR THE DESIGN AND IMPLEMENTATION OF LINEAR PHASE FINITE IMPULSE RESPONSE DIGITAL FILTERS

James N. Walbert

July 1981



US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating
or sponsoring activity is prohibited.

Additional copies of this report may be obtained
from the National Technical Information Service,
U.S. Department of Commerce, Springfield, Virginia
22161.

The findings in this report are not to be construed as
an official Department of the Army position, unless
so designated by other authorized documents.

*The use of trade names or manufacturers' names in this report
does not constitute indorsement of any commercial product.*

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
TECHNICAL REPORT ARBRL-TR-02346		
4. TITLE (and Subtitle) COMPUTER ALGORITHMS FOR THE DESIGN AND IMPLEMENTATION OF LINEAR PHASE FINITE IMPULSE RESPONSE DIGITAL FILTERS		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL REPORT
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James N. Walbert		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Ballistic Research Laboratory ATTN: DRDAR-BLP Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L161102AH80
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Armament Research & Development Command U.S. Army Ballistic Research Laboratory ATTN: DRDAR-BL Aberdeen Proving Ground, MD 21005		12. REPORT DATE JULY 1981
		13. NUMBER OF PAGES 61
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Software	Differentiation	Phase Delay
Digital Filter	Passband	
Linear Phase	Stopband	
Finite Impulse Response	Transition band	
Hilbert Transform	Convolution	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) jmk		
A FORTRAN program, published in the open literature, for the design of linear phase finite impulse response digital filters has been installed on the BRL CDC computer. Portions of this program have been extracted and combined to form a subroutine for filter design. Ancillary subroutines have been developed to assist in the formulation of filter design parameters. A subroutine for convolution of data with digital filters of finite odd length has also been written.		

TABLE OF CONTENTS

	Page
LIST OF ILLUSTRATIONS.	5
I. INTRODUCTION	7
II. A DESCRIPTION OF THE DIGITAL FILTER DESIGN PROGRAM	7
III. A DESCRIPTION OF THE DIGITAL FILTER DESIGN SUBROUTINE. . .	8
IV. CONSIDERATIONS IN THE USE OF THE DESIGN SOFTWARE	8
V. IMPLEMENTATION OF DIGITAL FILTERS.	11
VI. CONCLUSIONS.	15
VII. SUMMARY.	15
VIII. ACKNOWLEDGEMENTS	16
REFERENCES	17
APPENDIX A. A LISTING OF PROGRAM DESIGN	19
APPENDIX B. A LISTING OF SUBROUTINE FILTER.	41
APPENDIX C. SAMPLE OUTPUT FROM PROGRAM DESIGN	53
DISTRIBUTION LIST.	57

LIST OF ILLUSTRATIONS

Figure	Page
1. Frequency response of a low pass filter.	9
2. Frequency response of a low pass filter with narrow transition band.	10
3. Graphical construction of new end points for $\{x_i\}$	13

I. INTRODUCTION

In 1973, McClellan and Parks^{1,2} published a listing of a computer program for the design of finite-duration impulse-response digital filters. This program was unique in that the authors had developed a unified theory for the design of the four types of filters: bandpass, bandstop, Hilbert-transform, and differentiation. The resulting software is one of the most flexible digital filter design programs available.

In the analysis of ballistic data which has been converted from an analog voltage record to a digital time series, it is generally desirable to be able to isolate various signal components for individual study. Such components are usually identifiable by frequency content, and as a consequence, are ideally suited for separation or removal by digital filtering techniques. This report describes the adaptation of the filter design program to the CDC computer at BRL, the modification of a portion of this program into a subroutine, the development of subroutines to specify filter design parameters, and a convolution subroutine for filters of odd length. A complete description of design considerations for digital filters and definitions of related terms is beyond the scope of this report. Any of the cited references will provide the necessary information. This report does provide sufficient design information to allow the reader to implement digital filters; a subsequent BRL Technical Report will cover in greater detail specific application techniques.

II. A DESCRIPTION OF THE DIGITAL FILTER DESIGN PROGRAM

Only minor changes were made to the program as it appeared in reference 2. The program statement added was

```
PROGRAM DESIGN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7)
```

In the original program, when the value of the variable JPUNCH was input as 1, values of the filter coefficients were output to punched cards. In the program, as it exists on the CDC computer, TAPE7 may be specified in the jobstream to be any suitable device or file. The constants PI and PI2 (π and 2π , respectively) were extended to the full double precision word length for the CDC. The free-field input form of the original program was replaced with formatted input. Finally, a test for end-of-file on input was added to allow for multiple designs per computer run. A listing of program DESIGN is in Appendix A.

¹J.H. McClellan, T.W. Parks, "A Unified Approach to the Design of Optimum FIR Linear-Phase Digital Filters," IEEE Trans. Circuit Theory, CT-20(6), 697-701 (1973).

²J.H. McClellan, T.W. Parks, L.R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," IEEE Trans Audio Electroacoustics, AU-21(6), 506-526 (1973).

III. A DESCRIPTION OF THE DIGITAL FILTER DESIGN SUBROUTINE

For most applications to analysis of ballistic data, optimum digital filter design specifications are the result of a systematic trial-and-error investigation. Frequently, the design specifications change from one data event to the next because certain aspects of the experiment were non-repeatable. In view of these factors, it seemed appropriate to formulate a filter design subroutine for use in interactive analysis computer programs, thereby permitting tailoring of the filter design on a round-by-round basis.

Subroutine FILTER, a listing of which appears in Appendix B, is extracted from program DESIGN. It will design bandpass filters of up to 10 bands, but will not design Hilbert Transformers or differentiators. The grid density (LGRID) has been fixed at 16, but the subroutine otherwise retains the full flexibility of program DESIGN. All variable names used in program DESIGN are also retained.

The subroutine statement is

SUBROUTINE FILTER(NFILT,NBANDS,EDGE,FX,WTX,IPRINT,H), where NFILT is the filter length; NBANDS is the number of pass/stop bands; EDGE is an array containing the band edges, expressed as fractions of the sampling frequency; FX is an array containing the desired filter shape, (1. in the pass bands and 0. in the stop bands); WTX is an array containing the desired relative weighting in each band; IPRINT is a control variable for printing the coefficients (0-print coefficients, 1-don't print coefficients); and H is the array containing the filter coefficients on output. The variables NFILT, NBANDS, AND IPRINT are integers; the arrays EDGE,FX,WTX, and H are real, dimensioned $2 \times \text{NBANDS}$, NBANDS, AND $(\text{NFILT}+1)/2$, respectively. If NFILT is even, then the H array is dimensioned NFILT/2.

IV. CONSIDERATIONS IN THE USE OF THE DESIGN SOFTWARE

For the purposes of this discussion, assume that the data sequence x_i^n consists of points equally spaced in time; in particular, Δt will denote the time between two consecutive samples. The sampling frequency, f_s , is therefore $1/\Delta t$, and the bandwidth of the data is $.5f_s$. The bandwidth of the data represents the highest unaliased frequency present in the data, provided due care has been given to the sampling process.

The essence of the design algorithm is to approximate the desired filter response function on the frequency-amplitude plane from $-.5f_s$ to $+.5f_s$ on the frequency axis. The coefficients are designed in a normalized form on the interval $[-.5, .5]$. Moreover, the frequency response has either odd or even symmetry about the origin on the frequency axis, so that the design problem is completely determined by specifying the desired response on the normalized frequency interval $[0., .5]$.

In Figure 1, below, is shown the frequency response of a typical low pass filter. This is a two band filter: it has a pass band and a stop band.

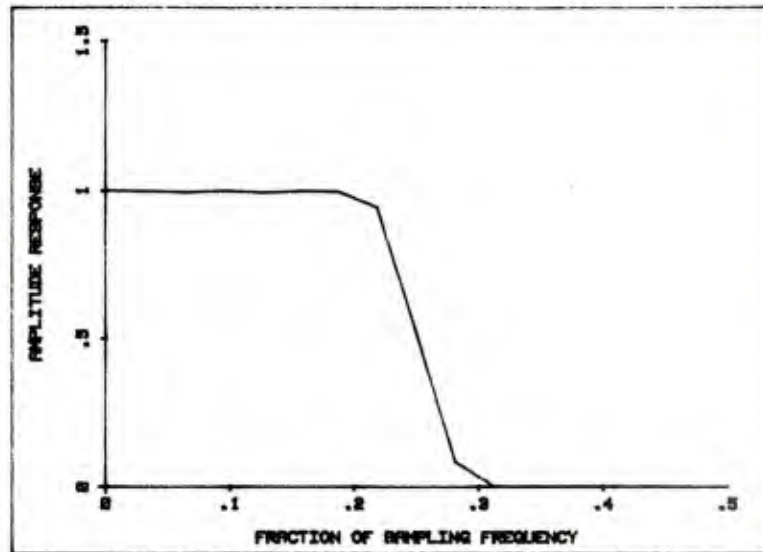


Figure 1. Frequency response of a low pass filter

The pass band is from 0. to .2, or to 40% of the bandwidth. The frequency $.2f_s$ is termed the cutoff frequency of the filter. It is a "pass" band since frequencies in this band are "passed" unaltered (i.e. are multiplied by 1). The stop band is from .3 to .5; frequencies in this band are "stopped" (i.e. multiplied by 0).

The frequency band from .2 to .3 is termed the transition band. Selection of the width of this transition band is somewhat critical in the design of a digital filter, for the following reason: as the transition band narrows, the slope of the frequency response (i.e. the filter roll-off) increases. As this slope increases, the design algorithm compensates by increasing the deviation from the desired response in the pass and stop bands. This deviation is called the "ripple", and results in increases and decreases of amplitude in the signal at those particular frequencies. An example of a filter designed with too narrow a transition band is shown in Figure 2.

In any application software, it is advisable to have the capability of viewing the frequency response of the filter prior to its application, in order to be certain of its characteristics. The design program, as a part of its printed output, lists the normalized frequencies at which the maximum and minimum amplitudes of the ripple occur. Also listed are the deviations from the desired design, which provide a measure of the amplitude error to be expected as a result of applying the filter to the data. (See Appendix C).

Referring to the example of Figure 1, the input variables to design this filter were assigned the following values:

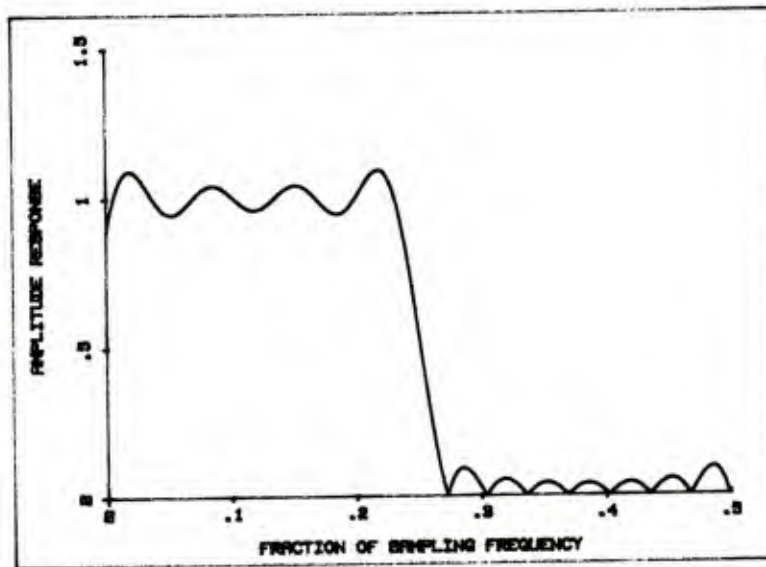


Figure 2. Frequency response of a low pass filter with narrow transition band

NFILT = 33

NBANDS = 2

EDGE(1) = 0

EDGE(2) = .2

EDGE(3) = .3

EDGE(4) = .5

FX(1) = 1.

FX(2) = 0.

WTX(1) = 10.

WTX(2) = 100.

While NFILT is specified as 33, only 17 distinct coefficients are returned, since the design is symmetric about 0. The sample output in Appendix C indicates the ordering of the 17 coefficients, although this is not the filter of Figure 1.

As can be seen in this example, the EDGE array specifies the normalized band edges. The FX array specifies the desired amplitude

response, which is usually (but not necessarily) 1 in the pass bands and 0 in the stop bands. The WTX array specifies a relative scaling of the magnitude of the deviation between the pass band and the stop band. In this example, the design algorithm allows 10 times less deviation in the stop band than in the pass band. This relative weighting may be adjusted arbitrarily to suit a particular need. For example, by relaxing the pass band weighting, say $WTX(1)=1.$, one could design a filter with a more narrow transition band.

For additional information concerning the design of digital filters, the reader is referred to references 1,2, and 3.

V. IMPLEMENTATION OF DIGITAL FILTERS

A digital filter is applied to a data sequence by convoluting the filter weights, or coefficients, with the data points. Specifically, if $\{x_i\}_{i=1}^n$ is a sequence of data points equally spaced in time, and if $\{h_k\}_{k=1}^N$ are the filter coefficients, where $N \leq n$, then the filtered data sequence $\{y_i\}_{i=N}^n$ has values given by

$$y_i = \sum_{k=1}^N h_k x_{i+1-k} \quad (1)$$

One notes that if $i \leq N-1$, then $i+1-k \leq N-k$, so that some subscripts of x in the summation may have values less than 1; we have no corresponding x values. There are two choices: either start the convolution process at $i=N$, or start at $i=1$ and modify k to avoid subscripts of x less than 1 until we get to the N th point. In the first case, $N-1$ data points at the beginning are unused, and the output sequence starts at $i=N$. In the second case, the first $N-1$ output points have not been transformed by the same set of filter coefficients as have the rest of the data; the first $N-1$ points are of questionable value. It will be noted that the same problem occurs for $i > n-N$. In what follows, we will discuss a method to avoid these difficulties. In particular, it will be shown that an n -point input sequence can be modified to provide n useful output points.

For nonreal-time applications, i.e.: for post-processing of data, one is in the admirable position of knowing in advance what is going to occur. That is, the convolution process can be numerically manipulated so as to provide one output point corresponding to each input point, with no lag. (Only filters of finite odd length, say $N = 2M + 1$, $M = 1, 2, \dots, 63$, will be discussed here.) This is accomplished simply by moving the filter coefficients M indices in Eq. (1), so that

$$y_i = \sum_{k=1}^{2M+1} h_k x_{i-M-1+k} \quad (2)$$

Eq. (2) implies that the filter coefficients are centered at the i^{th} data

point. If the coefficients are re-indexed as $\{h'_k\}_{k=-M}^M$, then Eq. (2) is more simply written as

$$y_i = \sum_{k=-M}^M h'_k x_{i-k} , \quad (3)$$

$$\text{where } h'_k = h_{k+M+1} .$$

Now, for the values $i=1,2,\dots,M,n-M+1,n-M+2,\dots,n-1,n$, Eq. (3) still has some values of $i-k$ for which there is no corresponding x .

It is necessary to provide M values at the beginning and M values at the end of the sequence $\{x_i\}_{i=1}^n$. This can be done with a minimum of frequency distortion by using an odd reflection of the first M and last M points. Specifically, for $i-k < 1$, define x_{i-k} by

$$x_{i-k} = 2x_1 - x_{k-i+2} . \quad (4)$$

Similary, for $i-k > n$, define x_{i-k} by

$$x_{i-k} = 2x_n - x_{k-i} . \quad (5)$$

Graphically, Eq. (4) reflects $x_2, x_3, \dots, x_M$ about the vertical line through x_1 and then about the horizontal line through x_1 . The points $x_{n-M+1}, x_{n-M+2}, \dots, x_{n-1}$ are reflected in a like manner about x_n , as shown in Figure 3.

This reflection process can be trivially incorporated into the convolution algorithm, as will be explained below.

The types of digital filters being considered here have an additional property which simplifies the convolution process: they are of either even or odd symmetry about their midpoint. That is,

$$h_k = \pm h_{-k} , \quad k=1,2,\dots,M . \quad (6)$$

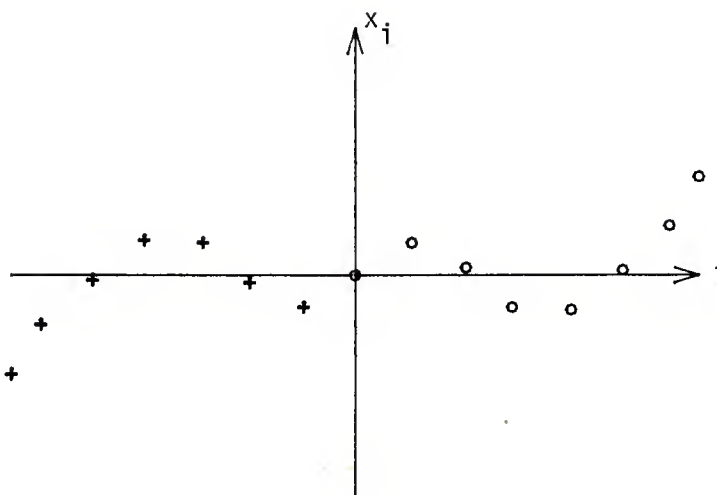


Figure 3. Graphical construction of new end points for x_i

As a consequence, Eq. (3) may be written as

$$y_i = h_0 x_i + 2 \sum_{k=1}^M h_k (x_{i-k} \pm x_{i+k}) \quad (7)$$

Whereas Eq. (3) requires $2M + 1$ multiplications and $2M$ additions to implement, Eq. (7) requires only $M + 1$ multiplications and M additions.

Utilizing Eqs. (4), (5), and (7), the following are three examples of convolution subroutines. The first, in FORTRAN, is in use on the BRL CDC system. The second is in standard BASIC. The third, in an enhanced BASIC, is in use on several BRL systems. In each case, x is the input/output array of length N . The $K=M+1$ filter coefficients are stored in the array H .

Example 1: FORTRAN Convolution Subroutine

```
SUBROUTINE CONVOL (H,K,X,N)
DIMENSION H(K),X(N),S(127),SAVE(63)
M = K-1
L = K + M
IF(L.GT.127) STOP
DO 5 I=1,M
S(I) = 2.*X(1) - X(K -I)
```

```

        S(L+1-I)=X(I)
        SAVE(I) = 2.*X(N) -X(N-I)
5      CONTINUE
        S(K)=X(1)
        LAST=N-M
        DO 20 I=1,N
          X(I) = 0.
          DO 10 J=1,M
            X(I)=X(I)+H(J)*(S(J)+S(L+1-J))
10      CONTINUE
          X(I)=X(I)+H(K)*S(K)
          DO 15 J=2, L
            S(J-1)=S(J)
15      CONTINUE
          IF(I.LE.LAST) S(L)=X(I+K)
          IF(I.GT.LAST) S(L)=SAVE(I-LAST)
20      CONTINUE
        RETURN
      END

```

Example 2: BASIC Convolution Subroutine

```

1000 SUBROUTINE Convolution (H,K,X,N)
1100 DIM H(K),X(N),S(127),SAVE(63)
1200 M=K-1
1300 L=K+M
1400 IF L>127 THEN STOP
1500 FOR I=1 TO M
1600 S(I) = 2.*X(1)-X(K-I)
1700 S(L+1-I)=X(I)
1800 SAVE(I)=2.*X(N)-X(N-I)
1900 NEXT I
2000 S(K)=X(1)
2100 Last=N-M
2200 FOR I=1 TO N
2300 X(I)=0
2400 FOR J=1 TO M
2500 X(I)=X(I)+H(J)*(S(J)+S(L+1-J))
2600 NEXT J
2700 X(I)=X(I)+H(K)*S(K)
2800 FOR J=2 TO L
2900 S(J-1)=S(J)
3000 NEXT J
3100 IF I<=Last THEN S(L)=X(I+K)
3200 If I> Last THEN S(L)=SAVE(I-Last)
3300 NEXT I
3400 RETURN
3500 SUBEND

```

In the third example, use is made of several matrix operations available in enhanced BASIC. The function DOT returns the dot product of the two input arrays. The function MAT REORDER rearranges the elements of one array according to the index order specified by another. In this example, the array B has the values 2,3,4,...,L,1, where $L=2M+1$, the filter length. Implementation of the routine in example 3 represents a decrease in execution time by a factor of 15 over the routine in example 2.

Example 3: BASIC Matrix convolution Subroutine

```

1000 SUBROUTINE Convolution(H,L,X,N,B)
1100 DIM H(L),X(N),S(127), Save(63),B(L)
1200 IF L>127 THEN STOP
1300 REDIM S(L)
1400 M= INT(L/2)
1500 FOR I=1 TO M
1600 S(I)=2.*X(1)-X(M+1-I)
1700 S(L+1-I)=X(I)
1800 Save(I)=2.*X(N)-X(N-I)
1900 NEXT I
2000 S(M+1)=X(1)
2100 Last=N-M
2200 FOR I=1 TO N
2300 X(I)=DOT(H,S)
2400 MAT REORDER S BY B
2500 IF I<=Last THEN S(L)=X(I+M+1)
2600 IF I>Last THEN S(L)=SAVE(I-Last)
2700 NEXT I
2800 RETURN
2900 SUBEND

```

VI. CONCLUSIONS

Digital filters have a wide range of application for numerical analysis of time-series data. The filter design program presented here has been found to be one of the most versatile available. The reflection principle described in this report seems to introduce the least additional frequency content into the data of any of the methods available. This same technique has been used to produce periodic continuation of essentially transient phenomena, facilitating the use of numerical filters in their analysis.

In a forthcoming BRL Technical Report, the author will discuss specific techniques for the application of digital filters to the analysis of ballistic data. The report will also develop in greater detail the theory and applicability of digital filters to analysis of time series.

VII. SUMMARY

An open literature FORTRAN computer program for the design of finite

impulse-response digital filters has been implemented on the BRL CYBER system. Algorithms have been developed and coded for the convolution of digital filters with time series data. These algorithms include a method for the removal of the filter delay, as well as elimination of the loss of data at the beginning and end of the particular data set being filtered.

VIII. ACKNOWLEDGEMENTS

The author is indebted to Mrs. Emma Wineholt, who made the necessary coding changes in program DESIGN and subroutine FILTER to convert them from IBM to CDC FORTRAN.

REFERENCES

1. J.H. McClellan, T.W. Parks, "A Unified Approach to the Design of Optimum FIR Linear-Phase Digital Filters," IEEE Trans. Circuit Theory, CT-20(6), 697-701 (1973).
2. J.H. McClellan, T.W. Parks, L.R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," IEEE Trans Audio Electroacoustics, AU-21(6), 506-526 (1973).

APPENDIX A
A LISTING OF PROGRAM DESIGN

```

C      PROGRAM DESIGN (INPUT,OUTPUT,TAPES=INPUT,TAPES=OUTPUT,TAPE7)
C      PROGRAM FOR THE DESIGN OF LINEAR PHASE FINITE IMPULSE
C      RESPONSE (FIR) FILTERS USING THE REMEZ EXCHANGE ALGORITHM
C      JIM MCCLELLAN, RICE UNIVERSITY, APRIL 13, 1973
C      THREE TYPES OF FILTERS ARE INCLUDED--BANDPASS FILTERS,
C      DIFFERENTIATORS, AND HILBERT TRANSFORM FILTERS
C
C      THE INPUT DATA CONSISTS OF 4 CARDS
C
C      CARD 1--FILTER LENGTH, TYPE OF FILTER, 1-MULTIPLE
C      PASSBAND/STOPBAND, 2-DIFFERENTIATOR, 3-HILBERT TRANSFORM
C      FILTER.  NUMBER OF BANDS, CARD PUNCH DESIRED, AND GRID
C      DENSITY.
C
C      CARD 2--BANDEGES.  LOWER AND UPPER EDGES FOR EACH BAND
C      WITH A MAXIMUM OF 10 BANDS.
C
C      CARD 3--DESIRED FUNCTION (OR DESIRED SLOPE IF A
C      DIFFERENTIATOR) FOR EACH BAND.
C
C      CARD 4--WEIGHT FUNCTION IN EACH BAND.  FOR A
C      DIFFERENTIATOR, THE WEIGHT FUNCTION IS INVERSELY
C      PROPORTIONAL TO F.
C
C      THE FOLLOWING INPUT DATA SPECIFIES A LENGTH 32 BANDPASS
C      FILTER WITH STOPBANDS 0 TO 0.1 AND 0.425 TO 0.5.  AND
C      PASSBAND FROM 0.2 TO 0.35 WITH WEIGHTING OF 10 IN THE
C      STOPBANDS AND 1 IN THE PASSBAND.  THE IMPULSE RESPONSE
C      WILL BE PUNCHED AND THE GRID DENSITY IS 32.  THIS IS THE
C      FILTER IN FIGURES 9 AND 10 IN THE TEXT.
C      SAMPLE INPUT DATA SETUP
C      32,1,2,1,32
C      0,0,1,0,92,0,35,6,425,0,5
C      0,1,0
C      10,1,10

```

```

000100
000110
000120
000130
000140
000150
000160
000170
000180
000190
000200
000210
000220
000230
000240
000250
000260
000270
000280
000290
000300
000310
000320
000330
000340
000350
000360
000370
000380
000390
000400
000410
000420
000430
000440

```

```

C THE FOLLOWING INPUT DATA SPECIFIES A LENGTH 32 WIDEBAND
C DIFFERENTIATOR WITH SLOPE 1 AND WEIGHING OF 1/P. THE
C IMPULSE RESPONSE WILL NOT BE PUNCHED AND THE GRID
C DENSITY IS ASSUMED TO BE 16. THIS IS THE FILTER IN
C FIGURES 17 AND 18 IN THE TEXT.
C 32,2,1,0,0
C 0,6.5
C 1.0
C 1.0
C
C
C COMMON P12,AD,DEV,X,Y,GRID,DES,WT,ALPHA,TEXT,NPONS,NGRID
C DIMENSION TEXT(66),AD(66),ALPHA(66),X(66),Y(66)
C DIMENSION H(66)
C DIMENSION DES(1045),GRID(1045),WT(1045)
C DIMENSION EDGE(20),FX(10),WTX(10),DEVIAT(10)
C DOUBLE PRECISION AD,DEV,X,Y
C DOUBLE PRECISION P12,PI
C P12=6.28318530717958600
C PI=3.14159265358979300
C
C THIS PROGRAM IS SET UP FOR A MAXIMUM LENGTH OF 128, BUT
C THIS UPPER LIMIT CAN BE CHANGE BY REDIMENSIONING THE
C ARRAYS TEXT,AD,ALPHA,X,Y,H TO BE NFMAX/2 + 2.
C THE ARRAYS DES, GRID, AND W1 MUST BE DIMENSIONED
C 16(NFMAX/2+2).
C
C NFMAX=128
C 100 CONTINUE
C JTYPE=0
C
C PROGRAM INPUT SECTION
C
C READ(5,1000) NFILT,JTYPE,NBANDS,JPUNCH,LGRID

```

```

000450
000460
000470
000480
000490
000500
000510
000520
000530
000540
000550
000560
000570
000580
000590
000600
000610
000620
000630
000640
000650
000660
000670
000680
000690
000700
000710
000720
000730
000740
000750
000760
000770
000780
000790

```

```

1000 FORMAT(14,I1,I2,I1,I4)
      IF(EOF(5).NE.0) GOTO 700
      IF(NFILT.GT.NFMAX.OR.NFILT.LT.3) CALL ERROR
      IF(NBANDS.LE.0) NBANDS=1
C
C GRID DENSITY IS ASSUMED TO BE 16 UNLESS SPECIFIED
C OTHERWISE
C
      IF(LGRID.LE.0) LGRID=16
      JB=2*NBANDS
      READ(5,1010) (EDGE(J),J=1,JB)
1010 FORMAT(10F8.0)
      READ(5,1020) (FX(J),J=1,NBANDS)
1020 FORMAT(10F8.0)
      READ(5,1030) (WTX(J),J=1,NBANDS)
1030 FORMAT(10F8.0)
      IF(JTYPE.EQ.0) CALL ERROR
      NEG=1
      IF(JTYPE.EQ.1) NEG=0
      NODU=NFILT/2
      NODU=NFILT-2*NODU
      NFCNS=NFILT/2
      IF(NODD.EQ.1.AND.NEG.EQ.0) NFCNS=NFCNS+1
      IF((LGRID*NFCNS).GT.(16*(NFMAX/2+2))) CALL ERROR
C
C SET UP THE DENSE GRID. THE NUMBER OF POINTS IN THE GRID
C IS (FILTER LENGTH + 1)*GRID DENSITY/2
C
      GRID(1)=EDGE(1)
      DELF=LGRID*NFCNS
      DELF=0.5/DELF
      IF(NEG.EQ.0) GO TO 135
      IF(EDGE(1).LT.DELF) GRID(1)=DELF
135 CONTINUE
      J=1

```

```

001140
001150
001160
001170
001180
001190
001200
001210
001220
001230
001240
001250
001260
001270
001280
001290
001300
001310
001320
001330
001340
001350
001360
001370
001380
001390
001400
001410
001420
001430
001440
001450
001460
001470
001480

      L=1
      LBAND=1
140 FUP=EDGE(L+1)
145 TEMP=GRID(J)
C
C   CALCULATE THE DESIRED MAGNITUDE RESPONSE AND THE WEIGHT
C   FUNCTION OF THE GRID.
C
      DES(J)=EFF(TEMP,FX,WTX,LBAND,JTYPE)
      WT(J)=WATE(TEMP,FX,WTX,LCAND,JTYPE)
      J=J+1
      GRID(J)=TEMP*DELF
      IF(GRID(J).GT.FUP) GO TO 150
      GO TO 145
150 GRID(J-1)=FUP
      DES(J-1)=EFF(FUP,FX,WTX,LBAND,JTYPE)
      WT(J-1)=WATE(FUP,FX,WTX,LCAND,JTYPE)
      LBAND=LBAND+1
      L=L+2
      IF(LBAND.GT.NBANDS) GO TO 160
      GRID(J)=EDGE(L)
      GO TO 140
160 NGRID=J-1
      IF(NEG.NE.NODD) GO TO 165
      IF(GRID(NGRID).GT.(0.5*DELF)) NGRID=GRID-1
165 CONTINUE
C
C   SET UP THE NEW APPROXIMATION PROBLEM WHICH IS EQUIVALENT
C   TO THE ORIGINAL PROBLEM
C
      IF(NFG) 170,170,150
170 IF(NODD.EQ.1) GO TO 200
      DO 175 J=1,NGRID
      CHANGE=DCOS(PI*GRID(J))
      DES(J)=DES(J)/CHANGE

```

```

175 WT(J)=WT(J)*CHANGE
    GO TO 200
180 IF(NODD.EQ.1) GO TO 190
    DO 185 J=1,NGRID
    CHANGE=USIN(PI*GRID(J))
    DES(J)=DES(J)/CHANGE
185 WT(J)=WT(J)*CHANGE
    GO TO 200
190 DO 195 J=1,NGRID
    CHANGE=USIN(PI2*GRID(J))
    DES(J)=DES(J)/CHANGE
195 WT(J)=WT(J)*CHANGE
C
C INITIAL GUESS FOR THE EXTREMAL FREQUENCIES---EQUALLY
C SPACED ALONG THE GRID
C
200 TEMP=FLOAT(NGRID-1)/FLOAT(NFCNS)
    DO 210 J=1,NFCNS
210 IEXT(J)=(J-1)*TEMP+1
    IEXT(NFCNS+1)=NGRID
    NMI=NFCNS-1
    NZ=NFCNS+1
C
C CALL REMEZ EXCHANGE ALGORITHM TO DO THE APPROXIMATION
C PROBLEM
C
    CALL REMEZ(EDGE,NBANDS)
C
C CALCULATE THE IMPULSE RESPONSE
C
    IF(NEG) 300,300,320
300 IF(NODD.EQ.0) GO TO 310
    DO 305 J=1,NM1
305 H(J)=0.5*ALPHA(NZ-J)
    H(NFCNS)=ALPHA(1)

```

```

      GO TO 350
310 H(1)=0.25*ALPHA(NFCNS)
   DO 315 J=2,NM1
315 H(J)=0.25*(ALPHA(NZ-J)+ALPHA(NFCNS+2-J))
   H(NFCNS)=0.5*ALPHA(1)+0.25*ALPHA(2)
      GO TO 350
320 IF(NODD.EQ.0) GO TO 330
   H(1)=0.25*ALPHA(NFCNS)
   H(2)=0.25*ALPHA(NM1)
   DO 325 J=3,NM1
325 H(J)=0.25*(ALPHA(NZ-J)-ALPHA(NFCNS+3-J))
   H(NFCNS)=0.5*ALPHA(1)-0.25*ALPHA(3)
   H(NZ)=0.0
      GO TO 350
330 H(1)=0.25*ALPHA(NFCNS)
   DO 335 J=2,NM1
335 H(J)=0.25*(ALPHA(NZ-J)-ALPHA(NFCNS+2-J))
   H(NFCNS)=0.5*ALPHA(1)-0.25*ALPHA(2)

C
C  PROGRAM OUTPUT SECTION
C
350 PRINT 360
360 FORMAT(1H1, 70(1H*)//25X,'FINITE IMPULSE RESPONSE (FIR)'/
125X,'LINEAR PHASE DIGITAL FILTER DESIGN'/
225X,'REMEZ EXCHANGE ALGORITHM'/)
   IF(JTYPE.EQ.1) PRINT 365
365 FORMAT(25X,'BANDPASS FILTER'/)
   IF(JTYPE.EQ.2) PRINT 370
370 FORMAT(25X,'DIFFERENTIATOR'/)
   IF(JTYPE.EQ.3) PRINT 375
375 FORMAT(25X,'HILBERT TRANSFORMER'/)
   PRINT 378,NFILT
378 FORMAT(15X,'FILTER LENGTH=',I3/)
   PRINT 380
380 FORMAT(15X,'***** IMPULSE RESPONSE *****')

```

```

001840
001850
001860
001870
001880
001890
001900
001910
001920
001930
001940
001950
001960
001970
001980
001990
002000
002010
002020
002030
002040
002050
002060
002070
002080
002090
002100
002110
002120
002130
002140
002150
002160
002170
002180

```

```

DO 381 J=1,NFCNS
K=NFILT+1-J
IF(NEG.EQ.0) PRINT 382,J,H(J),K
IF(NEG.EQ.1) PRINT 383,J,H(J),K
381 CONTINUE
382 FORMAT(20X,'H(',I3,')=',E15.8,' = H(',I4,')')
383 FORMAT(20X,'H(',I3,')=',E15.8,' = -H(',I4,')')
IF(NEG.EQ.1.AND.NODD.EQ.1) PRINT 384,NZ
384 FORMAT(20X,'H(',I3,') = 0.0')
DO 450 K=1,NBANDS,4
KUP=K+3
IF(KUP.GT.NBANDS) KUP=NBANDS
PRINT 385,(J,J=K,KUP)
385 FORMAT(/24X,4('BAND',I3,EX))
PRINT 390,(EDGE(2*J-1),J=K,KUP)
390 FORMAT(2X,'LOWER BAND EDGE',5F15.9)
PRINT 395,(EDGE(2*J),J=K,KUP)
395 FORMAT(2X,'UPPER BAND EDGE',5F15.9)
IF(JTYPE.NE.2) PRINT 400,(FX(J),J=K,KUP)
400 FORMAT(2X,'DESIRED VALUE',2X,5F15.9)
IF(JTYPE.EQ.2) PRINT 405,(FX(J),J=K,KUP)
405 FORMAT(2X,'DESIRED SLOPE',2X,5F15.9)
PRINT 410,(WTX(J),J=K,KUP)
410 FORMAT(2X,'WEIGHTING',6X,5F15.9)
DO 420 J=K,KUP
420 DEVIAT(J)=DEV/WTX(J)
PRINT 425,(DEVIAT(J),J=K,KUP)
425 FORMAT(2X,'DEVIATION',6X,5F15.9)
IF(JTYPE.NE.1) GO TO 450
DO 430 J=K,KUP
430 DEVIAT(J)=20.0*ALOG10(DEVIAT(J))
PRINT 435,(DEVIAT(J),J=K,KUP)
435 FORMAT(2X,'DEVIATION IN DB',5D15.9)
450 CONTINUE
PRINT 455,(GRID(IEX(J)),J=1,NZ)

```

```

002190
002200
002210
002220
002230
002240
002250
002260
002270
002280
002290
002300
002310
002320
002330
002340
002350
002360
002370
002380
002390
002400
002410
002420
002430
002440
002450
002460
002470
002480
002490
002500
002510
002520
002530

```

002740
002750
002760
002770
002780
002790
002800
002810

```
455 FORMAT(/2X,'EXTREMAL FREQUENCIES'/(2X,SF12.7))  
    PRINT 460  
460 FORMAT(/1X,70(1H*)/1H1)  
    IF(JPUNCH.NE.0) WRITE(7,2000) (H(J),J=1,NFCNS)  
2000 FORMAT(5E15.8)  
    IF(NFILT.NE.0) GO TO 100  
700 STOP  
    END
```

```

002620
002630
002640
002650
002660
002670
002680
002690
002700
002710
002720
002730
002740
002750
002760
002770
002780
002790
002800
002810
002820
002830
002840
002850
002860
002870
002880
002890
002900
002910
002920
002930
002940
002950
002960

SUBROUTINE REMEZ(EDGE,NBANDS)

C THIS SUBROUTINE IMPLEMENTS THE REMEZ EXCHANGE ALGORITHM
C FOR THE WEIGHTED CHEBYCHEV APPROXIMATION OF A CONTINUOUS
C FUNCTION WITH A SUM OF COSINES. INPUTS TO THE SUBROUTINE
C ARE A DENSE GRID WHICH REPLACES THE FREQUENCY AXIS, THE
C DESIRED FUNCTION ON THIS GRID, THE WEIGHT FUNCTION ON THE
C GRID, THE NUMBER OF COSINES, AND AN INITIAL GUESS OF THE
C EXTERNAL FREQUENCIES. THE PROGRAM MINIMIZES THE CHEBYCHEV
C ERROR BY DETERMINING THE BEST LOCATION OF THE EXTERNAL
C FREQUENCIES (POINTS OF MAXIMUM ERROR) AND THEN CALCULATES
C THE COEFFICIENTS OF THE BEST APPROXIMATION.
C
COMMON P12,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
DIMENSION EDGE(20)
DIMENSION IEXT(66),AD(66),ALPHA(66),A(66),Y(66)
DIMENSION DES(1045),GRID(1045),WT(1045)
DIMENSION A(66),P(65),Q(65)
DOUBLE PRECISION P12,UNUM,DDEN,DTEMP,A,P,Q
DOUBLE PRECISION AD,DEV,A,Y

C THIS PROGRAM ALLOWS A MAXIMUM NUMBER OF ITERATIONS OF 25
C
C
ITRMAX=25
DEVL=-1.0
NZ=NFCNS+1
NZZ=NFCNS+2
NITER=0
100 CONTINUE
IEXT(NZZ)=NGRID+1
NITER=NITER+1
IF(NITER.GT.ITRMAX) GO TO 400
DO 110 J=1,NZ
DTEMP=GRID(IEXT(J))
DTEMP=DCOS(DTEMP*P12)

```

```

110 X(J)=DTEMP
    JET=(NFCNS-1)/15+1
    DO 120 J=1,NZ
120 AD(J)=D(J,NZ,JET)
    UNUM=0.0
    DDEN=0.0
    K=1
    DO 130 J=1,NZ
    L=TEXT(J)
    DTEMP=AD(J)*DES(L)
    DNUM=UNUM+DTEMP
    DTEMP=K*AD(J)/WT(L)
    DDEN=DDEN+DTEMP
130 K=-K
    DEV=DNUM/DDEN
    NU=1
    IF(DEV.GT.0.0) NU=-1
    DEV=-NU*DEV
    K=NU
    DO 140 J=1,NZ
    L=TEXT(J)
    DTEMP=K*DEV/WT(L)
    Y(J)=DES(L)+DTEMP
140 K=-K
    IF(DEV.DE.DEVL) GO TO 150
    CALL OUCH
    GO TO 400
150 DEVL=DEV
    JCHNGE=0
    K1=TEXT(1)
    KNZ=TEXT(NZ)
    KLOW=0
    NUT=-NU
    J=1

```

C

```

002970
002980
002990
003000
003010
003020
003030
003040
003050
003060
003070
003080
003090
003100
003110
003120
003130
003140
003150
003160
003170
003180
003190
003200
003210
003220
003230
003240
003250
003260
003270
003280
003290
003300
003310

```

```

C SEARCH FOR THE EXTERNAL FREQUENCIES OF THE BEST
C APPROXIMATIONS
C
200 IF (J.EQ.NZZ) YNZ=COMP
   IF (J.GE.NZZ) GO TO 300
   KUP=IEXT(J+1)
   L=IEXT(J)+1
   NUT=-NUT
   IF (J.EQ.2) Y1=COMP
   COMP=DEV
   IF (L.GE.KUP) GO TO 220
   ERR=GEE(L,NZ)
   ERR=(ERR-DES(L))*WT(L)
   DTEMP=NUT*ERR-COMP
   IF (DTEMP.LE.0.0) GO TO 220
   COMP=NUT*ERR
210 L=L+1
   IF (L.GE.KUP) GO TO 215
   ERR=GEE(L,NZ)
   ERR=(ERR-DES(L))*WT(L)
   DTEMP=NUT*ERR-COMP
   IF (DTEMP.LE.0.0) GO TO 215
   COMP=NUT*ERR
   GO TO 210
215 IEXT(J)=L-1
   J=J+1
   KLOW=L-1
   JCHNGE=JCHNGE+1
   GO TO 200
220 L=L-1
225 L=L-1
   IF (L.LE.KLOW) GO TO 250
   ERR=GEE(L,NZ)
   ERR=(ERR-DES(L))*WT(L)
   DTEMP=NUT*ERR-COMP

```

003670
003680
003690
003700
003710
003720
003730
003740
003750
003760
003770
003780
003790
003800
003810
003820
003830
003840
003850
003860
003870
003880
003890
003900
003910
003920
003930
003940
003950
003960
003970
003980
003990
004000
004010

```

IF (UTEMP.GT.0.0) GO TO 230
IF (JCHANGE.LE.0) GO TO 225
GO TO 260
230 COMP=NUT*ERR
235 L=L-1
IF (L.LE.KLOW) GO TO 240
ERR=GEE(L,NZ)
ERR=(ERR-DES(L))*WT(L)
DTEMP=NUT*ERR-COMP
IF (DTEMP.LE.0.0) GO TO 240
COMP=NUT*ERR
GO TO 235
240 KLOW=IEXT(J)
IEXT(J)=L+1
J=J+1
JCHANGE=JCHANGE+1
GO TO 200
250 L=IEXT(J)+1
IF (JCHANGE.GT.0) GO TO 215
255 L=L+1
IF (L.GE.KUP) GO TO 260
ERR=GEE(L,NZ)
ERR=(ERR-DES(L))*WT(L)
DTEMP=NUT*ERR-COMP
IF (DTEMP.LE.0.0) GO TO 255
COMP=NUT*ERR
GO TO 210
260 KLOW=IEXT(J)
J=J+1
GO TO 200
300 IF (J.GT.NZ) GO TO 320
IF (K1.GT.IEXT(1)) K1=IEXT(1)
IF (KNZ.LT.IEXT(NZ)) KNZ=IEXT(NZ)
NUT=NUT
NUT=-NUT

```

```

004020 L=0
004030 KUP=K1
004040 COMP=YNZ*(1.00001)
004050 LUCK=1
004060 310 L=L+1
004070 IF (L.GE.KUP) GO TO 315
004080 ERR=GEE(L,NZ)
004090 ERR=(ERR-DES(L))*WT(L)
004100 DTEMP=NUT*ERR-COMP
004110 IF (DTEMP.LE.0.0) GO TO 310
004120 COMP=NUT*ERR
004130 J=NZZ
004140 GO TO 210
004150 315 LUCK=6
004160 GO TO 325
004170 320 IF (LUCK.GT.9) GO TO 350
004180 IF (COMP.GT.Y1) Y1=COMP
004190 K1=IEXT(NZZ)
004200 325 L=NGRID+1
004210 KLOW=KNZ
004220 NUT=-NUT1
004230 COMP=Y1*(1.00001)
004240 330 L=L-1
004250 IF (L.LE.KLOW) GO TO 340
004260 ERR=GEE(L,NZ)
004270 ERR=(ERR-DES(L))*WT(L)
004280 DTEMP=NUT*ERR-COMP
004290 IF (DTEMP.LE.0.0) GO TO 330
004300 J=NZZ
004310 COMP=NUT*ERR
004320 LUCK=LUCK+10
004330 GO TO 235
004340 340 IF (LUCK.EQ.6) GO TO 370
004350 DO 345 J=1,NFCNS
004360 345 IEXT(NZZ-J)=IEXT(NZ-J)

```

```

004370 IEXT(1)=K1
004380 GO TO 100
004390 350 KN=IEXT(NZZ)
004400 DO 360 J=1,NFCNS
004410 360 IEXT(J)=IEXT(J+1)
004420 IEXT(NZ)=KN
004430 GO TO 100
004440 370 IF(JCHNGE.GT.0) GO TO 100
004450 C
004460 C CALCULATION OF THE COEFFICIENTS OF THE BEST APPROXIMATION
004470 C USING THE INVERSE DISCRETE FOURIER TRANSFORM
004480 C
004490 400 CONTINUE
004500 NM1=NFCNS-1
004510 FSH=1.0E-06
004520 GTEMP=GRID(1)
004530 X(NZZ)=-2.0
004540 CN=2*NFCNS-1
004550 DELF=1.0/CN
004560 L=1
004570 KKK=0
004580 IF (EDGE(1) .EQ. 0. .AND. EDGE(2*NHANDS) .EQ. 0.5 ) KKK=1
004590 IF (NFCNS.LE.3) KKK=1
004600 IF (KKK.EQ.1) GO TO 405
004610 DTEMP=DCOS(P12*GRID(1))
004620 DNUM=UCOS(P12*GRID(NGRID))
004630 AA=2.0/(DTEMP-DNUM)
004640 BB=-(DTEMP+DNUM)/(DTEMP-DNUM)
004650 405 CONTINUE
004660 DO 430 J=1,NFCNS
004670 FT=(J-1)*DELF
004680 XT=DCOS(P12*FT)
004690 IF (KKK.EQ.1) GO TO 410
004700 XT=(XT-BB)/AA
004710 FT=ARCCOS(XT)/P12

```

```

410 XE=X(L)
    IF (XT.GT.XE) GO TO 420
    IF ((XE-XT).LT.FSH) GO TO 415
    L=L+1
    GO TO 410
415 A(J)=Y(L)
    GO TO 425
420 IF ((XT-XE).LT.FSH) GO TO 415
    GRID(1)=FT
    A(J)=SEE(1,NZ)
425 CONTINUE
    IF (L.GT.1) L=L-1
430 CONTINUE
    GRID(1)=GTEMP
    DDEN=P12/CN
    DO 510 J=1,NFCNS
        DTEMP=0.0
        DNUM=(J-1)*DDEN
        IF (NM1.LT.1) GO TO 505
        DO 500 K=1,NM1
            DTEMP=DTEMP+A(K+1)*DCOS(DNUM*K)
500 DTEMP=DTEMP+A(K+1)*DCOS(DNUM*K)
505 DTEMP=2.0*DTEMP+A(1)
510 ALPHA(J)=DTEMP
        DO 550 J=2,NFCNS
            ALPHA(J)=2*ALPHA(J)/CN
            ALPHA(1)=ALPHA(1)/CN
            IF (KKK.EQ.1) GO TO 545
            P(1)=2.0*ALPHA(NFCNS)*BB+ALPHA(NM1)
            P(2)=2.0*AA*ALPHA(NFCNS)
            Q(1)=ALPHA(NFCNS-2)-ALPHA(NFCNS)
            DO 540 J=2,NM1
                IF (J.LT.NM1) GO TO 515
                AA=0.5*AA
                BB=0.5*BB
515 CONTINUE

```

```

004720
004730
004740
004750
004760
004770
004780
004790
004800
004810
004820
004830
004840
004850
004860
004870
004880
004890
004900
004910
004920
004930
004940
004950
004960
004970
004980
004990
005000
005010
005020
005030
005040
005050
005060

```

```

P(J+1)=0.0
DO 520 K=1,J
  A(K)=P(K)
520 P(K)=2.0*HB*A(K)
  P(2)=P(2)+A(1)*2.0*AA
  JM1=J-1
DO 525 K=1,JM1
525 P(K)=P(K)+Q(K)+AA*A(K+1)
  JP1=J+1
DO 530 K=3,JP1
530 P(K)=P(K)+AA*A(K-1)
  IF (J.EQ.NM1) GO TO 540
DO 535 K=1,J
535 Q(K)=-A(K)
  Q(1)=Q(1)+ALPHA(NFCNS-1-J)
540 CONTINUE
DO 543 J=1,NFCNS
543 ALPHA(J)=P(J)
545 CONTINUE
  IF (NFCNS.GT.3) RETURN
  ALPHA(NFCNS+1)=0.0
  ALPHA(NFCNS+2)=0.0
  RETURN
END

```

```

005070
005080
005090
005100
005110
005120
005130
005140
005150
005160
005170
005180
005190
005200
005210
005220
005230
005240
005250
005260
005270
005280
005290
005300

```

005310
 005320
 005330
 005340
 005350
 005360
 005370
 005380
 005390
 005400
 005410
 005420
 005430
 005440
 005450

```

C      FUNCTION WATE(TEMP,FX,WTX,LBAND,JTYPE)
C
C      FUNCTION TO CALCULATE THE WEIGHT FUNCTION AS A FUNCTION
C      OF FREQUENCY.
C
      DIMENSION FX(5),WTX(5)
      IF(JTYPE.EQ.2) GO TO 1
      WATE=WTX(LBAND)
      RETURN
1     IF(FX(LBAND).LT.0.0001) GO TO 2
      WATE=WTX(LBAND)/TEMP
      RETURN
2     WATE=WTX(LBAND)
      RETURN
      END

```

005460
005470
005480
005490
005500

```
SUBROUTINE ERROR  
  PRINT 1  
  1 FORMAT(0 ***** ERROR IN INPUT DATA ***** 0)  
  STOP  
  END
```

005510
005520
005530
005540
005550
005560
005570
005580
005590
005600
005610
005620

```
C      FUNCTION EFF(TEMP,FX,WTX,LBAND,JTYPE)
C
C      FUNCTION TO CALCULATE THE DESIRED RESPONSE MAGNITUDE
C      AS A FUNCTION OF FREQUENCY.
C
      DIMENSION FX(5),WTX(5)
      IF(JTYPE.EQ.2) GO TO 1
      EFF=FX(LBAND)
      RETURN
1 EFF=FX(LBAND)*TEMP
      RETURN
      END
```

```

SUBROUTINE OUCH
PRINT 1
1 FORMAT(' ***** FAILURE TO CONVERGE *****')/
1,0PROBABLE CAUSE IS MACHINE ROUNDING ERROR'/
2,0THE IMPULSE RESPONSE MAY BE CORRECT'/
3,0CHECK WITH A FREQUENCY RESPONSE')
RETURN
END
DOUBLE PRECISION FUNCTION GEE(K,N)
C
C FUNCTION TO EVALUATE THE FREQUENCY RESPONSE USING THE
C LAGRANGE INTERPOLATION FORMULA IN THE BARYCENTRIC FORM
C
COMMON P12,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
DIMENSION DES(1045),GRID(1045),WT(1045)
DOUBLE PRECISION P,C,D,XF
DOUBLE PRECISION P12
DOUBLE PRECISION AD,DEV,X,Y
P=0.0
XF=GRID(K)
XF=DCOS(P12*XF)
D=0.0
DO 1 J=1,N
C=XF-X(J)
C=AD(J)/C
D=D+C
1 P=P+C*Y(J)
GEE=P/D
RETURN
END

```

```

005630
005640
005650
005660
005670
005680
005690
005700
005710
005720
005730
005740
005750
005760
005770
005780
005790
005800
005810
005820
005830
005840
005850
005860
005870
005880
005890
005900
005910
005920
005930

```


APPENDIX B
A LISTING OF SUBROUTINE FILTER

```

000100
000110
000120
000130
000140
000150
000160
000170
000180
000190
000200
000210
000220
000230
000240
000250
000260
000270
000280
000290
000300
000310
000320
000330
000340
000350
000360
000370
000380
000390
000400
000410
000420
000430
000440

SUBROUTINE FILTER(NFILT,NBANDS,EDGE,FX,WTX,IPRINT,H)
*****
SUBROUTINE FILTER - PGMR. JAMES N. WALBERT, NOVEMBER 1974
DESIGNS A DIGITAL FILTER OF UP TO 10 BANDS AND OF MAXIMUM LENGTH 127.
THE SUBROUTINE USES THE REMEZ EXCHANGE ALGORITHM TO FIND THE BEST
APPROXIMATION WHICH MINIMIZES CHEBYSHEV ERROR.
USAGE -
CALL FILTER(NFILT,NBANDS,EDGE,FX,WTX,IPRINT,H)
DESCRIPTION OF VARIABLES -
NFILT - FILTER LENGTH. MUST BE AN ODD NUMBER BETWEEN 3
AND 127, INCLUSIVE.
NBANDS - NUMBER OF PASS-STOP BANDS. MAXIMUM IS 10.
EDGE - ARRAY DEFINING BAND EDGES. DIMENSION IN CALLING
PROGRAM MUST BE 2*NBANDS. EDGE IS DEFINED BY
FREQUENCY/SAMPLING RATE. EDGE(1)=0.,
EDGE(2*NBANDS)=.5.
FX - DESIRED FUNCTION. FX IS 1. IN PASSBANDS AND 0. IN
STOPBANDS.
WTX - WEIGHTING FACTOR. USUALLY 10. IN PASSBANDS AND
100. IN STOPBANDS.
IPRINT - CONTROL VARIABLE. IF IPRINT=0, COEFFICIENTS ARE
PRINTED. IF IPRINT=1, COEFFICIENTS ARE NOT PRINTED.
H - FILTER COEFFICIENT ARRAY, NUMBERED FROM 1 TO
NFILT/2+1. THE ARRAY IS SYMMETRIC ABOUT H(NFILT/2+1).
THE H ARRAY SHOULD HAVE DIMENSION NFILT IN THE
CALLING PROGRAM.
FUNCTION SUBPROGRAMS CALLED -

```

```

C      DOUBLE PRECISION FUNCTION GEE
000450
C
000460
C
000470
C
000480
C
000490
C *****
000500
C
000510
000520
000530
000540
000550
000560
000570
000580
000590
000600
000610
000620
000630
000640
000650
000660
000670
000680
000690
000700
000710
000720
000730
000740
000750
000760
000770
000780
000790

      DIMENSION IEXT(66),AD(66),ALPHA(66),A(66),Y(66)
      DIMENSION DEVIAT(10),AMP(66),FRE(66),FXA(20)
      COMMON /XFRBLK/ DES(1045),GRID(1045),WT(1045),A(66)
      DIMENSION EDGE(1),FX(1),WTX(1),H(1)
      DOUBLE PRECISION AD,DEV,X,Y,PI2,PI
      DOUBLE PRECISION UNUM,DLEN,DTEMP,A
      PI2=6.283185307179586
      PI=3.141592653589793
      NFCNS=NFILT/2+1
      GRID(1)=EDGE(1)
      DELF=16*NFCNS
      DELF=0.5/DELF
      J=1
      L=1
      LHAND=1
      FUP=EDGE(L+1)
      TEMP=GRID(J)
      DES(J)=FX(LHAND)
      WT(J)=WTX(LHAND)
      J=J+1
      GRID(J)=TEMP+DELF
      IF(GRID(J).GT.FUP) GO TO 150
      GO TO 145
      GRID(J-1)=FUP
      DES(J-1)=FX(LHAND)
      WT(J-1)=WTX(LHAND)
      LHAND=LHAND+1
      L=L+2

```

```

160 IF(LBAND.GT.NBANDS) GO TO 160
200 GRID(J)=EDGE(L)
    GO TO 140
    NGRID=J-1
    TEMP=FLOAT(NGRID-1)/FLOAT(NFCNS)
210 DO 210 J=1,NFCNS
    IEXT(J)=(J-1)*TEMP+1
    IEXT(NFCNS+1)=NGRID
    NM1=NFCNS-1
    NZ=NFCNS+1
    DEVL=-1.
    NZZ=NFCNS+2
    NITER=0
1000 CONTINUE
    IEXT(NZZ)=NGRID+1
    NITER=NITER+1
    IF(NITER.GT.25) GO TO 4000
    DO 1100 J=1,NZ
    DTEMP=GRID(IEXT(J))
    DTEMP=DCOS(DTEMP*PI2)
    X(J)=DTEMP
    JET=(NFCNS-1)/15+1
    DO 1200 J=1,NZ
    D=1.
    DO 1193 LL=1,JET
    DO 1192 KK=LL,NZ,JET
1191 IF(KK-J)1191,1192,1191
    D=2.0*D*(X(J)-X(KK))
1192 CONTINUE
1193 CONTINUE
1200 AD(J)=1.0/D
    DNUM=0.0
    DDEN=0.0
    K=1
    DO 1300 J=1,NZ

```

```

000800
000810
000820
000830
000840
000850
000860
000870
000880
000890
000900
000910
000920
000930
000940
000950
000960
000970
000980
000990
001000
001010
001020
001030
001040
001050
001060
001070
001080
001090
001100
001110
001120
001130
001140

```

```

1300      L=IEXT(J)
          DTEMP=AD(J)*DES(L)
          DNUM=DNUM+DTEMP
          DTEMP=K*AD(J)/WT(L)
          DDEN=DDEN+DTEMP
          K=-K
          DEV=DNUM/DDEN
          NU=1
          IF(DEV.GT.0.0) NU=-1
          DEV=-NU*DEV
          K=NU
          DO 1400 J=1,NZ
            L=IEXT(J)
            DTEMP=K*DEV/WT(L)
            Y(J)=DES(L)+DTEMP
            K=-K
          1400      IF(DEV.GE.DEVL) GO TO 1500
                    PRINT 1401
                    FORMAT(1H0,***** FAILURE TO CONVERGE *****/,O RESPONSE MAY BE OK.)
                    GO TO 4000
          1500      DEVL=DEV
                    JCHNGE=0
                    K1=IEXT(1)
                    KNZ=IEXT(NZ)
                    KLOW=0
                    NUT=-NU
                    J=1
          2000      IF(J.EQ.NZ) YNZ=COMP
                    IF(J.GE.NZ) GO TO 3000
                    KUP=IEXT(J+1)
                    L=IEXT(J)+1
                    NUT=-NUT
                    IF(J.EQ.2) Y1=COMP
                    COMP=DEV
                    IF(L.GE.KUP) GO TO 2200

```

```

001150
001160
001170
001180
001190
001200
001210
001220
001230
001240
001250
001260
001270
001280
001290
001300
001310
001320
001330
001340
001350
001360
001370
001380
001390
001400
001410
001420
001430
001440
001450
001460
001470
001480
001490

```

001500
001510
001520
001530
001540
001550
001560
001570
001580
001590
001600
001610
001620
001630
001640
001650
001660
001670
001680
001690
001700
001710
001720
001730
001740
001750
001760
001770
001780
001790
001800
001810
001820
001830
001840

```

ERR=GEE(NZ,GRID(L),X,AD,Y)
ERR=(ERR-DES(L))*WT(L)
DTEMP=NUT*ERR-COMP
IF(UTEMP.LE.0.0) GO TO 2200
COMP=NUT*ERR
2100 L=L+1
IF(L.GE.KUP) GO TO 2150
ERR=GEE(NZ,GRID(L),X,AD,Y)
ERR=(ERR-DES(L))*WT(L)
DTEMP=NUT*ERR-COMP
IF(UTEMP.LE.0.0) GO TO 2150
COMP=NUT*ERR
GO TO 2100
2150 IEXT(J)=L-1
J=J+1
KLOW=L-1
JCHNGE=JCHNGE+1
GO TO 2000
2200 L=L-1
2250 L=L-1
IF(L.LE.KLOW) GO TO 2500
ERR=GEE(NZ,GRID(L),X,AD,Y)
ERR=(ERR-DES(L))*WT(L)
DTEMP=NUT*ERR-COMP
IF(UTEMP.GT.0.0) GO TO 2300
IF(JCHNGE.LE.0) GO TO 2250
GO TO 2600
2300 COMP=NUT*ERR
2350 L=L-1
IF(L.LE.KLOW) GO TO 2400
ERR=GEE(NZ,GRID(L),X,AD,Y)
ERR=(ERR-DES(L))*WT(L)
DTEMP=NUT*ERR-COMP
IF(UTEMP.LE.0.0) GO TO 2400
COMP=NUT*ERR

```

```

2400      GO TO 2350
          KLOW=IEXT(J)
          IEXT(J)=L+1
          J=J+1
          JCHNGE=JCHNGE+1
          GO TO 2000
2500      L=IEXT(J)+1
          IF(JCHNGE.GT.0) GO TO 2150
2550      L=L+1
          IF(L.GE.KUP) GO TO 2600
          ERR=GEE(NZ,GRID(L),X,AD,Y)
          ERR=(ERR-DES(L))*WT(L)
          DTEMP=NUT*ERR-COMP
          IF(DTEMP.LE.0.0) GO TO 2550
          COMP=NUT*ERR
          GO TO 2100
2600      KLOW=IEXT(J)
          J=J+1
          GO TO 2000
3000      IF(J.GT.NZ) GO TO 3200
          IF(K1.GT.IEXT(1)) K1=IEXT(1)
          IF(KN2.LT.IEXT(NZ)) KN2=IEXT(NZ)
          NUT1=NUT
          NUT=-NUT
          L=0
          KUP=K1
          COMP=YNZ*(1.00001)
          LUCK=1
          L=L+1
3100      IF(L.GE.KUP) GO TO 3150
          ERR=GEE(NZ,GRID(L),X,AD,Y)
          ERR=(ERR-DES(L))*WT(L)
          DTEMP=NUT*ERR-COMP
          IF(DTEMP.LE.0.0) GO TO 3100
          COMP=NUT*ERR

```

```

001850
001860
001870
001880
001890
001900
001910
001920
001930
001940
001950
001960
001970
001980
001990
002000
002010
002020
002030
002040
002050
002060
002070
002080
002090
002100
002110
002120
002130
002140
002150
002160
002170
002180
002190

```

3150	J=NZZ	002200
	GO TO 2100	002210
	LUCK=6	002220
	GO TO 3250	002230
3200	IF (LUCK.GT.9) GO TO 3500	002240
	IF (COMP.GT.Y1) Y1=COMP	002250
	K1=IEXT(NZZ)	002260
3250	L=NGRID+1	002270
	KLOW=KNZ	002280
	NUT=-NUT1	002290
	COMP=Y1*(1.00001)	002300
3300	L=L-1	002310
	IF (L.LE.KLOW) GO TO 3400	002320
	ERR=GEE(NZ,GRID(L),X,AD,Y)	002330
	ERR=(ERR-DES(L))*WT(L)	002340
	DTEMP=NUT*ERR-COMP	002350
	IF (DTEMP.LE.0.) GO TO 3300	002360
	J=NZZ	002370
	COMP=NUT*ERR	002380
	LUCK=LUCK+10	002390
	GO TO 2350	002400
3400	IF (LUCK.EQ.6) GO TO 3700	002410
	DO 3450 J=1,NFCNS	002420
3450	IEXT(NZZ-J)=IEXT(NZ-J)	002430
	IEXT(1)=K1	002440
	GO TO 1000	002450
3500	KN=IEXT(NZZ)	002460
	DO 3600 J=1,NFCNS	002470
3600	IEXT(J)=IEXT(J+1)	002480
	IEXT(NZ)=KN	002490
	GO TO 1000	002500
3700	IF (JCHANGE.GT.0) GO TO 1000	002510
4000	CONTINUE	002520
	NM1=NFCNS-1	002530
	FSH=1.0E-06	002540

```

X(NZZ)=-2.0
CN=2*NFCNS-1
DELF=1.0/CN
L=1
DO 4300 J=1,NFCNS
FT=(J-1)*DELF
XT=DCOS(PI2*FT)
XE=X(L)
IF(XT.GT.XE) GO TO 4200
IF((XE-XT).LT.FSH) GO TO 4150
L=L+1
GO TO 4100
A(J)=Y(L)
GO TO 4250
IF((XT-XE).LT.FSH) GO TO 4150
A(J)=GEE(NZ,FT,X,AD,Y)
CONTINUE
IF(L.GT.1) L=L-1
CONTINUE
DDEN=PI2/CN
DO 5100 J=1,NFCNS
DTEMP=0.
DNUM=(J-1)*DDEN
IF(NM1.LT.1) GO TO 5050
DO 5000 K=1,NM1
DTEMP=DTEMP+A(K+1)*DCOS(DNUM*FLOAT(K))
DTEMP=2.0*DTEMP+A(1)
ALPHA(J)=DTEMP
DO 5500 J=2,NFCNS
ALPHA(J)=2*ALPHA(J)/CN
ALPHA(1)=ALPHA(1)/CN
IF(NFCNS.GT.3) GO TO 304
ALPHA(NFCNS+1)=0.
ALPHA(NFCNS+2)=0.
CONTINUE
304

```

```

002550
002560
002570
002580
002590
002600
002610
002620
002630
002640
002650
002660
002670
002680
002690
002700
002710
002720
002730
002740
002750
002760
002770
002780
002790
002800
002810
002820
002830
002840
002850
002860
002870
002880
002890

```

```

002900      DO 305 J=1,NM1
002910      H(J)=0.5*ALPHA(NZ-J)
002920      H(NFCNS)=ALPHA(1)
002930      IF(IPRINT.EQ.1) GO TO 700
002940      PRINT 360
002950      360 FORMAT(1H1, 70(1H*)//25X,'FINITE IMPULSE RESPONSE (FIR)')/
002960      1 25X, 'LINEAR PHASE DIGITAL FILTER DESIGN'/
002970      2 25X, 'REMEZ EXCHANGE ALGORITHM'//
002980      PRINT 365
002990      365 FORMAT(25X, 'BANDPASS FILTER')/
003000      PRINT 378, NFILT
003010      378 FORMAT(15X, 'FILTER LENGTH = ',I5//)
003020      PRINT 380
003030      380 FORMAT(15X,'***** IMPULSE RESPONSE *****')
003040      DO 381 J=1,NFCNS
003050      K=NFILT+1-J
003060      PRINT 382,J,H(J),K
003070      CONTINUE
003080      381
003090      382 FORMAT(20X,'H(',I3,')=' ,E15.8,' = H(',I4,')')
003100      DO 450 K=1,NBANDS,4
003110      KUP=K+3
003120      IF (KUP.GT.NBANDS) KUP=NBANDS
003130      PRINT 385, (J,J=K,KUP)
003140      385 FORMAT(/24X,4('BAND',I3,8X))
003150      PRINT 390, (EDGE(2*J-1),J=K,KUP)
003160      390 FORMAT(2X,'LOWER BAND EDGE',5F15.9)
003170      PRINT 395, (EDGE(2*J),J=K,KUP)
003180      395 FORMAT(2X,'UPPER BAND EDGE',5F15.9)
003190      PRINT 400, (FX(J),J=K,KUP)
003200      400 FORMAT(2X,'DESIRED VALUE',2X,5F15.9)
003210      PRINT 410, (WTX(J),J=K,KUP)
003220      410 FORMAT(2X,'WEIGHTING',5X,5F15.9)
003230      DO 420 J=K,KUP
003240      420 DEVIAT(J)=DEV/WTX(J)
      PRINT 425, (DEVIAT(J),J=K,KUP)

```

```

425 FORMAT(2X,'DEVIATION',6X,5F15.9)
DO 430 J=K,KUP
430 DEVIAT(J)=20.0*ALOG10(DEVIAT(J))
PRINT 435, (DEVIAT(J),J=K,KUP)
435 FORMAT(2X,'DEVIATION IN DB',5F15.9)
450 CONTINUE
PI2=PI2
DO 452 J=1,NZ
AMP(J)=H(NFCNS)
FRE(J)=GRID(TEXT(J))
DO 451 NN=1,NM1
AMP(J)=AMP(J)+2.*H(NM1-NN+1)*COS(FRE(J)*PI2*FLOAT(NN))
451 CONTINUE
452 CONTINUE
PRINT 455, (FRE(J),J=1,NZ)
455 FORMAT(2X,'EXTREMAL FREQUENCIES',/(2X,8F12.7))
PRINT 456, (AMP(J),J=1,NZ)
456 FORMAT(2X,'MAGNITUDE OF FREQUENCY RESPONSES',/(2X,8F12.7))
PRINT 460
460 FORMAT(/1X, 70(1H*)/1H1)
NPT=2*NBANDS
DO 470 J=1,NPT
FXA(J)=FX((J+1)/2)
470 CONTINUE
CALL PLIDTA(FRE,AMP,NZ,EDGE,FXA,NPT)
700 RETURN
END

```

```

003250
003260
003270
003280
003290
003300
003310
003320
003330
003340
003350
003360
003370
003380
003390
003400
003410
003420
003430
003440
003450
003460
003470
003480
003490
003500
003510

```

```

C
C
GEE.001      06-NOV-79
DOUBLE PRECISION FUNCTION GEE(N,BLIP,X,AD,Y)
DIMENSION X(1),Y(1),AD(1)
DOUBLE PRECISION P12,X,Y,AD
P12=6.283185307179586
P=0.
XF=BLIP
XF=DCOS(P12*XF)
D=0.
DO 1 J=1,N
O=XF-X(J)
O=AD(J)/O
D=D+O
P=P+O*Y(J)
GEE=P/D
RETURN
END

```

```

003520
003530
003540
003550
003560
003570
003580
003590
003600
003610
003620
003630
003640
003650
003660
003670
003680
003690

```

1

APPENDIX C
SAMPLE OUTPUT FROM PROGRAM DESIGN

FINITE IMPULSE RESPONSE (FIR)
LINEAR PHASE DIGITAL FILTER DESIGN
REMEZ EXCHANGE ALGORITHM

BANDPASS FILTER

FILTER LENGTH= 33

**** IMPULSE RESPONSE ****

H(1)=	-.43343124E-02	=	H(33)
H(2)=	.28107133E-01	=	H(32)
H(3)=	.36576607E-01	=	H(31)
H(4)=	.48062215E-01	=	H(30)
H(5)=	.52151146E-01	=	H(29)
H(6)=	.44314948E-01	=	H(28)
H(7)=	.24178276E-01	=	H(27)
H(8)=	-.39524089E-02	=	H(26)
H(9)=	-.31592790E-01	=	H(25)
H(10)=	-.48276263E-01	=	H(24)
H(11)=	-.44888714E-01	=	H(23)
H(12)=	-.17057197E-01	=	H(22)
H(13)=	.32779728E-01	=	H(21)
H(14)=	.95199035E-01	=	H(20)
H(15)=	.15595394E+00	=	H(19)
H(16)=	.20005638E+00	=	H(18)
H(17)=	.21618581E+00	=	H(17)

	BAND 1	BAND 2	BAND
LOWER BAND EDGE	0.000000000	.120000000	
UPPER BAND EDGE	.100000000	.500000000	
DESIRED VALUE	1.000000000	0.000000000	
WEIGHTING	10.000000000	100.000000000	
DEVIATION	.350741255	.035074126	
DEVIATION IN DB-	.9100262960+01-	.2910026300+02	

EXTREMAL FREQUENCIES			
0.0000000	.0367647	.0753676	.1000000
.1310294	.1549265	.1825000	.2137500
.2762500	.3075000	.3387500	.3716382
.4361765	.4674265	.5000000	
			.1200000
			.2450000
			.4030882

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
12	Commander Defense Technical Info Center ATTN: DDC-DDA Cameron Station Alexandria, VA 22314	9	Commander US Army Armament Research and Development Command ATTN: DRDAR-LC, T. Moore W. Williver K. Reuben H. Fair S. Bernstein G. Demitrack W. Benson B. Knutulski G. Bubb
1	Director Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209		
1	Director Defense Nuclear Agency Arlington, VA 22209		Dover, NJ 07801
1	Commander US Army BMD Advanced Technology Center ATTN: BMDATC-M P. O. Box 1500 Huntsville, AL 35807	2	Commander US Army Armament Research and Development Command ATTN: DRDAR-SC, B. Shulman Mr. Webster Dover, NJ 07801
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCMDM-ST 5001 Eisenhower Avenue Alexandria, VA 22333	2	Commander US Army Armament Research and Development Command ATTN: DRDAR-SE Dover, NJ 07801
2	Commander US Army Armament Research and Development Command ATTN: DRDAR-TSS Dover, NJ 07801	1	Commander US Army Armament Research and Development Command ATTN: DRDAR-FU Dover, NJ 07801
1	Commander US Army Armament Research and Development Command ATTN: L. Goldsmith Dover, NJ 07801	2	Commander US Army Armament Research and Development Command ATTN: DRDAR-DP Dover, NJ 07801
		2	Commander US Army Armament Research and Development Command ATTN: DRDAR-QA Dover, NJ 07801

DISTRIBUTION LIST

<u>No. of</u> <u>Copies</u>	<u>Organization</u>	<u>No. of</u> <u>Copies</u>	<u>Organization</u>
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299	1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
1	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB-TL Watervliet, NY 12189	1	Commander US Army Harry Diamond Labs 2800 Powder Mill Road Adelphi, MD 20783
2	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB, T. Simkins T. Davidson Watervliet, NY 12189	1	Commander US Army Missile Command ATTN: DRSMI-R Redstone Arsenal, AL 35809
1	Commander US Army Aviation Research and Development Command ATTN: DRDAV-E 4300 Goodfellow Blvd. St. Louis, MO 63120	1	Commander US Army Missile Command ATTN: DRSMI-RBL Redstone Arsenal, AL 35809
2	Director US Army Research and Technology Laboratories (AVRADCOM) Ames Research Center Moffett Field, CA 94035	1	Commander US Army Missile Command ATTN: DRSMI-YDL Redstone Arsenal, AL 35809
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035	2	Commander US Army Mobility Equipment Research & Development Command Fort Belvoir, VA 22060
1	Commander US Army Communications Research and Development Command ATTN: DRDCO-PPA-SA Fort Monmouth, NJ 07703	2	Commander US Army Tank Automotive Research and Development Command ATTN: DRDTA-UL Technical Director Warren, MI 48090
		2	Project Manager Division Air Defense Gun ATTN: DRCPM-ADG Dover, NJ 07801

DISTRIBUTION LIST

<u>No. of</u> <u>Copies</u>	<u>Organization</u>	<u>No. of</u> <u>Copies</u>	<u>Organization</u>
2	Project Manager Cannon Artillery Weapons System ATTN: DRCPM-CAWS Dover, NJ 07801	1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL, Tech Lib White Sands Missile Range NM 88002
2	Project Manager Nuclear Munitions ATTN: DRCPM-NUC Dover, NJ 07801	1	Commander Naval Air Systems Command Washington, DC 20360
2	Project Manager Tank Main Armament Systems ATTN: DRCPM-TMA Dover, NJ 07801	2	Commander Naval Ordnance Systems Command Washington, DC 20360
1	Product Manager for 30mm Ammo. ATTN: DRCPM-AAH-30mm Dover, NJ 07801	1	Commander Naval Sea Systems Command Washington, DC 20362
2	Product Manager M110E2 Weapon System, DARCM ATTN: DRCPM-M110E2 Rock Island, IL 61299	2	Commander Naval Ship Research and Development Command Bethesda, MD 20084
3	Commander US Army Research Office P.O. Box 12211 ATTN: Technical Director Engineering Division Metallurgy & Materials Division Research Triangle Park NC 27709	1	Commander Naval Air Development Center, Johnsville Warminster, PA 18974
1	Commander US Army Research Office ATTN: Dr. J. Chandra Research Triangle Park NC 27709	1	Commander Naval Missile Center Point Mugu, CA 95041
		2	Commander Naval Surface Weapons Center Dahlgren, VA 22448
		2	Commander Naval Surface Weapons Center Silver Spring, MD 20910
4	Director US Army Mechanics and Materials Research Center ATTN: Director (3 cys) DRXMR-ATL (1 cy) Watertown, MA 02172	2	Commander Naval Weapons Center China Lake, CA 93555

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Commander Naval Research Laboratory Washington, DC 20375	2	Director National Aeronautics and Space Administration Langley Research Center Hampton, VA 23365
1	Superintendent Naval Postgraduate School ATTN: Dir of Lib Monterey, CA 93940	1	Director National Aeronautics and Space Administration Manned Spacecraft Center ATTN: Lib Houston, TX 77058
1	Commander Naval Ordnance Station Indian Head, MD 20640	1	BLM Applied Mechanics Consultants ATTN: Dr. A. Boresi 3310 Willett Drive Laramie, WY 82070
2	AFRPL ATTN: W. Andrepont T. Park Edwards AFB, CA 93523	1	CALSPAN Corp. ATTN: E. Fisher P. O. Box 400 Buffalo, NY 14225
2	AFATL Eglin AFB, FL 32542	1	S&D Dynamics, Inc. ATTN: Dr. M. Soifer 755 New York Avenue Huntington, NY 11743
2	AFWL Kirkland AFB NM 87117	1	Southwest Research Institute ATTN: P. Cox 8500 Culebra Road San Antonio, TX 78228
2	ASD Wright-Patterson AFB OH 45433	1	Stanford University Stanford Linear Accelerator Center ATTN: Eric Grosse, Numerical Analysis Consultant ALAC, P. O. Box 4349 Stanford, CA 94305
1	Director Lawrence Livermore Laboratory Livermore, CA 94550	1	Towson State University Department of Mathematics ATTN: Miss Margaret Zipp Towson, MD 21204
1	Director Los Alamos Scientific Laboratory Los Alamos, NM 87544		
1	Headquarters National Aeronautics and Space Administration Washington, DC 20546		

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	University of Delaware Department of Mathematics Department of Mechanical Engr. Newark, DE 19711		<u>Aberdeen Proving Ground</u>
		2	Dir, USAMSAA ATTN: DRXSY-D DRXSY-MP, H. Cohen
1	University of Illinois Department of Mathematics ATTN: Dr. Evelyn Frank Urbana, IL 61801	2	Dir, MTD ATTN: H. King P. Paules
		1	Cdr, USATECOM ATTN: DRSTE-TO-F
1	University of Kentucky Department of Computer Science ATTN: Prof. H. C. Thacher, Jr. 915 Patterson Office Tower Lexington, KY 40506	6	Cdr/Dir USACSL, EA ATTN: DRDAR-CL DRDAR-CLB DRDAR-CLB-PA DRDAR-CLD DRDAR-CLN DRDAR-CLY
3	University of Wisconsin-Madison Mathematics Research Center ATTN: Dr. John Nohel Dr. Carl de Boer Dr. George Box 610 Walnut Street Madison, WI 53706		
1	Virginia Commonwealth Univ. Department of Math. Sciences ATTN: Mr. V. Benokraitis 901 W. Franklin Richmond, VA 23284		
1	Virginia Polytechnic Institute Dept. of Aerospace Engineering ATTN: Prof. G. Inger Blacksburg, VA 24061		

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number _____

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) _____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) _____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: _____

Telephone Number: _____

Organization Address: _____
