

AD-A066 851

NAVAL UNDERWATER SYSTEMS CENTER NEW LONDON CONN NEW --ETC F/G 17/1
A PROGRAM TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL COMPUTER--ETC(U)
OCT 70 S R VAN DER VEEN

UNCLASSIFIED

NUSC/NL-TM-2211-309-70

NL

1 OF
ADA
066851



END
DATE
FILED
6-79
DDC

000980

ADA0 66851

DDC FILE COPY

LEVEL II

Most Project -3

Copy 52

Code _____

NUSC/NL Problem No.

A-407-00-00
SF 11 552 002-12862

(1) SC

OOVI LIBRARY COPY

(9) Technical memo,

NEW LONDON LABORATORY
NAVAL UNDERWATER SYSTEMS CENTER
NEW LONDON, CONNECTICUT 06320

(6)

A PROGRAM TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL COMPUTER.

(14)

by Steven R. van der Veen

NUSC/NL Technical Memorandum No. 2211-309-70

TM-2211-309-70
1 October 1979

(11)

INTRODUCTION

(16)

F11552

(17)

SF11552002

Filtering of underwater acoustic data is an operation which is typically performed in an analog manner. This operation is limited in flexibility due to the quantity and availability of hardware and limited in accuracy because of component tolerances, aging and environmental deterioration. It is the purpose of this memorandum to discuss the processing of broad band acoustic data using digital computer filtering on digitized data. This procedure is limited by the need for sampling rates several times the maximum frequency present in the signal, but bypasses the above limitations by allowing the simulation of a large number of filters. The accuracy is improved and further computer processing can be applied to the results.

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

ADMINISTRATIVE INFORMATION

This memorandum was prepared under NUSC/NL Project Title: Forward Scattering Studies For Sonar Design and Performance Prediction, S. R. Santaniello, Principal Investigator. The sponsoring activity was Naval Ships Systems Command, Code OOVI, J. Reeves, Program Manager.

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the New London Laboratory, Naval Underwater Systems Center.

405918

LB

000980
SF

ACCESSION NO.	White Section	Black Section	DISTRIBUTION/AVAILABILITY CODES <i>ALL INFORMATION IS UNCLASSIFIED</i>	Date Acq. and/or SPECIAL
	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
HTIS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	A	
DOC	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
UNANNOUNCED				
NOTIFICATION				

NUSC/NL Tech Memo
No. 2211-309-70

While the computer programs are presently implemented to simulate only Butterworth type low pass (LP) and band pass (BP) filter, they are general enough to allow expansion to include any filter which is representable in terms of a finite pole-zero transfer function configuration. The formulation presented here produces, under mild constraints, digital filters with the same impulse response as the equivalent analog filter.

THEORY

Using z-transform notation (ref a and b), a time

$$x(kT) \quad , \quad k=0,1,2\dots$$

can be expressed as a function of a complex variable z by the transform:

$$(1) \quad X(z) = \sum_{k=0}^{\infty} x(kT) \cdot z^{-k}$$

A linear filtering operation, such as that represented by the finite difference equation:

$$\sum_{i=0}^N a_i \cdot y(k-i) = \sum_{j=0}^M b_j \cdot x(k-j) \quad , \quad k=0,1,2\dots$$

can be represented in z-transform notation by the equation:

$$(1a) Y(z) = H(z) \cdot X(z), \text{ where } H(z) = \frac{\sum_{j=0}^M b_j z^{-j}}{\sum_{i=0}^N a_i z^{-i}}$$

This equation is analogous to the continuous-time filtering operation represented in Laplace transform notation as:

$$(2) \quad z(s) = H(s) \cdot x(s)$$

Letting the z-transform of a sampled time function $x(t)$ be denoted $X^*(z)$ and letting the sampling operator on a Laplace transform be denoted by $(x(s))^*$ (so that $(X(s))^* = X^*(z)$), we can examine the conditions that are sufficient to cause the sampled output of a filter to be equal to the output of the equivalent digital filter.

We will assume that $X^*(z)$ and $H^*(z)$ are equal to $X(z)$ and $H(z)$. That is, the z-transforms used in evaluating the output of the digital filter were derived by sampling the input to and impulse response of the analog filter, or by an equivalent method.

The relationship between $Z(s)$ and $Z^*(z)$ is derived in two forms by Freeman(a). They are:

$$(3) \quad Z^*(z) = \sum_{\text{residues of } \frac{Z(s)}{s^T - 1} \text{ at the poles of } Z(s)} \text{ residues of } \frac{Z(s)}{s^T - 1}$$

and where T^{-1} is the sampling frequency

$$(4) \quad Z^*(s) = \frac{1}{T} \sum_{l=-\infty}^{\infty} Z\left(s + \frac{2\pi i l}{T}\right), \quad l = \sqrt{-1}, \quad z = e^{\frac{sT}{2}}$$

Applying the sampling operator to eq.(2), we have:

$$Z^*(z) = (H(s) \cdot X(s))^*$$

Substituting from eq.(4), we get:

$$(4a) \quad Z^*(z) = \frac{1}{T} \sum_{l=-\infty}^{\infty} \left[H\left(s + \frac{2\pi i l}{T}\right) \cdot X\left(s + \frac{2\pi i l}{T}\right) \right]$$

If both X and H are bandlimited (i.e. if $X(s)$ and $H(s)$ are zero for all s such that $|s| > \frac{1}{T}$), then: (4a) reduces to:

$$\begin{aligned} Z^*(z) &= T \left(\frac{1}{T} \sum_{l=-\infty}^{\infty} H\left(s + \frac{2\pi i l}{T}\right) \right) \cdot \left(\frac{1}{T} \sum_{l=-\infty}^{\infty} X\left(s + \frac{2\pi i l}{T}\right) \right) \\ &= T \cdot H^*(z) \cdot X^*(z) = T \cdot H(z) \cdot X(z) \end{aligned}$$

Thus, if both X and H are bandlimited to less than half the sampling frequency, then the sampled output of the analog filter will be equal to the output of the digital filter amplified by the sampling period.

NUMERICAL METHOD

Many types of filters can be represented in terms of the pole-zero configuration of their transfer functions in the complex frequency plane.

Their transfer functions can then be written:

$$(5) \quad H(s) = K \left[\frac{\prod_{i=1}^M (s - r_i)}{\prod_{j=1}^N (s - p_j)} \right]$$

Using eq.(3) reduces (5) to:

$$H^*(z) = K \sum_{j=1}^N \frac{\prod_{i=1}^M (\bar{r}_j - r_i)}{(1 - e^{p_j T} z^{-1}) \prod_{l \neq j}^N (\bar{r}_j - \bar{r}_l)}$$

in the z-domain. Letting:

$$A_j = e^{p_j T} \quad \text{and} \quad K_j = K \frac{\prod_{i=1}^M (\bar{r}_j - r_i)}{\prod_{l \neq j}^N (\bar{r}_j - \bar{r}_l)}$$

we have a sum of 1st order terms:

$$H(z) = \sum_{j=1}^N \frac{K_j}{(1 - A_j z^{-1})}$$

The K_j and A_j will either be real or in complex conjugate pairs. Each conjugate pair can be combined to yield second order terms of the type:

$$\frac{C_{j1} - C_{j2} \cdot z^{-1}}{1 - B_{j1} \cdot z^{-1} + B_{j2} \cdot z^{-2}}$$

Thus, $H(z)$ can be represented as the sum of first and second order terms of the type seen in (1a), representing a difference equation in z-transform notation.

The second order terms are thus realized by the equations:

$$y_j(k) - B_{j1} \cdot y_j(k-1) + B_{j2} \cdot y_j(k-2) = C_{j1} \cdot x(k) - C_{j2} \cdot x(k-1) , k=0,1,2\dots$$

for as many second order terms as there are ($j=1,2,\dots$), while the first order terms are realized by the equations:

$$y_j(k) - B_j \cdot y_j(k-1) = C_j \cdot x(k) , k=0,1,2\dots , B_j = A_j , C_j = K$$

again for as many j as there are first order terms.

The outputs $y_j(k)$ are then summed to give the filter output $y(k)$ (for each k). Note that this is justified by the linearity of the z -transformation.

The filter constructed above is called a recursive numerical filter (RNF), where the term "recursive" refers to the fact that past outputs contribute algebraically to the present output.

PROGRAM INFORMATION

The computations above have been implemented in a series of subroutines that are called by the driver subroutine BTRRNF. BTRRNF computes the coefficients of a specified filter and passes them to a routine which actually performs filtering operations.

BTRRNF is called by the FORTRAN statement:

```
CALL BTRRNF(NPOLES,ITYPE,CF,BW,T,IFERR)
```

VARIABLE	PURPOSE	TYPE	INPUT/OUTPUT
NPOLES	The number of poles in LP version, where NPOLES<10	INTEGER	Input
ITYPE	A switch which determines whether a LP or BP filter will be generated. If ITYPE=0, then LP, If ITYPE=1, then BP. Note that shifting from LP to BP doubles the actual number of poles (see below)	INTEGER	Input
CF	The center frequency in Hz	REAL	Input
BW	The bandwidth in Hz	REAL	Input
T	The sampling period in secs	REAL	Input

IFERR	An error indicator If NPOLES .GT. 10, an error exists, IFERR returns as 0. If all goes well, IFERR returns as 1	INTEGER	Output
-------	--	---------	--------

TABLE 1

The filter is specified in the call, and if IFERR is returned as 1, the filter is loaded in the filtering subroutine.

If the filter type is BP (ITYPE=1), a low pass filter with the appropriate bandwidth is set up and then bandshifted to produce the BP filter by substituting $(CF) /s' + s'/(cf)$ for s in the complex frequency plane. This is the operation which doubles the number of poles as mentioned in table 1. Note also that the center frequency is the geometric mean of the 3 dB points, not the algebraic mean.

As mentioned above, a call to BTRRNF computes the coefficients of the appropriate filter and passes them to a subroutine called RNF. This is the filtering routine and may be called by using the FORTRAN statement:

CALL RNF(X,Y,N)

VARIABLE	PURPOSE	TYPE	INPUT/OUTPUT
X	The input array containing the data to be filtered	REAL array dimension X(N)	Input
Y	The output array to contain the filtered data	REAL array dimension Y(N)	Output

N	Number of points to be filtered and size of X and Y arrays	INTEGER	Input
---	--	---------	-------

Table II

For user's purposes, this subroutine can be treated as a black box which returns one output value for each input value. By its nature, it "remembers" previous inputs, even after it has returned control to the main program. The user need not filter all the data in a time series at one time, if, for example, the data are arranged in several records on a tape. Consecutive calls to RNF using part of the series at a time (in the right order, of course) will give the same results as if all the filtering were done in one call. The following calls give identical results:

CALL RNF(X,Y,1000)

and

CALL RNF(X,Y,500)
CALL RNF(X(501), Y(501), 500)

The first is slightly more economical, but the computed results are the same.

It follows that if the filter remembers previous inputs, there should be a way to make it forget them, in order to pass unrelated series through the same filter. This capability is provided in a call which brings the filter to a fully relaxed state. This call, in FORTRAN, is:

CALL ZERRNF

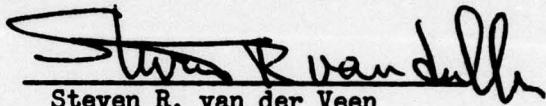
which zeros the memory elements of the filter.

BTRRNF and associated subroutines are located on FASTRAND file S1787/LIBRY and are available for general use.

NUSC/NL Tech Memo
No. 2211-309-70

SUMMARY

The subroutine BTRRNF allows the simulation of Butterworth filters on a digital computer. The method used is impulse-invariant in the sense that the computer output is identical to the sampled output of the equivalent analog filter. It is expected that the errors introduced by the use of a finite number system in the computer are much smaller than are the errors introduced in analog devices by variabilities in component specification.



Steven R. van der Veen
Physicist

NUSC/NL Tech Memo
No. 2211-309-70

REFERENCES

- a. Freeman, H., Discrete Time Systems, an Introduction to the Theory, J. Wiley and Sons, N.Y., 1965
- b. Rader, C. and Gold, B., "Digital Filter Design Techniques in the Frequency Domain", Proc. of the IEEE, Vol. 55, No. 2, Feb 1967

NJUSC/NL Tech Memo
No. 2211-309-70

APPENDIX A: SAMPLE PROGRAM

A record on FASTRAND file SRV/82875 was known to contain a bottom reflected signal embedded in noise. Since it was known that the noise level was high outside the frequency band 100-400 Hz, it was decided to pass the signal through a bandpass filter with that passband. A program listing of the sample program and the plotted CALCOMP output (Figure 1) follow, demonstrating the use of the BTRRNF subroutine.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

1* C
2* L PROGRAM TO DEMONSTRATE DIGITAL FILTERING ROUTINE BTTRNF
3* L
4* L THE FIRST RECORD ON FASTRAND FILE SRV/82875 WAS
5* L KNOWN TO CONTAIN A TIME SERIES WHICH WAS PREDOMINANTLY
6* L NOISE OUTSIDE THE FREQUENCY BAND 100-400 HZ.
7* L THEREFORE IT WAS USED HERE TO DEMONSTRATE THE USE OF
8* L THE BANPASS FILTER.
9* L
10* L DIMENSION X(2048),Y(2002),IBUF(4096)
11* L
12* L INITIALIZE PLOTTING ROUTINE AND READ IN RECORD
13* L
14* L CALL PLOTS(1000,4096,5) NUSC/NL Tech Memo
15* L CALL PLOT(0.,0.,200)
16* L CALL PLOT(0.,1.5,-3) No. 2211-309-70
17* L CALL FACTOR(.8)
18* L Y(1)=.005
19* L DO 3 I=2,2000

20* L Y(I)=Y(I-1)+.005
21* L 3 CONTINUE
22* L Y(2001)=0.
23* L Y(2002)=1.
24* L CALL FTRAN(1,'82875')
25* L CALL FTRAN(3,'SRV',L)
26* L IF(L)99,99,
27* L ISECTR=1
28* L CALL FTRAN(8,'SRV',ISECTR,2048,X,IFERR)
29* L 1 IF(IFERR+1)98,1,
30* L
31* L SET UP A 5-POLE BP FILTER CENTERED AT 200 HZ, 300 HZ
32* L WIDE FOR A SAMPLING RATE OF 5000 HZ.
33* L
34* L CALL BTTRNF(5,1,200.,300.,.0002,IFERR)
35* L IF(IFERR .NE. 1) GO TO 9/
36* L
37* L PLUT THE UNFILTERED SIGNAL
38* L
39* L CALL AXIS(0.,0.,'SIGNAL AMPLITUDE',16,3.,90.,-.05,.05,10.)
40* L X(2001)=-.05
41* L X(2002)=.05
42* L CALL LINE(Y*X,2000,1,0,0)
43* L CALL SYMBOL(3.,2.5,.15,'UNFILTERED DATA',0.,15)
44* L CALL PLOT(0.,4.,-3)
45* L
46* L FILTER 2000 POINTS OF THE X ARRAY AND PLUT THE OUTPUT
47* L
48* L CALL RNF(X,X,2000)
49* L CALL AXIS(0.,0.,'SIGNAL AMPLITUDE',16,3.,90.,-.05,.05,10.)
50* L CALL LINE(Y*X,2000,1,0,0)
51* L CALL SYMBOL(3.,2.5,.15,'FILTERED DATA',0.,15)
52* L CALL PLOT(12.,-4.,-3)
53* L
54* L STOP
55* L 99 STOP 99
56* L 98 STOP 98
57* L 97 STOP 97
58* L END

NUSC/NL Tech Memo
No. 2211-309-70

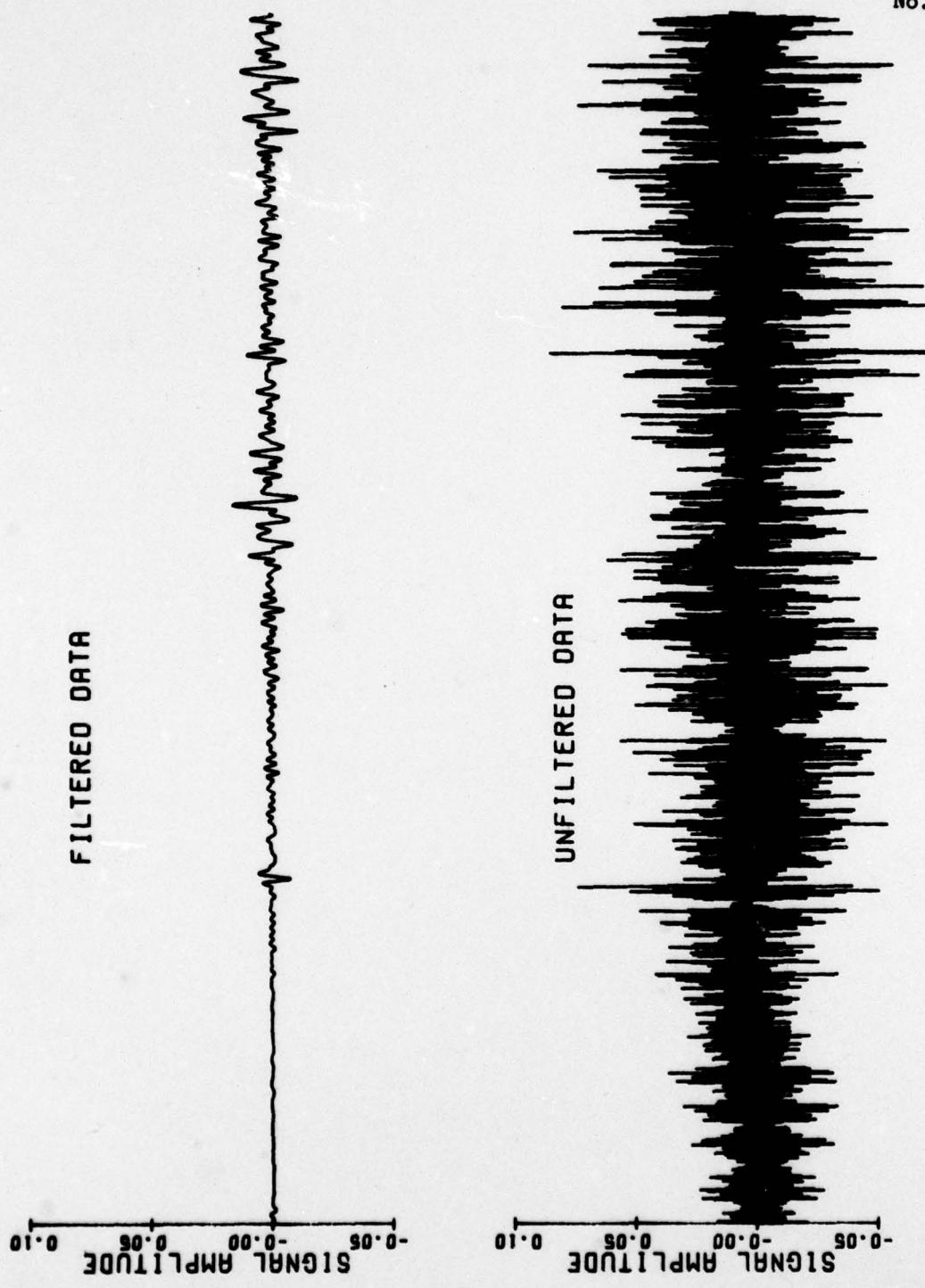


Figure 1: Plot of output from sample program
time vs. signal amplitude

NUSC/NL Tech Memo
No. 2211-309-70

APPENDIX B: PROGRAM LISTINGS

All routines written for this program were written in FORTRAN V for the UNIVAC 1108 with the exception of the subroutine RNF, which was written in 1108 Assembly Language. Listings of all programs follow.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

00101 1* SUBROUTINE BTRRNF(NPOLES,ITYPE,CF,BW,T,IFERR)
00101 2* C
00101 3* C ** SUBROUTINE BTRRNF EVALUATES THE COEFFICIENTS OF
00101 4* C ** A BUTTERWORTH FILTER. THE PARAMETER ITYPE
00101 5* C ** CONTROLS THE TYPE OF FILTER (LP OR BP). IF ITYPE
00101 6* C ** IS 0, A LOW PASS FILTER IS GENERATED. IF IT
00101 7* C ** IS 1, A BAND PASS FILTER IS GENERATED. NPOLES IS
00101 8* C ** THE NUMBER OF POLES IN THE LP VERSION. THIS
00101 9* C ** VERSION IS BANDSHIFTED IF A BP FILTER IS DESIRED.
00101 10* C ** BW IS THE BANDWIDTH OF THE FILTER, AND CF IS THE
00101 11* C ** CENTER FREQUENCY (IF ONE IS NECESSARY). T IS THE
00101 12* C ** SAMPLING INTERVAL. ALL ARE IN UNITS OF HZ OR
00101 13* C ** SECs. IFERR IS AN ERROR INDICATOR AND WILL BE
00101 14* C ** 1 ON A NORMAL RETURN. NPOLES MUST BE .LE. 10.
00101 15* C ** IN ALL CASES, IF IFERR IS SET TO 1, A FILTER
00101 16* C
00101 17* C ** HAS BEEN LOADED IN THE RNF FILTER SUBROUTINE.
00101 18* C ** SEE NUSC/NL TECH MEMO 2211-309-70. 'A PROGRAM
00101 19* C ** TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL
00101 20* C ** COMPUTER'
00101 21* C
00103 22* DOUBLE PRECISION POLE(2,5),POLES(2,10),ZERO(2,1),
00103 23* *ZEROS(2,10),SCALE,TC

00103 24* C
00103 25* C ** CONVERT NPOLES,T TO STORE IN PROGRAM
00103 26* C
00104 27* C TC=T
00105 28* C N=NPOLES
00106 29* C N1=0
00106 30* C
00106 31* C ** N MUST BE .LE. 10. TEST IT
00106 32* C
00107 33* C IF(N .GT. 10) GO TO 100
00107 34* C
00107 35* C ** LOCATE POLES OF N-POLE LP FILTER WITH BANDWIDTH BW
00107 36* C
00111 37* C CALL BTRWRT(N,BW,POLE,SCALE,M1,M2)
00111 38* C
00111 39* C ** IF ITYPE .EQ. 1, BANDPASS FILTER IS REQUESTED.
00111 40* C ** GENERATE IT AT STATEMENT 10
00111 41* C
00112 42* C IF(ITYPE .NE. 0) GO TO 10
00112 43* C
00112 44* C ** ZERO ALL STORAGE IN RNF TO PREPARE IT TO
00112 45* C ** ACCEPT NEW FILTER COEFFS
00112 46* C
00114 47* C CALL LUQRNF
00114 48* C
00114 49* C ** COMPUTE COEFFS OF FILTER AND PASS THEM TO RNF
00114 50* C
00115 51* C CALL RECUMS(ZERO,N1,POLE,M1,M2,SCALE,TC)
00115 52* C
00115 53* C ** ALL O.K. RETURN
00115 54* C
00116 55* C IFERR=1
00117 56* C RETURN

NUSC/NL Tech Memo
No. 2211-309-70

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

00117 57* C ** CONVERT FROM LOWPASS TO BANDPASS IN S-DOMAIN
00117 58* C
00117 59* C
00120 60* 10 CALL LPTOBP(CF,POLE,M1,M2,ZERO,N1,POLES,M3,M4,
00120 61* +ZEROS,N2,SCALE,IFRR)
00120 62* C
00120 63* C ** TEST FOR ERROR IN LPTOBP
00120 64* C ** IF IFRR IS 0, AN ERROR EXISTS NUSC/NL Tech Memo
00120 65* C No. 2211-309-70
00121 66* C IF(IFRR .EQ. 0) GO TO 100
00121 67* C
00121 68* C ** ZERO ALL STORAGE IN RNF TO PREPARE IT TO
00121 69* C ** ACCEPT NEW FILTER COEFFS
00121 70* C
00123 71* C CALL LODRNF
00123 72* C
00123 73* C ** COMPUTE COEFFS OF FILTER AND PASS THEM TO RNF
00123 74* C
00124 75* C CALL RECURS(ZEROS,N2,POLES,M3,M4,SCALE,TC)
00124 76* C
00124 77* C ** ALL O.K. RETURN
00124 78* C
00125 79* C IFERR=1
00126 80* C RETURN
00126 81* C

00126 82* C ** ERROR RETURN WITH IFERR SET TO 0
00126 83* C
00127 84* C 100 IFERR=0
00130 85* C RETURN
00131 86* C END

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

NUSC/NL Tech Memo
No. 2211-309-70

SUBROUTINE RECURS(ZERO,N,POLE,M1,M2,SCALE,T)

** THIS SUBROUTINE PERFORMS THE IMPULSE-INVARIANT TRANSFORM FROM THE
** S-PLANE TO THE Z-PLANE. DOUBLE PRECISION IS REQUIRED FOR
** ACCURACY. NOTE THAT RECURS WILL NOT HANDLE MULTIPLE POLES.

** THE ZEROS OF THE FILTER ARE TO BE N IN NUMBER, AND STORED IN ZERO(I,J)
** WHERE ZERO(1,J) IS THE REAL PART OF THE J-TH ZERO, AND ZERO(2,J) IS THE
** IMAGINARY PART OF THE J-TH ZERO.
** THE POLES OF THE FILTER ARE TO BE STORED IN POLE(I,J) WHERE I HAS THE
** SAME MEANING AS BEFORE. IF J IS LESS THAN OR EQUAL TO M1, THE POLE IS A
** REAL POLE AND ITS IMAGINARY PART WILL BE ZERO. IF J IS GREATER THAN M1
** THE POLE WILL BE ONE OF A COMPLEX PAIR. ONLY ONE OF A COMPLEX PAIR IS
** INCLUDED IN THE POLE ARRAY, THE OTHER BEING PRESENT BY IMPLICATION.
** THERE ARE TO BE M1 REAL POLES AND M2 CONJUGATE POLE PAIRS. SCALE IS
** THE FILTER AMPLIFICATION CONSTANT, AND T IS THE SAMPLE INTERVAL. ALL
** NON-INTEGER VARIABLES ARE TO BE DOUBLE PRECISION EXCEPT FOR THE
** ARRAY CONTAINING THE COMPUTED COEFFICIENTS FOR THE RNF ROUTINE.

** SEE NUSC/NL TECH MEMO 2211-309-70. 'A COMPUTER PROGRAM
** TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL COMPUTER'

DOUBLE PRECISION T,SCALE,TBAR
DOUBLE PRECISION Z(2),P(2),TEMP,A(2),POLE(2,20),ZERO(2,20),TMP(2)

REAL COEFF(4)

MTOT=M1+M2

DO 100 II=1,MTOT

Z(1)=1.0E0

P(1)=1.0E0

P(2)=0.0E0

Z(2)=0.0E0

** TEST FOR NO ZEROS

IF(N .EQ. 0) GO TO 111

** COMPUTE PRODUCT OF (POLE(II)-ZERO(I)) FOR ALL I AND STORE IN Z

DO 110 II=1,N

TEMP=Z(1)*(POLE(1,II)-ZERO(1,1))- Z(2)*(POLE(2,II)-ZERO(2,1))

Z(2)=Z(2)*(POLE(1,II)-ZERO(1,1))+ Z(1)*(POLE(2,II)-ZERO(2,1))

110 Z(1)=TEMP

** COMPUTE PRODUCT OF (POLE(I)-POLE(I)) FOR ALL I .NE. II AND STORE
** IN P

118 GO TO 120 I=1,M1UT
IF(I .LE. II) GO TO 129
TEMP=P(1)*(POLE(1+II)-POLE(1,I)) + P(2)*(POLE(2+II)-POLE(2,I))
P(2)=P(2)*(POLE(1+II)-POLE(1,I)) + P(1)*(POLE(2+II)-POLE(2,I))
P(1)=TEMP

** TEST FOR REAL POLE

NUSC/NL Tech Memo
No. 2211-309-70

129 IF(I .LE. M1) GO TO 130

** CONJUGATE POLE PAIR. COMPUTE TERM FOR CONJUGATE OF POLE(I)

TEMP=P(1)*(POLE(1+II)-POLE(1,I)) - P(2)*(POLE(2+II)+POLE(2,I))
P(2)=P(2)*(POLE(1+II)-POLE(1,I)) + P(1)*(POLE(2+II)+POLE(2,I))
P(1)=TEMP

130 CONTINUE

120 CONTINUE

** LOCATE POLE IN Z-PLANE

A(1)= DEEXP(POLE(1+II)*T)+DCOS(POLE(2,II)*T)
A(2)= DEEXP(POLE(1+II)*T)+DSIN(POLE(2,II)*T)

** COMPUTE Z/P*SCALE*T. AMPLIFICATION CONSTANT

TEMP=P(1)*P(1)+P(2)*P(2)
TBANK=SCALE/TEMP*T
TMP(1)= TBANK*(P(1)*Z(1)+P(2)*Z(2))
TMP(2)= TBANK*(P(1)*Z(2)-P(2)*Z(1))

** TEST FOR REAL POLE

IF(II .LE. M1) GO TO 150

** COMPLEX POLE--SECOND ORDER FILTER

Coeff(1)=2.*IMP(1)
Coeff(2)=2.*(A(1)* TMP(1)+A(2)* TMP(2))
Coeff(3)=2.*A(1)
Coeff(4)=A(1)*A(1)+A(2)*A(2)

** LOAD COEFFICIENTS

CALL ORD2(Coeff(1),Coeff(2),Coeff(3),Coeff(4))
GO TO 100

** REAL POLE--FIRST ORDER FILTER

150 Coeff(1)=TMP(1)
Coeff(2)=A(1)

** LOAD COEFFICIENTS

CALL ORD1(Coeff(1),Coeff(2))
100 CONTINUE
RETURN
END

00101 1* SUBROUTINE LP10BP(CF,P1,M1,M2,Z1,N1,P2,M3,M4,
00101 2* *Z2,N2,SC,IFERR)
00101 3* C
00101 4* C ** THIS SUBROUTINE PERFORMS THE BANDSHIFT NECESSARY
00101 5* C ** TO CONVERT A POLE-ZERO SET CENTERED ON THE REAL
00101 6* C ** AXIS TO A PAIR OF SETS CENTERED AT +- (CF). IT
00101 7* C ** DOES THIS BY THE TRANSFORM S=SPRIME/CF+CF/SPRIME.
00101 8* C ** THE POLES AND ZEROS TO BE TRANSFORMED ARE IN P1
00101 9* C ** AND Z1 AND THE RESULTS OF THE TRANSFORM ARE IN
00101 10* C ** P2 AND Z2. NOTE THAT EACH POLE PRODUCES 2
00101 11* C ** POLES AND 1 ZERO(AT THE ORIGIN), WITH THE OPPOSITE
00101 12* C ** OCCURRING FOR EACH ZERO.
00101 13* C
00101 14* C ** SEE NUSC/NL TECH MEMO 2211-309-70. 'A PROGRAM
00101 15* C ** TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL
00101 16* C ** COMPUTER'
00101 17* C
00103 18* DOUBLE PRECISION P1(2,10),P2(2,10),Z1(2,5),Z2(2,5),
00103 19* *A,B,C,D,R,PHI,PI,CFT
00104 20* PI=3.14159265358979336800
00105 21* CFT=CF*PI*2.000

NUSC/NL Tech Memo
No. 2211-309-70

00106 22* N2=0
00107 23* M3=0
00110 24* M4=0
00111 25* IFERR=1
00111 26* C
00111 27* C ** MM=NUMBER OF POLES
00111 28* C ** NN=NUMBER OF ZEROS
00111 29* C
00112 30* MM=M1+2*M2
00113 31* NN=N1
00113 32* C
00113 33* C ** EACH POLE PRODUCES A ZERO AT THE ORIGIN AND EACH
00113 34* C ** ZERO A POLE. THESE WILL CANCEL. COMPUTE THE
00113 35* C ** DIFFERENCE IN NUMBER. IF MULTIPLE POLES OCCUR
00113 36* C ** AT THE ORIGIN (IF MM>NN .LT. -1), AN ERROR WILL
00113 37* C ** OCCUR IN RECURS. SO TEST MM>NN.
00113 38* C
00114 39* II=MM-NN
00115 40* IF(II)1,7,5
00120 41* 1 IF(II+1)2,3,3
00120 42* C
00120 43* C ** TOO MANY ZEROS
00120 44* C
00123 45* 2 IFERR=0
00123 46* C
00123 47* C ** PUT POLES AT ORIGIN, EVEN IF THERE ARE MORE THAN
00123 48* C ** ONE
00123 49* C
00124 50* 3 DO 4 I=-1,II,-1
00127 51* M3=M3+1
00130 52* P2(1,M3)=-1.0-05
00131 53* 4 P2(2,M3)=0.000
00133 54* GO TO 7

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDG

00133 55* C
00133 56* C ** MORE POLES THAN ZEROS, PUT ZEROS AT ORIGIN
00133 57* C
00134 58* 5 DO 6 I=1,11
00137 59* N2=N2+1
00140 60* Z2(1,N2)=0.000
00141 61* 6 Z2(2,N2)=0.000
00143 62* 7 CONTINUE
00144 63* IF(MM)11,11,
00147 64* MTUT=M1+M2
00150 65* ICNT=M3
00150 66* C
00150 67* C ** SOLVE TRANSFORMATION EQUATION(CF/P2+P2/CF-P1(1)=0)
00150 68* C ** FOR NEW POLES
00150 69* C
00151 70* DO 10 I=1,MTUT
00154 71* A=P1(1,I)
00155 72* B=P1(2,I)
00156 73* C=2.000*A*B
00157 74* D=A*A-B*B-4.000*CFT*CFT
00160 75* PHI=DATAN2(C,D)/2.000
00161 76* R=DSQRT(DSQRT(C+C*D*D))
00162 77* C=R*DCOS(PHI)/2.000
00163 78* D=R*DSIN(PHI)/2.000
00164 79* ICNT=ICNT+1

00165 80* P2(1,ICNT)=A/2.000+C
00166 81* P2(2,ICNT)=B/2.000+D
00166 82* C
00166 83* C ** TEST FOR P1(I) NON-REAL
00166 84* C
00167 85* IF(I .GT. M1) GO TO 9
00167 86* C
00167 87* C ** P1(I) IS REAL, SECOND ROOT CONJUGATE OF FIRST
00167 88* C
00171 89* M4=M4+1
00172 90* GO TO 10
00172 91* C
00172 92* C ** P1(I) COMPLEX, FIND SECOND ROOT. (ROOTS DUE TO
00172 93* C ** CONJUGATE OF P1(I) ARE EQUAL TO THE CONJUGATE OF
00172 94* C ** ROOTS DUE TO P1(I))
00172 95* C
00173 96* 9 M4=M4+1
00174 97* ICNT=ICNT+1
00175 98* P2(1,ICNT)=A/2.000+C
00176 99* P2(2,ICNT)=B/2.000+D
00177 100* M4=M4+1
00200 101* 10 CONTINUE
00202 102* 11 CONTINUE
00203 103* IF(NN)21,<1,

NUSC/NL Tech Memo
No. 2211-309-70

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

NUSC/NL Tech Memo
No. 2211-309-70

00203 104* C
00203 105* C ** SOLVE TRANSFORMATION EQUATION(CF/Z2+z2/CF-Z1(I)=0)
00203 106* C ** FOR NEW ZEROS
00203 107* C
00206 108* DO 20 I=1,NN
00211 109* N2=N2+1
00212 110* A=Z1(1,I)
00213 111* B=Z1(Z,I)
00214 112* C=2.000*A*B
00215 113* D=A*A-B*B-4.000*CFT*CFT
00216 114* PHI=DATAN2(C,U)/2.000
00217 115* R=DSQRT(DSQRT(C*C+D*D))
00220 116* C=R*DCOS(PHI)/2.000
00221 117* U=R*DSIN(PHI)/2.000
00222 118* Z2(1,N2)=A/2.000+C
00223 119* Z2(2,N2)=B/2.000+U
00224 120* N2=N2+1
00225 121* Z2(1,N2)=A/2.000-C
00226 122* Z2(2,N2)=B/2.000-U
00227 123* 20 CONTINUE
00231 124* 21 CONTINUE
00232 125* RETURN
00233 126* END

00101 1* SUBROUTINE BTRWRT(N,BW,POLES,SCALE,M1,M2)
00103 2*
00103 3*
00103 4* C DOUBLE PRECISION POLES(2,N),BWC,PI,SCALE,XIND
00103 5* C ** BTRWRT LOCATES THE POLES OF AN N-POLE BUTTERWORTH
00103 6* C ** FILTER WITH BANDWIDTH BW.
00103 7* C ** SEE NUSC/NL TECH MEMO 2211-309-70. 'A PROGRAM
00103 8* C ** TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL
00103 9* C ** COMPUTER'
00103 10* C
00104 11*
00105 12*
00106 13*
00107 14*
00110 15*
00111 16*
00113 17*
00114 18*
00115 19*
00116 20*
00117 21*
00120 22*
00121 23*
00124 24*
00125 25*
00126 26*
00127 27*

00130 28*
00132 29*
00133 30*

SUBROUTINE BTRWRT(N,BW,POLES,SCALE,M1,M2)
DOUBLE PRECISION POLES(2,N),BWC,PI,SCALE,XIND
** BTRWRT LOCATES THE POLES OF AN N-POLE BUTTERWORTH
** FILTER WITH BANDWIDTH BW.
** SEE NUSC/NL TECH MEMO 2211-309-70. 'A PROGRAM
** TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL
** COMPUTER'
PI=3.141592653589793368D0
SCALE=1.0D0
BWC=2.0D0*PI*BW
ICT=0
M1=0
IF (N/2+2 .EQ. N) GO TO 10
SCALE=BWC
ICT=1
M1=1
POLES(1,1)=-BWC
POLES(2,1)=0.0D0
10 M2=N/2
DO 20 I=1,M2
ICT=ICT+1
XIND=(N-1+2*I)*PI/N/2.0D0
POLES(1,ICT)=BWC*DCOS(XIND)
POLES(2,ICT)=BWC*DSIN(XIND)

20 SCALE=SCALE*BWC*BWC
RETURN
END

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDG

NUSC/NL Tech Memo
No. 2211-309-70

00100	1*	C	
00100	2*	C	** IMMED PROVIDES IMMEDIATE LINKS BETWEEN RECURS
00100	3*	C	** AND KMF, AND CAN BE BYPASSED WITH APPROPRIATE
00100	4*	C	** CALLS IF DESIRED, BY PROVIDING ENTRY POINTS
00100	5*	C	** 'ORD1(A,B)' AND 'ORD2(A,B,C,D)' AND THEN STORING
00100	6*	C	** A, B, C, AND D IN ARRAYS
00100	7*	C	
00100	8*	C	** SEE NUSC/NL TECH MEMO 2211-309-70. 'A PROGRAM
00100	9*	C	** TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL
00100	10*	C	** COMPUTER'
00100	11*	C	
00101	12*		SUBROUTINE IMMED
00103	13*		ENTRY ORD1(A,B)
00105	14*		CALL ORD1(A,B)
00106	15*		RETURN
00107	16*		ENTRY ORD2(A,B,C,D)
00111	17*		CALL ORD2(A,B,C,D)
00112	18*		RETURN
00113	19*		END

W1 ASM KNF PKT

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

- KNF IS THE PROGRAM THAT ACTUALLY IMPLEMENTS THE FILTER GENERATED BY SUBROUTINE NKF.
- IT CONTAINS ENTRY POINTS FOR LOADING COEFFICIENTS,
- FOR RELAXING THE FILTER, FOR ZEROING ALL COEFFICIENT STORAGE, AND FOR
- ACTUALLY FILTERING AN ARRAY OF POINTS.
- SEE NUSC/NL TECH MEMO 2211-309-70, 'A PROGRAM TO SIMULATE BUTTERWORTH FILTERS ON A DIGITAL COMPUTER'

NUSC/NL Tech Memo
No. 2211-309-70

REGNAM

S(0)

XPAST	RES	1
YY1	RES	10
YY2	RES	10

YY11	RES	10
N1ORDE	RES	1
N2ORDE	RES	1
CC1	RES	10
CC2	RES	10
BB1	RES	10
BB2	RES	10
CC11	RES	10
BB11	RES	10
SAVREG	RES	4

S(1)

RNF*

SX	X1,SAVREG
SX	X2,SAVREG+1
SX	X3,SAVREG+2
SX	X4,SAVREG+3
LX	X3,0,X11
LXI+XU	X3,1
LX	X4,1,X11
LXI+XU	X4,1
LX	X2,*2,X11
ANX+XU	X2,1

OUTLOO

SC	A0
LX	X1,N2ORDE
JGD	X1,LOOP2
J	FIRSTO

LOOP2

LA	A2,CC1,X1
FM	A2,0,X3
SA	A2,A4
LA	A2,CC2,X1
FM	A2,XPAST
FAN	A4,A2
LA	A2,YY2,X1
FM	A2,BB2,X1
FAN	A4,A2
LA	A2,YY1,X1
SA	A2,YY2,X1
FM	A2,BB1,X1
FA	A4,A2

- STORE DATA ON LOC CNTR 0
- PREVIOUS VALUE OF INPUT
- PREV VAL OF OUTPUT, 2NU ORD FILTS
- NEXT TO LAST VAL OF OUTPUT, 2ND ORDER FILTS
- PREV VAL OF OUTPUT, 1ST ORD FILTS
- NUMBER OF FIRST ORDER FILTERS
- NUMBER OF 2NU ORDER FILTERS
- C-SUB-1 2NU ORDER FILTS
- C-SUB-2 2ND ORDER FILTS
- B-SUB-1 2NU ORD FILTS
- B-SUB-2 2ND ORD FILTS
- C-SUB-1 1ST ORD FILTS
- B-SUB-1 1ST ORD FILTS
- TEMP STORAGE TO SAVE REGISTERS
- CODE UNDER LOC CNTR 1
- ENTRY FOR CALL RNF(X,Y,N)
- SAVE REG X1
- SAVE REG X2
- SAVE REG X3
- SAVE REG X4
- ADDR(X(1)) INTO X3
- 1 INTO X3 INCREMENT
- ADDR(Y(1)) INTO X4
- 1 INTO X4 INCREMENT
- N INTO X2 FOR LOOPING
- INITIALIZE X2
- OUTER LOOP FOR N FILTER OPS
- 0 INTO A0(Y-ACCUMULATOR)
- N2ORDE INTO X1
- TEST X1 NOT 0 AND INIT FOR LOOP
- N2ORDE 0, RUN 1ST ORD FILTS
- LOOP FOR 2ND ORD FILTS
- CC1-SUB-X1 INTO A2
- A2 TIMES X-SUB-X3 INTO A2
- A2 INTO A4
- CC2-SUB-X1 INTO A2
- A2 TIMES XPAST INTO A2
- A4 MINUS A2 INTO A4
- YY2-SUB-X1 INTO A2
- A2 TIMES BB2-SUB-X1 INTO A2
- A4 MINUS A2 INTO A4
- YY1-SUB-X1 INTO A2
- A2 INTO YY2-SUB-X1
- A2 TIMES BB1-SUB-X1 INTO A2
- A4 PLUS A2 INTO A4

FIRSTO	SA	A4,YY1,X1	• A4 INTO YY1-SUB-X1
	FA	AU,A4	• AO PLUS A4 INTO AO(Y ACCUMULATOR)
	JGD	X1,LOOPZ	• NEXT 2ND ORD FILT, IF ANY
	LX	X1,N1ORDE	• START OF CODE FOR 1ST ORD FILTS
	JGD	X1,LOOP1	• N1ORDE INTO X1
LOOP1	J	STOREY	• TEST X1 NOT 0 AND INIT FOR LOOP
	LA	A2,CC11,X1	• N1ORDE 0, STORE Y
	FM	A2,0,X3	• LOOP FOR 1ST ORD FILTS
	SA	A2,A4	• CC11-SUB-X1 INTO A2
	LA	A2,BB11,X1	• A2 TIMES X-SUB-X3 INTO A2
	FM	A2,YY11,X1	• A2 INTO A4
	FA	A4,A2	• BB11-SUB-X1 INTO A2
	SA	A4,YY11,X1	• A2 TIMES YY11-SUB-X1 INTO A2
	FA	A0,A4	• A4 PLUS A2 INTO A4
STOREY	JGD	X1,LOOP1	• A4 INTO YY11-SUB-X1
	LA	A4,0,*X3	• AO PLUS A4 INTO AO(Y ACCUMULATOR)
	SA	A4,XPAST	• NEXT 1ST ORD FILT, IF ANY
	SA	A0,0,*X4	• X-SUB-X3 INTO A4 THEN INCREMENT
	JGD	X2,OUTLOO	• X3 FOR NEXT X VALUE
	LX	X1,SAVREG	• A4 INTO XPAST
	LX	X2,SAVREG+1	• AO INTO Y-SUB-X4 THEN INCREMENT
	LX	X3,SAVREG+2	• X4 FOR NEXT Y
	LX	X4,SAVREG+3	• NEXT POINT FOR FILTERING
	J	4,X11	• RESTORE X1
LOURNF*			• RESTORE X2
	LX,XU	X12,92	• RESTORE X3
ZERNF*	J	S+2	• RESTORE X4
			• RETURN
			• ENTRY FOR CALL LOURNF
			• ZEROS ALL STORAGE
			• 92 INTO X12
LOOPZ	LX,XU	X12,30	
	SZ	XPAST+X12	• ENTRY FOR CALL ZERRNF
	JGC	X12,LOOPZ	• ZEROS ALL NON-COEFF. STORAGE
	J	1,X11	• 30 INTO X12
ORDER2*			• LOOP ON ZEROING STEP
	SX	X1,SAVREG	• 0 INTO XPAST+X12
	LX	X1,N2ORDE	• ZERO NEXT LOC, IF ANY
	LA	A0,*0,X11	• RETURN
	SA	A0,CC1,X1	• ENTRY FOR CALL ORDER2(A,B,C,D)
	LA	A0,*1,X11	• PASSES 2ND ORD FILT COEFFS
	SA	A0,CC2,X1	• SAVE REG X1
	LA	A0,*2,X11	• N2ORDE INTO X1
	SA	A0,BB1,X1	• A INTO AO
	LA	A0,*3,X11	• AO INTO CC1-SUB-X1
	SA	A0,BB2,X1	• B INTO AO
	AX,XU	X1,1	• AO INTO CC2-SUB-X1
	SX	X1,N2ORDE	• C INTO AO
	LX	X1,SAVREG	• AO INTO BB1-SUB-X1
ORDER1*	J	5,X11	• D INTO AO
	SX	X1,SAVREG	• AO INTO BB2-SUB-X1
			• INCREMENT X1
			• STORE NEW N2ORDE
			• RESTORE X1
			• RETURN
			• ENTRY FOR CALL ORDER1(A,B)
			• PASSES 1ST ORD FILT COEFFS
			• SAVE REG X1

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

NUSC/NL Tech Memo
No. 2211-309-70

LX	X1.N1ORDE	
LA	A0.*0,X11	• N1ORDE INTO X1
SA	A0.CC11,X1	• A INTO A0
LA	A0.*1,X11	• A0 INTO CC11-SUB-X1
SA	A0.BB11,X1	• B INTO A0
AX,XU	X1.1	• A0 INTO BB11-SUB-X1
SX	X1.N1ORDE	• INCREMENT X1
LX	X1.SAVREG	• STORE NEW N1ORDE
J	3,X11	• RESTORE X1
END		• RETURN