THE CEPSTRUM, THE CEPSTRALLY SMOOTHED LOG SPECTRUM AND THE CHIRP Z-TRANSFORM

5 January 1972

TM-4857/200/00

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited
The Cepstrum, the Cepstrally Smoothed Log Spectrum and the Chirp-Z Transform

Several digital signal processing algorithms useful in speech research are presented: the cepstrum pitch detector; the cepstrally smoothed log spectrum; and the Chirp Z-transform. FORTRAN codes and examples accompany the discussions.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th></th>
<th>LINK B</th>
<th></th>
<th>LINK C</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
<td>WT</td>
</tr>
<tr>
<td>Cepstrum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cepstrally Smoothed Log Spectrum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chirp Z-Transform</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital Signal Processing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pitch detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formant Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speech Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The work reported herein was supported by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC15-67-C-0149, ARPA Order No. 1327, Amendment No. 4, Program Code No. 2D30 and 2P10.

THE CEPSTRUM, THE CEPSTRALLY SMOOTHED LOG SPECTRUM AND THE CHIRP Z-TRANSFORM

by

H. Barry Rice

5 January 1972

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U. S. Government.

Distribution of this document is unlimited.
ABSTRACT

Several digital signal processing algorithms useful in speech research are presented: the cepstrum pitch detector, the cepstrally smoothed log spectrum, and the Chirp Z-transform. FORTRAN codes and examples accompany the discussions.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>CEPSTRUM PITCH DETECTOR</td>
<td>1</td>
</tr>
<tr>
<td>3.</td>
<td>CEPSTRALLY SMOOTHED LOG SPECTRUM</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>CHIRP Z-TRANSFORM</td>
<td>9</td>
</tr>
<tr>
<td>5.</td>
<td>REFERENCES</td>
<td>16</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1.</td>
<td>Cepstrum Pitch Detector Program Listing</td>
<td>2</td>
</tr>
<tr>
<td>2.</td>
<td>Cepstrum Example</td>
<td>4</td>
</tr>
<tr>
<td>3.</td>
<td>Cepstrally Smoothed Log Spectrum Program Listing</td>
<td>5</td>
</tr>
<tr>
<td>4.</td>
<td>Unsmoothed Log Spectrum Example</td>
<td>7</td>
</tr>
<tr>
<td>5.</td>
<td>Cepstrally Smoothed Log Spectrum Example</td>
<td>8</td>
</tr>
<tr>
<td>6.</td>
<td>Chirp Z-Transform Program Listing</td>
<td>10</td>
</tr>
<tr>
<td>7.</td>
<td>Subroutine CONTOR Program Listing</td>
<td>12</td>
</tr>
<tr>
<td>8.</td>
<td>Subroutine POWER Program Listing</td>
<td>13</td>
</tr>
<tr>
<td>9.</td>
<td>Double Precision Chirp Z-Transform Program Listing</td>
<td>14</td>
</tr>
</tbody>
</table>
1. **INTRODUCTION**

The development of the Fast Fourier Transform (FFT) led to the realization of a number of FFT applications to speech research. Chief among these are the cepstrum pitch detector for automatic detection of the pitch period of voiced speech, the cepstrally smoothed log spectrum for formant selection, and the Chirp Z-transform for narrow-band frequency analysis. Each of these applications is discussed in the following sections and references are given for more detailed presentations. In addition, complete FORTRAN codes and examples accompany the discussions.

2. **CEPSTRUM PITCH DETECTOR**

In 1967, Noll [1] described a method for computing the pitch period of voiced speech using the cepstrum. The cepstrum is defined as the square of the Fourier transform of the logarithm power spectrum and has a strong peak corresponding to the pitch period of the voiced-speech segment being analyzed. Schafer and Rabiner [2] have also described this procedure in a way which allows the computer code to be easily constructed. The FORTRAN program listing is shown in Figure 1. The input data are the digitized speech sample $S(I)$, the number $N$ of samples $S(I)$ (which must be compatible with the FFT algorithm used), the sampling period $T$ (in sec.), and a rough estimate of the pitch period $PP$. The latter parameter is used to establish the length of the Hamming window (viz., $4*PP$) through which the signal $S(I)$ is passed.

As an example, a truncated Fourier series

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos 2\pi kwt + b_k \sin 2\pi kwt)$$
SUBROUTINE CEPS(S,N,T,C,PP,X)

CEPSTRUM CALCULATION

REFERENCE: R. W. SCHAFER & L. R. RABINER,
"SYSTEM FOR AUTOMATIC FORMANT ANALYSIS OF VOICED SPEECH,"

S(I) = DIGITIZED SPEECH SAMPLE
T = SAMPLING PERIOD (IN SEC)
C(I) = CEPSTRUM OF S(I)
PP = PITCH PERIOD ESTIMATE
X(I) = COMPLEX CEPSTRUM
DIMENSION S(N),C(N),X(N)
COMPLEX X

MULTIPLY SAMPLE BY HAMMING WINDOW
XN=N
TWOPI=6.283185317178
DURVIN=4.*PP
DO 17 I=1,N
XI=I-1
IF (XI*T-DURVIN) 10,10,15
10 ARG=TWOPI*XI*T/DURWIN
S(I)=S(I)*(.54-.46*COS(ARG))
GO TO 17
15 S(I)=0.
17 CONTINUE
DO 20 I=1,N
20 X(I)=CMPLX(S(I),0.)

FIND M SUCH THAT N = 2**M
K=N
I=0
30 K=K/2
I=I+1
IF (K-I) 40,40,30
40 M=I
CALL FFT(X,M,N-1)
DO 50 I=1,N
A=REAL(X(I))
B=AIMAG(X(I))
S(I)=10.*ALOG10(A*A+B*B)
50 X(I)=CMPLX(S(I),0.)
CALL FFT(X,M,N-1)
DO 60 I=1,N
X(I)=X(I)/XN
60 C(I)=REAL(X(I))
RETURN
END

Figure 1. Cepstrum Pitch Detector Program Listing
was generated with the following parameters:

\[
\begin{align*}
m &= 5, & \omega &= 500 \\
a_n &= 0, & a_0 &= 0 \\
a_1 &= 0, & b_1 &= 1 \\
a_2 &= 0, & b_2 &= 1/2 \\
a_3 &= 0, & b_3 &= 1 \\
a_4 &= 0, & b_4 &= 0 \\
a_5 &= 0, & b_5 &= 1/2.
\end{align*}
\]

and a set of 512 data points was obtained by sampling \( f(t) \) at \( t = (j-1)T \) \((j = 1, 2, \ldots, 512)\) for \( T = .0001 \) (10,000 samples/sec.) A pitch period estimate of 5 msec. was used. The resulting cepstrum is shown in Figure 2. Note that a strong peak occurs at \( t = 2 \) msec., the actual pitch period of the signal.

3. **CEPSTRALLY SMOOTHED LOG SPECTRUM**

As shown by Schafer and Rabiner [2], the spectral envelope can be obtained by cepstrally smoothing the log spectrum. This smoothing is accomplished by low-pass filtering the log magnitude of the DFT. To this end, the cepstrum is multiplied by a low-time filter whose cut-off is less than the pitch period, and then transformed by the DFT to produce the smoothed spectral envelope. The FORTRAN program listing is given in Figure 3.

As an illustration, the unsmoothed log spectrum of the 512 data points of section 2 was first computed, using an FFT algorithm; this is shown in Figure 4. Figure 5 shows the cepstrally smoothed log spectrum.
Figure 2. Cepstrum Example
SUBROUTINE PITCH(S,N,T,C,P,SPEC)

ESTIMATION OF PITCH PERIOD AND SPECTRAL ENVELOPE

REFERENCE: R. W. SCHAFER & L. R. RABINER,
"SYSTEM FOR AUTOMATIC FORMANT ANALYSIS OF VOICED SPEECH.,"

S(I) = DIGITIZED SPEECH SAMPLE
N = NUMBER OF VALUES S(I)
T = SAMPLING PERIOD ( IN SEC. )
P = PITCH PERIOD
SPEC(I) = CEPSTRALLY SMOOTHED LOG SPECTRUM ( SPECTRAL ENVELOPE )

DIMENSION S(N),SPEC(N),C(N)
COMPLEX CC(512)
CALL CEPS(S,N,T,C,P,CC)
TAUMIN = MINIMUM EXPECTED PITCH PERIOD
TAUMIN=.001
TAUMAX = MAXIMUM EXPECTED PITCH PERIOD
TAUMAX=.080
SEARCH THE CEPSTRUM FOR A STRONG PEAK IN THE REGION
TAUMIN < J*T < TAUMAX

JMIN=TAUMIN/T+1.
JMAX=TAUMAX/T+1.
PEAK=C(JMIN)
DO 30 I=JMIN,JMAX
10 IF (PEAK=C(I)) 20,20,30
20 PEAK=C(I)
INDEX=I
30 CONTINUE
P=FLOAT(INDEX-1)*T

LOW - PASS FILTER THE LOG MAGNITUDE OF THE DFT

Figure 3. Cepstrally Smoothed Log Spectrum Program Listing
PI=3.141592653589
T1=.8*P
DT=.1*P
T1PD=T1+DT
DO 50 I=1,N
XI=I-1
   IF (XI*T-T1) .GE. 80,40,40
   IF (XI*T-T1PD) .GE. 50,60,60
50   CC(I)=CC(I)*.5*(1.+C0S(PI*(XI*T-T1)/DT))
   GO TO 80
60   CC(I)=CMPLX(0.,0.)
   CONTINUE
C
C   FIND M SUCH THAT N = 2**M
K=N
   I=0
90   K=K/2
   I=I+1
   IF (K-I) .GE. 100,100,90
100  M=I
C
CALL FFT(CC,M,N,-1)
DO 110 I=1,N
110  SPEC(I)=S.*REAL(CC(I))
C
ADD TO SPECTRUM EQUALIZING CURVE TO SMOOTHED SPECTRAL ENVELOPE
C
DO 140 K=1,N
   W=FLOAT(K-1)/FLOAT(N)*T)
   IF (W-.3000.) 120,120,130
120  SPEC(K)=SPEC(K)+10.*C0S(PI*W/3000.)
   GO TO 140
130  SPEC(K)=SPEC(K)+10.
140  CONTINUE
C
RETURN
END

(Cont'd) Figure 3. Cepstrally Smoothed Log Spectrum Program Listing
4. **CHIRP Z-TRANSFORM**

The Chirp Z-transform (CZT) is a computational algorithm for efficiently evaluating the Z-transform of a sequence of n samples at m points in the Z-plane which lie on circular or spiral contours beginning at an arbitrary point. Its importance to speech analysis stems from its ability to efficiently evaluate the Z-transform off the unit circle, the required contour for evaluation of the DFT. In this way, the contour can be made to pass closer to the poles and zeros of the system, reducing the bandwidths and sharpening the transfer function. A detailed description is given in references [2] and [3].

The FORTRAN program listing of the CZT is shown in Figure 6. Subroutine CZT must be used in conjunction with three other subroutines: FFT, CONTOR, and POWER. The FFT used in this case is algorithm I of SDC TM-4857/100/00, "A Comparison of FFT algorithms". Subroutine CONTOR establishes an appropriate contour over which to evaluate the CZT. Figure 7 is a listing of CONTOR for establishing parameters for a circle of radius .95. Subroutine POWER, given in Figure 8, is used to replace the computation of high powers of the exponential function by iterated multiplications. This has yielded slightly more stability in the code. However, even in its double-precision form (Figure 9), the CZT has been found to be unstable for as few as 16 data points.
SUBROUTINE CZT(N,X,M,A,W)
C
CHIRP Z-TRANSFORM
C
REFERENCE: L.R. RABINER, R.W. SCHAER & G. M. RADER,
"THE CHIRP Z-TRANSFORM AND ITS APPLICATION ",
C
N = NUMBER OF INPUT DATA POINTS X(I)
X = INPUT ARRAY OF COMPLEX DATA POINTS
M = NUMBER OF OUTPUT POINTS DESIRED
A,W = PARAMETERS DETERMINING THE CONTOUR OVER WHICH
THE CZT IS TO BE EVALUATED (SEE LISTING OF SUBROUTINE CONTO)
C
COMPLEX X(512),A,W,V(512)
COMPLEX C,D,PWR
C
FIND THE SMALLEST POWER OF TWO WHICH IS GREATER THAN OR EQUAL
TO N+M-1
C
K=N+M-1
I=1
5 K=K/8
IF (K-1) 10,80,10
10 I=I+1
GO TO 5
20 L=8**((I+1)
DO 30 J=1,N
XJ=J-1
CALL POWER(W,XJ/2,PWR)
C=PWR/A
CALL POWER(C,XJ,PWR)
30 X(J)=PWR*X(J)
DO 40 J=N+1,L

Figure 6. Chirp Z-Transform Program Listing
40  X(J)=CMPLX(0.,0.)
    CALL FFT(X,I+1,-1)
    DO 50 J=1,M
        XJ=J-1
        CALL POWER(W,-XJ*XJ/2.,PWR)
    50       V(J)=PWR
    IF (L-(N+M-1)) 55,68,55
    55       DO 60 J=M+1,L-N+1
    60       V(J)=CMPLX(0.,0.)
    68       XL=L
    DO 70 J=L-N+2,L
        XJ=J-1
        CALL POWER(W,-(XL-XJ)*(XL-XJ)/2.,PWR)
    70       V(J)=PWR
    CALL FFT(V,I+1,-1)
    DO 80 J=1,L
    80       V(J)=V(J)*X(J)
    CALL FFT(V,I+1)
    DO 90 J=1,L
    90       V(J)=V(J)/XL
    DO 100 J=1,M
        XJ=J-1
        CALL POWER(W,XJ*XJ/2.,PWR)
    D=PWR
    100      X(J)=V(J)*D
    RETURN
    END

(Cont'd) Figure 6. Chirp Z-Transform Program Listing
SUBROUTINE CONTOR(NPTS, T, BF, BW, AO, THO, WO, PHI0)

C COMPUTES AN APPROPRIATE CONTOUR ON WHICH TO EVALUATE THE CZT

C INPUT: NPTS = NUMBER OF VALUES OF CEPSTRUM TO BE USED
T = SAMPLING PERIOD (IN SEC.)
BF = LOWEST FREQUENCY OF REGION OVER WHICH THE
NARROW-BAND FREQUENCY ANALYSIS IS TO BE PERFORMED
BW = BANDWIDTH OF REGION

C OUTPUT: M = NUMBER OF OUTPUT VALUES TO BE COMPUTED BY CZT
AO = RADIUS OF CONTOUR
THO: CENTERS THE ANALYSIS ON THE FREQUENCY REGION
OF INTEREST
WO: TAKEN HERE TO BE = 1 TO MAKE THE CONTOUR AN ARC
OF A CIRCLE
PHI0: DETERMINES THE FREQUENCY SPACING OF THE SPECTRAL
SAMPLES

C
TWOP1=6.283185317178
AO=EXP(-.0314)
THO=BF*T/TWOP1
WO=1.
PHI0=BW*T/(FLOAT(M-1)*TWOP1)
RETURN
END

Figure 7. Subroutine CONTOR Program Listing
SUBROUTINE POWER(Z, A, PWR)
C
COMPUTES Z**A FOR LARGE VALUES OF A
C
DOUBLE PRECISION Z, W, P, X, PWR, A, R
IF (A-20.)* 5, 5, S
5 PWR=Z**A
RETURN
C
REPRESENT A AS A = 10*N + R, WHERE R IS LESS THAN 10
8 N=A/10.
R=A-10.*FLOAT(N)
W=Z**10
P=Z**R
C
DETERMINE WHETHER N IS EVEN OR ODD:
C
IF EVEN, FIND K SUCH THAT N = 2*K
C
IF ODD, FIND K SUCH THAT N = 2*K + 1
C
K=N/2
M=MOD(N, 2)
X=W
IF (K.EQ.1) GO TO 25
DO 20 I=1, K-1
20 W=W*X
25 PWR=W*X
IF (M) 40, 40, 30
30 PWR=PWR*X
40 RETURN
END

Figure 8. Subroutine POWER Program Listing
SUBROUTINE CZT(XR, XI, N, M, AO, THO, W0, PHO)
DIMENSION YR(128), YI(128), YRS(128), YIS(128)
DIMENSION VR(128), VI(128), VRS(128), VIS(128)
DIMENSION GR(128), GI(128), XR(128), XI(128)
DOUBLE PRECISION YR, YI, PHO, VR, VI
DOUBLE PRECISION FJ, XP, AO, W0, PWR, PR, PI, TWP0I, ARG
PI = 3.14159265
TWP01 = 6.2831853
K = N + M - 1
I = 1
5 K = K / 2
IF ( K - 1) I.C. 20, 10
10 I = I + 1
GO TO 5
20 L = 2**(I + 1)
DO 30 J = 1, N
FJ = J - 1
XP = FJ * FJ / 8.
CALL POWER(AO, - FJ, PWR)
CALL POWER(W0, XP, PR)
ARG = PI + FJ*(FJ + PHO + 2.* THO)
ARG = DMOCK(ARG, TWP01)
YR(J) = PWR * PR * XR(J) * DCOS(ARG)
YI(J) = PWR * PR * XR(J) * DSINC(ARG)
30 CONTINUE
DO 40 J = N + 1, L
YR(J) = 0.
YI(J) = 0.
40 CONTINUE
DO 45 J = 1, L
YRS(J) = SNGL(YR(J))
YIS(J) = SNGL(YI(J))
45 CONTINUE
CALL FFT(YRS, YIS, I + 1, L, -1)
DO 50 J = 1, M
FJ = J - 1
XP = - FJ * FJ / 8.
CALL POWER(W0, XP, PWR)
ARG = PI + FJ * FJ + PHO
ARG = DMOCK(ARG, TWP01)
VR(J) = PWR * DCOS(ARG)
VI(J) = - PWR * DSINC(ARG)

Figure 9. Double Precision Chirp Z-Transform Program Listing
CONTINUE
IF (L-(N+M-1)) 55, 66, 55
55 DO 60 J=M+1, L-N+1
VR(J)=0.
60 VI(J)=0.
66 FL=L
DO 70 J=L-N+2, L
FJ=J-1
XP=-(FL-FJ)*(FL-FJ)/2.
CALL POWER(W0, XP, PR)
ARG=PI*(-2.*XP)*PHO
ARG=DMOD(ARG, TWOPI)
VR(J)=PR*DCOS(ARG)
VI(J)=PR*DSIN(ARG)
70 CONTINUE
DO 75 J=1, L
VR(J)=SNGL(VR(J))
75 VI(J)=SNGL(VI(J))
CALL FFT(VRS, VI$, I+1, L, -1)
DO 80 J=1, L
GR(J)=VRS(J)*YRS(J)-VIS(J)*YIS(J)
80 GI(J)=VRS(J)*YIS(J)+VIS(J)*YRS(J)
CALL FFT(GR$, GI, I+1, L, -1)
DO 90 J=1, L
GR(J)=GR(J)/FL
90 GI(J)=GI(J)/FL
DO 100 J=1, M
FJ=J-1
XP=FJ*FJ/2.
CALL POWER(W0, XP, PR)
ARG=PI*FJ*FJ*PHO
ARG=DMOD(ARG, TWOPI)
XR(J)=PR*GR(J)*DCOS(ARG)-GI(J)*DSIN(ARG)
XI(J)=PR*GI(J)*DCOS(ARG)+GR(J)*DSIN(ARG)
100 CONTINUE
RETURN
END

(Cont'd) Figure 9. Double Precision Chirp Z-Transform Program Listing
REFERENCES

