

HDL-TM-90-23

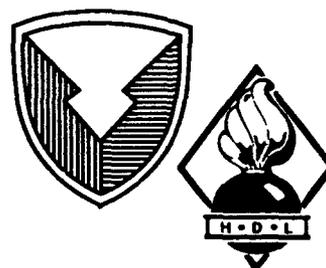
November 1990

The Fast Hartley Transform

by Mark Hyosun Mar

AD-A230 370

DTIC
ELECTE
DEC 26 1990
S B D
60



U.S. Army Laboratory Command
Harry Diamond Laboratories
Adelphi, MD 20783-1197

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.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

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

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 1990	3. REPORT TYPE AND DATES COVERED Final, 1 Nov 88 to 30 Jan 89	
4. TITLE AND SUBTITLE The Fast Hartley Transform			5. FUNDING NUMBERS PE: 6.21.20.A	
6. AUTHOR(S) Mark Hyosun Mar				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Harry Diamond Laboratories 2800 Powder Mill Road Adelphi, MD 20783-1197			8. PERFORMING ORGANIZATION REPORT NUMBER HDL-TM-90-23	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Laboratory Command 2800 Powder Mill Road Adelphi, MD 20783-1145			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES AMS code: 612120.H250011 HDL PR: XE79E3				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report describes the Fast Hartley Transform (FHT), which is twice as fast as the Fast Fourier Transform (FFT), uses only half the computer memory, and is somewhat better for applications such as spectral analysis, signal processing, and convolution. The FHT output is generated with a Fortran computer code and compared to the results of a typical FFT algorithm.				
14. SUBJECT TERMS FHT (fast Hartley transform), FFT (fast Fourier transform), EMP (electromagnetic pulse)			15. NUMBER OF PAGES 27	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	17. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

Contents

	Page
Background	5
1. Introduction	5
2. General Description of FHT	5
3. Comparison	7
4. Performance	9
5. Conclusions	13
Acknowledgments	14
References	14
Distribution	25

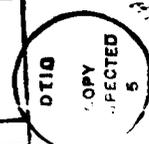
Appendices

A.—Fortran Program FHT.FOR	15
B.—Fortran Program BODE.FOR	21

Figures

1. Generalized EMP waveform	10
2. Fast Hartley transform of EMP waveform	11
3. Fast Fourier transform of EMP waveform	11
4. Calculated double exponential EMP spectrum	12
5. Comparison of figure 4 with figures 2 and 3	12

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



Background

The purpose of this paper is to report the results of testing the Fast Hartley Transform (FHT) and comparing it with the Fast Fourier Transform (FFT). All the definitions and equations in this paper are quoted and cited from the series of references. The author of this report developed a Fortran program which computes the Hartley transform. He tested the program with a generalized electromagnetic pulse waveform and verified the result with the known value.

1. Introduction

Fourier analysis is an essential tool to obtain frequency domain information from transient time domain signals. The FFT is a popular tool to process many of today's audio and electromagnetic signals. System frequency response, digital filtering of signals, and signal power spectrum are the most practical applications of the FFT. However, the Fourier integral transform or the FFT requires the computer resources appropriate to the complex arithmetic operations. On the other hand, the FHT can accomplish the same results faster and requires fewer computer resources.[1] The FHT is twice as fast as the FFT, uses only half the computer resources, and so could be more useful than the FFT in typical applications such as spectral analysis, signal processing, and convolution.[2] This paper presents a Fortran computer program for the FHT algorithm along with a brief description and compares the results and performance of the FHT and the FFT algorithms.

2. General Description of FHT

Equation (1) defines the analytic form of the Hartley transform, and (2) shows its inverse transform, which switches the frequency function back into the time domain [1]:

$$H(f) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(t) \text{cas}(2\pi ft) dt, \quad (1)$$

$$X(t) = \int_{-\infty}^{\infty} H(f) \text{cas}(2\pi ft) df, \quad (2)$$

where $\text{cas}(2\pi ft) = \cos(2\pi ft) + \sin(2\pi ft)$.

The cas function was introduced by R. V. L. Hartley, who first proposed the Hartley transform in 1942 [3].

The above equations are very similar to the Fourier transform, equation (3), and its inverse, equation (4) [1].

$$F(f) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(t)e^{-j2\pi ft} dt , \quad (3)$$

$$X(t) = \int_{-\infty}^{\infty} F(f)e^{j2\pi ft} df , \quad (4)$$

where $e^{j2\pi ft} = \cos(2\pi ft) + j \sin(2\pi ft)$ and $e^{-j2\pi ft} = \cos(2\pi ft) - j \sin(2\pi ft)$. (These are known as Euler's formulas.)

Note the electrical engineering convention of labeling the imaginary unit i as j (i equals the square root of -1).

These four equations deal only with continuous time variables. In the real world, however, signals are sampled at discrete intervals of time. So there are discrete transforms that approximate the Fourier integral and the Hartley integral. The discrete forms of the Hartley transform (DHT) pairs are [1]

$$H(f) = \frac{1}{N} \sum_{t=0}^{N-1} X(t) \text{cas}(2\pi ft/N) , \quad (5)$$

and

$$X(t) = \sum_{f=0}^{N-1} H(f) \text{cas}(2\pi ft/N) . \quad (6)$$

The discrete Fourier transform (DFT) pairs are very similar to the DHT [1].

$$F(f) = \frac{1}{N} \sum_{t=0}^{N-1} X(t)e^{-j2\pi ft/N} , \quad (7)$$

and

$$X(t) = \sum_{f=0}^{N-1} F(f)e^{j2\pi ft/N} . \quad (8)$$

As can be seen in equation (7), the DFT requires $(N - 1)$ complex multiplications and $(N - 1)$ complex additions to compute each output point (the first term in the sum involves $\exp(j * 0) = 1$ and therefore does not require a multiplication). Thus, to compute N output points, $N(N - 1)$ complex multiplications and the same number of complex additions are required. Now each complex multiplication requires four real multiplications and two real additions. Hence, for computa-

tion of all the output points for N -point data, the DFT requires $4N(N-1)$ real multiplications and $4N(N-1)$ real additions. To overcome this computational requirement, the FFT algorithm (Cooley and Tukey algorithm) was developed for the machine computation of a complex Fourier transform [4]. The FFT uses a permutation process to bisect the data sequence until data pairs are reached. The fundamental concept for the permutation process is that it is faster to divide the data set into pairs, compute the transform of the pairs individually, and recombine them to make the entire transform rather than to compute the transform as a whole data set. The Fourier transform of the time domain data set can be obtained by superimposing all permuted data pairs. An N -point FFT, where N is a power of 2, requires $2N \log_2 N$ real multiplications and $3N \log_2 N$ real additions, which means a factor of about 200 times less multiplication for N equals 1024 time data points [5]. Similarly, Bracewell developed a fast algorithm for the Hartley transform. However, to use the algorithm, the decomposition formula that expresses a complete DHT in terms of its half-length subsequences is required. Bracewell has shown the following decomposition formula by application of the shift and the similarity theorems for the DHT [6]. Using an equivalent concept for the Fourier transform, the similar decomposition formula can be defined [1].

$$H(f) = H_1(f) + H_2(f) \cos(2\pi f/N_s) + H_2(N_s - f) \sin(2\pi f/N_s), \quad (9)$$

$$F(f) = F_1(f) + F_2(f)e^{j2\pi f/N_s}, \quad (10)$$

where N_s is the number of elements in the half-length sequence, and thus $N_s = N/2$ for a data set of N elements

As can be seen, there is one important difference between equations (9) and (10). While the FFT decomposition formula is symmetric, the FHT decomposition formula is asymmetric because of the sine coefficients on both $H(f)$ and $H(N_s - f)$. This asymmetric matrix processing requires special handling such as retrograde indexing for computer implementation. The retrograde indexing behavior can be described by "using an independent variable as an index for the elements multiplied by the sine coefficients. This index decreases while the other indexes increase [1]."

3. Comparison

The major difference between the two transform algorithms is the real function \cos in the Hartley transform and the complex exponential term in the Fourier transform. Since real arithmetic is much simpler

than complex computation, the FHT is faster than the FFT and requires fewer computer resources. Furthermore, the complex Fourier spectrum can be obtained from the Hartley transform. It is faster to generate the Fourier transform and power spectrum with the FHT than with the FFT, because the FHT algorithm uses real rather than complex quantities and so requires fewer floating-point operations.

If a function can be expressed uniquely into even and odd parts, then from the even and odd parts the original function can be uniquely reconstructed. Based upon the symmetrical property for the even function and the asymmetrical property for the odd function, the following relationships can be established [6].

Let $H(f) = E(f) + O(f)$, where $E(f)$ and $O(f)$ are the even and odd parts of $H(f)$, respectively. Then

$$E(f) = \frac{H(f) + H(-f)}{2} = \int_{-\infty}^{\infty} V(t) \cos 2\pi f t dt \quad (11)$$

and

$$O(f) = \frac{H(f) - H(-f)}{2} = \int_{-\infty}^{\infty} V(t) \sin 2\pi f t dt . \quad (12)$$

According to equations (11) and (12), the Fourier transform can be obtained from the Hartley transform simply by reflections and additions of the even and odd parts. Since the real part of the FFT is equal to the even part of the FHT, and the imaginary part of the FFT equals the negative odd part of the FHT, the real and imaginary parts of the FFT can be obtained from the FHT according to the following equations [1]:

$$F_r = H(f) + H(N - f) \quad (13)$$

$$F_{im} = H(f) - H(N - f) , \quad (14)$$

where F_r is the real portion of the complex Fourier transform, F_{im} is the imaginary portion, and N is the number of elements in the data set.

The power spectrum can be obtained directly from the FHT using the following equation [1]:

$$P_s(f) = [H(f)^2 + H(N - f)^2] / 2 , \quad (15)$$

where P_s is the power spectrum.

While the FFT computes the square of the real and imaginary parts and sums the two values at a given frequency, the FHT squares and sums the two values of the Hartley transform at the positive and negative frequencies. Since the FHT computes the energy content in

the positive and negative frequency domain, a factor of two is required in the above equation.

The convolution theorem is almost identical between the Hartley and the Fourier transforms. Equation (16) summarizes the convolution theorem for the Hartley transform, corresponding to equation (17) for the Fourier transform [1].

$$f_1(t) \otimes f_2(t) = H_1(f) H_{2e}(f) + H_1(-f) H_{2o}(f) , \quad (16)$$

where $H_{2e}(f)$ is the even part of $H_2(f)$ and $H_{2o}(f)$ is the odd part of $H_2(f)$.

$$f_1(t) \otimes f_2(t) = F_1(f) F_2(f) \quad (17)$$

The \otimes symbol denotes the convolution operation.

Note that if one of the functions being convolved is either even or odd, then the convolution theorem for the Hartley transform reduces to the particularly simple form indicated below [1]:

$$f_1(t) \otimes f_2(t) = H_1(f) H_2(f) . \quad (18)$$

4. Performance

The Fortran program, FHT.FOR, is presented in appendix A as developed from the basic program presented by Bracewell [2]. Using the generalized double exponential waveform as a typical electromagnetic pulse (EMP) electric field, a time domain data file is generated. The generalized double exponential EMP electric field time behavior is given by

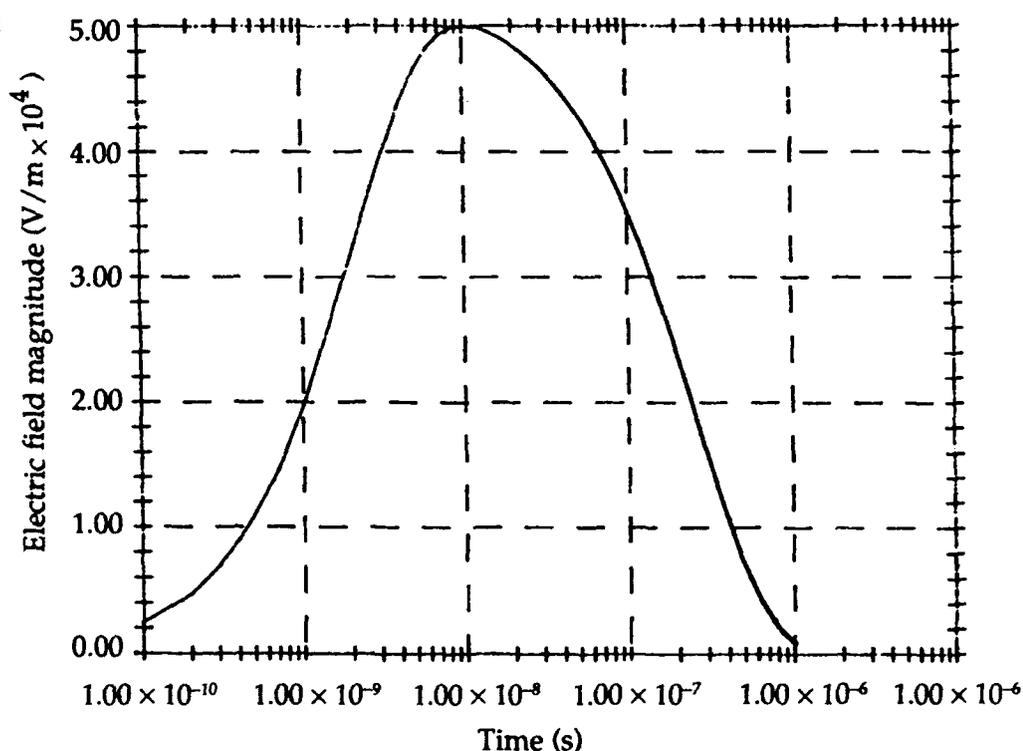
$$E(t) = 5.25 \cdot 10^4 [\exp(-4 \cdot 10^6 t) - \exp(-4.76 \cdot 10^8 t)] , \quad (19)$$

in volts per meter, where t is in seconds. This pulse has a peak value of 50 kV/m, a 10- to 90-percent risetime of about 5 ns, and a time to half-value of about 200 ns [7]. Figure 1 plots the time waveform of this constructed high-altitude EMP electric field. The time domain waveform has 1024 equally spaced points with a time increment of 1 ns. The user can choose any number of data points consistent with the available computer memory size, but must be a power of 2 for algorithm simplicity and additional execution speed. The sample data have to be equally spaced for the FHT algorithm. If more than 1024 data points are required, the FHT.FOR program has to be modified to expand the size of the arrays. The time increment can be controlled also, and 1 ns is used as an example. As the FHT.FOR program executes, the user has an option to choose a format for the transformed

output data. One format has frequency and magnitude only while the other one includes frequency and the real and imaginary parts of the FFT spectrum. Figure 2 shows the FHT.FOR output, and figure 3 shows the FFT output as generated with the HOB0II signal-processing software package [8]. As can be seen from figures 2 and 3, the FFT and FHT results appear to be identical. If drawn on one graph, they would overlay. The real and imaginary parts of the frequency spectrum obtained from the FHT algorithm were verified with the FFT output from HOB0II. An inverse FFT applied to the FHT results reproduces the input waveform which verifies the accuracy of the FHT algorithm.

As can be noticed in figures 2 and 3, both transform algorithms produce a nonphysical behavior at higher frequencies. To explore this deviation and compare these transforms with the original double exponential EMP spectrum, another Fortran program, BODE.FOR, was developed. This program is included in appendix B. The BODE.FOR program computes the double exponential frequency spectrum using the analytic form of Fourier transform for the differences of two exponentials [7]. The calculated spectrum is shown in figure 4, according to the output of BODE.FOR. In figure 5, the FHT and FFT are superimposed on the analytic spectrum. Both transforms trace the calculated spectrum, except for discrepancies at the high

Figure 1. Generalized EMP waveform.



frequency as previously mentioned. Since the double exponential pulse has a risetime of about 5 ns, there is an early time slope discontinuity between the origin and the first sample data point for a 1-ns sample time increment. This situation can be improved if we take

Figure 2. Fast Hartley transform of EMP waveform.

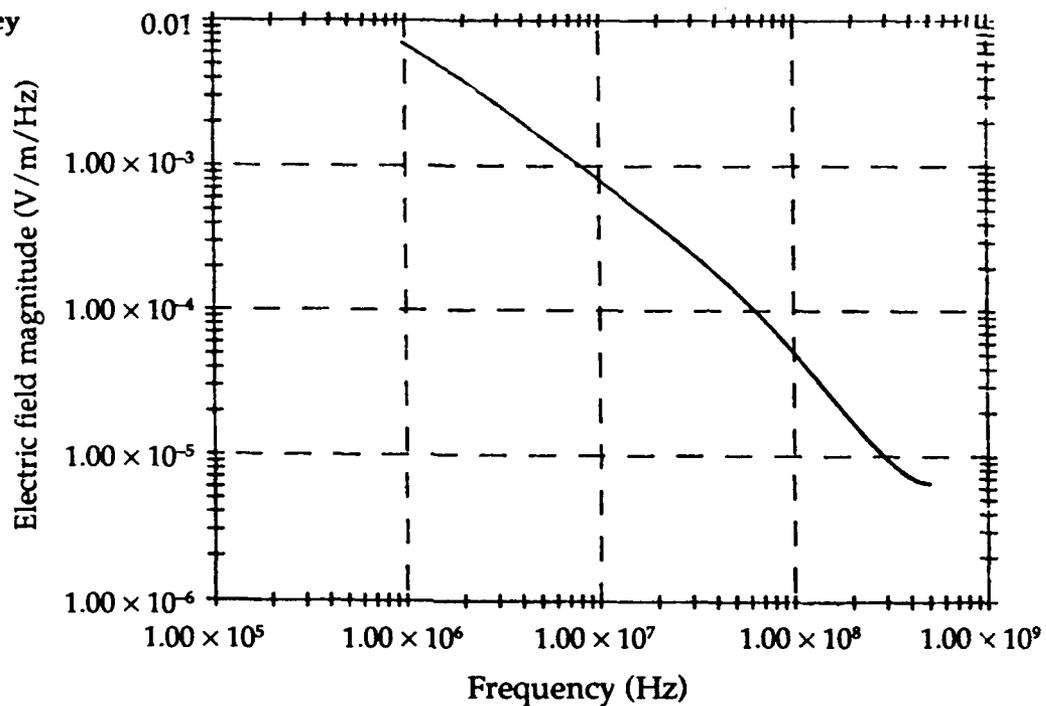


Figure 3. Fast Fourier transform of EMP waveform.

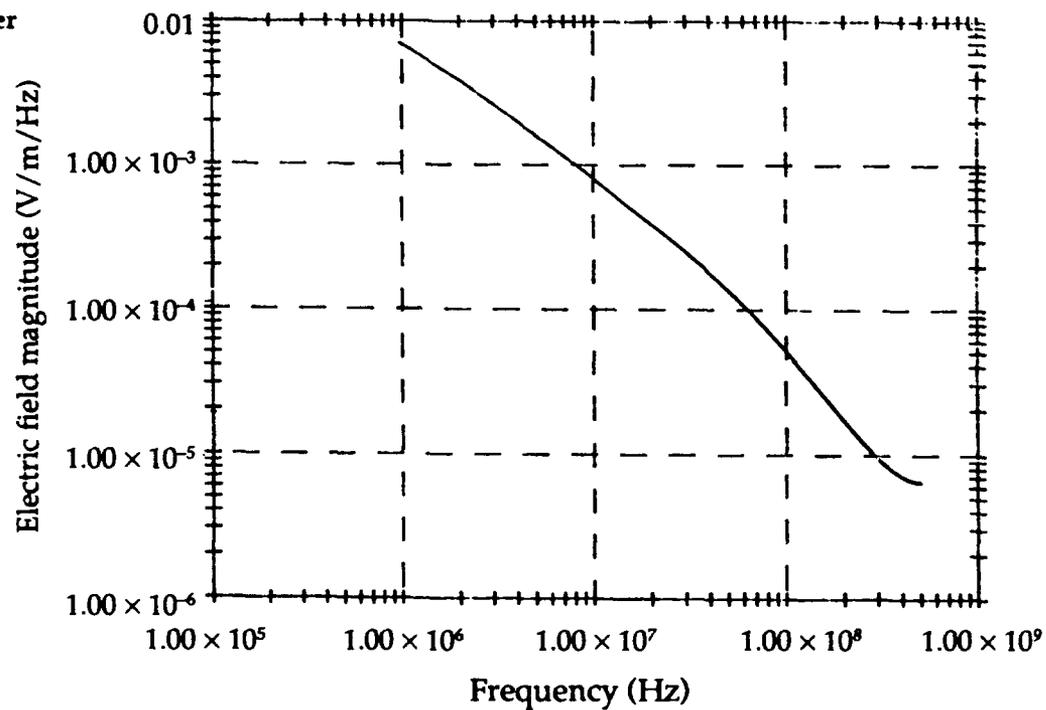


Figure 4. Calculated double exponential EMP spectrum.

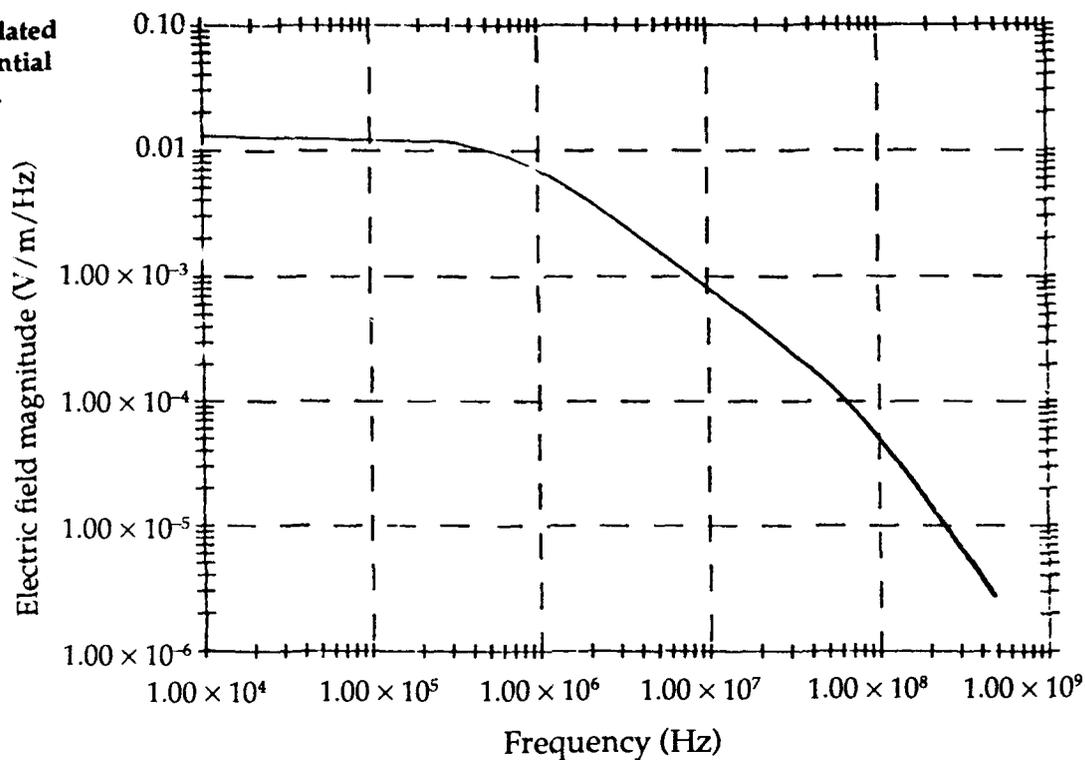
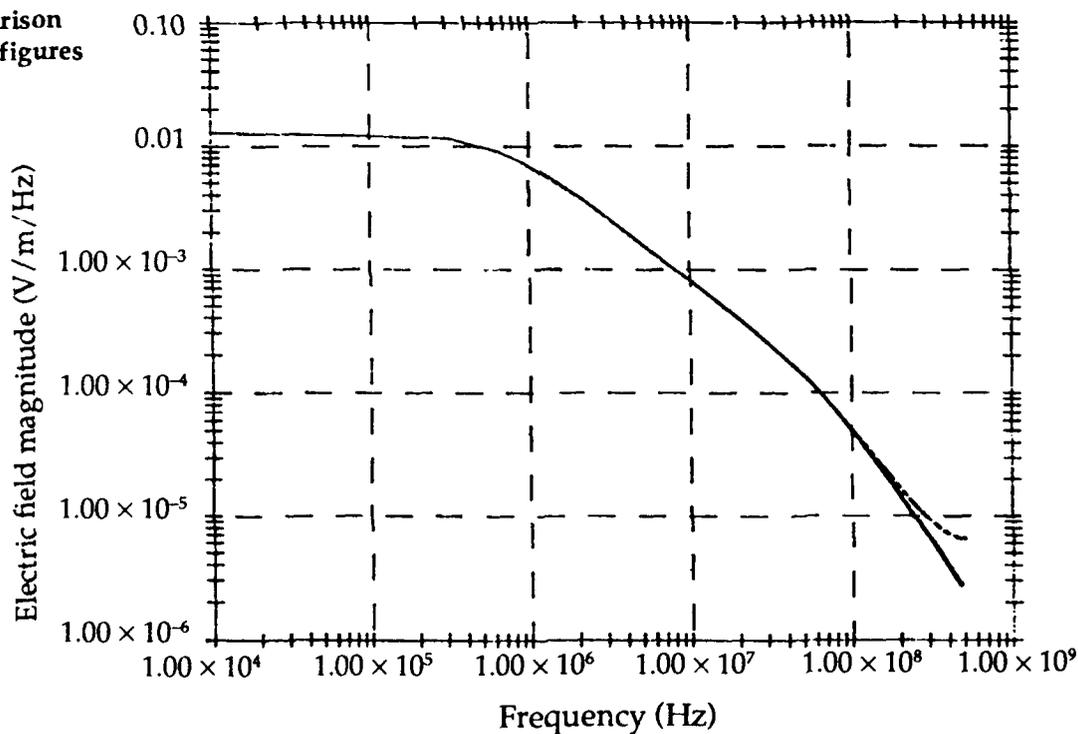


Figure 5. Comparison of figure 4 with figures 2 and 3.



Note:
Solid line, calculated double exponential EMP spectrum.
Dotted line, FFT and FHT.

more data points in the early time, which requires a smaller time increment. Another difference between the analytic frequency spectrum and that obtained with the FHT and FFT transforms is that the low frequency limit is undetermined. In order to obtain the low frequency value of the transformed data, the time domain data set has to have a much larger time window, which would require more time to process the transform. The time increment could also be increased, but would lead to a loss of the high-frequency content. According to the Nyquist sampling theorem, 200,000 time domain data points are required with a time increment of 0.5 ns to generate a double exponential frequency spectrum with the FHT or FFT, which would be similar to figure 4. The Nyquist sampling theorem requires that the sampling rate must be at least twice the highest frequency of interest in the waveform being sampled [9].

When the speed of two transforms is compared, theoretically, the FHT is faster than the FFT; however, it is hard to compare the run time of these two programs for the transform process itself because the programs used in this effort, FHT.FOR and HOB011, are implemented differently. Also, for all practical purposes both codes process the transform within a few seconds.

5. Conclusions

Since the FHT uses only real valued functions, there is no need for complex calculations, which implies faster run-times and less computer memory to process a signal in comparison to a typical FFT algorithm. Finally, the FHT uses fewer operations to transform a given signal, so there are fewer round-off errors.

The example illustrated does not prove that the FHT is superior to the FFT, although it demonstrates that the FHT is fully compatible with the FFT. However, if large amounts of data are being manipulated, there could be a significant difference in speed and the amount of memory required. The FHT offers better performance using fewer computational resources.

The FHT.FOR program can be further developed to perform a convolution that requires only real arithmetic. An FHT convolution algorithm would be faster and easier to implement than a similar algorithm developed for the FFT.

Acknowledgments

I would like to thank Andrew Hermann, who advocated this subject for my first project. I would particularly like to thank William Coburn for his guidance and encouragement throughout this project.

References

Mark A. O'Neill, *Faster Than Fast Fourier*, BYTE (April 1988), pp 293–300.

Ronald N. Bracewell, *The Fast Hartley Transform*, Proceedings of the IEEE, **72**, 8 (August 1984), 1010–1018.

R. V. L. Hartley, *A More Symmetrical Fourier Analysis Applied to Transmission Problems*, Proceedings of the IRE, **30** (1942), 144.

J. W. Cooley and J. W. Tukey, *An Algorithm for the Machine Computation of Complex Fourier Series*, Mathematical Computing, **19** (1965), 297.

D. Ronald Fannin, William H. Tranter, and Rodger E. Ziemer, *Signals and Systems*, MacMillan Publishing Co., Inc., New York (1983), 411.

Ronald N. Bracewell, *The Fast Hartley Transform*, Oxford University Press, Inc., New York (1986), 11.

Bell Laboratories, *EMP Engineering and Design Principles*, New Jersey: Technical Publication Department (1975).

Robert J. Balestri, *HOB0 Installation and User Manual*, Booz, Allen & Hamilton, Inc., Version HDL-1.0.

Robert W. Ramirez, *The FFT Fundamentals and Concepts*, Tektronix, Inc., New Jersey (1985), 115.

Appendix A.—Fortran Program FHT.FOR

\$LARGE

- C This routine generates the fast Hartley transform (FHT)
 C and the complex form of fast Fourier transform (FFT)
 C according to a given time domain function.

```
REAL L2
INTEGER P7,T,U,S2,Q,D,E,SS,SO,P,OPTION
DIMENSION F(10,1024),R(1024),X(1024),FR(1024),RMAG(1024) DIMENSION
S(1024),C(1024),M(I),RR(1024),XX(1024) CHARACTER*20 FILNAM,FILENAM
```

- C Opening the output files.

```
WRITE(*,*) 'TYPE OUTPUT FILENAME FOR THE TIME DOMAIN' READ(*,*)
FILNAM
OPEN(3,FILE=FILNAM,STATUS='NEW')
WRITE(*,*) 'TYPE OUTPUT FILENAME FOR THE FREQUENCY SPECTRUM'
READ(*,*) FILENAM
OPEN(10,FILE=FILENAM,STATUS='NEW')
```

- C Initialization.

```
WRITE(*,*) 'HOW MANY DATA POINTS.#'
WRITE(*,*) 'POWER OF 2 BUT LESS OR EQUAL TO 1024' READ(*,*) NU
P = 0
1 NUM = NU/2
P = P + 1
NU = NUM
IF (NUM. GE .2) GOTO 1
N4 = 2**(P-2)
N2 = N4 + N4
N = N2 + N2
N7 = N - 1
P7 = P - 1
WRITE(*,*) 'WHAT IS YOUR TIME INCREMENT ?' READ(*,*) DTM
WRITE(*,*) 'WHAT ARE YOUR PARAMETERS ?' WRITE(*,*) 'E(t) = EO*(EXP(-
AP*t)-EXP(-BT*t))' WRITE(*,*) 'EO=?, AP=?, BT=?'
READ(*,*) EO,AP,BT
TM = 0.0
```

APPENDIX A

C Generating the time domain waveform.

```
DO 10 I = 0,N7
  FNF = EO*(EXP(-AP*TM)-EXP(-BT*TM)) WRITE(3,*) TM,FNF
  TM = TM + DTM
  DF = 1.0/((N-1)*DTM)
  FR(I) = I*DF
  F(O,I) = FNF
  F(1,I) = FNF
10 CONTINUE
```

C Generating the power of 2 numbers.

```
I = 1
M(O) = 1
M(1) = 2
20 M(1+1) = M(I) + M(I)
  I = I + 1
  IF (I. LT .P) GOTO 20
```

C Get the sin and cos coefficients.

```
PI = 4*ATAN(1.0)
W = 2*PI/N
A = 0
DO 30 I = 1,N
  A = A + W
  S(I) = SIN(A)
  C(I) = COS(A)
30 CONTINUE
```

C Start permutation.

```
J = -1
I = -1
50 I = I + 1
  T = P
40 T = T - 1
  J = J - M(T)
  IF (J. GE .-1) GOTO 40
  J = J + M(T+1)
  IF (I. LE .J) GOTO 50
```

```

T = F(O,I+1)
F(O,I+1) = F(O,J+1)
F(O,J+1) = T
IF (I. LT .N-3) GOTO 50

```

C First stage.

```

DO 60 I = O,N-2,2
    F(I,I) = F(O,I) + F(O,I+1) F(I,I+1) = F(O,I) - F(O,I+1)
60 CONTINUE

```

C Second stage.

```

U = P7
SS = 4
DO 90 L = 2,P7
    S2 = SS + SS
    U = U - 1
    SO = M(U-1)
    DO 100 Q = O,N7,S2
        I = Q
        D = I + SS
        F(L+1,I) = F(L,I) + F(L,D) F(L+1,D) = F(L,I) - F(L,D) K = D - 1
        DO 110 J = SO,N4,SO
            I = I + 1
            D = I + SS
            E = K + SS
            Y = F(L,D)*C(J) + F(L,E)*S(J) Z = F(L,D)*S(J) - F(L,E)*C(J) F(L+1,I) = F(L,I)
            + Y
            F(L+1,D) = F(L,I) - Y
            F(L+1,K) = F(L,I) + Z
            F(L+1,E) = F(L,K) - Z
            K = K + 1
110        CONTINUE
            E = K + SS
100        CONTINUE
            SS = S2
90        CONTINUE

```

C Normalizing the fast Hartley transform's magnitude.

```

RMAG(O) = F(L,O)/(DF*N)

```

APPENDIX A

```
RR(O)   = F(L,O)/(DF*N)
XX(O)   = 0
```

- C Select either magnitude of the FHT
 C or the real and imaginary part of the FFT.

```
WRITE(*,*) 'SELECT THE DESIRED OUTPUT' WRITE(*,*) '1 = MAGNITUDE, 2 =
REAL & IMAGINARY' READ(*,*) OPTION
IF (OPTION.EQ.1) THEN
```

- C The fast Hartley transform's magnitude is derived from the
 C power spectrum

```
DO 120 I = N-1,(N/2)+1,-1
  F(L,I)   = F(L,I)/(N*DF)
  F(L,N-I) = F(L,N-I)/(N*DF)
  RMAG(I) = SQRT((F(L,I)**2 + F(L,N-I)**2)/2) WRITE (10,*) FR(N-I), RMAG(I)
120 CONTINUE
```

- C Get the real and imaginary parts of the fast Fourier
 C transform from the fast Hartley transform.

```
ELSE
DO 130 I = N-1,(N/2)+1,-1
  B      = F(L,N-I)
  RR(I) = F(L,I) + B
  XX(I) = F(L,I) - B
  IF (I.LE.N-I) THEN
    J = I
  ELSE
    J      = N - I
    RR(J) = RR(I)
    XX(J) = -XX(I)
  ENDIF
  R(I) = RR(I)/(2*N*DF)
  X(I) = XX(I)/(2*N*DF)
  WRITE(10,*) FR(N-I),R(I),X(I)
130 CONTINUE
```

```
ENDIF
END
```

Appendix B.—Fortran Program BODE.FOR

C This is a program to compute the analytic frequency spectrum
 C of a double exponential pulse.
 C Use COMPLEX constants, variables, operations, and functions.

C Input variables (all real) :
 C First : Starting frequency
 C Last : Stopping frequency
 C INC : Additive frequency increment

C Intermediate variables, all complex
 C D1 : First factor in denominator
 C D2 : Second factor in denominator
 C These factors are set up for using the CMPLX function, which
 C converts from the form of two REAL values, representing the
 C real and imaginary parts of the complex numbers, to the form
 C of one Fortran COMPLEX number.

```
REAL*8 K, FIRST, LAST, INC, OMEGA, ABSVAL, FRE COMPLEX*16 E, D1,D2
CHARACTER*20 FILNAM
```

C Opening the output file.

```
WRITE(*,*) 'TYPE OUTPUT FILENAME'
READ(*,*) FILNAM
OPEN(3,FILE=FILNAM,STATUS='NEW')
```

C Read parameters, validate.

```
WRITE(*,*) 'FIRST = ?, LAST = ?, INC = ?' READ(*,*) FIRST, LAST, INC
IF (FIRST. GE .LAST) THEN
WRITE(*,*) 'INVALID DATA; PROGRAM ABORTED' STOP
END IF
```

C Set frequency to starting value.

```
OMEGA = FIRST*8*ATAN(1.0)
```

C While OMEGA <= LAST

```
10 CONTINUE
   IF (OMEGA. LE .LAST) THEN
```

APPENDIX B

```
D1 = CMPLX(4.0D6,0MEGA)
D2 = CMPLX(4.76D8,0MEGA)
E = 2.47D13/(D1*D2)
```

C Get complex absolute value, = magnitude of output.

```
ABSBAL = CDABS(E)
FRE = OMEGA / (8*ATAN(1.0)) WRITE(3,*) FRE,ABSVAL
```

C Incrementing frequency.

```
FRE = INC + FRE
OMEGA = FRE*8*ATAN(1.0)
GOTO 10
END IF
END
```

Distribution

Administrator
Defense Technical Information Center
Attn DTIC-DDA (2 copies)
Cameron Station, Building 5
Alexandria, VA 22304-6145

Director
Defense Communications Agency
Attn Command & Control Center
Washington, DC 20305

Director
Defense Communications Engineering
Center
Attn Technical Library
1860 Wiehle Ave
Reston, VA 22090

Director
Defense Nuclear Agency
Attn TITL, Technical Library Div
Washington, DC 20305

Department of the Army
Concept Analysis Agency
8120 Woodmont Ave
Bethesda, MD 20014

Director
U. S. Army Ballistic Research Laboratory
Attn Director, SLCBR-D
Attn Experimental Fabrication
Div, SLCBR-EF
Attn Foreign Intelligence, SLCBR-DD-F
Attn Launch-Flight Div, SLCBR-LF
Attn Safety & Health, SLCBR-DD-H
Attn Security, SLCBR-DD-S
Attn SLCBR-DD-T (STINFO)
Attn Systems Engineering & Concepts
Analysis Div, SLCBR-SE
Attn Terminal Ballistics Div,
Attn Vulnerability/Lethality
Div, SLCBR-VL
Aberdeen Proving Ground, MD 21005

Commander
U. S. Army Comm-Elec Engr
Instal Agency
Attn Tech Lib
Ft. Huachuca, AZ 85613

Commander
U. S. Army Communications
Command
USA COMMO Agency, WS
White Sands Missile Range, NM 88002

Commander
U. S. Army Communications Research &
Development Command
Attn DRCPM-ATC, Ofc of the PM
Attn DRCPM-MSCS, Ofc of the PM Multi-
Service Communications Sys
Attn DRCPM-TF, Ofc of the PM
Tactical Fire Direction Sys/Field
Artillery Tactical Data Systems
(TACFIRE/FATDS)
Attn DRDCO-COM, Center for
Communications Sys
Attn DRDCO-PPA, Plans, Programs
& Analysis Dir
Attn DRDCO-SEI, Sys Engineering
& Integration (CENSEI)
Attn DRDCO-TCS, Center for
Tactical Computer Sys
Ft. Monmouth, NJ 07703

Chief
U. S. Army Communications Sys Agency
Ft. Monmouth, NJ 07703

Commander
U. S. Army Communications-Electronics
Materiel Readiness Command
Ft. Monmouth, NJ 07703

U. S. Army Electronics Technology
& Devices Laboratory
Attn SLCET-DD
Ft. Monmouth, NJ 07703

Commandant
U. S. Army Engineer Center & School
Attn Library
Ft. Belvoir, VA 22060-5291

Director
U. S. Army Materiel Systems
Analysis Activity
Attn AMXSY-MP
Attn X5 (W3JCAA)
Aberdeen Proving Ground, MD 21005

Commander
U. S. Army Missile & Munitions
Center & School
Attn ATSK-CTD-F
Redstone Arsenal, AL 35809

Commander
U. S. Army Ordnance Center &
School Library
Bldg 3071
Aberdeen Proving Ground, MD 21005

Distribution (cont'd)

U. S. Army Satellite Communications
Agency
CORADCOM
Ft. Monmouth, NJ 07703

Commander
U. S. Army Signal Center & Ft. Gordon
Attn ATZHCE, Dir of Communications &
Electronics
Attn ATZHTD, Dir of Combat
Developments
Attn ATZHTD-A-AV, C-E
Management/Audio-Visual
Branch
Attn ATZHTD-A-C, CTAD Branch
Attn ATZHTD-A-DA, C-E Data/
Avionics Branch
Attn ATZHTD-A-M, C-E
Maintenance Branch
Attn ATZHTD-D, Design &
Development Div
Attn ATZHTD-D, Doctrine Branch
Ft. Gordon, GA 30905

Director
U. S. Army Signals Warfare Center
Attn DELSW-CAC, Project Manager—
Control & Analysis Centers Office
Attn DELSW-CE,
Communications/EW Div
Attn DELSW-DT, Tactical Data
Systems Div
Attn DELSW-EE, Electronics/EW
Division
Attn DELSW-GD, General Support
Systems Div
Attn DELSW-M, Resource Management &
Plans Division
Attn DELSW-OS, Office of the
Scientific Advisor
Attn DELSW-RA, Analysis &
Applications Div
Attn DELSW-SS, Signal Security Office
Vint Hill Farms Station
Warrenton, VA 22186-5000

Commandant
U. S. Army War College
Attn Library
Carlisle Barracks, PA 17013

Commander
Naval Air Development Center
Attn Technical Library
Warminster, PA 18974

Superintendent
Naval Postgraduate School
Attn Library, Code 2124
Monterey, CA 93940

Commander
Naval Surface Weapons Center
Attn DX-21 Library Div
Dahlgren, VA 22448

Commander
Naval Telecommunications Command, HQ
Attn Technical Library
4401 Mass Ave, NW
Washington, DC 20390

Assistant Secretary of the Navy
Research, Engineering, & Systems
Attn Dept of the Navy
Washington, DC 20350

Commander
AF Electronic Systems Division
Attn WO, Dep for Control &
Communications Sys
L. G. Hanscom AFB, MA 01730

Dept of the Air Force, HQ
Attn LT COL Ronald L. Kercher, Chief
6585th Test Group (AFSC) Radar Target
Scatter Facility
Holloman AFB, NM 88330

Commander
HQ, Air Force Systems Command
Attn Technical Library
Andrews AFB
Washington, DC 20334

Superintendent
HQ, U. S. Air Force Academy
Attn Tech Lib
USAF Academy, CO 80840

HQ, USAF/SAMI
Washington, DC 20330

AMES Laboratory
Attn Dept of Energy
IOWA State University
AMES, IA 50011

Director
Applied Technology Laboratory
Attn DAVDL-ATL-TSD, Tech Library
AVRADCOM
Ft. Eustis, VA 23604

Boeing Military Airplane Company
Attn Cletus Sutter
M/S: K80-10 3801 South Oliver
Wichita, KS 67210

Distribution (cont'd)

Catholic University of America
Attn Cardinal Station
PO Box 232
Washington, DC 20017

Director
Electronics Technology & Devices
Laboratory
Attn DELET-M, Microwave &
Signal Processing Devices Div
Ft. Monmouth, NJ 07703

Institute for Telecommunications Sciences
Attn Library
National Telecommunications &
Info Admin
Boulder, CO 80303

Director
NASA
Attn Technical Library
Langley Research Center
Hampton, VA 23665

Director
NASA
Attn Technical Library
John F. Kennedy Space
Center, FL 32889

Director
NASA
Attn Technical Library
Lewis Research Center
Cleveland, OH 44135

National Communications System
Attn Library
Office of the Manager
Washington, DC 20305

Director
National Security Agency
Attn Technical Library
Ft. Meade, MD 20755

Director
Research & Technology
Laboratories (AVSCOM)
AMES Research Center
Moffett Field, CA 94035

Rutgers University
Attn Department of Ceramics
College of Engineering
New Brunswick, NJ 08903

Sandia Laboratories
Attn Livermore Laboratory
PO Box 969
Livermore, CA 94550

Sandia National Laboratories
PO Box 5800
Albuquerque, NM 87185

Commander
Technical Support Activity
Attn DELSD-L, Tech Lib Dir
Ft. Monmouth, NJ 07703

University of California
Attn School of Engineering
Los Angeles, CA 90024

US Army Laboratory Command
Attn Technical Director, AMSLC-TD

USAISC
Attn Record Copy, AMSLC-IM-VA
Attn Technical Reports Branch,
AMSLC-IM-VP (2 copies)

Harry Diamond Laboratories
Attn D/Division Directors
Attn Library, SLCHD-TL (3 copies)
Attn Library, SLCHD-TL (Woodbridge)
Attn Chief, SLCHD-NW (W. Vault)
Attn Chief, SLCHD-NW-CS (L. Jasper)
Attn Chief, SLCHD-NW-E (J. Ingram)
Attn Chief, SLCHD-NW-EH (J. Beilfuss)
Attn Chief, SLCHD-NW-EP (J. Miletta)
Attn Chief, SLCHD-NW-ES (T. Bock)
Attn Chief, SLCHD-NW-P (J. Corrigan)
Attn Chief, SLCHD-NW-RP (J. McGarrity)
Attn Chief, SLCHD-NW-RS (F. Jack Agee)
Attn Chief, SLCHD-NW-TN (S. Share)
Attn Chief, SLCHD-NW-TS (J. Halpin)
Attn Mark Mar (15 copies)