National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

DREA CR/97/407

PROGRAM **TPULSE** FOR GENERATING SIMULATED ACOUSTIC DATA

by
David R .Chang

EVANS COMPUTER APPLICATIONS Ltd.,
6424 Norwood Street,
Halifax, NS B3L 2L3

CONTRACTOR REPORT

Prepared for

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

DEFENSIO PER SCIENTIAM

Canada

National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

DREA CR/97/407

PROGRAM **TPULSE** FOR GENERATING SIMULATED ACOUSTIC DATA

by
David R .Chang

EVANS COMPUTER APPLICATIONS Ltd.,
6424 Norwood Street,
Halifax, NS B3L 2L3

Scientific Authority D.Vance Crowe

W7707-5-3329
Contract Number

March 1996

CONTRACTOR REPORT

Prepared for

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

DEFENSIO PER SCIENTIAM

Canada

ABSTRACT

This manual describes the program TPULSE which produces test data for verifying the operation of DREA time series and frequency domain analysis programs. TPULSE generates simulated acoustic time series data consisting of continuous wave (CW) signals, frequency modulated (FM) signals, and/or Gaussian noise. The output is written to standard DREA files on tape or disk.

The software is designed for use with a DEC Alpha AXP computer using the OpenVMS operating system.

RÉSUMÉ

Le manuel décrit le programme TPULSE, qui produit des données d'essai destinées à la vérification du fonctionnement des programmes d'analyse de séries chronologiques et d'analyse dans le domaine fréquence du CRDA. Le programme TPULSE génère des données acoustiques simulées, en séries chronologiques, comprenant des signaux en onde entretenue (CW), des signaux modulés en fréquence (FM) et/ou du bruit gaussien. La sortie est écrite dans des fichiers standard CDRA, sur bande ou sur disque.

Le logiciel est conçu pour être utilisé sur un ordinateur DEC Alpha AXP se servant du système d'exploitation OpenVMS.

TABLE OF CONTENTS

I. OPERATIONAL SECTION

1.0 PURPOSE

The purpose of the DEC C program TPULSE is to generate test signals consisting of simulated acoustic time series data.

2.0 HARDWARE ENVIRONMENT

The program was written for a DEC VAX or Alpha AXP computer. Time series data can be output to disk or magnetic tape.

3.0 SOFTWARE ENVIRONMENT

The program was written in DEC "relaxed" ANSI-89 C for use under the VMS or OpenVMS operating system. The relaxed ANSI C compiler follows the ANSI C standard, but also accepts additional Digital keywords and predefined macros that do not begin with an underscore.

Usage is made of disk I/O routines written by Bruce Skinner of DREA. Filtering routines written by L. Rabiner et al are used (L. Rabiner, 1979). A modified version of a widget library by Richard Culshaw is used.

4.0 OUTLINE OF WORKING

The processing parameters are read from a user-specified ASCII control file. If the program is being run interactively, it can be used to create or update the control file.

The input and output file specifications are read from the standard input unit, which is the user's terminal if the program is being run interactively. The simulated acoustic time series data are output to a standard DREA binary file on tape or disk.

From one to 64 channels of simulated acoustic time series data can be generated. Each channel contains composite data for up to 20 pulses. The same pulses (except for the case of uncorrelated Gaussian noise pulses) appear in all channels, but their start times can differ from channel to channel. Each pulse type can be continuous wave (CW), linear frequency modulated (LFM), exponential frequency modulated (EFM), hyperbolic frequency modulated (HFM), correlated Gaussian noise (CGAUS; correlated meaning correlated across channels), uncorrelated Gaussian noise (UGAUS), or user-defined (USER1). A pulse can have one of the following types of envelopes or windows: square (SQR), Hanning (HAN), 1/8 cycle cosine (TUK), linear rise and linear fall (LINLIN), linear rise and exponential fall (LINEXP), or user-defined shading (SH1).

4.1 Processing Details

4.1.1 Definitions

```
Let:
T         = Time in seconds from the beginning of the file.
TOFF      = Start time of pulse in seconds from the beginning of the
              data file.
T0        = Pulse length in seconds.
TR        = Time in seconds relative to the start of the pulse.
              TR = T - TOFF  where 0 <= TR <= T0.

F0        = Start frequency in Hz for the pulse.
F2        = Stop frequency in Hz for the pulse.
F(TR)     = Frequency in Hz for the pulse at time TR.
PHI(TR)   = Angle in radians for the pulse at time TR.

A0(TR)    = Unshaded pulse amplitude in volts at time TR. For a CW or
              FM pulse, A0(TR) is a constant (time-independent).
A(TR)     = Shaded pulse amplitude in volts at time TR.

S(T)      = Pulse signal in volts at time T.
              For TOFF <= T <= TOFF+T0:
                a) For CW or FM pulses: S(T) = A(TR) * SIN[PHI(TR)]
                                             = A(T-TOFF) * SIN[PHI(T-TOFF)]
                b) For Gaussian noise pulses: S(T) = A(TR)

              For T<TOFF or T>TOFF+T0:    S(T) = 0
```

4.1.2 Pulse Types

Continuous Wave (CW)

```
F(TR)= F0
PHI(TR) = 2.0 * PI * F0 * TR
```

Linear Frequency Modulated (LFM)

```
Let ALPHA = (F2-F0) / T0.
Then F(TR) = F0 + ALPHA * TR
  and PHI(TR) = 2.0 * PI * (F0 + ALPHA * TR/2.0) * TR
```

Exponential Frequency Modulated (EFM)

Let GAMMA = (1.0/T0) * LN(F2/F0)
  where LN is the natural logarithm function.
Then F(TR) = F0 * EXP(GAMMA*TR)
  and PHI(TR) = 2.0 * PI * F0 * (EXP(GAMMA*TR) - 1.0) / GAMMA
  where EXP is the exponential function.

Hyperbolic Frequency Modulated (HFM)

Let BETA = (F2-F0)/(F2*T0).
Then F(TR) = F0 / (1.0-BETA*TR)
  and PHI(TR) = -(2.0*PI*F0/BETA) * LN(1.0-BETA*TR)
  where LN is the natural logarithm function.

Correlated (CGAUS) or Uncorrelated (UGAUS) Gaussian Noise

A0(TR), the unshaded pulse amplitude, has a Gaussian distribution whose RMS value is specified by the user. A finite input response (FIR) filter with a Kaiser (Io-sinh) window design limits the noise to a user-defined bandwidth. The filter coefficients are scaled so that the maximum filter response is approximately 1.

Correlated noise pulses are the same across all channels, whereas uncorrelated pulses differ between channels.

User-Defined Pulse (USER1)

This is defined via function sigus1 in the file TPULSE_USER.C. The default function is for an LFM pulse.

4.1.3 Envelope (Window or Shading)

Let:
A0(TR)   = Unshaded pulse amplitude in volts.
SHADE(TR) = Shading factor.
A(TR)   = Shaded pulse amplitude at time TR. A(TR)=A0(TR)*SHADE(TR).

Square Window (SQR)

SHADE(TR)=1.0

1/N Cycle Cosine or Tukey Window (TUK)

Let:
TFRAC = T0/N where N >= 2.

For TR < TFRAC, SHADE(TR)=0.5*[1.0-COS(PI*TR/TFRAC)]
For TR > T0-TFRAC, SHADE(TR)=0.5 * {1.0-COS[PI*(TR-T0+2.0*TFRAC)/TFRAC] }
For TFRAC <= TR <= T0-TFRAC, SHADE(TR)=1.0.

The Hanning (HAN) window in this program is the same as a 1/2 cycle cosine window.

Window with Linear Rise and Linear Fall (LINLIN)

Let:
TRISE = Rise time for window.
TFALL = Fall time for window.

For TR < TRISE, SHADE(TR) = TR/TRISE
For TRISE <= TR <= T0-TFALL, SHADE(TR) = 1.0
For TR >= T0-TFALL, SHADE(TR) = 1.0 - (TR - T0 + TFALL)/TFALL

Window with Linear Rise and Exponential Fall (LINEXP)

Let:
TRISE = Rise time for window.
TFALL = Fall time for window.
AMPEND = Amplitude at end of fall time as a fraction of the amplitude at the start of the fall time. AMPEND = 0.01.

For TR < TRISE, SHADE(TR) = TR/TRISE
For TRISE <= TR <= T0-TFALL, SHADE(TR) = 1.0
For TR >= T0-TFALL, SHADE(TR) = EXP((LN(AMPEND)/TFALL) * (TR - T0 + TFALL))

where EXP is the exonential function and LN is the natural logarithm function.

User-Defined Window (SH1)

This is defined via function shadesh1 in the file TPULSE_USER.C. The default function is for a 1/4 cycle cosine window.

5.0 INPUTS

5.1 Terminal Input

There are two types of terminal input:

a) Control file editing (available only in interactive mode).
b) Specification of input and output files (available in interactive or batch mode).

5.1.1 Control File Editing

Control file editing is done via a widget (window gadget) type interface. The screen is divided into widgets or boxes. There are three types of widgets for user interaction:

1) A command widget carries out a function.
2) An input widget allows one to input a parameter value.
3) A toggle widget lets the user to choose from a set of parameter values.

A widget is activated by pressing the key (activation character) listed on its left.

To Edit a File:

1. On the first page of widgets, press 'C' (Control file) to specify the name of the input control file. Then press 'E' (Edit control file) to create/update this control file; a second page of widgets will appear.
2. The second page allows channel parameter editing. Edit the channel parameters as desired, then press 'D' (pulse Data) to switch to the third page of widgets.
3. The third page permits editing of pulse parameters. Pressing 'N' (pulse No.) lets one select a particular pulse. Pressing 'T' or 'E' toggles the pulse type and envelope respectively. When done editing the pulse parameters, press 'X' (eXit) to return to page two.
4. On page two, press 'O' (Output control file) if you wish to change the name of the control file to which the parameters will be saved. Press 'X' (save and eXit) to write the parameters to the output control file and to return to page one.

For details on the meaning of the channel and pulse parameters, refer to the description of the control file (section 5.2).

To Process File(s):

On the first page of widgets, press 'C' (Control file) to specify the name of the default control file. Then press 'P' (Process control file), and one will be prompted for input and output file specifications as per section 5.1.2 below.

To Stop the Program:

On the first page of widgets, press 'S' (Stop program).

5.1.2 Input and Output File Specifications

The relevant program messages are listed in section 10.0. The terminal input follows these general rules:

1. Each parameter has a default value as listed in parentheses. To use the default, just press <RETURN>. Otherwise, enter the desired value followed by <RETURN>.

2. To get help about a parameter, enter "?" as the first character of a line followed by <RETURN>.

3. Typing CTRL/Z as the first character of a line will cause the program to stop.

5.2 Input Control File

This ASCII file contains processing selections. There are one or more free format parameters per record, with the parameters separated by blanks or tabs (not commas). Input strings are not enclosed in quotation marks.

5.2.1 Control File Records

In the "Type" column, "Double" means a double precision number and "String[N]" means a string of up to N characters.

Initial records:

Record	Parameter	Type
1	Sampling frequency (FS) in Hz. Must be > 0.0.	Double
2	Length of data file in seconds. The output file will contain at least this amount of data. (It will probably contain more data since an integral number of logical records and blocks of data are output.)	Double
3	Noise filter rolloff in dB/octave (10 to 200). (This is the filter used to limit the bandwidth of Gaussian noise pulses.)	Double
	Stopband attenuation in dB for the noise filter (10 to 200). In practice one may not achieve this much attenuation due to integer roundoff of the output time series data.	Double

(Record 3 is continued on next page.)

Record	Parameter	Type
3 cont'd	Minimum noise filter length (samples; 3 to 600).	Integer
	Maximum noise filter length (samples; 3 to 600). Must be > minimum noise filter length. Larger noise filter lengths allow sharper filter cutoffs at the band edges.	Integer
4	Number of channels (1 to 64).	Integer
5	Number of pulses (NPULSE; 1 to 20) for each channel.	Integer

Following this record are 2*NPULSE records, two for each of the pulses. The records for the Jth (J>=1) such pulse contains:

Record	Parameter	Type
4+2*J	Pulse type: CW for continuous wave. LFM for linear frequency modulated. EFM for exponential frequency modulated. HFM for hyperbolic frequency modulated. CGAUS for correlated Gaussian noise (correlated across all channels). UGAUS for correlated Gaussian noise (not correlated between channels). USER1 for user-defined pulse.	String[5]
	Start frequency (F0) for pulse in Hz.	Double
	Stop frequency (F2) for pulse in Hz. Ignored for a CW pulse. For a CGAUS or UGAUS pulse, the stop frequency must be > the start frequency.	Double
	Type of envelope (window or shading) for pulse: SQR for square window. HAN for Hanning window. TUK for 1/8 cycle cosine window. LINLIN for window with linear rise, flat middle, and linear fall. LINEXP for window with linear rise, flat middle, and exponential fall. SH1 for user-defined window.	String[6]
	Pulse length (T0) in seconds. If < 0 (SQR window only), pulse will extend to the end of the file. (Note that the end of the file may extend beyond the time given in record 2 above.)	Double

Record	Parameter	Type
5+2*J	Rise time in seconds for a LINLIN, LINEXP or SH1 window. Ignored for other window types.	Double
	Fall time in seconds for a LINLIN, LINEXP or SH1 window. Ignored for other window types. For a LINEXP pulse, this is the time for the ampltude to fall to 0.01 of its value at the start of the fall. The rise time + fall time cannot exceed the pulse length.	Double
	Start time (TOFF) of channel one's pulse in seconds relative to the beginning of the file. Must be >= 0.	Double
	Inter-channel delay in seconds for start time of pulses (STDELAY). If the channel one's pulse starts at time TOFF, the channel K pulse will start at time TOFF + (K-1)*STDELAY. STDELAY can be positive or negative, but no pulse can start before the beginning of the file or finish after the end of the file.	Double
	RMS pulse amplitude in dB. SEE WARNING NOTE BELOW. 1) For a CGAUS or UGAUS pulse, this is in dB re 1 volt/sqrt(Hz). Maximum allowable value is 20.0*ALOG10(5.0/(3.0*SQRT(0.5*SAMPLING_FREQ))). 5.0 appears in this equation, since voltages are limited to +/- 5 volts. The factor of 3.0 appears since it is assumed that the maximum noise value will be <= 3.0 * the RMS noise amplitude. 2) For other pulses, this in dB re 1 volt. The maximum allowable value is 20.0*ALOG10(5.0/SQRT(2.0)) = 10.9691 dB. The 5.0 appears in this equation since the time series data must be in the range +/- 5 volts.	Double

WARNING:

If there are overlapping pulses, then the values for the RMS pulse amplitudes should be less than the indicated allowable maxima. The RMS amplitudes should be such that the estimated peak signal voltage is <= 5.0 volts. The estimated peak signal voltage is the sum of the amplitudes (peak values) of the non-noise pulses and three times the RMS voltage values of the noise pulses. Should the estimated peak signal voltage exceed 5.0 volts, the program will issue a warning message recommending how much all the RMS amplitudes should be reduced, and the user has the option of not proceeding with the test data generation. (Note that for simplicity, the program treats all pulses as having SQR shading when computing the estimated peak signal voltage.)

5.2.2 Sample Control File

The following file generates 300 seconds of data for two channels. There are three pulses: a 30 second 100 Hz CW pulse, a 10 second EFM pulse with frequency range 200 to 300 Hz, and a UGAUS pulse which spans the entire file and has a frequency range 0 to 200 Hz.

Note that comments can be included after the parameter values.

```
1000.0                    ! Sampling frequency
300.0                     ! File length (sec)
50.0 55.0 27 300          ! Filter rolloff,  attenuation, min. & max. length
2                         ! No. of channels
3                         ! No. of pulses
CW 100.0 100.0 HAN 30.0        ! Type, freqs, shading, length
0.0 0.0 10.0 -10.0 -4.0        !   Rise, fall, start, delay, amplitude (dB)
EFM 200.0 300.0 LINLIN 10.0    ! Type, freqs, shading, length
1.0 1.0 5.0 1.0 1.0            !   Rise, fall, start, delay, amplitude (dB)
UGAUS 0.0 200.0 SQR -1.0       ! Type, freqs, shading, length
0.0 0.0 0.0 0.0 -30.0          !   Rise, fall, start, delay, amplitude (dB)
```

5.3 User-Defined Functions

The source code for TPULSE is located in directory [CHANG.SONAR] which is currently on disk PS_72: in the DREA FIRM cluster. The command procedure file MFMAKE.COM in [CHANG.SONAR] can be used to compile and link the software.

The user-defined pulse (USER1) is defined via function sigus1 in the file TPULSE_USER.C. The default function is for an LFM pulse.

The user-defined envelope or window (SH1) is defined via function shadesh1 in the file TPULSE_USER.C. The default function is for a 1/4 cycle cosine window.

To link one's own functions to TPULSE, one should:

1. Copy TPULSE_USER.C to your own directory.
2. Modify your version of TPULSE_USER.C as desired.
3. Create subdirectory [.VAX] (VAX machine) or [.AXP] (Alpha AXP machine) which will hold your own version of TPULSE.EXE.
4. Define the logical name DC_DCL (used by MFMAKE.COM) via
   ```
   DEFINE DC_DCL PS_#:[CHANG]
   ```
   where PS_# is PS_72 for FIRM.
5. Create your version of TPULSE.EXE via:
   ```
   @PS_#:[CHANG.SONAR]MFMAKE TPULSE USER_DIR
   ```
   where PS_# is the appropriate disk, and USER_DIR is your directory which contains TPULSE_USER.C. For example on the FIRM cluster:
   ```
   @PS_72:[CHANG.SONAR]MFMAKE TPULSE PS_121:[SMITH.MYSONAR]
   ```

6.0 OUTPUTS

6.1 Terminal Output

This consists of informative messages and prompts for user input. The program messages are listed in section 10.0.

6.2 Output Report File

This ASCII file contains analysis and error information about the data generation. The file usually has the type ".REP".

6.3 Output Time Series Data File

This binary file contains the generated time series in standard DREA format (D. Caldwell, 1987). The data will be real (as opposed to complex) and will be in the form of 16-bit integers.

If the program is going to use a magnetic tape for output, it will automatically allocate and logically mount the tape, if this has not already been done. Therefore it is not necessary to use the VMS DCL commands ALLOCATE and MOUNT prior to running the program.

7.0 OPERATION

7.1 Initiate

The VAX version of TPULSE is in [CHANG.SONAR.VAX] and the Alpha AXP version is in [CHANG.SONAR.AXP]. These files are currently on disk PS_72: in the FIRM cluster.

For example, to run the AXP version of TPULSE on FIRM, enter:
RUN PS_72:[CHANG.SONAR.AXP]TPULSE

7.2 Execute

During program execution, the user is required to provide a number of inputs by way of the terminal keyboard as outlined in sections 5.1.1 and and 10.0.

7.3 Input and Output Data

The user manually inputs data via the terminal. Data are input automatically from the control files.

Data are output automatically to the terminal and to time series data and report files.

7.4 Error Procedures

Data input via the terminal are checked for syntax errors and to ensure that the parameters lie within acceptable range limits. If any errors are detected, an error message is printed on the terminal, and the user is prompted to input via the terminal the correct value for the erroneous parameter.

Data input from files are checked for errors. If an error is detected, an error message is output to the terminal and to the report file (if any).

7.5 Terminate

In interactive mode, this is normally done by activating the "stop program" ('S') widget. In batch mode, one usually terminates the program by answering "Q" when the program asks: "Name of input control file (Q to quit)?". It is also possible to stop the program by typing CTRL/Z whenever it prompts for terminal input as per section 10.0.

It is possible that when the program terminates, the output tape may not have been dismounted and deallocated. If so, one can use the DCL DISMOUNT and DEALLOCATE commands to do this.

8.0 ACCURACY

The time series data are mostly computed using double precision 64-bit floating-point numbers (15 decimal digits precision). The resulting floating point data are converted to 16-bit integers (maximum 5 decimal digits precision).

9.0 EXECUTION TIME

The program prints a value for the CPU time spent generating the data. This time includes the time spent writing the data blocks. (It does not include the time taken to write the header block(s) at the start of the file.)

10.0 PROGRAM MESSAGES

Messages 1 to 5 appear for each time series data file to be created:

No.	Message	Explanation/Response
1	Name of the input control file (Q to quit) (...) ?	Name of the input ASCII control file. If in interactive mode, entering "Q" will cause the widget-based user interface to appear (see section 5.1.1). If in batch mode, entering "Q" will stop the program.
2	Name of output time series file (...) ?	Name of the output time series data file in the form system::device:filename.
3	Using exabyte (8 mm) tape for time series output (...) ?	This message will appear if the user has requested time series output to a magnetic tape. Answer "Y" if you are using an 8 mm tape drive.
4	Initialize time series output tape (...) ?	This message will appear if the user has requested time series output to a magnetic tape. Answer "Y" if you want to initialize the tape by writing a VOL label at its start; warning - this will destroy any information already on the tape. Answer "N" if you don't want to initialize the tape.
5	Name of the output report file (...) ?	Name of the output ASCII report file.
6	... already mounted. Proceed (...) ?	When the program attempted to mount the output tape, it found that the tape had already been mounted. The prompt "Proceed (...) ?" will appear if the program is being run interactively. Answer "Y" to proceed with processing or "N" to abort processing.
7	Volume name for ... tape (...) ?	This message will appear if the user has requested that the output tape be initialized. Enter up to 6 characters consisting of uppercase letters, numbers or the following other characters: ! " % ' ( ) * + , - . / : ; < = > ?

No.	Message	Explanation/Response
8	Error: Output tape ... has no VOL1 label.	The tape has no VOL1 label and the user did not request that the tape be initialized.
9	Warning: Total no. of overflows has exceeded ...	Several overflows have occurred when computing the time series output. One or more of the pulse amplitudes given in the control file are too big.

11.0 PROCEDURE FILE USAGE

The user may wish to run TPULSE via a command procedure file such as the following one. The comments in braces ({ }) are not part of the file.

```
$! RUNNING ON VAX OR ALPHA AXP?
$ARCH_TYPE := VAX
$IF F$GETSYI("HW_MODEL") .GE. 1024 THEN ARCH_TYPE := AXP
$RUN PS_72:[CHANG.SONAR.'ARCH_TYPE']TPULSE
SIGLFM.CON          {Control File}
SIGLFM.DAT             {Time series file}
SIGLFM.REP             {Report file}
SIGEFM.CON          {Control file}
MUA10:SIGEFM.DAT        {Time series file}
Y                       {Yes, this is an 8 mm tape}
N                       {No, do not initialize tape}
SIGEFM.REP              {Report file}
Q                     {Control file - Q to quit}
$EXIT
```

12.0 REFERENCES

[1] D. Caldwell, "A Standard for the DREA Data Descriptor Block", DREA Technical Communication 87/302, January 1987.

[2] J. F. Kaiser, "Nonrecursive Digital Filter Design Using the Io-Sinh Window Function", Proc. 1974 IEEE Symp. Circuits and Syst., pp. 20-24, April 1974; also in "Selected Papers in Digital Signal Processing II", IEEE Press, New York, N.Y., 1975.

[3] L. Rabiner, C. McGonegal, and D. Paul, "Programs for Digital Signal Processing", IEEE Press, New York, N. Y., 1979, pp. 5.2-1 to 5.2-19.

II. TECHNICAL SECTION

21.0 PROGRAM AND DATA FILES

ASCII source files (.C and .H files) associated with the program are in directory [CHANG.SONAR], while binary files (.OBJ and .EXE files) are in [CHANG.SONAR.VAX] for VAX machines and [CHANG.SONAR.AXP] for Alpha AXP machines. These directories are currently on disk PS_72: in the DREA FIRM cluster.

Files in [CHANG.SONAR]

DBIO_ROUTINES.H	Function prototypes for the DBIO disk I/O functions.
MFLIB.H	Function prototypes for library MFLIB.
MTAPEIO.H	Function prototypes for library MTAPEIO.
WIDGET.H	Function prototypes for library WIDGETLIB.
FWFIR.C	FIR filter functions.
FWFIR.H	Function prototypes for fwfir.c.
TPULSE.C	Source for main program TPULSE.
TPULSE.H	Function prototypes for TPULSE.
TPULSE_CON.C	Control file I/O functions.
TPULSE_FUNC.C	Functions used by TPULSE.
TPULSE_WIDGET.C	Widget-based user interface.
TPULSE_USER.C	User functions used by TPULSE.
TPULSE_DRAFT.TXT	User manual for the software in draft form. Program DAISY in [CHANG.UTIL] is used to format the report in final contractor report form.
MFMAKE.COM	Command procedure file to create the program. See section 22.0 below.

22.0 COMPILING AND LINKING THE SOFTWARE

This is done via the MFMAKE command procedures. The resultant .LIS, .OBJ or .EXE files are stored in the [.VAX] or [.AXP]subdirectory depending on whether the procedure is being run on a VAX computer or an Alpha AXP computer.

To use MFMAKE enter @MFMAKE TPULSE.

See section 5.3 above regarding linking TPULSE with user-defined functions.

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA
(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.) EVANS COMPUTER APPLICATIONS LTD. 6424 Norwood Street, Halifax, NS B3L 2L3	2. SECURITY CLASSIFICATION (Overall security of the document including special warning terms if applicable.) Unclassified

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title.)

Program **TPULSE** for Generating Simulated Acoustic Data

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

David R. Chung

5. DATE OF PUBLICATION (Month and year of publication of document.) February 1997	6a. NO. OF PAGES (Total containing information. Include Annexes, Appendices, etc.) 19	6b. NO. OF REFS. (Total cited in document.) 3

6. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

DREA Contractor Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the reseach and development. include the address.)
Defence Research Establishment Atlantic,
P.O. Box 1012,
Dartmouth, N.S. B2Y 3Z7

9a. PROJECT OR GRANT NUMBER (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.) 1.c.c.	9b. CONTRACT NUMBER (If appropriate, the applicable number under which the document was written.) W7707-5-3329
10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number mrst be unique to this document.) DREA CR / 97 / 407	10b. OTHER DOCUMENT NUMBERS (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
(   ) Distribution limited to defence departments and defence contractors; further distribution only as approved
(   ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
(   ) Distribution limited to government departments and agencies; futher distribution only as approved
(   ) Distribution limited to defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic annoucement of this document. This will normally correspond to the Document Availability (11). However, where futher distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

DCDO3 2/06/87

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This manual describes the program TPULSE which produces test data for verifying the operation of DREA time series and frequency domain analysis programs. TPULSE generates simulated acoustic time series data consisting of continuous wave (CW) signals, frequency modulated (FM) signals, and/or Gaussian noise. The output is written to standard DREA files on tape or disk. The software is designed for use with a DEC Alpha AXP Computer using the OpenVMS operating system.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Time Series
Acoustic Test Signals
Calibration
Signal Processing
DEC Alpha

#503376