(1)

National Defence Défense nationale

DATAGUN: DYNAMIC TANK GUN ANALYSIS UTILITY Software User's Manual

by

J.S. Bird
Navigation and Integrated Systems Section
Electronics Division

and

T. Nguyen
S&S Software Ltd.

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 89-27

PCN
0318E

September 1989
Ottawa

ABSTRACT

This note describes a software package for preliminary analysis of digital signals. The program was written in VAX FORTRAN for a tank gun dynamics experiment and performs such techniques as digital filtering, integration, and estimation of power spectra and correlation functions.

RÉSUMÉ

Cette note décrit un logiciel pour l'analyse préliminaire de signaux digitaux. Le programme a été écrit en VAX FORTRAN pour l'analyse d'une expérience sur la dynamique d'un canon de battaille et accomplit des techniques telles que la filtration digitale, l'intégration, et l'estimation des spectres de puissance et des fonctions de corrélation.

EXECUTIVE SUMMARY

This note describes an interactive software package that can be used for preliminary signal processing of recorded sensor data. The program, written in VAX FORTRAN performs such operations as digital filtering, integration, and estimation of power spectra and correlation functions.

It was developed to assist in the preliminary analysis of some recorded sensor data from a set of field trials on the gun dynamics of a Leopard C1 tank. It has provided valuable results that are currently being used in the design of a predictive Kalman filtering algorithm for use in a proposed Dynamic Muzzle Referencing System. ~~This~~ system is currently under joint development by DREV, ~~DREO and Ernst Leitz~~ Canada.

The package is very easy to use, is very fast and can be easily modified to handle different data sets.

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 BACKGROUND

A data analysis utility has been developed to allow simple, yet rapid and interactive analysis of a general data file. It allows combinations of operations such as digital filtering, numerical integration, Fourier transforms (FFT'S), power spectral density and auto-correlation function estimation, and probability density function estimation to be performed on the data and plotted for the user.

It was written to help in the analysis of the Leopard C1 tank Dynamic Muzzle Reference System (DMRS) data [1], but can be modified, through a configuration file, to work on other data files as well. This document describes the package, how to use it, the input and output formats, and how to modify it for other data sets.

1.2 SCOPE OF THIS DOCUMENT

This document is intended for those who will be using or maintaining this utility. The objectives of this document are to allow a new user to run the program and to allow a competent programmer to complete the implementation of the program and to produce the required code for updates or enhancements.

Instructions for running the program are given in Section 4.6 and verbal descriptions of the effects of each of the routines appear in Section 3.1. A description of the assumed format of the input and output data files is given in Sections 4.1 to 4.3 for those users wishing to use the files directly in other programs.

The algorithms for each of the routines are outlined in the appendices. These show the basic steps along with some of the mathematical formulae and calls to IMSL [2] routines used in the implementation.

1.3 PROGRAM TASKS

The program described in this document will be called the Dynamic Tank Gun Analysis Utility (DATAGUN).

DATAGUN is an easy to use, menu-driven utility which guides the user through options and prompts for input. It performs analysis on data which was collected from an experiment on a Leopard C1 tank conducted at the Land Engineering Test Establishment (LETE) in October 1988 [1]. Below is a list of the program tasks implemented in DATAGUN:

a) FILTER
b) INTEGRATE
c) FFT (PSD)
d) CORRELATE
e) PROBABILITY DENSITY
f) RESOLVE - TURRET (a specific routine for the tank data)
g) PLOT

Each of these tasks is described in detail in Section 3.

2.0 COMPUTER EQUIPMENT IDENTIFICATION

DATAGUN will operate on VAX/VMS. The computer programming anguage is VAX-11 FORTRAN. A VT100 or compatible terminal is equired. Tektronix 4014 graphics can be generated for DEC raphics terminals operating in TEK401x mode and an optional ssociated LN03 laser printer. Some specific calls to VMS un-time libraries are made. As well, the IMSL mathematical unction library [2] is used.

3.0 PROGRAM DESCRIPTION

DATAGUN will function as a utility which allows a user to erform data analysis on a generic input data file. This section escribes each of the analysis routines and the resulting output f DATAGUN.

.1 DESCRIPTION OF PROGRAM MODULES

DATAGUN initially displays a main menu which lists all the ptions available for data analysis techniques as follows:

1. Select Binary Data File
2. Filter
3. Integrate
4. FFT (PSD)
5. Correlate
6. Probability Density
7. Resolve - Turret
8. Save buffer
9. Plot
10. Exit

Options 2 (Filter), 3 (Integrate), 4 (FFT/PSD), 5 (Correlate), 6 (Probability Density) and 7 (Resolve - Turret) are dependent on Option 1 (Select Data File); i.e., Option 1 must be selected before any of the above mentioned options. After execution of each selected option, control will be transferred back to main menu, whereupon the processed data is available in a buffer for further processing (e.g. to perform a PSD of a filtered signal). At any time, the user can hit CTRL-Z to reach the previous display and prompt without any action being taken.

3.1.1 **Option 1 - Select Data File Routine**

This subroutine either prompts the user to enter the input data file name or it lists the available binary data files for user selection. It then reads the data from the input file and stores it in a two-dimensional array buffer for later processing. The files are assumed to be in binary FORTRAN format (see Section 4.2). The routine allows the user to select part or all of the data file.

Start and stop records can be specified either with explicit record numbers or in time units (seconds), in which case the correct record numbers are calculated from the default sampling period of the data, dt, specified in the DATAGUN.INC configuration file (Section 4.4).

Specific data fields can be selected. By default all fields are read with time assumed to be in field 1 and all the data signals in each of the remaining fields. The default number of sensor channels and the character strings which contain the names of each of the signals are specified in the DATAGUN.INC file. If only certain fields are required, the user can enter them as, for example, "2,4,7" for channels 2, 4, and 7.

The array buffer that is returned is a two-dimensional array where each row contains time in column 1 and each of the selected channels in the remaining columns. In the example above, where channels 2, 4, and 7 are selected, the first column of the buffer contains time in seconds, and columns 2,3, and 4 of the buffer contain the data from sensor channels 2,4, and 7 respectively that were read from the data file.

3.1.2 **Option 2 - Filter Routine**

This subroutine prompts the user to select the desired columns of the data buffer to be filtered. It then applies a low-pass digital filter algorithm to the selected columns. The default filter currently implemented is a fourth-order, low-pass Butterworth filter with a -6 dB point at 20 Hz. The user can change the order and cutoff frequency of the filter but not the low-pass Butterworth structure. The filtered signals replace the original signals in the data buffer. The filtered buffer can be saved in a file under a name chosen by the user or with the default name (the selected input data filename appended by "_FIL.BIN"). Control is returned to the main menu to allow further processing of the filtered data buffer.

3.1.3 **Option 3 - Integration Routine**

This subroutine applies a simple trapezoidal integration rule to the data buffer. First it prompts the user for the columns to be integrated, then it computes the average of each selected column. The averages are displayed and the user can optionally remove the average or the "linear trend" from the selected columns before integration. Appendix A describes the algorithm for linear trend removal.

The selected columns as modified are then integrated according to the trapezoidal rule and the integrated signals replace the original signals in the data buffer. Finally, the integrated buffer can be saved in a file. The default file name is the name of the selected input data file appended by "_INT.BIN". Control is returned to the main menu to allow further processing of the integrated data buffer.

3.1.4 **Option 4 - FFT Routine**

This subroutine has the capability of computing the following:

1. FFT (Fast Fourier Transform)
2. PSD (Power Spectral Density)
3. XSD (Cross Spectral Density and Coherence Function)

The FFT subroutine first prompts the user to select from the above functions and to identify which column(s) in the data buffer to use. It then computes the mean values of the selected columns and asks the user if the means should be subtracted from the data. A window function is applied to the data to reduce the effect of a finite record length. Currently one of three types of window functions, $W(i)$, can be chosen. These are:

1. Rectangular
2. Tukey
3. Hanning

These are sketched in Appendix B. Typically, the Tukey or Hanning windows should be used since they reduce the high frequency error introduced in the FFT by using no window at all (i.e., a rectangular window). However, the inherent frequency resolution will be somewhat lower [3, Ch. 11] than that obtained when the rectangular window is used.

After the window is selected and applied to the selected columns, the program will call the FFTCF routine, which resides in the IMSL library, to compute the Fast Fourier Transform. The PSD is estimated from the square of the FFT divided by T (the length of the data record in seconds). After the PSD is computed, it can be normalized as required by the Parseval energy theorem (see any reference on signal theory, e.g. [4]) or left as it is. (The normalization may be required to account for the loss of power in the time signal due to the application of the window function - the algorithm for PSD normalization is given in Appendix C.)

The results of the PSD or FFT calculations replace the columns of the buffer which were selected. The time field is also replaced by a frequency field in hertz and the number of records is halved due to the FFT operations.

If the cross power spectral density (XSD) function estimate of two columns is requested by the user, the XSD will be returned in the first column specified and the "coherence" function [3, Ch. 6] will be returned in the second. Again the time field is replaced by frequency. The algorithms for the XSD and coherence function estimates are given in Appendix D.

Finally the FFT subroutine allows the user to save the processed buffer in a file. The default file name is the name of the selected input data file appended by "_FFT.BIN". Control is returned to the main menu to allow further processing of the data buffer.

3.1.5 Option 5 - Correlate Routine

This subroutine has the capability of computing the following:

1. Autocorrelation
2. Cross-correlation

The Correlate subroutine first prompts the user to choose one of the above functions. If "1" (autocorrelation) is selected, the program will prompt for channel selection. If "2" (cross-correlation) is selected, it will prompt for the selection of two channels. The user is then prompted for a window function (see FFT routine). The subroutine will call FFTCF and FFTCB, which reside in the IMSL library, for computing the Fast Fourier Transform and its inverse respectively. The algorithms used in the Correlate routine are outlined in Appendix E.

The results of the auto-correlation replace the data in the selected columns of the buffer. The cross-correlation, if requested, is returned in the first field selected. The time field is replaced by a "time lag" field, which is equivalent to the original time field, except it always starts from 0.

Finally the Correlate subroutine allows the user to save the processed buffer in a file. The default file name is the name of the selected input data file appended by "_COR.BIN". Control is returned to the main menu to allow further processing of the data buffer.

3.1.6 **Option 6 - Probability Density Routine**

This subroutine estimates the probability density function of each selected column in the buffer by calculating its histogram.

The user is prompted to select columns of the data buffer. The maximum and minimum values of the signals in each column are found, and these ranges are divided into several (NBINS=100 by default) intervals called "bins". The number of times the signal falls into each bin is found and stored in a counter for that bin. The histogram is plotted as a normalized bin count versus signal value.

Finally, the histogram is written out to a file. The default file name is the name of the selected input data file appended by "_HST.BIN". and the average and standard deviation is written to a log file called *_HST.LOG. The data buffer is **unchanged** and returned to the main menu.

The format of the *_HST.BIN file is alternate fields of signal value and bin counters, i.e. the FORTRAN write statement is:

```
DO I=1,NBINS
  WRITE(LUN,1010) (SIG_VALUE(I,J),BIN_COUNTER(I,J),J=1,NCHANS)
END DO
```

where j corresponds to the channel selected and i is the bin number. Thus there are NBINS records (100 by default) in this file and 2*NCHAN fields in each record (where NCHAN is the number of channels selected).

The *_HST.LOG file is an ASCII file which lists the mean and standard deviations (1σ level) of each of the selected channels.

3.1.7 Option 7 - Resolve-Turret Routine

This subroutine is specific to the tank sensor configuration. It resolves the turret roof accelerometer measurements into local-level coordinates based on the roof mounted gyro signal. The algorithm for this routine is outlined in Appendix F. The resolved accelerations replace the original acceleration fields in the buffer.

Finally, the routine allows the user to save the processed buffer in a file. The default file name is the name of the selected input data file appended by "_RES.BIN".

N.B.: This routine requires that the turret gyro signal be the first channel selected from the input data file and the turret accelerometers be the last two. Control is returned to the main menu to allow further processing of the data buffer.

3.1.8 Option 8 - Save Buffer Routine

This routine allows the user to save the data buffer at the main menu level. The default file name is the name of the selected data file appended by "_.TMP.BIN". Control is returned to the main menu to allow further processing of the data buffer.

3.1.9 Option 9 - Plot Routine

This subroutine allows the user to plot the data with a locally developed plotting package called GPlot. This subroutine plots the data on Tektronix 4014 type terminals or DEC VT2xx/VT3xx terminals with TEK 4014 emulation. If a graphics terminal is not available, the routine has a fallback mode called VTPlot which will operate on any DEC VT100 series terminal. It has the capability of plotting either directly from the data buffer or from an input data file. When a processed data file (with an underscore in the filename) is plotted, the first record is a header (Section 4.3) and should be skipped over when specifying plotting parameters.

3.2 PROGRAM LOGIC FLOW DIAGRAM

The flow of logic through the main routines is very simple. It is depicted in Figure 1.

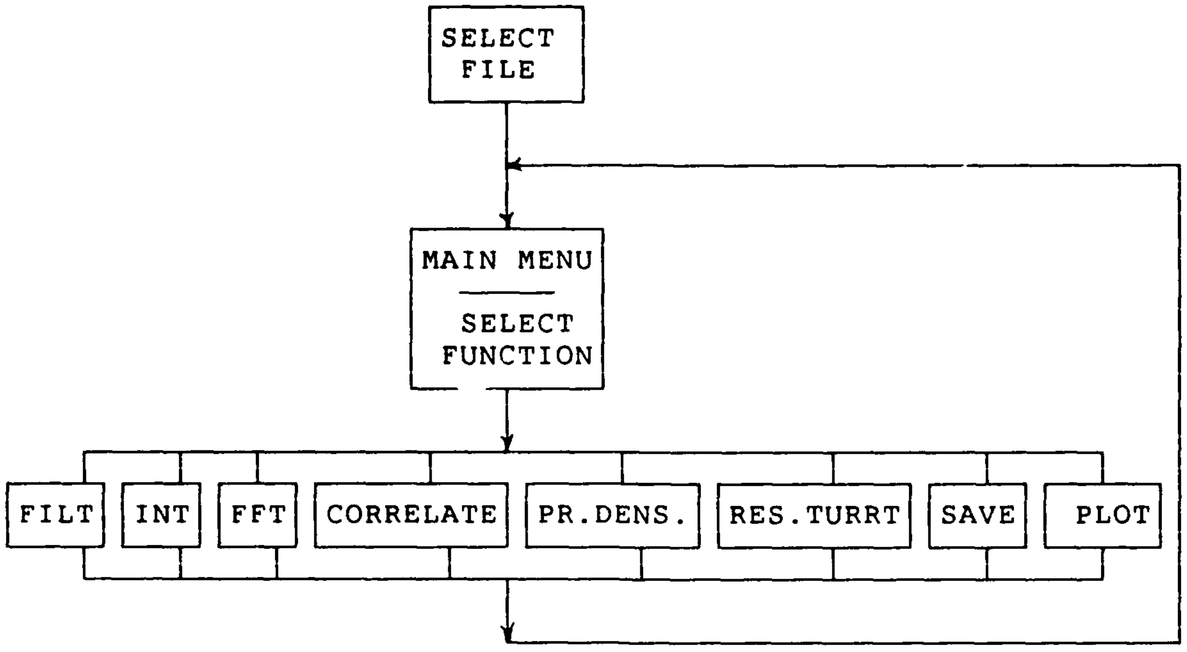

Figure 1: Program Logic Flow Diagram

4.0 OPERATING ENVIRONMENT

This section describes some of the pertinent information the user requires to execute the program and understand the resulting files produced by DATAGUN.

4.1 DATA FILE IDENTIFICATION

Below are the default data file naming conventions and extensions as used by DATAGUN. The notation "xxx" represents the root of the data filename initially chosen by the user.

xxx.BIN	Binary input data file
xxx_FIL.BIN	"Filter" routine output data file
xxx_INT.BIN	"Integrate" routine output data file
xxx_FFT.BIN	"FFT" routine output data file
xxx_COR.BIN	"Correlation" routine output data file
xxx_HST.BIN	"Histogram" routine output data file
xxx_RES.BIN	"Resolve-Turret" output data file
xxx_TMP.BIN	Saved data file at the main menu level

4.2 INPUT/OUTPUT FILE FORMATS

The input data file contains records of data, each record consists of NCHAN+1 fields of FORTRAN 'A4' format binary numbers which represent the signals recorded from NCHAN sensors. The first field contains time in increments of dt seconds; the next NCHAN fields contain the signal data recorded from various sensors. Each field of data occupies four (4) bytes (A4 FORTRAN format). A one-byte blank (' ') carriage control character precedes the data in each record.

<u>Current Configuration of raw data files:</u>

Field 1:	Time (sec)
Field 2:	Turret Pitch rate (mrad/s)
Field 3:	End of barrel pitch rate (mrad/s)
Field 4:	Mantlet Pitch rate (mrad/s)
Field 5:	End of barrel down acceleration (m/s^2)
Field 6:	Gun (center) down acceleration minus End-of-barrel down acceleration (m/s^2)
Field 7:	Turret down acceleration (m/s^2)
Field 8:	Turret forward acceleration (m/s^2)

4.3 HEADER RECORDS

When a processed data file is written out by DATAGUN, the file name always includes an underscore ('_') character. A header record is written out as the first record of the file. When a processed file is later read in by DATAGUN, the required underscore in the filename signifies that the first record is a header record. The header record contains the following information:

- Number of channels
- Number of records
- Index numbers to identify the names of the channels.

4.4 GLOBAL VARIABLES

The FORTRAN "include" file DATAGUN.INC contains some global variables which the user can change. The number of channels, sampling time interval (dt), and the names of the channels can be changed in the DATAGUN.INC file and the program recompiled for different data sets.

4.5 REQUIRED SYSTEM LIBRARY MODULES

Below are the system library modules referred by DATAGUN:

- FFTCF, FFTCB (IMSL library)
- LIB$SPAWN, LIB$WAIT (VAX system run-time library).

4.6 CONDITIONS FOR INITIALIZATION AND EXECUTION

The terminal must be set up with a "nowrap" option at DCL level to ensure proper 132 column operation when plotting on VT100 and compatible terminals. This operation is taken care of in the DATAGUN.COM execution command file. The source code can be compiled and linked and the executable can be run as follows:

```
To compile:        $ for [nguyen]DATAGUN,gplot2,vtplot

To link:           $ link  [nguyen]DATAGUN,gplot2,vtplot, -
                           [bird.draw]draw_on_plot, -
                           term-noecho,
                           imsl/lib, -
                           plot10$library:plot10/lib

To run:            $ @[nguyen]DATAGUN.com
```

(These are correct as of 1 September 89.)

5.0 CONCLUSIONS

In summary, DATAGUN was developed as a modular data analysis program for use in studying sensor data recorded during a series of tank gun dynamics field trials. It has been written with sufficient generality so as to be useful in other applications as well. For the tank gun data, the program has provided valuable results that are currently being used in the design of a predictive Kalman filtering algorithm to be used in a dynamic muzzle referencing system for a future tank fire control system.

REFERENCES

1. J. S. Bird, Measurement of Tank Gun Dynamics in Support of a Dynamic Muzzle Referencing System, DREO Technical Note October 1989 (in preparation).

2. IMSL User's Manual, Volume 2, Chapter 6, Math/library (FORTRAN subroutines for mathematical applications) Version 1.0, April 1987.

3. J. S. Bendat and A. G. Piersol, Random Data - Analysis and Measurement Procedures, 2nd ed. New York: Wiley & Sons, 1986.

4. F. G. Stremler, Introduction to Communications Systems, 2nd ed. Reading, Mass.: Addison-Wesley, 1982.

APPENDIX A

ALGORITHM FOR LINEAR TREND REMOVAL

As described in Chapter 10 of [3], the linear trend is determined from the coefficients of best straight line fit through the data points, y_i, as shown in Figure A-1.

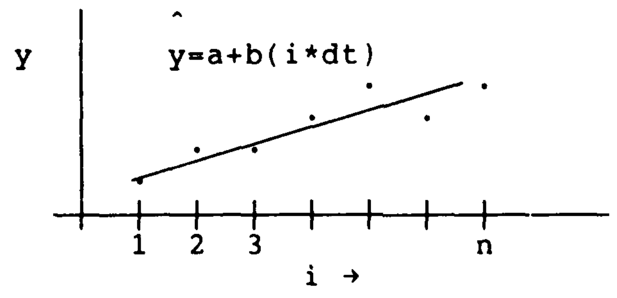

Figure A-1: Best Straight Line

The equation of the best (least squares) straight line, $\hat{y}(t)$, is

$$\hat{y}(t) = a + bt$$

$$\hat{y}_i = a + b(i*dt), \quad i=1,\ldots,n$$

where

$$a = \frac{2(2n+1)S - 6A}{n(n-1)}$$

$$b = \frac{12A - 6(n+1)S}{n\, dt(n-1)(n+1)}$$

n = number of points

dt = sample interval (seconds)

$$S = \Sigma y_i$$

$$A = \Sigma i*y_i$$

To remove the trend from the data:

$$z(t) := y(t) - \hat{y}(t)$$

i.e., $z_i = y_i - \hat{y}_i = y_i - a - b(i*dt), \quad i = 1,\ldots,n$

APPENDIX B

WINDOW FUNCTIONS USED IN THE FFT ROUTINES

The window functions currently implemented include a rectangular window, $W_r(i)$, a Tukey window, $W_t(i)$, and a Hanning window, $W_h(i)$, where

$$W_r(i) = \begin{cases} 1, & 0 \le i \le n-1 \\ 0, & \text{otherwise} \end{cases}$$

$$W_t(i) = \begin{cases} \tfrac{1}{2}(1-\cos(i\pi/a)), & 0\le i\le a \\ 1, & a\le i\le b \\ \tfrac{1}{2}(1+\cos((i-b)\pi/a)), & b\le i\le n-1 \\ 0, & \text{otherwise} \end{cases}$$

$$a = 0.1n$$
$$b = 0.9n$$

$$W_h(i) = \begin{cases} 1 - \cos^2(i\pi/n), & 0 \le i \le n-1 \\ 0, & \text{otherwise} \end{cases}$$

These are sketched in Figure B-1.

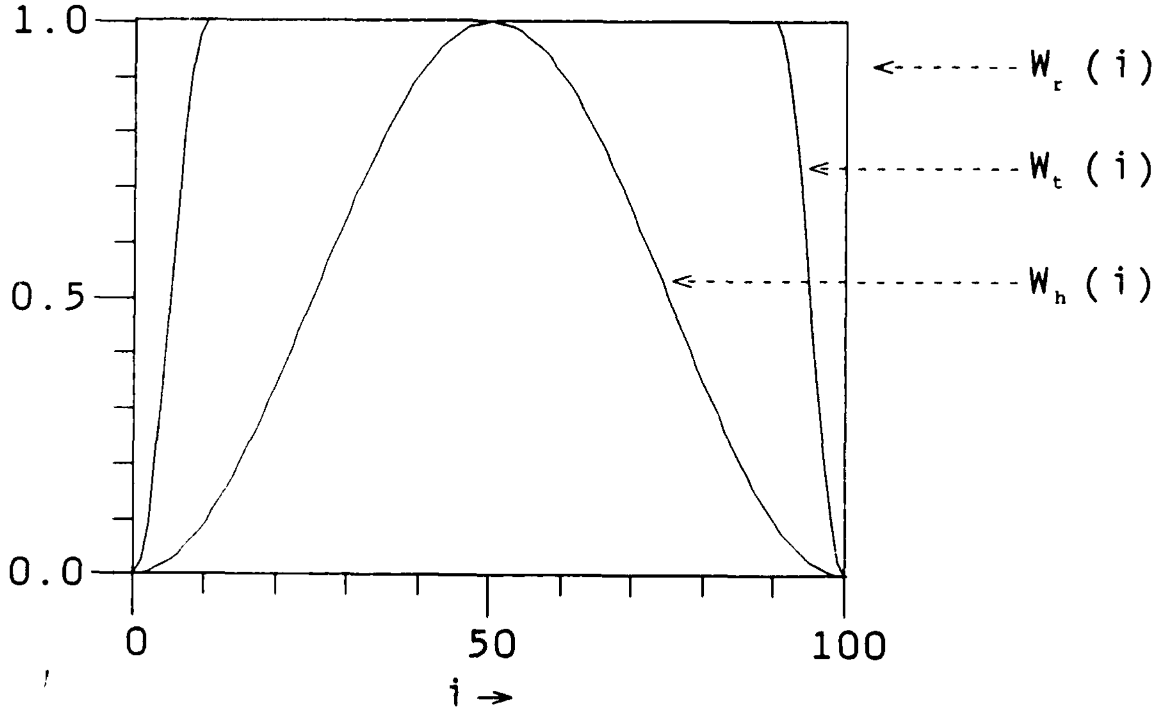

Figure B-1: Window Functions

APPENDIX C

ALGORITHM FOR PSD NORMALIZATION

PSD normalization is carried out in the following steps:

1. Find the area under the square of the time series after the average is removed but before the window is applied.

2. Find the area under the PSD curve.

3. Normalize the PSD by multiplying it by the result of step 1 divided by step 2.

APPENDIX D

ALGORITHM FOR CROSS-SPECTRAL DENSITY AND COHERENCE

As described in Chapter 11 of [3], the algorithms are:

<u>Cross-Spectral Density</u>

1. Select 2 fields of the buffer x(i), y(i), i=0,...,n-1.
2. Apply window and average removal if desired.
3. Load them into one complex array, A:

   $$A(i) = x(i) + j\ y(i) \qquad i = 0,\ldots,n-1$$

   where $j = \sqrt{-1}$

4. Compute the FFT of A, return the result in Z(i), i=0,...,n-1.

5. Compute 2 new complex arrays from the result:

   $$X(i) = \frac{Z(i) + \mathrm{CONJ}(Z(n-i))}{2}$$

   $$Y(i) = \frac{Z(i) - \mathrm{CONJ}(Z(n-i))}{2j}$$

   $i = 0,\ldots,n-1$

   where CONJ() represents complex conjugate, and Z(n) := Z(0).

6. Compute the one-sided cross-spectral density Gxy:

   $$Gxy(k) = \left| \frac{2 * \mathrm{CONJ}(X(k)) * Y(k)}{n\ dt} \right|$$

   $k = 0,\ldots,n/2 - 1$.

7. Return the result, Gxy(k), in the first field specified and the coherence function (below) in the second field. Reduce the number of records by half.

8. Replace the time field (Field 1) by a frequency field in hertz

Coherence Function

1. Select 2 fields of the buffer.
2. Apply window / average removal if desired.

3. Compute the cross-spectral density of these two fields and power spectral density of each field separately.

   Gxy(i) = Cross-Spectral Density
   Gxx(i) = PSD of field 1
   Gyy(i) = PSD of field 2.

4. Compute Coherence function, Γ:

$$\Gamma(i) = \sqrt{\frac{|Gxy(i)|^2}{|Gxx(i)|\,|Gyy(i)|}}$$

$$i = 0,\ldots,n/2 - 1 .$$

APPENDIX E

ALGORITHM FOR AUTO AND CROSS-CORRELATION

As described in Chapter 11 of [3], the algorithms are:

<u>Autocorrelation</u>

1. Select fields.

2. Augment each selected field with n zeros, where

   n = number of records

3. Apply weighting function if requested.

4. Compute FFT of each channel (which now has 2n records). Multiply result by dt; call it X, say.

5. Compute the complex autospectrum, Sxx, of each channel:

$$Sxx(i) = \frac{|X(i)|^2}{(2n)(dt)} \qquad i = 0,\ldots,2n-1.$$

6. Compute the inverse FFT of Sxx for each channel using the IMSL routine FFTCB, call it R.

8. Discard the last n records of the R vector, leaving the first n.

9. Normalize the R vector to obtain the autocorrelation function for each channel, Rxx(i):

$$Rxx(i) = n * |R(i)|/(n-i) \qquad i = 0,\ldots,n-1$$

10. Put Rxx(i) back in the corresponding fields in the buffer.

11. Replace the time field with the time lag (which is essentially the same as the time field except it starts from 0).

Algorithm for Cross Correlation

1. Select two channels, x(i), y(i), i=0,...,n-1
2. Augment each channel with n zeros.
3. Load the first channel into the real part of a complex array, A(i), i=0,...,2n-1, and the second into the imaginary part.
4. Apply desired weighting functions.
5. Compute FFT of A with the IMSL routine FFTCF, call the result Z(i), i=0,...2n-1.
6. Compute the complex cross spectral density, Sxy, as in steps 5 and 6 of Appendix D except the number of point is 2n instead of n.
7. Compute the inverse FFT of Sxy, call it R.
8. Discard the last n records of R, leaving the first n.
9. Normalize the R vector to obtain the cross-correlation function, Rxy(i), for the two channels

   $$Rxy(i) = n * |R(i)|/(n-i) \qquad i = 0,...,n-1$$

10. Return Rxy(i) back in the first channel specified in step 1 in the buffer.
11. Replace the time field with the time lag (which is essentially the same as the time field except it starts from 0).

APPENDIX F

ALGORITHM FOR RESOLVE-TURRET

1. Prompt for initial turret pitch angle, θ_0.

2. Integrate Turret pitch rate:

   $\theta(0) = \theta_0$

   $\theta(i+1) = \theta(i) + \frac{1}{2}(\omega(i) + \omega(i+1))\ dt, \quad 0 \le i \le n-2$

   where

   $\theta(i)$ = turret pitch angle (mrad)

   dt = sampling interval

   ω = measured turret pitch rate

3. Find local-level forward & up accelerations, A_{xl}, A_{zl}

   $A_{xl}(i) = \cos(\theta(i))\ A_{xt}(i) - \sin(\theta(i))\ A_{zt}(i)$

   $A_{zl}(i) = \sin(\theta(i))\ A_{xt}(i) + \cos(\theta(i))\ A_{zt}(i)$

   where

   $A_{xt}(i)$: Measured turret forward acceleration

   $A_{zt}(i)$: Measured turret up acceleration

   $i=0,\ldots,n-1$

4. Replace the measured turret accelerations in the data buffer with the resolved (local-level) accelerations.

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.)

Department Of National Defence
Defence Research Establishment Ottawa
Ottawa, Ontario K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification 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.)

"DATAGUN: Dynamic Tank Gun Analysis Utility - Software User's Manual" (U)

4 AUTHORS (Last name, first name, middle initial)

Bird, Jeffrey S. & Nguyen, T.

5 DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (total cited in document)
September 1989	19	4

7 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.)

Technical Note

8 SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Department of National Defence
Defence Research Establishment Ottawa
Ottawa, Ontario K1A 0Z4

9a PROJECT OR GRANT NO (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)	9b. CONTRACT NO (if appropriate, the applicable number under which the document was written)
PCN 0316E	
10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)
DREO TECHNICAL NOTE 89-29	

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; further 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 announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

UNLIMITED

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

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 offical languages unless the text is bilingual).

This note describes a software package for preliminary analysis of digital signals. The program was written in VAX FORTRAN for a tank gun dynamics experiment and performs such techniques as digital filtering, integration, and power spectrum and correlation function estimation.

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 is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Digital Signal Processing Software
Muzzle Referencing Systems

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM