



# ***ADVANCED CODAS*** **USER'S MANUAL**

***Software Release Level 3***

***Manual Revision I***

## **Copyright Notice**

Copyright © 1988-2009 by Dataq Instruments, Inc. ALL RIGHTS RESERVED. The documentation contained herein is the exclusive property of Dataq Instruments, Inc., except as otherwise indicated and shall not be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise without expressed written authorization from Dataq Instruments, Inc. The distribution of this material outside the company may occur only as authorized by Dataq Instruments, Inc. in writing.

Dataq Instruments' hardware and software products are not designed to be used in the diagnosis and treatment of humans, nor are they to be used as critical components in any life-support systems whose failure to perform can reasonably be expected to cause significant injury to humans.

**REPRODUCTION OF ANY PART OF CALC SOFTWARE, BEYOND THAT PERMITTED BY SECTION 117 OF THE 1976 UNITED STATES COPYRIGHT ACT, WITHOUT PERMISSION OF DATAQ INSTRUMENTS, INC. IS UNLAWFUL.**



241 Springside Dr  
Akron, Ohio 44333 USA  
[www.dataq.com](http://www.dataq.com)  
Telephone: 330/668-1444  
FAX: 330/666-5434

Designed and manufactured in the  
*United States of America*



# TABLE OF CONTENTS

<b>AN INTRODUCTION TO ADVANCED CODAS</b>	6A-1
Common Features	
Waveform Differentiation Features	
Waveform Integration Features	
Waveform Arithmetic Functions	
Waveform Moving Average Features	
Waveform Peak and Valley Capture (PVC) Features	
Waveform Rectification Features	
Report Generator Features	
Waveform Copy Feature	
<b>GETTING READY TO USE ADVANCED CODAS</b>	6A-3
Installing Advanced CODAS Routines	
Accessing Advanced CODAS Routines	
Invoking CALC from COPY AND PASTE	
Invoking CALC Using the CALC HOT KEY FUNCTION	
Invoking Advanced CODAS from DOS	
<b>WORKING WITH CALC MENUS</b>	6A-6
General	
Movement Through CALC Menu Items	
Entering Item Arguments	
Activating Menu Selections	
<b>WAVEFORM ARITHMETIC OPERATIONS USING ADVANCED CODAS</b>	6A-7
General	
WAO Operators and Built-in Functions	
Combining Functions and Operators Into WAO Program Formulas	
Engineering Notation With the WAO Program	
Formula and Calculated Error Detection	
WAO and Dynamic Range Errors	
WAO DOS Command Line and CALC Menu Syntax	
<b>WAVEFORM DERIVATIVES USING ADVANCED CODAS</b>	6A-12
General	
Automatic Engineering Unit Calibration	
Selecting a Smoothing Factor	
Waveform Derivative DOS Command Line and CALC Menu Syntax	
<b>WAVEFORM INTEGRATION USING ADVANCED CODAS</b>	6A-14
General	
Automatic Engineering Unit Calibration	
Extended Features of Advanced CODAS's Integrator	
SUM and PULSE Modes of Integration	
Signal RECTIFICATION Before Integration	
Optional HOLD Feature	
Adjustable RESET INTEGRATION PERIOD	

## ADVANCED CODAS

Selectable LEVEL RESET  
Resetting on an EXTERNAL WAVEFORM EVENT  
Waveform Integral DOS Command Line and CALC Menu Syntax

### **WAVEFORM MOVING AVERAGE USING ADVANCED CODAS** 6A-20

General  
Automatic Engineering Unit Calibration  
Applying Advanced CODAS's Moving Average Program  
Waveform Moving Average DOS Command Line and CALC Menu Syntax

### **WAVEFORM PEAK AND VALLEY CAPTURE USING ADVANCED CODAS** 6A-23

General  
Peak and Valley Capture (PVC) Operation  
Peak Capturing in the Presence of Waveform Artifact  
Good Data Acquisition Practice Leads to Good PVC Performance  
Editing PVC-selected Event Markers  
Peak Capturing Channel Ones's Waveform  
PVC DOS Command Line and CALC Menu Syntax

### **WAVEFORM RECTIFICATION USING ADVANCED CODAS** 6A-29

General  
Automatic Engineering Unit Calibration  
Advanced CODAS's Waveform Rectification Features  
    Positive-only Half-wave Rectification  
    Negative-only Half-wave Rectification  
    Absolute Value  
Waveform Rectification DOS Command Line and CALC Menu Syntax

### **REPORT GENERATION OF PEAK/VALLEY EVENTS USING ADVANCED CODAS** 6A-32

General  
Defining the Parameters Reported by RG  
RG Data File Creation and Storage Formats  
RG DOS Command Line and CALC Menu Syntax

### **WAVEFORM COPY FEATURE USING ADVANCED CODAS** 6A-35

General  
Waveform Copy DOS Command Line and CALC Menu Syntax

# AN INTRODUCTION TO ADVANCED CODAS

The Advanced CODAS package from Dataq Instruments is a waveform integration, differentiation, moving average, peak and valley capture, and rectification utility that operates on an existing waveform data file created by AT-CODAS, WINDAQ/Pro, or WINDAQ/Pro+ data acquisition software.

## **Common Features**

Except for the peak and valley capture function, Advanced CODAS is a waveform-oriented analysis package, meaning that the package accepts waveforms as an input and generates waveforms as an output. The output waveform may be the integral, derivative, mathematical combination, moving average, or a rectified result of the input waveform. Advanced CODAS permits output waveforms to be inserted directly into the original file that contained the input waveform. Two formats are supported: 1) The resulting waveform may be overwritten into an existing waveform channel, thereby destroying the original contents of that channel, or 2) The resulting waveform may be appended to the file containing the input waveform provided the number of channels contained in the file before appending is less than 29 for AT-CODAS and WINDAQ Waveform Browser without the EXP channel multiplexer option, or less than 144 for WINDAQ Waveform Browser with the EXP channel multiplexer option.

## **Waveform Differentiation Features**

Advanced CODAS's waveform differentiation program allows any input waveform to be differentiated with respect to time. Following insertion into the file, the differentiated waveform will be scaled automatically to the engineering units of the input waveform divided by seconds. For example an input waveform with millivolts as units becomes mV/sec when differentiated. When reviewed through the waveform playback program, the differentiated waveform will carry all attributes of an acquired channel. Timing measurements can be made from the differentiated channel, and when cursors and channel annotation are enabled, the value of the differentiated channel may be read directly from the screen in the appropriate units. Filtering of the differentiated waveform is accomplished by allowing the selection of the number of points over which the derivative is calculated. Step sizes from 2 (no filtering) to 64 may be chosen.

## **Waveform Integration Features**

Similar to differentiation, Advanced CODAS's waveform integration program allows any input waveform to be integrated with respect to time. Following insertion into the file, the integrated waveform will be scaled into the engineering units of the input waveform multiplied by seconds (i.e., millivolts becomes mV•sec). When reviewed through the waveform playback program, the integrated waveform will carry all attributes of an acquired channel. Timing measurements can be made from the integrated channel, and when cursors and channel annotation are enabled, the value of the integrated channel may be read directly from the screen in the appropriate units.

Unlike all other commercially available integration utilities, Advanced CODAS's integration routine allows control over reset conditions and signal polarity. Waveforms may be integrated over bipolar, positive only, negative only, or absolute value transitions relative to zero reference. Reset may be programmed to occur: upon zero crossing of the input waveform, after  $n$  samples (where  $2 \leq n \leq 32,767$ ), at a specified level, or as triggered by a different channel (such as an ECG waveform). Another option allows

0 integrated values to be displayed in a sample/hold fashion where the previous integrated value is held until a new value is available. When this sample/hold feature is enabled, reset transitions of the integrated waveform are not displayed.

## **Waveform Arithmetic Functions**

This operation allows any number of waveforms contained in the data file to be combined mathematically according to a user-defined, arbitrary formula. Supported operations are: +, -, \*, ÷, exp,  $\log_{10}$ ,  $\log_e$ , sqrt, and abs. Another operation supports phase shifting of the input waveform in precise increments.

## **ADVANCED CODAS**

### **Waveform Moving Average Features**

Advanced CODAS's waveform moving average program allows waveform filtering independent of Advanced CODAS's waveform derivative program. It can be used to achieve waveform smoothing in the presence of noise, or with the proper filter factor the mean value of a given periodic waveform may be extracted or subtracted from the input waveform. The moving average program may be applied over 2 to 10,000 samples.

### **Waveform Peak and Valley Capture (PVC) Features**

This Advanced CODAS mode allows the cycle-by-cycle extraction of peak, valley, or peak and valley data points from a periodic waveform. PVC may be applied to a variety of waveforms whenever it is necessary to reduce a large number of continuous waveform data values to one or two values per waveform cycle, representing the maximum, minimum, or maximum and minimum signal excursion within the cycle. For example, an arterial pressure waveform may be peak captured for systolic, diastolic, or systolic and diastolic pressures on a beat-by-beat basis.

Unlike other Advanced CODAS functions, PVC does not generate a waveform as a result. Instead, the input waveform (or any other channel in the file) is tagged by PVC with positive- and negative-going event markers to indicate cyclic peak and valley points respectively. The Advanced CODAS Report Generator (included) is then used to extract waveform values coincident with the event markers, and to generate a report of these values in one of two data storage formats.

### **Waveform Rectification Features**

Another mode of Advanced CODAS allows input waveforms to be converted into positive only, negative only, or absolute value transitions relative to zero reference without integration. The application of waveform rectification is useful for verifying a stable zero reference baseline before integration.

### **Report Generator Features**

The Advanced CODAS Report Generator is used to generate a report of peak and valley detected waveforms. The Report Generator may be configured to report the value of an input waveform coincident with a peak and/or valley tag, the period of the waveform, or its mean value on a cycle-by-cycle, or 2 – 32,767 cycle average. Supported Report generator file formats are a spreadsheet/ASCII-compatible format, and ASYST/ASYSTANT format.

### **Waveform Copy Feature**

With this feature, you can copy a selected channel or portion of a channel from a source file to a selected channel or portion of a channel in a destination file. You can transfer the calibration, screen scaling, and event markers of the copied channel, while leaving the sample rate of the destination file alone.

# GETTING READY

## TO USE ADVANCED CODAS

### INSTALLING ADVANCED CODAS ROUTINES

Before installing Advanced CODAS, verify the following:

- Your computer contains at least 640K of memory.
- You are using PC-DOS or MS-DOS version 3.0 or greater.
- You are using AT-CODAS or WINDAQ Waveform Browser playback and analysis software.

You will find the Advanced CODAS software on the same *The WinDag Resource* CD that you installed your WinDag software from. When you insert the CD, a welcome window will appear. Select “Install Software” and then select “Advanced CODAS.” You will need the password you acquired when you purchased Advanced CODAS to complete the installation.

One subdirectory containing both Advanced CODAS files and AT-CODAS or WINDAQ files allows the waveform playback programs of either to access the integration, differentiation, moving average, peak and valley capture, rectification, and report generation routines provided by Advanced CODAS. A summary of each Advanced CODAS file is as follows:

<b>ARITH . EXE</b>	ARITH.EXE is the arithmetic functions program. This program is automatically accessed by the menu-oriented program CALC.EXE, or may be run directly from DOS.
<b>AVERAGE . EXE</b>	AVERAGE.EXE is the waveform moving average program. This program is automatically accessed by the menu-oriented program CALC.EXE, or may be run directly from DOS.
<b>CALC . EXE</b>	CALC.EXE is a menu-oriented user interface that is accessed by the waveform playback program whenever CALC is invoked. CALC.EXE may also be accessed from the DOS level by typing “CALC”. When CALC is in use, it automatically accesses all waveform analysis programs as required.
<b>DERIV . EXE</b>	DERIV.EXE is the waveform differentiation program. This program is automatically accessed by the menu-oriented program CALC.EXE, or may be run directly from DOS.
<b>INTEGRAL . EXE</b>	INTEGRAL.EXE is the waveform integration program. This program is automatically accessed by the menu-oriented program CALC.EXE, or may be run directly from DOS.
<b>PEAK . EXE</b>	PEAK.EXE is the waveform peak and valley capture program. This program is automatically accessed by the menu-oriented program CALC.EXE, or may be run directly from DOS.

## ADVANCED CODAS

**RECTIFY.EXE** RECTIFY.EXE is the waveform rectification program. This program is automatically accessed by the menu-oriented program CALC.EXE, or may be run directly from DOS.

**REPORT.EXE** REPORT.EXE is the peak and valley report generator. This program is automatically accessed by the menu-oriented program CALC.EXE, or may be run directly from DOS.

### ACCESSING ADVANCED CODAS ROUTINES

CALC is a menu-driven, post-recording analysis program that allows the integration, differentiation, mathematical combination, moving average, peak and valley capture, rectification, and report generation of specified waveforms. CALC and its supporting waveform analysis routines may be accessed three ways:

1. The CALC menu may be accessed as an option from the COPY/PASTE menu (refer to the COPY AND PASTE FUNCTION in the *Post Acquisition using POST or WINDAQ* chapter of the AT-CODAS Users Manual or in the WINDAQ Users Manual).
2. For AT-CODAS users, the CALC menu may be accessed through the playback software's CALC HOT KEY function (ALT+F6 is the key sequence. Refer to the CALC (ADVANCED CODAS) FUNCTION in the *Post Acquisition using POST or WINDAQ* chapter of the AT-CODAS Users Manual for complete details).
3. Individual Advanced CODAS routines may be accessed directly from DOS.

### Invoking CALC from COPY AND PASTE

The COPY AND PASTE FUNCTION of playback software allows any range of waveform information to be extracted from a waveform data file. The extracted waveform information can then be converted into a data storage format as specified through the COPY/PASTE menu of AT-CODAS or through the Save As dialog box of WINDAQ Waveform Browser. When the CALC option is selected (using either method), the following events take place:

1. A new data file is created on disk carrying the file name you supplied when COPY/PASTE was first invoked (using AT-CODAS) or when the Save As dialog box was opened (using WINDAQ Waveform Browser).
2. The specified range of waveform information is extracted (copied) from the currently active waveform data file, then pasted into the newly created file using the CODAS (with data file header) format. ***Note that CALC will not produce accurate results if the data is pasted in a compressed format!***
3. CALC is run and initialized with the default file name of that containing the pasted waveforms. At this point, the CALC first-level menu will be displayed and you may execute your required CALC function(s) on the data file containing the pasted waveforms, or even change the default file name to operate on another file. When CALC is exited, control returns to the playback software in the state that existed just before COPY/PASTE was selected (using AT-CODAS) or just before Save As was selected (using WINDAQ Waveform Browser). The waveform data file from which waveforms were copied remains as the active waveform playback file. You may change this file to the one containing the pasted waveforms by invoking the SWITCH FILES FUNCTION. Refer to the SWITCH FILES FUNCTION in the AT-CODAS Users Manual or in the WINDAQ Waveform Browser Users Manual for complete details.

### Invoking CALC Using the Playback Software's CALC HOT KEY Function (AT-CODAS User's Only)

Unlike the COPY/PASTE method of invoking CALC (which operates on a *specified subset* of the currently-active waveform data file), the CALC HOT KEY function (ALT-F6) operates on the *entire* currently-active waveform data file. The following sequence of events describes the operation of the CALC HOT KEY function:

1. Invoke the CALC (ADVANCED CODAS) FUNCTION using ALT+F6 keystrokes. Refer to the CALC (ADVANCED CODAS) FUNCTION in the *Post Acquisition using POST or WINDAQ* chapter of the AT-CODAS Users Manual for complete details.



2. The CALC first-level menu is displayed. CALC's default file name is the same as that in effect prior to the time HOT KEY was enabled.
3. You may now operate on the file using CALC functions. When CALC is exited, control is returned to the waveform playback software with the active file being the one that was open when the HOT KEY was enabled. Assuming you did not change the file name from within CALC, the waveforms contained in the file should now reflect the operations performed by CALC.

### Invoking Advanced CODAS Routines From DOS

Assuming a CODAS-acquired waveform data file exists, each Advanced CODAS routine may be invoked directly from DOS using the following methods:

1. CALC  
Typing this syntax from the DOS command prompt invokes the first-level menu of CALC. No default conditions are established.
2. CALC *filename*  
Typing this syntax from the DOS command prompt invokes the first-level menu of CALC. "*filename*" is used as the default file name.
3. CALC *filename* A  
CALC *filename* D  
CALC *filename* G  
CALC *filename* I  
CALC *filename* M  
CALC *filename* P  
CALC *filename* R  
Typing any of these commands from the DOS prompt invokes CALC from its (A)rithmetic functions, (D)erivative, Report (G)enerator, (I)ntegral, (M)oving Average, (P)eak and valley capture, or (R)ectify menu respectively. In all three cases, "*filename*" is used as the default file name.
4. CALC ?  
Displays a help screen describing the syntax required to enter CALC at the menu level pertaining to any Advanced CODAS function.
5. FUNCTION *argument*  
Where *argument* forms a complete command line for the specified Advanced CODAS FUNCTION. It is possible to invoke any Advanced CODAS analysis routine with a complete command argument that allows you to bypass CALC's menus entirely. Such an approach is most useful for repetitive Advanced CODAS operations where the analysis procedure is always the same, or where you need to process a file, or a number of different files in a defined manner. This approach also requires less memory since the CALC menu is not loaded. A DOS batch file may be written that allows the flexibility of enabling many complex operations with a single command. Refer to your computer's DOS manual for further information related to batch file processing. Refer to the appropriate section of this chapter for a complete treatment of DOS command line syntax for waveform arithmetic functions, derivative, integration, moving average, peak and valley capture, rectification, and report generation Advanced CODAS operations.

# WORKING WITH CALC MENUS

## GENERAL

Assuming CALC is accessed from the DOS level or from waveform playback software, a user interface consisting of several menus that allow you to specify any Advanced CODAS parameter in a straight forward manner will be displayed. The menus used in CALC are designed to allow you to modify all menu parameters any number of times before executing the specific Advanced CODAS operation. The specific action configured by you at any menu level is activated only after pressing the escape key (ESC) on your keyboard.

## MOVEMENT THROUGH CALC MENU ITEMS

Generally, the cursor moves from the top of the menu item through the last (or bottom) menu item each time the ENTER key is pressed. When at the bottom menu item, the cursor moves back to the top menu item after pressing the ENTER key. If you attempt to enter an invalid argument for any menu item, the cursor will fail to move to the next menu item. Blank entries, those where only the ENTER key was pressed with no arguments, are ignored and force the cursor to the next menu item.

## ENTERING MENU ITEM ARGUMENTS

For each individual menu parameter, a pair of brackets is provided (<>) between which only valid arguments are displayed. In many cases, a menu item will be provided with a default parameter, such as a default input channel number, or default output channel number. These, of course, may be changed but they do provide a convenient starting point for analysis.

The initial cursor position is immediately following the valid entry brackets (<>). An argument is entered into a menu item by typing it at the cursor position, then pressing the ENTER key. If your argument is valid, the entry will appear within the brackets, your typed response following the bracket will be erased, and the cursor will move to the next menu item. If your entry is invalid, the cursor will simply reinitialize to its starting position, and fail to move to the next menu item. Your invalid response remains displayed until either of two inputs is provided:

1. Pressing the ENTER key alone will cause the cursor to move to the next menu item. The previous invalid response is erased, and the argument accepted for that menu item is either the default, or the last valid entry.
2. Upon detecting any keyboard entry other than ENTER, the invalid response will be erased and your new entry will appear at the cursor. Upon pressing ENTER, this new entry will be either accepted or rejected as described.

## ACTIVATING MENU SELECTIONS

Following your entry of specific menu items, action is taken at any menu level by pressing the ESC key on your keyboard. The last entry on any menu level determines whether the software will (C)ontinue or (E)xit. If this selection was "C", then the appropriate analysis action will be activated. Following completion of the selected analysis procedure, control will be returned to the activating menu with the cursor positioned at the top menu item. Note that at this point, the Continue/Exit selection has been changed to (E)xit thus allowing you to simply press ESC to exit and return to the first-level menu. Note also that upon changing the input file name, or entering a blank line (assuming you want to operate again on the same file) the state of the Continue/Exit selection changes to (C)ontinue.

Should (E)xit be enabled from the first-level menu at the time the ESC key is pressed, control will be returned either to the waveform playback program, or DOS depending upon which was active at the time CALC was enabled.

# WAVEFORM ARITHMETIC OPERATIONS USING ADVANCED CODAS

## GENERAL

The Waveform Arithmetic Operation (WAO) program of Advanced CODAS may be used to generate calculated channels as a function of user-supplied formulas. The program supports a host of operations including phase shifting and fully automatic calibration of the output waveform. Since all processing by the arithmetic operations program is performed using floating point math, use of a math coprocessor will enhance the speed with which these operations are carried out.

## WAO OPERATORS AND BUILT-IN FUNCTIONS

WAO supports 8 operators and 7 functions. Each of the 8 operators is described as follows according to its syntax, precedence in relation to other operators, and full description.

<u>Operator Syntax</u>	<u>Precedence</u>	<u>Description</u>
+	1	Arithmetic addition
-	1	Arithmetic subtraction
*	2	Arithmetic multiplication
/	2	Arithmetic division
Negation (-)	3	Arithmetic negation (e.g., -2, -12, -123)
**	4	Exponentiation (e.g., 2.7 <sup>2</sup> , 2 <sup>2.7</sup> )
( )	5	Parentheses to change the order that arithmetic operations are performed
#	—	Number following the “#” character specifies data of an absolute channel number

The WAO program also supports 7 functions as follows:

<u>Function Syntax</u>	<u>Description</u>
ABS(x)	Absolute value of argument (x)
SQRT(x)	Square root of argument (x)
EXP(x)	e raised to the power of argument (x) *
LOG(x)	Natural logarithm of argument (x) *
LOG10(x)	Common logarithm of argument (x)
#Y:x	Phase shift waveform channel “Y” “x” samples (1 to 20 samples maximum)
:x	Phase shift output channel “x” samples

\* e = 2.71828...

## COMBINING FUNCTIONS AND OPERATORS INTO WAO PROGRAM FORMULAS

WAO formulas are generated much like any other computer language with two exceptions. Unique to the WAO program are two symbols that perform specific operations, they are the pound symbol (#) and the colon (:).

Data for a channel is denoted by “#” followed by the channel number. This approach allows channel numbers to be differentiated from constants. The channel number may be followed by an optional colon and positive delay (number of sample intervals ranging from 1 to 20) to allow phase shifting of the channel data.

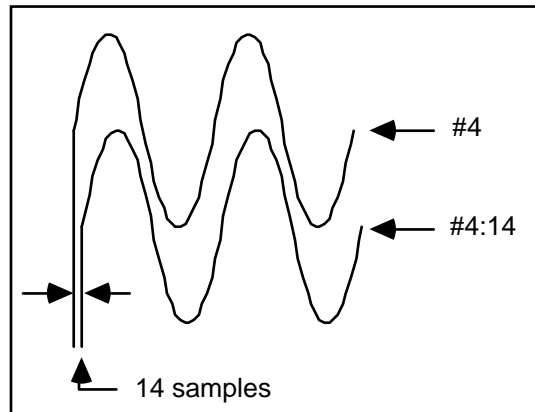
Proper formula generation can best be described by way of example:

## ADVANCED CODAS

<u>Desired Operation</u>	<u>Resulting WAO Formula*</u>
Channel 1 plus Channel 2	#1+#2
The quantity Channel 1 plus Channel 2, divided by the quantity Channel 1 times Channel 8	(#1+#2) / (#1*#8)
Common log of Channel 14	log10 (#14)
The quantity Channel #1 plus 573.2 divided by .02881	(#1+573.2) / .02881
The quantity of the absolute value of channel 3 plus the natural log of channel 15, divided by the quantity of Channel 3 raised to the power of Channel 7	(abs (#3)+log (#15)) / (#3**#7)
Channel 4 right phase shifted 14 samples†	#4 : 14†

\*Note that WAO functions are not case sensitive (e.g., LOG ≡ log) and that spaces must not appear in the formula.

†Result of a waveform phase shifted 14 samples from the original:



### ENGINEERING NOTATION WITH THE WAO PROGRAM

As with other Advanced CODAS utilities, the WAO program supports automatic engineering notation of its output (or calculated) channels. However since it is not possible to predict how the output relates to time due to the diverse range of possible calculations, the output units name is a concatenation of up to 4 unique input channel numbers in the order of first appearance in the WAO formula. Channel numbers above 9 are represented alphabetically, with A for channel 10 through T for channel 29. If you are using an EXP channel multiplexer, you have access to as many as 144 channels. In this case, the standard ASCII character set is used to represent channels beyond channel 35. For example, the result of the operation (#1+#4+#9+#12) would carry engineering units annotation of "149C". You can change this annotation whenever desired from playback software by enabling the window containing the waveform, pressing F9, then entering a space followed by up to 4 characters of annotation (for AT-CODAS) or by enabling the window containing the waveform, choosing Low Calibration... from the Edit menu, then entering up to four characters of annotation (for WINDAQ Waveform Browser).

### FORMULA AND CALCULATED ERROR DETECTION

Error detection and reporting is supported by the WAO program both from the CALC menu and DOS command line levels. However, error handling is different in each case:

MENU-LEVEL ERROR DETECTION

When operating the WAO program from the menu level, two lines are reserved within the menu area for error reporting. One at the bottom of the screen, and one directly underneath the user formula line. The type of message appearing in each area depends upon whether the error was calculated, or the result of an improperly formatted formula.

MENU-LEVEL FORMULA ERROR DETECTION

Whenever an improperly formatted expression is entered at the formula prompt of the menu, the message

**Fix expression (ESC clears it)**

appears at the bottom of the menu screen immediately after attempting to evaluate the operation. This message indicates that the formula provided does not adhere to certain rules governing the syntax of WAO program formula expressions. At the same time the above message is displayed, a companion message located directly beneath the erroneous formula provides more detail about the nature of the syntax error. One of the following 13 possible error messages may be displayed:

<u>Error Message</u>	<u>Erroneous Example</u> <sup>(5)</sup>	<u>Corrected Example</u>
illegal digit	# <u>A</u> +1	#1+1
illegal channel number <sup>(1)</sup>	# <u>3</u> 0+1	#2+1
illegal delay value <sup>(2)</sup>	#1: <u>2</u> 5	#1:20
illegal function	<u>l</u> n(#1)	log(#1)
illegal operator	#1 <u>%</u> #2	#1*#2
missing operator	(#1+1) <u>_</u> (#2+2)	(#1+1)*(#2+2)
missing operand	#1+ <u>_</u>	#1+#2
missing ")"	(#1+1) <u>_</u>	#1+1 or (#1+1)
extra ")"	(#1+1) <u>_)</u>	#1+1 or (#1+1)
missing "("	sqrt <u>#</u> 1)	sqrt(#1)
missing digit	#1+ <u>#_</u>	#1+#2
too many constants <sup>(3)</sup>	--	--
illegal constant <sup>(4)</sup>	--	--

- (1) Specified channel number must not exceed the total number in the file.
- (2) Phase shift argument must not exceed 20.
- (3) Total number of constants appearing in the formula must not exceed 21.
- (4) Constants specified in the formula must fall within the range of  $-1.79E^{-308}$  to  $1.79E^{308}$ .
- (5) Each underlined character indicates where WAO error detection code places the cursor to indicate the location of the problem.

Immediately upon detecting an error in the formula, WAO software enters an editor mode that allows a convenient method for correcting the formula. The → and ← cursor control keys may be used to move the cursor across the characters of the formula. HOME and END keys move to the beginning and end of the formula line, the INS key may be used to toggle between the insert and overwrite typing modes, pressing the DEL key deletes the character at the cursor, and pressing ESC clears the entire formula line.

MENU-LEVEL CALCULATED ERROR DETECTION

If while generating the calculated channel an arithmetic error occurs, the software will display

**Arithmetic error at TBF = X**

at the bottom of the menu screen. Quantity "X" indicates the elapsed time in seconds from the beginning of the file (TBF stands for Time from Beginning of File) where the error occurred. When you return to the playback program, the cursor will automatically be positioned at the exact point where the error exists. At the same time the above message is displayed, a companion message located directly beneath the formula provides more detail about the nature of the arithmetic error. In addition, the first character of the formula operator that was involved with the error is displayed with intensified video. One of the following messages will be displayed in the formula area:

## ADVANCED CODAS

<u>Error Message</u>	<u>Expanded Description</u>
fp divide by 0	The quantity 0 appeared in the denominator
fp overflow	The operation exceeded the range of double precision floating point numbers
fp func domain error	An illegal arithmetic operation was attempted, such as the log of a negative number
fp func singular error	An operation occurred that forced the result to $\pm$ infinity, such as $\log(0)$

### COMMAND-LEVEL FORMULA ERROR DETECTION

When the WAO program is invoked from the DOS command line, formula error detection is still maintained. The same formula error messages described above for menu-level operations will be echoed to the screen when an invalid formula is attempted. In addition, the formula will be echoed starting with the offending character. For example, the following messages will be echoed to the screen upon attempting the operation **ARITH DATA.DAT 4 1+LN(#2)+12.5**:

```
LN(#2)+12.5
illegal function
```

### COMMAND-LEVEL CALCULATED ERROR DETECTION

If an arithmetic error occurs while generating the calculated channel, the screen will echo the formula starting with the operator or operand involved with the error along with an error message describing the nature of the error. For example, should you attempt **ARITH DATA.DAT 5 #1/#2** and channel 2 at one point equals zero, the following messages will be echoed to the screen:

```
/#2
fp divide by 0 at TBF = x
```

As with menu-level operations, the value equated to TBF defines the location of the offending operation relative to the beginning of the data file in seconds. When you run the playback program, the cursor will automatically be positioned at the exact point where the error exists.

## WAO AND DYNAMIC RANGE ERRORS

The WAO program performs all calculations using double precision floating point operations. At the end of a calculation operation, the floating point numbers are converted to 14-bit signed integers, written to the CODAS or WINDAQ file, and the channel's m and b scaling constants are adjusted accordingly. During the conversion process to 14-bit precision, the highest value floating point numbers are scaled to display at the top of the window, and the lowest at the bottom. The presence of transient information in the resulting waveform may damage lower value accuracy since a one count change in the 14-bit number represents a comparatively huge step change in the floating point output.

## WAO DOS COMMAND LINE and CALC MENU SYNTAX

**ARITH** *argument*

*argument:*

[d:] [path] filename [.ext] Y *expression*

where: **[d:] [path] filename [.ext]** specifies the file name containing the waveforms to be manipulated. The disk drive specifier, path, and file name extension are optional. A basic file name must be supplied.

### CALC Menu Prompt

Output channel number

### DOS Command Line

**Y** is the channel number within the file to contain the calculated waveform where  $2 \leq Y \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $2 \leq Y \leq 144$  for WINDAQ EXP multiplexer users. If **Y**  $\leq$  the total number of channels contained in the file before the operation, then the result is placed in waveform channel **Y** and the original contents of channel **Y** are *overwritten and destroyed*. If **Y**  $>$  the total number of channels contained in the file before the operation, then the result is appended to the input file with the channel number equal to the original total number of channels + 1.

Enter expression

This is the formula containing input channel numbers, constants, and functions combined mathematically and according to the rules established earlier in this section to generate the desired output channel Y.

Examples:

```
ARITH DATA.DAT 4 #1+#2-#3
```

Using DATA.DAT as the input waveform data file, subtract channel 3 from the sum of channels 1 and 2 and store the result as waveform channel 4.

```
ARITH C:\RUN1\TEST.DAT 15 #1*LOG(#2-#3)
```

Using TEST.DAT located on drive C:, subdirectory RUN1 as the input waveform data file, multiply channel 1 times the natural log of the difference of channel 3 from channel 2. Store the result as waveform channel 15.

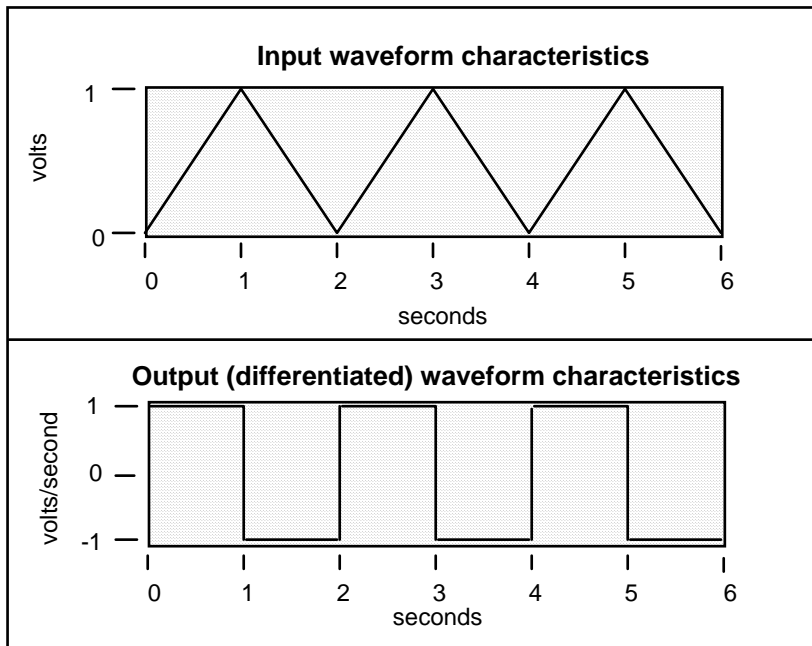
# WAVEFORM DERIVATIVES USING ADVANCED CODAS

## GENERAL

The waveform derivative program provided with Advanced CODAS allows any waveform within a CODAS- or WINDAQ-acquired data file to be differentiated with respect to time. The differentiated waveform is placed within the same file by either inserting the waveform as an additional channel (assuming less than 29 original channels), or by overwriting an existing channel. If the latter approach is selected, care should be taken to ensure that the original content of the overwritten channel is not pertinent since it will be **destroyed**.

## AUTOMATIC ENGINEERING UNIT CALIBRATION

When a waveform is differentiated by Advanced CODAS, the calibration information associated with the data is automatically adjusted so that when the differentiated waveform is reviewed through the waveform playback program a display of its value at any given point is in appropriate engineering units. The engineering unit tag associated with the calculated waveform (and displayed on the waveform playback program screen) will always be "#/s"; where "#" is the channel number of the waveform used as an input to create the derivative, and "/s" denotes 'per second'. In this manner, the differentiated channel's unit of measure is always the unit of measure of the input waveform divided by seconds. Note that the engineering units tag of any enabled channel may be changed at any time from the waveform playback program by pressing either the F9 or F10 key, then entering up to a 4-character (the first character must be an alpha) descriptor (for AT-CODAS users) or by choosing either Low Calibration... or High Calibration... from the Edit menu, then entering up to a four character descriptor in the Engr. Units = text box (for WINDAQ Waveform Browser users).



**FIGURE 6A.1**

The derivative program of Advanced CODAS generates the first derivative of an input waveform. Just as this ideal example demonstrates, the differential waveform generated by Advanced CODAS is scaled to the appropriate unit of measure to allow a complete evaluation of the result. Assuming that the input waveform was CODAS or WINDAQ channel 4, the engineering unit tag on the output waveform would be displayed as '4/s' on the waveform playback program screen to indicate that the unit of measure is channel four's unit of measure divided by seconds (in this case: volts per second).



**SELECTING A SMOOTHING FACTOR**

A feature of Advanced CODAS's waveform derivative program is built-in smoothing of the differentiated waveform. As you may be aware, noise present on an input waveform will be greatly magnified when the waveform is passed through a differentiating function. This is true of not only digital differentiators such as Advanced CODAS, but of analog differentiators as well. A feature common to the later to help compensate for noise in the output signal is a low pass filter with selectable corner frequencies. The smoothing factor parameter of Advanced CODAS's waveform differentiator accomplishes the same noise-quieting result. The value of the smoothing factor determines the number of samples evaluated from the input waveform for determining a slope. It follows that the greater the smoothing factor, the more heavily filtered the output signal becomes. A smoothing factor of 2 (no filtering) yields an output signal that is much more sensitive to rates of change in the input waveform than a smoothing factor of 11. Clearly, a balance must be achieved when selecting a smoothing factor so that the resulting waveform is not cluttered with so much noise as to make evaluation difficult, while ensuring that true rates of change in the input waveform are not masked by a smoothing factor that is too great. This digital filtering technique comes without the disadvantages of analog filtering which causes substantial phase shift of the output vs. input signals.

Generally, a smoothing factor between 3 and 7 yields the best performance, keeping in mind that a smoothing factor of 2 should be used under only the most ideal circumstances. Advanced CODAS supports smoothing factors ranging from 2 to 64.

**WAVEFORM DERIVATIVE DOS COMMAND LINE and CALC MENU SYNTAX**

DERIV *argument*

*argument:*

[d:] [path] filename [.ext] D X Y Z

where: **[d:]**[path]**filename**[.ext] specifies the file name containing the waveform to be differentiated. The disk drive specifier, path, and file name extension are optional. A basic file name must be supplied.

CALC Menu Prompt

Input channel number

DOS Command Line

**X** is the channel number of the waveform to be differentiated where  $1 \leq X \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $1 \leq X \leq 144$  for WINDAQ EXP multiplexer users.

Output channel number

**Y** is the channel number within the file to contain the differentiated waveform where  $2 \leq Y \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $2 \leq Y \leq 144$  for WINDAQ EXP multiplexer users. If **Y** ≤ the total number of channels contained in the file before the derivative operation, then the result is placed in waveform channel **Y** and the original contents of channel **Y** are *overwritten and destroyed*. If **Y** > the total number of channels contained in the file before the derivative operation, then the result is appended to the input file with the channel number equal to the original total number of channels + 1.

Smoothing factor

**Z** specifies the smoothing factor to be applied to the differentiated waveform where  $2 \leq Z \leq 64$ . Higher values of **Z** apply greater smoothing constraints on the resulting waveform. A factor of 2 chosen for **Z** produces no smoothing of the resultant waveform.

Example:

DERIV DATA.DAT 1 4 3

Using DATA.DAT as the input waveform data file, take the derivative of waveform channel 1 and store the result as waveform channel 4 using a smoothing factor of 3.

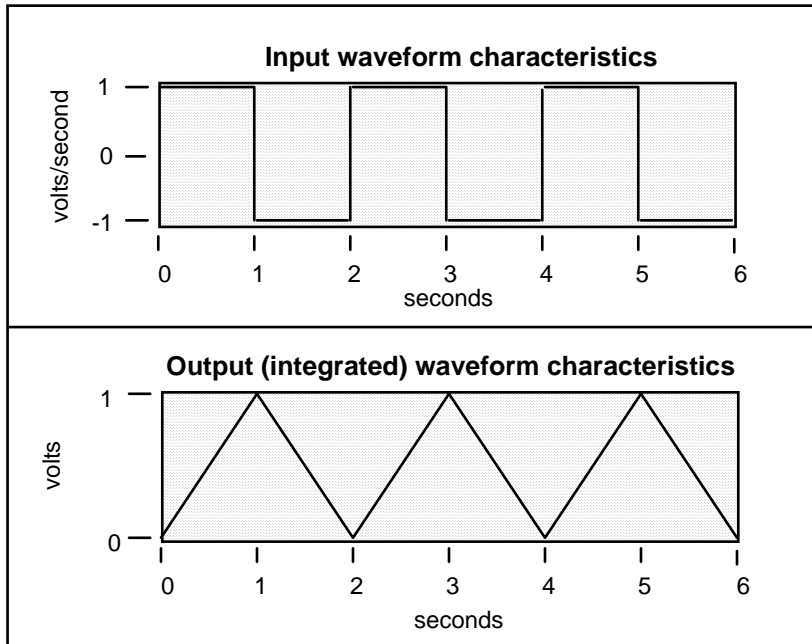
# WAVEFORM INTEGRATION USING ADVANCED CODAS

## GENERAL

The waveform integration program provided with Advanced CODAS allows any waveform within a CODAS-acquired data file to be integrated with respect to time. The integrated waveform is placed within the same file by either inserting the waveform as an additional channel (assuming less than 29 original channels), or by overwriting an existing channel. If the latter approach is selected, care should be taken to ensure that the original content of the overwritten channel is not pertinent since **it will be destroyed**.

## AUTOMATIC ENGINEERING UNIT CALIBRATION

When a waveform is integrated by Advanced CODAS, the calibration information associated with the data is automatically adjusted so that when the integrated waveform is reviewed through the waveform playback program a display of its value at any given point is in appropriate engineering units. The engineering unit tag associated with the calculated waveform (and displayed on the waveform playback program screen) will always be "#\*s"; where "#" is the channel number of the waveform used as an input to create the integral, and "\*s" denotes 'times seconds'. In this manner, the integrated channel's unit of measure is always the unit of measure of the input waveform multiplied by seconds. Note that the engineering units tag of any enabled channel may be changed at any time from the waveform playback program by pressing either the F9 or F10 key, then entering up to a 4-character (the first character must be an alpha) descriptor (for AT-CODAS users) or by choosing either Low Calibration... or High Calibration... from the Edit menu, then entering up to a four character descriptor in the Engr. Units = text box (for WINDAQ Waveform Browser users).



**FIGURE 6A.2**

The integration program of Advanced CODAS generates the integral of an input waveform. Just as the above ideal example illustrates, the integrated waveform generated by Advanced CODAS is scaled to the appropriate unit of measure to allow a complete evaluation of the result. Assuming that the input waveform was CODAS channel 4, the engineering unit tag on the output waveform would be displayed as '4\*s' on the waveform playback program screen to indicate that the unit of measure is channel four's unit of measure multiplied by seconds (in this case: volts).

**EXTENDED FEATURES OF ADVANCED CODAS's INTEGRATOR**

Besides integration, Advanced CODAS's integrator program supports many features not available on any other commercially available software product:

**1. SUM and PULSE Modes of Integration.**

The sum mode of integration reflects that which is commonly available in software integration products. The input waveform is continually integrated without providing the ability to reset the integral.

The pulse mode of signal integration forces the integrator to reset whenever the input signal crosses or equals zero. It is important to note that *zero* is defined as *zero reference* which does not need to be (and in most cases is not) equal to zero volts. Rather, zero is defined by the value of the input waveform that was *calibrated* through the waveform playback program to be equal to zero regardless of the actual voltage applied to the A-D converter. Remember that through use of the waveform playback program's ENGINEERING UNITS FUNCTION (for AT-CODAS users) or WAVEFORM CALIBRATION FUNCTION (for WINDAQ Waveform Browser users), you have the ability to set zero reference at any place on an input signal's waveform. This ability combined with signal integration relative to *zero reference* allows the integrator to reset as a function of actual signal value rather than as a function of signal voltage.

**2. Signal RECTIFICATION Before Integration.**

Another unique feature of Advanced CODAS's integrator is its ability to allow you to define whether integration will be applied to positive and negative, positive only, negative only, or the absolute value of input signal transitions relative to *zero reference*. Refer to FIGURE 6A.4 for a graphic demonstration of how waveform rectification can assist in waveform analysis of integrated waveforms.

**3. An Optional HOLD Function.**

For every integration mode of operation, a hold feature can be enabled that forces CALC to hold the peak integral value that was calculated before a reset. In applications where only the peak integral value is of interest, use of the hold feature yields an uncluttered, histogram-like waveform. For applications that require a determination of the integral value at discrete steps before reset, the hold option may be disabled. Refer to FIGURE 6A.4 for a graphic example of the integrator's output with the hold feature enabled.

**4. Adjustable RESET INTEGRATION PERIOD.**

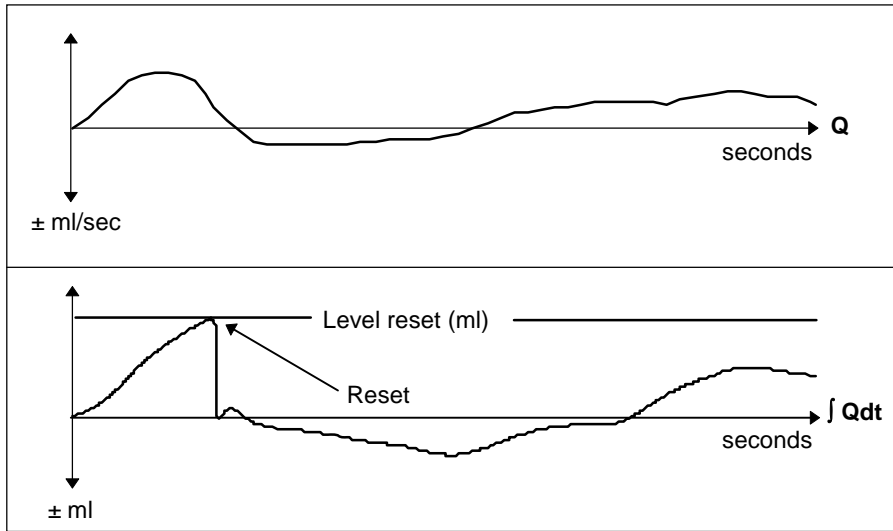
Advanced CODAS's integration program does not force the integral to reset only on zero crossings. By supplying a sample number period between 2 and 32,767 samples, the integral will reset upon reaching the specified sample count. For example, given a waveform sampled at 1,000 samples per second, specifying an integration period of 1000 will force the integral to reset every one second ( $1/1,000 \text{ seconds/sample} \cdot 1,000 \text{ samples} = 1 \text{ second}$ ). A running integral (one that never resets) of a waveform may be generated by specifying an integration period of 1.

**5. Selectable LEVEL RESET.**

Another method of achieving integration reset is by specifying a specific level in the output waveform's engineering unit of measure. For example, if the waveform used as the input to the integrator is calibrated in quarts per second, specifying a level reset of 100 quarts will force reset when this quantity is reached in the integrated waveform. Levels within the range of  $-1.79 \times 10^{308}$  to  $1.79 \times 10^{308}$  units may be specified. Refer to FIGURE 6A.3 for a graphic example of the level reset function. When you need to integrate an entire waveform stream and reset only when the specified level is achieved, you must set the reset integration period to 1. This forces integration software to reset only upon a valid level threshold.

**6. Resetting on an EXTERNAL WAVEFORM EVENT**

The final method for resetting the integral is upon an event occurring on a different waveform channel. An event is defined as a peak marker, valley marker, or either occurring on another channel within the same file. A typical application for the external reset function is resetting integration as a function of an ECG waveform. Using this approach, you would first peak capture the ECG waveform using Advanced CODAS's Peak and Valley Capture program, then specify the external reset trigger option from either the CALC integral menu or DOS command line. Each time a peak and/or valley marker on the ECG waveform is encountered by integration software, the waveform being integrated will reset.



**FIGURE 6A.3**

Demonstrating the use of the level reset function, the integrated waveform is set up to reset at a specific value. When the value is reached, integration is reset.

**WAVEFORM INTEGRAL DOS COMMAND LINE and CALC MENU SYNTAX**

INTEGRAL *argument*

*argument:*

[d:] [path] filename [.ext] X Y A Z [L] [T\_M] [H]

where:

**[d:][path]filename[.ext]** specifies the file name containing the waveform to be integrated. The disk drive specifier, path, and file name extension are optional. A basic file name must be supplied.

CALC Menu Prompt

Channel number to integrate

DOS Command Line

**X** is the channel number of the waveform to be integrated where  $1 \leq X \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $1 \leq X \leq 144$  for WINDAQ EXP multiplexer users.

Output channel number

**Y** is the channel number within the file to contain the integrated waveform where  $2 \leq Y \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $2 \leq Y \leq 144$  for WINDAQ EXP multiplexer users. If  $Y \leq$  the total number of channels contained in the file before the integral operation, then the result is placed in waveform channel **Y** and the original contents of channel **Y** are *overwritten and destroyed*. If  $Y >$  the total number of channels contained in the file before the CALC operation, then the result is appended to the input file with the channel number equal to the original total number of channels +1.

Integration mode

**A** specifies the type of integration to be performed as follows:

If **A**=“**B**” bipolar, then channel **X** will be integrated over both positive and negative transitions relative to zero reference.

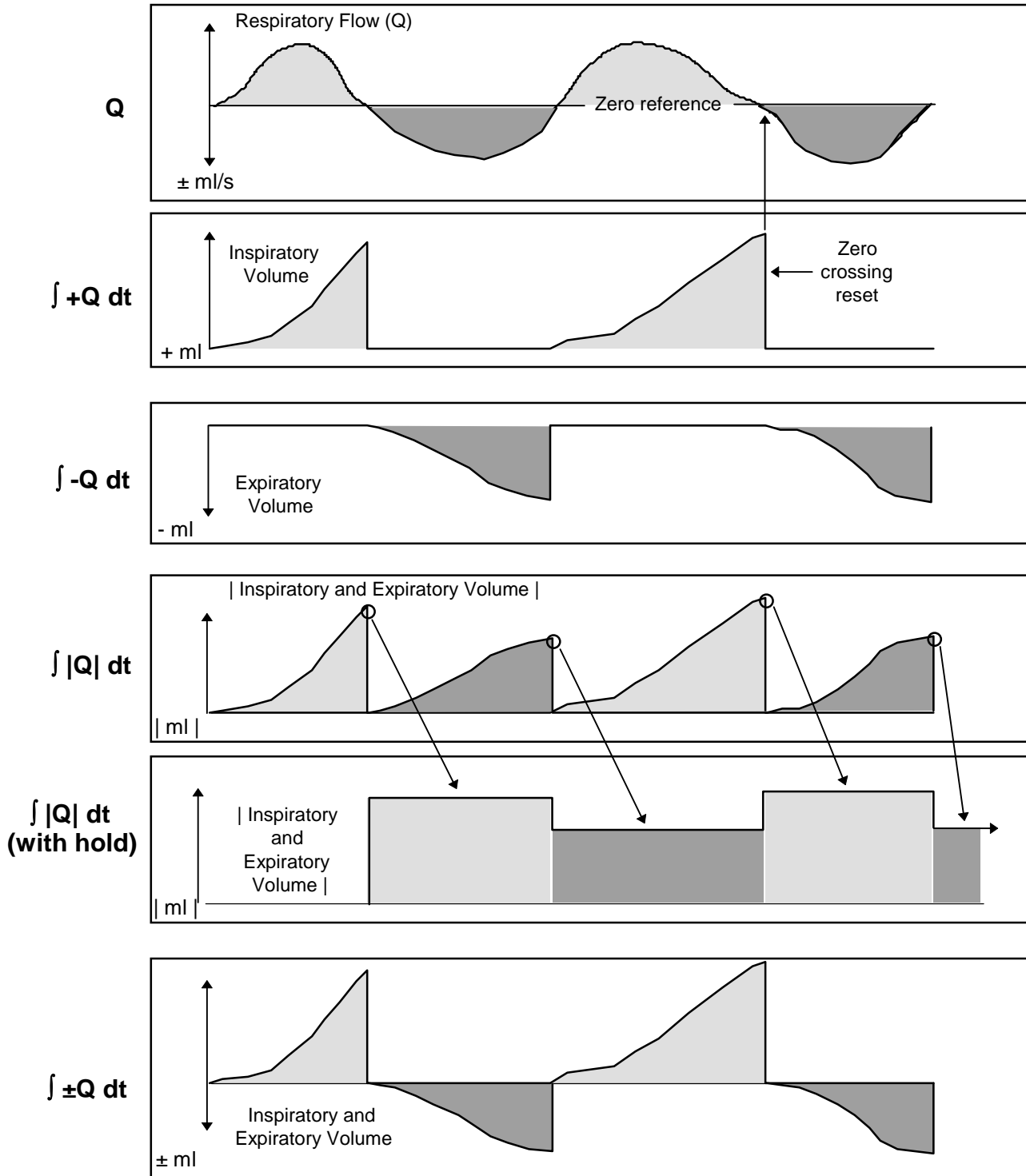
If **A**=“+”, then only the positive-going transitions of channel **X** relative to zero reference will be displayed and integrated.

If **A**=“-”, then only the negative-going transitions of channel **X** relative to zero reference will be displayed and integrated.

- If **A**="A"bsolute, then the integral will be applied to the absolute value of channel **X** relative to zero reference.
- Integral reset period**      **Z** specifies the reset period of the integral where  $0 \leq Z \leq 32767$ . If **Z** = 0, then the integral will reset upon every zero crossing of the input waveform. If **Z** = 1, then the specified waveform is integrated from beginning to end without recognizing zero-crossing as a valid reset condition. Reset may still occur upon a specified level or external event if these features are enabled. If  $2 \leq Z \leq 32767$ , then the integral will reset after every **Z**th sample to a maximum of 32,767 samples.
- Level reset units**      "L"evel reset, when included, specifies the maximum number of units relative to the output waveform that will be accumulated before a reset will occur. For example, given an input waveform calibrated in milliliters per second, a value of 100 ml could be entered for level which would cause the integrator to reset every 100 ml. Values supplied for level reset may range from  $-1.79 \times 10^{308}$  to  $1.79 \times 10^{308}$  units.
- External reset**      An external "T"rigger may be enabled to force reset of the integral as a function of a peak and/or valley event occurring on another channel within the same file. The variable "T" defines the channel number containing the reset trigger event where  $2 \leq T \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $2 \leq T \leq 144$  for WINDAQ EXP multiplexer users, and variable "M" may assume one of three states: "+", "-", or "B". "M" specifies whether reset will occur on a positive-going (peak) event, a negative-going (valley) event, or both. Note that variables **T** and **M** must be separated by a space when implementing this feature from a DOS command line.
- Hold integrated waveform?**      "H"old, when included, specifies that the last integrated value will be held until the next is calculated. When **H** is not specified, reset transitions will be displayed.
- Example:**
- ```
INTEGRAL D:\DATA\TEST.1 3 2 A 0
```
- Using TEST.1 located in subdirectory DATA on drive D: as the input waveform data file, take the integral of channel 3 using the absolute value (full-wave) rectification mode and using zero crossings to reset the integrator. The result is stored in waveform channel 2.
- ```
INTEGRAL C:\DATA\LABWORK\RUN2.DAT 4 8 B 32000
```
- Using file RUN2.DAT located in subdirectory LABWORK which is a subdirectory of DATA, take the integral of waveform channel 4 with no rectification (bipolar) and store the result in channel 8. This is a running integral resetting after 32000 data points
- ```
INTEGRAL D:\DATA\TEST.8 7 3 + 20 H
```
- Using waveform file TEST.8 located in subdirectory DATA on drive D:, integrate channel 7 using the positive-only rectification mode and resetting every 20 samples. The output waveform is channel 3 and each 20-point integration value will be held until the next is available. Refer to FIGURE 6A.3.
- ```
INTEGRAL DATA.DAT 2 3 - 32000 134
```
- Using Waveform file DATA.DAT, integrate waveform channel 2 storing the result as waveform channel 3. Rectify the input waveform so that only negative transitions relative to zero reset are integrated. Reset every 32,000 samples, or upon 134 integrated units, which ever occurs first.
- ```
INTEGRAL DATA.DAT 2 3 - 32000 134 14 +
```

**ADVANCED CODAS**

Using Waveform file DATA.DAT, integrate waveform channel 2 storing the result as waveform channel 3. Rectify the input waveform so that only negative transitions relative to zero reset are integrated,. Reset every 32,000 samples, upon 134 integrated units, or upon a positive-going external event on channel 14 which ever occurs first.



**FIGURE 6A.4**

Waveform rectification and reset upon zero crossing of the input signal allows great flexibility in the presentation of integrated waveforms. Note how the hold option yields an uncluttered trace when only the peak value of the integral on a cycle-by-cycle basis is required.

# WAVEFORM MOVING AVERAGE USING ADVANCED CODAS

## GENERAL

In cases where a given waveform is cluttered with noise, where a mean needs to be extracted from a periodic signal, or where a slowly drifting baseline needs to be removed from a higher frequency signal, Advanced CODAS's moving average filter may be applied to achieve the desired result. Its wide span of smoothing factors (2 - 10,000) allow a great deal of flexibility in waveform moving average applications. Typically, a smaller smoothing factor is applied when in noise reduction applications while larger factors are applied to generate the mean value of an input waveform or to remove the affects of a slowly drifting baseline. The moving average filter may also be used in conjunction with (but before) Advanced CODAS's waveform derivative program, thus prefiltering the signal before differentiating. Such an approach may be used when it is necessary to isolate the derivative function from its built-in smoothing capability. In these cases, a smoothing factor of 2 (no filtering) should be chosen from Advanced CODAS's derivative program.

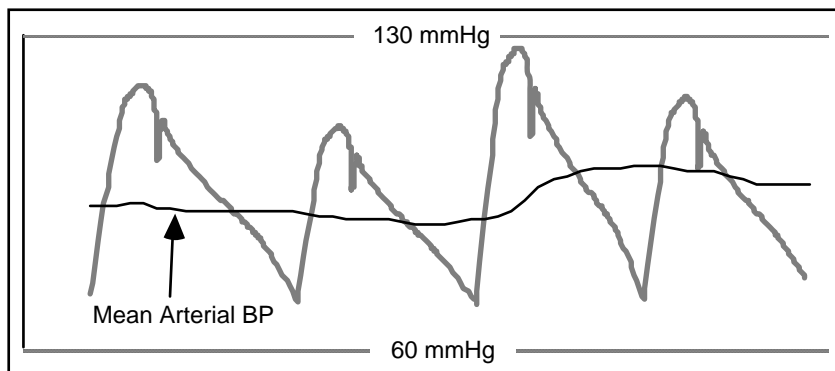
## AUTOMATIC ENGINEERING UNIT CALIBRATION

When a waveform is averaged by Advanced CODAS, the calibration information associated with the data is automatically adjusted so that when the moving averaged waveform is reviewed through the waveform playback program a display of its value at any given point is in appropriate engineering units. The engineering unit tag associated with the moving averaged waveform (and displayed on the waveform playback program screen) will always be "MA#"; where "#" is the channel number of the waveform used as an input to create the moving averaged waveform, and "MA" implies (M)oving (A)verage. The only exception to this rule is when a moving average factor of 1 is specified. In this case, the input channel is copied without modification to the output file including engineering units. Note that the engineering units tag of any enabled channel may be changed at any time from the waveform playback program by pressing either the F9 or F10 key, then entering up to a 4-character (the first character must be an alpha) descriptor (for AT-CODAS users) or by choosing either Low Calibration... or High Calibration... from the Edit menu, then entering up to a four character descriptor in the Engr. Units = text box (for WINDAQ Waveform Browser users).

## APPLYING Advanced CODAS's MOVING AVERAGE PROGRAM

The application of Advanced CODAS's moving average program, as with all other moving average functions, is an exercise in trial and error and intuition. When it is necessary to remove a noise component from an input waveform, only you can judge the suitability of the result; whether the number of averaged points selected was too high, too low, or just right. Such subjective criterion exists since what may appear to one user to be a properly filtered waveform, may appear to another to be too noisy. As a general rule, you should start with a low filter factor, then increase it if necessary until the desired result is achieved. Also, realize that more than one iteration may be needed to converge on an acceptable result.



**FIGURE 6A.5**

This figure demonstrates the ability of Advanced CODAS's moving average filter to generate a mean waveform from a periodic input waveform.

When using Advanced CODAS's moving average program to display the mean value of an input signal, the smoothing factor should be selected so that the number of points contained in the moving average window is proportional to the number of sampled points contained in one typical signal cycle. A good first order approximation for the frequency response of the mean signal is as follows:

$$\text{Frequency}_{\text{mean}} = (1.6 \times \text{Sample Rate}) \div n$$

Where:  $n$  = The number of points contained in the moving average filter.

Sample Rate = The per channel sample rate at which the input signal was originally acquired in Hertz.

Frequency<sub>mean</sub> = The bandwidth of the mean signal in Hertz.

This formula may be algebraically rearranged as follows to solve for  $n$ :

$$n = (1.6 \times \text{Sample Rate}) \div \text{Frequency}_{\text{mean}}$$

The use of this formula can best be described by example: Assume that an arterial blood pressure waveform was sampled at 250Hz, and that a mean ABP signal with a frequency response of 1Hz needs to be generated. Plugging these numbers into the above equation solves for the moving average span ( $n$ ).

$$n = (1.6 \times 250) \div 1 = 400 \text{ samples}$$

Using 400 as the moving average smoothing constant is a good first order approximation for the desired result.

The waveform moving average utility may also be used to simulate a high pass filter by specifying smoothing factors less than zero. When a moving average factor less than zero is specified, the resulting output is the input signal minus the moving averaged result.

## ADVANCED CODAS

### WAVEFORM MOVING AVERAGE DOS COMMAND LINE and CALC MENU SYNTAX

AVERAGE *argument*

*argument:*

[d:] [path] filename [.ext] X Y Z

where: **[d:][path]filename.ext** specifies the file name containing the waveform to be moving averaged. The disk drive specifier, path, and file name extension are optional. A basic file name must be supplied.

#### CALC Menu Prompt

Input channel number

#### DOS Command Line

**X** is the channel number of the waveform to be moving averaged where  $1 \leq X \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users, or  $1 \leq X \leq 144$  for WINDAQ EXP multiplexer users.

Output channel number

**Y** is the channel number within the file to contain the moving averaged result where  $2 \leq Y \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users, or  $2 \leq Y \leq 144$  for WINDAQ EXP multiplexer users. If **Y**  $\leq$  the total number of channels contained in the file before the moving average operation, then the result is placed in waveform channel **Y** and the original contents of channel **Y** are *overwritten and destroyed*. If **Y**  $>$  the total number of channels contained in the file before the CALC operation, then the result is appended to the input file with the channel number equal to the original total number of channels +1.

Smoothing factor

**Z** specifies the moving average span to be applied to the input waveform where  $2 \leq Z \leq 10,000$  samples for low pass filter simulation, or  $-9,999 \leq Z \leq -2$  samples for high pass filter simulation. Higher values of **|Z|** apply greater smoothing constraints on the resulting waveform. If **Z** = 1 then no smoothing is applied and the specified input channel may be copied without modification to the specified output channel. This feature may be used to peak capture waveform channel 1, which may not be done on channel 1 directly.

Example:

AVERAGE DATA.DAT 2 5 350

Using DATA.DAT as the input waveform data file, take the moving average of waveform channel 2 and store the result as waveform channel 5 using a 350 point moving average.

# WAVEFORM PEAK AND VALLEY CAPTURE USING ADVANCED CODAS

## **GENERAL**

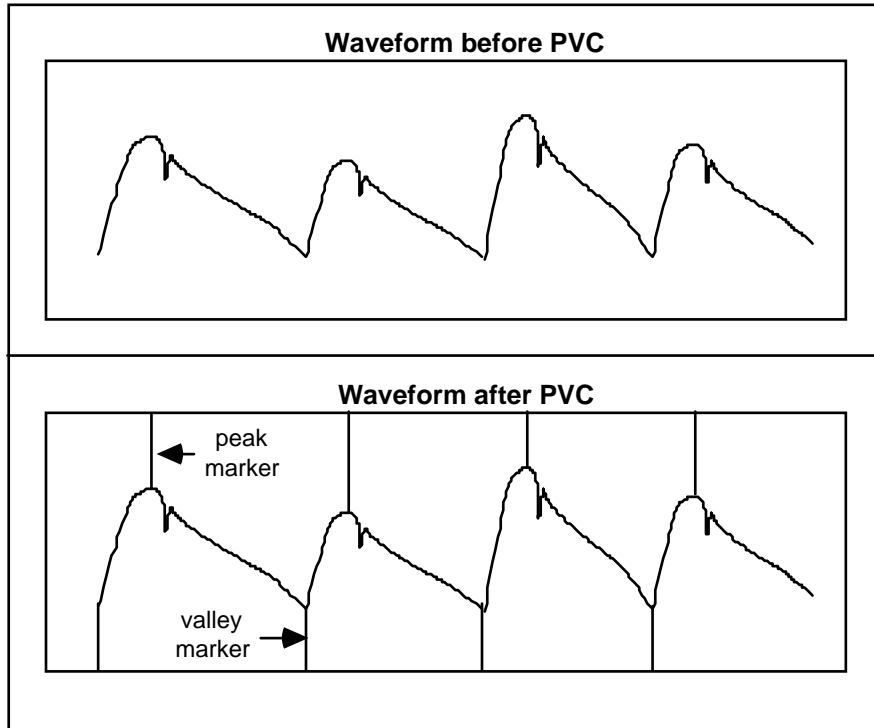
Anyone with the most basic knowledge of physiological waveforms can glance at a blood pressure waveform, determine where the heart cycle begins and ends, then measure the minimum (diastolic), maximum (systolic), and mean pressures. Another simple calculation yields the heart rate. The problem is not one of procedure, but time. Consider a more typical application involving not one, but several hundred heart cycles and the need to calculate with reasonable accuracy the four blood pressure parameters on a beat-by-beat basis. The task becomes overwhelming. Without question, the process may be simplified by “eyeballing” large groups of waveforms for average maximum, minimum, mean, and rate values. But what happens to accuracy and all-important repeatability?

The obvious solution is to computerize. Everyone knows that computers excel in their ability to evaluate large quantities of numbers. Although it is true that for any number of samples a computer can be programmed to report the highest and lowest values, it fails in its ability to duplicate the intuitive process applied by an experienced researcher when he or she identifies the beginning and end of a heart cycle for example. Since the accurate determination of systolic, diastolic, mean, and rate information depends on an equally accurate determination of heart cycle end points, the computer-based solution fails.

The development of Dataq Instruments' Peak Capture Algorithm focused on solving the end point determination problem since this was the only barrier preventing the computer from automating the labor-intensive process described above. Through innovative software design, and by making the researcher part of the evaluation process, the Peak Capture Algorithm represents the first computer-based solution to the automatic detection of peak, valley, mean, and period information on either a cycle-by-cycle or a 2 to 32,767 cycle average basis.

## **PEAK AND VALLEY CAPTURE (PVC) OPERATION**

PVC does not generate waveforms. Instead, it uses any waveform channel (except channel 1) as an input, then places positive- and/or negative-going event markers directly on the waveform to indicate cyclic peak and/or valley waveform points.

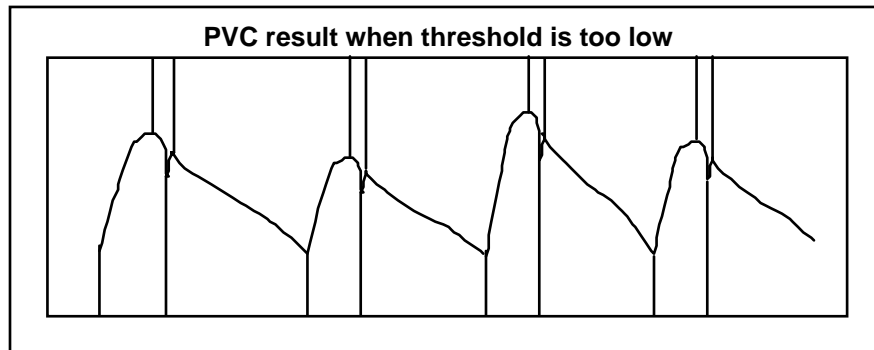


**FIGURE 6A.6**

The Peak and Valley Capture (PVC) function generates positive- and negative-going event marker tags on the input waveform corresponding to cycle-by-cycle maximum and minimum waveform values. PVC may also be configured to detect peaks only, or valleys only. Note that event markers must be enabled from the waveform playback utility before peak and/or valley markers will be displayed.

The procedure used by PVC software to determine cyclic peak and valley waveform values involves the use of a threshold detector. Through command arguments from either the DOS or menu levels, you control the threshold setting as a percentage within the range of 1% to 99%. The default value of the threshold detector is 30%.

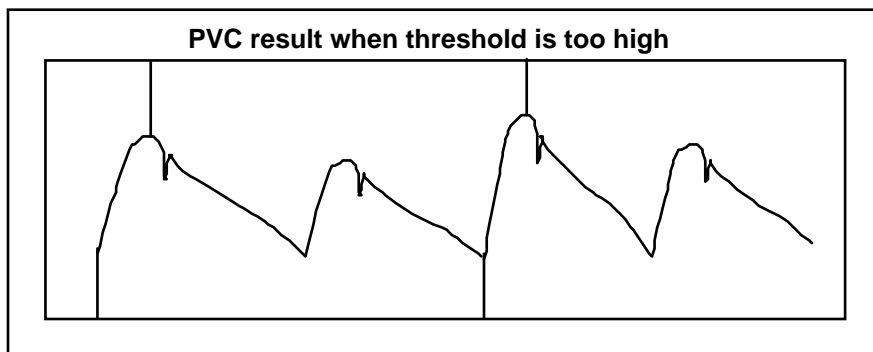
The default threshold level is an extremely good starting point for most waveforms encountered in physiology. Even so, you may have the need to adjust the default threshold level to accurately respond to some waveforms depending upon how they were acquired. FIGURE 6A.6 shows the result of a properly peak captured waveform. FIGURE 6A.7 shows the same waveform that was peak captured using a threshold level that was too low. In contrast, FIGURE 6A.8 shows the same waveform that was peak captured using a threshold level that was too high.



**FIGURE 6A.7**

This figure demonstrates what can happen when the threshold level of the PVC function is too small. The algorithm interprets minor changes in waveform inflections as valid peaks and valleys.

Increasing the threshold percentage will mask these minor inflections, leaving only true cyclic peaks and valleys detected.



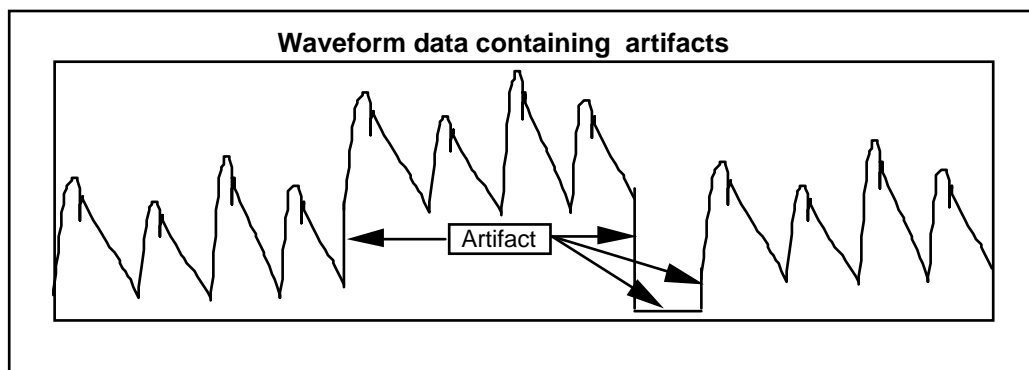
**FIGURE 6A.8**

This figure demonstrates what can happen when the threshold level of the PVC function is too large. The algorithm fails to detect cyclic changes in waveforms that are smaller in amplitude than others in the waveform stream. Decreasing the threshold percentage will increase the sensitivity of the algorithm to allow detection of smaller amplitude, albeit valid waveform cycles.

#### **PEAK CAPTURING IN THE PRESENCE OF WAVEFORM ARTIFACT**

Another consideration when using PVC is to ensure that the waveform data being peak captured is free of artifact (see FIGURE 6A.9) such as that caused by gain changes, loose signal connections, and large step changes caused by stopping then starting disk storage during data acquisition. Should the waveform you need to peak capture contain such artificial components, there are two approaches that may be used to avoid confusing the PVC algorithm:

1. Try peak capturing the waveform as-is using the default threshold level of 30%. If your result indicates that more sensitivity is needed (which will probably be the case if step changes are present) try decreasing the threshold level until the waveform is properly peak captured. In the presence of very large step changes, it may not be possible to properly peak capture the waveform using this approach.
2. Isolate clean waveform data from artifact through use of the waveform playback program's COPY/PASTE FUNCTION. Separate data files may be created, each containing clean waveform data that may be peak captured individually.



**FIGURE 6A.9**

This figure is an example of a waveform containing artifacts which may interfere with PVC's ability to extract proper peak and/or valley information. Approach such situations with caution, and try using the techniques described above to extract useful information.

## ADVANCED CODAS

### GOOD DATA ACQUISITION PRACTICE LEADS TO GOOD PVC PERFORMANCE

The time to start thinking about peak and valley capture of waveform data is *during data acquisition*. The PVC algorithm is a very forgiving utility that can adjust to many variations in waveform quality. However, it can have difficulty if poor data acquisition procedures are used. Listed below are some suggestions related to data acquisition practice that will help you maximize your productive use of PVC:

1. Ensure that the dynamic range of the A-D converter is maximized by closely matching the full scale voltage of your input signal to that of the A-D converter. A-D cards with programmable gain can be used to great advantage in this area by selecting gain factors on a channel-by-channel basis to compensate for varying full scale channel voltages.
2. Minimize noise present on signals to be acquired. Be sure to use quality, shielded signal cable at all points in the signal path.
3. Avoid poor signal connections that could cause intermittent signal drop-outs.
4. Do not make a habit of starting and stopping data acquisition to disk indiscriminately. When acquisition to disk is halted and then started again, an event marker is automatically inserted in channel 1's data stream so that when the waveforms are played back through the waveform playback program, it will be obvious when a discontinuity could exist. This practice will help you avoid peak capturing across data acquisition session boundaries.

### EDITING PVC-SELECTED EVENT MARKERS

The points selected by PVC may be reviewed through use of the waveform playback program. For AT-CODAS user's: run the waveform playback program and select the SEARCH-FOR-EVENT FUNCTION (F8) followed by an "E" keystroke to enable the display of positive and negative event markers. For WINDAQ user's: run the waveform playback program and choose Event Markers... from the View menu. In either case, refer to the SEARCH-FOR-EVENT FUNCTION in the *Post Acquisition using POST or WINDAQ* chapter of the AT-CODAS Users Manual or in the WINDAQ Waveform Browser Users Manual for complete details. Your review of a peak captured waveform will yield one of three possible results:

1. PVC-selected minimum and/or maximum events will be grossly in error.
2. The waveform will be properly peak captured, cycle-by-cycle, end-to-end.
3. Except for a few minor instances, the waveform will be properly peak captured.

In the event of item (1), you should return to PVC and attempt to properly peak capture the input waveform by selecting a different threshold value. Increase or decrease the threshold value depending upon the results of your first attempt, and by using the concepts described in the PEAK AND VALLEY CAPTURE OPERATION section above.

In the event of item (2), no further PVC operations need to be performed. You may proceed with the report generator to produce a point-value file.

You have two options in the event of item (3): You can rerun PVC with a slightly different threshold value in an attempt to correct the few peak capture errors, or you can use the event marker editing facility of the waveform playback program to insert and/or delete event peak and valley markers.

Assuming that peak and valley markers are enabled for display, you are in the waveform playback program with a peak captured waveform displayed, and the data cursor is displayed (WINDAQ user's), the markers may be edited as follows (note that marker editing cannot be used when the waveform compression factor is greater than 1, the waveform must be uncompressed to edit markers):

1. Enable the channel for peak and valley marker editing operations. A channel is enabled for peak and valley marker editing operations by pressing the number (or letter) of the display window containing the waveform to

be modified. For example, say window two is displaying the peak and valley captured result of channel five. You would press the 2 key to enable the channel displayed in window 2 (in this case, channel 5) for marker editing. When pressed, the 2=5 annotation near the bottom of the screen (in AT-CODAS) or in the right or left annotation margin (in WINDAQ) will be highlighted, indicating channel 5 is enabled for marker editing operations.

2. Move to the desired point in the waveform. To move to the next displayed peak or valley marker in the enabled channel, proceed as follows:

#### **AT-CODAS User's**

CTRL+→ moves to the next peak or valley marker by scrolling the waveform in a left (or right) direction depending upon the state of the waveform movement preference switch specified from the CODAS configuration or installation program.

CTRL+← moves to the next peak or valley marker by scrolling the waveform in a right (or left) direction depending upon the state of the waveform movement preference switch specified from the CODAS installation program.

#### **WINDAQ User's**

Choose Next Mark... from the Search menu. This moves the window in a positive time direction to the next enabled event marker, positioning the data cursor directly on top of the event marker.

Choose Previous Mark... from the Search menu. This moves the window in a negative time direction to the last enabled event marker, positioning the data cursor directly on top of the event marker.

3. Edit the file.

#### **AT-CODAS User's**

To delete a marker, press the DEL key on your computer keyboard's key pad while the the waveform playback program cursor is directly over a peak or valley marker.

To insert a peak or valley marker, position the waveform playback program's cursor at the point on a waveform where you'd like a marker to be placed. Pressing the INS key the first time will place a peak marker (positive-going) at the point of waveform/cursor intersection. Pressing the INS key a second time will place a valley marker (negative-going) at the point of waveform/cursor intersection. Alternate pressing of the INS key will toggle between peak and valley marker placement.

#### **WINDAQ User's**

To delete a marker, choose Delete Mark from the Edit menu. This deletes an event marker that is coincident with the waveform cursor.

To insert a peak or valley marker, choose Insert Mark from the Edit menu. This inserts a peak marker (positive-going) at the point of waveform cursor intersection. Repeating the Edit Insert Mark action toggles the peak marker to a valley marker (negative-going).

### **PEAK CAPTURING CHANNEL ONE's WAVEFORM**

As with all Advanced CODAS functions, channel one's analog waveform is considered sacred by PVC meaning that operating on this channel is not directly possible. However, channel one, as well as any other channel may be duplicated by using Advanced CODAS's moving average utility and specifying a moving average factor of 1. Specify the input channel number as that to be duplicated.

### **PVC DOS COMMAND LINE AND CALC MENU SYNTAX**

PEAK *argument*

*argument:*

## ADVANCED CODAS

[d:] [path] filename [.ext] X [C] Y [Z] [S]

where: **[d:][path]filename.ext** specifies the file name containing the waveform to be peak captured. The disk drive specifier, path, and file name extension are optional. A basic file name must be supplied.

### CALC Menu Prompt

### DOS Command Line

Channel to peak capture

**X** is the channel number of the waveform to be peak/valley captured where  $2 \leq X \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $2 \leq X \leq 144$  for WINDAQ EXP multiplexer users. Note that channel 1 may be specified only if peak and/or valley markers are to be written to another channel.

Enter channel number to mark

**C** is an optional channel number within the file to contain the peak captured waveform where  $2 \leq X \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $2 \leq X \leq 144$  for WINDAQ EXP multiplexer users. The default value of **C** is equal to variable **X** providing  $X \neq 1$ .

Select capture mode

**Y** is the capture mode which defines whether the peak/valley capture utility will flag peaks, valleys, or peaks and valleys. Valid entries are:

**Y** = "P" for peaks only

**Y** = "V" for valleys only

**Y** = "PV" or "VP" for peaks and valleys

Select threshold level

**Z** specifies the peak/valley capture utility threshold level where  $1 \leq Z \leq 99$ . The default value for **Z** is 30%. If Peak is given a negative threshold ( $-99 \leq Z \leq -1$ ), its operation is limited to the data between the cursor and the time marker. This is helpful for use on files in which the amplitude changes too much for a single value to work on the whole file. Refer to the PEAK AND VALLEY CAPTURE OPERATION section for details on proper threshold adjustment.

(not displayed)

**S** optionally specifies the smoothing factor applied by the peak detection algorithm. The default value of **S** is 11, which is adequate for most applications. Lower values should be specified when the waveform to be peak captured has been under-sampled. Higher values should be specified in the presence of noise. The range of **S** is  $2 \leq S \leq 64$ .

Example:

PEAK DATA.DAT 2 VP

Peak capture channel 2 of file DATA.DAT for peaks and valleys using the default threshold value of 30%.

PEAK D:\LABDATA\TEST1.DAT 5 4 P 25

Peak capture channel 5 of file TEST1.DAT located on drive D: and subdirectory LABDATA for peaks only using a threshold value of 25%. The peak markers are written to channel 4.



# WAVEFORM RECTIFICATION USING ADVANCED CODAS

## GENERAL

You may find it desirable to rectify a waveform as an interim step before integrating. Although Advanced CODAS's integration utility supports rectification directly, the rectified waveform is not available as an intermediate result. From the perspective of presentation, you can use Advanced CODAS's rectification utility to generate the rectified waveform to one channel, then generate the integral of the rectified waveform to another channel. A hard copy can be generated of all three waveforms (input, rectified, integrated) to fully document analysis flow.

On a more practical side, the full-wave rectification capability of Advanced CODAS can be applied to accurately establish a zero reference in the presence of base line noise. FIGURE 6A.10 illustrates a raw EMG signal acquired in the presence of a high degree of base line noise, and with zero reference established at the midpoint of the bipolar signal (0 volts). Passing this waveform through Advanced CODAS's full-wave (absolute value) rectifier yields the second waveform  $|EMG|$ . Using the waveform playback program's ENGINEERING UNITS FUNCTION (for AT-CODAS users) or WAVEFORM CALIBRATION - FIXED METHOD (for WINDAQ users), zero reference may be reestablished at a point above the previous zero reference to eliminate the entire band of base line noise. The signal of FIGURE 6A.10 labeled as  $+|EMG|$ , is the waveform that will be integrated if a rectify mode of positive-only is selected as an integration option. Note that signal  $+|EMG|$  is never displayed, but rather interpreted in the manner shown by the integrator.

## AUTOMATIC ENGINEERING UNIT CALIBRATION

When a waveform is rectified by Advanced CODAS, the calibration information associated with the data file is automatically adjusted so that when the rectified waveform is reviewed through the waveform playback program a display of its value at any given point is in appropriate engineering units. The engineering unit tag associated with the calculated waveform (and displayed on the waveform playback program screen) will always be "OR#"; where "#" is the channel number of the waveform used as an input to create the rectified waveform, and O describes the applied operation. O may assume one of three possible values: "+" denotes positive only rectification, "-" denotes negative only rectification, and "A" denotes absolute or full wave rectification. For example, assume that waveform channel 1 is full wave rectified and the result is stored to channel 3. Channel 3's engineering unit tag would be "AR1". Likewise, if the operation was positive only half wave, channel 3's engineering units tag would be "+R1". Note that the engineering units tag of any enabled channel may be changed at any time from the waveform playback program by pressing either the F9 or F10 key, then entering up to a 4-character (the first character must be an alpha) descriptor (for AT-CODAS users) or by choosing either Low Calibration... or High Calibration... from the Edit menu, then entering up to a four character descriptor in the Engr. Units = text box (for WINDAQ Waveform Browser users).

## Advanced CODAS's WAVEFORM RECTIFICATION FEATURES

The Advanced CODAS waveform rectification program provides three rectification modes:

### 1. Positive-only half-wave rectification.

Applying this rectification mode to a waveform will yield only positive waveform transitions relative to zero reference.

### 2. Negative-only half-wave rectification.

Applying this rectification mode to a waveform will yield only negative waveform transitions relative to zero reference.

### 3. Absolute value full-wave rectification.

Applying this rectification mode to a waveform will force negative portions of the input waveform to be "folded" above zero reference to join the positive transitions, thus forming the absolute value.

## ADVANCED CODAS

### WAVEFORM RECTIFICATION DOS COMMAND LINE and CALC MENU SYNTAX

RECTIFY *argument*

*argument:*

[d:] [path] filename [.ext] X Y Z

where: **[d:][path]filename[.ext]** specifies the file name containing the waveform to be rectified. The disk drive specifier, path, and file name extension are optional. A basic file name must be supplied.

CALC Menu Prompt  
Input channel number

DOS Command Line

**X** is the channel number of the waveform to be rectified where  $1 \leq X \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $1 \leq X \leq 144$  for WINDAQ EXP multiplexer users.

Output channel number

**Y** is the channel number within the file to contain the rectified waveform where  $2 \leq Y \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $2 \leq Y \leq 144$  for WINDAQ EXP multiplexer users. If  $Y \leq$  the total number of channels contained in the file before the rectification operation, then the result is placed in waveform channel **Y** and the original contents of channel **Y** are *overwritten and destroyed*. If  $Y >$  the total number of channels contained in the file before the rectification operation, then the result is appended to the input file with the channel number equal to the original total number of channels + 1.

Rectification mode

**Z** specifies the rectification mode as follows:

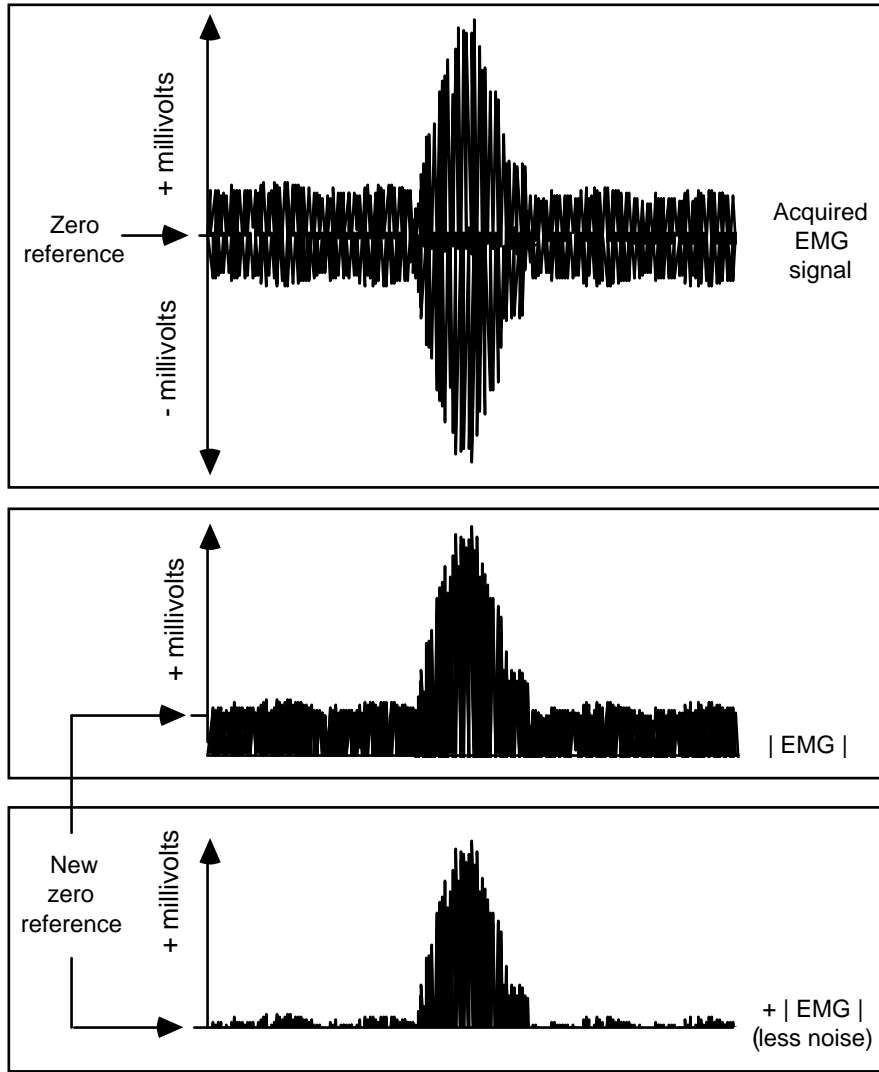
**Z** = "+" passes signal transitions that are greater than or equal to zero reference (positive half-wave rectification).

**Z** = "-" passes signal transitions that are less than or equal to zero reference (negative half-wave rectification).

**Z** = "A" applies an absolute value function so that all signal transitions less than zero reference are folded above zero reference by an equal amount (full-wave rectification).

Example: RECTIFY DATA.DAT 1 6 A

Using waveform data file DATA.DAT, apply the absolute value (full-wave rectification) function to waveform channel 1 storing the result in channel 6. Refer to FIGURE 6A.10.



**FIGURE 6A.10**

This figure demonstrates the application of Advanced CODAS's full-wave rectifier to eliminate the effects of base line noise. The input signal is full-wave rectified to obtain its absolute value. Next, zero reference is reestablished at a level above the original zero reference. When integrated using the positive-only integration mode, the last waveform (+|EMG|) becomes the integrator's input thereby preventing base line noise from distorting the result.

# REPORT GENERATION OF PEAK/VALLEY EVENTS USING ADVANCED CODAS

## GENERAL

The Report Generator (RG) allows peak and valley event markers contained on a waveform (and placed there by the PVC or the peak/valley marker editor) to be extracted as point values. These values may then be placed in a spreadsheet or ASYST and ASYSTANT compatible file along with additional information describing cycle intervals, and sample numbers corresponding with the peak and valley points. RG essentially finishes what was started by PVC by allowing a convenient way to report critical cyclic waveform point-values in a format suitable for further analysis and plotting.

## DEFINING THE PARAMETERS REPORTED BY RG

Whenever it is enabled to report on a peak captured waveform, RG will report six values per cycle (or group of cycles if a cyclic average is selected). These, in order of appearance in the output file, are:

1. **Waveform value at valley marker** — If valley markers are present on the input waveform, the value at waveform/valley marker intersection will be reported in the waveform's unit of measure. If valley markers are not present in the waveform, a value of zero (0) will be reported.
2. **Waveform value at peak marker** — If peak markers are present on the input waveform, the value at waveform/peak marker intersection will be reported in the waveform's unit of measure. If peak markers are not present in the waveform, a value of zero (0) will be reported.
3. **Waveform mean value** — The cyclic mean value of the waveform will be reported, where a cycle is defined by you to be one of the following:
  - From peak marker to peak marker
  - From valley marker to valley marker
  - From valley marker to peak marker
  - From peak marker to valley marker

For example, in the case of an arterial blood pressure waveform a cycle could be defined as valley to valley. The mean value of the waveform would be reported as that calculated between these markers.

Mean values are calculated by RG as the summation of all data points between the indicated markers, divided by elapsed time.

4. **Cycle timing** — Using the cycle definition defined in item (3) above, RG may be configured to calculate and report the cycle interval in seconds, cycle rate in Hertz (cycles per second), or cycle rate in cycles per minute.
5. **Sample number at valley** — If valley markers are present on the input waveform, the sample number at waveform/valley marker intersection will be reported relative to the beginning of the data file. Should the first data point in the waveform contain a valley marker, its sample number would be reported as zero (0). Likewise, should the 1,000th point in the waveform contain a valley marker, its sample number would be reported as "999." If valley markers are not present in the waveform, a value of "-1" will be reported. The sample number may be used to plot valley values versus time, since multiplying the sample number by the reciprocal of the waveform's sample rate per channel yields the location of the value in time.
6. **Sample number at peak** — If peak markers are present on the input waveform, the sample number at waveform/peak marker intersection will be reported relative to the beginning of the data file. Should the first data point in the waveform contain a peak marker, its sample number would be reported as zero (0). Likewise, should the 1,000th point in the waveform contain a peak marker, its sample number would be reported as "999." If peak markers are not present in the waveform, a value of "-1" will be reported. The sample number may be

used to plot peak values versus time, since multiplying the sample number by the reciprocal of the waveform's sample rate yields the location of the value in time.

Note that all values described above may be reported on a cycle-by-cycle, or on a 2 – 32,767-cycle average basis.

### RG DATA FILE CREATION AND STORAGE FORMATS

RG can generate two data file storage formats, each containing the same information: spreadsheet compatible, and ASYST(ANT) compatible.

When configured to create a data file in the spreadsheet storage format, RG generates an ASCII file that may be imported directly to a spreadsheet using (for example) Lotus's IMPORT command. The file may also be typed to the computer's screen or printed to a line printer using DOS's TYPE and PRINT commands. The name of the RG-created file is fixed and carries the same basic name as the waveform file used as the input but with a ".LX" extension. X will be a number ranging from 2 to 29 for AT-CODAS and WINDAQ non-EXP multiplexer users or from 2 to 144 for WINDAQ EXP multiplexer users and will be equal to the channel number of the input waveform file being reported. For example, if channel number 5 in file TEST.DAT is reported, the Lotus-compatible file created on disk will carry a file name of "TEST.L5".

When configured to create a data file in the ASYST(ANT) storage format, RG generates a floating point binary file that may be imported directly to either of two packages provided by ASYST Software Technologies: ASYST or ASYSTANT. The name of the RG-created file is fixed and carries the same basic name as the waveform file used as the input but with a ".AX" extension. X will be a number ranging from 2 to 29 for AT-CODAS and WINDAQ non-EXP multiplexer users or from 2 to 144 for WINDAQ EXP multiplexer users and will be equal to the channel number of the input waveform file being reported. For example, if channel number 16 in file RUN1.DAT is reported, the ASYST(ANT)-compatible file created on disk will carry a file name of "RUN1.A16".

Imported data is handled by ASYST or ASYSTANT as a two dimensional array where one dimension is the number of cycle sets imported, and the second dimension is always 6. The second dimension (handled as rows by ASYSTANT and ASYST) are each of the six data categories described in the parameters section above.

Both file formats contain the following comments which describe various parameters of the data file:

1. Name of the parent data file from which the peak captured data was derived.
2. The channel number of the peak captured waveform being reported.
3. The engineering units of the reported waveform.
4. The sample rate at which the peak captured waveform was acquired.
5. The cycle definition used by RG (may be V-V, P-P, P-V, or V-P)
6. The number of cycles over which point values are averaged (1 – 32,767)
7. This file MUST BE TRANSPOSED!!!\*

\* This comment is found in ASYST(ANT) data files only.

The last comment, " This file MUST BE TRANSPOSED!!!", is found in ASYST(ANT) data files only and serves as a reminder to transpose the data file before operating on it. The transpose command (TRANS) inverts the sense of the two dimensional array generated when an RG file is imported to ASYST or ASYSTANT. For example, assume data for 500 cycles is imported by ASYSTANT. The two dimensional array initially created and stored in an array variable (assume array variable R is used) has a dimension of "500 x 6". Placing array variable R on the stack, then typing "TRANS" changes the sense of the array to "6 x 500". After restoring the transposed array to R, analysis may begin.

## ADVANCED CODAS

### RG DOS COMMAND LINE AND CALC MENU SYNTAX

REPORT *argument*

*argument:*

[d:] [path] filename [.ext] X Y Z [R] [A]

where: **[d:][path]filename.ext** specifies the file name containing the waveform to be reported. The disk drive specifier, path, and file name extension are optional. A basic file name must be supplied.

#### CALC Menu Prompt

Channel to report

#### DOS Command Line

**X** is the channel number of the peak captured waveform to be reported where  $1 \leq X \leq 29$  for AT-CODAS and WINDAQ non-EXP multiplexer users or  $1 \leq X \leq 144$  for WINDAQ EXP multiplexer users.

Select cycle definition

**Y** defines waveform cycle end points used for calculating mean values and reporting the cycle interval in seconds, rate in terms of cycles per second (Hertz), or rate in terms of cycles per minute. Valid **Y** entries are

**Y** = "PP" for cycles defined from peak to peak

**Y** = "VV" for cycles defined from valley to valley

**Y** = "PV" for cycles defined from peak to valley

**Y** = "VP" for cycles defined from valley to peak

Select file output format

Variable **Z** defines the format of the output file to be either spreadsheet or ASYST and ASYSTANT compatible:

**Z** = "A" for a ASYST and ASYSTANT compatible file format

**Z** = "L" for a spreadsheet compatible file format

Select cycle units

Optional variable **R** defines whether a cycle is reported as an interval (in seconds), or as a rate in Hertz (cycles per second) or cycles per minute. The default state of **R** is interval.

**R** = "I" reports cycle time in seconds (default)

**R** = "H" reports cycle rate in Hertz (cycles per second)

**R** = "M" reports cycle rate in cycles per minute

Select cycle average

Optional variable **A** defines the number of consecutive cycles to average before reporting where  $1 \leq A \leq 32,767$ . The default value of **A** is 1; each cycle value is reported without averaging.

Example:

```
REPORT DATA1.DAT 4 VP A
```

Generates file "DATA1.A4" containing cycle-by-cycle point values of waveform channel 4 of file DATA1.DAT. Output format is ASYST and ASYSTANT compatible, with a cycle definition of valley to peak. Cycle timing will be reported in seconds.

```
REPORT D:\TESTS\SET.1 12 PP L M 8
```

Generates file "SET.L12" in subdirectory TESTS containing 8-cycle average point values of waveform channel 12 of file SET.1. Output format is spreadsheet compatible, with a cycle definition of peak to peak. Cycle timing will be reported as a rate in cycles per minute units.

# WAVEFORM COPY FEATURE USING ADVANCED CODAS

## GENERAL

The Waveform Copy Feature allows you to copy to a destination file starting at a specified destination point, from a source channel in the same file or another file. The copied information includes calibration, units, scaling, and user annotation, but excludes acquisition channel, method, input type, gain, and input level range.

In WinDaq Waveform Browser, set Options Select Marker Display to “Samples from Marker” and set Options Time Marker at the start of the file to view the point numbers required by the Copy Channel program when copying part of a destination or source channel.

The default destination channel is the number of channels plus 1, which adds a new channel to the file. The default start point in the destination channel is 0, which copies to the start of the file, and other values of this parameter leave the specified number of data values in the destination channel (or baseline when adding a channel) untouched to the left of the first data copied from the source channel.

The default source file is the same as the destination and the default source channel is 1. The default start point in the source is 0, which copies from the start of the file, and other values of this parameter specify the number of data values to skip in the source channel. If some of the source data would be copied past the end of the destination file, it would be discarded.

Although the Advanced CODAS menus disallow channel 1 as a destination channel to protect acquisition starts and event marks, COPYCHAN itself, when invoked from a command line, allows this if both starting points are zero. However, the feature will copy the event mark information from the trailer of the source file to the destination (or part of it if the destination file is shorter), along with the acquisition start and end times from the source file header. It is also recommended, but not required, that the source channel be 1 so the Option Event Marks will show corresponding event marks in its channel 1 data. To see the format of a COPYCHAN command line, type COPYCHAN with no parameters from an MS-DOS prompt.

Some warning messages:

“Source sample rate differs from that of destination file and will be ignored.” Copy Channel never changes the sample rate per channel in the destination file, so the time scale of the data just copied from the source is incorrect.

“Part of destination not written had calibration changed to that of source.” This means that some of the original data remains, calibrated inappropriately. There is no message when adding a channel, in which case points outside the copied data have the value of the calibration intercept (baseline after WWB Scaling Show Dynamic Range).

“Source and dest ranges overlap, and updated dest may be used as source.” The source and destination channels are the same channel in the same file. Since the start point in the source is to the left of the start point in the destination, data copied from the source phase shifted into the destination may have been reread and reshifted later in the process. This causes corruption to the channel data instead of just phase shifting it. This can be avoided by making a copy of the channel to be shifted first.

### CALC Menu Prompt

### DOS Command Line

Destination file name

[d:][path]filename[.ext] This specifies the destination file to which you are copying a selected channel or portion of a channel

Destination channel number

Specifies the channel that will display the copied data.

## ADVANCED CODAS

|                                    |                                                                                           |
|------------------------------------|-------------------------------------------------------------------------------------------|
| Start point in destination channel | Specifies at what point on the channel the data will begin to be displayed.               |
| Source filename                    | [d:][path]filename[.ext] This specifies the source filename of the data you want to copy. |
| Source channel number              | Specifies the channel in the source file that contains the data you want to copy.         |
| Start point in source channel      | Specifies the point in the source file that you would like to begin copying.              |