SINEWAVE SYNTHESIS

INSTRUCTION MANUAL

(VAX)

Philip Rubin
Version 2.1
August 11, 1982

TABLE OF CONTENTS

I. INTRODUCTION

The SineWave Synthesizer (SWS) creates an output sampled data file (in PCM format) from an input specification that includes information about the sinewave frequencies, the sinewave amplitudes, and timing.
Timing information is specified as vertical time slices (in msec.),
for which there are corresponding sinewave frequencies and amplitudes.

As presently configured, SWS can handle a maximum of 20 sinewaves and 999 time slices. The first time slice should be at 0. msec.

There are 4 modes of input in SWS:

I Input by prompt: in which the computer explicitly prompts you for the appropriate input information.

A Input by art : amplitude, frequency, and timing values are specified by using the cross-hair cursors and the keyboard.

RS Input from a disk file: reads in input information from a previously created disk file (see also "ER").

RI Input of ILS data: reads information created by the special ILS "ANAL" procedure.

Default extensions used in SWS include:

.SWI - Sine Wave Input
.SWO - Sine Wave Output (in PCM format)
.SER - serial SYN file.
.CSW - Sine Wave Command file

For a list of SWS commands, type M<cr> in response to the SWS> prompt. A short description of each command is provided.

SWS output support includes access to WENDY, VAX audio output, and a facility for saving synthesized data in PCM format files. Sinewave data files (.SWI) can be edited with EDT or EDI or with SWS's own editor commands.

The default sampling frequency in SWS is 10,000 Hz.

II. RUNNING THE SINE WAVE SYNTHESIZER

To run SWS, type in response to the DCL prompt, $:

SWS<cr>

SWS will run on any terminal, but graphics capabilities are possible only when using graphics terminals.

If you wish to generate PCM data using SWS, you must first assign an output channel. This should be done before entering SWS by using AUDIO (e.g. $AUDIO RES/OUT<cr>).

Getting started with SWS involves first giving it some input. This can be done by art (see below) if using a graphics terminal, by prompt from within the program (see also below), or by creating a data file using one of the VAX editors.

A sinewave synthesis input file consists of a time-slice by time-slice listing of the sinewave frequency and amplitude values. The first item in this file should be the number of sinewaves. Then, for each time slice, the starting time of the slice must be entered followed by pairs of frequencies (in Hz) and amplitudes (from 0. to 1.) for each of the number of sinewaves specified. Below is an example of such a file:

```
2
0
100,0.
200,0.
90
100,.5
500,1.
180
100,0.
200,0.
```

Frequencies are entered in Hz, with a resolution of 5 decimal places.
Amplitude values are on a linear scale for 0. to 1. (6 decimal places).

Figure 1 shows the output for this sinewave data. The token is 180 msec. in duration. There are 2 sinewaves. The bottom one is at a constant frequency of 100 Hz and rises and falls in amplitude. The top sinewave starts at 200 Hz, rises to 500 Hz by 90 msec., and then falls back to 200 Hz. It also rises and falls in amplitude. The abscissa of the display represents time; the ordinate represents frequency. The height of sinewaves corresponds to amplitude. Sine waves of 0. amplitude are not displayed. Vertical lines within a sinewave represent time slices.

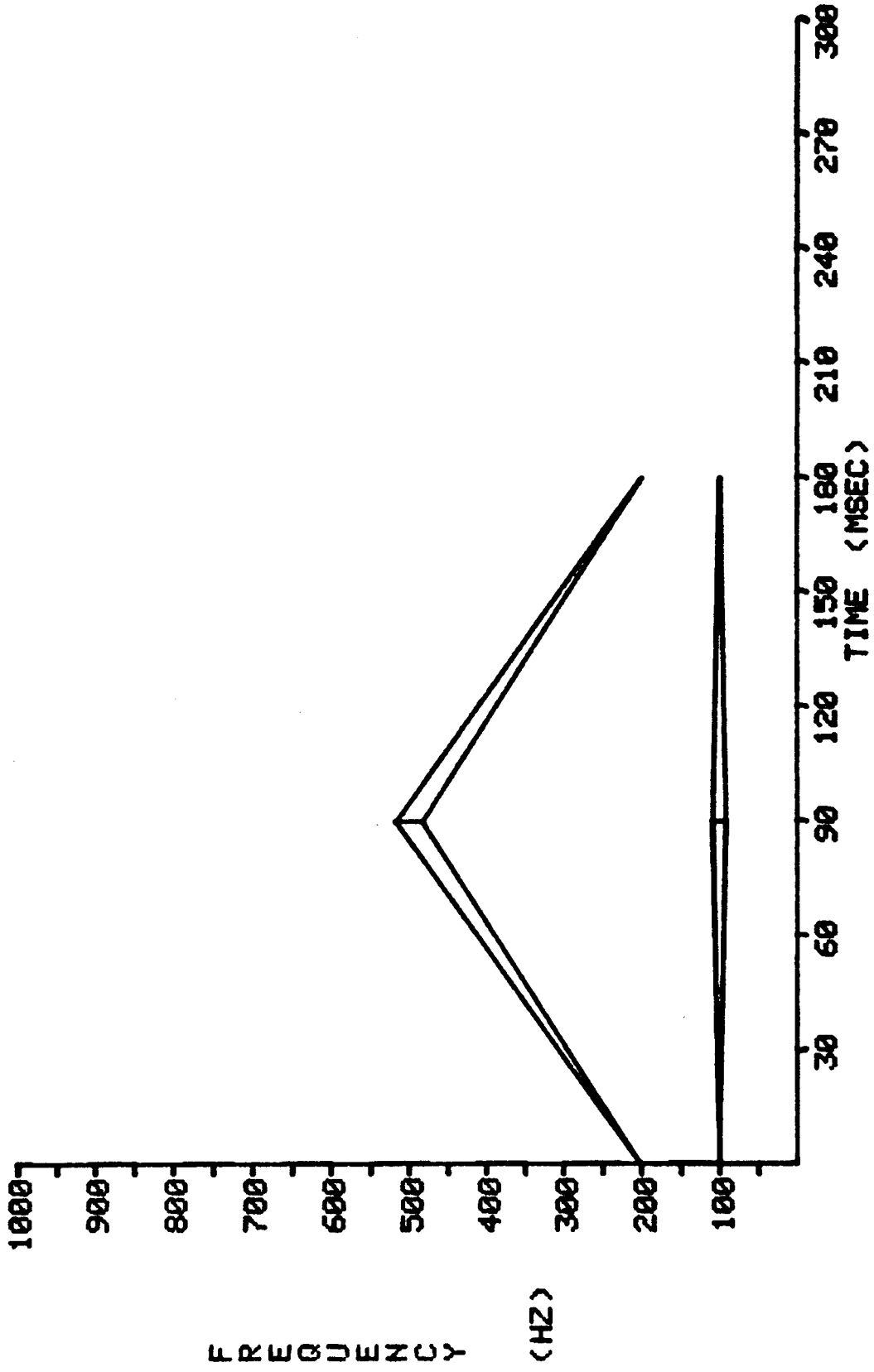

Figure 1: Sine Wave Display

III. COMMAND LINE INFORMATION

Command Line Format:

<CMD> [TEXT1,][TEXT2,][VALUE],...[VALUE]<cr>

CMD — 1 to 3 letter SWS command.

TEXT1 — optional text string (depends upon the specific command).

TEXT2 — optional text string (depends upon the specific command).

VALUE — optional numeric argument or arguments (depends upon the specific command; floating point values can be used).

Unspecified arguments (or blank commas) will result in the use of default values, if appropriate, else the user will be prompted for the values.
If the user types "?" in place of a value, the system will prompt for all succeeding values for that command.

SWS commands can be concatenated on a single line through the use of the "&" symbol.
For example, SWS>ER BAH&G&P 3<cr> will result in SWS reading the file BAH.SWI and then generating sinewave output information, if possible, and then playing the result through the output system 3 times.

SWS also has a command file input mode. Using the system editor, a list of SWS commands can be created. In response to the SWS> prompt, this command file can be invoked by typing the name of the command file preceded by an at sign "@".

Example: Assume we have created the following file with the editor.

```
ER BAH
G
P 3
```

We will call this file FOO.CSW. In SWS : SWS>@FOO
will open the file, begin reading it, and executing the SWS commands. The result will be the same as that in the previous example.
Nested command files are not allowed.

Notation: In this documentation, the angle bracket "<>" is being used to represent two things. First, <> is being used to enclose a control key. E.g. <cr> means press the NEW LINE, RETURN, or CARRIAGE RETURN key. "<>" can also be used to indicate some variable name. E.g. <filename> means enter some variable filename of your choosing.
[] indicates an optional variable argument.
Underlining indicates messages typed by the computer.

IV. MENU OF SWS COMMANDS

I	Input data by prompt	:	EDITOR Commands:
A	input data by Art	:	
RS	Read Sinewave data	:	EL List sinewave data
RI	Read ILS data	:	ER Read sinewave data
L	List sinewave data	:	EO Overlay from file
D	Display	:	ED Delete sinewave data
DA	Display Amplitude	:	EM Multiply
G	Generate PCM data	:	EA Add
P	Play PCM data	:	EI Interpolate
PC	Play Comparison	:	EH Harmonic creation
SP	Save PCM data	:	EE Edit time and sinewave data
SS	Save Sinewave data	:	ES Shift sinewave
SY	SYN data creation	:	EQ sQuash sinewave frequency
HO	Harmonic Oscillator	:	
DD	Derivative Display	:	MISC. Commands:
AV	AVerage frequency, amplitude	:	
SET	SET variables, flags	:	H Help
SHO	SHOw variables, flag	:	? info. about SWS
W	Wendy	:	M Menu of SWS commands
HC	HardCopy of screen	:	X eXit SWS

V. SWS COMMANDS

A SINEWAVE DATA INPUT BY ART.

Values for sinewaves are specified by drawing a point, with an associated amplitude, at the desired frequency. Entries must be made a slice at a time. Thus, if you have 3 sinewaves, and you are at the first time slice, you first enter the value for sinewave 1, then sinewave 2, and finally sinewave 3. You then proceed to time slice 2, starting over again with sinewave 1.
The first time slice should begin at 0. msec.
Time slices should be ordered across time.

To enter a value for a sinewave, position the cross-hair cursors and enter a number from 0 to 9. This number corresponds to an amplitude on a scale of 0 to 10, with 0 being silence and 9 being almost maximum amplitude. If you have more than 1 sinewave, use the vertical cross-hair to select the time position. Once this has been selected for the first sinewave, it should not be moved for the other sinewaves at this time slice. Instead, only the horizontal cross-hair should be used to select the frequency. When you have finished entering the value for the final sinewave, at the last desired time slice, press "T" to terminate input. When finished, the display will be redrawn. The data can later be saved in a .SWI file for reentry or editing.

Command Format:

A<cr>

AV AVERAGE frequency and amplitude:

Reports the average frequency and amplitude for all sinewaves, between the specified stop and end times.

Command Format:

AV [start],[end]<cr>

START : starting time in msec. (def.= 0.).

END : ending time in msec. (def.= END).

D DISPLAY sinewave data on the screen:

Command Format:

```
D [start],[end],[sw #]<cr>

    START : starting time in msec. (def.=  0.).
    END   : ending   time in msec. (def.= END).
    SW #  : # of sinewave to be displayed
            (default=ALL).
```

The time and frequency axis maxima will be set to automatic scaling unless the TMAX and FMAX flags have been altered using the "SET" command. You can use these flags to personalize the display size. See SET for further information. Be aware that the frequency axis is constrained to units of 500 Hz and the time axis to 100 msec. units. If frequency data exceed the FMAX value, automatic scaling will be used. However, the time axis (TMAX) can be a subset of the total time slice data.

DA DISPLAY AMPLITUDE:

Plots time in msec. vs. amplitude (on a scale of 0. - 1.) on the screen. (See Figure 2.)

Command Format:

```
DA [start],[end],[sw #]<cr>

    START : starting time in msec. (def.=  0.).
    END   : ending   time in msec. (def.= END).
    SW #  : # of sinewave to be displayed
            (default= 1 ).
```

As with the "D" command, the time scale can be overriden by changing the TMAX flag. See "D" and "SET" for further information.
The time axis is constrained to units of 100 msec.

DD DERIVATIVE DISPLAY:

This command performs a difference equation, using a single sinewave as input, and displays the output. Either frequency or amplitude values can be used. (See Figure 3.)

Note: The time-slice to time-slice intervals must be equal across the entire data set.

Derivative 0: raw data
Derivative 1: velocity
Derivative 2: acceleration

Command Format:

DD [frkamp,][d #],[sw #],[scale],[scmin],[scmax]<cr>

FRKAMP : frequency (F) or amplitude (A) calculation (def.=F).
D # : derivative # (0-3; def. = 1).
SW # : sinewave # (def. = 1).
SCALE : automatic (0) or manual (1) (def. = 0).
Automatic scale adjusts the display size.
Manual scale permits displays that are larger than those that are automatically scaled.
If SCALE = 1, the following are used:
SCMIN : display minimum (def.=automatic).
SCMAX : display maximum (def.=automatic).

EA EDIT: ADD.

Adds a constant factor to either the frequency, amplitude, or time slice values.

Command Format:

EA [mode,][start],[end],[sw #],[add]<cr>

MODE : type of data to modify, where:
F=frequency (D)
A=amplitude
T=time
START : starting time in msec. (def.= 0.).
END : ending time in msec. (def.= END).
SW # : # of sinewave to be added to (default=ALL).
ADD : additive factor.

ED DELETE sinewave data.

Command Format:

ED [start],[end],[sw #],[type]<cr>

START : starting time in msec. (def.= 0.).
END : ending time in msec. (def.=END).
SW # : sinewave to be deleted (def.=ALL).
TYPE : type of deletion (def.= 0).
0 = Normal. Does not justify time slice values if entire slices are deleted.
1 = Compress. Adjust time slices. If an entire sinewave is deleted, the total number of sinewaves will be decreased by one.

EE EDIT: EDIT time and sinewave data.

Using this command, time slice information and sinewave frequency and amplitude values can be modified, starting at the specified time slice.

This command involves interactive entry.
The present time slice and frequency-amplitude values will be printed.
To change values, enter new values (both for frequency and amplitude).
If a value is to be left unchanged, hit <cr>.
Typing "P" for time or frequency values will bring you back to the previous time slice, if one exists.
Typing "X" for a time or frequency value will exit the EDIT routine and return you to the SWS> prompt.
Time slice values must be properly ordered.

Command Format:

EE [start],[sw #]<cr>

START : starting time in msec. (def.= 0.).
SW # : sinewave to be edited (def.=ALL).

EH EDIT: HARMONIC creation.

Creates an harmonic of an existing sinewave by multiplying or adding to the frequency value.
The amplitudes can also be scaled.

Command Format:

```
EH [mode,][start],[end],[sw #],[value],[amult]<cr>

    MODE  : type of frequency modification, where:
                M=multiply frequencies (D)
                A=add to   frequencies
    START : starting time in msec. (def.=  0.).
    END   : ending   time in msec. (def.= END).
    SW #  : # of sinewave to be modified
              (default=ALL).
    VALUE : value used to get new frequencies:
              If MODE="M", VALUE=frequency multiplier.
              If MODE="A", VALUE=additive constant in Hz.
    AMULT : amplitude multiplier   (D= 1. ).
```

EI EDIT: INTERPOLATE.

Interpolation for frequency, amplitude or time slice values.
Interpolation is linear between two user-specified values.

Command Format:

```
EI [mode,][start],[end],[sw #],[val1],[val2]<cr>

    MODE  : type of data to modify, where:
                F=frequency (D)
                A=amplitude
                T=time
    START : starting time in msec. (def.=  0.).
    END   : ending   time in msec. (def.= END).
    SW #  : # of sinewave to be interpolated
              (default=ALL).
    VAL1  : starting interpolation value.
    VAL2  : ending   interpolation value.
```

EL EDIT: LIST.

Lists the present sinewave data on the selected listing device.

Command Format:

```
EL [start],[end],[sw #]<cr>

    START : starting time in msec.  (def.= 0.).
    END   : ending   time in msec.  (def.=END).
    SW #  : sinewave to be listed   (def.=ALL).
```

Notes:

The listing can be sent to the printer by setting the LIST flag to LP (see "SET").
This command is the same as the "L" command.

EM EDIT: MULTIPLY.

Multiplies either the frequency, amplitude, or time slice values by a constant factor.

Command Format:

```
EM [mode,][start],[end],[sw #],[mult]<cr>

    MODE  : type of data to modify, where:
              F=frequency (D)
              A=amplitude
              T=time
    START : starting time in msec. (def.= 0.).
    END   : ending   time in msec. (def.= END).
    SW #  : # of sinewave to be multiplied
             (default=ALL).
    MULT  : multiplicative factor.
```

EO EDIT: OVERLAY.

Overlay data from a diskfile on top of the present sinewave data.

Command Format:

```
EO <filename><cr>

    FILENAME : name of a sinewave input file
               (default extension = .SWI ).
```

The overlay data can correspond exactly to the present sinewave data (base data), to a subset of the base data, can be appended to the base data, etc. At any point that the overlay data overlaps the base data, the time values must correspond exactly.

EQ EDIT: SQUASH frequency values.

Compresses or expands a sinewave's frequency values.
A constant value is entered for a sinewave. Departures of the sinewave's frequency values from this constant are multiplied by the specified multiplier value. If this multiplier value is 0., the resulting frequency values we all be set to the constant. If the multiplier value is 1., the frequency values will be left unchanged.

Command Format:

EQ [start],[end],[sw #],[const],[mult]<cr>

START : starting time in msec. (def.= 0.).
END : ending time in msec. (def.= END).
SW # : # of sinewave to be modified
(default=ALL).
CONST : frequency constant.
MULT : departure multiplier.

ER EDIT: READ.

Read data from a sinewave input file.

This command is the same as the "RS" command, except that it will not automatically generate either a display or PCM data. To display, use the "D" command. To generate PCM data, use the "G" command.

Command Format:

ER <filename><cr>

FILENAME : name of a sinewave input file
(default extension = .SWI).

ES EDIT: SHIFT sinewave.

This command lets you shift an entire sinewave forward or backwards in time. The amount to be shifted, which can be positive or negative, is specified in terms of the number of slices. The time slice values for the entire file must be uniform. That is, the time difference, from slice to slice, must be equal in all cases.

Command Format:

ES [# shift],[sw #]<cr>

SHIFT : the number of slices to shift.
Can be positive or negative.
SW # : the number of the sinewave that
you want to shift (D.=1).

You must have at least 3 time slices to use this command.

*** NOTE ***
Be careful when shifting a sinewave backwards in time (when # SHIFT<0). Data can lost at the lefthand edge.

G GENERATE PCM DATA:

This command generates PCM data from the input sinewave data.

This PCM data can be played out using the "P" command, looked at in WENDY ("W"), or saved in a disk file ("SP").

In general, SWS synthesizes sinewaves - that's why its called a sinewave synthesizer. However, different waveform types or "voice qualities" can be produced. These can be obtained by changing the VOICE flag using the SET command. At present, the voice qualities include:
(1) sinewave; (2) sawtooth; (3) square wave; and (4) band-passed noise.

The default sampling rate for output generation is determined by the SAMPR flag (the default is 10000 Hz). This flag can also be changed with the SET command. PCM output through the VAX hardware can only be at 10,000 or 20,000 Hz.

See SET for further information on the VOICE and SAMPR flags.

Command Format:

G<cr>

H HELP.

HELP provides information about SWS commands.

Command Format:

H <command code><cr>

COMMAND CODE : 1 to 3 letter SWS command.

HC HARDCOPY.

This command makes a hardcopy of the Tektronix screen.

Notes:
This command can only used on a Tektronix terminal that is attached to a hardcopy device.
Make sure that the hardcopy device has been turned on, and is warmed up.

Command Format:
HC<cr>

HO HARMONIC OSCILLATOR:

Damped harmonic motion model for changes in sinewave frequency or amplitude.

This is a special command that is, at present, being used for experimental purposes. HO simulates the oscillatory motion of a linear spring, including damping, based upon the entry of values for stiffness of the spring, mass, resistance, resting length, and initial displacement and velocity. Either frequency or amplitude values can be made to oscillate. In the default condition, three parallel sinewaves are created: the base wave, and the third and fifth harmonics. Optionally, up to nine ordered harmonics can be generated. (See Figure 4.)

Command Format:

HO <codes,><tinc>,<dur>,<k>,<m>,<r>,<rl>,<id>,<iv>,<fv>,<#h><cr>

CODES : 2-letter code, where:
First letter indicates frequency (F) or amplitude (A) motion (default=F).
Second letter indicates whether a new file should be created (N) or an old file be modified (O) (def.=N).
TINC : time increment in msec. (def. = 10.).
DUR : duration in msec. (def. = 1000.).
K : stiffness.
M : mass.
R : resistance.
RL : resting length (FREQ. or AMP.).
(If FREQ, min.=1., max.=1000., def.=500.).
(If AMP , min.=0., max.= 1., def.=0.5).
ID : initial displacement (FREQ. or AMP.).
(If FREQ, min.=-1000., max.=1000.).
(If AMP , min.= -1., max.= 1.).
IV : initial velocity.
FV : fixed modifier value (FREQ. or AMP.).
If using FREQ, FV is amplitude (def.= 0.5).
If using AMP , FV is frequency (def.= 500.).
#H : # of harmonics:
(Max. = 9 ; default = 3rd and 5th).

I INPUT sinewave data by prompt:

Values in SWS are entered on a frame-by-frame basis. In SWS these frames are known as "time slices".
The value for the first time slice should be 0. This corresponds to a time of 0. msec.

This routine steps through the sinewaves time-slice by time-slice. A single value should be entered for each time slice. Then, for each sinewave, 2 values should be entered: one for the frequency and one for the amplitude. Frequency is in Hz. Amplitude runs on a linear scale from 0. to 1. After the values for all sinewaves in a time slice have been entered, continue with the next time slice.

The letters "R" and "I" can be substituted for frequency and/or amplitude entries.

"R": repeat value entered for previous time slice.
"I": interpolate from last numeric value until the next numeric value.

Command Format:

I<cr>

L LIST:

Lists the present sinewave data on the selected listing device.

Command Format:

L [start],[end],[sw #]<cr>

START : starting time in msec. (def.= 0.).
END : ending time in msec. (def.=END).
SW # : sinewave to be listed (def.=ALL).

Notes:

The listing can be sent to the printer by setting the LIST flag to LP (see "SET").
This command is the same as the "EL" command.

M MENU.

Prints the menu of SWS commands.

Command Format:

```
M<cr>
```

P PLAY PCM DATA:

Play speech through the VAX output system. An output channel should have been previously assigned using the VAX AUDIO command.

"P" can be used to play out any file in PCM format by specifying the name of the file as the first argument.

If no filename is specified, "P" will play out the default output generation buffer.

Command Format:

```
P  [filename,][# reps.],[isi]<cr>

    FILENAME : stored PCM format file
               (def. extension is .SWO).
               If no filename is specified, the
               default buffer will be played.
    # REPS.  : # of repetitions (def.=  1).
    ISI      : ISI in seconds   (def.= 0.5).
```

Note: Sinewave data can be generated at any sampling rate, but at the present time our output system supports only 10000 Hz and 20000 Hz.

PC PLAY COMPARISON:

The "PC" command plays out the default speech buffer, followed by an ISI, followed by a user-specified stored PCM format file, followed by an ISI.

Command Format:

```
PC <filename>,[# reps.],[isi]<cr>

    FILENAME : stored PCM format file
               (def. extension is .SWO).
    # REPS.  : # of repetitions of the
                    sequence    (def.=  1).
    ISI      : ISI in seconds   (def.= 0.5).
```

RI READ ILS data:

This command reads data that was created by an ILS-compatible routine. This data file contains information about the first 5 formants of an analyzed file, including their bandwidths and formant amplitudes. To produce this file, you must be familiar with the VAX version of ILS. The file is produced by using the "ANAL" procedure.

Note that because of the characteristics of the ILS LPC-based analysis, the converted sinewave values will need careful editing.

Command Format:

RI<cr>

RS READ SINEWAVE DATA FROM A DISK FILE.

Reads sinewave data from a file, displays the data, generates the PCM buffer, and attempts to play out this buffer one time.

Command Format:

RS <filename (def.extension = .SWI)><cr>

Sample input file:

```
          2
          0.
          500.,0.
          1000.,0.
          50.
          750,.5
          1000.,.2
```

Line 1: the number of sinewaves.
Line 2: the time of the first time slice in msec.
Line 3: sinewave 1 frequency, amplitude pair.
Line 4: sinewave 2 frequency, amplitude pair.
Line 5: the time of the second time slice in msec.
Line 6: sinewave 1 frequency, amplitude pair.
Line 7: sinewave 1 frequency, amplitude pair.
The file would then continue with the next time slice.

SET

SET is used to set the status of various flags in the program.

Command Format:

SET <flag>=<value><cr>

The flags and their values (default=D) are:

LIST	listing device	: TT (D) or LP.
SAMPR	sampling rate in Hz	: (D=10000.).
VOICE	voice quality, where:	
	1 = sinewave (D)	
	2 = sawtooth	
	3 = square wave	
	4 = band-passed noise.	
TMAX	time max. for display	(D=scale).
FMAX	frequency max. for display	(D=scale).

The "scale" flags for TMAX and FMAX provide for automatic scaling for the time and frequency axes. Be aware that the frequency axis is constrained to units of 500 Hz and the time axis to 100 msec. units. Automatic scaling expands the display, within the above constraints, to provide a picture of all the data.

SHO

SHO shows the status of selected variables and flags.

The options are:

LIST	listing device.
SAMPR	sampling rate in Hz.
VOICE	voice quality.
TMAX	time max. for display.
FMAX	frequency max. for display.
N	number of slices in buffer.
ALL	all of the above.

Command Format:

SHO <choice><cr>

SP SAVE PCM data in a disk file:

This command copies the PCM data in the SWS buffer to a new file, whose filename is specified by the user. The new file is written in Haskins PCM format.

Command Format:

SP <filename><cr>

FILENAME : name of output file.
(Default extension = .SWO).

SS SAVE SINEWAVE data:

This command saves the input sinewave data in an ASCII file.
The data can be edited with the SWS editor functions, or with the system editors.

Command Format:

SS <filename><cr>

FILENAME : name of output ASCII file.
(default extension = .SWI).

SY SYN data creation:

This command converts sinewave input information into data that is compatible with the IGM serial synthesizer. This information is written to an output file.

Amplitude scaling is used.
The fundamental frequency (F0) is simulated.

Command Format:

SY <filename>,[amult],[arange],[achg],[f0lo],[f0hi],[f0chg]<cr>

FILENAME: name of output SYN file
(def. extension is .SER.)
AMULT : output amplitude multipler (D=1.).
ARANGE : % range for amp. above/below mean (D=20).
ACHG : max. change/pitch pulse in db (D=1.).
F0LO : F0 lower limit in Hz (D=100.).
F0HI : F0 upper limit in Hz (D=120.).
F0CHG : max. change/pitch pulse in Hz (D=1.).

W WENDY:

Invoke WENDY, the Waveform Editor and Display program.
WENDY is a separate program from SWS and is called as a sub-process.

To examine the default speech generation file, do the following upon entering WENDY:
Type: TOOLS:SWS.WEN<ctrl-A> .

This reads in a WENDY command file. If you are going to be using this procedure often, copy the file TOOLS:SWS.WEN to your own directory.

This WENDY command file does a number of things.
It creates two ports in time mode, reads the default output file into port 0, and displays it. It also creates the following three procedures:

0<cr>	play the contents of port 0.
1<cr>	play the contents of port 1.
X<cr>	exit WENDY.

When you exit WENDY, you will be returned to SWS.

Command Format:

W<cr>

X EXIT:

Typing "X" in response to the SWS> prompt exits the Sinewave Synthesis program and returns you to the DCL prompt, $.

Command Format:

X<cr>

? INFORMATION concerning SWS:

This command provides supplementary information concerning the Sinewave Synthesis program, SWS.

Command Format:

?<cr>

[Insert Figure 2 here: Amplitude Display.]

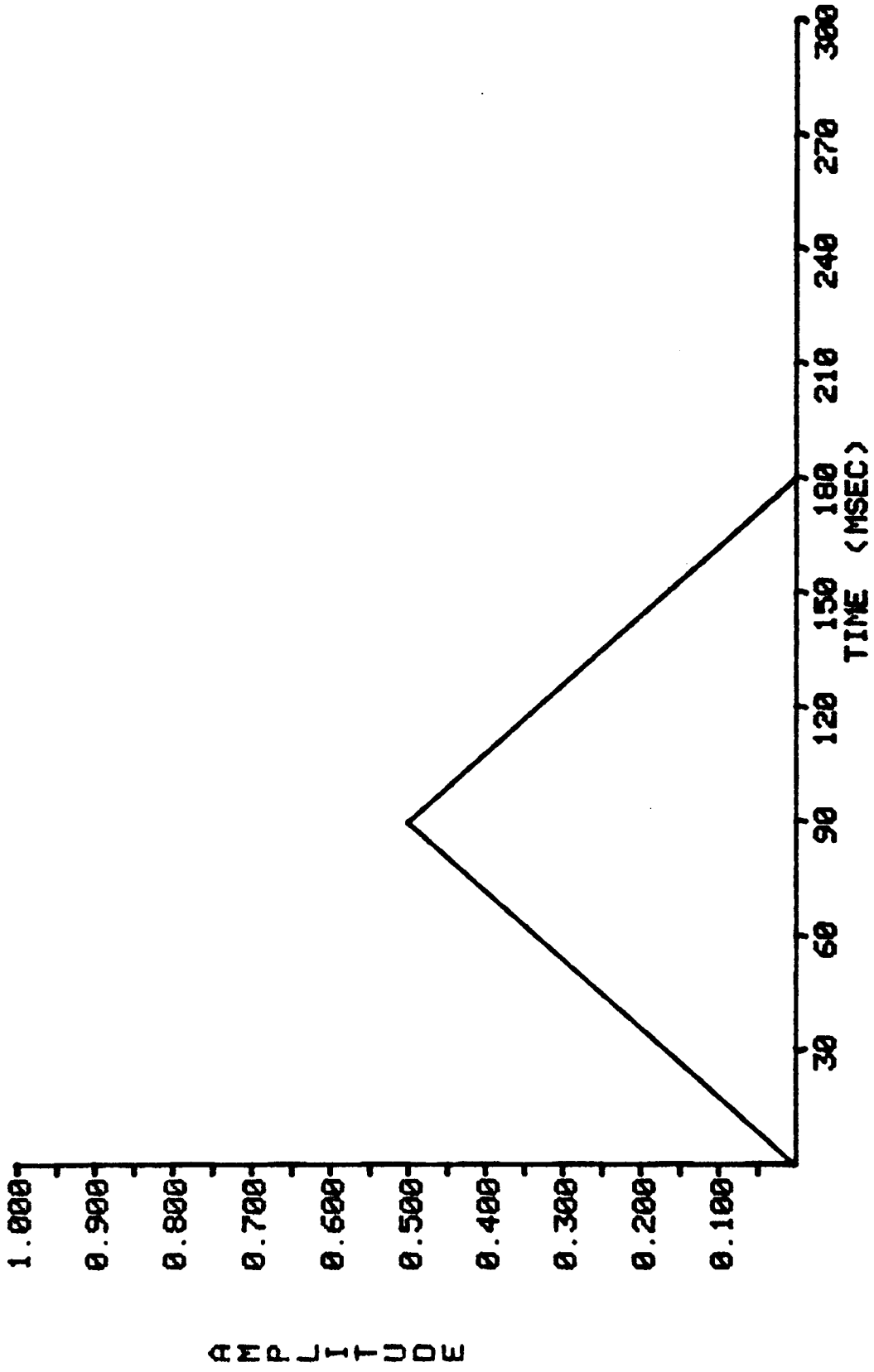

Figure 2: Amplitude Display

[Insert Figure 3 here: Derivative Display.]

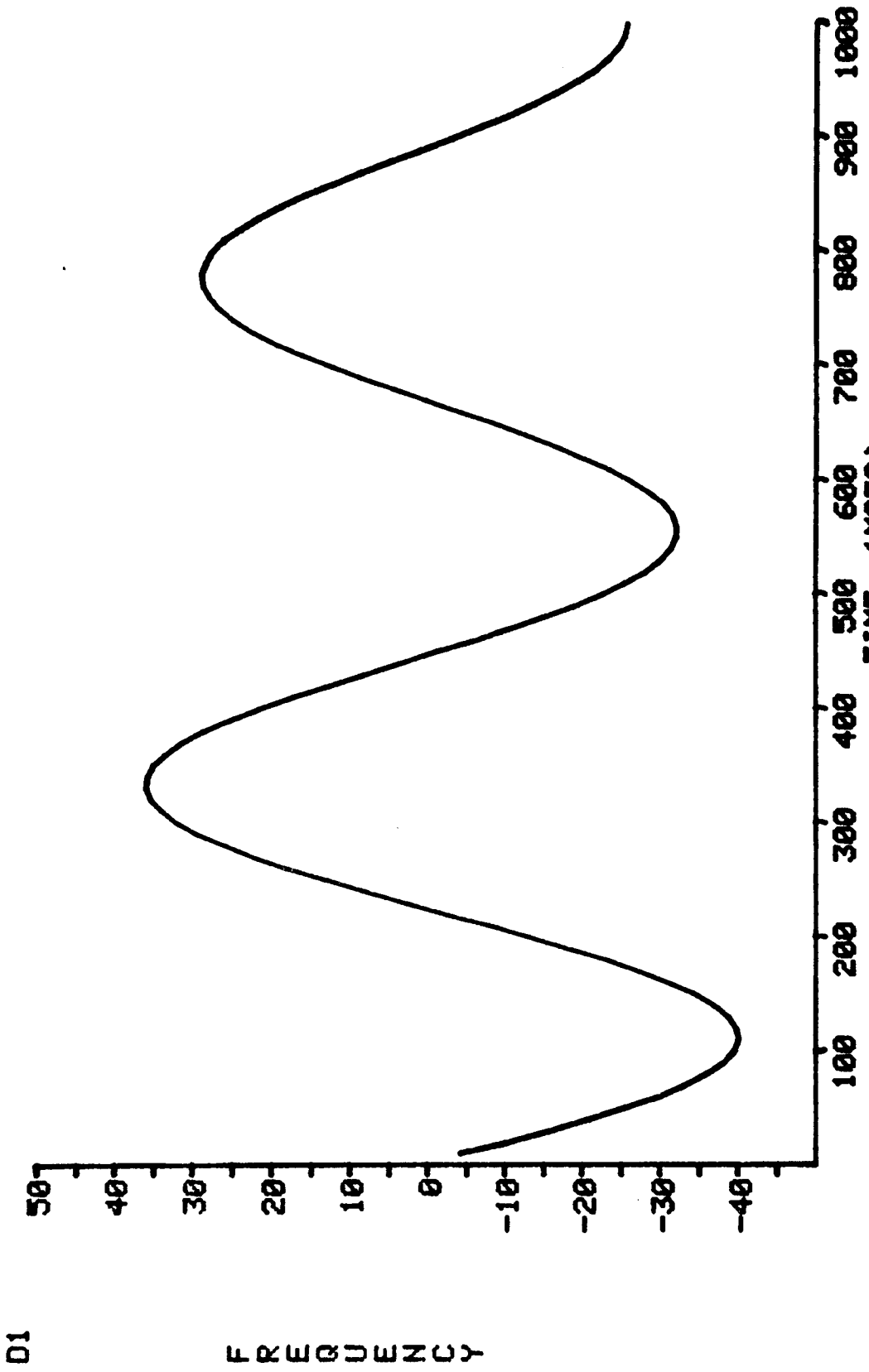

Figure 4. Derivative Display (first derivative on frequency)

[Insert Figure 2 here: Amplitude Display.]

[Insert Figure 4 here: Harmonic Motion Display.]

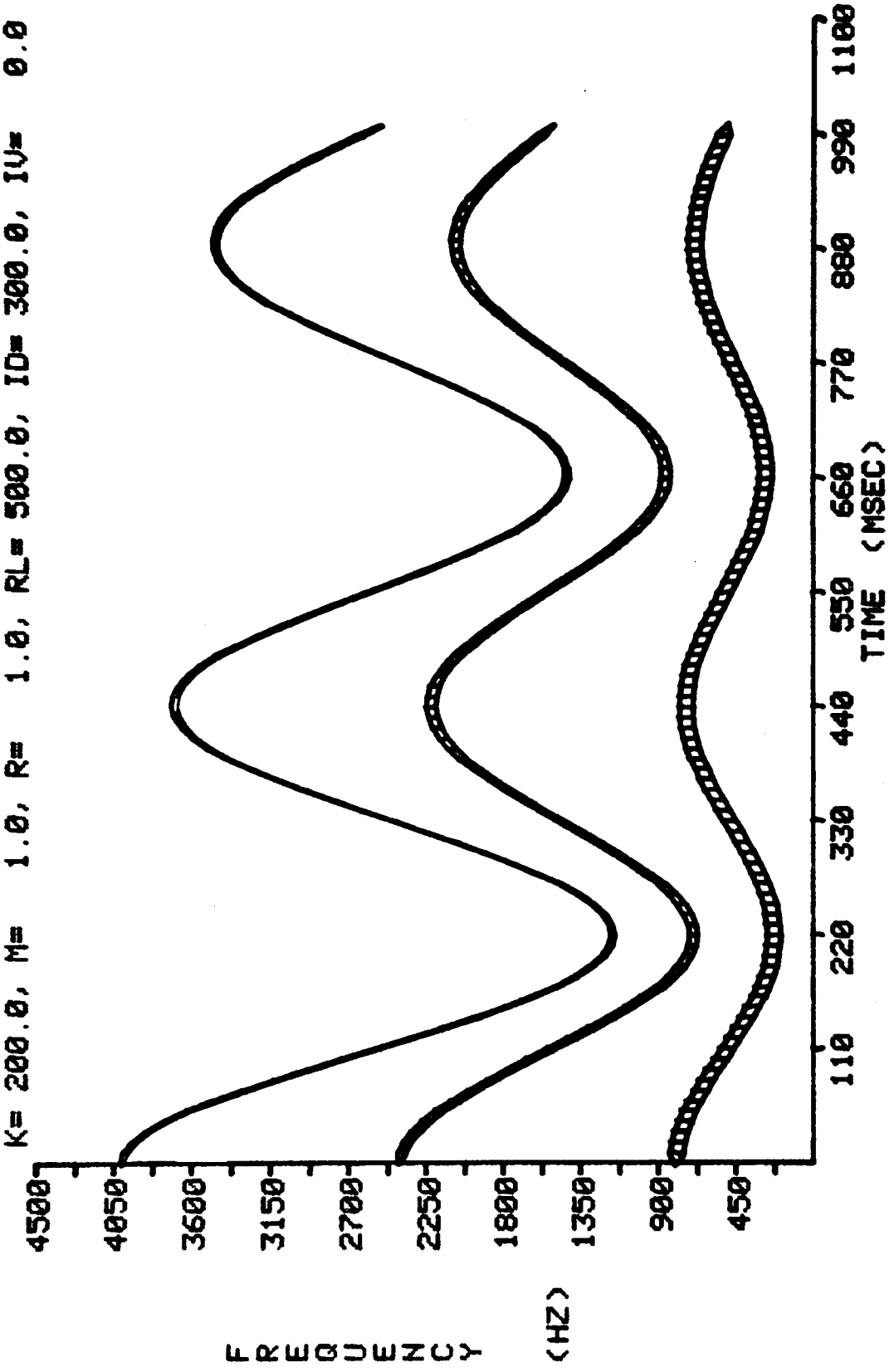

Figure 3: Harmonic Motion Display

VI. SYSTEM TRANSPORTABILITY NOTES

SWS, the Haskins Laboratories Sinewave Synthesizer, is implemented on a DEC VAX-11/780, using the VAX/VMS operating system, V3.0. The usual operations terminal is a Tektronix, which allows storage mode graphics. The resolution of its screen, for display purposes, is 1024 (x) by 780 (y). SWS is written in DEC VAX Fortran. SWS can be made to run on other computer systems, if a variety of transportability limitations are considered and properly addressed.

Limitations on transportability fall into the following categories:

VAX FORTRAN EXTENDED FEATURES.
SWS makes use of certain extended features of Vax FORTRAN, the most notable of these are:

PARAMETER statement - defines a constant.

Variable repetition factors in format field descriptors.
E.g.: <n>F7.0 .

OPEN and CLOSE file statements (these might have to be changed on your system).

DO-WHILE; DO-ENDO; and IF-THEN-ELSE structures.

INCLUDE statement - inserts an ASCII text file, including executable Fortran statements, into the program at compile time. If you don't have an INCLUDE statement, use your editor to insert the INCLUDE files at the appropriate places. These files are those with the .INC extensions.

GRAPHICS CALLS - SWS uses Fortran-callable graphics routines, which in turn call lower-level Macro assembler Tektronix primitives. These routines reside in a library called TEKGR.OLB. When SWS is sent to outside users this library is not supplied; however, documentation for the library will be included. The routines in this library are relatively standard graphics routines and they can be replaced by similar routines that are used on your own system.

WENDY - SWS invokes WENDY, the Waveform Editor and Display program, as a sub-process. WENDY is not supplied with SWS. All references to it should be removed from the code.

AUDIO OUTPUT - audio output is handled by issuing a low-level output call (QIO) to the Haskins Audio Output System, and is not directly handled by SWS. You should substitute routines appropriate to your installation.

FILES:
SWS makes use of a number of specific files. These include:

SY:SWSSWO.TMP;1. This is a contiguous binary file created on the user's area when output PCM data is generated. This file is deleted when the user exits the program.

SYS$HELP:SWS.AID. This is an ASCII text file, used to provide general information about SWS, when using the "?" command.

SYS$HELP:SWS.HLB. This is the text "help" library used by the routine SWSHELP.

TOOLS:SWS.WEN. This is an ASCII WENDY command file.

VII. SWS COMMAND SUMMARY

A SINEWAVE DATA INPUT BY ART (graphics entry).
A<cr>

AV AVERAGE FREQUENCY AND AMPLITUDE.
AV [start time (D=0.)],[end time (D=END)]<cr>

D DISPLAY SINEWAVE DATA ON THE SCREEN.
D [start time (D=0.)],[end time (D=END)],[sw # (D=ALL)]<cr>

DA DISPLAY AMPLITUDE.
DA [start time (D=0.)],[end time (D=END)],[sw # (D=1)]<cr>

DD DERIVATIVE DISPLAY.
DD [f or a],[d #],[sw #],[scale],[scmin],[scmax]<cr>

EA EDIT: ADD.
EA <mode: f,a or t>,[start time (D=0)],[end time (D=END)],
[sw # (D=ALL)],[additive factor]<cr>

ED EDIT: DELETE SINE WAVE DATA.
ED [start time],[end time],[sw #],[0=normal,1=compress]<cr>

EE EDIT: EDIT TIME AND SINEWAVE DATA (interactive entry).
EE [start time (D=0.)],[sw # (D=ALL)]<cr>

EH EDIT: HARMONIC CREATION.
EH [mode: m or a,][start time (D=0)],[end time (D=END)],
[sw # (D=ALL)],[val. of multiplier or adder],[amp.mult.(D=1)]<cr>

EI EDIT: INTERPOLATE.
EI <mode: f,a or t>,[start time (D=0)],[end time (D=END)],
[sw # (D=ALL)],[val1],[val2]<cr>

EL EDIT: LIST SINEWAVE DATA.
EL [start time (D=0.)],[end time (D=END)],[sw # (D=ALL)]<cr>

EM EDIT: MULTIPLY.
EM <mode: f,a or t>,[start time (D=0)],[end time (D=END)],
[sw # (D=ALL)],[multiplier]<cr>

EO EDIT: OVERLAY SINEWAVE DATA FROM A DISK FILE.
EO <filename (def. extension = .SWI) ><cr>

EQ EDIT: SQUASH FREQUENCY VALUES.
EQ [start time (D=0)],[end time (D=END)],[sw # (D=ALL)],
[frequency constant],[departure multiplier]<cr>

ER EDIT: READ SINEWAVE DATA FROM A DISK FILE.
ER <FILENAME (def. extension = .SWI) ><cr>

ES EDIT: SHIFT SINEWAVE.
ES [# slices to shift],[sw # (D=1)]<cr>

G GENERATE PCM DATA: G<cr>

H HELP: H <command code><cr>

HC HARDCOPY OF TEKTRONIX SCREEN.
HC<cr>

HO HARMONIC OSCILLATOR.
HO [f or a,]<time increment (d=10.)>,<duration (d=1000.)>,
<k>,<m>,<r>,<rl>,<id>,<iv>,<fv>,<# harm. (D=3rd,5th)><cr>

I INPUT SINEWAVE DATA BY PROMPT (interactive entry).
I<cr>

L LIST SINEWAVE DATA.
L [start time (D=0.)],[end time (D=END)],[sw # (D=ALL)]<cr>

M MENU OF SWS COMMANDS: M<cr>

P PLAY PCM DATA.
P [filename,][# repetitions],[isi in sec.]<cr>

PC PLAY COMPARISON.
PC <filename>,[# repetitions],[isi in sec.]<cr>

RI READ ILS DATA: RI<cr>

RS READ SINEWAVE DATA FROM A DISK FILE, DISPLAY AND GENERATE.
RS <filename (def. extension = .SWI) ><cr>

SET SET FLAGS AND VARIABLES.
SET <flag>=<value><cr>

SHO SHOW FLAGS AND VARIABLES.
SHO <choice><cr>

SP SAVE PCM DATA IN A DISK FILE.
SP <filename (def. extension = .SWO) ><cr>

SS SAVE SINEWAVE DATA IN A DISK FILE.
SS <filename (default extension = .SWI) ><cr>

SY SYN DATA CREATION.
SY <filename (d.ext.=.SER)>,[amp.mult.(D=1)],[% amp range (D=20)],
[db change (D=1)],[f0lo (D=100)],[f0hi (D=120)],[f0chg (D=1)]<cr>

W WENDY: W<cr>

X EXIT : X<cr>

? INFORMATION CONCERNING SWS: ?<cr>

```
MENU of SWS commands:

INP    INPut data by prompt              :   EDITOR Commands:
ART    input data by ART                 :    EL    List sinewave data
RSW    Read SineWave data                :    ER    Read sinewave data
RFB    Read FBA data                     :    EO    Overlay from file
LIST   LIST    sinewave data             :    EI    Insert  from file
DIS    DISplay sinewave data             :    ED    Delete sinewave data
AMP    AMPlitude display                 :    EM    Multiply
WAVE   WAVeform  display                 :    EA    Add
VAD    Velocity/Acc. Display             :    ET    inTerpolate
GEN    GENerate PCM data                 :    EN    Numerical value set
PLAY   PLAY PCM data                     :    EH    Harmonic creation
COM    play COMparison                   :    EE    Edit time & sinewave data
SPC    Save PCM data                     :    ES    Shift  sinewave
SSW    Save SineWave data                :    EQ    sQuash sinewave frequency
SSE    Save SErial   SYN data            :    EC    Copy time slice values
SPR    Save PaRallel SYN data            :    EF    Flip time slice values
OSC    harmonic OSCillator               :
AVE    AVErage frequency, amplitude      :   MISC. Commands:
                                         :    VAX   VAX DCL command
SET    SET  variables, flags             :    MACRO MACRO definition
SHOW   SHOW variables, flags             :    SUM   SUMmary of SWS
                                         :    MENU  MENU of  commands
RES    REServe a channel                 :    HELP  HELP for commands
REL    RELease a channel                 :    EXIT  EXIT SWS

SWS>
```