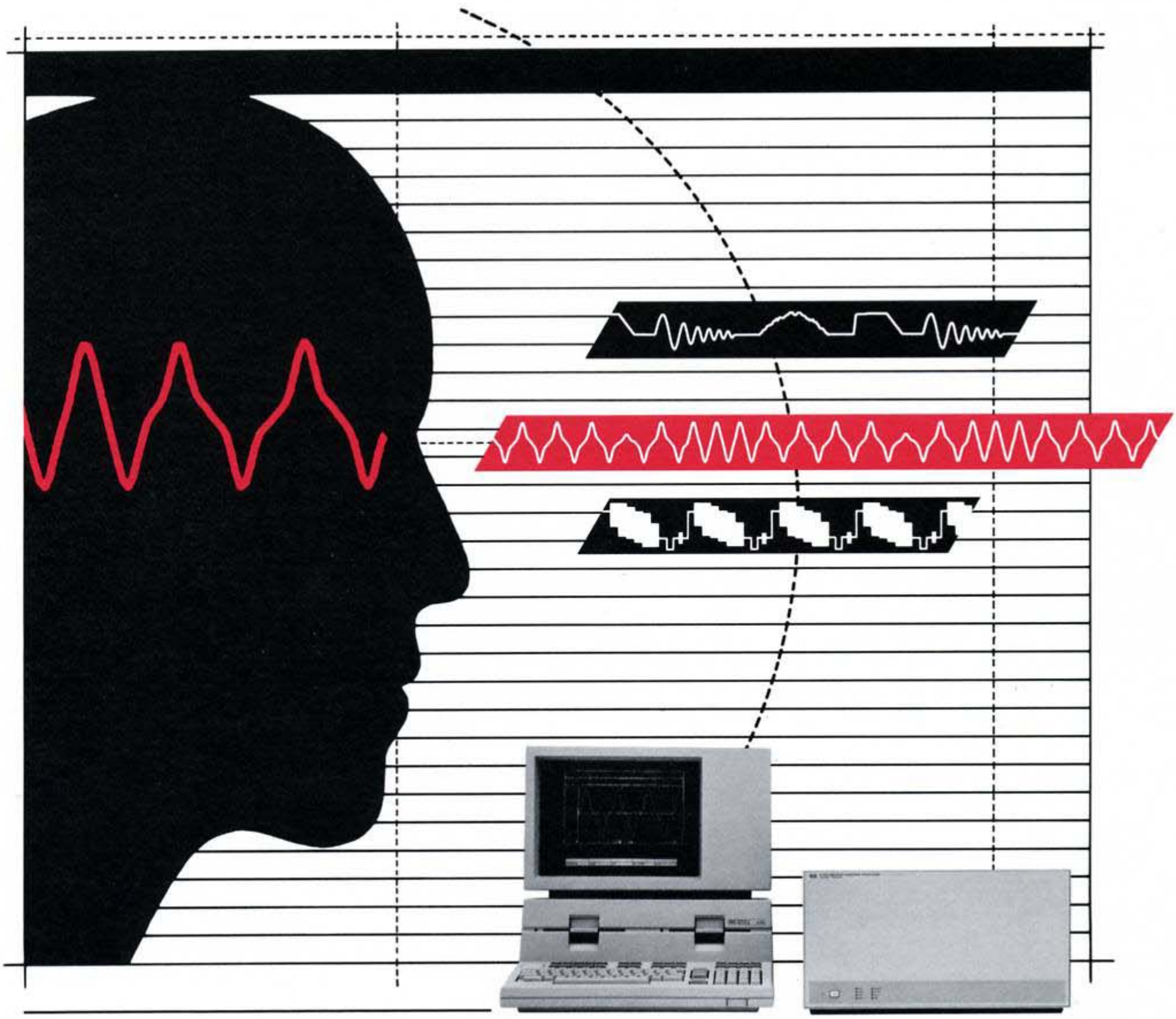
APPLICATION NOTE 314-2

Synthesizing Magnetic Disc Read and Servo Signals with the HP 8770S Arbitrary Waveform Synthesizer System



HEWLETT PACKARD

Synthesizing Magnetic Disc Read and Servo Signals with the HP 8770S Arbitrary Waveform Synthesizer System

This application note discusses how the HP 8770S can be used to provide magnetic disc signals to test read and servo circuitry. Examples of typical test signal developments will be covered, as well as how to add precise distortions and errors for simulating actual operating waveforms. General familiarity with the HP 8770S is assumed. If you have not already been introduced to the system, reading the data sheet and product note will be helpful (available at no charge from your local HP sales and service office; ask for HP literature numbers 5954-6355 and 5954-6360).

HP 8770S: A Powerful Magnetic Disc Test Signal Source

With the increasing performance of today's disc memories, it becomes essential to be able to accurately simulate operating signals to test the performance of disc circuitry. The HP 8770S is a versatile system for providing test signals to test magnetic disc drives at either the board or circuit level. The HP 8770S can synthesize signals with frequency components from dc to 50 MHz for complete simulation of high data rate signals. The HP 8770S can also generate signals with precise amounts of distortion and errors to help find the performance margin of disc drive circuitry.

The HP 8770S consists of the HP 8770A Arbitrary Waveform Synthesizer, the HP 11775A Waveform Generation Software, and an HP 9816A or 9836A technical computer. The HP 8770S synthesizes waveforms through digital synthesis: a user designs the waveform with the software, downloads the waveform data to the synthesizer, and the synthesizer then generates the waveform through a digital-to-analog converter.

The HP 8770S has the performance necessary to generate magnetic disc test signals. The HP 8770S uses a 12-bit digital-to-analog converter for generating high resolution test signals that help find a system's marginal condition. The synthesizer's large 128K internal waveform memory means complete disc sectors and other long period waveforms can be simulated with one signal source. High stability and repeatablility mean the HP 8770S can provide calibration signals to other pieces of test equipment in remote locations, maintaining high correlation.

REPRESENTATION OF MAGNETIC DISC READ SIGNALS

Because of the varied modulation formats in use today, there are many different types of magnetic disc read signals. For this application note, MFM (Modified Frequency Modulation) disc signals are the waveforms choosen for the examples. Normally, MFM disc read signals are modeled as the summation of Lorentzian pulses with each pulse occurring whenever there is a transition in the disc's magnetic state. The closer these transitions, the more the disc signal may be represented with a simpler model of summed sine waves harmonically and phase related. This note uses the sine wave model for disc read signal examples.

Common MFM test signals are the 1F, 2F and 110 waveforms (**Figure 1**). The 1F signal represents the lowest magnetic reversal rate and therefore the lowest frequency signal. The 2F signal represents the highest magnetic reversal rate and thus the highest frequency signal. The 110 signal represents an intermediate magnetic reversal rate between the 1F and 2F signals.

Figure 1. MFM test signals created by summation of sine waves. The HP 8770S can generate signals with frequency components up to 50 MHz.

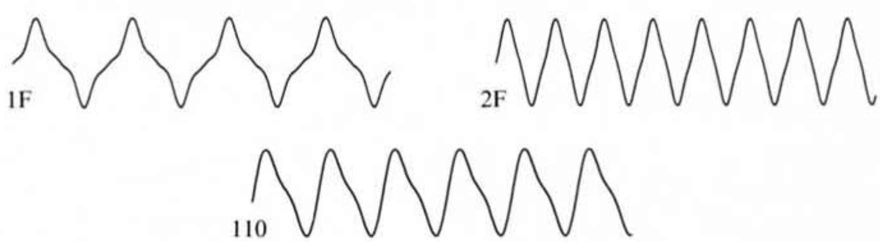
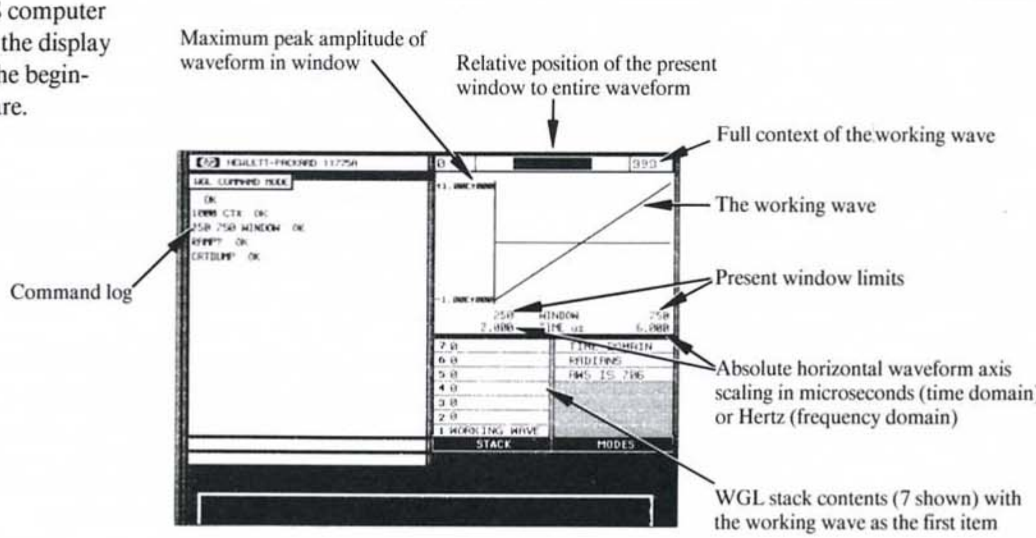


Figure 2. Typical HP 8770S computer screen. The working wave in the display is a RAMP, which is one of the beginning waveforms of the software.



CREATION OF READ TEST SIGNALS

Test signals are easy to create with the HP 11775A Waveform Generation Software. This software features the Waveform Generation Language (WGL), which is a powerful tool for creating waveforms in either the time or the frequency domain. **Figure 2** shows the computer display of the HP 8770S software. The left side of the computer screen contains a running log of executed commands, the lower right corner displays items on the stack, and the upper right corner displays the waveform. In **Figure 2**, the displayed waveform is a RAMP, one of the language's "seed" waveforms used in the following examples. The software works with reverse Polish notation and uses the stack in its computations. For example, to find the sum of 2 plus 2, the order of operation would be

2 <ENTER> 2 +

This places 2 and 2 on the stack and then adds them together, returning the sum as the first item on the stack.

Because the HP 8770S creates an analog signal from discrete waveform samples, three WGL terms describing these samples have evolved. The first term is "working wave," which refers to the linear array that holds the waveform samples. The number of discrete samples that make up the working wave is referred to as the "waveform context," or just "context." A waveform context can be set between 10 and 16384 (16k) discrete elements in the standard system configuration. "Window" refers to the discrete elements in the working wave that are presently being operated on. A window is user-selected and consists of a minimum of 3 elements and a maximum number of elements equal to the full context of the working wave. By using the **WINDOW** command, a waveform can be designed a section at a time.

Appendix 1 lists the WGL software commands used in this application note.

1F Signal Creation

The first example shows how to create a 1F signal as a sine wave added with its 3rd and 5th harmonics. The 3rd harmonic is 1/3 the fundamental's amplitude and 180 degrees out-of-phase. The 5th harmonic is 1/9 the fundamental's amplitude and in-phase. More harmonics, different amplitudes and phases can be used if desired. Other disc signal models can be used as the basis for creating test waveforms, as well.

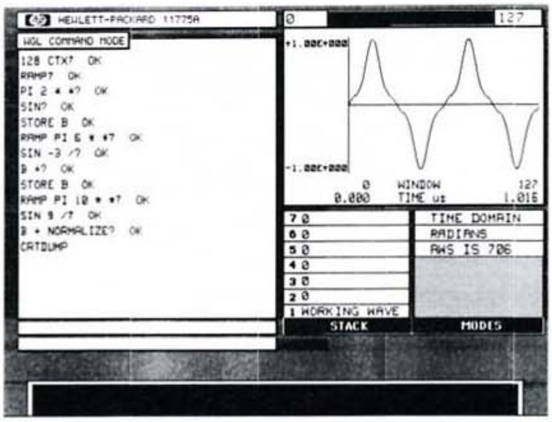
The data for the 1F signal is first created with the HP 8770S software. The 1F signal will be approximately 1.95 MHz, typical of the transfer rate of present disc systems. Because the HP 8770S has an 8 ns sampling rate, 64 sample points for each cycle produces the 1F signal of 1.95 MHz (512 ns period). Using a different number of points changes the period and thus the frequency of the signal.

Two cycles of the 1F signal are created in this example to better show the creation process. The following WGL software commands generate the 1F signal.

128 CTX?	Set the number of waveform points to 128.
RAMP?	Start with a linearly increasing series of waveform samples.
PI 2 * *?	Have the samples go from −2 pi to + 2 pi to set the necessary phase information for 2 cycles of a sine wave.
SIN?	Take the sin of the display information to give the samples for 2 sine wave cycles.
STORE B	Save the fundamental for later use.
	Next, generate the samples for the other harmonics and add them to the fundamental.
RAMP PI 6 * *?	This gives the phase information for the 6 cycles of the 3rd harmonic.
SIN -3 /?	Dividing by −3 sets both the amplitude and phase for the 3rd harmonic relative to the fundamental.
B +?	Recall the previously stored fundamental and add it to the 3rd harmonic.
STORE B	Save the summed signal.
RAMP PI 10 * *?	Phase information for 10 cycles of the 5th harmonic.
SIN 9 /?	Take the sin and divide by 9 to set the amplitude level relative to the fundamental.
B + NORMALIZE?	Add the 1st and 3rd harmonics to complete the 1F signal, and then normalize it to a maximum absolute amplitude of 1.
STORE B?	Save the 1F signal.

Figure 3 shows the computer display after creating the 1F signal. The executed commands are listed on the left side of the display.

Figure 3. HP 8770S computer screen shows software commands used to create 1.95 MHz 1F signal.

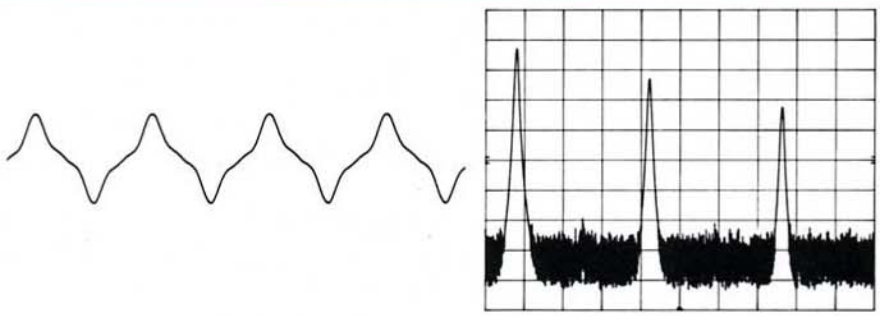


To generate the 1F signal, the waveform data is sent to the HP 8770A synthesizer with the following command string:

DOWNLOAD GO	The software will prompt for a name and a scaling factor before saving the waveform data. Press the **RETURN** key to return the default values. The **GO** command starts the generation of the signal.

Figure 4 shows an oscilloscope and spectrum analyzer display of the generated 1F signal. The oscilloscope's display matches the created signal on the computer display, and the spectrum analyzer shows that the signal does consist of 3 sine waves with the desired amplitudes. For small signals, the synthesizer has a 110 dB attenuator (10 dB steps) that can set the full scale amplitude range from 2 V p–p to 6.3 μV p–p.

Figure 4. Oscilloscope and spectrum analyzer display of HP 8770S synthesized 1F signal.



2F and 110 Signal Creation

2F and 110 test signals are created with the same sine wave model as the 1F signal. The 2F signal is simulated with a sine wave of 3.9 MHz added with its 3rd harmonic at 1/9 the amplitude and 180 degrees out of phase:

128 CTX?	Set the context for 4 cycles of the fundamental.
RAMP PI 4 * *?	Phase information for the 2F fundamental.
SIN STORE C?	Save the fundamental.
RAMP PI 12 * *?	Phase information for the 3rd harmonic.
SIN -9/ C +?	Set the amplitude and phase relationship and then add it to the stored fundamental.

STORE C	Save the completed signal.
DOWNLOAD GO	Generate the 2F signal.

The 110 disc signal is created as a sine wave of 2.93 MHz added with its 2nd harmonic at 1/3 the amplitude:

128 CTX?	Set the context for 3 cycles of the fundamental.
RAMP PI 3 * *?	Phase information for the fundamental.
SIN STORE D?	Save the fundamental.
RAMP PI 6 * *?	Phase information for the 2nd harmonic.
SIN 3/ D+?	Set the amplitude relation and add it to the fundamental.
STORE D?	Save the completed signal.
DOWNLOAD GO	Generate the 110 signal.

SETTING SIGNAL FREQUENCY

The **FREQ** command in the HP 8770S software gives the necessary information needed to generate a sine wave at a given frequency, simplifying the creation of head read test signals. The parameters that the **FREQ** command act on are the maximum number of points to be used, and the desired frequency of the test signal. For example, the following command

1000 4E6 FREQ

instructs the software to return the parameters needed to create a 4 MHz sine wave using fewer than 1000 points. The command returns the context length, the number of cycles needed to create the desired sine wave, and the frequency error in Hertz. The context information is the first item on the stack, the number of sine wave cycles in the context is second and the frequency error is third. The following commands create the 4 MHz sine wave using the **FREQ** command.

1000 4E6 FREQ	Get the context and number of cycles.
CTX RAMP?	Set the context to the first value returned with **FREQ** and place a **RAMP** there.
*** PI *?**	This sets the phase information necessary based on the number of cycles. The first * multiplies the **RAMP** by the value for the number of sine wave cycles returned by **FREQ**.
SIN ?	Take the sin of the phase information to create the 4 MHz signal.

This signal can be used as the fundamental of a read signal. Creation of harmonics is simple by multiplying the number of sine wave cycles by the number of the desired harmonic. For example, to get the 3rd harmonic, the number of cycles returned with the **FREQ** command is multiplied by 3.

ADDING ERRORS

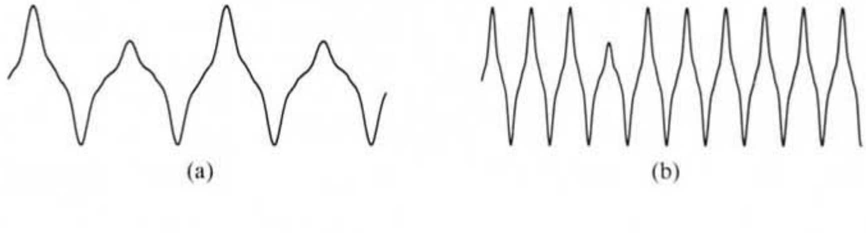
The major contribution of the HP 8770S for disc test signal generation is synthesizing test signals with precise amounts of errors. This section gives examples of several different types of read signal errors that the HP 8770S can generate with set error amplitudes and rates.

Missing Bits

The creation of a missing bit is easy because the HP 8770S synthesizes waveforms point-by-point and thus error conditions can also be set point-by-point. To create a missing bit on the previously stored 1F signal, the following example shows how to "zoom in" on one transition of the 1F signal and lower the amplitude of that section to the desired missing bit level.

128 CTX?	Set the waveform display context the same as for the stored 1F signal.
B?	Recall the stored 1F signal.
MARKER	Rotate the computer knob controlling the marker to find the location of the first 0 crossing of the desired transition. Place the location on the stack by pressing the **ENTER** key (location 64 in this example).
MARKER	Locate the second 0 crossing of the desired transition (location 96 in this example).
WINDOW?	Zoom in on the desired transition.
0.5 *?	Reduce the transition's amplitude 50 %.
FULL?	Look at the entire waveform and verify that the missing bit occurs in the desired location.
STORE E	Store the missing bit waveform for later use.
DOWNLOAD GO	Send the data to the synthesizer for subsequent generation.

In this example, the missing bit occurs once out of every 4 transitions, generally a much higher rate than would normally occur in an actual system (**Figure 5a**). Realistic error rates can be obtained in two ways. The first way is to simply start with more transitions before creating the missing bit (**Figure 5b**). The second way uses the synthesizer's sequencer to set the error rate. Using the sequencer reduces the amount of memory needed to set the error rate.



(a) (b)

Figure 5. One transition of 1F signal is reduced to simulate a missing bit (a). Realistic error rates can be attained using either more memory (b), or using the HP 8770S synthesizer's memory sequencer.

The sequencer can program any section of memory to repeat many times. These memory sections (packets) can follow in any order and be repeated different numbers of times. The missing bit rate is set by first downloading the waveform data that does not contain the the missing bit to the synthesizer, and then doing the same for the waveform data that does contain the missing bit. After that, the repetition rate of each packet is specified by the user to determine the desired error rate.

The following example shows how to set the missing bit rate to once every 56 signal transitions. The memory packet with the missing bit is expanded to 512 memory elements because the synthesizer requires a packet duration of at least 344 elements before it can move on to the next packet.

The first step is to download all the data for the 1F signals with and without the missing bit into the synthesizer:

128 CTX?	Set the correct context for recalling the stored 1F signal.
B?	Recall the 1F signal.
DOWNLOAD	Download the 1F signal to the synthesizer, giving it the name "**ONEF**."
512 CTX?	Set the context for the missing bit packet.
128 CLONE?	Repeat the 1F signal throughout the expanded display context, creating 16 transitions.
0 127 WINDOW?	Set the display window to recall the missing bit signal.
E?	Recall the missing bit signal.
FULL?	Show the entire signal.
DOWNLOAD	Download the missing bit signal to the synthesizer, giving this the name "**MISS**."

Now that the waveform data is stored in the synthesizer, the sequencer information is sent to start generation of the test signal with the specified error rate. The **NEWSEQ** command initializes the sequencer memory in preparation for new sequencer information. The new sequencer information is created with the **PACKET** command. **PACKET** asks for the packet name in the synthesizer, the desired number of repetitions, and the packet advance condition. For this example, the **AUTO** advance is used. Each packet will be subsequently generated in the order that the packet information is entered into the synthesizer. The following commands set the error rate and start generation of the missing bit signal:

NEWSEQ	Initialize the synthesizer's sequencer memory.
PACKET	Create the information for the packet without the missing bit. When prompted for the name, enter "**ONEF**." Set the number of repetitions to 10 for 40 total signal transitions (4 transitions * 10 repetitions). Set for **AUTO** packet advance.

PACKET	Create the information for the packet with the missing bit. Enter the name "**MISS**," set 1 repetition, and **AUTO** advance
GO	Start the synthesizer generating the test signal with the packet information.

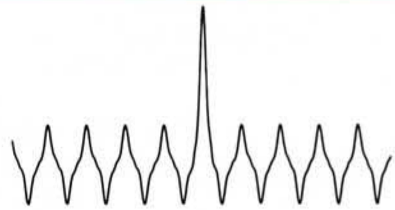


Figure 6. Synthesis of extra bit on the 1F signal.

Extra Bit

An extra bit error signal can be simulated in much the same way as the missing bit signal by increasing the amplitude of a single transition (**Figure 6**). The commands needed to generate the extra bit are very close to the ones for the missing bit:

128 CTX B?	Recall the 1F signal.
64 96 WI?	"Zoom-in" on one transition.
4 *?	Increase the amplitude for the error (400% in this example).
FULL?	View the entire signal.
DOWNLOAD GO	Send the waveform to the synthesizer. Use **AUTO** scaling in the **DOWNLOAD**.

The maximum output signal amplitude from the synthesizer is 1 V p–p. When the extra bit signal is downloaded, the software will scale the data automatically to the 1 V p–p output level.

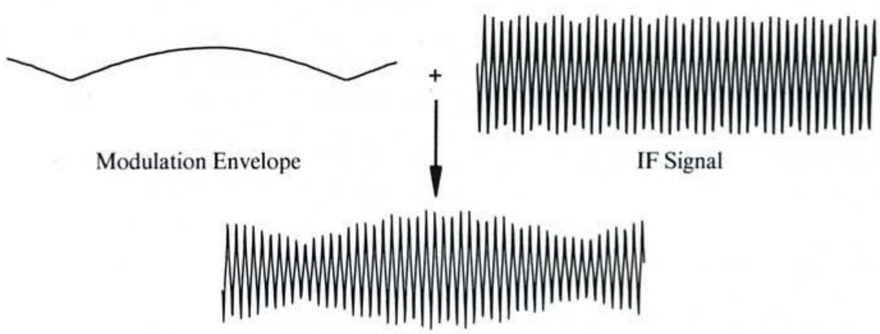


Figure 7. The HP 8770S can generate test signals with AM to simulate varying disc flying heights.

External AM

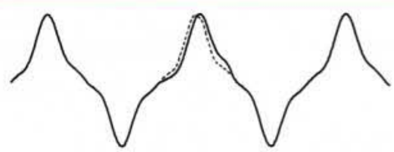
The HP 8770S can generate test signals where the entire waveform has amplitude modulation to simulate the effect of varying flying heights (**Figure 7**). The following example shows how AM can be imposed on the 1F signal with a 65.536 μs AM period (15.26 kHz).

8192 CTX	Set the context for 65.536 μs.

RAMP PI *?	Create the modulating waveform. This will be a raised sine wave. This is the phase information for one cycle of the sine wave.
SIN 2+?	Take the sine of the phase information and raise it by 2 for an AM depth of 50%.
STORE A	Save the AM waveform.
0 127 WI B?	Recall the 1F signal.
FULL 128 CLONE?	Repeat the 1F signal through the entire 65.536 μs period.
A *?	Modulate the 1F signal with the stored AM signal.
DOWNLOAD GO	Send the signal to the synthesizer for generation.

The amount of peak-to-peak AM can be varied to find the clip points of the disc read circuitry. If longer period AM signals are required, more memory can be used (larger context), or the synthesizer's memory sequencer can repeat different parts of memory to generate the longer signal. With either method, care must be taken that the waveform data does not have any discontinuities that will cause glitches in the final signal.

Figure 8. The HP 8770S can simulate peak shift on test signals.



Peak Shift

Peak shifts are signal peaks occurring before or after they should because magnetic transitions are so close together that there is intersymbol interference. The HP 8770S can generate test signals that simulate varying amounts of peak shift. These peak-shifted signals can be used to find where a bit may fall out of the data window, and to determine how sensitive the clock recovery and phase-locked loop circuitry is to the distortion (**Figure 8**). The following example shows how to simulate peak shift on one signal transition:

128 CTX B?	Recall the 1F signal.
64 96 WI?	"Zoom-in" on one transition.
STORE E	Save for modification.
RAMP EXP?	Create an exponential signal.
E * NORMALIZE?	Shape the stored transition by the exponential and normalize the resultant amplitude to the rest of the signal.
FULL?	Observe the signal.
DOWNLOAD GO	Generate the signal.

The HP 8770S has a time resolution of 8 ns for setting peak shift. Smaller amounts of peak shift are possible by using the external sampling clock input of the synthesizer. The external clock can be from 10 MHz to 130 MHz. A synthesizer with its output frequency modulated can be used for the external clock input. A clock signal with FM causes the synthesizer's sample rate, and thus the signal period, to vary over time, generating peak shifts on the entire signal.

Figure 9. Noise can be added to test signals in varying amounts for sensitivity tests.

Adding Noise and Glitches

The noise capability of the HP 8770S can be used to determine how many data errors occur for a given noise level (**Figure 9**). The noise is based on a random number generator. The amplitude of this noise can be scaled to any level relative to the signal. The following example sets a signal-to-noise ratio of approximately 20 dB on the stored 1F signal:

128 CTX	Set the context for the stored 1F signal.
NOISE 10 /?	Set noise to approximately 20 dB below the 1F signal.
B+?	Add scaled noise to the 1F signal.
DOWNLOAD GO	Generate the noisy 1F signal.

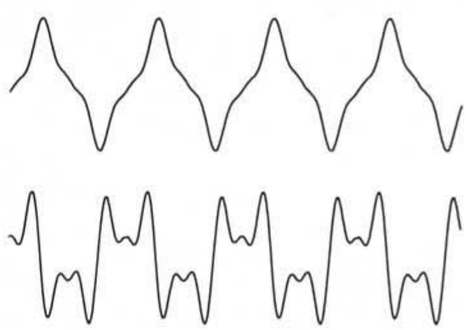
The HP 8770S will generate this signal repeatedly with the same noise level. In this example, the period of the noise is the same as 2 cycles of the 1F signal. The noise period can be increased relative to the 1F period by increasing the length of the 1F signal (larger context) before adding the noise.

Glitches can also be added with the **MARKCHANGE** command. With this command, the test signal can be changed point-by-point to add glitches of any level or duration to test the error rate as a function of the severity of glitches.

TESTING DIFFERENTIATOR CIRCUITS

After a read test signal has been created, it can also be used to test the differentiator circuits that follow the head amplifiers. The HP 8770S software has a **DIFF** command that differentiates a waveform in a given window (**Figure 10**). With this command, it is possible to separately test the head amplifiers and differentiating circuits, and determine their performance parameters directly at each stage.

Figure 10. The HP 8770S can differentiate signals for testing disc head detector circuits after the head amplifiers.



CREATING SERVO TEST SIGNALS

Important elements in proper disc operation are the servo circuits. These circuits are responsible for positioning the read head over the proper data track and keeping the head on track. The HP 8770S can synthesize servo test signals that simulate on- and off-track conditions, as well as a track seek. These signals can be used to calibrate servo writers, or to characterize the response of various parts of servo circuits either at the board level or system level.

Just as there are many different types of read signals, there are many different types of servo signals. For demonstration purposes, the following example is a dedicated quad di-bit servo signal. This example signal consists of one sync pulse and then 3 pulses representing 3 different tracks (**Figure 11**). The amplitudes of the 3 track pulses indicate the track position of the head. Relative amplitudes and timing of the pulses are randomly chosen for purposes of this example and, of course, can be changed for specific applications.

Figure 11. Example servo signal synthesized with the HP 8770S. Period is 2.048 μs, and duration of pulses is 256 ns.

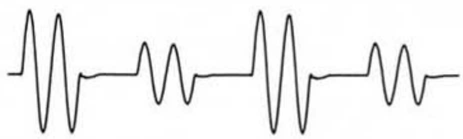


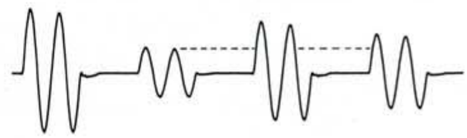
Figure 11 shows a servo signal with a 2048 ns period (256 memory elements) generated by the HP 8770S. The signal is created by using the **WINDOW** command to put 2 cycles of a sine wave in the selected positions. The following commands create the example servo signal.

256 CTX	Set the context for the desired period.
0 LOAD	Clear the display.
1 32 WI?	Set the timing for the sync pulse, which is 256 ns duration.
RAMP PI 2 * * SIN?	Create 2 cycles of a sine wave.
STORE B	Save the sine wave cycles for later use.
65 96 WI?	Move to the first track position.
B 2/?	Insert 2 sine wave cycles at 1/2 amplitude.
129 160 WI?	Move to second track.
B?	Insert 2 sine wave cycles.
193 224 WI?	Move to last track position.
B 2/?	Insert 2 sine wave cycles again at 1/2 amplitude.
FULL?	View the entire servo signal.
DOWNLOAD GO	Generate the signal.

This signal can be the on-track servo signal for track #2. Adding noise to this signal in varying amounts can be used to test the noise sensitivity of the detection circuitry. The on-track servo signal can also be modified to simulate an off-track signal. This off-track signal can be varied to test the sensitivity and linearity of the servo error detection circuitry. To create the off-track signal, the track pulses only need to be varied in amplitude during the creation process (**Figure 12**).

Another servo test application is using the HP 8770A's memory sequencer to simulate a track seek. The track seek signal can be generated by first storing in the synthesizer's memory different servo signals that represent the head moving from one track to another. The sequencer can then be programmed to repeat each waveform many times and in the desired order to simulate the seek signal.

Figure 12. Relative amplitudes of the track pulses are easy to change for simulating off-track conditions.



SAVING AND RETRIEVING WAVEFORMS

After creating a test signal, it is simple to save the test waveform data on a magnetic disc for later use. The **SAVEWAVE** command stores the waveform in the present display **WINDOW** to a mass storage device. A mass storage device can be specified, or the default device is used.

The **GET** command brings into the present **WINDOW** a previously stored waveform. An aid to retrieving waveforms is to put information about the length of the waveform in the name of the stored waveform. For example, a name for a 1F signal with a context of 128 might be **ONEF_128**. Knowing the length of the stored waveform facilitates setting the proper **WINDOW** length before the waveform is brought into the display with **GET**.

APPLICATION COMMANDS AND PROGRAMS

The HP 8770S software allows the definition of new, user-selectable programming commands. These new commands can be optimized so that one command will do all the work and bookkeeping necessary for creating test signals. Existing software commands or other user-defined commands can be combined to form new commands.

To define a new command, the **DEFINE** capability of the software is used. Commands typed between **DEFINE** and **END** are executed whenever the new command is used.

The following example creates a new command called **SINE**. **SINE** will generate a sine wave at a user-defined frequency. The **FREQ** command is used to find the proper amount of memory needed to generate the desired sine wave.

DEFINE SINE	This enters the **DEFINE** mode and sets the new command's name.
FREQ CTX	Call the **FREQ** command to find and set the proper memory length with the context number returned by **FREQ** (length returned as first item on the stack).
RAMP * PI * SIN	Create the necessary number of sine wave cycles based on the number of cycles returned by **FREQ**.

END Close the definition.

SINE will now automatically create a sine wave when called and passed the same parameters as **FREQ**:

< Maximum amount of memory to be used >< Frequency > SINE ?

An example application program for generating disc read signals is provided with the HP 11775A software. This program creates MFM read test signals based on the sine wave model, and also servo test signals.

To load this sample application program, follow the instructions below:

1. Put the HP 11775A Application Programs disc containing **"DISC_APP"** in the computer disc drive.

2. Type **"DISC_APP" $ GET**

3. Press the **EDIT** key (for the HP 9816A type **EDIT** and press the **RETURN** key).

4. Press the **EXIT** softkey to compile the **DISC_APP** program.

5. Type **DISC_APP** to run the program.

SUMMARY

This note shows a few practical examples of how the HP 8770S system can provide many signals that are useful in the development and testing of disc drive circuitry. Its 50 MHz bandwidth and 12-bit output means that the system can precisely generate the high speed signals with high resolution needed for effective disc read head and servo circuitry testing. The large memory enables simulating complicated signals, both ideal and with errors, for testing disc drive circuitry.

APPENDIX 1:
Glossary of WGL Commands Used in this Application Note

CLONE — Duplicates the specified number of waveform sample elements in the working wave as many times as necessary to fill the present window.

CTX — This command sets the length of the full working waveform.

DEFINE — Used to create a new command.

DOWNLOAD — Downloads the waveform information to the synthesizer. The **DOWNLOAD** command will prompt for a waveform name and a scaling factor. For the examples in this note, the **AUTO** scaling factor is used. The command will then automatically scale the waveform to an amplitude between −1V and +1V. Any name less than 6 characters is acceptable, with the first character a letter.

END — Used to signify the end of a new command definition.

EXP — Natural antilogarithm of e.

FULL — Selects all the waveform elements for manipulation.

GO — Starts the generation of waveforms by the synthesizer.

NEWSEQ — Initiates a sequencer program definition.

NORMALIZE — Scales the waveform elements in the present window between −1 and +1.

PI — This is the constant pi.

PACKET — Defines a waveform Packet for the Sequencer with name, number of repetitions and triggering information

RAMP — Fills all elements of the present window with linearly increasing values starting with −1. The incremental increase is 2/(# of elements in the **WINDOW**). For example, a **RAMP** generated in a window of 200 elements would have the following values: −1.00, −0.99 . . . 0.98, 0.99.

STORE A — This is the command that saves the top item on the stack (a number or waveform). Other memory locations available are B, C, D and E.

WINDOW — This command allows the user to "zoom in" on sections of the working waveform.

*** / + SIN** — These are the multiply, divide, add and sine math commands of the software.

? — The question mark updates the waveform display.

5954-6357 January 1986 Printed in U.S.A.