Measurement Guide and Programming Examples

N9068A Phase Noise Measurement Application

This manual provides documentation for the following instrument:

Agilent Technologies N9020A MXA Signal Analyzer
Agilent Technologies N9010A EXA Signal Analyzer

Agilent Technologies

Manufacturing Part Number: N9068-90005
Supersedes: N9068-90003

Printed in USA

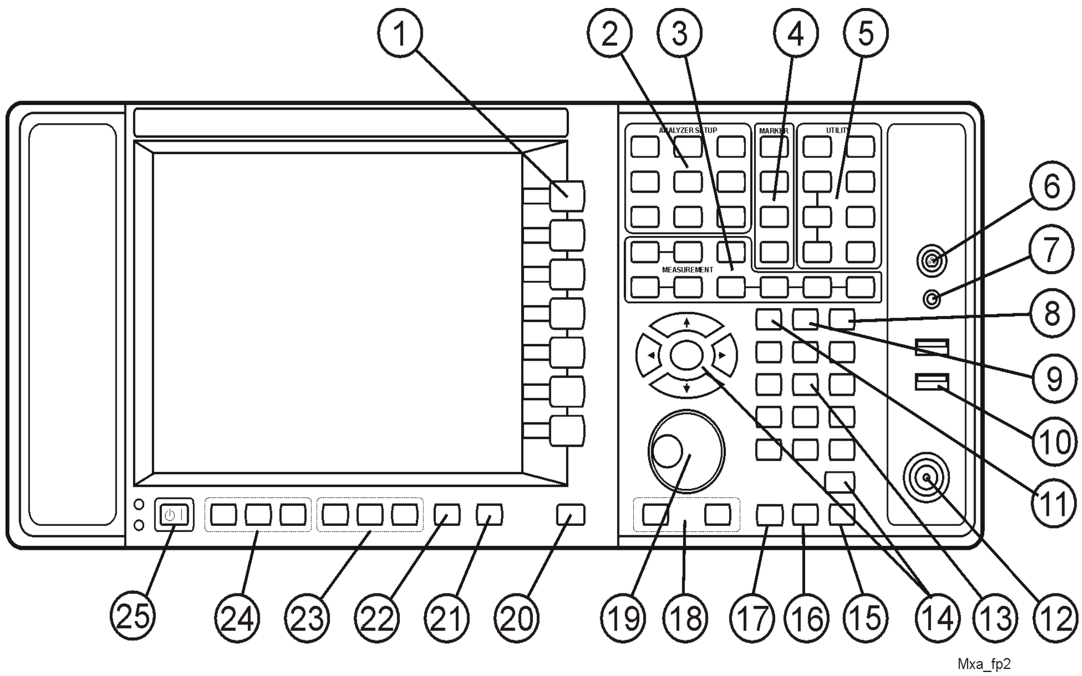
August 2007

Contents

1 Front and Rear Panel Features

Front Panel Features

Front-Panel Connectors and Keys



Item		Description
#	Name	
1	Menu Keys	Key labels appear to the left of the menu keys to identify the current function of each key. The displayed functions are dependent on the currently selected Mode and Measurement, and are directly related to the most recent key press.
2	Analyzer Setup Keys	These keys set the parameters used for making measurements in the current Mode and Measurement.
3	Measurement Keys	These keys select the Mode, and the Measurement within the mode. They also control the initiation and frequency of measurement.
4	Marker Keys	Markers are often available for a measurement, to measure a very specific point/segment of data within the range of the current measurement data.
5	Utility Keys	These keys control system-wide functionality like: • instrument configuration information and I/O setup, • printer setup and printing, • file management, save and recall, • instrument presets.
6	Probe Power	Supplies power for external high frequency probes and accessories.

Item		Description
#	Name	
7	Headphones Output	Headphones can be used to hear any available audio output.
8	Back Space Key	Press this key to delete the previous character when entering alphanumeric information. It also works as the Back key in Help and Explorer windows.
9	Delete Key	Press this key to delete files, or to perform other deletion tasks.
10	USB Connectors	Standard USB 2.0 ports, Type A. Connect to external peripherals such as a mouse, keyboard, DVD drive, or hard drive.
11	Local/Cancel/ (Esc) Key	If you are in remote operation, Local: • returns instrument control from remote back to local (the front panel). • turns the display on (if it was turned off for remote operation). • can be used to clear errors. (Press the key once to return to local control, and a second time to clear error message line.) If you have not already pressed the units or Enter key, Cancel exits the currently selected function without changing its value. Esc works the same as it does on a pc keyboard. It: • exits Windows dialogs • resets input overloads • clears errors • aborts printing • cancels operations.
12	RF Input	Connector for inputting an external signal. Make sure that the total power of all signals at the analyzer input does *not* exceed +30 dBm (1 watt).
13	Numeric Keypad	Enters a specific numeric value for the current function. Entries appear on the upper left of the display, in the measurement information area.
14	Enter and Arrow Keys	The Enter key terminates data entry when either no unit of measure is needed, or you want to use the default unit. The arrow keys: • Increment and decrement the value of the current measurement selection. • Navigate help topics. • Navigate, or make selections, within Windows dialogs. • Navigate within forms used for setting up measurements. • Navigate within tables. **NOTE** The arrow keys cannot be used to move a mouse pointer around on the display.
15	Menu/ (Alt) Key	Alt works the same as a pc keyboard. Use it to change control focus in Windows pull-down menus.
16	Ctrl Key	Ctrl works the same as a pc keyboard. Use it to navigate in Windows applications, or to select multiple items in lists.

Item #	Item Name	Description
17	Select / Space Key	Select is also the Space key and it has typical pc functionality. For example, in Windows dialogs, it selects files, checks and unchecks check boxes, and picks radio button choices. It opens a highlighted Help topic.
18	Tab Keys	Use these keys to move between fields in Windows dialogs.
19	Knob	Increments and decrements the value of the current active function.
20	Return Key	Exits the current menu and returns to the previous menu. Has typical pc functionality.
21	Full Screen Key	Pressing this key turns off the softkeys to maximize the graticule display area.
22	Help Key	Initiates a context-sensitive Help display for the current Mode. Once Help is accessed, pressing a front panel key brings up the help topic for that key function.
23	Speaker Control Keys	Enables you to increase or decrease the speaker volume, or mute it.
24	Window Control Keys	These keys select between single or multiple window displays. They zoom the current window to fill the data display, or change the currently selected window. They can be used to switch between the Help window navigation pane and the topic pane.
25	Power Standby On/Off	Turns the analyzer on. A green light indicates power on. A yellow light indicates standby mode. **NOTE** The front-panel switch is a standby switch, *not* a LINE switch (disconnecting device). The analyzer continues to draw power even when the line switch is in standby. The main power cord can be used as the system disconnecting device. It disconnects the mains circuits from the mains supply.

Overview of Key Types

The keys labeled **FREQ Channel**, **System**, and **Marker Function** are all examples of front-panel keys. Most of the dark or light gray keys access menus of functions that are displayed along the right side of the display. These displayed key labels are next to a column of keys called menu keys.

Menu keys list functions based on which front-panel key was pressed last. These functions are also dependant on the current selection of measurement application (**Mode**) and measurement (**Meas**).

If a menu key function's numeric value can be changed, it is called an active function. The function label of the active function is highlighted after that key has been selected. For example, press **AMPTD Y Scale**. This calls up the menu of related amplitude functions. Note the function labeled **Reference Level** (the default selected key in the Amplitude menu) is highlighted. **Reference Level** also appears in the upper left of the display in the measurement information area. The displayed value indicates that the function is selected and its value can now be changed using any of the data entry controls.

Some menu keys have multiple choices on their label like On/Off or Auto/Man. The different choices are selected by pressing the key multiple times. Take an Auto/Man type of key as an example. To select the function, press the menu key and notice that Auto is underlined and the key becomes highlighted. To change the function to manual, press the key again so that Man is underlined. If there are more than two settings on the key, keep pressing it until the desired selection is underlined.

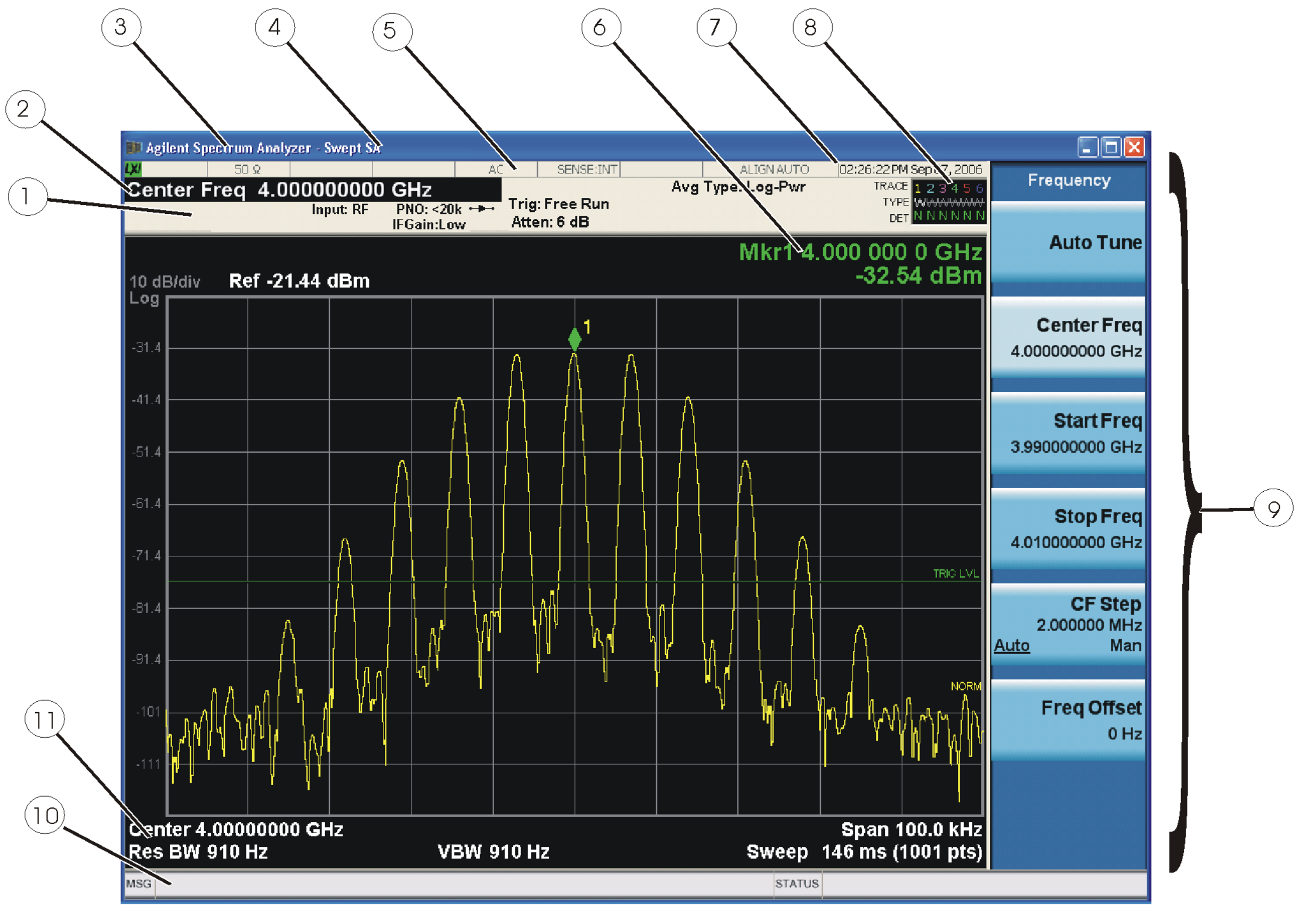
When a menu first appears, one key label will be highlighted to show which key is the default selection. If you press **Marker Function**, the **Marker Function Off** key is the menu's default key, and it will be highlighted. Some of the menu keys are grouped together by a yellow bar running behind the keys near the left side. When you press a key within the yellow bar region, such as **Marker Noise**, the highlight will move to that key to show it has been selected. The keys that are linked by the yellow bar are related functions, and only one of them can be selected at any one time. For example, a marker can only have one marker function active on it. So if you select a different function it turns off the previous selection. If the current menu is two pages long, the yellow bar could include keys on the second page of keys.

In some key menus, a key label will be highlighted to show which key has been selected from multiple available choices. And the menu is immediately exited when you press one of the other keys. For example, when you press the **Select Trace** key (in the **Trace/Detector** menu), it will bring up its own menu of keys. The **Trace 1** key will be highlighted. When you press the **Trace 2** key, the highlight moves to that key and the screen returns to the **Trace/Detector** menu.

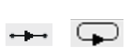
If a displayed key label shows a small solid-black arrow tip pointing to the right, it indicates that additional key menus are available. If the arrow tip is not filled in solid then pressing the key the first time selects that function. Now the arrow is solid and pressing it again will bring up an additional menu of settings.

Display Annotations

This section describes the display annotation as it is on the Spectrum Analyzer Measurement Application display. Other measurement application modes will have some annotation differences.

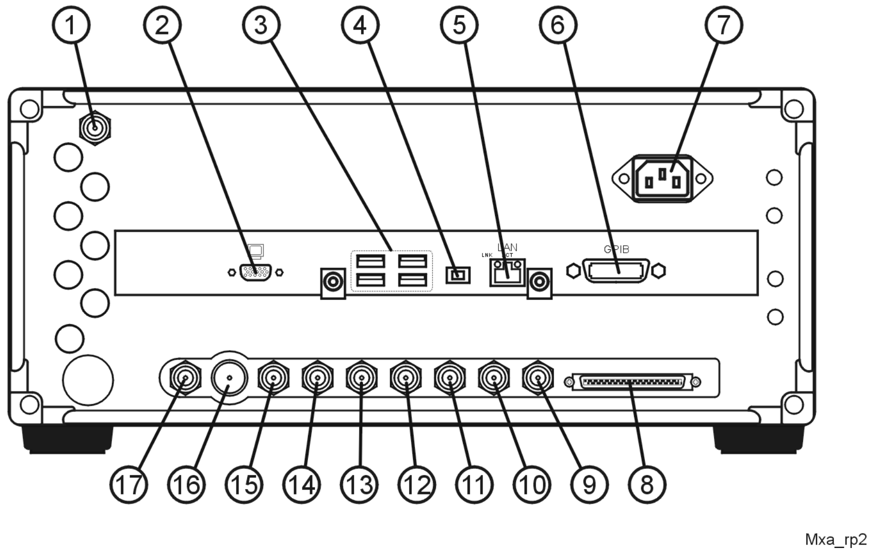


Displayannot

Item	Description	Function Keys
1	Measurement bar - Shows general measurement settings and information.  Indicates single/continuous measurement. Some measurements include limits that the data is tested against. A Pass/Fail indication may be shown in the lower left of the measurement bar.	All the keys in the Analyzer Setup part of the front panel.
2	Active Function (measurement bar) - when the current active function has a settable numeric value, it is shown here.	Currently selected front panel key.

Item	Description	Function Keys
3	Banner - shows the name of the selected measurement application and the measurement that is currently running.	**Mode**, **Meas**
4	Measurement title (banner) - shows title information for the current Measurement, or a title that you created for the measurement.	**Meas** **View/Display**, **Display**, **Title**
5	Settings panel - displays system information that is not specific to any one application. • Input/Output status - green LXI indicates the LAN is connected. RLTS indicate Remote, Listen, Talk, SRQ • Input impedance and coupling • Selection of external frequency reference • Setting of automatic internal alignment routine	**Local** and **System, I/O Config** **Input/Output**, **Amplitude**, **System** and others
6	Active marker frequency, amplitude or function value	**Marker**
7	Settings panel - time and date display.	**System**, **Control Panel**
8	Key labels that change based on the most recent key press.	Softkeys
9	Displays information, warning and error messages. Message area - single events, Status area - conditions	
10	Measurement settings for the data currently being displayed in the graticule area. In the example above: center frequency, resolution bandwidth, video bandwidth, frequency span, sweep time and number of sweep points.	Keys in the Analyzer Setup part of the front panel.

Rear-Panel Features



Item		Description
#	Name	
1	EXT REF IN	Input for a 1 to 50 MHz external frequency reference signal.
2	MONITOR	Allows connection of an external VGA monitor.
3	USB Connectors	Standard USB 2.0 ports, Type A. Connect to external peripherals such as a mouse, keyboard, printer, DVD drive, or hard drive.
4	USB Connector	USB 2.0 port, Type B. USB TMC (test and measurement class) connects to an external pc controller to control the instrument and for data transfers over a 480 Mbps link.
5	LAN	A TCP/IP Interface that is used for remote analyzer operation.
6	GPIB	A General Purpose Interface Bus (GPIB, IEEE 488.1) connection that can be used for remote analyzer operation.
7	Line power input	The AC power connection. See the product specifications for more details.
8	Digital Bus	Reserved for future use.
9	Analog Out	Reserved for future use.
10	TRIGGER 2 OUT	A trigger output used to synchronize other test equipment with the analyzer. Configurable from the Input/Output keys.
11	TRIGGER 1 OUT	A trigger output used to synchronize other test equipment with the analyzer. Configurable from the Input/Output keys.
12	Sync	Reserved for future use.

Item		Description
#	Name	
13	TRIGGER 2 IN	Allows external triggering of measurements.
14	TRIGGER 1 IN	Allows external triggering of measurements.
15	Noise Source Drive +28 V (Pulsed)	Reserved for future use.
16	SNS Series Noise Source	Reserved for future use.
17	10 MHz OUT	An output of the analyzer’s internal 10 MHz frequency reference signal. It is used to lock the frequency reference of other test equipment to the analyzer.

Front and Rear Panel Symbols

| This symbol is used to indicate power ON (green LED).

This symbol is used to indicate power STANDBY mode (yellow LED).

~ This symbol indicates the input power required is AC.

The instruction documentation symbol. The product is marked with this symbol when it is necessary for the user to refer to instructions in the documentation.

The CE mark is a registered trademark of the European Community.

N10149 The C-Tick mark is a registered trademark of the Australian Spectrum Management Agency.

ICES/NMB-001
ISM GRP.1 CLASS A

This is a marking of a product in compliance with the Canadian Interference-Causing Equipment Standard (ICES-001).

This is also a symbol of an Industrial Scientific and Medical Group 1 Class A product (CISPR 11, Clause 4).

The CSA mark is a registered trademark of the Canadian Standards Association International.

This symbol indicates separate collection for electrical and electronic equipment mandated under EU law as of August 13, 2005. All electric and electronic equipment are required to be separated from normal waste for disposal (Reference WEEE Directive 2002/96/EC).

To return unwanted products, contact your local Agilent office, or see http://www.agilent.com/environment/product/ for more information.

2 Making Measurements

Introduction

This chapter provides procedures for making the following Phase Noise measurements:

- Monitor Spectrum - See "Monitor Spectrum Measurements" on page 31

NOTE Monitor Spectrum does not measure Phase Noise, but is used initially to assure the instrument is correctly tuned to the carrier frequency.

- Log Plot - See "Log Plot - with Residual FM, Jitter, RMS Noise" on page 17
- Spot Frequency - See "Spot Frequency Measurements" on page 27

Detailed Instructions are also provided to help you with the following:

- Improving Measurement Accuracy
  - — Smoothing, and Averaging
  - — Signal Tracking
  - — Viewing Signal Drift
- Using Cancellation for Log Plot Measurements
  - — Measuring and Displaying the Displayed Average Noise Level Floor (DANL Floor)
  - — Using a DANL Floor Reference for Cancellation
  - — Creating a Low Phase Noise Reference
  - — Using a Reference Trace for Cancellation
  - — Saving and Recalling Reference Traces

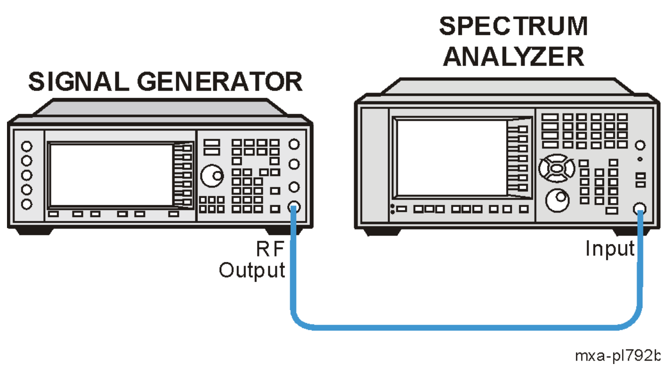
Log Plot - with Residual FM, Jitter, RMS Noise

This chapter explains how to make a Log Plot measurement on a signal source. A Log Plot measures single-sideband phase noise (in dBc/Hz) versus offset frequencies on the horizontal axis shown in logarithmic scale. This allows you to view the phase noise behavior of the signal under test across many decades of offset frequencies.

A Log Plot measurement trace will also provide Residual FM, Jitter, and RMS Noise data when the appropriate frequency marker is used. See "Using Markers to make Integrated Noise Measurements of RMS Noise, Jitter, and Residual FM" on page 20.

This example shows a signal generator under test set up to transmit RF power. The transmitting signal is connected to the RF input port of the instrument. Connect the equipment as shown.

Figure 2-1 **Log Plot Measurement System**



1. Using the appropriate cables and adapters, connect the output signal of the signal generator to the RF input of the analyzer.
2. For best frequency accuracy, connect a BNC cable between the 10 MHz REF IN port of the signal generator (if available) and the 10 MHz EXT REF OUT port of the analyzer.

Basic Log Plot Measurement Procedure

Step 1. Enable the Phase Noise Mode:

Press **Mode**, Phase Noise.

Step 2. Preset the analyzer mode:

Press **Mode Preset**.

Step 3. Verify the RF carrier is present, and the instrument is tuned to the frequency of interest. Use the Monitor Spectrum measurement to view the RF carrier.

Press **Meas**, **Monitor Spectrum**

Step 4. Set the measurement center frequency:

Press **FREQ Channel**, enter a numerical frequency using the front-panel keypad, and select a units key, such as **MHz**.

Step 5. If the RF carrier is not at the center of the display you can use the RPG knob to scroll the carrier to the center of the display.

Step 6. Initiate the Log Plot measurement:

Press **Meas**, **Log Plot**

Step 7. If you are not sure of the carrier frequency use the Auto Tune function to tune the measurement:

Press **Auto Tune**

If the carrier is too far off for the Auto Tune function to determine the carrier frequency, an error will be displayed at the lower right-hand corner of the display. Go back to the Monitor Spectrum measurement and retune the instrument.

Step 8. To allow the measurement to track a drifting signal:

Press **FREQ/Channel**, **Signal Tracking**.

Step 9. To view a table of tabular results at decade offsets, turn ON the Decade Table:

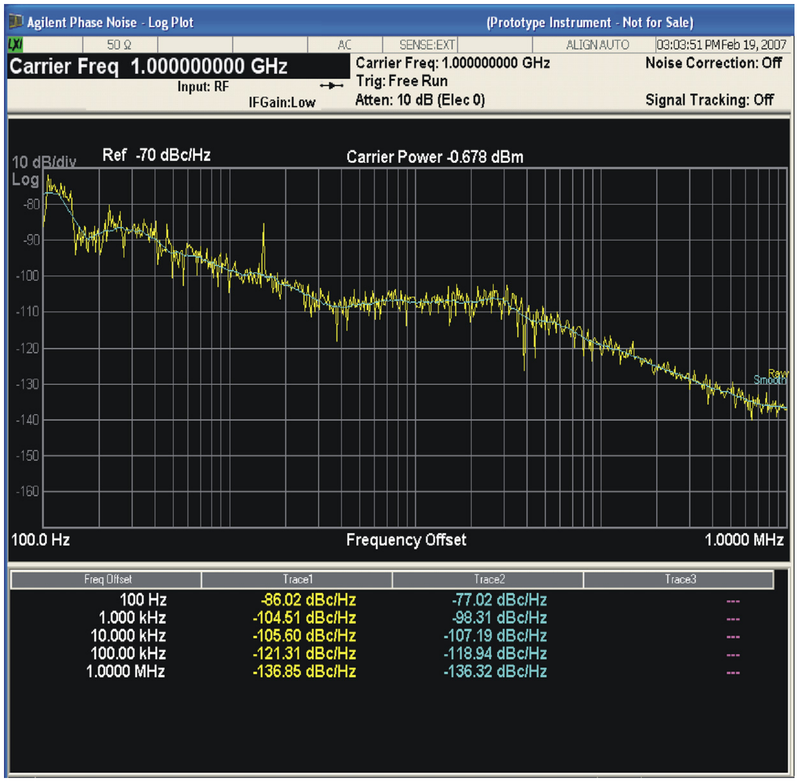
Press **Meas Setup**, **More 1 of 2**, **Decade Table ON**.

Step 10. Select a Phase Noise measurement Offset Range:

Press **SPAN X scale**, **Start Offset** or **Stop Offset** and input a value.

Step 11. The default display (Figure 2-2 on page 19) shows the **Current** (yellow trace) and smoothed (blue trace) data.

Figure 2-2 **Log Plot Measurement - (Default View)**



Step 12. If you are measuring offsets > 1 MHz you can use the Overdrive feature to increase the dynamic range of the instrument. Overdrive uses the electronic step attenuator if available, or the mechanical step attenuator to optimize input signal levels. The measurement is slowed by the respective attenuator switching time. To select Overdrive:

Press **Meas Setup**, **Advanced**, **Overdrive ON**

For more information see "Advanced Features - Using AM Rejection and Overdrive" on page 46.

Step 13. To make viewing the display easier, you can view either the Raw or Smoothed data:

Press **Trace/Detector**, select a **Trace** (**1**, or **2**) and then select whether the data displayed is **Raw**, **Smoothed**, or **Blank**.

Step 14. To make a measurement repeatedly, press **Cont**.

Step 15. You have the option to trade measurement speed for improved accuracy:

Press **FREQ/Channel**, **Tracking**, **Accuracy** and choose between **High (Slowest)** and **Low (Fastest)**. **Medium** is the default measurement speed and accuracy.

For more information see "Log Plot Measurements" on page 43.

Using Markers to make Integrated Noise Measurements of RMS Noise, Jitter, and Residual FM

When the basic Phase Noise measurement (above) is complete you can measure other noise parameters using markers.

Step 1. To activate a marker:

Press **Marker Function**, select a marker number (**1**-**3**) and select from the following:

- **RMS Noise** - Select units in degrees or radians
- **Jitter** - Marker values provided in seconds
- **Residual FM** - Marker values provided in Hz

For more information on these topics see "Phase Noise Measurements" on page 43.

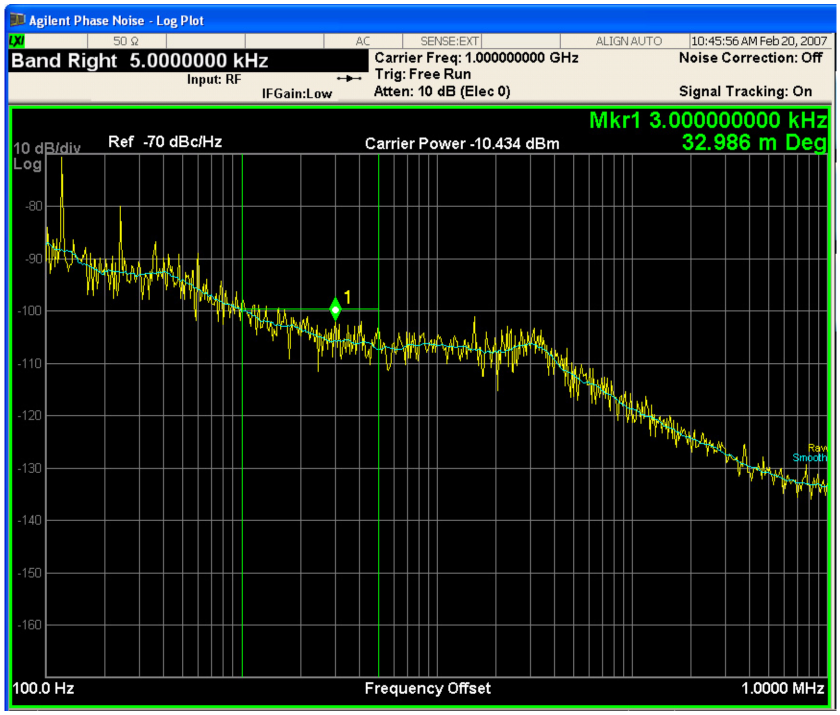
Step 2. Set an integration band:

Press **Band Adjust**, and enter values for **Band/Interval Span, Band/Interval Left** or **Band/Interval Right**. The interval is that portion of the measurement spectrum between the minimum or maximum measurement frequency and the integration band marker. The minimum band/Interval Left value is 100.00000 Hz.

Band markers appear as green vertical lines in the plot, with the center frequency marked as a diamond (see Figure 2-3 on page 21). You can also adjust the RPG knob to set the markers visually, on screen. Because the Log Plot is not linear, and because the marker center location is shown in the linear center of the band, the Band/Interval Left marker will be farther from the center than the Band/Interval Right marker.

The values displayed for the integration bandwidth are based on the last measurement acquisition. Unless the **CONT** button is active, the measurement data will not be refreshed unless either the **SINGLE** or **Restart** button is pressed.

Figure 2-3 **Log Plot Measurement - (Integrated Noise View)**



Using Log Plot to Measure Displayed Average Noise Level (DANL Floor)

The DANL floor of a signal/spectrum analyzer limits measurement of the smallest input signals, those usually found at the extreme offset frequencies. When the amplitude of a signal under test approaches the level of the DANL floor, significant measurement error can occur. To validate a low signal level measurement, you can measure the DANL floor of the analyzer and determine whether the measured phase noise is from the generator, or is due to the DANL contribution.

Step 1. Enable the Phase Noise Mode:

Press **Mode**, Phase Noise.

Step 2. Preset the analyzer mode:

Press **Mode Preset**.

Step 3. Set the measurement center frequency:

Press **FREQ Channel**, enter a numerical frequency using the front-panel keypad, and select a units key, such as **MHz**.

Step 4. Initiate the Log Plot measurement:

Press **Meas**, **Log Plot**

Step 5. Select DANL Floor as the measurement type:

Press **Meas Setup**, **Meas Type**, **DANL Floor**.

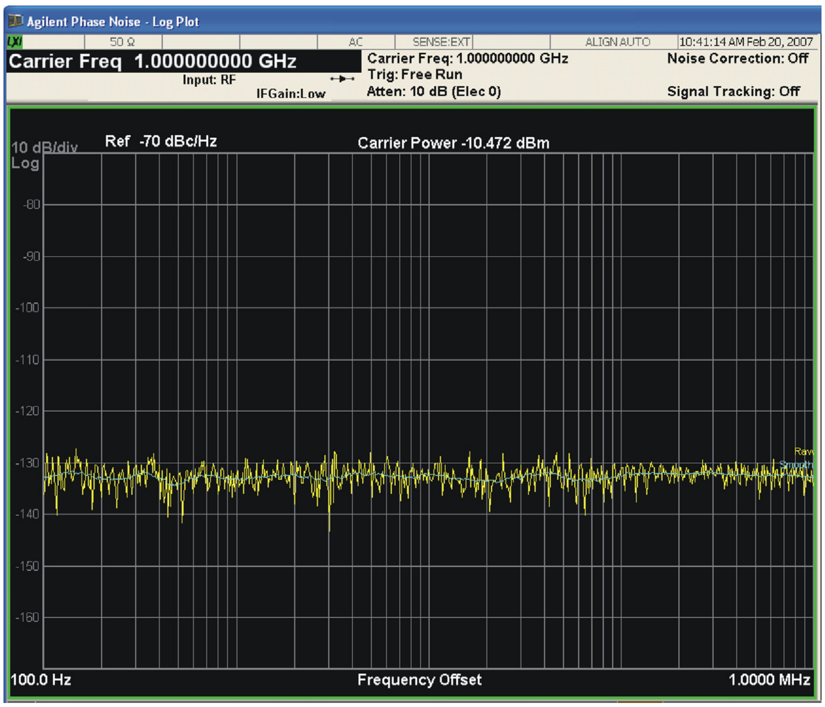
Step 6. Turn on averaging to increase measurement accuracy:

Press **Meas Setup**, **Avg/Hold Number** and enter **20**.

Step 7. Press **Restart**

Step 8. The default display shown below shows the **Current** (yellow trace) and smoothed (blue trace) data.

Figure 2-4 Log Plot Measurement - (DANL Floor Meas Type Default View)



Step 9. Use a marker to see precise DANL Floor levels at a specified frequency, or turn on the Decade Table:

Press **Meas Setup**, **More 1 of 2**, **Decade Table ON**.

Step 10. To make viewing the display easier, you can view either the Raw or Smoothed data:

Press **Trace/Detector**, select a **Trace** (**1**, or **2**) and then select whether the data displayed is **Raw**, **Smoothed**, or **Blank**.

Using a DANL Floor Trace for Phase Noise Measurement Cancellation

Once you have measured the DANL Floor of the analyzer, you can use the trace data to cancel the instrument contribution to the measurement.

Step 1. Complete a DANL Floor measurement as described in "Using Log Plot to Measure Displayed Average Noise Level (DANL Floor)" on page 21.

Step 2. Store a DANL Floor measurement trace as a Reference Trace as described in "Creating a Low Phase Noise Signal Reference Trace" on page 23.

Step 3. Use the DANL Floor Reference Trace with the Cancellation function as described in "Using a Phase Noise Reference Trace for Cancellation" on page 23.

Creating a Low Phase Noise Signal Reference Trace

You can use a known low phase noise source to create a reference trace that be used to characterize another source. The reference trace must cover the same frequency range as your intended measurement.

NOTE For a reference trace to be valid it must cover the same frequency range as your intended measurement.

Step 1. Set up the analyzer to measure the test signal phase noise over the desired frequency range. Press **Measure**, **Log Plot**.

Step 2. Connect a low phase noise signal source to the input of your analyzer and set it to the desired output frequency.

Step 3. Measure the phase noise of your reference signal. Press **Meas Setup**, **Meas Type**, **Phase Noise** and **Restart**. This measures and displays the phase noise of your test signal.

You now have a reference trace available that you can either use immediately or save for later use. See the information about saving and restoring traces later in this section.

Using a Phase Noise Reference Trace for Cancellation

Cancellation is a process where a reference trace is "subtracted" from a Log Plot measurement trace, providing a more accurate result. The reference trace can be of the internal DANL Floor of the analyzer, or a Log Plot trace of a known lower phase noise source for comparison.

NOTE Cancellation may NOT be performed when making a DANL Floor measurement.

Not all Log Plot measurements will benefit from Cancellation. Neither will Cancellation improve all measurements. In general, as the phase noise of the DUT approaches the internal DANL Floor of the analyzer, cancellation provides significant improvement in accuracy, but if the noise levels approach too closely, the results become less accurate. For more information See "Log Plot Cancellation" on page 48.

In the following example, the trace that will be used for reference is trace 2 from the previous analyzer DANL Floor measurement. As trace 2 is normally used by the instrument to show smoothed data, the reference trace will be stored in trace 3, which is not normally used by the instrument.

NOTE Traces saved using a Log Plot measurement are different than other measurement traces. Traces saved from any other measurement except Spectrum Analysis Mode are not compatible with those saved using Log Plot measurements.

Step 1. Complete the acquisition of a Reference trace as detailed in the procedure “Creating a Low Phase Noise Signal Reference Trace” on page 23.

Step 2. Copy the smoothed trace data in Trace 2 (cyan blue) to Trace 3 (magenta pink) for reference:

Press **Trace/Detector**, **More (1 of 2)**, **Copy/Exchange**, **From Trace**, **Trace 2**

Press **To Trace, Trace 3**

Press **Copy Now**

You now have a reference trace stored in Trace number 3.

Step 3. Go to the Cancellation menu under the Meas Setup menu:

Press **Meas Setup**, **More 1 of 2** and **Cancellation**.

Step 4. Select the trace that you wish to use as the reference trace (trace 3):

Press **Ref Trace**, **Trace 3**

Step 5. Set the Threshold value if required, although you will not normally have to change this value. The noise cancellation measurement compares your current measurement with the reference trace on a point by point basis. At each point, the current measurement has to exceed the reference trace by at least the threshold level. If the difference between the source trace and the reference trace is less then the threshold level, then the source trace is assumed to be exactly the threshold level above the reference level. To set the threshold level:

Press **Threshold**, and then enter a threshold level in dB.

Step 6. Turn the cancellation ON:

Press **Cancellation ON** to toggle the setting to **ON**.

Any trace that is displaying smoothed data will change immediately to reflect the noise cancellation.

Saving Traces for Retrieval

All traces, including the reference traces used for noise cancellation measurement, can be saved to floppy disk or to the analyzer's own internal file system (C:). All traces are saved in CSV format. In this example the Reference Trace 3 from the above example will be saved.

Step 1. Select the trace you would like to save and the storage location:

Press **Save**, **Data (Export), Trace**, **Trace 3**

Step 2. Select the storage location and filename:

Press **Save As...**

Save As... launches a menu of navigation buttons and a Windows Explorer window which opens to the default trace location:

`D:\\User_My_Documents\Instrument\desktop\My Documents\PNOISE\data\LPL\traces`

You may use another directory or drive as desired. The default file name is "`Trace_00000.csv`" and the number automatically increments, adding 1 to the last number in the file. If you want to use the default file location and file name:

Press **Save**

For future use, this reference trace can be saved to the analyzer's own internal file system (the C: drive), to a network drive, or to a removable drive via the analyzer's USB port.

It is easiest to use a mouse to browse to another file location, and a keyboard to rename files, but the menu keys, Tab and Arrow buttons can be used effectively.

For more information see the **Save/Recall** section in Phase Noise Help.

Recalling Traces

All traces, including the reference traces used for noise cancellation measurement, can be loaded from a floppy disk or from the analyzer's own internal file system (C:). All traces to be recalled must be saved in CSV format.

NOTE For a reference trace to be valid it must cover the same frequency range as your intended measurement.

Step 1. Select the trace you would like to recall:

Press **Recall**, **Data (Import)**, and select a trace to receive the trace data.

Step 2. Press **Open...**

Open... launches a menu of navigation buttons and a Windows Explorer window which opens to the default trace location:

`D:\\User_My_Documents\Instrument\desktop\My Documents\PNOISE\data\LPL\traces`

Browse to the location, select the file name and you want to open then:

Press **Open**

It is easiest to use a mouse to navigate to another file location, but the buttons can be used, if desired.

For more information see the **Save/Recall** section in Phase Noise Help.

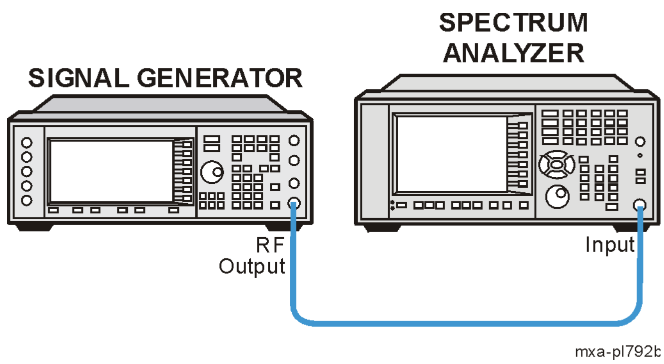
NOTE This method of saving and loading traces in a Log Plot measurement is different from the normal method on ESA and PSA series Spectrum Analyzers. Traces saved from any other measurement except Spectrum Analysis Mode are incompatible with Log Plot measurements.

Spot Frequency Measurements

A spot frequency measurement is a single sideband measurement of the phase noise at a specified offset frequency from the main carrier signal. The average value of the trace points displayed on the screen is indicated by a blue line. The analyzer is normally set up to display a continuous sweep, although a single measurement can be performed by pressing by setting the **Sweep** option to **Single**.

This example shows a signal generator under test set up to transmit RF power. The transmitting signal is connected to the RF input port of the instrument. Connect the equipment as shown.

Figure 2-5 Spot Frequency Measurement System



1. Using the appropriate cables and adapters, connect the output signal of the signal generator to the RF input of the analyzer.
2. For best frequency accuracy, connect a BNC cable between the 10 MHz REF IN port of the signal generator (if available) and the 10 MHz EXT REF OUT port of the analyzer.

Measurement Procedure

Step 1. Enable the Phase Noise Mode:

Press **Mode**, Phase Noise.

Step 2. Preset the analyzer mode:

Press **Mode Preset**.

Step 3. Verify the RF carrier is present, and the instrument is tuned to the frequency of interest. Use the Monitor Spectrum measurement to view the RF carrier.

Press **Meas**, **Monitor Spectrum**

Step 4. Set the measurement center frequency:

Press **FREQ Channel**, enter a numerical frequency using the front-panel keypad, and select a units key, such as **MHz**.

Step 5. If the RF carrier is not at the center of the display you can use the RPG knob to scroll the carrier to the center of the display.

Step 6. Initiate the Spot Frequency measurement:

Press **Meas**, **Spot Frequency**

Step 7. If you are not sure of the carrier frequency use the Auto Tune function to tune the measurement:

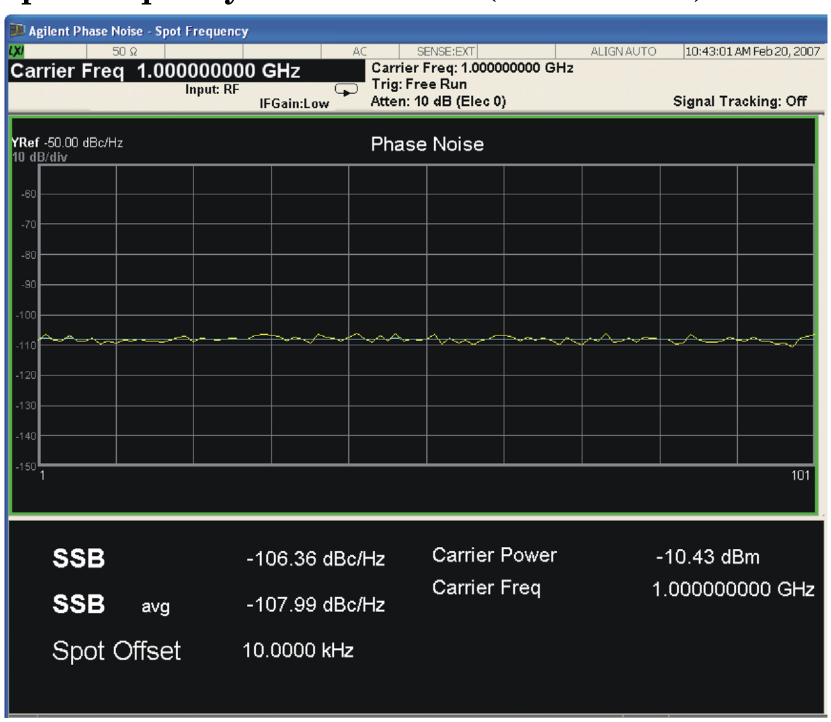
Press **Auto Tune**

If the carrier is too far off for the Auto Tune function to determine the carrier frequency, an error will be displayed at the lower right-hand corner of the display. Go back to the Monitor Spectrum measurement and retune the instrument.

Step 8. The default display shown below shows the **Current** (yellow trace) and smoothed (blue trace) data. The measured value of Phase Noise in dBc/Hz is the vertical scale, and the horizontal scale is in Samples.

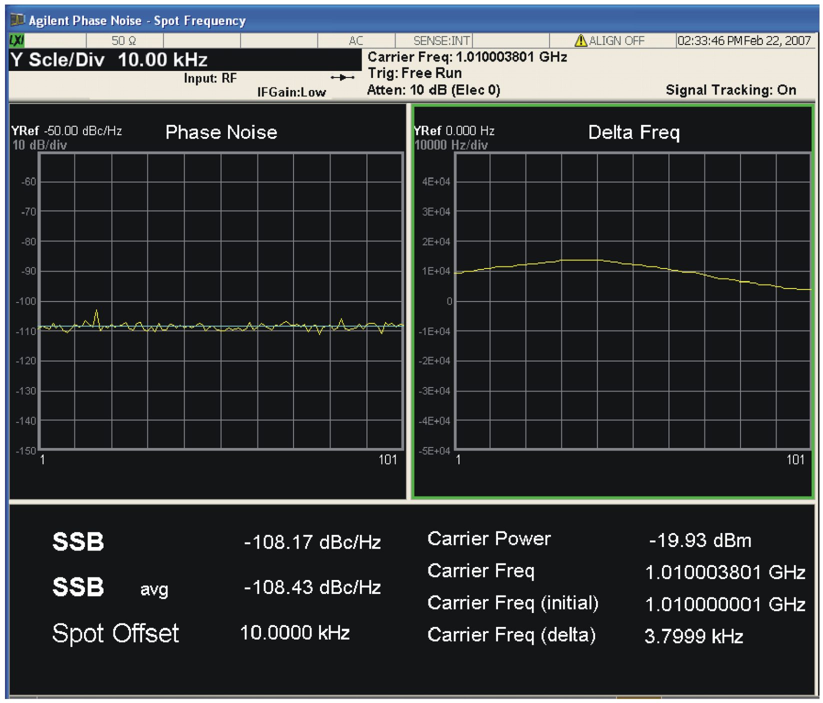
Press **CONT** to view a continuous scrolling display of measurement data.

Figure 2-6 Spot Frequency Measurement - (Default View)



Step 9. The Spot Frequency measurement can be set up to track a drifting signal by pressing **Frequency** and then setting **Signal Track** to **On**. When signal tracking is **On**, another graph showing the change in carrier frequency over time is shown next to the spot frequency trace. The sample numbers in the horizontal scale of both graphs correspond to each other.

Figure 2-7 Spot Frequency Measurement - Signal Tracking ON



Step 10. You have the option to trade measurement speed for accuracy:

Press **FREQ/Channel**, **Tracking**, **Accuracy** and choose between **High (Slowest)** and **Low (Fastest)**. **Medium** is the default measurement speed and accuracy. Using the Fastest measurement will meet the instrument specifications, but is a good choice for strong, stable signals. Use Slowest for drifting or weaker signals.

Step 11. Video filtering can be applied to the active trace when making measurements. Off by default, it is especially useful if a lot of averaging or smoothing is required to settle a measurement. To filter a trace:

Press **Meas Setup**, **Filtering** and select a value depending on the severity of the trace irregularity. Maximum produces the most pronounced effect, but taking the longest time to achieve.

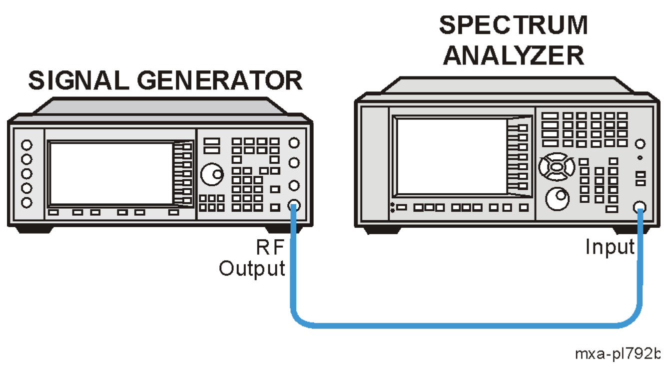
For more information see "Spot Frequency Measurements" on page 44.

Monitor Spectrum Measurements

This chapter explains how to make a Monitor Spectrum measurement on a signal source. Monitor Spectrum measurements show a spectrum domain display of the signal.

This example shows a signal generator under test set up to transmit RF power. The transmitting signal is connected to the RF input port of the instrument. Connect the equipment as shown.

Figure 2-8 **Monitor Spectrum Measurement**



1. Using the appropriate cables and adapters, connect the output signal of the signal generator to the RF input of the analyzer.
2. For best frequency accuracy, connect a BNC cable between the 10 MHz REF IN port of the signal generator (if available) and the 10 MHz EXT REF OUT port of the analyzer.

Measurement Procedure

Step 1. Enable the Phase Noise Mode:

Press **Mode**, Phase Noise.

Step 2. Preset the analyzer mode:

Press **Mode Preset**.

Step 3. Set the measurement center frequency:

Press **FREQ Channel**, enter a numerical frequency using the front-panel keypad, and select a units key, such as **MHz**.

Step 4. Set the measurement span frequency:

Press **SPAN X Scale**, enter a numerical span using the front-panel keypad, and select a units key, such as **MHz**.

Step 5. Initiate the measurement:

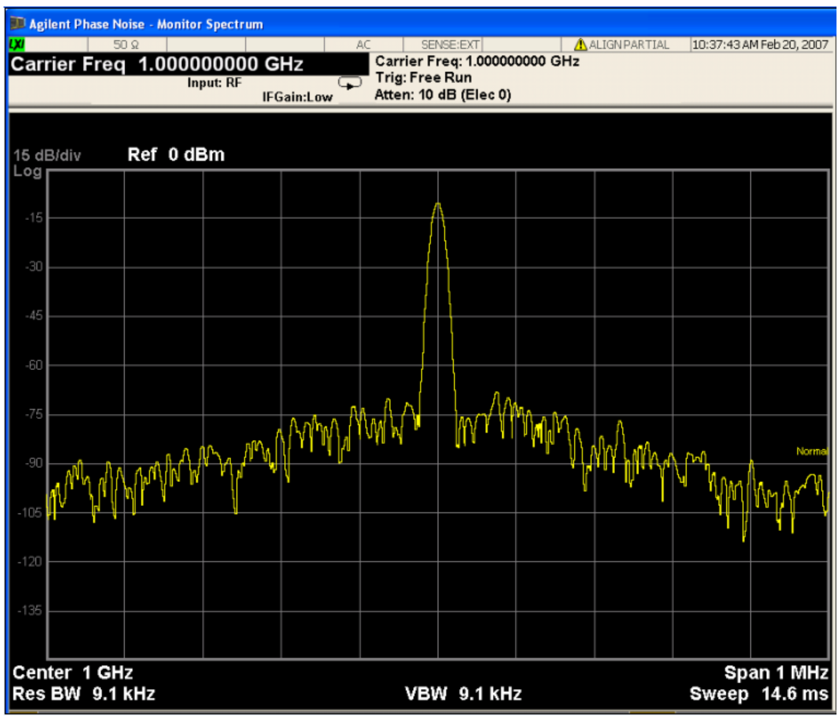
Press **Meas**, **Monitor Spectrum**.

NOTE A display with a Spectrum window appears when you activate a Spectrum measurement. Changes to the **FREQ**, **Span**, or **AMPTD** settings will affect only the active window.

The default display shows the **Current** (yellow trace) data. To make viewing the display easier, you can view either the **Current** trace or **Average** separately.

- Press **Trace/Detector, Select Trace** and select the trace(s) desired for display, then toggle **Display** to **Show**.

Figure 2-9 Monitor Spectrum Measurement - Spectrum and I/Q Waveform (Default View)



Step 6. To make a measurement repeatedly, press **Cont**.

3 Phase Noise Measurement Concepts

Introduction

This chapter includes the following topics:

- What is Phase Noise?
  - — Definition
  - — Thermal Noise
  - — Other Noise Contributions
  - — Single-Sideband Noise
  - — AM Noise
- About Log Plot and Spot Frequency Measurements
- Improving Phase Noise Measurements
  - — Smoothing, Averaging, and Filtering
  - — Signal Tracking
  - — Slowly Drifting Signals
  - — System Noise Floor
  - — Advanced Features - AM Rejection and Overdrive
- Using Cancellation for Log Plot Measurements
  - — Creating DANL Floor Reference
  - — Using DANL Reference for Cancellation
  - — Creating a Phase Noise Reference Trace
  - — Saving and Restoring Reference Traces
- Additional Phase Noise Documentation

What is Phase Noise?

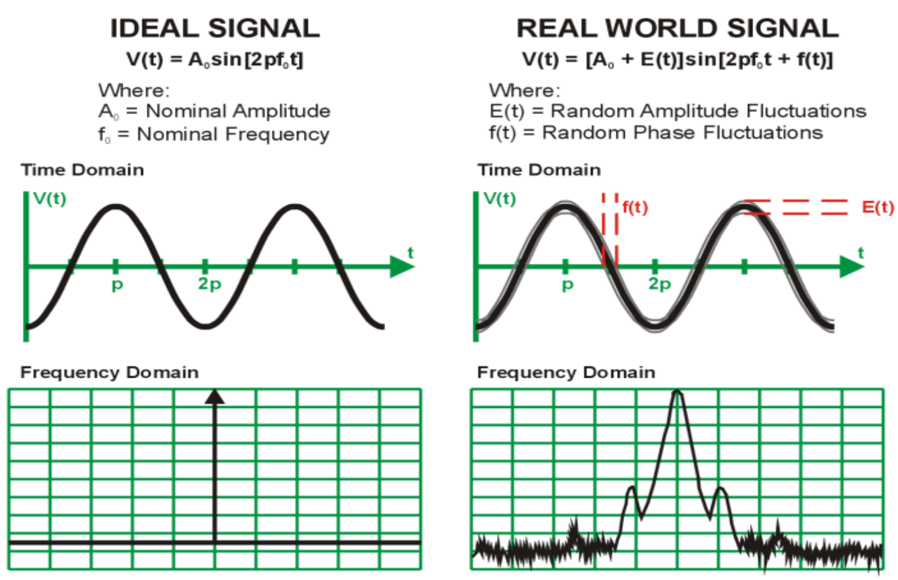
Phase Noise is the term used to describe the aggregate noise power of unwanted modulation products close to a signal, at a specific offset from the actual carrier frequency. As this power is higher near the carrier but can extend far into the sidebands, the usual offsets are decadal to allow logarithmic scale plots of the power levels. Noise power contributions may be due to several various mechanisms, and each will affect the carrier at different offsets. Among these are Thermal Noise, Flicker Noise, and White Noise.

Before we get to the formal definitions of phase noise, let's look at the difference between an ideal signal (a perfect oscillator) and a real world signal. In the frequency domain, the ideal signal is represented by a single spectral line (Figure 3-1).

In the frequency domain, the real world signal is no longer a single discrete spectral line. It is now represented by spread of spectral lines, both above and below the nominal signal frequency in the form of modulation sidebands due to random amplitude and phase fluctuations.

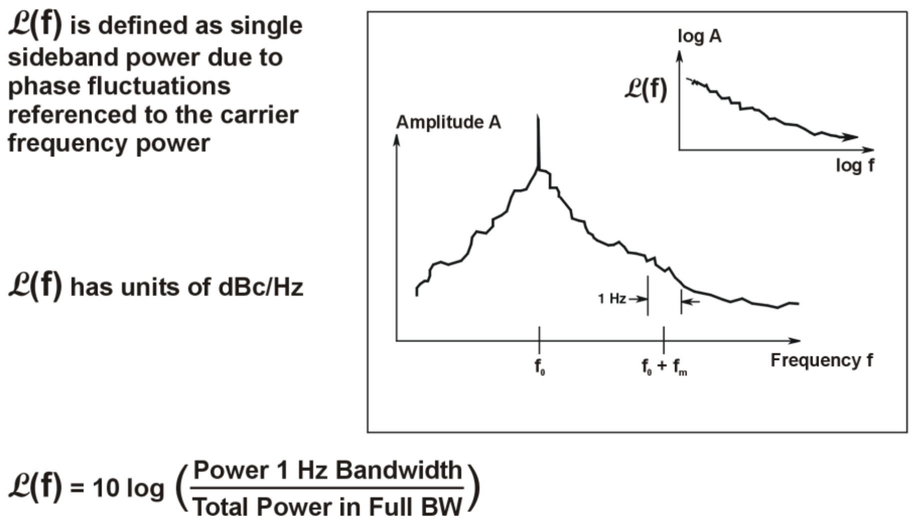
There are always small, unwanted amplitude and phase fluctuations present on a signal. Notice that frequency fluctuations are actually an added term to the phase angle portion of the time domain equation. Because phase and frequency are related, you can discuss equivalently unwanted frequency or phase fluctuations.

Figure 3-1 **Ideal RF Signal vs. Real-World Signal**



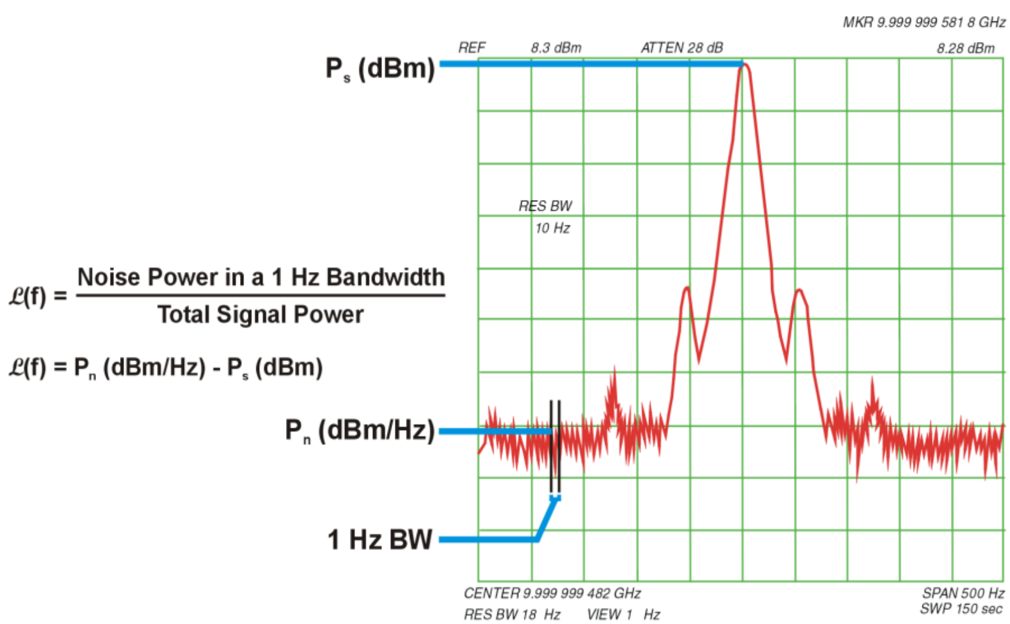
Historically, the most widely used phase noise unit of measure has been total single sideband power within a one hertz bandwidth at a frequency **f** away from the carrier referenced to the carrier frequency power. This unit of measure is represented as a Script L(f) in units of dBc/Hz (Figure 3-2).

Figure 3-2 Phase Noise Unit of Measure



When measuring phase noise directly with a RF spectrum analyzer, the Script L(f) ratio is the noise power in a 1 Hz bandwidth, offset from the carrier at the desired offset frequency, relative to the carrier signal power (Figure 3-3).

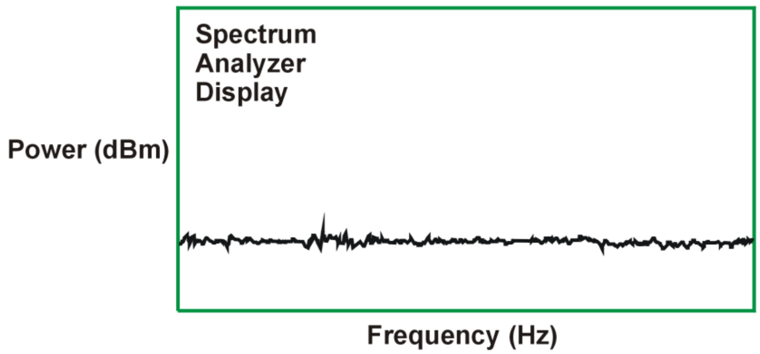
Figure 3-3 L(f) Ratio



Thermal Noise

Thermal noise (kT) is the mean available noise power from a resistor at a temperature T in degrees Kelvin (K). Ambient temperature is assumed to be 290 K. As the temperature of the resistor increases, the kinetic energy of its electrons increases and more power becomes available. Thermal noise is broadband and virtually flat with frequency (Figure 3-4).

Figure 3-4 Thermal Noise Power (Np)



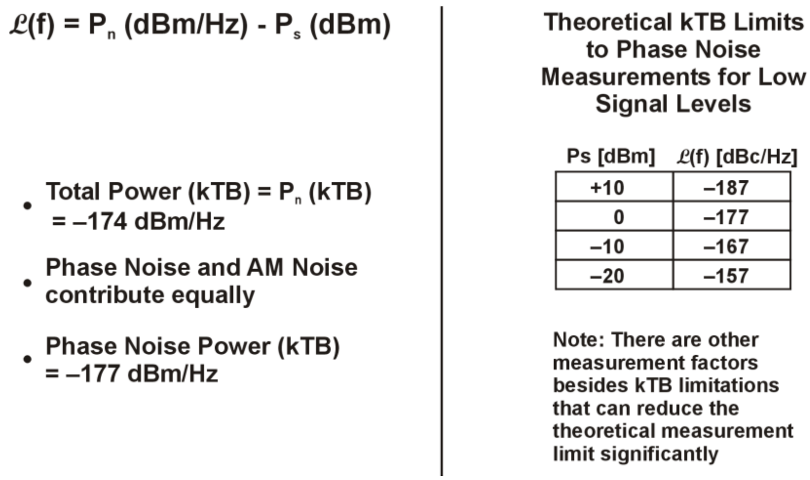
$$N_p = kTB$$

$$\text{For } T = 290\text{ K},\ N_p = -204\ \frac{\text{dB (Watts)}}{\text{Hz}} = -174\ \frac{\text{dBm}}{\text{Hz}}$$

Thermal noise can limit the extent to which you can measure phase noise. The bandwidth term B above is equal to 1 for 1 Hz. Thermal noise as described by kT at room temperature is -174 dBm/Hz. Since phase noise and AM noise contribute equally to kT, the phase noise power portion of kT is equal to -177 dBm/Hz (3 dB less than the total kT power).

If the power in the carrier signal becomes a small value, for example -20 dBm, the limit to which you can measure phase noise power is the difference between the carrier signal power and the phase noise portion of kTB (-177 dBm/Hz - (-20 dBm) = -157 dBc/Hz). Higher signal powers allow phase noise to be measured to a lower dBc/Hz level (Figure 3-5).

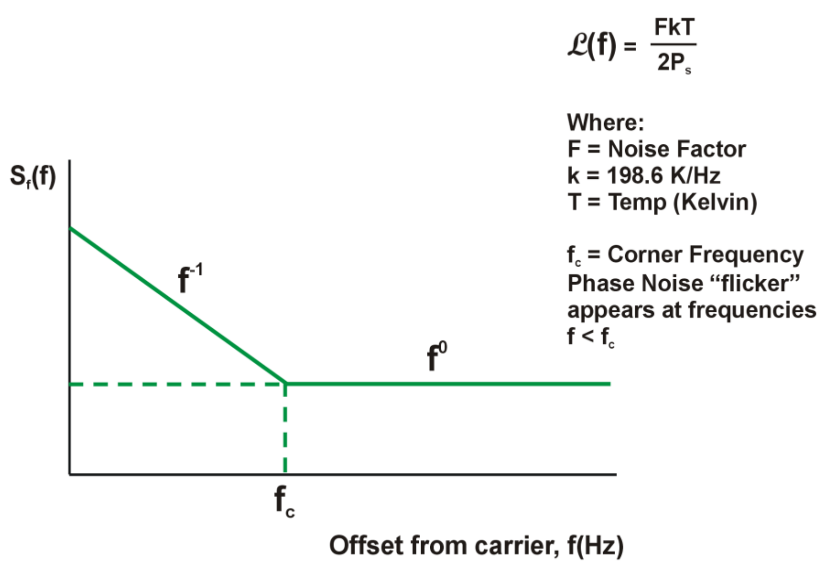
Figure 3-5 Thermal Noise Effects on Phase Noise Measurements (1 Hz BW)



Other Noise Contributions

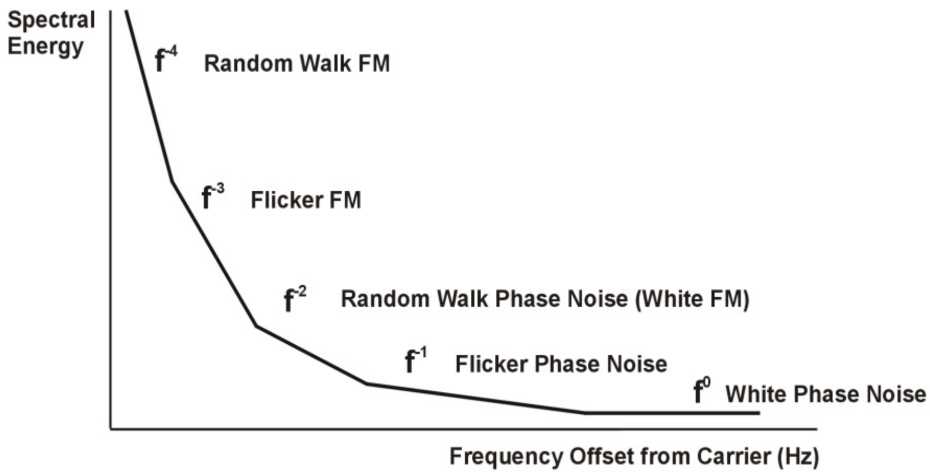
In addition to a thermal noise floor that has an approximately constant level with frequency, active devices exhibit a noise flicker characteristic that intercepts the thermal noise at an empirically determined corner frequency (fc). For offsets below fc, phase noise increases with f^{-1}. This noise is caused by defects within semiconductor lattice structures resulting in combination-recombination of charge carriers (Figure 3-6). Flicker noise power is approximately -120 dBc/Hz @ 1 Hz offset.

Figure 3-6 Flicker Noise



The distribution of other sources of phase noise energy can be described in the terms given in Figure 3-7. Each of these characteristic noise distributions is due to a distinct process in the source circuitry.

Figure 3-7 Typical Phase Noise Distribution



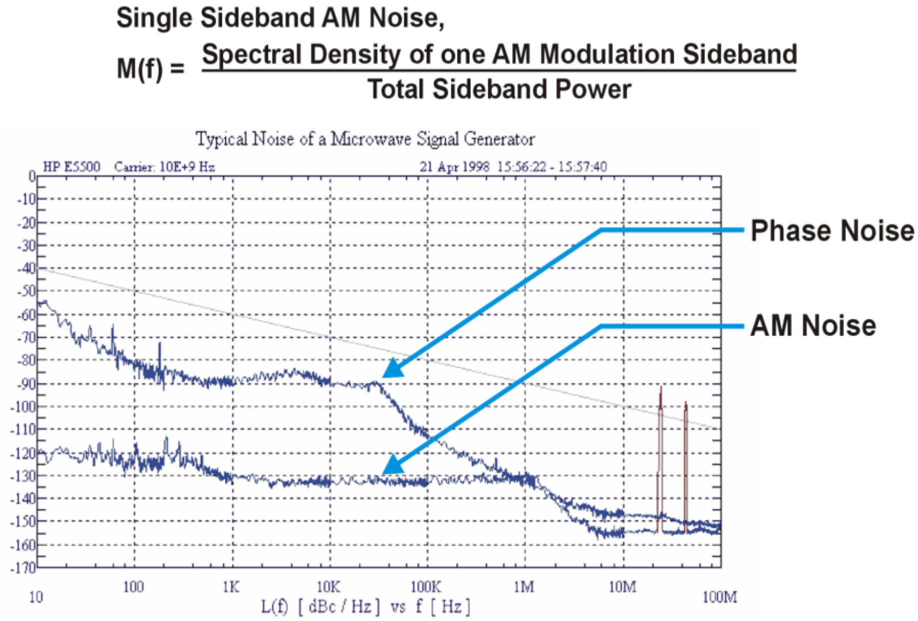
AM Noise

All carriers-with-noise have upper and lower sidebands. These sidebands can alternatively be expressed as “FM” and “AM” sidebands. Most signals measured by phase noise test systems have very little AM sideband power relative to the FM sideband power, so for most signals, measuring the upper (or lower) sideband is equivalent to measuring the FM sideband (phase noise). If the sidebands are due to broadband noise, though, instead of phase noise, they have equal AM and FM power and the upper sidebands have 3 dB more power than the FM sidebands.

AM noise, described here as M(f), is the power density of amplitude noise in a one hertz bandwidth relative to the carrier power. The example shown here indicates that while AM noise can often be found to be much less than phase noise, there can be offset frequencies at which the AM noise can be equal to or even exceed the value of the phase noise.

The AM Rejection feature in the Phase Noise Measurement Application removes the AM noise contribution from your phase noise measurements for offsets <1 Mhz. See “Advanced Features - Using AM Rejection and Overdrive” on page 3-46.

Figure 3-8 AM Noise



Residual FM

Residual FM is a familiar measure of frequency instability commonly used to specify noise inside a data communications bandwidth. Residual FM is the total rms frequency deviation in a specified bandwidth. Commonly used bandwidths have been 50 Hz to 3 kHz, 300 Hz to 3 kHz, and 20 Hz to 15 kHz. Only the short-term frequency instability occurring at rates within the bandwidth is included. No information regarding the relative weighting of instability is conveyed. Therefore, the energy distribution within the bandwidth is lost.

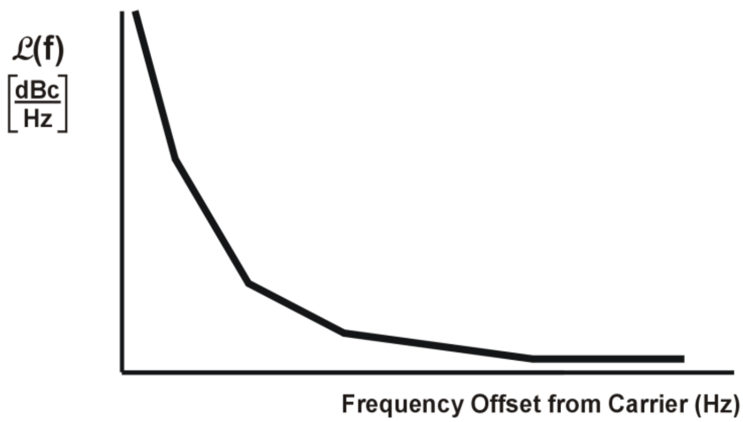
Since spurious signals are detected as FM sidebands, the presence of large spurious signals near the signal under test can greatly increase the measured level of residual FM. You can use the Monitor Spectrum measurement to determine whether these interfering signals are present.

Single-Sideband Noise

L(f) is an indirect measure of noise energy easily related to the RF power spectrum observed on a spectrum analyzer. The historical definition is the ratio of the power in one phase modulation sideband per hertz, to the total signal power. L(f) is usually presented logarithmically as a plot of phase modulation sidebands in the frequency domain, expressed in dB relative to the carrier power per hertz of bandwidth [dBc/Hz].

This historical definition is confusing when the bandwidth of the phase variations are well below 1 Hz because it is possible to have phase noise density values that are greater than 0 dBc/Hz even though the power in the modulation sideband is not greater than the carrier power.

Figure 3-9 Single-Sideband Phase Noise Definition



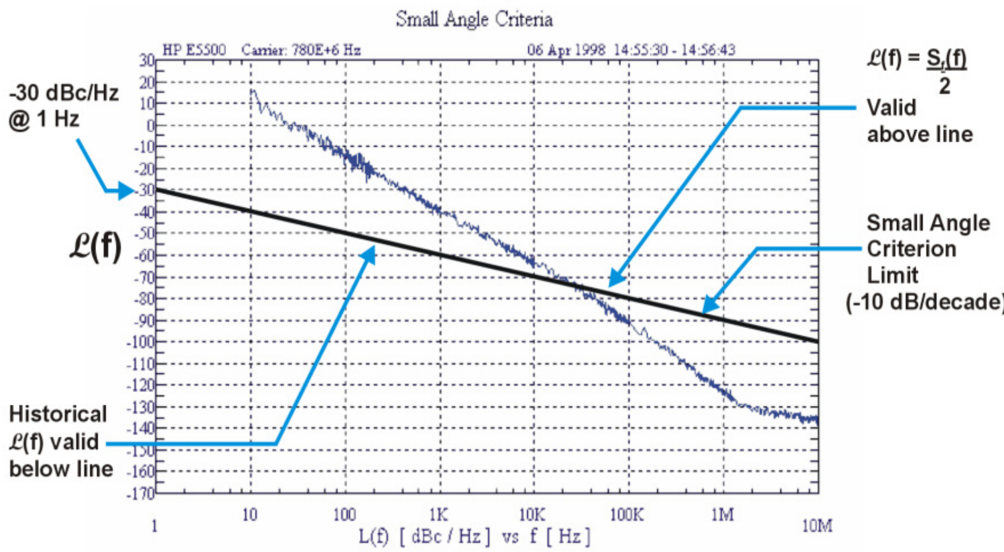
$$\mathcal{L}(f) = \frac{\text{Power Density (one phase modulation sideband)}}{\text{Carrier Power}} \left[\frac{\text{dBc}}{\text{Hz}}\right]$$

$$\mathcal{L}(f) = \frac{S_f(f)}{2}$$

Measurements of L(f) with a spectrum analyzer typically measure phase noise when the phase variation is much less than 1 radian. Phase noise measurement systems, however, measure Sf(f), which allows the phase variation to exceed this small angle restriction. On this graph, the typical limit for the small angle criterion is a line drawn with a slope of -10 dB/decade that passes through a 1 Hz offset at -30 dBc/Hz. This represents a peak phase deviation of approximately 0.2 radians integrated over any one decade of offset frequency.

The plot of L(f) resulting from the phase noise of a free-running VCO illustrates the confusing display of measured results that can occur if the instantaneous phase modulation exceeds a small angle (Figure 3-10). Measured data, $S_f(f)/2$ (dB), is correct, but historical L(f) is obviously not an appropriate data representation as it reaches +15 dBc/Hz at a 10 Hz offset (15 dB more power at a 10 Hz offset than the total power in the signal). The new definition of L(f) $=S_f(f)/2$ allows this condition, since $S_f(f)$ in dB is relative to 1 radian. Results >0 dB simply mean than the phase variations being measured are >1 radian.

Figure 3-10 Single-Sideband Phase Noise L(f)



Phase Noise Measurements

Log Plot Measurements

The log plot measurement gives a display of dBc/Hz versus logarithmic frequency offset for the single sideband measurement. Trace 1, which is the yellow trace, displays the point-by-point data as measured. Trace 2, the cyan blue trace, displays a smoothed version of trace 1. The amount of smoothing is determined by the current setting of the smoothing parameter. With the default settings, marker 1 is set to a frequency offset of 10 kHz, and the phase noise at that frequency is displayed numerically.

NOTE The Trace numbers, trace data and marker data referred to (above) apply if you are using the factory default settings, but these can be changed.

If the analyzer is set up to perform single sweeps, the **Measure Log Plot** softkey or the **Restart** key allow a measurement to be repeated with a single key press. This is useful for seeing effects of circuit changes where the carrier and offset frequencies of interest do not change. The analyzer can also be set up to perform continuous sweeps. In this case a new measurement will be started as soon as the previous one has completed. To make continuous sweeps press the **Cont** (continuous sweep) front panel key.

Up to 12 markers can be used to display various parameters of the measurement, although the default display only shows data for one marker.

Phase noise measurement results can be integrated over a selected frequency range to get the total RMS (root mean squared) noise in a given bandwidth. The frequency limits used for integration may be selected by pressing **Marker** then **RMS Noise**. Use the RPG knob or front panel keys to select the starting point of your frequency range, and then select whether to display the result in radians or in degrees, or in seconds if **RMS Jitter** is selected. Now you can use the RPG knob or front panel keys to select the end point of your frequency range. The results are displayed in radians, degrees or seconds, depending on your previous selection.

RMS Residual FM over a specified range can also be displayed using markers. Using a **Normal** marker, use the RPG knob or front panel keys to position the marker at the start of your frequency range. Then press **Residual FM**, and use the RPG knob (or the front panel keys) to position the second marker at the end point of your frequency range. The display will show your frequency range and the measured RMS residual FM over this range. RMS phase noise measurements are based on the log plot data which is a single-sideband measurement. The RMS phase noise results are mathematically corrected to properly represent the true RMS phase deviation.

Spot Frequency Measurements

A spot frequency measurement is a single sideband measurement of the phase error at a specified offset frequency from the main carrier signal. The average value of the trace points displayed on the screen is indicated by a magenta pink line. The analyzer is normally set up to display a continuous sweep, although a single measurement can be performed by pressing by setting the sweep to **Single**.

The analyzer can be set up to track a drifting signal by pressing **Frequency** and then setting **Signal Track** to **On**. When signal tracking is **On**, a graph with a trace showing the change in frequency against time is shown next to the spot frequency plot.

Improving Phase Noise Measurements

Smoothing, Averaging and Filtering

Repeatability of a measured trace can be improved in several different ways. Smoothing is used with log plot measurements while trace averaging is used with spot frequency measurements. Video filtering can be used with both types of measurements.

The smoothing process averages a number of adjacent trace points from the raw trace, typically Trace 1, and displays the smoothed result in second trace, typically Trace 2, for a log plot measurement. Smoothing is faster than averaging or filtering, but less accurate than either. Loss of accuracy is particularly noticeable when a trace has sudden changes in amplitude, for example when a carrier has a large discrete signal such as a spurious sideband. To smooth a trace, choose the **Smoothing** softkey in **Meas Setup**, and then adjust it between 0.00% and 16.0% using either the front panel keys or the RPG knob. While inside the log plot measurement each level of smoothing can be tried without having to make a new measurement.

Video filtering can be applied to the active trace when making measurements. Additional video filtering improves the repeatability of the measurement but makes the measurement process slower. To filter a trace, choose the **Filtering** softkey in **Meas Setup**.

The averaging process (trace averaging) measured each point on multiple independent sweeps and converges on the average at each point. Trace averaging and video filtering require about the same amount of additional time to reach the same level of repeatability. The biggest difference is that with video filtering, the trace is smooth as it is being created, while with trace averaging, intermediate less smooth results are visible.

The averaging process measures each frequency point multiple times, and then calculates and plots the average value.

Signal Tracking

Signal tracking can be used in all measurements to track a slowly drifting signal. When it is enabled (**On**) , the measurement will follow a slowly drifting signal by reacquiring the carrier signal at the beginning of each trace.

If the signal is not tracked correctly (such as might happen with a rapidly drifting signal), the analyzer may not completely compensate for the drift, with the off-frequency measurement causing the phase noise to appear either higher or lower than it actually is.

Slowly Drifting Signals

Spot frequency and log plot measurements can be made on slowly drifting signals by making use of the signal tracking function, although the measured value may be slightly less accurate.

DANL Floor

The system noise floor can have a significant effect on low phase noise measurements such as those that will typically be found at large frequency offsets. The system noise floor can be measured using one of two methods. See Using Log Plot to Measure Displayed Average Noise Level (DANL Floor) on page 21 for a procedure on measuring DANL Floor. See Using a DANL Floor Trace for Phase Noise Measurement Cancellation on page 22 for a procedure on using the trace to improve accuracy.

Advanced Features - Using AM Rejection and Overdrive

AM Rejection

All carriers-with-noise have upper and lower sidebands. These sidebands can alternatively be expressed as “FM” and “AM” sidebands. Most signals measured by phase noise test systems have very little AM sideband power relative to the FM sideband power, so for most signals, measuring the upper (or lower) sideband is equivalent to measuring the FM sideband (phase noise). If the sidebands are due to broadband noise instead of phase noise, they have equal AM and FM power and the upper sidebands have 3 dB more power than the FM sidebands.

AM Rejection is ON by default for Phase Noise measurements for offsets <1 MHz. Therefore, if a measurement shows a step change in phase noise at 1 MHz offset, you can either change the offset or turn AM rejection OFF to determine whether AM Noise is the reason for the observed step change.

Overdrive

If you are measuring offsets > 1 MHz you can use the Overdrive feature to increase the dynamic range of the instrument. Overdrive uses the electronic step attenuator if available, or the mechanical step attenuator to optimize input signal levels. The measurement is slowed by the respective attenuator switching time. Although measurement accuracy may improve with increased dynamic range, the measurement uncertainty increases slightly with attenuator switching. To select Overdrive:

Press **Meas Setup**, **Advanced**, **Overdrive ON**

Overdrive has the advantage of allowing higher dynamic range at the extreme offsets where broadband input noise of the analyzer might otherwise dominate over the analyzer's phase noise in setting the available dynamic range. Overdrive has the disadvantage of requiring attenuation changes between measuring the carrier power and measuring the noise density offset from that carrier. The accuracy of the phase noise measurement is the accuracy of the ratio of these two. Therefore, phase noise amplitude accuracy is degraded (by the attenuator switching uncertainty) at those offsets where overdrive occurs when it is allowed. Also, switching the attenuator slows the throughput.

Log Plot Cancellation

When you make a phase noise measurement on a given signal, the measurement result that you get is actually a combination of three different noise sources. The first is the phase noise of the signal that you are measuring. If this noise is very small, it can be hidden by the two other noise sources which are generated by the analyzer itself.

The first internal noise source is the phase noise generated by the analyzer as a side-effect of measuring an input signal. The second source is the Displayed Average Noise Level (DANL) of the analyzer. The DANL is the internally generated noise of the analyzer regardless of whether or not an input signal is present, so the DANL is derived from the noise figure of the analyzer. The DANL Floor is broadly flat across the spectrum and represents the absolute noise level below which measurements cannot be made because the signal gets lost in the analyzer noise.

The third noise source is the analyzer's own phase noise. At far offset frequencies, the analyzer's phase noise is often below the analyzer's noise floor (DANL). The DANL floor of an analyzer thus limits the range over which an analyzer can measure phase noise. By making a log plot measurement of the analyzer's DANL noise floor, you are able to characterize the DANL limitation on phase noise measurements.

If you make a phase noise measurement without any input signal, that measurement represents the absolute noise floor (DANL) of the analyzer. If you reference this absolute noise floor to the carrier amplitude, the DANL floor becomes a relative limit below which phase noise sidebands cannot be measured.

The Log Plot measurement accuracy on low phase noise DUTs can be improved by using the cancellation feature to remove the affects of the analyzer's internal noise. This is done by comparing a stored reference measurement with the DUT's measured phase noise.

The stored reference measurement can be generated two ways.

- If you have a signal source that has much better phase noise than the analyzer's phase noise, then you can measure that source and know that the resulting trace represents the analyzer's internal phase noise when an input signal is present.
- If you do not have a good low-phase noise source, you can make a reference measurement with no input signal. This gives you a measurement of the analyzer's noise floor (DANL).

A reference trace from a good source that is relatively free of phase noise will let you compensate for both the phase noise and the DANL of the analyzer. A reference trace that is derived from the DANL only compensates for the DANL portion of the noise, but this may be adequate for measurement conditions where the analyzer DANL is the limiting factor (for offsets above some f_{limit}, where f_{limit} is typically between 1 to 10 MHz).

For future use, this reference trace can be saved to the analyzer's own internal file system (the C: drive), to a network drive, or to a removable drive via the analyzer's USB port. It can then be used later to automatically subtracted from any subsequent log plot measurement to give you a more accurate result.

Cancellation Procedure Overview

Step 1. Set up the analyzer as needed to measure the test signal's phase noise. (i.e. Use the same frequency range as needed for your intended DUT measurement.) See "Using Log Plot to Measure Displayed Average Noise Level (DANL Floor)" on page 21

Step 2. Create a reference trace. Create either a DANL reference or a signal phase noise reference. (See "Using a DANL Floor Trace for Phase Noise Measurement Cancellation" on page 22

Step 3. Save the reference trace. See "Creating a Low Phase Noise Signal Reference Trace" on page 23.)

Step 4. Set up the analyzer so it is making a log plot measurement of the DUT's phase noise and turn on cancellation using the saved reference trace data.

Using Cancellation for Log Plot Measurements

Many phase noise measurements do not benefit from cancellation. If the phase noise of your DUT is more than 10 dB higher then the analyzer noise, then cancellation has almost no effect on the calculated measurement data. The effectiveness of using the cancellation function also has a lower limit. When the phase noise of your DUT gets very close to the analyzer noise (within about 0.1 dB), the logarithmic nature of the calculation results in large, invalid cancellation values. The following table shows error cancellation values that will be applied to the measurement results for various DUT to analyzer phase noise ratios.

Phase Noise of DUT relative to Phase Noise of Analyzer	**Measurement Error Before Cancellation[a]**	**Threshold Δ Required for Maximum Cancellation**
20 dB	0.043 dB	20.0 dB
10 dB	0.41 dB	10.41 dB
0 dB	3.01 dB	3.01 dB
–5.87 dB	6.87 dB	1.0 dB
–10 dB	10.41 dB	0.41 dB
–16.33 dB	16.43 dB	0.1 dB
–20 dB	20.04 dB	0.04 dB
–26.83 dB	26.84 dB	0.01 dB

a. Only considers error due to additive affects of analyzer noise and DUT noise.

Additional Information

The documents listed below provide more information on making phase noise measurements. They can be obtained through your local Agilent sales representative.

Agilent Part Number	Title
5952-0292	AN 150 Spectrum Analyzer Basics
5968-2081E	AN 1309 Pulsed Carrier Phase Noise Measurements
5989-5718EN	Using Clock Jitter Analysis to Reduce BER in Serial Data Applications
5988-6082EN	AN 1397-1 Lowering Cost & Improving Interoperability by Predicting Residual BER:
5954-6365	AN 343-1 Vector Modulation Measurements
5952-8203	AN 283-1 Applications and Measurements of Low Phase Noise Signals Using 8662A

4 Programming Examples

- The programming examples were written for use on an IBM compatible PC.
- The programming examples use C, Visual Basic, or VEE programming languages.
- The programming examples use VISA interfaces (GPIB, LAN, or USB).
- Some of the examples use the IVI-COM drivers.

  Interchangeable Virtual Instruments COM (IVI-COM) drivers: Develop system automation software easily and quickly. IVI-COM drivers take full advantage of application development environments such as Visual Studio using Visual Basic, C# or Visual C++ as well as Agilent's Test and Measurement Toolkit. You can now develop application programs that are portable across computer platforms and I/O interfaces. With IVI-COM drivers you do not need to have in depth test instrument knowledge to develop sophisticated measurement software. IVI-COM drivers provide a compatible interface to all. COM environments. The IVI-COM software drivers can be found at the URL: http://www.agilent.com/find/ivi-com
- Most of the examples are written in C, Visual Basic, VEE, or LabVIew using the Agilent VISA transition library.

  The Agilent I/O Libraries Suite must be installed and the GPIB card, USB to GPIB interface, or Lan interface USB interface configured. The latest Agilent I/O Libraries Suite is available: www.agilent.com/find/iolib
- The STATus subsystem of commands is used to monitor and query hardware status. These hardware registers monitor various events and conditions in the instrument. Details about the use of these commands and registers can be found in the manual/help in the Utility Functions section on the STATus subsystem.

Visual Basic is a registered trademark of Microsoft Corporation.

Available Programing Examples

The following examples work with a Spectrum Analyzer. These examples use one of the following programming languages: Visual Basic® 6, Visual Basic.NET®, MS Excel®, C++, ANSI C, C#.NET, and Agilent VEE Pro.

These examples are available in either the “progexamples” directory on the Agilent Technologies Spectrum Analyzer documentation CD-ROM or the “progexamples” directory in the analyzer. The file names for each example is listed at the end of the example description. The examples can also be found on the Agilent Technologies, Inc. web site at URL:

http://www.agilent.com/find/sa_programming

NOTE These examples have all been test and validated as functional in the Spectrum Analyzer mode. They have not been tested in all other modes. However, they should work in all other modes except where exceptions are noted.

Programming using Visual Basic® 6, Visual Basic.NET® and MS Excel®:

- *Transfer Screen Images* from your Spectrum Analyzer using Visual Basic 6

  This example program stores the current screen image on the instrument flash memory as “D:\PICTURE.PNG”. It then transfers the image over GPIB or LAN and stores the image on your PC in the current directory as “PICTURE.PNG”. The file “D:\PICTURE.PNG” is then deleted on the instrument flash memory.

  File name: _screen.bas

- *Binary Block Trace* data transfer from your Spectrum Analyzer using Visual Basic 6

  This example program queries the IDN string from the instrument and then reads the trace data in Spectrum Analysis mode in binary format (Real,32 or Real,64 or Int,32). The data is then stored to a file "bintrace.txt". This data transfer method is faster than the default ASCII transfer mode, because less data is sent over the bus.

  File name: bintrace.bas

Programming using C++, ANSI C and C#.NET:

- *Serial Poll for Sweep Complete* using C++

  This example demonstrates how to:

1. Perform an instrument sweep.
2. Poll the instrument to determine when the operation is complete.
3. Perform an instrument sweep.

File name: _Sweep.c

- *Service Request Method (SRQ)* determines when a measurement is done by waiting for SRQ and reading Status Register using C++.

  This example demonstrates how:

  1. Set the service request mask to assert SRQ when either a measurement is uncalibrated or an error message has occurred,
  2. Initiate a sweep and wait for the SRQ interrupt,
  3. Poll all instruments and report the nature of the * interrupt on the spectrum analyzer.

  The STATus subsystem of commands is used to monitor and query hardware status. These hardware registers monitor various events and conditions in the instrument. Details about the use of these commands and registers can be found in the manual/help in the Utility Functions section on the STATus subsystem.

  File name: _SRQ.C

- *Relative Band Power Markers* using C++

  This example demonstrates how to set markers as Band Power Markers and obtain their band power relative to another specified marker.

  File name: _BPM.c

- *Trace Detector/Couple Markers* using C++

  This example demonstrates how to:

  1. Set different types of traces (max hold, clear and write, min hold)
  2. Set markers to specified traces
  3. Couple markers

  Note: The Spectrum Analyzer is capable of multiple simultaneous detectors (i.e. peak detector for max hold, sample for clear and write, and negative peak for min hold).

  File name: _tracecouple.c

- *Phase Noise* using C++

  This example demonstrates how to:

  1. Remove instrument noise from the phase noise
  2. Calculate the power difference between 2 traces

  File name: _phasenoise.c

Programming using Agilent VEE Pro:

- *Transfer Screen Images* from my Spectrum Analyzer using Agilent VEE Pro

  This example program stores the current screen image on the instrument flash memory as "`D:\scr.png`". It then transfers the image over GPIB and stores the image on your PC in the desired directory as "`capture.gif`". The file "`D:\scr.png`" is then deleted on the instrument flash memory.

  File name: _ScreenCapture.vee

- *Transfer Trace Data* data transfer using Agilent VEE Pro

  This example program transfers the trace data from your Spectrum Analyzer. The program queries the IDN string from the instrument and supports Integer 32, real 32, real 64 and ASCII data. The program returns 1001 trace points for the signal analyzer.

  File name: transfertrace.vee

Programming Fundamentals

SCPI Language Basics

This section is not intended to teach you everything about the SCPI (Standard Commands for Programmable Instruments) programming language. The SCPI Consortium or IEEE can provide that level of detailed information. For more information refer to the websites for the IEEE Standard 488.1 (IEEE Standard Digital Interface for Programmable Instrumentation).

Topics covered in this chapter include:

- “Command Keywords and Syntax” on page 58
- “Creating Valid Commands” on page 58
- “Special Characters in Commands” on page 59
- “Parameters in Commands” on page 60
- “Putting Multiple Commands on the Same Line” on page 63

Command Keywords and Syntax

A typical command is made up of keywords set off by colons. The keywords are followed by parameters that can be followed by optional units.

Example: `SENSe:FREQuency:STARt 1.5 MHZ`

The instrument does not distinguish between upper and lower case letters. In the documentation, upper case letters indicate the short form of the keyword. The lower case letters, indicate the long form of the keyword. Either form may be used in the command.

Example: `Sens:Freq:Star 1.5 mhz`

is the same as `SENSE:FREQ:start 1.5 MHz`

NOTE The command `SENS:FREQU:STAR` would not be valid because `FREQU` is neither the short, nor the long form of the command. Only the short and long forms of the keywords are allowed in valid commands.

Creating Valid Commands

Commands are not case sensitive and there are often many different ways of writing a particular command. These are examples of valid commands for a given command syntax:

<table>
<tr><th>Command Syntax</th><th>Sample Valid Commands</th></tr>
<tr><td><code>[SENSe:]BANDwidth[:RESolution] <freq></code></td><td>The following sample commands are all identical. They will all cause the same result.
• <code>Sense:Band:Res 1700</code>
• <code>BANDWIDTH:RESOLUTION 1.7e3</code>
• <code>sens:band 1.7KHZ</code>
• <code>SENS:band 1.7E3Hz</code>
• <code>band 1.7kHz</code>
• <code>bandwidth:RES 1.7e3Hz</code></td></tr>
<tr><td><code>MEASure:SPECtrum[n]?</code></td><td>• <code>MEAS:SPEC?</code>
• <code>Meas:spec?</code>
• <code>meas:spec3?</code>
The number 3 in the last meas example causes it to return different results then the commands above it. See the command description for more information.</td></tr>
<tr><td><code>[:SENSe]:DETector[:FUNCtion] NEGative|POSitive|SAMPle</code></td><td>• <code>DET:FUNC neg</code>
• <code>Detector:Func Pos</code></td></tr>
<tr><td><code>INITiate:CONTinuous ON|OFF|1|0</code></td><td>The sample commands below are identical.
• <code>INIT:CONT ON</code>
• <code>init:continuous 1</code></td></tr>
</table>

Special Characters in Commands

<table>
<tr><th>Special Character</th><th>Meaning</th><th>Example</th></tr>
<tr><td>|</td><td>A vertical stroke between parameters indicates alternative choices. The effect of the command is different depending on which parameter is selected.</td><td>Command: <code>TRIGger:SOURce EXTernal|INTernal|LINE</code>
The choices are external, internal, and line.
Ex: <code>TRIG:SOURCE INT</code>
is one possible command choice.</td></tr>
<tr><td></td><td>A vertical stroke between keywords indicates identical effects exist for both keywords. The command functions the same for either keyword. Only one of these keywords is used at a time.</td><td>Command: <code>SENSe:BANDwidth|BWIDth:OFFSet</code>
Two identical commands are:
Ex1: <code>SENSE:BWIDTH:OFFSET</code>
Ex2: <code>SENSE:BAND:OFFSET</code></td></tr>
</table>

Special Character	Meaning	Example
[ ]	Keywords in square brackets are optional when composing the command. These implied keywords will be executed even if they are omitted.	Command: [SENSe:]BANDwidth[:RESolution]:AUTO The following commands are all valid and have identical effects: Ex1: bandwidth:auto Ex2: band:resolution:auto Ex3: sense:bandwidth:auto
< >	Angle brackets around a word, or words, indicates they are not to be used literally in the command. They represent the needed item.	Command: SENS:FREQ <freq> In this command example the word <freq> should be replaced by an actual frequency. Ex: SENS:FREQ 9.7MHz.
{ }	Parameters in braces can optionally be used in the command either not at all, once, or several times.	Command: MEASure:BW <freq>{,level} A valid command is: meas:BW 6MHz, 3dB, 60dB

Parameters in Commands

There are four basic types of parameters: booleans, keywords, variables and arbitrary block program data.

OFF|ON|0|1
(Boolean) This is a two state boolean-type parameter. The numeric value 0 is equivalent to OFF. Any numeric value other than 0 is equivalent to ON. The numeric values of 0 or 1 are commonly used in the command instead of OFF or ON. Queries of the parameter always return a numeric value of 0 or 1.

keyword The keywords that are allowed for a particular command are defined in the command syntax description.

Units Numeric variables may include units. The valid units for a command depend on the variable type being used. See the following variable descriptions. The indicated default units will be used if no units are sent. Units can follow the numerical value with, or without, a space.

Variable A variable can be entered in exponential format as well as standard numeric format. The appropriate range of the variable and its optional units are defined in the command description.

The following keywords may also be used in commands, but not all commands allow keyword variables.

- DEFault - resets the parameter to its default value.
- UP - increments the parameter.
- DOWN - decrements the parameter.
- MINimum - sets the parameter to the smallest possible value.
- MAXimum - sets the parameter to the largest possible value.

The numeric value for the function's MINimum, MAXimum, or DEFault can be queried by adding the keyword to the command in its query form. The keyword must be entered following the question mark.

Example query: `SENSE:FREQ:CENTER? MAX`

Variable Parameters

<integer>	is an integer value with no units.
<real>	Is a floating point number with no units.
<freq> <bandwidth>	Is a positive rational number followed by optional units. The default unit is Hertz. Acceptable units include: Hz, kHz, MHz, GHz.
<time> <seconds>	Is a rational number followed by optional units. The default units are seconds. Acceptable units include: ks, s, ms, µs, ns.
<voltage>	Is a rational number followed by optional units. The default units are Volts. Acceptable units include: V, mV, µV, nV
<current>	Is a rational number followed by optional units. The default units are Amperes. Acceptable units include: A, mA, µA, nA.
<power>	Is a rational number followed by optional units. The default units are W. Acceptable units include: mAW, kW, W, mW, µW, nW, pW.
<ampl>	Is a rational number followed by optional units. The default units are dBm. Acceptable units include: dBm, dBmV, dBµV.
<rel_power> <rel_ampl>	Is a positive rational number followed by optional units. The default units are dB. Acceptable units include: dB.
<percent>	Is a rational number between 0 and 100. You can either use no units or use PCT.

<angle>
<degrees> Is a rational number followed by optional units. The default units are degrees. Acceptable units include: DEG, RAD.

<string> Is a series of alpha numeric characters.

<bit_pattern> Specifies a series of bits rather than a numeric value. The bit series is the binary representation of a numeric value. There are no units.

Bit patterns are most often specified as hexadecimal numbers, though octal, binary or decimal numbers may also be used. In the SCPI language these numbers are specified as:

- Hexadecimal, #Hdddd or #hdddd where ‘d’ represents a hexadecimal digit 0 to 9 and ‘a’ to ‘f’. So #h14 can be used instead of the decimal number 20.
- Octal, #Oddddd or #oddddd where ‘d’ represents an octal digit 0 to 7. So #o24 can be used instead of the decimal number 20.
- Binary, #Bdddddddddddddd or #bdddddddddddddd where ‘d’ represents a 1 or 0. So #b10100 can be used instead of the decimal number 20.

Block Program Data

Some parameters consist of a block of data. There are a few standard types of block data. Arbitrary blocks of program data can also be used.

<trace> Is an array of rational numbers corresponding to displayed trace data. See FORMat:DATA for information about available data formats.

A SCPI command often refers to a block of current trace data with a variable name such as: Trace1, Trace2, or Trace3, depending on which trace is being accessed.

<arbitrary block data> Consists of a block of data bytes. The first information sent in the block is an ASCII header beginning with #. The block is terminated with a semi-colon. The header can be used to determine how many bytes are in the data block. There are no units. You will not get block data if your data type is ASCII, using `FORMat:DATA ASCII` command. Your data will be comma separated ASCII values.

Block data example: suppose the header is #512320.

- The first digit in the header (5) tells you how many additional digits/bytes there are in the header.

- The 12320 means 12 thousand, 3 hundred, 20 data bytes follow the header.
- Divide this number of bytes by your current data format (bytes/data point), either 8 (for real,64), or 4 (for real,32). For this example, if you are using real64 then there are 1540 points in the block.

Putting Multiple Commands on the Same Line

Multiple commands can be written on the same line, reducing your code space requirement. To do this:

- Commands must be separated with a semicolon (;).
- If the commands are in different subsystems, the key word for the new subsystem must be preceded by a colon (:).
- If the commands are in the same subsystem, the full hierarchy of the command key words need not be included. The second command can start at the same key word level as the command that was just executed.

SCPI Termination and Separator Syntax

All binary trace and response data is terminated with <NL><END>, as defined in Section 8.5 of IEEE Standard 488.2-1992, *IEEE Standard Codes, Formats, Protocols and Common Commands for Use with ANSI/IEEE Std 488.1-1987*. New York, NY, 1992. (Although one intent of SCPI is to be interface independent, <END> is only defined for IEEE 488 operation.)

The following are some examples of good and bad commands. The examples are created from a theoretical instrument with the simple set of commands indicated below:

```
[:SENSe]
      :POWer
            [:RF]
            :ATTenuation 40dB

:TRIGger
      [:SEQuence]
      :EXTernal [1]
            :SLOPe
                  POSitive

[:SENSe]
      :FREQuency
            :STARt
      :POWer
      [:RF]
            :MIXer
                  :RANGe
                  [:UPPer]
```

Bad Command	Good Command
`PWR:ATT 40dB`	`POW:ATT 40dB`
The short form of `POWER` is `POW`, not `PWR`.	
`FREQ:STAR 30MHz;MIX:RANG -20dBm`	`FREQ:STAR 30MHz;POW:MIX:RANG -20dBm`
The `MIX:RANG` command is in the same `:SENSE` subsystem as `FREQ`, but executing the `FREQ` command puts you back at the `SENSE` level. You must specify `POW` to get to the `MIX:RANG` command.	
`FREQ:STAR 30MHz;POW:MIX RANG -20dBm`	`FREQ:STAR 30MHz;POW:MIX:RANG -20dBm`
`MIX` and `RANG` require a colon to separate them.	
`:POW:ATT 40dB;TRIG:FREQ:STAR 2.3GHz`	`:POW:ATT 40dB;:FREQ:STAR 2.3GHz`
`:FREQ:STAR` is in the `:SENSE` subsystem, not the `:TRIGGER` subsystem.	
`:POW:ATT?:FREQ:STAR?`	`:POW:ATT?;:FREQ:STAR?`
`:POW` and `FREQ` are within the same `:SENSE` subsystem, but they are two separate commands, so they should be separated with a semicolon, not a colon.	
`:POW:ATT -5dB;:FREQ:STAR 10MHz`	`:POW:ATT 5dB;:FREQ:STAR 10MHz`
Attenuation cannot be a negative value.	

Improving Measurement Speed

There are a number of things you can do in your programs to make them run faster:

"Turn off the display updates" on page 65

"Use binary data format instead of ASCII" on page 65

"Minimize the number of GPIB transactions" on page 66

"Consider using USB or LAN instead of GPIB" on page 66

"Minimize DUT/instrument setup changes" on page 66

"Avoid automatic attenuator setting" on page 67

"Avoid using RFBurst trigger for single burst signals" on page 67

Turn off the display updates

`:DISPlay:ENABle OFF` turns off the display. That is, the data may still be visible, but it will no longer be updated. Updating the display slows down the measurement. For remote testing, since the computer is processing the data rather than a person, there is no need to display the data on the analyzer screen.

Use binary data format instead of ASCII

The ASCII data format is the instrument default since it is easier for people to understand and is required by SCPI for *RST. However, data input/output is faster using the binary formats.

`:FORMat:DATA REAL,64` selects the 64-bit binary data format for all your numerical data queries. You may need to swap the byte order if you are using a PC rather than UNIX. **`NORMal`** is the default byte order. Use **`:FORMat:BORDer SWAP`** to change the byte order so that the least significant byte is sent first. (Real,32 which is smaller and somewhat faster, should only be used if you do not need full resolution for your data. Some frequency data may require full 64-bit resolution.)

When using the binary format, data is sent in a block of bytes with an ASCII header. A data query would return the block of data in the following format: `#DNNN<nnn binary data bytes>`

To parse the data:

- Read two characters (#D), where D tells you how many N characters follow the D character.
- Read D characters, the resulting integer specifies the number of data bytes sent.
- Read the bytes into a real array.

For example, suppose the header is #512320.

- The first character/digit in the header (5) tells you how many additional digits there are in the header.

- The 12320 means 12 thousand, 3 hundred, 20 data bytes follow the header.
- Divide this number of bytes by your current data format (bytes/data point), 8 for real,64. For this example, there are 1540 data points in the block of data.

Minimize the number of GPIB transactions

When you are using the GPIB for control of your instrument, each transaction requires driver overhead and bus handshaking, so minimizing these transactions reduces the time used.

- You can reduce bus transactions by sending multiple commands per transaction. See the information on “Putting Multiple Commands on the Same Line” in the SCPI Language Basics section.
- If you are making the same measurement multiple times with small changes in the measurement setup, use the READ command. It is faster then using INITiate and FETCh.

Consider using USB or LAN instead of GPIB

USB and LAN allow faster data input and output. This is especially important if you are moving large blocks of data. You will not get this improved throughput using LAN if there is excessive LAN traffic (that is, your test instrument is connected to a very busy enterprise LAN). You may want to use a private LAN that is only for your test system.

Minimize DUT/instrument setup changes

- Some instrument setup parameters are common to multiple measurements. You should look at your measurement process with an eye toward minimizing setup changes. If your test process involves nested loops, make sure that the inner-most loop is the fastest. Also, check if the loops could be nested in a different order to reduce the number of parameter changes as you step through the test.
- Are you are using the measurements under the **Meas** key? Remember that if you have already set your Meas Setup parameters for a measurement, and you want to make another one of these measurements later, use READ:<meas>?. The MEASure:<meas>?. command resets all the settings to the defaults, while READ changes back to that measurement without changing the setup parameters from the previous use.
- Are you are using the Measurements under the **Meas** key? Remember that *Mode Setup* parameters remain constant across all the measurements in that mode (for example, center/channel frequency, amplitude, radio standard, input selection, trigger setup). You do not have to re-initialize them each time you change to a

different measurement.

Avoid unnecessary use of *RST

Remember that while *RST does not change the current Mode, it presets all the measurements and settings to their factory defaults. This forces you to reset your analyzer's measurement settings even if they use similar mode setup or measurement settings. See Minimize DUT/instrument setup changes. (Also note that *RST may put the instrument in single measurement/sweep for some modes.)

Avoid automatic attenuator setting

Many of the one-button measurements use an internal process for automatically setting the value of the attenuator. It requires measuring an initial burst to identify the proper attenuator setting before the next burst can be measured properly. If you know the amount of attenuation or the signal level needed for your measurement, just set it.

Note that spurious types of measurements must be done with the attenuator set to automatic (for measurements like: output RF spectrum, transmit spurs, adjacent channel power, spectrum emission mask). These types of measurements start by tuning to the signal, then they tune away from it and must be able to reset the attenuation value as needed.

Avoid using RFBurst trigger for single burst signals

RFBurst triggering works best when measuring signals with repetitive bursts. For a non-repetitive or single burst signals, use the IF (video) trigger or external trigger, depending on what you have available.

RFBurst triggering depends on its establishment of a valid triggering reference level, based on previous bursts. If you only have a single burst, the peak detection nature of this triggering function, may result in the trigger being done at the wrong level/point generating incorrect data, or it may not trigger at all.

Are you making a single burst measurement? To get consistent triggering and good data for this type of measurement application, you need to synchronize the triggering of the DUT with the analyzer. You should use the analyzer's internal status system for this.

The first step in this process is to initialize the status register mask to look for the "waiting for trigger" condition (bit 5). Use **`:STATus:OPERation:ENABle 32`**

Then, in the measurement loop:

1. **`:STATus:OPERation:EVENt?`** This query of the operation event register is to clear the current register contents.
2. **`:READ:PVT?`** initiates a measurement (in this example, for GSM

power versus time) using the previous setup. The measurement will then be waiting for the trigger.

Make sure the attenuation is set manually. Do NOT use automatic attenuation as this requires an additional burst to determine the proper attenuation level before the measurement can be made.

3. Create a small loop that will serial poll the instrument for a status byte value of binary 128. Then wait 1 msec (100 ms if the display is left on/enabled) before checking again, to keep the bus traffic down. These two commands are repeated until the condition is set, so we know that the trigger is armed and ready.
4. Trigger your DUT to send the burst.
5. Return the measurement data to your computer.

NOTE This process cannot be done with the current VXI plug-n-play driver implementation. You will need to use the previous SCPI commands.

Programming in C Using the VTL

The programming examples that are provided are written using the C programming language and the Agilent VTL (VISA transition library). This section includes some basic information about programming in the C language. Note that some of this information may not be relevant to your particular application. (For example, if you are not using VXI instruments, the VXI references will not be relevant).

Refer to your C programming language documentation for more details. (This information is taken from the manual “VISA Transition Library”, part number E2090-90026.) The following topics are included:

Typical Example Program Contents

The following is a summary of the VTL function calls used in the example programs.

`visa.h` This file is included at the beginning of the file to provide the function prototypes and constants defined by VTL.

`ViSession` The `ViSession` is a VTL data type. Each object that will establish a communication channel must be defined as `ViSession`.

`viOpenDefaultRM` You must first open a session with the default resource manager with the `viOpenDefaultRM` function. This function will initialize the default resource manager and return a pointer to that resource manager session.

`viOpen` This function establishes a communication channel with the device specified. A session identifier that can be used with other VTL functions is returned. This call must be made for each device you will be using.

`viPrintf`
`viScanf` These are the VTL formatted I/O functions that are patterned after those used in the C programming language. The `viPrintf` call sends the IEEE 488.2 `*RST` command to the instrument and puts it in a known state. The `viPrintf` call is used again to query

for the device identification (`*IDN?`). The `viScanf` call is then used to read the results.

`viClose` This function must be used to close each session. When you close a device session, all data structures that had been allocated for the session will be de-allocated. When you close the default manager session, all sessions opened using the default manager session will be closed.

Linking to VTL Libraries

Your application must link to one of the VTL import libraries:

32-bit Version:

`C:\VXIPNP\WIN95\LIB\MSC\VISA32.LIB` for Microsoft compilers

`C:\VXIPNP\WIN95\LIB\BC\VISA32.LIB` for Borland compilers

16-bit Version:

`C:\VXIPNP\WIN\LIB\MSC\VISA.LIB` for Microsoft compilers

`C:\VXIPNP\WIN\LIB\BC\VISA.LIB` for Borland compilers

See the following section, "Compiling and Linking a VTL Program" for information on how to use the VTL run-time libraries.

Compiling and Linking a VTL Program

32-bit Applications

The following is a summary of important compiler-specific considerations for several C/C++ compiler products when developing WIN32 applications.

For Microsoft Visual C++ version 2.0 compilers:

- Select `Project | Update All Dependencies` from the menu.
- Select `Project | Settings` from the menu. Click on the `C/C++` button. Select `Code Generation` from the `Use Run-Time Libraries` list box. VTL requires these definitions for WIN32. Click `OK` to close the dialog boxes.
- Select `Project | Settings` from the menu. Click on the `Link` button and add `visa32.lib` to the `Object / Library Modules` list box. Optionally, you may add the library directly to your project file. Click `OK` to close the dialog boxes.
- You may wish to add the include file and library file search paths. They are set by doing the following:

  1. Select `Tools | Options` from the menu.
  2. Click `Directories` to set the include file path.

3. Select `Include Files` from the `Show Directories For` list box.

4. Click `Add` and type in the following:
   `C:\VXIPNP\WIN95\INCLUDE`

5. Select `Library Files` from the `Show Directories For` list box.

6. Click `Add` and type in the following:
   `C:\VXIPNP\WIN95\LIB\MSC`

For Borland C++ version 4.0 compilers:

- You may wish to add the include file and library file search paths. They are set under the `Options | Project` menu selection. Double-click on `Directories` from the `Topics` list box and add the following:

  ```
  C:\VXIPNP\WIN95\INCLUDE
  C:\VXIPNP\WIN95\LIB\BC
  ```

16-bit Applications

The following is a summary of important compiler-specific considerations for the Windows compiler.

For Microsoft Visual C++ version 1.5:

- To set the memory model, do the following:

  1. Select `Options | Project`.

  2. Click `Compiler`, then select `Memory Model` from the `Category` list.

  3. Click the `Model` list arrow to display the model options, and select `Large`.

  4. Click `OK` to close the `Compiler` dialog box.

- You may wish to add the include file and library file search paths. They are set under the `Options | Directories` menu selection:

  ```
  C:\VXIPNP\WIN\INCLUDE
  C:\VXIPNP\WIN\LIB\MSC
  ```

  Otherwise, the library and include files should be explicitly specified in the project file.

Example Program

This example program queries a GPIB device for an identification string and prints the results. Note that you must change the address.

```
/*idn.c - program filename */

#include "visa.h"
```

```
#include <stdio.h>

void main ()
{
     /*Open session to GPIB device at address 18 */
     ViOpenDefaultRM (&defaultRM);
     ViOpen (defaultRM, GPIB0::18::INSTR", VI_NULL,
       VI_NULL, &vi);

     /*Initialize device */
     viPrintf (vi, "*RST\n");

     /*Send an *IDN? string to the device */
     printf (vi, "*IDN?\n");

     /*Read results */
     viScanf (vi, "%t", &buf);

     /*Print results */
     printf ("Instrument identification string: %s\n", buf);

     /* Close sessions */
     viClose (vi);
     viClose (defaultRM);
}
```

Including the VISA Declarations File

For C and C++ programs, you must include the `visa.h` header file at the beginning of every file that contains VTL function calls:

```
#include "visa.h"
```

This header file contains the VISA function prototypes and the definitions for all VISA constants and error codes. The `visa.h` header file includes the `visatype.h` header file.

The `visatype.h` header file defines most of the VISA types. The VISA types are used throughout VTL to specify data types used in the functions. For example, the `viOpenDefaultRM` function requires a pointer to a parameter of type `ViSession`. If you find `ViSession` in the `visatype.h` header file, you will find that `ViSession` is eventually typed as an unsigned long.

Opening a Session

A session is a channel of communication. Sessions must first be opened on the default resource manager, and then for each device you will be using. The following is a summary of sessions that can be opened:

- A **resource manager session** is used to initialize the VISA system. It is a parent session that knows about all the opened sessions. A resource manager session must be opened before any other session can be opened.

- A **device session** is used to communicate with a device on an interface. A device session must be opened for each device you will be using. When you use a device session you can communicate without worrying about the type of interface to which it is connected. This insulation makes applications more robust and portable across interfaces. Typically a device is an instrument, but could be a computer, a plotter, or a printer.

NOTE All devices that you will be using need to be connected and in working condition prior to the first VTL function call (`viOpenDefaultRM`). The system is configured only on the *first* `viOpenDefaultRM` per process. Therefore, if `viOpenDefaultRM` is called without devices connected and then called again when devices are connected, the devices will not be recognized. You must close **ALL** resource manager sessions and re-open with all devices connected and in working condition.

Device Sessions

There are two parts to opening a communications session with a specific device. First you must open a session to the default resource manager with the `viOpenDefaultRM` function. The first call to this function initializes the default resource manager and returns a session to that resource manager session. You only need to open the default manager session once. However, subsequent calls to `viOpenDefaultRM` returns a session to a unique session to the same default resource manager resource.

Next, you open a session with a specific device with the `viOpen` function. This function uses the session returned from `viOpenDefaultRM` and returns its own session to identify the device session. The following shows the function syntax:

viOpenDefaultRM (*sesn*);

viOpen (*sesn*, *rsrcName*, *accessMode*, *timeout*, *vi*);

The session returned from `viOpenDefaultRM` must be used in the *sesn* parameter of the `viOpen` function. The `viOpen` function then uses that session and the device address specified in the *rsrcName* parameter to open a device session. The *vi* parameter in `viOpen` returns a session identifier that can be used with other VTL functions.

Your program may have several sessions open at the same time by creating multiple session identifiers by calling the `viOpen` function multiple times.

The following summarizes the parameters in the previous function calls:

sesn This is a session returned from the `viOpenDefaultRM` function that identifies the resource manager session.

rsrcName	This is a unique symbolic name of the device (device address).
accessMode	This parameter is not used for VTL. Use VI_NULL.
timeout	This parameter is not used for VTL. Use VI_NULL.
vi	This is a pointer to the session identifier for this particular device session. This pointer will be used to identify this device session when using other VTL functions.

The following is an example of opening sessions with a GPIB multimeter and a GPIB-VXI scanner:

```
ViSession defaultRM, dmm, scanner;
.
.
viOpenDefaultRM(&defaultRM);
viOpen (defaultRM, "GPIB0::22::INSTR", VI_NULL,
    VI_NULL, &dmm);
viOpen (defaultRM, "GPIB-VXI0::24::INSTR", VI_NULL,
    VI_NULL, &scanner);
.
.
viClose (scanner);
viClose (dmm);
viClose(defaultRM);
```

The above function first opens a session with the default resource manager. The session returned from the resource manager and a device address is then used to open a session with the GPIB device at address 22. That session will now be identified as **dmm** when using other VTL functions. The session returned from the resource manager is then used again with another device address to open a session with the GPIB-VXI device at primary address 9 and VXI logical address 24. That session will now be identified as **scanner** when using other VTL functions. See the following section for information on addressing particular devices.

Addressing a Session

As seen in the previous section, the *rsrcName* parameter in the `viOpen` function is used to identify a specific device. This parameter is made up of the VTL interface name and the device address. The interface name is determined when you run the VTL Configuration Utility. This name is usually the interface type followed by a number. The following table illustrates the format of the *rsrcName* for the different interface types

Interface	**Syntax**
VXI	VXI [*board*]::*VXI logical address*[::INSTR]
GPIB-VXI	GPIB-VXI [*board*]::*VXI logical address*[::INSTR]

Interface	**Syntax**
GPIB	GPIB [*board*]::*primary address*[::*secondary address*][::INSTR]

The following describes the parameters used above:

board This optional parameter is used if you have more than one interface of the same type. The default value for *board* is 0.

VSI logical address This is the logical address of the VXI instrument.

primary address This is the primary address of the GPIB device.

secondary address This optional parameter is the secondary address of the GPIB device. If no secondary address is specified, none is assumed.

INSTR This is an optional parameter that indicates that you are communicating with a resource that is of type **INSTR**, meaning instrument.

NOTE If you want to be compatible with future releases of VTL and VISA, you must include the INSTR parameter in the syntax.

The following are examples of valid symbolic names:

XI0::24::INSTR Device at VXI logical address 24 that is of VISA type INSTR.

VXI2::128 Device at VXI logical address 128, in the third VXI system (VXI2).

GPIB-VXI0::24 A VXI device at logical address 24. This VXI device is connected via a GPIB-VXI command module.

GPIB0::7::0 A GPIB device at primary address 7 and secondary address 0 on the GPIB interface.

The following is an example of opening a device session with the GPIB device at primary address23.

```
ViSession defaultRM, vi;

.
.

viOpenDefaultRM (&defaultRM);

viOpen (defaultRM, "GPIB0::23::INSTR", VI_NULL,VI_NULL,&vi);

.
.
```

```
viClose(vi);

viClose (defaultRM);
```

Closing a Session

The viClose function must be used to close each session. You can close the specific device session, which will free all data structures that had been allocated for the session. If you close the default resource manager session, all sessions opened using that resource manager will be closed.

Since system resources are also used when searching for resources (viFindRsrc) or waiting for events (viWaitOnEvent), the viClose function needs to be called to free up find lists and event contexts.