

“Using SystemVue to Integrate Wireless PHY Design, Validate, and Test”

Application Note

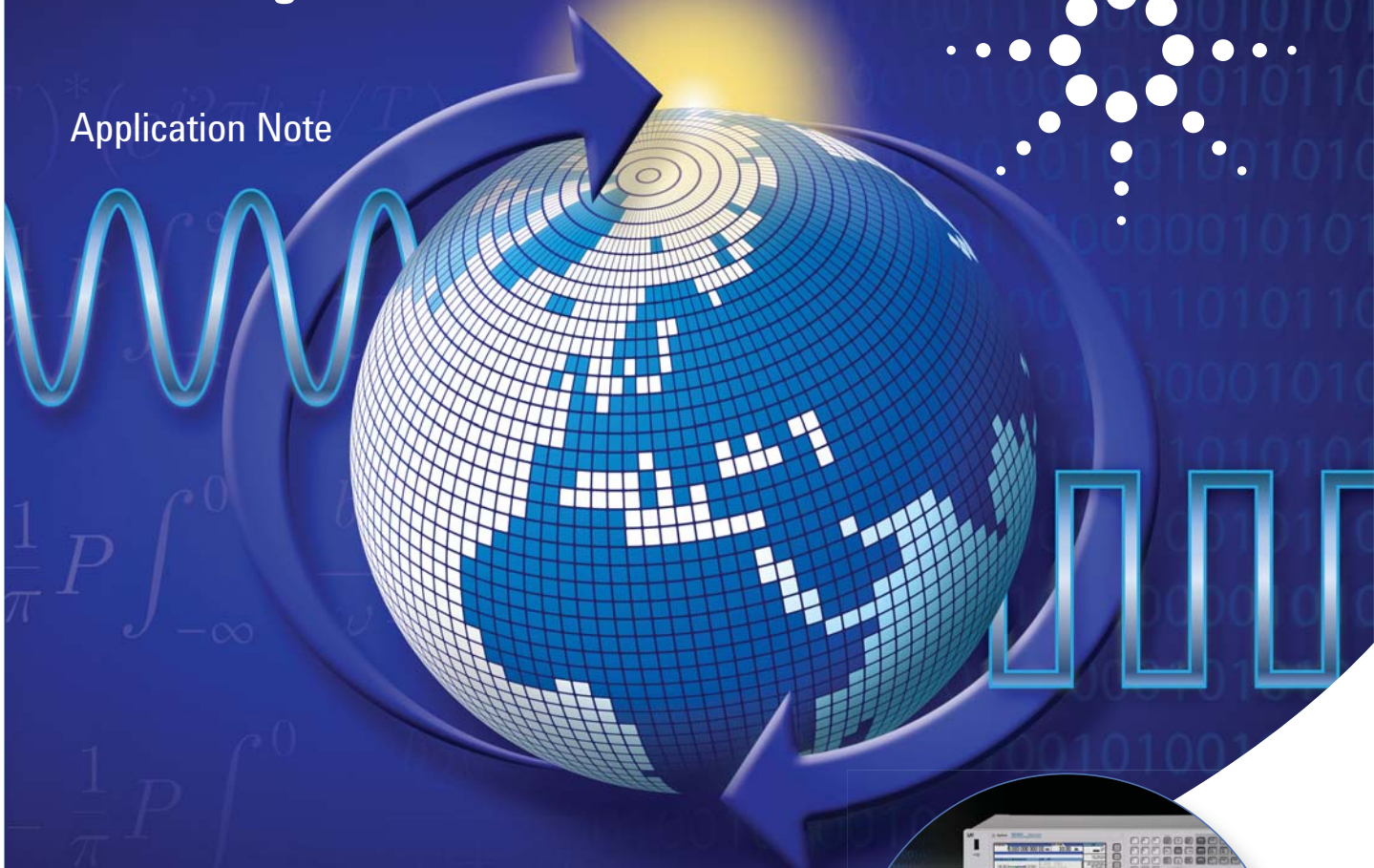


Table of Contents

Introduction	2
Getting Started with DVT	7
Hands-On Lab Exercises	18
Conclusion.....	35
Appendix: Configuration Reference	36



Agilent Technologies

What is Design Validate Test (“DVT”)?

This application note introduces an integrated approach to model-based design for Wireless and Defense applications, called “Design-Validate-Test” (DVT). As a hybrid approach to challenging communications system design, DVT uses stimulus/response test equipment to bring real-world effects into the design of communications architectures and physical layer algorithm development. Agilent Technologies’ SystemVue software is used to integrate these domains into a single, versatile development platform. This application note describes the basic steps for connecting SystemVue to Agilent wireless test equipment, and then applies the integrated environment to a few practical examples. Future notes in this series will explore other applications beyond Wireless verification.

Motivation and Benefits of “DVT”

Traditionally, RF and DSP designers have worked separately to produce architectures within their respective tool domains. However, today’s emerging communications and aerospace/defense systems require cross-domain trade-offs and debugging to achieve extreme linearity, bandwidth and robustness to interference.

There are several benefits of the DVT approach. Modelers gain practical performance insight from measurement-based validation of initial algorithmic designs, to help:

- diagnose and solve cross-domain problems early in R&D;
- reduce excess design margin in both baseband/DSP and RF transceiver architectures; and
- create an accelerated model-based design methodology that sits above traditional RF and baseband hardware design flows, and connects to them, making them more powerful.

The DVT process produces increased, early confidence that communications and aerospace/defense designs will not only work in the real world, but also achieve superior results, given available processing power, analog performance and environmental conditions.

Design Challenges Addressed by “DVT”

Systems development for Layer 1 communications is traditionally split between baseband/DSP and RF/analog signal teams. While baseband/DSP and RF components may initially be architected together, and even verified together as a hardware prototype, there are intermediate states of component development that are difficult to validate until all components are ready for cross-domain integration.

The channel is often unaccounted for influence: that is, the combined effects of the external signaling environment, including propagation, fading, traffic from other users, interference, jamming, and impairments of the air interface. Another design challenge is functional gaps. R&D groups may concentrate on one component of an overall system and, due to vendor or IP barriers, not have access to complementary parts to complete a virtual system. SystemVue’s DVT approach can fill these intermediate verification gaps (Figure 1). Table 1 shows some of the communications design and verification challenges that can be addressed by SystemVue using a DVT approach.

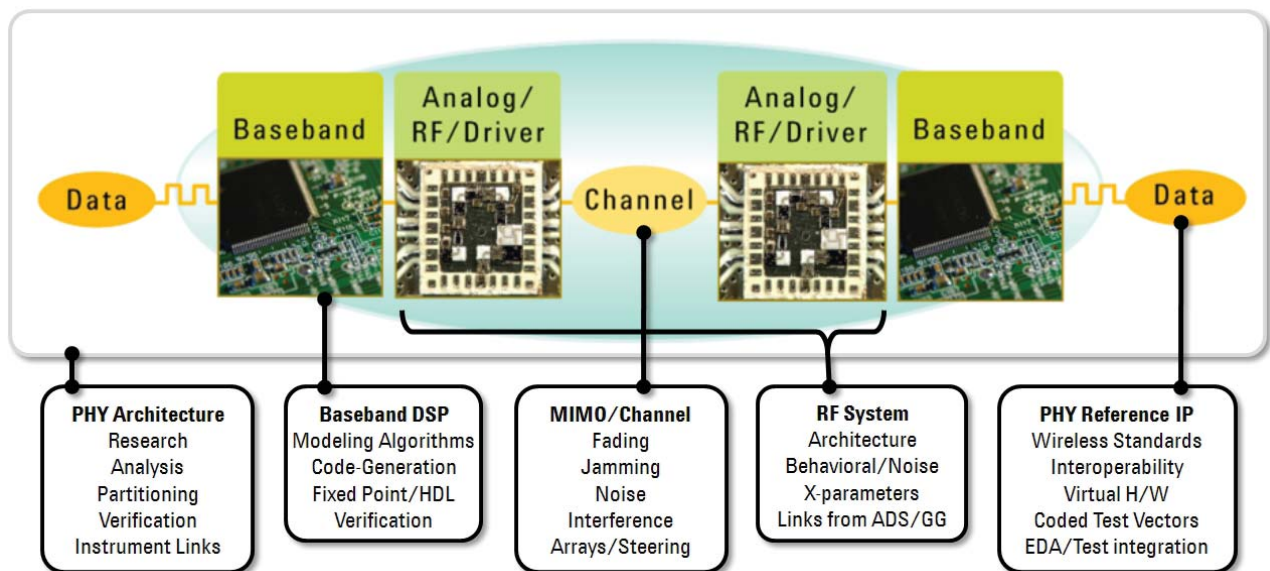


Figure 1. SystemVue can fill gaps in key areas to enable system architects to continuously evaluate the link-level performance throughout a cross-domain communications design effort.

The DVT approach is able to “virtualize” the missing pieces of an integrated system, including the RF. This allows both the component designer and the system architect to maintain a link-level perspective of physical layer performance from the first day of a project.

At any point in the mutual baseband/RF design flow, an individual algorithm, hardware block or analog subsystem can be represented using either simulations or measurements, and the performance evaluated, both at a detailed physical signaling level (e.g., digital debug using a logic analyzer), as well as at the link-level (e.g., system throughput or BER/FER). This is what is meant by cross-domain model-based design.

Design problems addressed	Verification problems addressed
<ul style="list-style-type: none"> Simulation of missing transmit or receive hardware, to complete a virtual system Environmental effects, and precisely-degraded signaling environments in the investigative/architecture stage Continuous verification of systems that are partially-implemented, or mixed-hierarchy prototypes Improved RF/analog modeling fidelity Unified approach to RF & baseband/DSP 	<ul style="list-style-type: none"> Early pre-commercial standards support for test instruments Platform for structured verification scripting and test plan execution Consistency of approaches and measurements for validation of algorithms vs. hardware MIMO hardware multiplicity for early R&D validation

Table 1. A sampling of communications design and verification challenges addressed by a “Design-Validate-Test” approach, based on Agilent SystemVue.

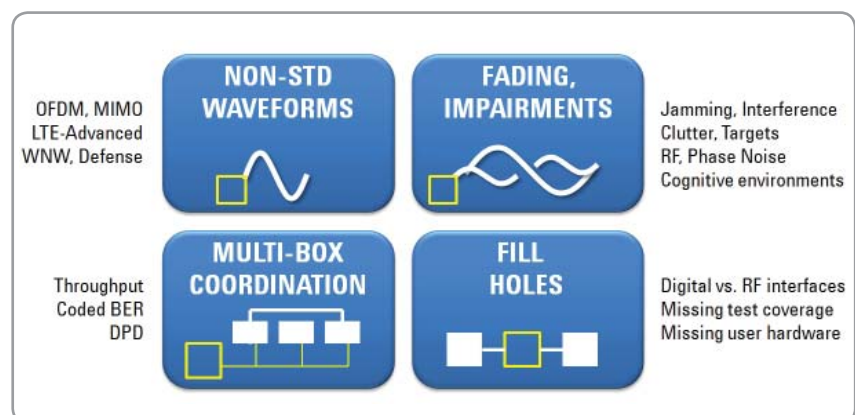


Figure 2. Connecting to test improves the fidelity of system architecture design. Conversely, leveraging the design platform for early R&D validation creates new value for test.

Software-Defined Instruments Methodology

With the DVT approach, wideband stimulus/response test equipment abstracts into a software-defined instrument within the design process, increasing the fidelity of the simulations at the earliest possible phases of design. As a design transitions to either hardware evaluation, coverage testing of an emerging standard or more comprehensive physical layer pre-compliance, it then becomes necessary to test a greater volume of test vectors, as well as to run signals at their true speeds and eliminate speed bottlenecks. Simulation can still play a role in later phases of R&D. For example, simulation can reduce the urgency for multi-channel MIMO hardware testing and bridge gaps in incomplete systems (e.g., a working handset that needs to interact with a basestation). Simulation can help begin the true hardware evaluation phases earlier than previously possible.

The elegance of the Agilent DVT approach is that the design environment (SystemVue) transitions along with the user into an integrated test environment, thus promoting high levels of reuse for assets, scripting and IP (Figure 3). The DVT approach can be applied just as easily to multi-mode SoC and ASIC development; to FPGA and embedded DSP/ASSP development; to board-level integration; and to RF technologies. As an example, SystemVue's LTE-Advanced reference library can be effectively used to deliver an instrument-grade verification experience (e.g., BER/throughput testing) to an SoC algorithm designer who has not yet taped-out his first wafer, or prototyped using an FPGA.

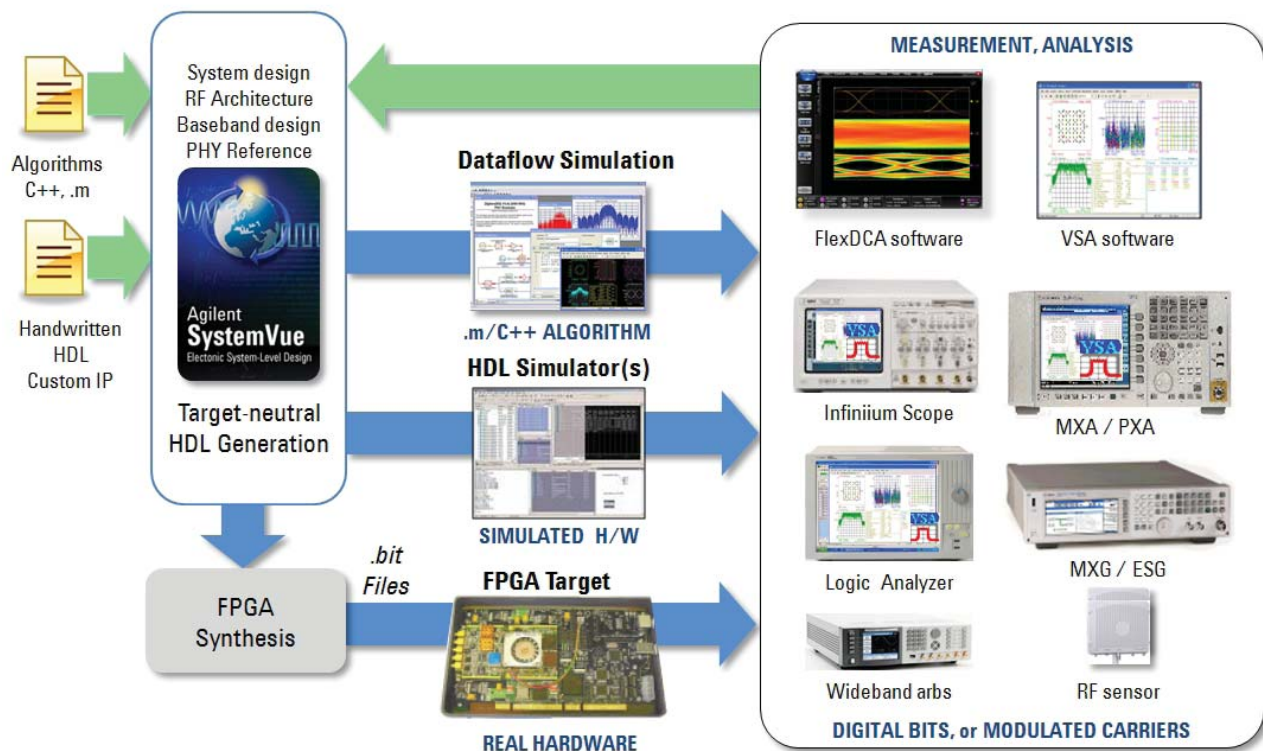


Figure 3. SystemVue is a platform for cross-domain, model-based design of challenging baseband-RF communications systems.

Connecting SystemVue to Test Equipment

This application note is the first of a series. It concentrates on interaction with basic RF stimulus/response test equipment, to establish:

- a connected signal source capability between SystemVue and the Agilent MXG/ESG families of signal generators,
- the connected signal analysis capability between SystemVue and Agilent's families of signal analyzers, greatly leveraging the Agilent 89600 VSA software (Figure 4).

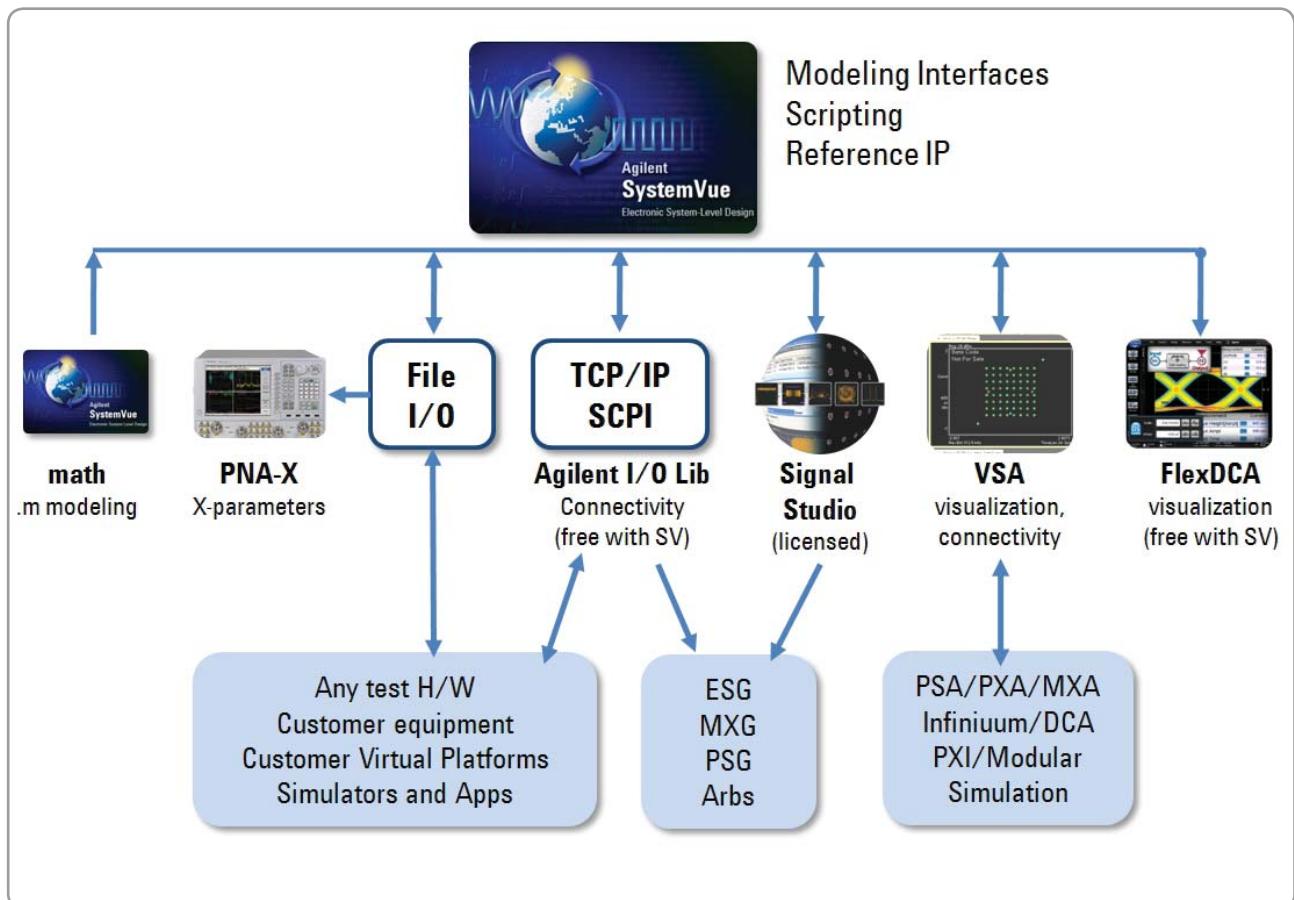


Figure 4. Shown here are ways that SystemVue connects algorithms to the outside world.

This document assumes the reader has a basic familiarity with digital communications concepts, with SystemVue, and with Agilent signal sources and analyzers. An overview is given, followed by a short, self-guided hands-on exercise, so the reader can begin to integrate instrumentation into the simulation environment.

Getting Started

Many configurations are theoretically possible, so a simple case is presented first (Figure 5). Further application notes in this series will elaborate on this foundation with more detailed configurations, scripting, and applications built on top of these basic pieces.

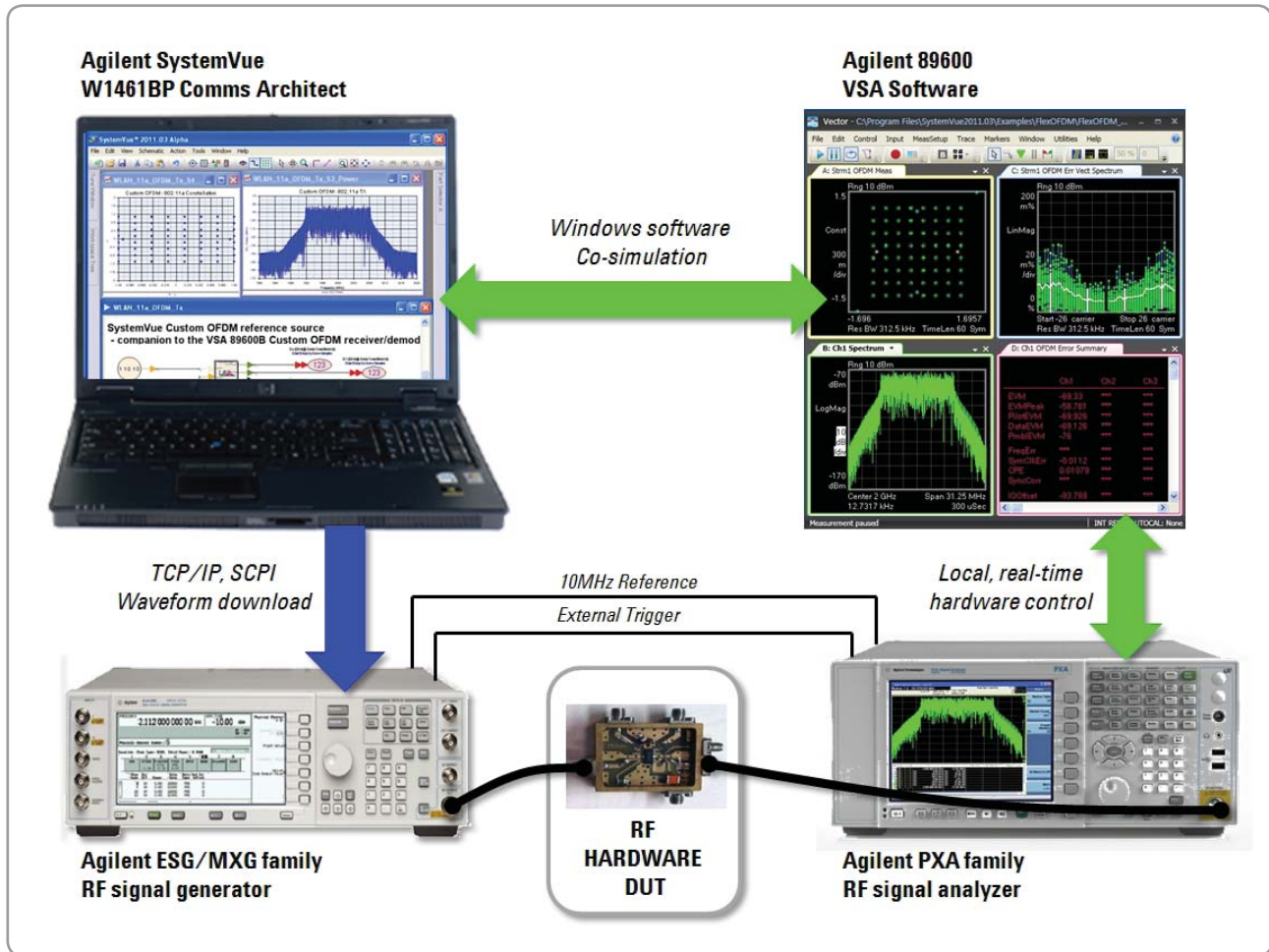


Figure 5. Typical cabling diagram for a basic "Design-Validate-Test" integration, based on SystemVue.

Later, this application note will lead the user step-by-step through some basic connections and tests with SystemVue.

First, however, it is necessary to review the source and sink capabilities of SystemVue.

Creating and Downloading Signals for Test

This section discusses the main parameters for interfacing with instruments and their corresponding functions.

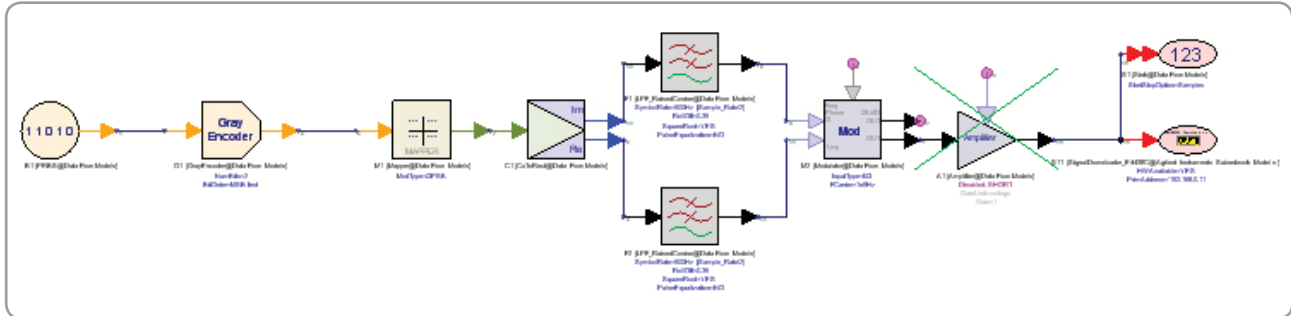


Figure 6. A QPSK signal is created in simulation, then a signal sink (far right) is used to accumulate the waveform samples for download to a signal generator.

Figure 6 shows a basic transmitter design in SystemVue. The two circular items at the right of the schematic are “sinks” where data is collected during simulation. The sink on the bottom right can be used to interface with the Agilent E443nC Series ESG signal generators. It can also be used for Agilent PSG or MXG vector signal generators. This sink model is available in the SystemVue Algorithm Library of the Parts Selector (the right edge of the SystemVue environment) under “Instruments”.

Tip: When using the SystemVue Parts Selector, typing the word “signal” and clicking the Search button (or pushing the Enter key) is a fast way to find the **SignalDownloader** model Without memorizing its name or location.

Figure 7 shows the open dialog box for the signal downloader model.

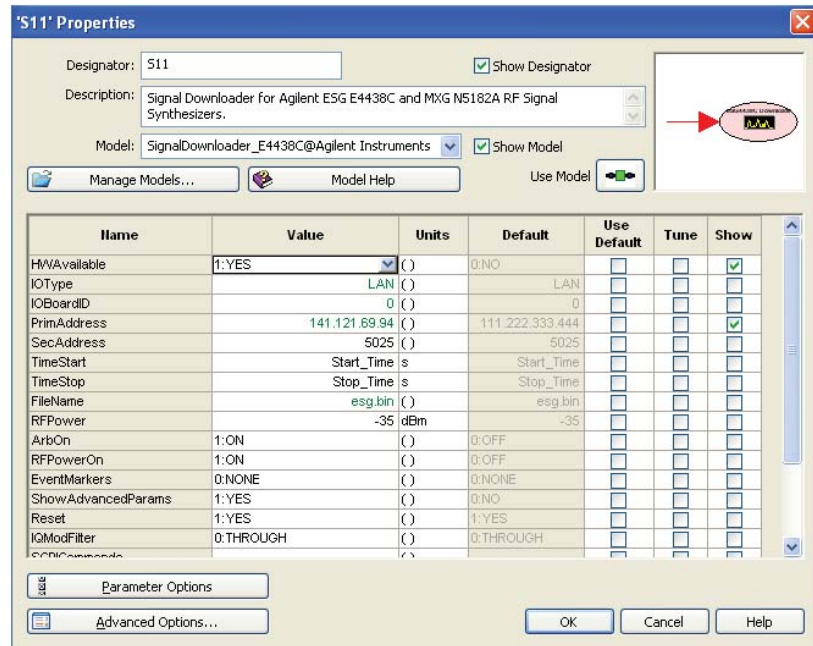


Figure 7. Double-click on the signal downloader sink in the schematic to open this dialog screen, to see more detailed parameters and choices.

Some of the parameters of the open signal download model are:

1. **HW Available.** This parameter has a pull down menu intended for easy toggle between download and non-download. Note that the default is “No”.
2. **IO Type and IO BoardID.** In a LAN connection, the IOBoardID is set to 0. SystemVue instrument connection examples typically assume a LAN connection
3. **PrimAddress and SecAddress.** Here, users enter the IP address (for LAN/TCP/IP) or GPIB address of the signal generator. If the IP address is not known, it can be checked by pressing “Utility,” followed by “LAN Setup” on the ESG signal generator’s front panel. The SecAddress parameter is the LAN port number.
4. **TimeStart and TimeStop.** These required parameters define when to begin and end data collection by the ESG sink. The I and Q inputs for the ESG sink are floating point, so a *TimedToCx* element followed by a *CxToRect* element can convert the timed signal into a floating point rectangular (I and Q) representation. The *TimedToCx* converts a timed signal to a complex I and Q representation on a single output pin, while the *CxToRect* splits the I and Q components into separate pins. Depending on the options, the number of samples collected (Stop-Start+1) must be in the range of 60 to 67,018,496 samples and should be an even number. The last sample is discarded if (Stop-Start+1) is odd. Carefully consider the Start and Stop parameters for the ESG sink. The ESG sink downloads the I and Q samples from SystemVue to the ESG’s arbitrary waveform generator following completion of the SystemVue simulation. These samples are then repeated continuously every 67,018,496th sample to create a periodic signal. Any signal shorter than 67,018,496 samples in duration looks like a “zero-padded” signal as it repeats in the ESG signal generator.

It is useful to select a **Start** value which accounts for any delays in the data being input into the ESG sink. For example, if there is a 100-tap Finite Impulse Response (FIR) filter at the input of the ESG sink, then no signal data will come out of the FIR until 50 samples (one-half of the filter length). The “TimeStart” parameter could then be set to 50, so no data will be gathered until there is valid signal data at its input pins.

The **Stop** parameter can then be selected to download between 60 and 67,018,496 samples. If a signal format with a frame structure like cdma2000, 1xEV or WLAN is being simulated, the Stop parameter can be selected such that an integer number of frames is downloaded into the ESG’s arbitrary waveform generator. To get a better idea of maximum memory, users should first review the Signal Generator Programming Guide (Agilent part# E4400-90505).

5. **FileName.** This parameter specifies the file name for the downloaded waveform. The Arb file in the ESG will be overwritten if repeated simulations are performed. To view or select the downloaded arbitrary waveform data file on the ESG, select **Local > Mode > Arb Waveform Generator > Dual Arb > Select Waveform**. Select the desired waveform and press “Select Waveform.” Next, press “Mod On/Off” and “RF On/Off” to turn on modulation and RF output power.

6. **RF Power.** This parameter is the signal amplitude to be set on the signal generator. (The RF power can also be over-ridden directly on the front panel of the signal generator after downloading, by pressing “Local” then “Amplitude” on the front panel) Regardless of the signal amplitude in the simulation, the instrument will re-normalize the amplitude to this RMS power level.
7. **Arb_On.** If the ARB_On parameter is set to automatic, the ESG starts generating the signal immediately after simulation data is downloaded. If it is set to manual, waveform generation must be turned on at the ESG front panel.
8. **RFPowerOn.** This parameter turns on the RF power. The RF power can also be turned on with the hard key “RF on” by first pressing the local key.
9. **EventMakers.** This parameter controls whether a pulse will be sent out at the instrument's Event1 and Event2 outputs. The pulses coming out of the two outputs have the same characteristics. The pulse rises at the sample indexed by the **MarkerStart** parameter and lasts the number of samples determined by the **MarkerLength** parameter. For example, assume the waveform contains 600 samples (e.g., 600 IQ pairs). If you want a pulse to start at the 1st sample and last 10 sampling periods, simply set the MarkerStart = 1 and MarkerLength = 10.

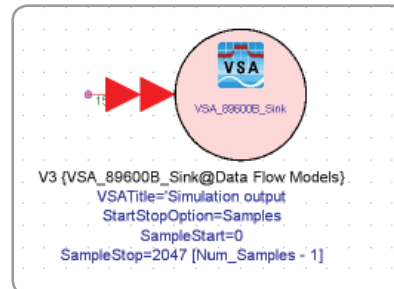
Advanced Parameters. If “YES” is selected, a drop down list of additional parameters is enabled in the GUI. Advanced parameters have options for adding additional SCPI control and other options for configuring the source parameters.

Note that MIMO configuration is also a capability of this signal sink. More details are available for configuration of MIMO applications. These details can be found in the model help menu.

Using the SignalDownloader model is very helpful to provide appropriate signal formatting and fast download for the Agilent signal generators. Understanding the dynamics of how waveforms play in the signal generator still should be understood. Some considerations include, but are not limited to: ALC hold, memory types and appropriate windowing or other techniques to help ensure smooth transitions between waveform segments. These topics are covered more comprehensively in Agilent’s Signal Generation documentation, such as the *Signal Generator Programming Guide(s)*.

Visualizing Results Using the 89600 VSA Software

Another type of sink related to instrumentation is the VSA sink.



In this configuration, you do not necessarily connect to hardware. Instead, you use the sink in simulation to verify and troubleshoot your design using the GUI provided by the 89600 vector signal analyzer software. This capability gives you the same interface, measurement and demodulation algorithms for both design and test, enabling faster troubleshooting of issues during the design and prototype phases. You can also use the VSA sink to record a waveform at virtually any point of your design, play it back in hardware, or use it as a reference versus a real measurement on your Device Under Test (DUT).

To use a VSA sink, a timed complex IQ or Envelope data type must be the input to the model. Some of the key parameters of this model include:

- **SetupFile.** This is an important feature to help expedite the setup time for analysis. Once your analysis is complete and have an open VSA window, you can save the setup file. After that, you open up the model and either type in the path name or browse to it using the browse tab at the bottom of the model GUI. It is also important to note that this setup file can be used as an alternate to entering some of the other properties (such as symbol rate). Reading the help file will provide more details on this topic.
- **Start** and **Stop.** This parameter sets at what point in the simulation the VSA will start and stop its acquisition.
- **ContinuousMode.** This parameter can be used to take data continuously or only a limited amount of data.
- **RecordMode.** This value enables the recording of a .sdf file. This file can be transported to other versions of VSA or replayed at will. As previously mentioned, this is very useful for referencing various designs and implementations. In addition, the recording could be used as a source in other simulations using the VSA **Source** parameter (described later).

For more information, double-click to open the model dialog, and select “Model Help”.

Using Signal Analyzers as Sources in Simulation

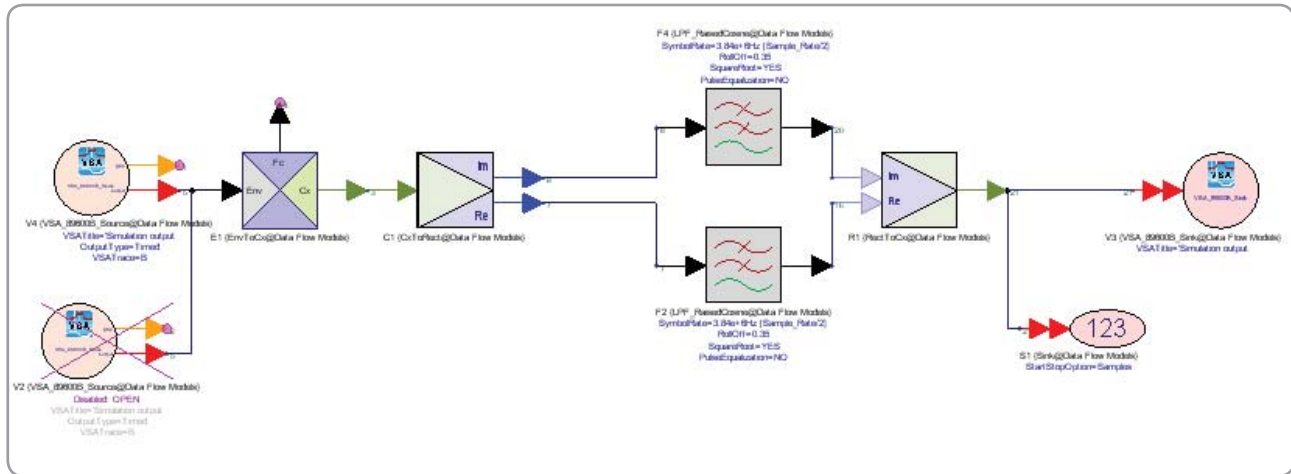


Figure 8. The connected signal analysis capability between SystemVue and Agilent's signal analyzer families is illustrated here.

This section discusses the connected signal analysis capability as shown in Figure 8. This combines powerful signal analysis testing with the flexibility of simulation. Measured signals on the testbench are read into simulation for further analysis and post-processing. With this capability, the 89600 software (version 7.0 or later with Options 105 and 300) simulation source element reads measured IQ data from the X-Series signal analyzer, E4440A PSA or the Infiniium oscilloscope on the testbench, back into the SystemVue simulation environment for further analysis in simulation.

The next section discusses setting up the 89600 software simulation source to read measured signals back into simulation.

Using the 89600 VSA to Capture a Waveform

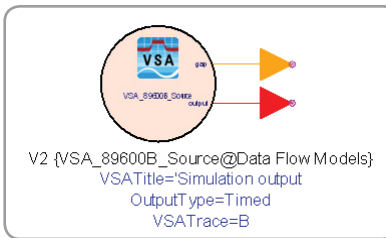


Figure 9. This model shows the 89600 software configured as a SystemVue simulation source, injecting either recorded or live measurement waveforms into the simulation as a stimulus.

As shown in Figure 9, the 89600 software simulation source has various parameters, including:

1. **VSATitle.** This parameter defines the text labeling to appear at the top of the measurement display when the simulation is launched.
2. **ControlSimulation.** The simulation duration in the SystemVue DSP schematic window can be controlled from sources or sinks in the schematic window. This parameter defines how long the signal source produces data and whether the source will control the simulation. When this parameter is set to “NO,” the signal source keeps the simulation running long enough to cover one pass through the data. When this parameter is set to “YES,” the signal source runs for the duration defined by the sink(s) controlling the simulation duration.
3. **Output Type.** The DSP schematic window supports a variety of different signal types. The Timed signal format is most typically used for RF signals represented with a modulated carrier frequency. The various signal types include:
 - *Timed*—Timed data is output at the 89600 VSA center frequency and time step (see TStep below). The data is complex or baseband, depending on the 89600 VSA’s zoom state and requires time-domain measurements from the 89600. An example 89600 trace data type is Main Time.
 - *Frequency*—Spectrum data is output as pairs of complex numbers. The first number is the real frequency (the imaginary part is zero). The second number is the complex voltage at that frequency. You can connect a Distributor2 to the VSA89600Source data port to separate the signal. This requires frequency-domain measurements from the 89600 VSA. An example 89600 trace data type is Spectrum.
 - *Demod Errors*—Demodulation error data is output as sets offloading point values. The number of values and required 89600 VSA configuration varies with demodulator type.
 - *Complex Scalar*—With this type of signal, complex numbers are output. If the 89600 VSA measurement data is real-valued, the imaginary part of the output values is zero. An example 89600 VSA trace data type is Error Vector Time in Digital Demod mode.

- *Float Scalar*—With this type of signal, floating point numbers are output. This requires real-valued 89600 VSA measurement data. An example 89600 trace data type is IQ Mag Error in Digital Demod mode.
 - *Integer Scalar*—With this type of signal, integer numbers are output. This requires real-valued 89600 VSA measurement data. It is useful for sourcing demodulation symbols when the 89600 VSA tracedata type is Syms/Errs.
4. **Pause.** The **RecordingFile** parameter can be used to automatically recall a recording into the 89600 VSA during simulation start-up. If “Pause” is set to YES, the setup file and recording files are loaded before the “Pause” dialog is displayed.
 5. **VSATrace.** This parameter defines which trace will provide measurement data.
 6. **RepeatData.** This parameter defines data that is transferred into SystemVue from the hardware-based measurement. “Single Pass” supplies data from a single measurement. “Repeat” acquires a single measurement and repetitively sources it into the simulation. “Re-acquire” repeatedly acquires and sources new measurements. The data across measurements in the “Repeat” and “Re-acquire” modes are not time continuous.
 7. **TStep.** This parameter specifies the time step associated with the simulation source data being output. It should generally be set to the same time step used for SystemVue simulation. A value of 0 seconds produces data with the time step set by the 89600 VSA, based on the frequency span selected. When reading from a recorded file (effectively downsampling the data if set to a larger value), this parameter should be set to a value that is equal to or larger than the time step associated with a recorded file.
 8. **SetupFile.** This parameter can be used to automatically recall an 89600 VSA setup file during the simulation start-up and is useful for setting up the 89600 VSA configuration when repetitively performing simulation. The setup file can be created by configuring the 89600 VSA software simulation source as desired, selecting **File > Save > SaveSetup** on the menu bar and then specifying the name of the *.set file to be saved in the /data subdirectory of the SystemVue project. Setup files in locations other than the /data subdirectory can also be specified by providing the relative path (e.g., ../.././ filename.setup) or absolute path (e.g., C:/my_dir/filename.setup) of the setup file. The **SetupUse** parameter defines how the setup file is to be recalled. If Pause is set to YES, the setup file is loaded before the Pause dialog is displayed.

Note: the latest versions of the 89600 VSA software can read both “.set” and a newer file format “.setx” that supports newer features and window formats. The newer format is not readable by older versions of the VSA software.

9. **RecordingFile.** This parameter specifies whether new measurement hardware data should be gathered or if previously recorded data can be used as the simulation signal source. If no recording file is specified, then measurement hardware data is gathered. Recorded (.sdf) files are useful for performing repetitive simulations when the measurement hardware data is not expected to change from one simulation to the next, or when it is desirable to perform simulation without the presence of measurement hardware. If Pause is set to YES, the recording file is loaded before the Pause dialog is displayed.
10. **SetupUse.** This parameter defines when an 89600 VSA setup file (specified in the **SetupFile** parameter) will be recalled. Options are:
 - *always* — recall setup file on every simulation run;
 - *once* — recall setup file only when 89600 is started; and
 - *no* — do not recall setup file.
11. **AutoCapture.** Setting this parameter to YES will automatically initiate a capture recording of the hardware input data when the simulation is launched. The model begins sourcing data into the simulation when the capture is complete. Time-based measurements can be continuous, except for wrapped values when a playback loop occurs. The recording length can be controlled via an 89600 VSA setup file.
12. **DefaultHardware.** If this parameter is set to YES, the 89600 VSA automatically selects an appropriate hardware configuration. Otherwise, the 89600 VSA will use previously selected hardware settings. Note that the 89600 VSA software must first be installed before selecting signal analyzer hardware.

More information regarding the 89600 VSA software installation process and how to configure it for signal analyzer hardware is available at www.agilent.com/find/89600. Simply select the applicable product category and then Manuals, Guides, and Service Notes. When the 89600 VSA software is started (**Start > Programs > Agilent 89600 VSA > Vector Signal Analyzer**), it retrieves supported hardware information from the resource manager table and determines the appropriate installed hardware configuration. The 89600 VSA's Select Hardware dialog box (**Utilities > Hardware**) displays this information. The devices are grouped and listed by their hardware type, as indicated by the tab header. The information is tailored to the specific device and includes address, location and name (or description).

SystemVue's Math Language and Alternate Ways to Connect .m Files

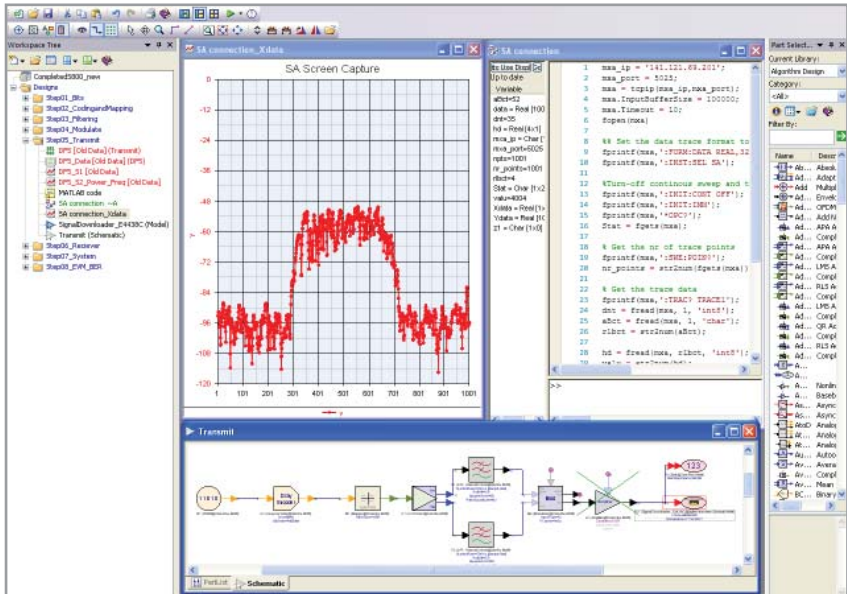


Figure 10. A signal trace captured manually from an Agilent MXA signal analyzer, using .m code.

SystemVue's built-in math language capability offers another option to instrument connectivity. Using a TCP/IP object, SystemVue can connect to instrumentation and send SCPI commands and/or send/receive data (controlled using ascii .m scripting files). The example in Figure 10 shows a trace capture from an Agilent MXA using .m code (as seen on the right side of the screen). Right-clicking in the workspace tree and selecting **Add -> Equation** opens an interface that enables you to write math language code and review applicable data. In addition, the ESG downloader model used in a previous section is based on a traditional SystemVue sink with .m files in the equation editor. You have the option of creating more logic in your download model. For more details, see the Signal Download model help.

Another complementary capability is to use MATLAB directly from within SystemVue, in conjunction with SystemVue's own math language parser. Using MATLAB with SystemVue's instrument control "toolbox" offers an expanded set of instrument interfacing capability. SystemVue provides examples of IQ and screen trace data acquisition (from a spectrum analyzer) using SystemVue's native math language capability. Direct MATLAB integration into SystemVue also provides additional plotting capability.

Waveform File Generation and Wideband Arbitrary Waveform Generation

SystemVue offers an arbitrary waveform generator (AWG) file output model that allows designers to output waveform files to use with their Agilent arbs for wideband analysis and signal generation (Figure 11). SystemVue can generate a multitude of separate files that can be sequenced in a coordinated test plan, or used simply as an available library of known test vectors.

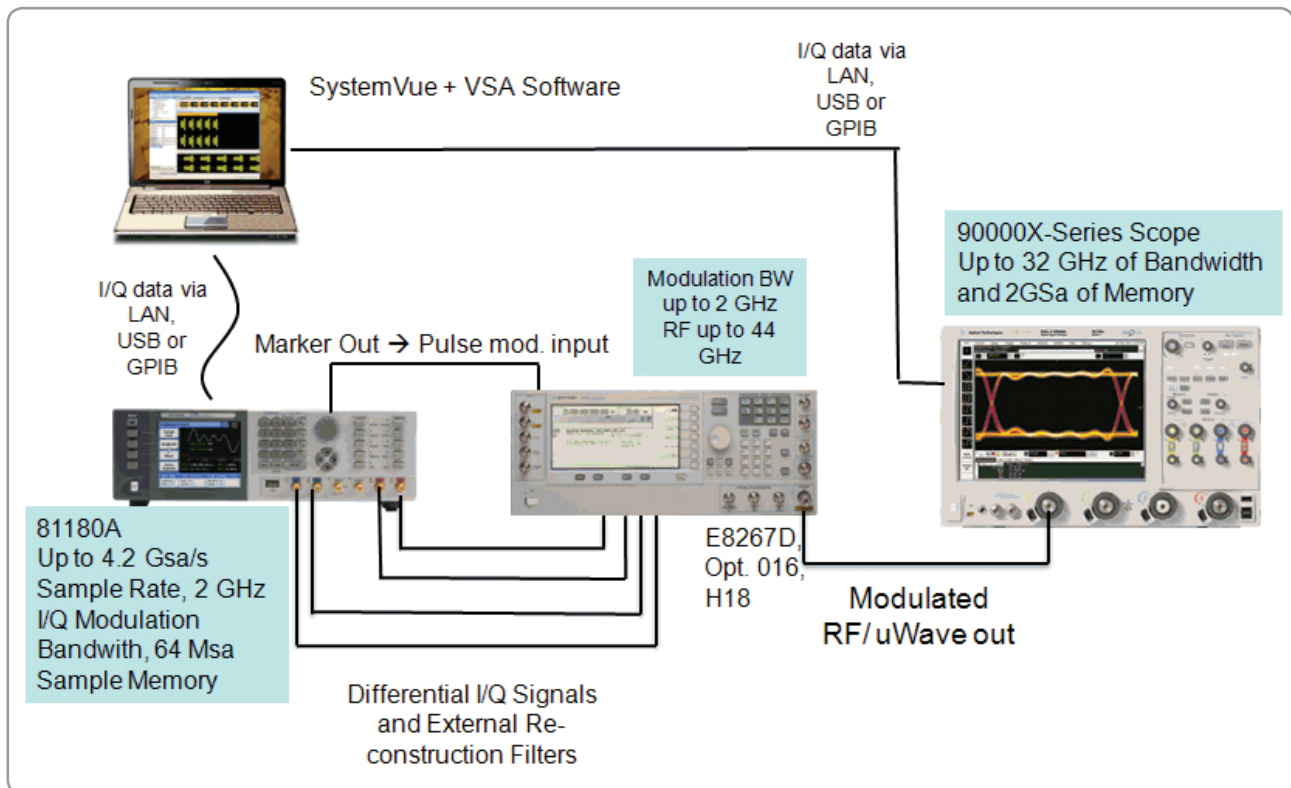


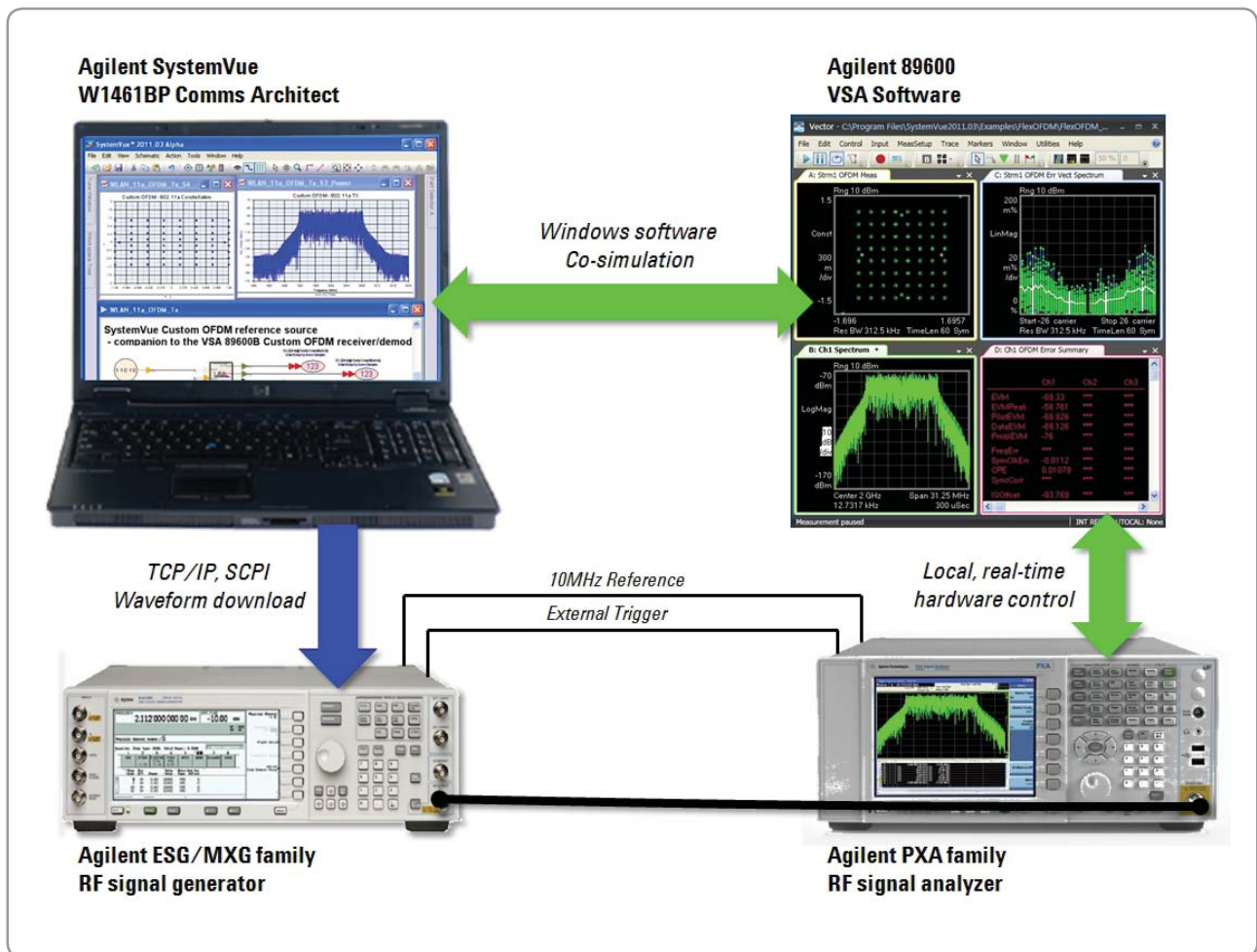
Figure 11. SystemVue's AWG files can be used with Agilent's multi-GHz wideband arbs for both signal generation and analysis in emerging comms and aerospace/defense.

Generated AWG files are easily integrated into the software used for the instrument, since they have proper header properties and extensions. In some cases, the AWG calibration is also contained inside SystemVue. More information regarding this topic is available in the SystemVue documentation, as well as in other Agilent application notes.

Hands-on Exercises

This short lab is intended to give the reader a better feel for how to integrate instrumentation with SystemVue. It is a continuation of a “Getting Started” tutorial for SystemVue. Only a minimal amount of understanding is required to engage these exercises. To run them, the Agilent 89600 VSA software must already be installed on the target, your PC. You will also need access to an Agilent signal generator and signal analyzer. Please consult the configuration list in the Appendix for supported hardware. This set-up was tested with the MXA and the MXG.

To begin, open the SystemVue 2011 example called “Instruments\Integrating_SV_with_instruments”, and open the “Example1_Transmit” folder. If you do not have this example, you may also visit <http://www.agilent.com/find/eesof-systemvue-dvt> for additional links and resources.



Example 1: Transmitter Subcircuit

First, open the “Transmit” schematic, as shown in Figure 12. The primary purpose of this example is to show how to download a simple waveform to an Agilent signal generator using the *SignalDownloaderModel*. It will also review how the 89600 VSA software sink works in the SystemVue environment.

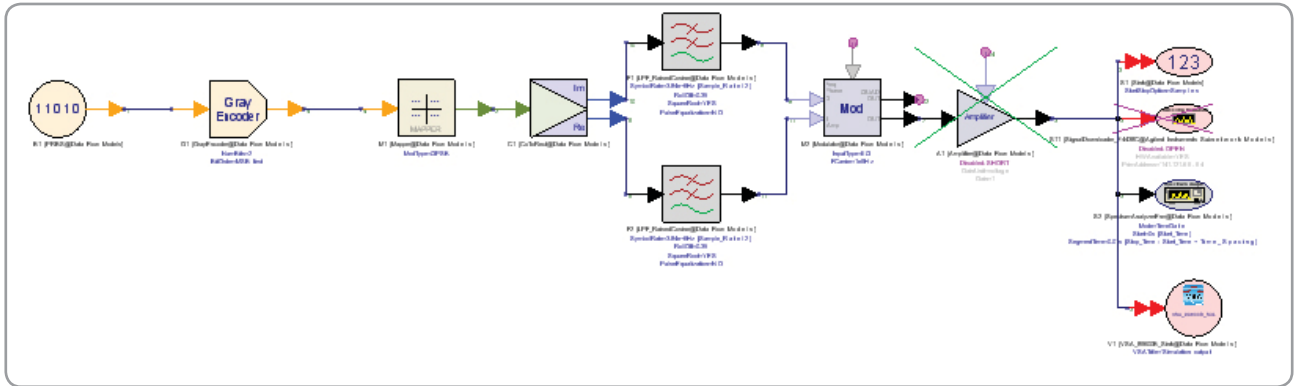


Figure 12. A typical QPSK transmitter circuit is used to create the signal.

First, simulate the circuit and view the spectrum by clicking the DF1_S2_Power_Freq graph shown in Figure 13.

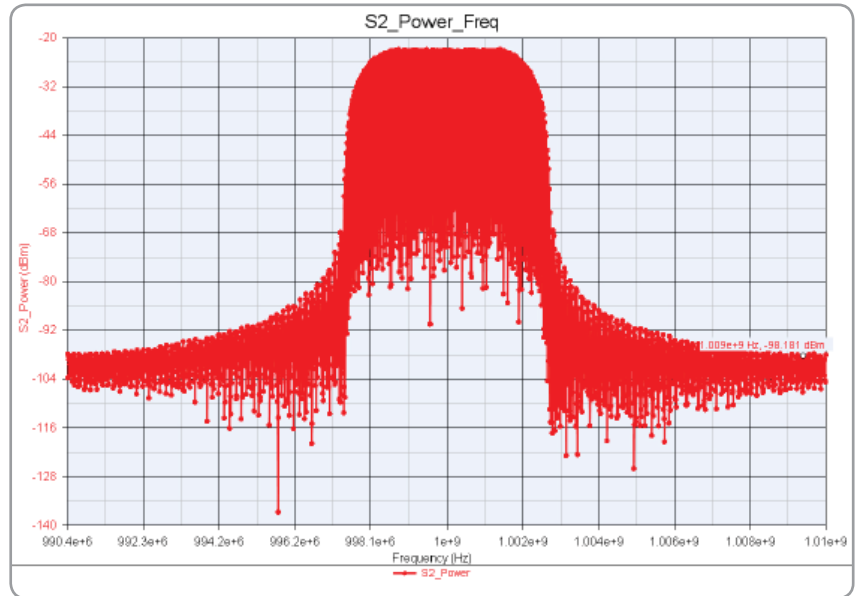


Figure 13. Shown here is the QPSK spectrum at the output sink..

To show the capability of generating “real world” signals we introduce impairments in the modulator or demodulator and study the effect on EVM and BER. For example, you can introduce gain imbalance, phase imbalance, IQ rotation, IQ origin offset, or other parameters in the modulator as shown in Figure 14. You can do this by clicking on this icon and editing as shown in Figure 15.

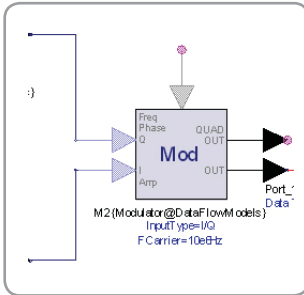



Figure 14. Impairments can be introduced in the modulator or demodulator.

Name	Value	Units	Default	Use Default	Tune	Show
InputType	0:IQ	()	0:IQ	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
FCarrier	220e6	Hz	0.2e6	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
InitialPhase	0	deg	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AmpSensitivity	1	()	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ConjugatedQuadrature	0:NO	()	0:NO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MirrorSignal	0:NO	()	0:NO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ShowIQ_Impairments	1:YES	()	0:NO	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
GainImbalance	0.0	()	0.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PhaseImbalance	0.0	deg	0.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I_OriginOffset	0.0	()	0.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q_OriginOffset	0.0	()	0.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IQ_Rotation	0.0	deg	0.0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 15. To introduce impairments, simply edit the highlighted parameters change the values.

Feel free to simulate and re-plot the graphs. Next, select the amplifier and enable this part by clicking the short button  (Figure 16). Edit the impairments.

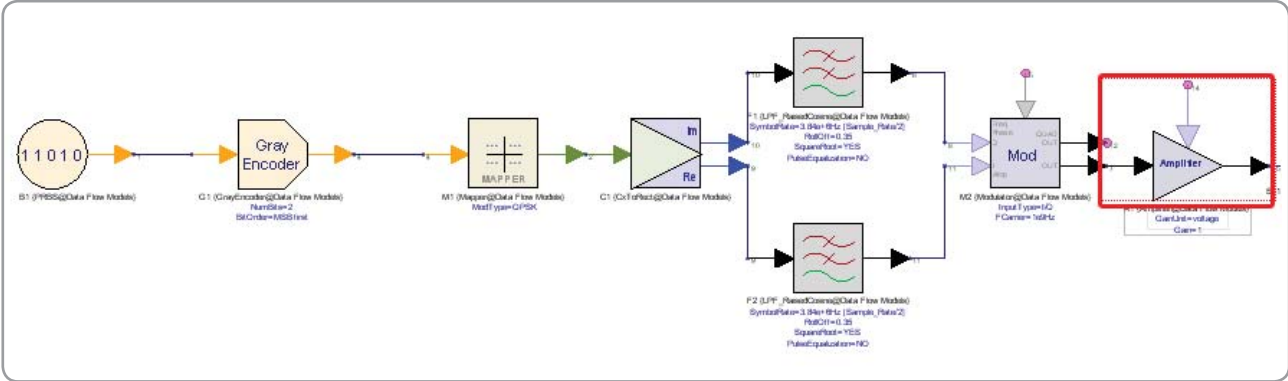


Figure 16. The output amplifier is selected in the transmitter circuit.

In the example in Figure 17, a significant amount of distortion has been introduced. Run the simulation and review your plot (Figure 18). Can you see that distortion was added?

'A1' Properties

Designator: A1 Show Designator

Description: Nonlinear Amplifier with Noise Figure

Model: Amplifier@Data Flow Models Show Model

Manage Models... Model Help Use Model

Name	Value	Units	Default	Use Default	Tune	Show
GainUnit	1: dB	()	0: voltage	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Gain	1	()	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NoiseFigure	10	()	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
GCType	3: TOI+dBc1	▼	0: none	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TOIout	40	dBm	0.1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dBc1out	10	dBm	0.01	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ReIR	50	Ohm	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 17. Distortion is introduced using the menu shown above.

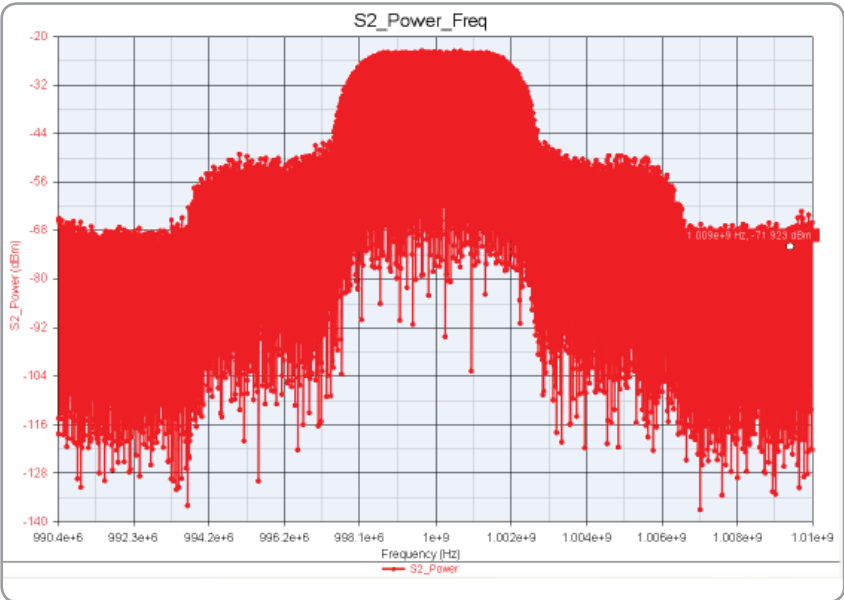


Figure 18. The updated simulation result confirms that nonlinear distortion has been added.

In Figures 19 and 20, we compare simulated signal data versus data measured using the VSA software (89601B).

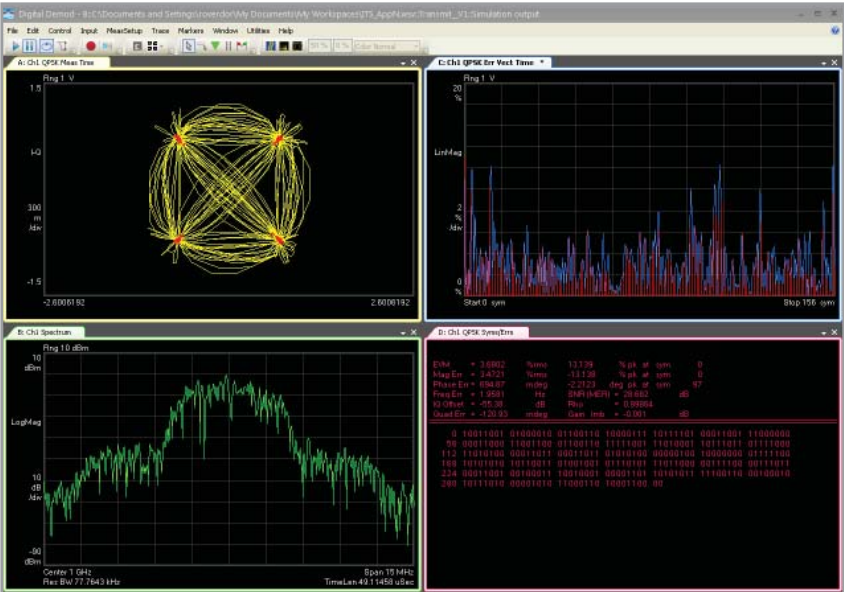


Figure 19. This graph shows simulated signal data. Compare the EVM vs. the hardware measurement in Figure 20.

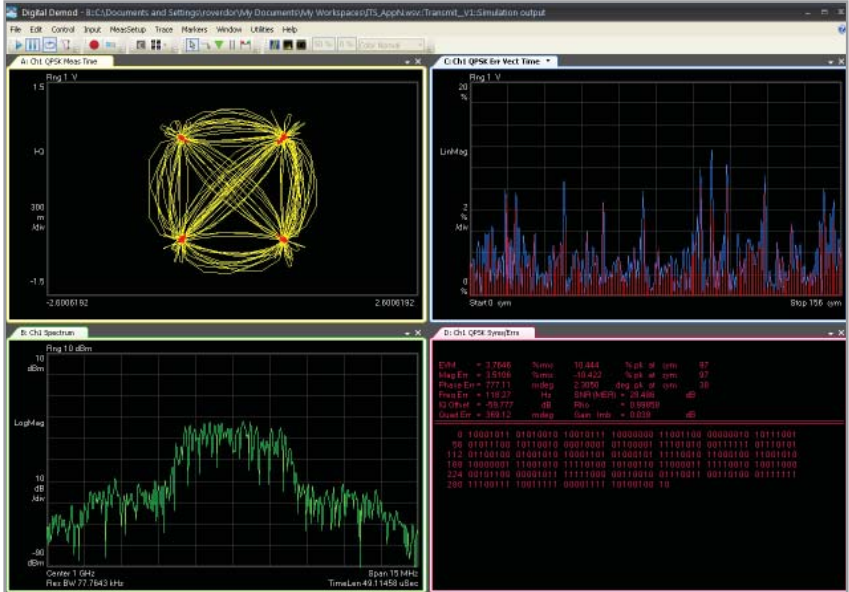


Figure 20. Shown here is the corresponding hardware measurement, using an Agilent MXG source and a PSA analyzer.

As you can see by comparing Figures 19 and 20, the measured EVM in Figure 20 is slightly higher than the simulation. This is expected because the instrumentation hardware degrades the signal slightly.

Using the Signal Generator

To use the signal generator, first verify that it is connected to the LAN interface of your PC.

1. Select the "Run..." icon under the Windows "Start" menu and type "cmd" in the Open box.
2. When the window appears, type "ping" and then the signal generator's IP address.

If there is a reply, you should be able to perform the download from SystemVue. The signal downloader model is shown in Figure 21 (highlighted portion of the image). Double click on this parameter to open its dialog box.

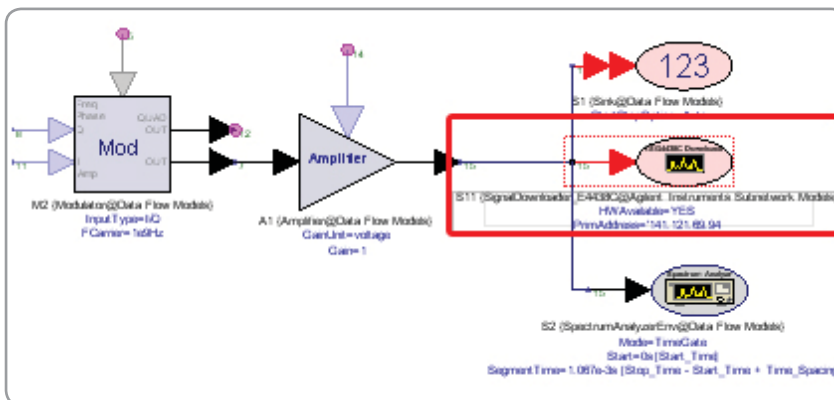


Figure 21. Shown here is the signal download model (highlighted, middle sink on the right).

Edit the model as shown in Figure 22, and enter the IP address of your signal generator in the *PrimAddress* parameter.

Name	Value	Units	Default	Use Default	Tune	Show
HWAvailable	1: YES	()	0: NO	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
IOType	LAN	()	LAN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IOBoardID	0	()	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PrimAddress	141.121.69.94	()	111.222.333.444	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
SecAddress	5025	()	5025	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TimeStart	30*Time_Spacing	us	Start_Time	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TimeStop	Stop_Time	s	Stop_Time	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FileName	esg.bin	()	esg.bin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RFPower	-35	dBm	-35	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ArbOn	1: ON	()	0: OFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RFPowerOn	1: ON	()	0: OFF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EventMarkers	1: EVENT1	()	0: NONE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MarkerStart	1	()	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MarkerLength	10	()	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ShowAdvancedParams	0: NO	()	0: NO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 22. Double-click the signal downloader model to set up the parameters.

On your first connection, you should set any power levels and simulated impairments to their nominal settings so that a cleaner signal is analyzed, making it easier to troubleshoot your configuration. Once you have finished editing parameters, close the model's GUI and press the simulate button. When you have confirmed that the signal has been successfully downloaded, you can further adjust your settings and introduce simulated impairments. Simply re-simulate to download a new waveform.

Using the VSA Sink

By clicking the “open” button to enable the VSA Sink model, you can record the signal (at the node where you would add the ESG model). You can find this by typing VSA into the “Filter By:” box in the Part Selector. Make sure the current library is Algorithm Design. For many of these demonstrations, we show screen shots using the older version of the 89600 VSA software (release 12 or earlier, or version “A”), but you can also use the 89600 “B” version as well (for version 13 or later). The model now defaults to the B version. You can select the appropriate version of the VSA software using the model pull-down menu as shown in Figure 23.

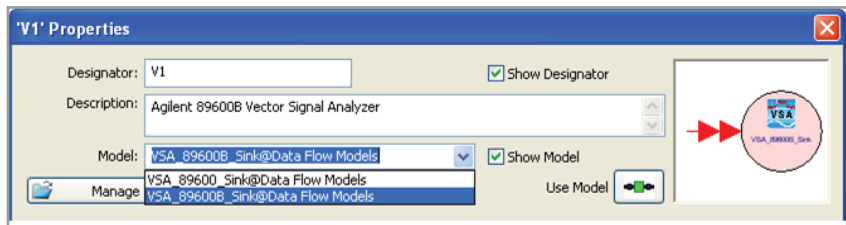


Figure 23. For the “A” version of VSA, select the “89600_Sink” in the model pull down menu.

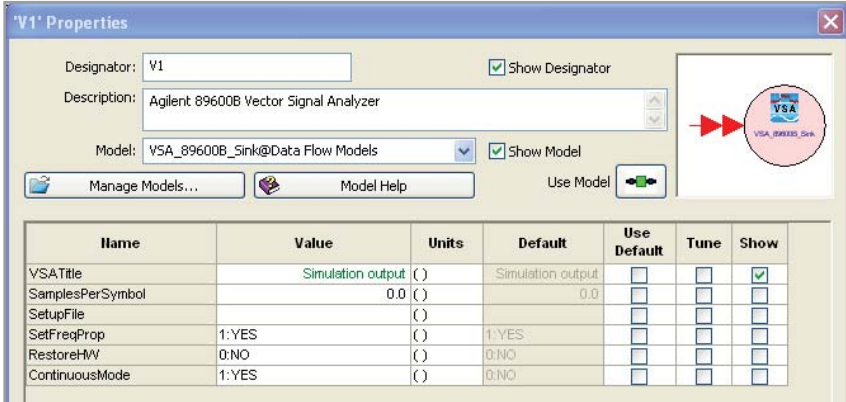


Figure 24. The model of this VSA sink has been successfully changed to the VSA “B” model. Note that the “SetupFile” parameter is currently blank.

Example 1: Transmitter Subcircuit

When you run the simulation with defaulted VSA settings (i.e. –file .set or .setx no setup), you should see the image in Figure 25. Next we will configure the measurement, and save the set-up file (see Figure 24).

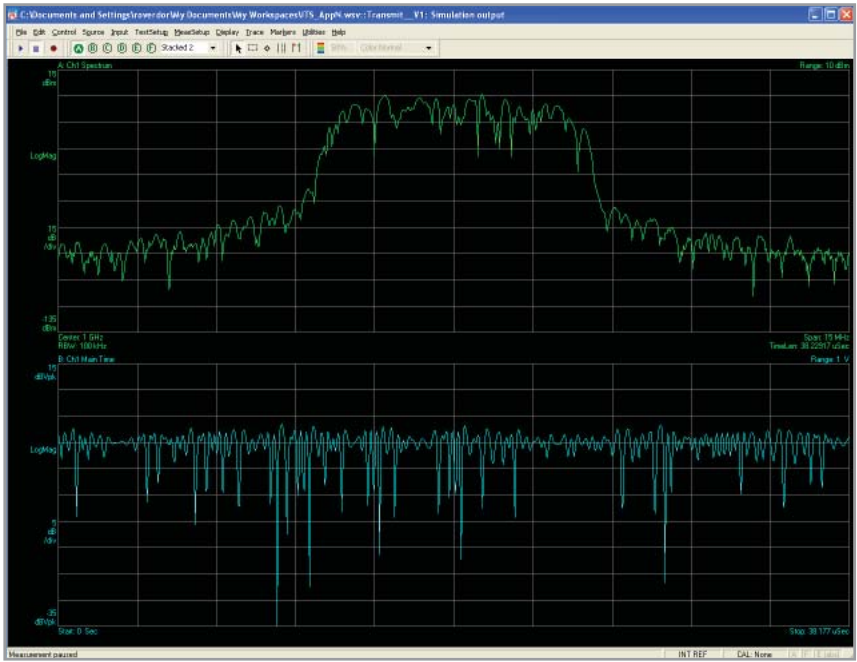


Figure 25. This waveform results from simulation with no setup file.

Next, let's configure the VSA's digital demodulation to analyze the QPSK waveform. Under the VSA Demod Properties, set the parameters to QPSK at 3.84 MHz Symbol Rate, with 5 points/symbol, as shown in Figure 26.

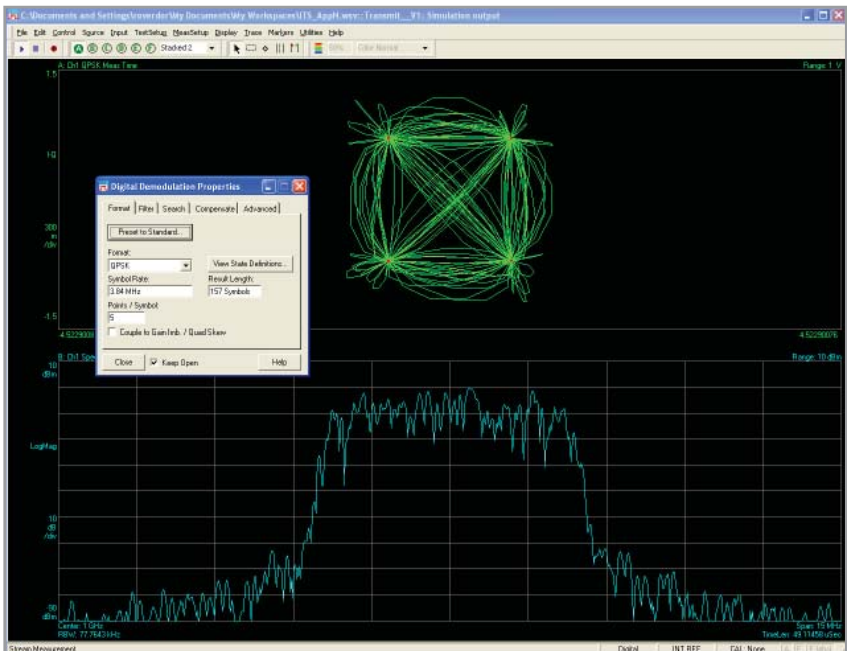


Figure 26. To analyze the QPSK waveform, you must first configure the measurement.

Now, save the setup file. (We saved ours as "ITS.set.") Save the file by clicking **File -> Save -> Setup**. Once this setup file has been saved, return to the SystemVue VSA sink parameters (Figure 24), and click the cursor on the value in the "SetupFile" parameter. Now click the "Browse" button to find the location of the new setup file. Change it to "ITS.set", and press OK.

The result should be similar to Figure 27.

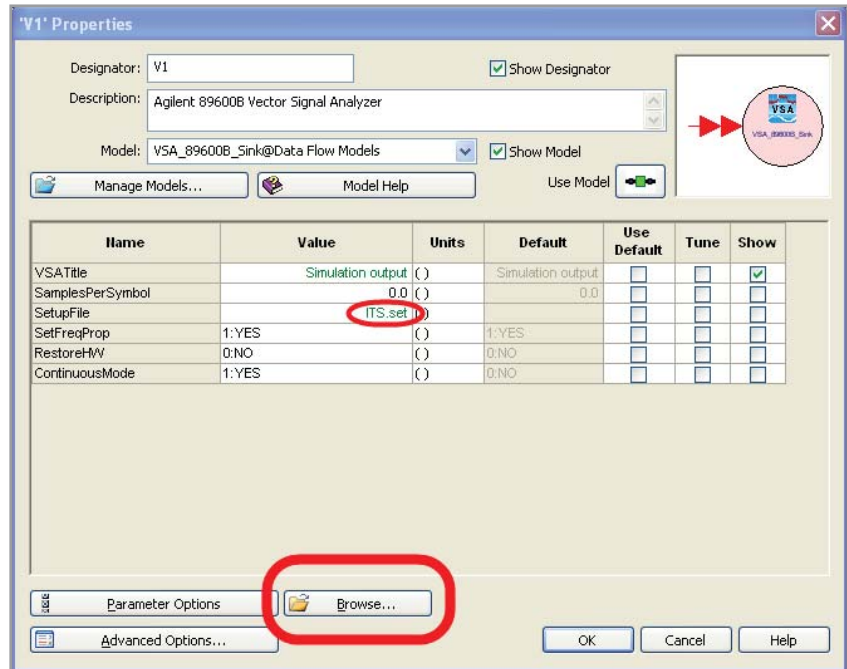


Figure 27. Use this menu to browse to the saved file.

Setting up a Recording

To set up a recording, follow these steps:

1. In the VSA, go to **Input -> Recording -> Acquire** 16 records.
2. Now, configure the triggering. Do **Input -> Trigger -> Type External**.
3. You should see the following: **Control -> Record**.
4. **File -> Save -> Save Recording**: Save the recording as (in the workspace file you are working from): "ITS_rec.sdf".

Tip: If you are making many small adjustments and re-simulating, allow your VSA session to continue to run. It will re-simulate very quickly. When you are done with your session, return to the SystemVue menus and perform "Action/Exit Vector Signal Analyzer" (figure 28). This is a clean way of ending any VSA processes running in Windows, seen or unseen.

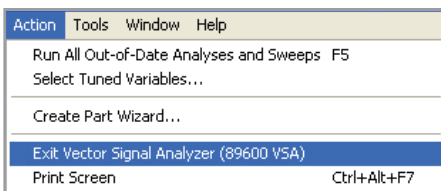


Figure 28. Exit the VSA by selecting "Action/Exit Vector Signal Analyzer" from SystemVue.

Using Math Language to Acquire Trace Data from a Signal Analyzer

Both “math language” (SystemVue’s native capability) and MATLAB can be used to capture trace data out of an analyzer, and plot results. To obtain the math script for this section, you will need to recall the completed example workspace.

To check to see if the signal was downloaded correctly, use the “Autotune” feature of your X-Series signal analyzer to find the signal.

Now, run the math language script shown in Figure 29. Note that the script uses MATLAB to plot the results (the added script is in the MATLAB code note in the Example1_Transmit folder). To only use the native math language parser, run the script and plot the results using the X versus Y plotter in SystemVue (right click on the Variable data). The X-axis data is “Xdata”, while the Y-axis is “Ydata”. This is one way to capture trace data out of the analyzer manually, without requiring a full license for the VSA software.

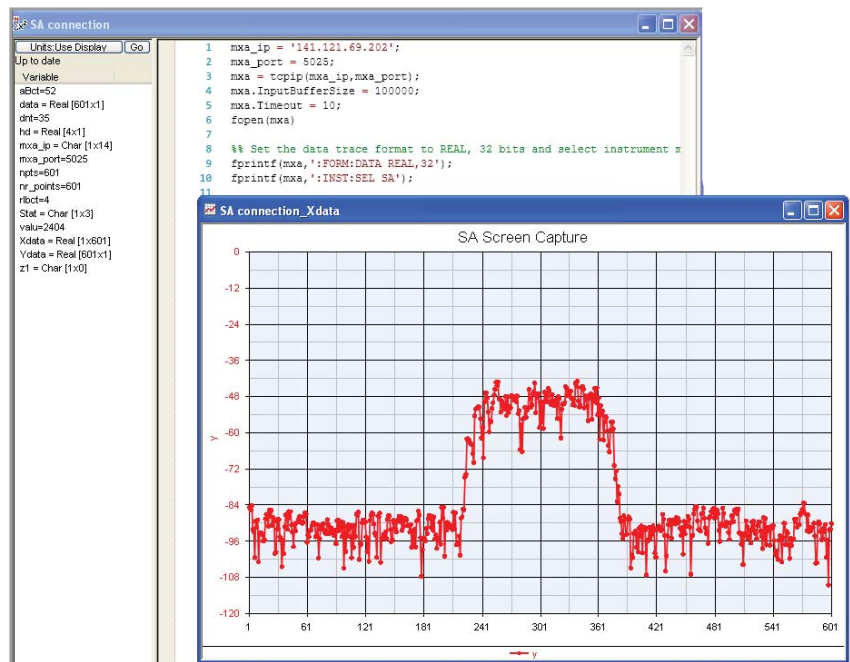


Figure 29. Run this math language script to see if the download was done correctly.

As we did in the previous section, it is worthwhile to compare the captured trace to the original simulation result, as a sanity check. As you can see in Figure 30, we have virtually the same spectrum. The differences largely stem from how many samples we have taken from each. You can further enhance the X and Y axis notation directly in the .m code in the “custom equation” area of the plot window.

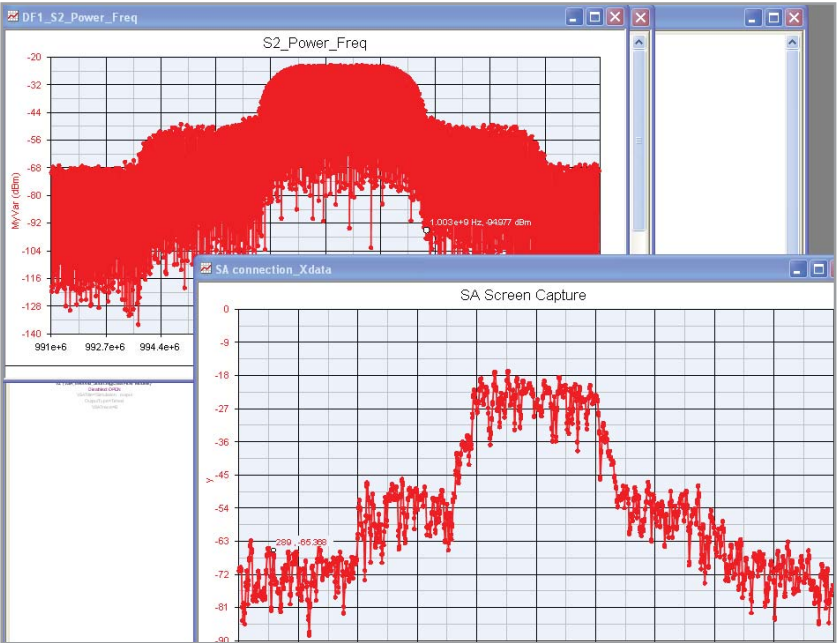


Figure 30. These graphics compare the simulated (upper left) versus measured results of the impaired signal.

Before completing this section, please close all of the plots, datasets and schematic windows that may be open, and save your work.

Example 2: Receiver Subcircuit

In this example, we use both a VSA source and VSA sink in the same design. The VSA instances on the left of the schematic take either a live or recorded waveform and inject it into the system. The complex envelope is then digitally filtered in SystemVue, and sent to another simulation-only VSA for visualization.

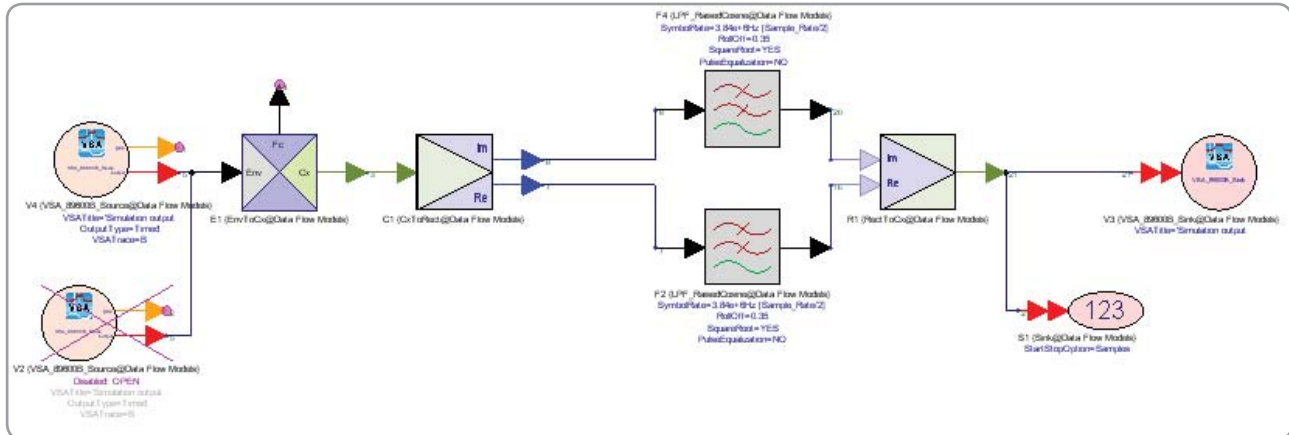


Figure 31. Two VSA parts are used in this schematic, one to play back a recorded waveform, and another capture a live waveform from a signal analyzer.

In Figure 31, two VSA sources are available, with the lower one being de-activated. The top source (currently in “open” state) is used to acquire data directly from a live signal analyzer. The bottom source (show de-activated) allows us to use the recording we made earlier.

Case 1: Live Signal Analyzer is Available

For those with a signal analyzer, please enable the top VSA source, and disable the lower VSA source, as shown in Figure 32.

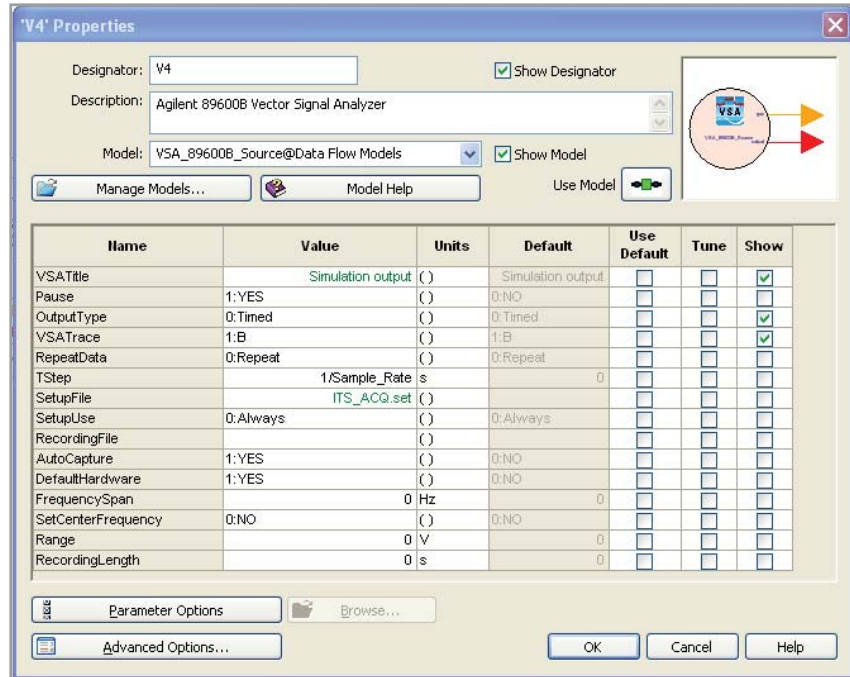


Figure 32. Select the top VSA source and click on the “Use Model” button in the upper right until it is activated (as shown). Also make sure the other VSA source is deactivated.

Setting up the acquisition from hardware is very similar to the recording we did of the simulated data. The record length and triggering needs to be set up along with the Sample Rate settings to ensure the simulation is successful. In the VSA source parameter, the PAUSE button is set to yes. The simulation will pause and a window will open that asks when you are ready for the acquisition from the signal analyzer. Note that this is a small window and may be hidden behind other windows. Look for it on your Windows taskbar, or toggle to it using <alt><tab> on your Windows keyboard.

If the VSA screen looks similar to what is shown in Figure 33, then click OK. The data capture will then complete, and the regular SystemVue simulation will begin, using that signal.

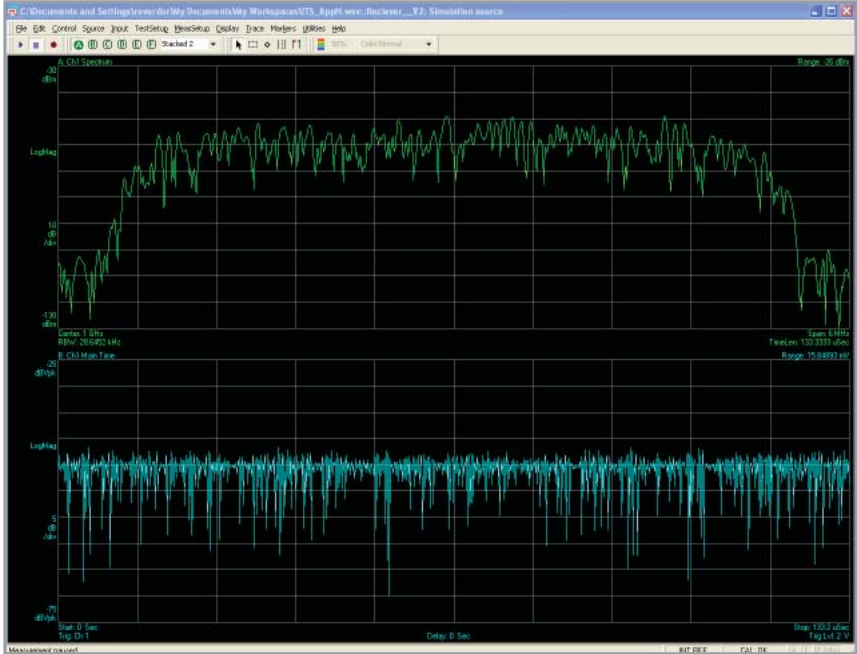


Figure 33. This VSA screen shows the acquisition of data from hardware.

Case 2: Live Signal Analyzer is Not Available

If you do not have access to a suitable Agilent signal analyzer, then use the recorded waveform that has been provided. De-activate the top model to an “open,” then re-activate the bottom model. For this model, you will not want to use the default hardware settings or a specified setup file. Instead, use the “Browse” tab at the bottom of the VSA model window (shown in figure 34) to select the .sdf file you recorded earlier.

The VSA sink parameters should be the same as the transmit example.

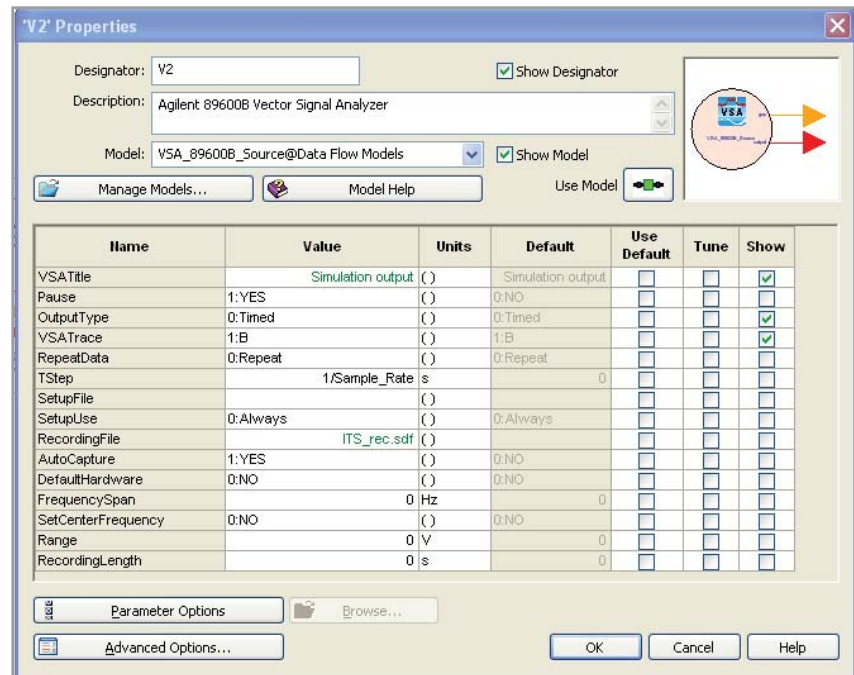


Figure 34. Set up the VSA sink parameters the same as you did for the transmit example.

In both cases, you should see what is shown in Figure 35.

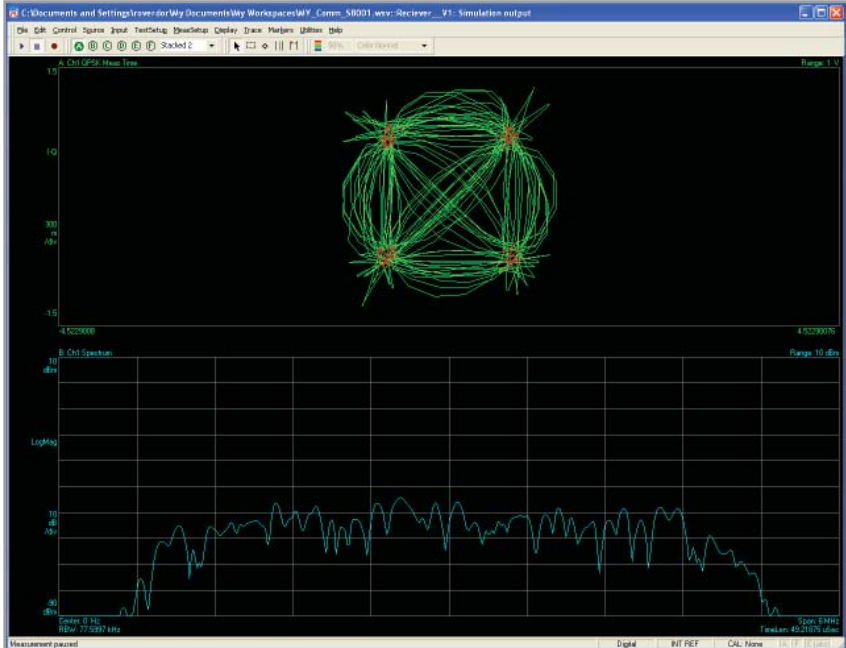


Figure 35. Note that the IQ constellation of the captured waveform is “noisy”. Common issues include filtering that has been performed twice, a mismatch in filter parameters such as RRC “alpha”, and samples per symbol..

The IQ constellation should look noisy. You may remember we have already applied the measurement filter in SystemVue to this signal. The VSA applies an additional raised cosine (RC) filter. If enabled, this extra filtering can be removed by selecting **Meas Setup -> demod properties -> Filter -> Measurement filter -> Off**.

Conclusion

Using the information in this application note, you have now successfully created a modulated RF waveform and downloaded it to an Agilent signal generator. Then, you pulled a measured waveform out of an Agilent signal analyzer and used it in a simulation. With these basic stimulus/response skills, you are ready to begin building more sophisticated verification strategies for the modeling environment, as well as for hardware verification.

Appendix - Configurations

Instrument category	Agilent family name	Model numbers	Frequency range	Connectivity (note 1)	Required options	Recommended options
RF signal generators	ESG	E443*C	250 kHz – 6 GHz	1,2	501, 502, 503, 504, or 506	• 506/UNJ, 602, 1E5, UNB
	PSG	E8267D	250 kHz – 44 GHz	1,2	520, 532, or 544	• UNX/UNY, 1EH, H18/016, 602
	MXG	N5182A N5162A	100 kHz – 6 GHz	1,2	501, 503, or 506	• UNV, UNZ, 012, 019, 1EA, 1EL
RF signal analyzers	PSA	E4440A, E4443A, E4445A, E4446A, E4447A, E4448A	3 Hz – 50 GHz	1,2,3	B7J, 122, or 140	• 122 (80 MHz analysis BW) • 123 (pre-selector bypass) • 110 (internal preamp) • 89600 VSA
	PXA	N9030A	3 Hz – 26.5 GHz	1,2,3	503, 508, 513 or 526	• B1X (140 MHz analysis BW) • P03, P08, P13, or P26 preamps • 89600 VSA Software
	MXA / EXA	N9020A, N9010A	20 Hz/9 kHz – 26.5 GHz	1,2,3	503, 508, 513, or 526	• B25 (25 MHz analysis bandwidth) • P03, P08, P13, or P26 (internal pre-amplifier) • 89600 VSA
Logic analyzers		16900	BB	1,2,3	001, 004, 016, 032, 064	064 with 64 Msa/sec

Table 1. Supported General-Purpose Measurement Instruments

Note 1:

SystemVue connects to this instrument via these connectivity methods:

1. File I/O
2. SCPI commands over TCP/IP, using the Agilent VISA 15.5 drivers, or later
3. 89600 VSA, option 300 physical connectivity engine
4. IVI-com and IVI-C drivers

Software category	Agilent family name	Model numbers	Connectivity to SystemVue	Required options	Recommended options
Vector Signal Analyzer software	VSA	89600A	Live co-simulation	• 105 (connect to SV)	• AYA (digital demod)
		89600B		• 200(Basic Vector Signal Analysis)	• BHF (custom OFDM)
			SDF file I/O	• 300 (Hardware Connectivity)	• Any format options
FlexDCA software	Flex DCA	N1010A	Co-simulation only	• Basic environment runs for free with SystemVue	• Jitter • Sim
Signal Studio software	SS	N76xx series	Encrypted WFM file I/O only	• N7624/25 requires SystemVue W1910 • N7615 requires SystemVue W1911	

Table 2. Supported Measurement Software

Wideband stimulus platforms	Maximum signal bandwidth	Maximum sample rate	Maximum memory depth	Typical/Max SFDR	Max MIMO channels
N6030A/N8241A	500 MHz, 1 GHz IQ	1.25 GSa/sec	16 MSa	50 dB/65 dB	2
		Fixed			
M9330A with option M16	500 MHz, 1 GHz IQ	1.25 GSa/sec	16 Msa	75 dB nominal	16
81180A	1.5 GHz, 2 GHz IQ	10 MSa/sec – 4.2 GSa/sec	64 MSa	56 dB/64 dB	1
M8190A**	5 GHz	100 MSa/sec – 12 GSa/sec	2 GSa	56 dB/64 dB	1

** Support for the M8190A is expected sometime after the SystemVue 2011.03 release.

Table 3. Arbitrary Waveform Generator Platforms, At a Glance

Digitization platforms	Maximum carrier frequency	Maximum signal bandwidth ¹	Maximum sample rate/second	Max memory depth @ Max sampling rate	SFDR ²	Max MIMO channels
90000 Series Infiniium oscilloscope	baseband	32 GHz	40 GSa/sec	2 GSa	55 dB	4
M9211A IF Digitizer with option M51		3000 MHz	4 GSa/sec	512 MSa	55 dB	3
M9210A Digitizing Scope		1400 MHz	2 GSa/sec	256 MSa		6
with option M51			4GSa/sec	512 MSa		3
M9392 Modular VSA (PXI format)	26.5 GHz	250 MHz		128 MSa	70dB	1
M9202A IF Digitizer with option F02		1000 MHz	2 GSa/sec	128 MSa	65dB	1
N9030A PXA , with option B1X	26.5 GHz	140 MHz	400 MSa/sec	128 MSa	75 dB	1
E444XA PSA with options 122, 123	50 GHz	80 MHz	200 MSa/sec	128 MSa	75 dB	1
N9020A MXA with option B253	26.5 GHz	25 MHz	90 MSa/sec	4 Msa	70 dB	1

1. *Dependent on carrier frequency for digitizers*
2. *These numbers are nominal (rounded) to give the user a general sense of relative performance, please use instrument specification guide for specific use case*

Table 4. Vector Signal Analysis Platforms, At a Glance

For more information about SystemVue, please visit us on the web:

Product information

<http://www.agilent.com/find/eesof-systemvue>

Product Configurations

<http://www.agilent.com/find/eesof-systemvue-configs>

Request a 30-day Evaluation

<http://www.agilent.com/find/eesof-systemvue-evaluation>

Downloads

<http://www.agilent.com/find/eesof-systemvue-latest-downloads>

Helpful Videos

<http://www.agilent.com/find/eesof-systemvue-videos>

Technical Support Forum

<http://www.agilent.com/find/eesof-systemvue-forum>



Agilent Email Updates

www.agilent.com/find/emailupdates

Get the latest information on the products and applications you select.

www.agilent.com
www.agilent.com/find/eesof-systemvue

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at:

www.agilent.com/find/contactus

Americas

Canada	(877) 894 4414
Brazil	(11) 4197 3500
Mexico	01800 5064 800
United States	(800) 829 4444

Asia Pacific

Australia	1 800 629 485
China	800 810 0189
Hong Kong	800 938 693
India	1 800 112 929
Japan	0120 (421) 345
Korea	080 769 0800
Malaysia	1 800 888 848
Singapore	1 800 375 8100
Taiwan	0800 047 866
Other AP Countries	(65) 375 8100

Europe & Middle East

Belgium	32 (0) 2 404 93 40
Denmark	45 70 13 15 15
Finland	358 (0) 10 855 2100
France	0825 010 700*
	*0.125 €/minute
Germany	49 (0) 7031 464 6333
Ireland	1890 924 204
Israel	972-3-9288-504/544
Italy	39 02 92 60 8484
Netherlands	31 (0) 20 547 2111
Spain	34 (91) 631 3300
Sweden	0200-88 22 55
United Kingdom	44 (0) 118 9276201

For other unlisted Countries:

www.agilent.com/find/contactus

Revised: October 14, 2010

Product specifications and descriptions in this document subject to change without notice.

© Agilent Technologies, Inc. 2011
Printed in USA, March 29, 2011
5990-7757EN



Agilent Technologies