# Radar System Design and Interference Analysis Using Agilent SystemVue

## Application Note

By David Leiss, Sr. Consultant EEsof EDA

*Anurag Bhargava, Application Engineer EEsof EDA Agilent Technologies*

## Introduction

This application note outlines key features of Agilent's SystemVue software for performing radar system design and jammer/interferer analysis. Some of the key areas to be discussed include how to implement a radar Chirp waveform, design an RF chain for the transmitter and receiver, and perform pulse-compression analysis using fast Fourier transform (FFT) based convolution. Finally, the radar system is tested in the presence of unwanted interference and jamming signals to study the impact of such unwanted impairments on radar performance.
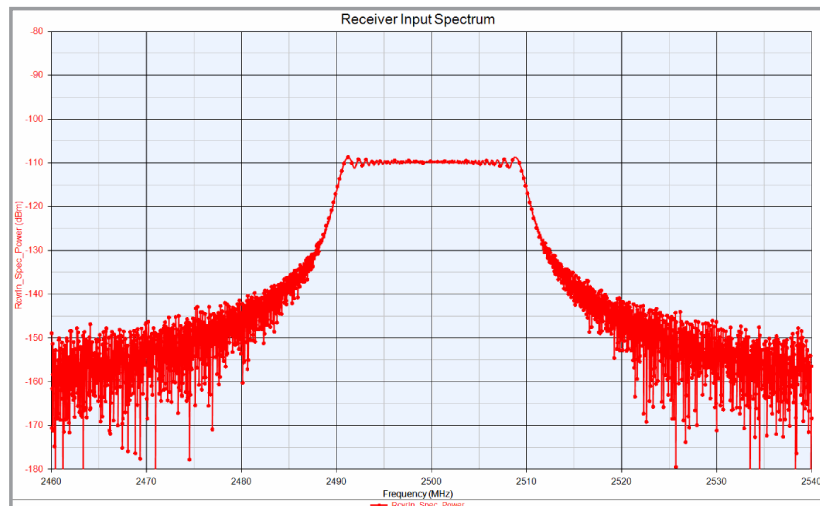


**Agilent Technologies**

# 1.0 Custom Signal Generation

## 1.1 LFM Chirp for Radar System Design

SystemVue offers a flexible platform to create custom signals. In the example shown in Figure 1-1, SystemVue floating-point Data Flow components were used to model the LFM Chirp source. The integrator on the left increments time until the value for the pulse period is achieved, causing it to reset and start over. The values for u (μ) and wc (ω$_c$) are computed as shown in Figure 1-1.
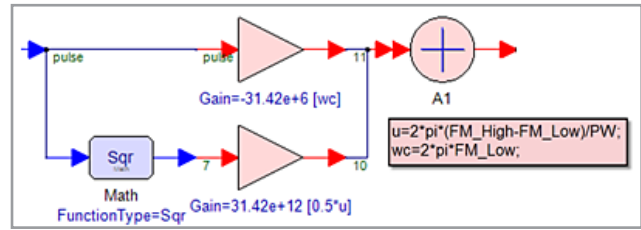
(1-1a)
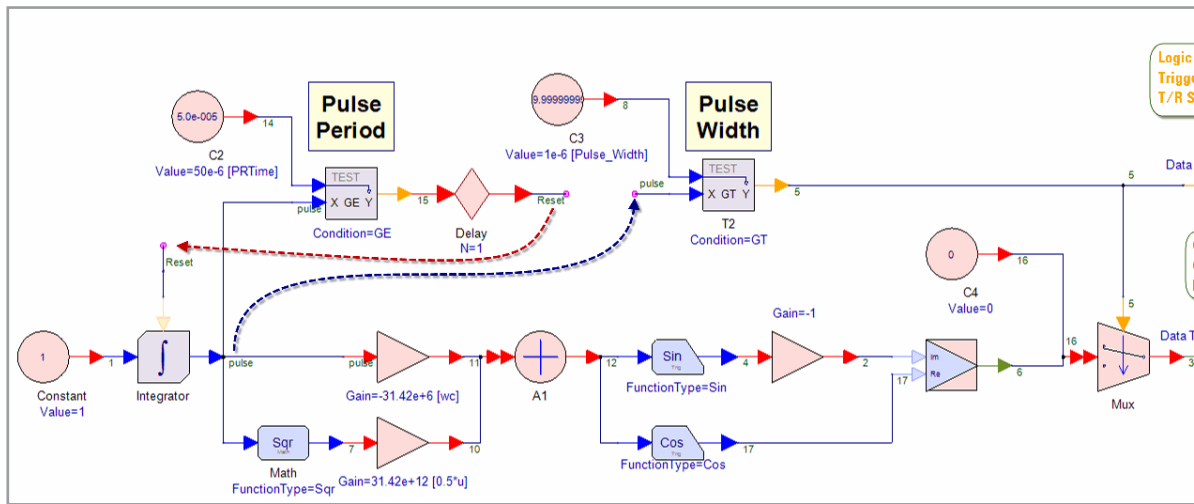
(1-1b)

*Figure 1-1. Custom signal generation using SystemVue DSP library blocks*

# 1.0 Custom Signal Generation, Continued

## 1.2 Custom Signal Generation using MathLang

SystemVue offers built-in m-code compatible syntax to be used throughout the program. In Figure 1-2, the LFM Chirp source is defined in a MathLang component.

## 1.3 Custom Signal Generation using Third-Party Tools

SystemVue offers direct links with C++, HDL and MATLAB®. Any custom signals written using these languages can be easily brought into SystemVue as shown in Figures 1-3 and 1-4. Co-simulation with MATLAB allows the user to incorporate pre-existing m-code files.
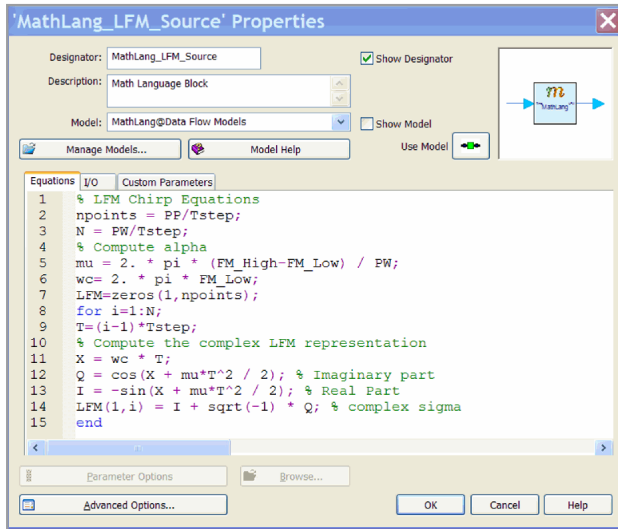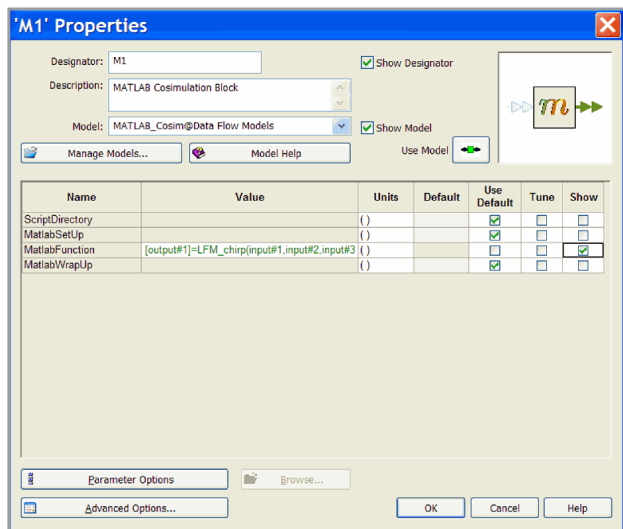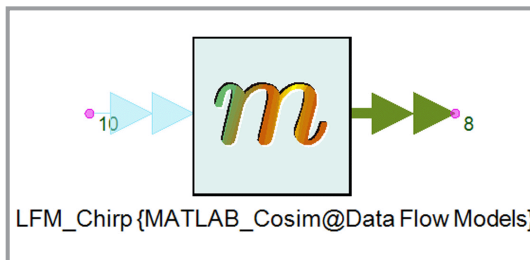
Figure 1-2. Custom signal generation using MATH Language in SystemVue

(1-3a. MATLAB co-simulation link)

(1-3b)
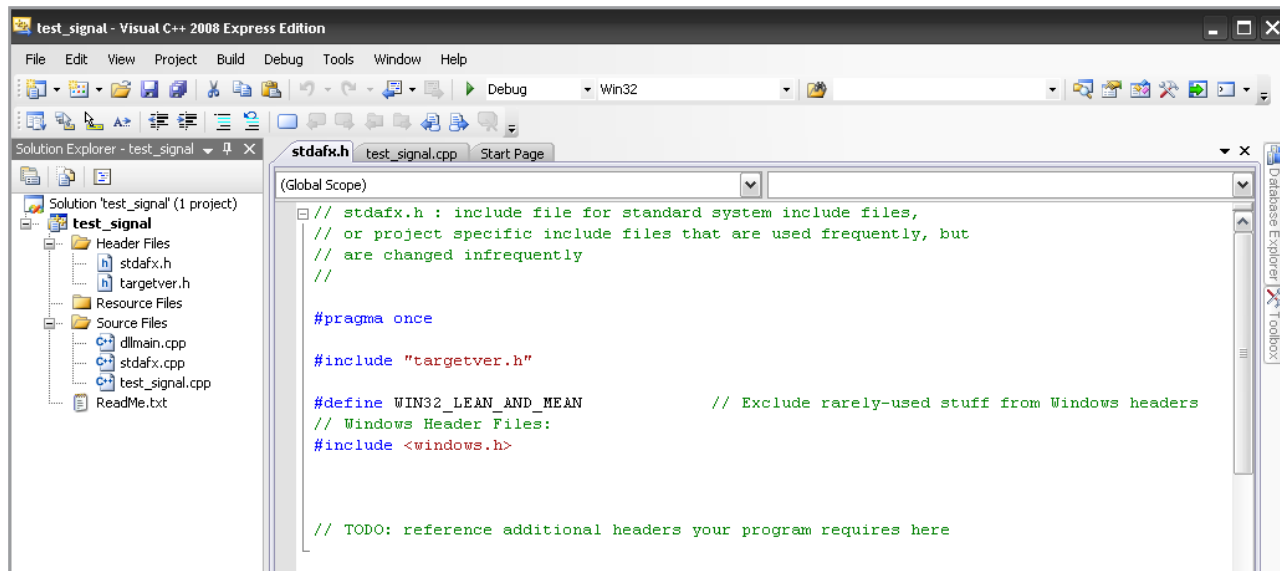
Figure 1-3. Linking MATLAB script with SystemVue

Figure 1-4. Custom waveform code in C++

# 2.0 LFM Chirp IF Generation

In this application note, Math Language is used to generate the LFM waveforms used for radar system design (Figure 2-1).

Note that the MathLang component does not provide SystemVue with the sample rate information that its built-in models do. It is recommended, therefore, that a sample rate component be added after a MathLang component that is used as a source to implicitly define the sample rate (Figure 2-1). The CxToEnvelope component defines the complex waveform as an RF Envelope waveform where I&Q, plus time and carrier frequency are defined. The spectral result, centered around 500 MHz, is shown in Figure 2-1.
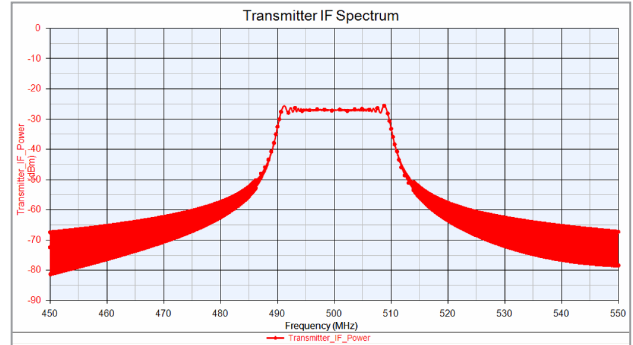


(2-1a)
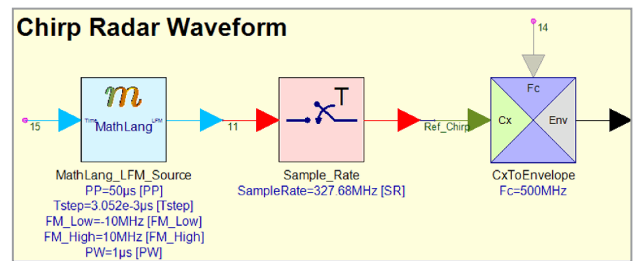


(2-1b)

*Figure 2-1. LFM Chirp signal generation and IF spectrum. Note that the Sample_Rate component declares the time step for the un-timed complex numeric data emerging from the MathLang source.*
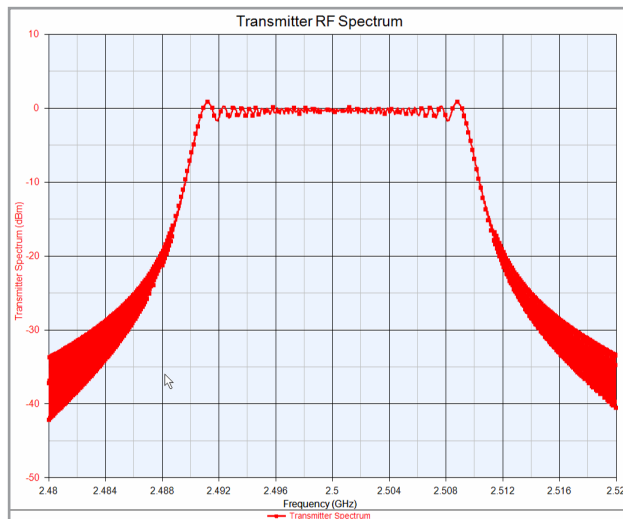


*Figure 3-1. Transmitter RF spectrum*

# 3.0 Transmitter RF Design

The RF chain of the radar transmitter was designed using the Data Flow RF block library in SystemVue, which offers users a variety of RF models to implement the RF section of the system (Figures 3-1 and 3-2). Real-world impairments such as nonlinearities, LO phase noise and mixer leakage products can also be incorporated into the simulation.
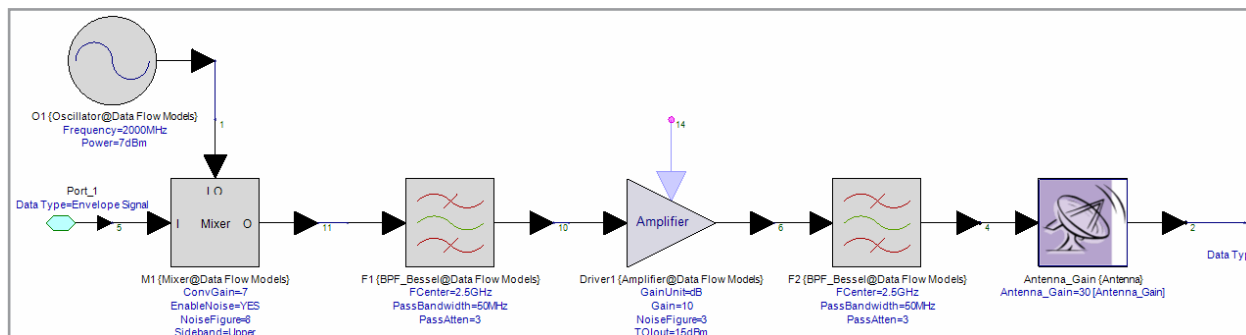


*Figure 3-2. RF section of radar transmitter using the RF block library in SystemVue*

# 4.0 Radar Propagation Loss Modeling

The radar propagation path modeling was performed using SystemVue library blocks to implement standard Math equations for propagation delay and free-space propagation loss (Figure 4-1 and 4-2).
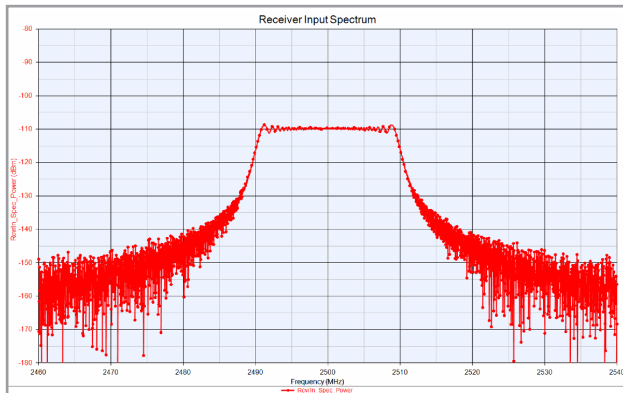


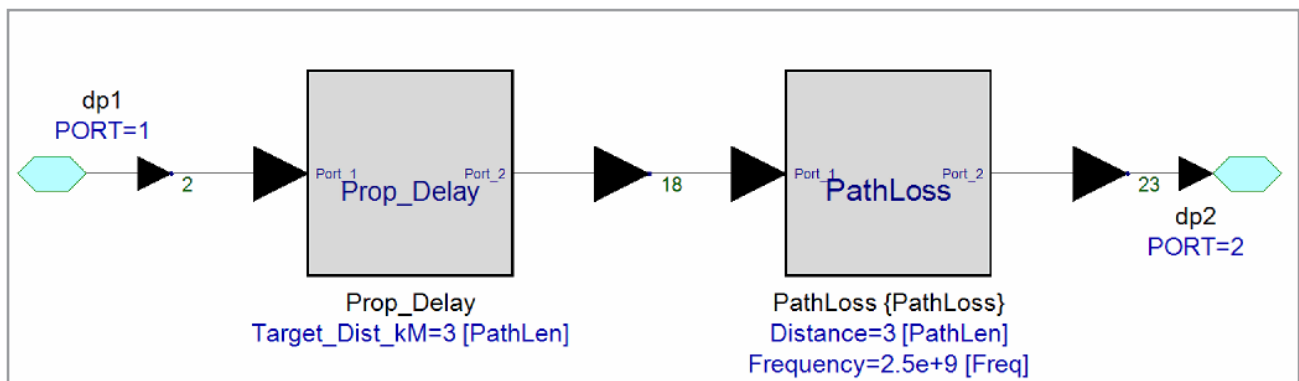Figure 4-1. Radar signal as seen at receiver input



Figure 4-2. Implementing radar propagation loss and propagation delay models

# 5.0 Radar Receiver Design

The radar front-end (FE) receiver was designed using RF blocks. Various budget analyses were performed to optimize the performance as shown in Figure 5-1 to Figure 5-5.
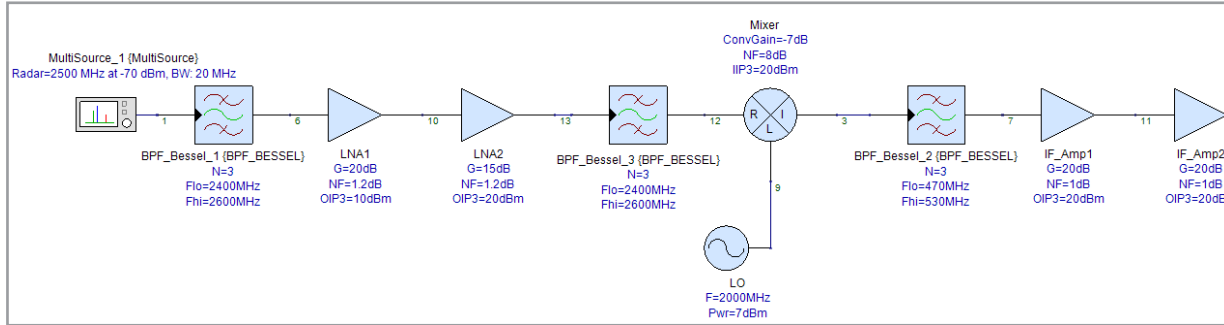


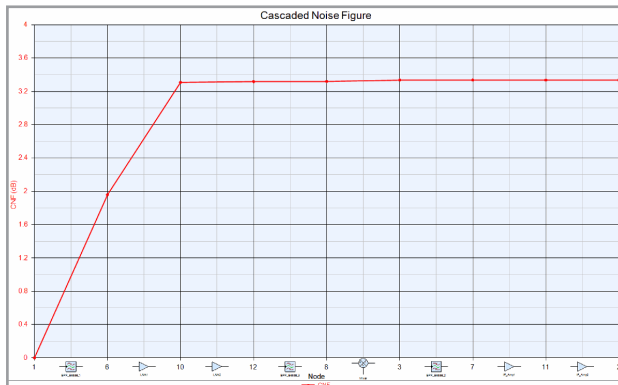*Figure 5-1. Radar front-end receiver design using RF block library in SystemVue*



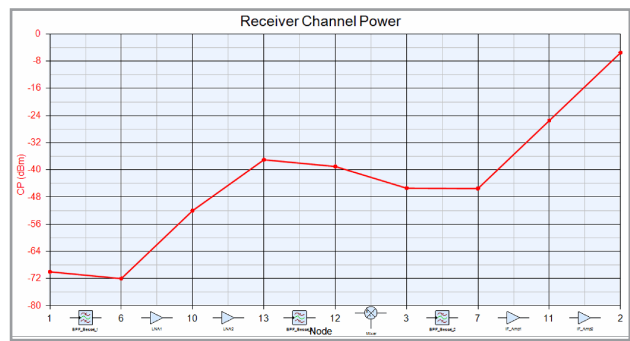*Figure 5-2. Noise-figure budget analysis of receiver front end*
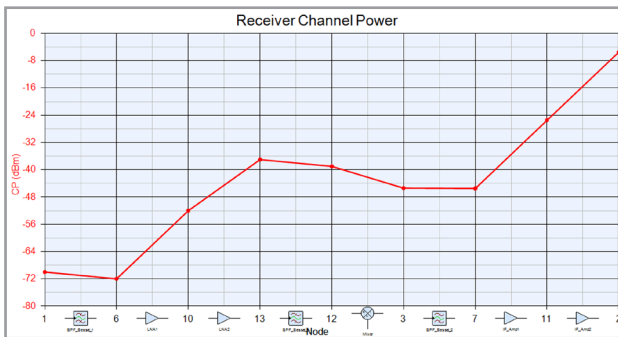


*Figure 5-3. Channel-power budget of receiver front end*
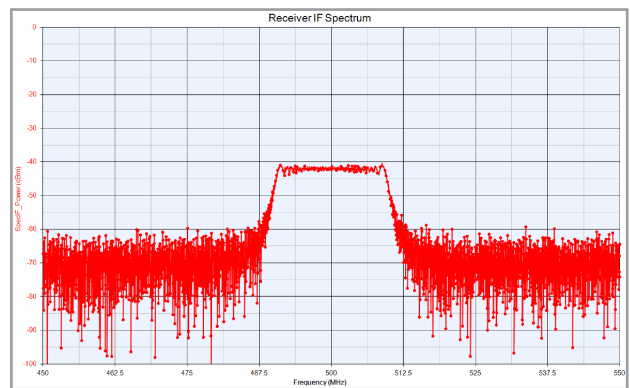


*Figure 5-4. Cascaded-gain budget of receiver front end*



*Figure 5-5. Receiver output IF Spectrum*

6

# 6.0 Receiver Signal Processing

The receiver IF was filtered, down converted and down sampled for analog-to-digital conversion. The post processing algorithm was designed using SystemVue DSP library blocks as shown in Figure 6-1 to Figure 6-5.
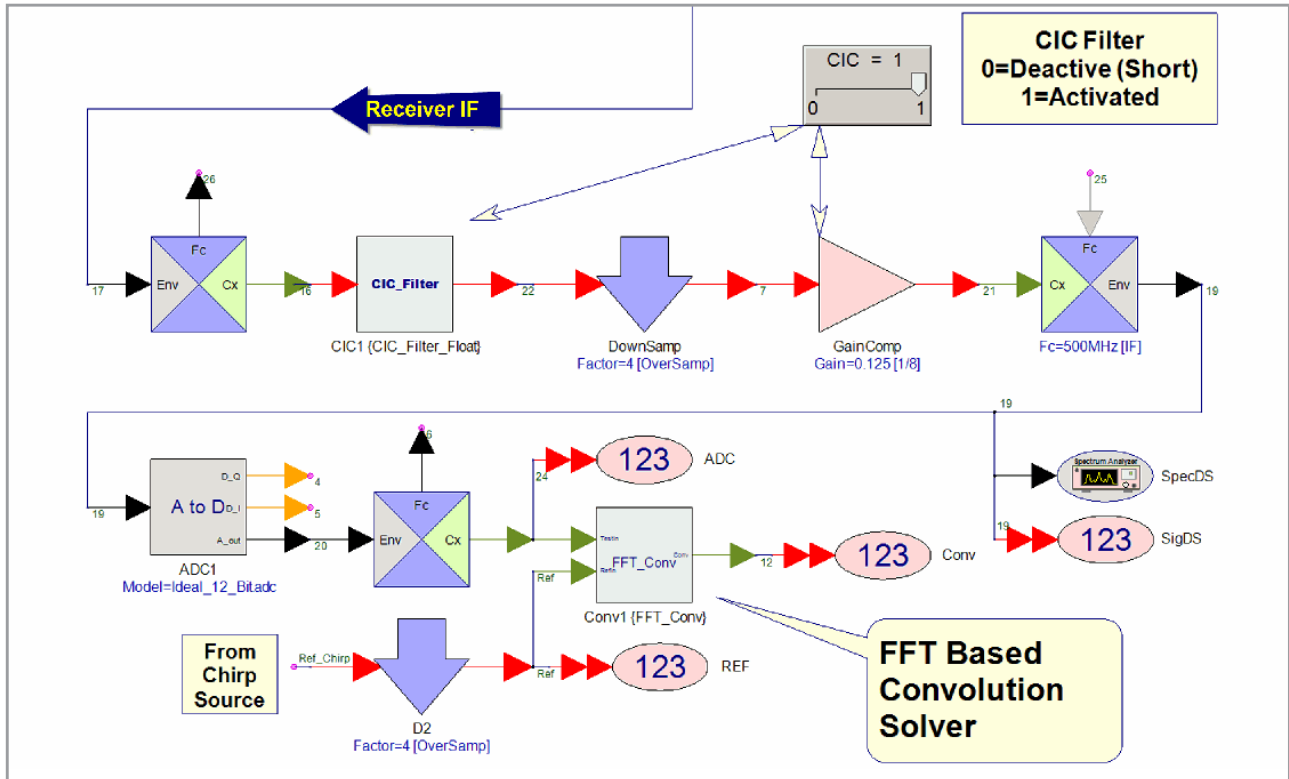


*Figure 6-1. Top-level view of receiver IF post processing*

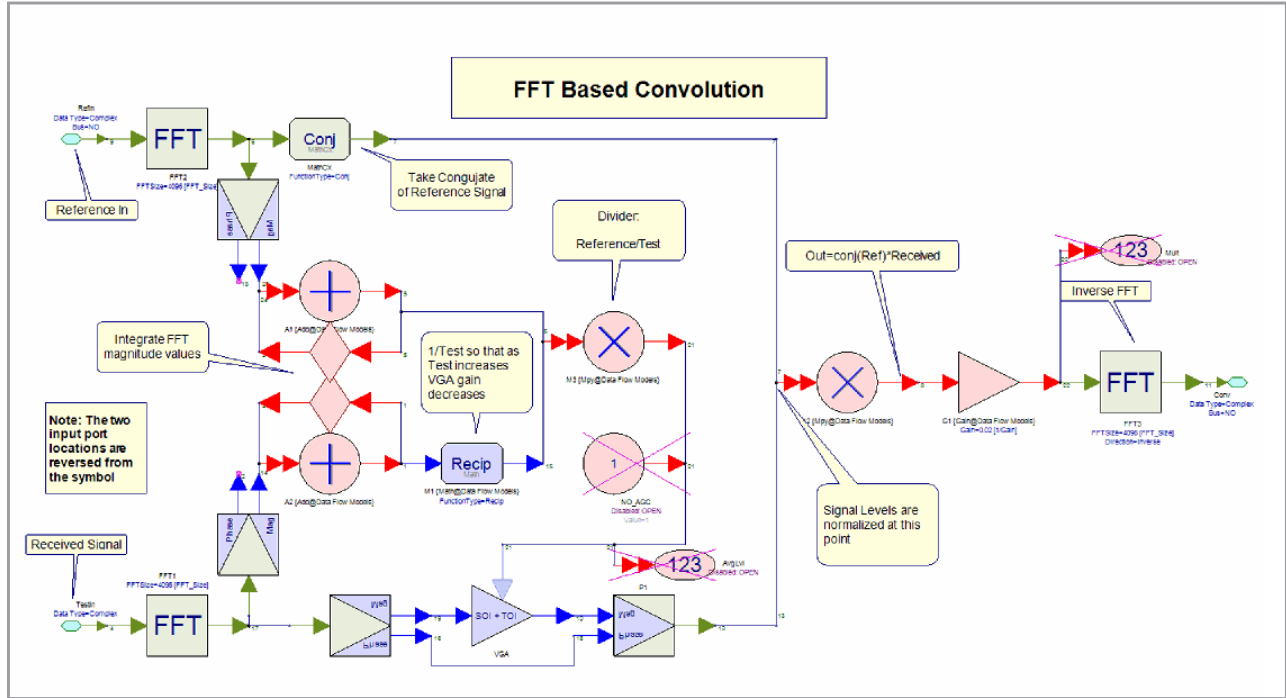# 6.0 Receiver Signal Processing, Continued



Figure 6-2. FFT-based convolution algorithm (pulse compression) using SystemVue DSP library blocks
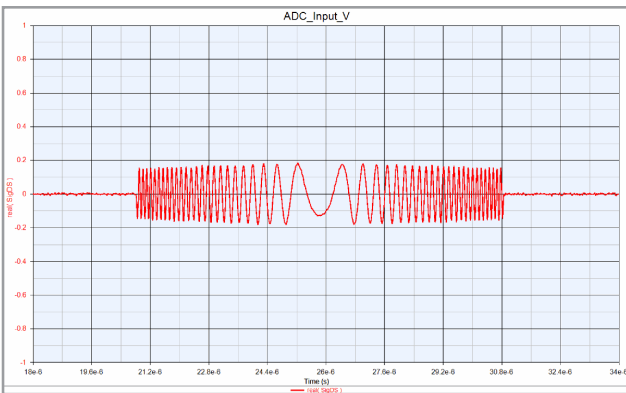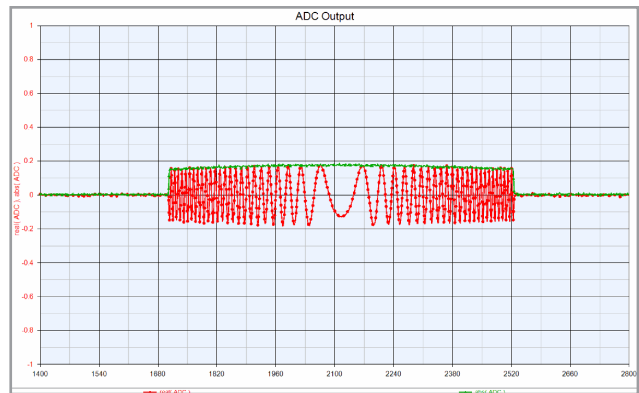


Figure 6-3 Time-domain waveform at ADC input



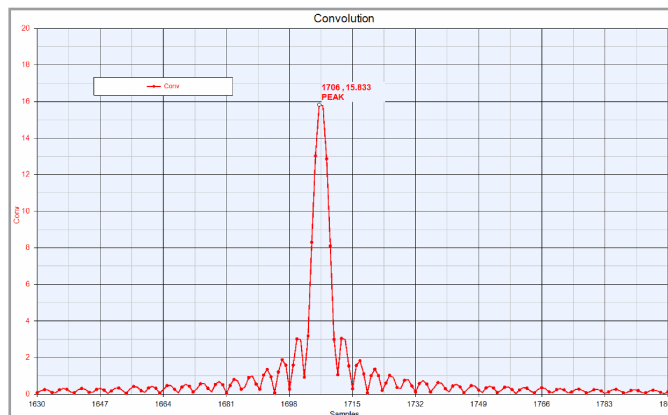Figure 6-4. Time-domain reconstructed data at ADC output



Figure 6-5. Compressed pulse after post processing

# 7.0 Jammer/Interference Analysis

Jammer/interference analysis can be performed on the radar system design as shown in Figure 7-1 to Figure 7-4. Note that jammers/interferers can be varied to have the different levels of amplitude and frequency needed to perform radar receiver fidelity analysis and what-if analysis.
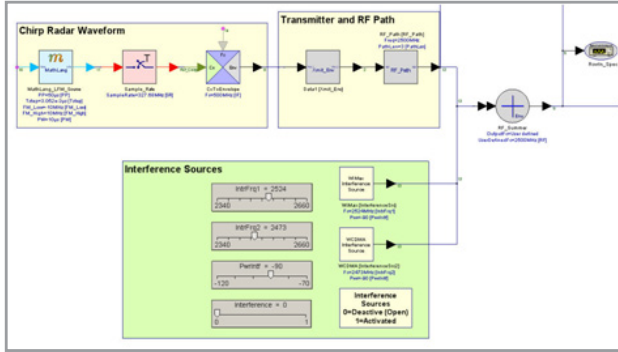


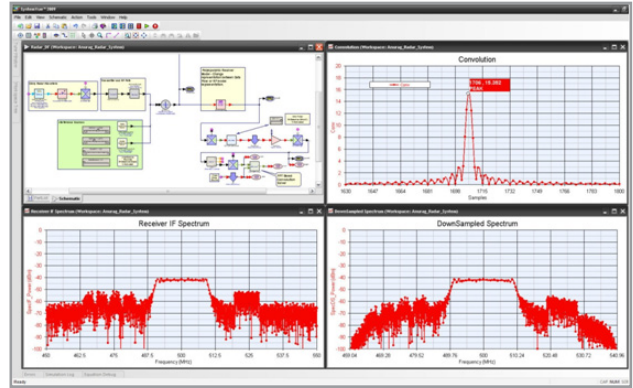Figure 7-1. Creating and combining interferer source with radar waveform



Figure 7-2. Composite spectrum at radar receiver input



Figure 7-3. Various waveforms in the presence of jammers/interferers
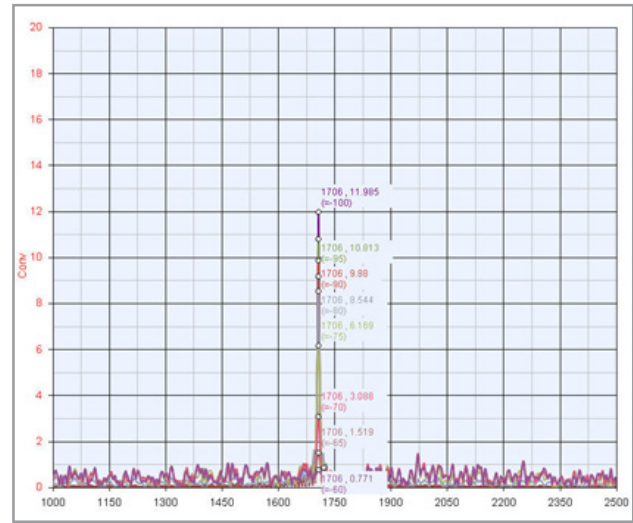


Figure 7-4 Sweeping interference power to see its effect on compressed-pulse peak detection

# 8.0 Integrating software and instruments for advanced system-level design

While it is always possible to design and test complex defense systems separately, ideally the engineer would like these two domains to be inter-linked to enable a fully integrated approach to system development and testing. This can be accomplished using the native instruments link option in SystemVue. This option allows both the designer and test engineer to link a variety of test and measurement equipment with SystemVue in order to achieve following objectives:

• Create and download arbitrary signal to instruments

• Create near real-life signals for realistic system integration testing in the lab environment

• Capture signals from instruments and design remaining blocks by taking real device distortions into account

## 8.1 Integrated Test Setup

A fully integrated approach to system development and testing can be used to address the case study presented in this application note. The test setup, as shown in Figure 8-1, requires:

• SystemVue software installed on a PC

• An Agilent vector signal generator (VSG), in this case the N5182A MXG

• An Agilent PXA Signal Analyzer, in this case the N9030A PXA

In this case study, the LFM Chirp Radar waveform is used. It is designed using SystemVue software with different specifications and downloaded onto the VSG (Figures 8-2 and 8-3). SystemVue allows data to be downloaded from any node in the design. In this case, the node selected is the one where there is a combined spectrum of main radar return coming to the receiver input and where interfering signals can be added to perform receiver signal processing fidelity test (Figure 8-4).
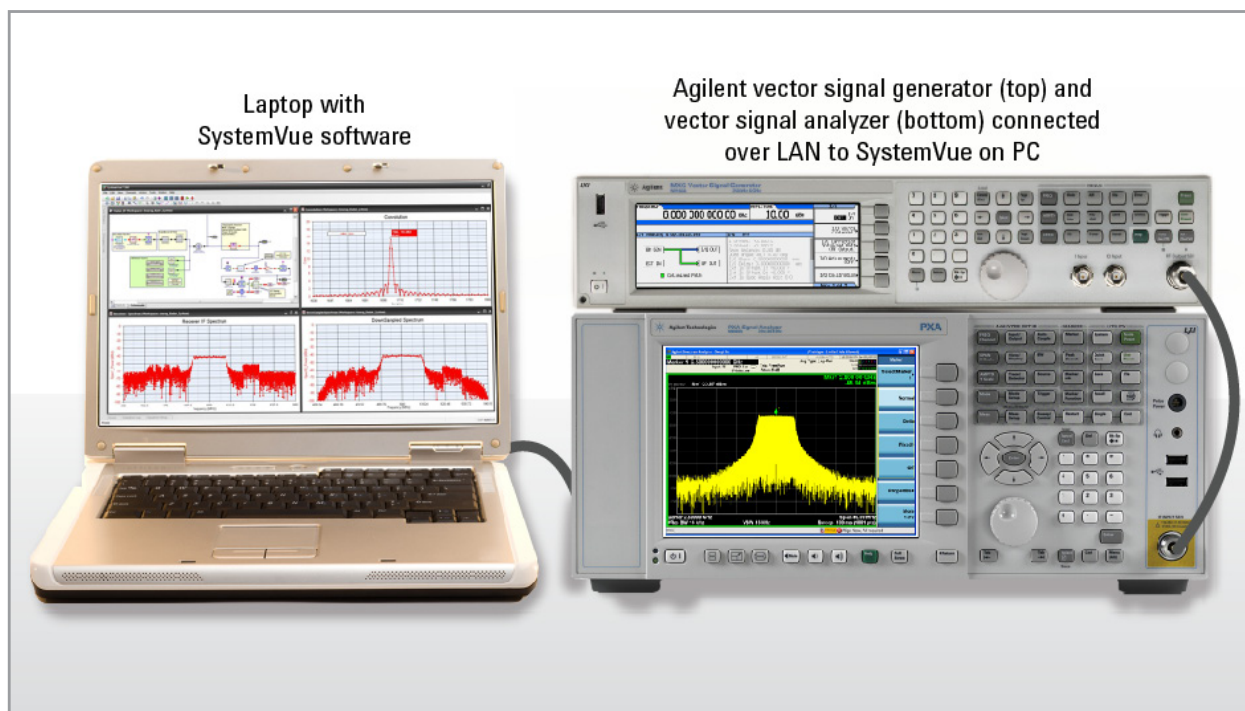


*Figure 8-1. Custom signal download to a VSG*

## 8.0 Integrating software and instruments for advanced system-level design, Continued
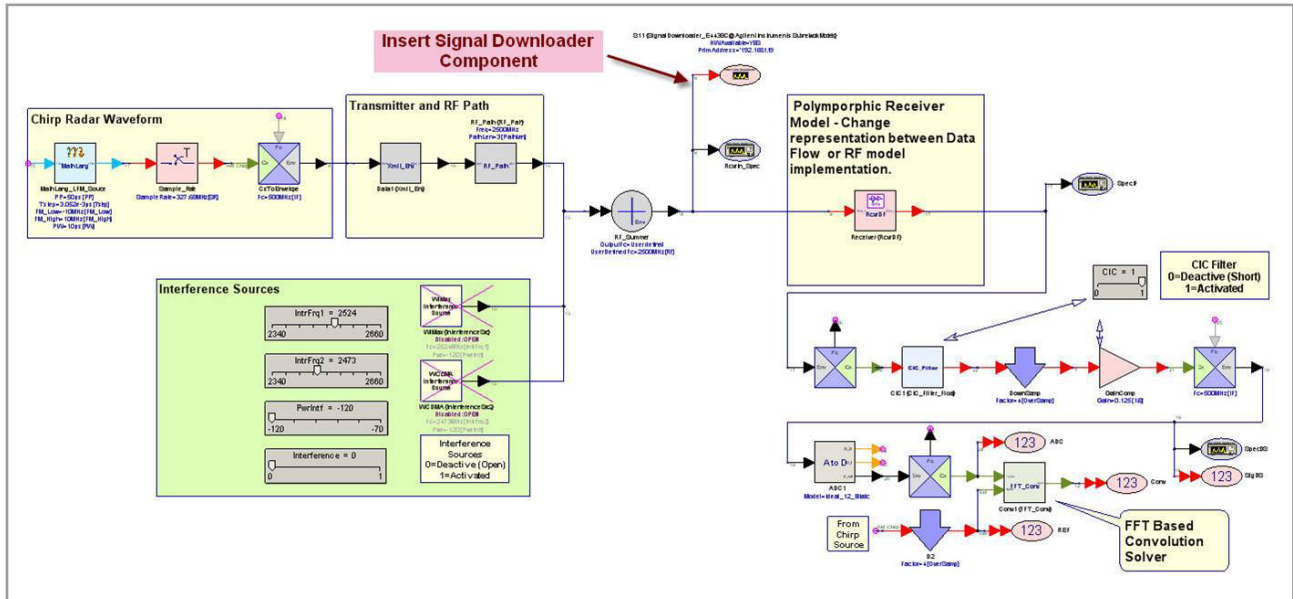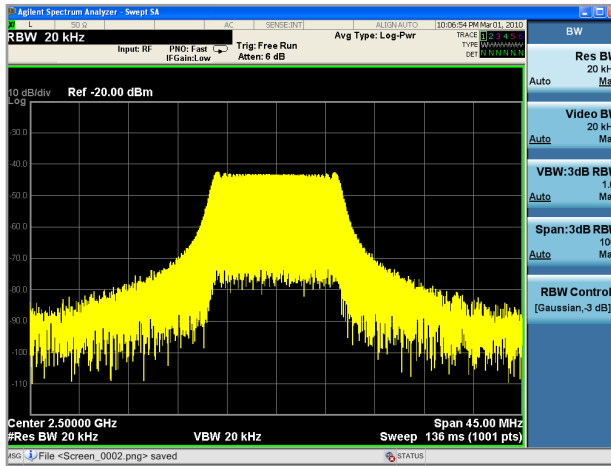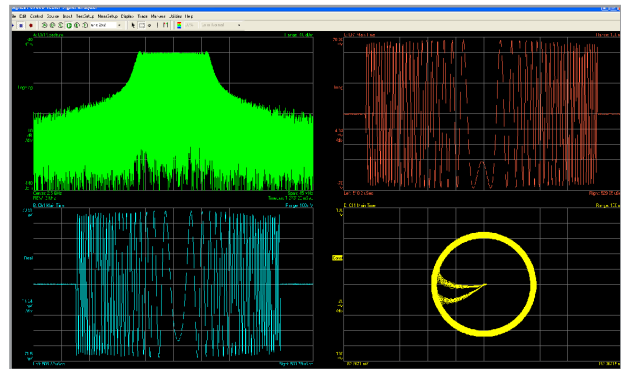


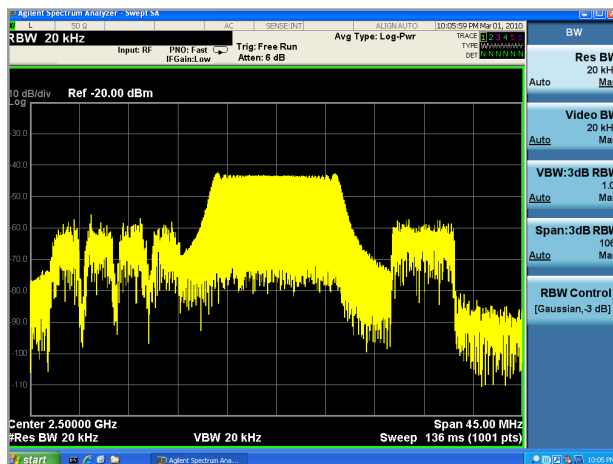*Figure 8-2. SystemVue workspace with signal downloader component to download custom waveform to VSG*
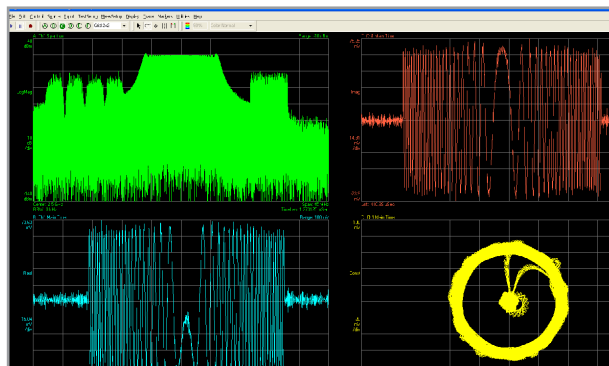


(8-3a)



(8-3b)

*Figure 8-3. Ideal LFM Chirp waveform being analyzed using spectrum analyzer and its vector signal analysis using 89600 VSA software running on the N9030A PXA*
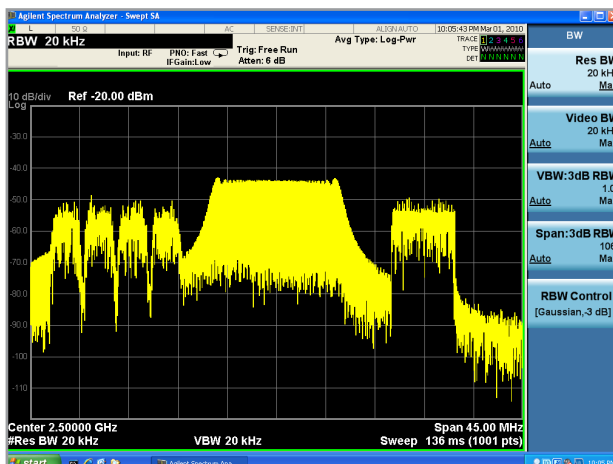
# 8.0 Integrating software and instruments for advanced system-level design, Continued
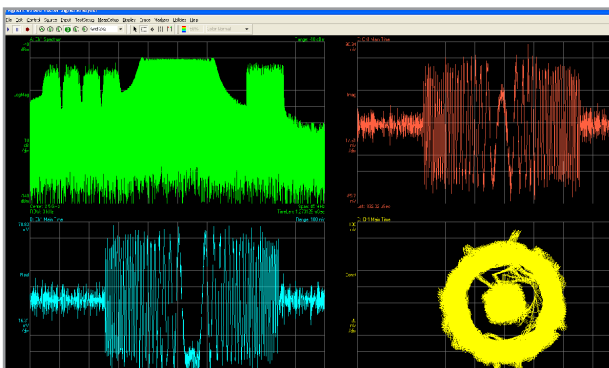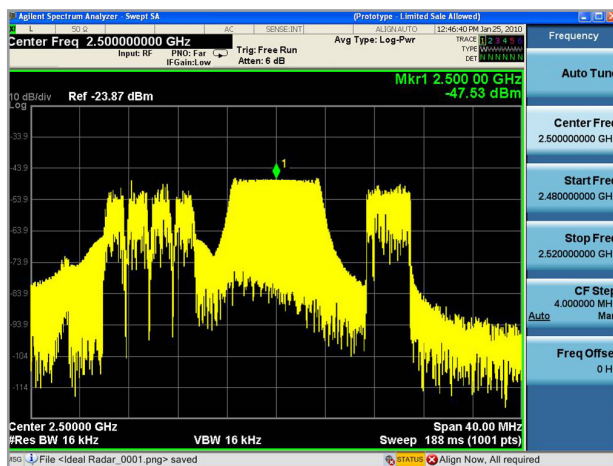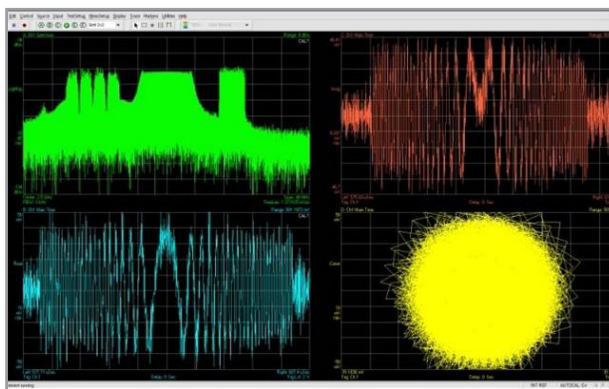

(8-4a)


(8-4b)


(8-4c)


(8-4d)


(8-4e)


(8-4f)

*Figure 8-4. Composite LFM radar with interferer signal and interference power sweep to understand receiver behavior under different conditions*

# 9.0 Conclusion

Agilent SystemVue provides a very flexible platform for implementing complex aerospace and defense systems such as radar. With its unique "Envelope" simulation technology, for example, all aspects of system design, including RF and digital sub-systems, can be handled easily. Some of the other key benefits of SystemVue for the design and test of complex aerospace and defense systems are:

- Its RF architecture capability allows for quick and accurate RF system design, while its budget analysis capability helps fine tune and optimize the RF system performance.

- A unique Real-Time Tuning and Sweep feature allow any parameter to be varied and its effect on system performance quickly analyzed.

- It enables a complete fixed-point digital implementation, whereby real DSP systems can be designed and bit-true and cycle accurate fixed-point VHDL or Verilog codes can be generated automatically for FPGA implementations.

- It provides a seamless integration with instruments like VSGs, signal analyzers, scopes, and logic analyzers that can be used to create and download standard or custom test vectors for complex system verification. Data from the device-under-test can also be captured and brought into software to design the system's signal processing section or for RF system design and optimization.

- It provides direct integration with third-party digital/DSP tools such as ModelSim, MATLAB and C++. Integrating such tools/third-party IP into one platform enables complete system design and validation.

# 10.0 References

Additional radar resources from Agilent:

- **www.agilent.com/find/radar**
- SystemVue home page
  **www.agilent.com/find/eesof-systemvue**
- SystemVue videos:
  **www.agilent.com/find/eesof-systemvue-videos**

MATLAB is a U.S. registered trademark of The Math Works, Inc.

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

**Agilent Technologies**