

## Radar Emitter Simulation Using Vector Signal Generators

*Presented by:*



With the advent of microwave signal generators and spectrum analyzers with vector capability, the engineer can now generate pulsed microwave signals with precise control over output power, amplitude envelope, and modulation within the pulse. These precision signals can be used as a standard to evaluate the performance of subsystems and to troubleshoot system problems. The devices to be tested are typically radar warning receivers and ELINT(Electronic intelligence) receivers. The purpose of this paper is to help the design engineer generate and evaluate complex radar signals using standard microwave test equipment.

The approach taken is to show how to implement a pulse test pattern generator. The pulse test pattern generator produces a series of pulses that will stress one specific parameter in the receiver, such as the ability to correctly identify two closely spaced pulses. The pulse test pattern generator has the flexibility to simulate a static emitter, but lacks the memory depth or real time processing to simulate movement of the emitter over time. Experience has shown that a series of complex pulse patterns will enable the user to perform roughly 80% of the tests necessary to evaluate the performance of a system. The final 20% of the receiver testing is typically done on a test range using real emitters.

**At the completion of this module you will have an understanding of how to produce pulsed waveforms using I & Q sample points**

Agenda

- Tools needed for the generation and analysis of pulsed waveforms
- Creating simple pulses using Matlab
- Creating pulses with compression using Matlab
- Other methods and tools for radar signal simulation

## Agilent E8267C PSG Vector Signal Generator

- 250KHz - 20GHz Frequency Coverage
- Superior level accuracy
- High power +18dBm
- Excellent phase noise
- AM/FM/PM and pulse modulation capabilities
- Internal 80MHz I/Q - Arbitrary Waveform Generator
- Optional Extended Analog IQ Inputs with 1GHz RF Bandwidth



The general-purpose test equipment needed to evaluate a receiver is a microwave signal generator capable of producing the signals required for the test and a microwave spectrum analyzer capable of verifying the signal's characteristics. The equipment should cover a frequency range of 0.5-18GHz. If the receiving system must process phase or frequency coded pulses, then the generator must be able to produce these signals.

The Agilent E8267C Vector Performance Signal Generator that covers 250KHz to 20GHz meets the needs for a general-purpose signal source to test radar warning receivers and ELINT systems. The generator is a member of the PSG line and provides excellent output power, low phase noise option, analog modulation, and digital communication modulation common to that line. The generator also provides new wide-bandwidth I/Q modulation with an internal arbitrary waveform generator providing 80MHz of modulation bandwidth. The generator offers an optional 1GHz analog IQ modulator with external inputs. When used with an external baseband source, the generator could produce the signal typical of synthetic aperture radars or imaging radars. This paper will focus on the use of the internal arbitrary waveform generator to produce complex radar signals.

## Performance Spectrum Analyzer Agilent E4440A

- Up to 50 GHz Frequency Coverage
- Best accuracy:
  - 0.62 dB @ 1 GHz,
  - 3.24 dB @ 50 GHz
- Modern connectivity:  
GPIB, LAN, floppy disk
- Optional Vector Signal Analysis on External PC
  - 8MHz VSA BW using Internal Digitizer
  - Up to 36MHz VSA BW using External VXI Digitizer



To capture and demodulate signals the E4440A Performance Spectrum Analyzer was used with the 89601 Vector Analysis Software. This configuration provided frequency coverage of 26.5GHz with up to 36MHz of analysis bandwidth. Using the internal digitizer provides 8MHz of analysis bandwidth and using an external 89611A VXI digitizer provides 36MHz of analysis bandwidth.

## Vector Signal Analyzer Configurations



IF and Baseband  
or RF



Ultra Wide Bandwidth



Microwave



The IF and Baseband product, which is the 89611A, captures 70MHz with 36MHz of analysis bandwidth. The sample rate is 95MHz with 800 Msamples as the maximum record length. This provides over 8 seconds of capture waveform.

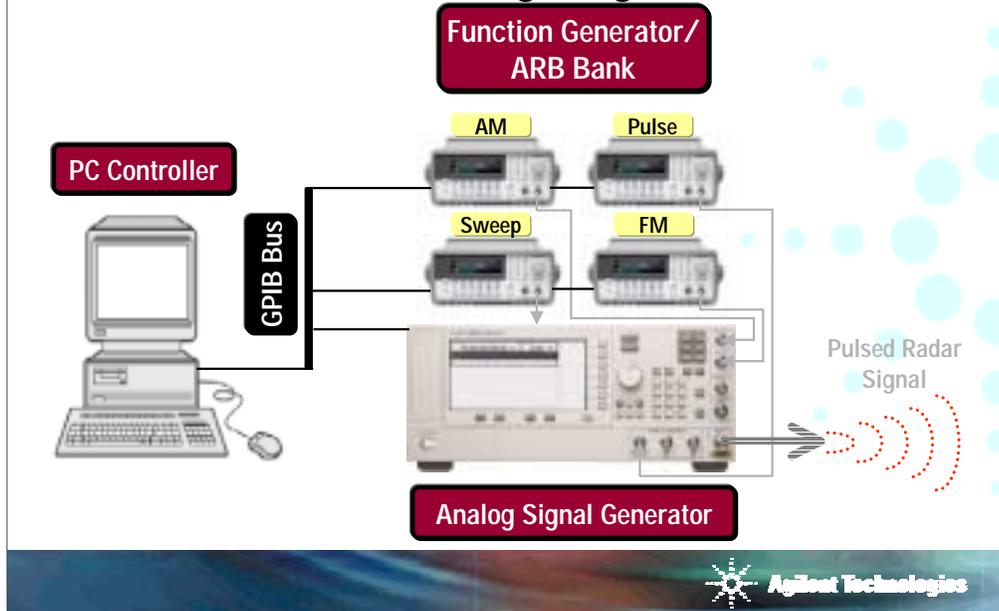
As a general comment, if the signal you are trying to capture occupies significantly less than 36MHz in this case, the system will digitally down convert the waveform and thus capture a much longer signal. So as you reduce the bandwidth occupancy from 36MHz, then you will significantly increase the period of time you can capture the waveform.

The RF products are contained within VXI mainframes. The 89640 covers dc to 2.7GHz and the 89641A covers dc to 6GHz. Both of these configurations provide 36MHz of analysis bandwidth, with at least 8.4 seconds of capture waveform.

The microwave configuration uses 2 products the E4440A Precision Spectrum Analyzer with option H70 and the 89611A Digitizer. The combination of these two products provide 3Hz to 26GHz of frequency coverage. The analysis and 36MHz with again 95MHz of sample rate and 800 Msample of memory depth.

The final configuration is the Ultra wide bandwidth configuration. This uses the 54832B Infiniium Oscilloscope and the 89601A Vector Signal Analysis software. This configuration covers dc to 1GHz with 1GHz of analysis bandwidth. It uses a sample rate of 4GHz with 8 Msample of memory depth. This provides 2ms of capture waveform.

## Traditional radar signal generation



Because most radar systems are custom built, signal generators are used in all stages of the design process. Given the broad range of applications, a variety of signal generation solutions have emerged to provide a test stimulus for the employed radar system. These range from home brew solutions based on golden devices to sophisticated signal generation solutions like the obsolete Agilent FASS (frequency agile signal simulator) system.

A typical setup using analog techniques to generate radar test signals is illustrated on the slide. This is a simple system configured using standard test equipment. In this setup, multiple function generators with basic ARB functionality are used to drive the Pulse, AM, FM, and Sweep inputs of an analog microwave signal generator. Using this setup a variety of radar test signals can be generated, ranging from pulsed CW... to shaped pulses using AM... to complex chirps using FM and Sweep. This is adequate for some applications, however the pulse shaping capability is limited and managing the timing relationships associated with pulsing modulated signals can be complicated. A PC is often used to control the instruments over the GPIB bus. Because of the system complexity, there is also a high cost associated with maintaining a system like this.

**Aerospace Defense Symposium 2003**

## Complex radar signal generation

**Features**

- Arbitrary I/Q waveform generation
- Deep waveform playback memory
- Flexible waveform sequencing
- Mass storage for waveforms
- Repeatability of digital signals

**Value**

- Simplified test setup
- Custom pulse shaping
- Flexible test pattern generation
- Create a library of pulse patterns and quickly retrieve for playback

Agilent Technologies

Rather than using traditional analog techniques, the PSG vector signal generator utilizes I/Q modulation to provide flexible radar signal generation, unparalleled in a standard off-the-shelf product.

Now, arbitrary waveforms representing pulsed radar signals can be defined in the time domain using industry standard tools, like Matlab or ADS, and played back using the PSG's internal baseband generator. This technique enables custom pulse shaping, accommodates pulse compression, including FM chirp and Barker coding, and eliminates many of the synchronization issues associated with pulsing modulated signals using traditional analog techniques. In addition, the baseband generator's deep playback memory and waveform sequencing capability facilitates the generation of complex radar test patterns.

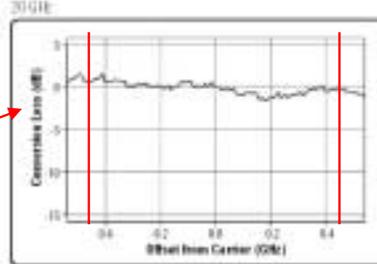
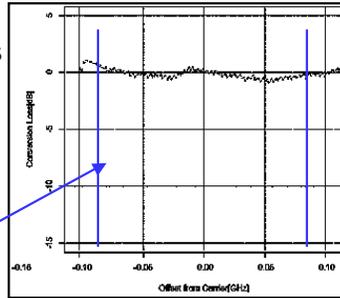
## PSG Vector Signal Generator

External I/Q inputs



**Standard**  
 160 MHz RF modulation bandwidth  
 External I/Q inputs

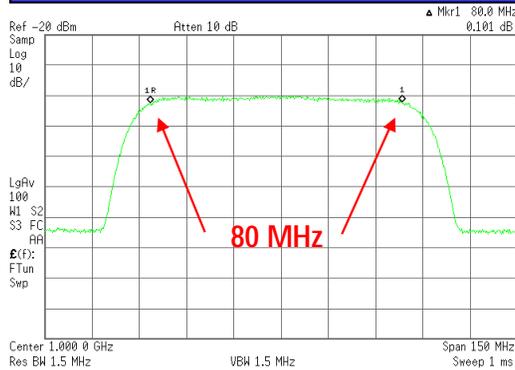
**Option 015**  
 Up to 1 GHz RF modulation bandwidth  
 Enhanced bandwidth external I/Q inputs  
 (uncalibrated, unlevelled, unfiltered)



The base PSG product has 160MHz of standard RF modulation BW for external I/Q. For satellite customers and others needing very wide modulation bandwidth we have provided option 015 which allows up to 1 GHz (that's right 1 GHz) of RF modulation bandwidth for external I/Q. Option 015 provides direct access to the modulator with out calibration. This means the inputs are uncalibrated, unlevelled and unfiltered. On the right, we are showing the typical frequency response of the different I/Q inputs.

## Internal Baseband Generator RF Modulation Bandwidth

### 80 MHz RF Modulation Bandwidth



### Applications

- Generate broadband signals
  - MMDS
  - LMDS
  - OFDM
  - Chirp Radar
  - Satellite
  - Multi-Carrier

The wide modulation bandwidth available in the new PSG means that you can generate the broadband signals you've wanted to test, without worrying about roll off at the band edges. Multicarrier and WLAN are no problem. If you've wanted a way to pre-distort your arbitrary waveform files to remove 3<sup>rd</sup> and 5<sup>th</sup> order intermodulation products, now you can. And if the high speed internal modulation source still isn't enough, you can drive the external I/Q ports to achieve 160 MHz modulation bandwidth or optionally up to 1 GHz.

## Firmware Personalities

### Arbitrary waveform mode – Custom modulation

#### Clock

Sample rate:  
1 Hz to 100MHz  
Resolution :  
0.001 Hz

#### Non-volatile storage

Standard: 1 Msample (5 MB)  
Option 005: 1.2 Gsample (6 GB)

#### Markers

Four marker signals

#### Connectivity

10BaseT LAN  
GPIB  
RS-232

#### Waveform segments

Segment length:  
60 samples to 64 Msamples  
Maximum number of segments:  
4,096 (64 Msamples memory)

#### Waveform sequences (continuously repeating)

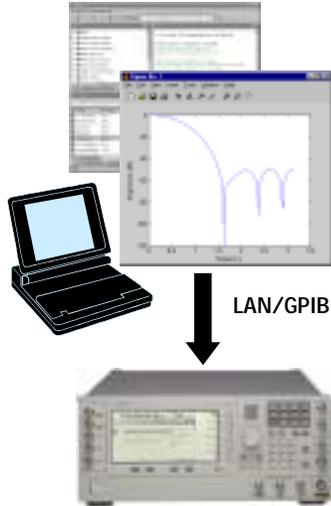
Maximum number of sequences:  
16,384  
Maximum segments/sequence:  
32,768  
Maximum segment repetitions:  
65,536



The ARB has variable sample clock, waveform segment storage and the ability to create sequences of segments. These features allow simple to complex waveform play back. Option 005 can be purchased to expand the waveform storage to 1.2 Gsample. The four marker signals can be set to trigger external events for example, a marker can be placed on the start of the waveform, center of the pulse, rising edge, or falling edge.

## PSG/ESG Download Assistant

Direct download from MATLAB® to signal generator



- Download data
- Play waveforms
- Set sample rate
- Add markers
- Control PSG using SCPI
- All from the MATLAB® command line

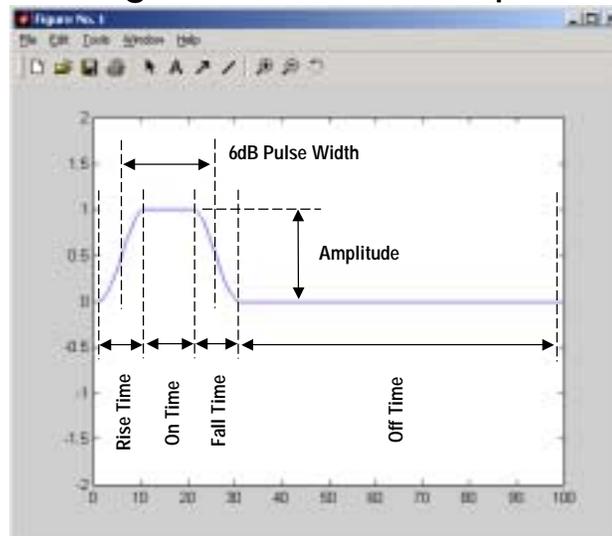
**FREE software**



Creating custom radar signals requires a software-programming tool capable of dealing with complex array math and displaying the information in a usable format. While there are several tools available, Matlab is widely used and commonly available. Agilent has chosen to support Matlab as a waveform builder for the PSG. The problem to be dealt with is how to move the digital waveforms from Matlab into the arbitrary waveform generator inside the PSG.

Agilent has developed 'Download Assistant' for Matlab to enable users to easily download their IQ waveforms into the arbitrary waveform generator's memory. In addition to downloading waveforms, it allows the user to send any SCPI command to the signal generator to control the instrument state. 'Download Assistant' adds keywords to Matlab to format and download arrays of data through common GPIB interface cards or a LAN interface into the signal generator. The examples used in this paper demonstrate the use of 'Download Assistant' with Matlab 6.0 or above to create, download, and generate radar signals. 'Download Assistant' and the programming examples used in this paper can be obtained for free at the Agilent web site: [www.agilent.com/find/psg](http://www.agilent.com/find/psg).

## Building the Pulse Envelope



To adequately test ELINT receiver performance, a wide variety of test signals are needed. The user may wish to simulate various types of radar emitters or to simulate the multiple modes of operation for a single type of radar. This requires the test engineer to control the basic pulse parameters: center frequency, power, Pulse Width (PW), and Pulse Repetition Interval (PRI). Doing a reasonable simulation of a simple emitter also requires control of the rise time and fall time of the pulse. Shaping the rising and falling edge of the pulse enables the user to control the frequency spectrum of the waveform.

## Generate a Simple Pulse

```

sampclk = 100e6;           % ARB Sample Clock for playback

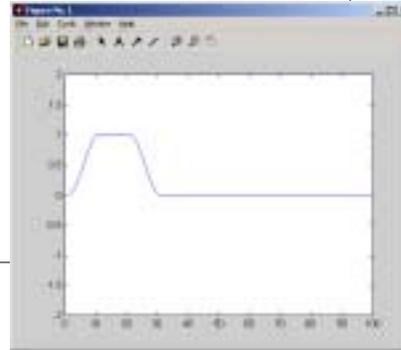
n=10;                      % number of pts in the rise & fall time
ramp=-1:2/n:1-2/n;        % ramp from -1 to almost +1 over n pts
rise=(1+sin(ramp*pi/2))/2; % raised cos rise-time shape
on=ones(1,10);            % on-time characteristics
fall=(1+sin(-ramp*pi/2))/2; % raised cos fall-time shape
off=zeros(1,70);         % defines the off-time characteristics

% build the pulse envelop
i=[rise on fall off];

% plot the i-samples and scale the plot
plot(i)
axis ([0 length(i) -2 2])

% set the q-samples to all zeroes
q=zeros(1,length(i));

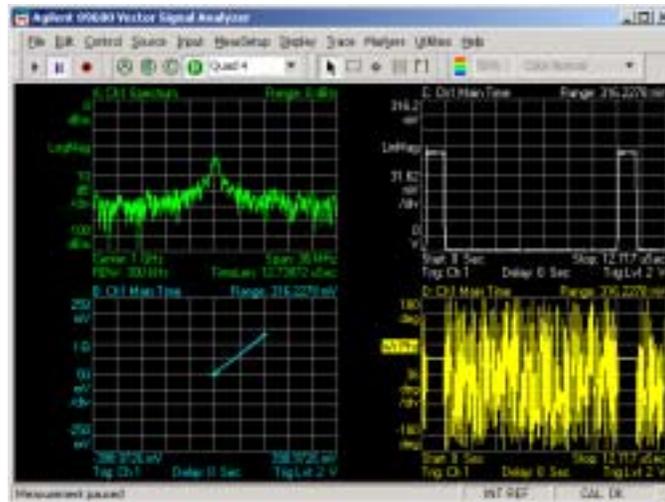
IQData=[i + (j * q)];
    
```



To generate the signal, we will build an array in Matlab that describes the In-Phase and Quadrature time domain waveforms and download the arrays into the signal generator. Shown here is a plot of a typical pulsed waveform and the code used to generate the waveform.

The file Pulse.m is a complete Matlab program to generate and download a simple pulse into the PSG. The source code for all of the programming examples used in this paper can be downloaded for free at the Agilent web site: [www.agilent.com/find/psg](http://www.agilent.com/find/psg).

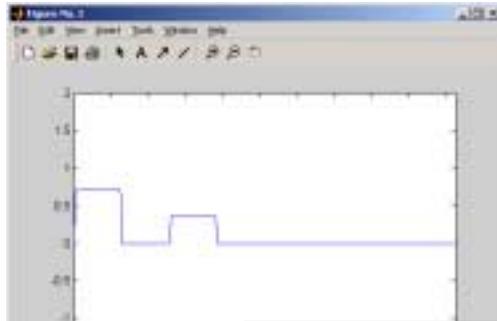
## Simple Pulse (Pulse.m)



This is a screen capture from the Vector Signal Analyzer to analyze the signal.

The user should be able to justify the displayed results on the analyzer with the Matlab program Pulse.m. The upper left hand plot shows the frequency spectrum of the signal. The upper right hand plot shows the linear amplitude time domain waveform. The lower left hand plot shows the IQ vector for the signal. The lower right hand plot displays the phase of the signal versus time. Observe linear amplitude plot and note that the amplitude of the signal is constant within a single pulse and between pulses. From the phase plot versus time; note that the phase of the pulse is constant within a single pulse and between pulses. This infers that the Vector PSG is coherent in frequency and phase with the Vector Signal Analyzer.

## Creating a Doublet



- Create a variable that defines the separation between pulses
- Create two separate pulses
- Concatenate the pulses, separation, and off-time

```
separation=zeros(1,128); % separation between the pulses
% define arrays which contain the pulse envelope for each pulse
pulse1 = [rise on fall];
pulse2 = .5*[rise on fall];
% concatenate and scale the pulses
i = .707*[pulse1 separation pulse2 off];
```

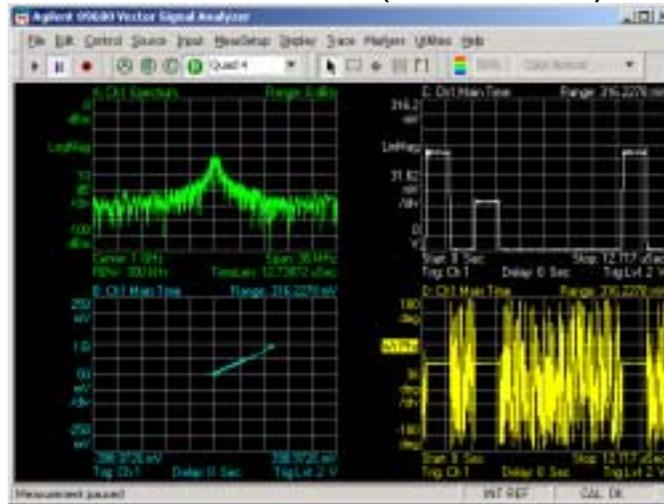
Note: Complete source code listing at [www.agilent.com/find/psg](http://www.agilent.com/find/psg)



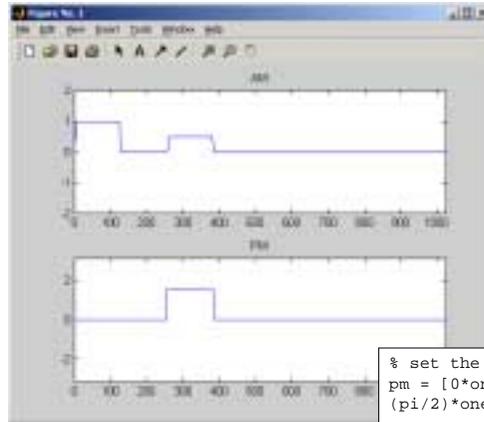
Having done the hard work of building a pulsed waveform and describing how each part of the program operates to create the signal, new waveforms may easily be built to stress some aspect of the system under test. The next signal is a pulse doublet, which are two pulses placed very close in time. When testing an ELINT system, a doublet is used to verify the ability of the system to correctly identify two closely spaced pulses rather than a single long pulse. Often a series of doublets will be created with a different amount of separation between each doublet. Also an amplitude change will be introduced between the pulses to place additional stress on the system. Shown is the plots of a pulse doublet.

Generating a pulse doublet using Pulse.m as a starting point is fairly straightforward. Refer to code above. Define three new variables: pulse1, pulse2, and separation. The variables pulse1 and pulse2 will contain the amplitude envelope of each pulse. The amplitude of each pulse can be scaled independently of the other. The variable separation will define the number of points (and the time) between the two pulses. Finally we concatenate the new pulses with separation between them.

## Pulse Doublet (Doublet.m)



## Creating Phase Shift Between Pulses



- Create an array that contains the desired phase for each pulse
- Convert AM and PM waveform to I and Q

Note: Complete source code listing at [www.agilent.com/find/psg](http://www.agilent.com/find/psg)

```
% set the phase of the two pulses
pm = [0*ones(1,length(pulse1)) separation
      (pi/2)*ones(1,length(pulse1)) off];

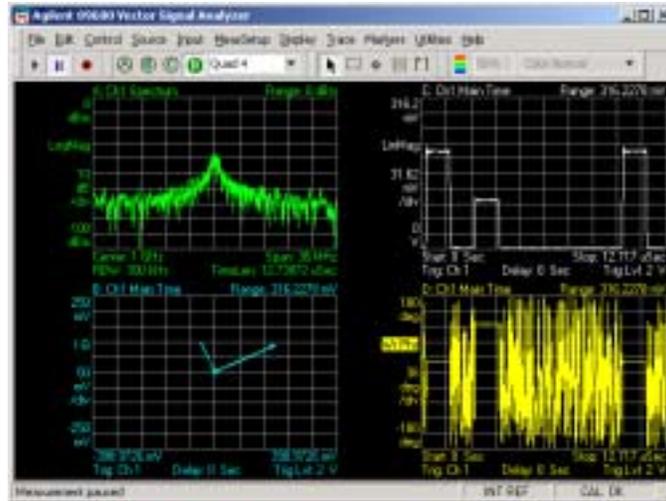
% convert am and pm to i and q
i = .707*am.* cos(pm);
q = .707*am.* sin(pm);
```



The next example will add a  $\pi/2$  phase shift (90 degrees) between the two pulses. Note that in the previous examples, the amplitude envelope was placed in the real array and zeros was placed in the imaginary array. For signals that require no phase or frequency modulation, that technique works fine and simplifies the math. For this example, the signals will be specified in terms of an am and pm array and converted into IQ.

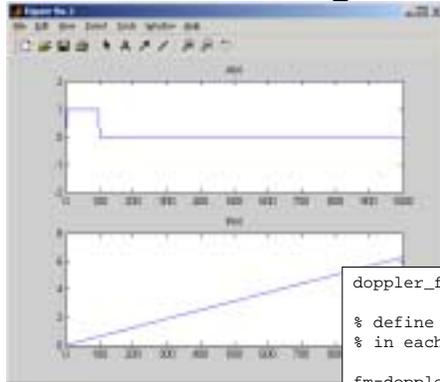
The array for pm contains a constant but different phase during the on-time of each pulse. Pulse1 is set to 0 degrees and pulse2 is set to  $\pi/2$ . Multiplying the am waveform times the sine or cosine of the pm waveform performs the IQ conversion. The period following the variable am instructs Matlab to multiply the arrays on an element by element basis.

## Pulse Doublet with Phase Offset (Phase\_Offset\_Doublet.m)



Note in screen capture the  $\pi/2$  phase shift between pulse1 and pulse2 shown in both the IQ plot and the phase versus time plot.

## Creating a Doppler Shift



- Create an array that contains the desired doppler frequency in each sample
- Integrate the frequency array to create a phase vs. time array

Note: Complete source code listing at [www.agilent.com/find/psg](http://www.agilent.com/find/psg)

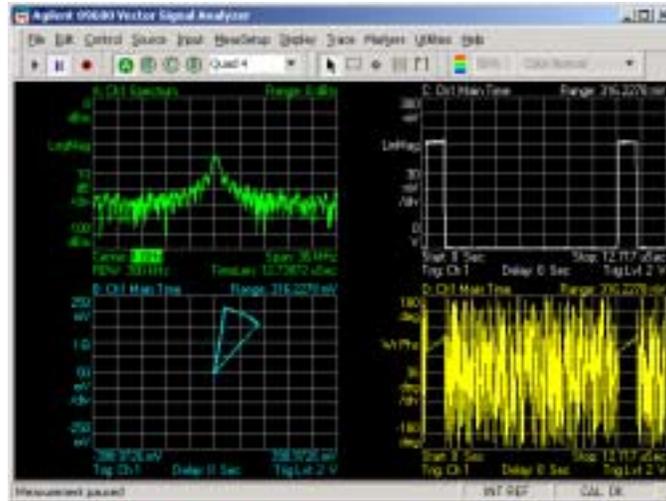
```
doppler_freq = 100e3;      % doppler freq in Hz
% define an array which contains the doppler freq
% in each sample
fm=doppler_freq*ones(1,length(am));
% use an integral to translate from fm to pm
pm=(2*pi/sampclk)*cumsum(fm);
% convert am and pm to i and q and scale amplitude
i=.707*am.* cos(pm);
q=.707*am.* sin(pm);
```



Creating a pulsed waveform with a constant Doppler frequency shift requires the introduction of a new technique. The idea is to build a waveform that contains the Doppler offset frequency from the carrier versus time, then integrate the waveform to produce phase versus time. Remember that phase is simply the integral of frequency. As shown above, an array *fm* is produced that contains a constant offset frequency in Hz. The length of the array is equal to the entire *am* pulsed waveform. The *fm* waveform is integrated using the function *cumsum*, which is the cumulative sum of the elements of the array. The new array must be scaled by  $2\pi/\text{sampclk}$  to obtain units of radians. The *am* and *pm* waveforms are converted into IQ and scaled for downloading into the signal generator.

Note that math shown in code allows the user to specify the Doppler offset frequency in Hertz versus time and, given the sample clock in Hertz, calculate the *pm* waveform. This is very powerful. While this example produces a static Doppler offset within a single pulse, the technique can easily be extended to produce a Doppler trajectory for a moving emitter. The primary limitation of the technique is the 32Mbytes of memory within the signal generator to play the waveforms. With 100MHz sample clock, the 32Mbytes of memory will allow the production of 32msec of unique signal.

## Pulse with Doppler Frequency Offset (Doppler.m)

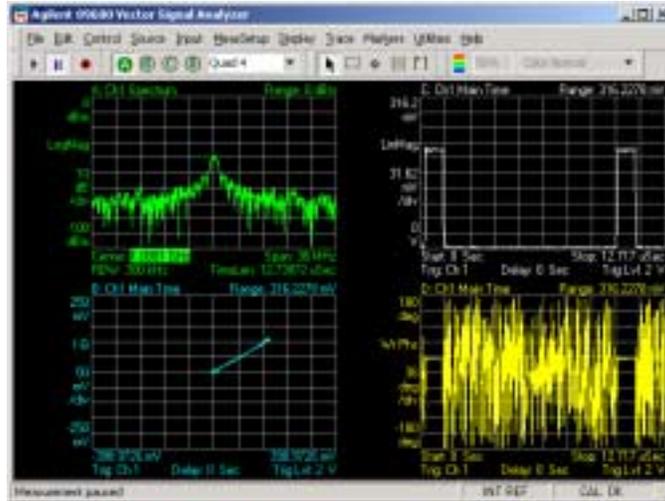


Now to discuss the two vector displays of the Doppler signal. Above, the vector signal analyzer is tuned to the exact center frequency of the signal generator. The phase versus time plot in the lower right hand corner shows a ramp in phase versus time during the on-time of the pulse. The IQ display in the lower left hand corner shows an arc of phase. From the parameters in the Matlab program doppler.m, the pulse width can be calculated to be 1 microsecond and the Doppler shift set to 100KHz. The phase shift generated by a 100KHz Doppler shift over 1 microsecond should be:

$$(100\text{KHz}) * (360^\circ/\text{cycle}) * (1\text{e-}6\text{sec}) = 36^\circ$$

Note that the displays show 36° of phase shift during the on-time of the pulse.

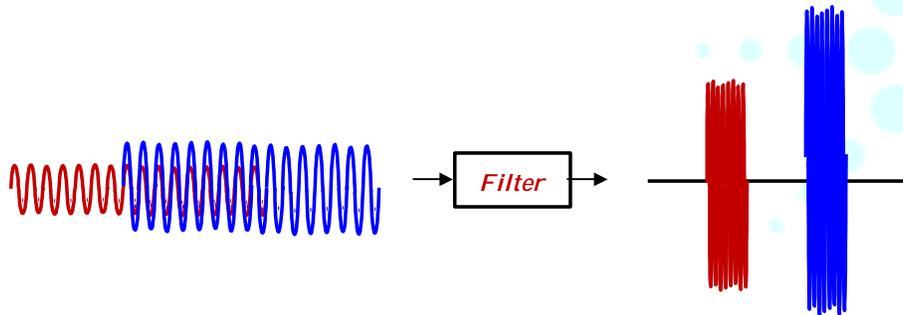
## Pulse with Doppler Frequency Offset (Doppler.m)



Above, the vector signal analyzer has been tuned to the Doppler offset frequency. Note that the phase of the pulse is constant and stable over at least two pulses. This infers the signal generator and arbitrary waveform generator are coherent with the vector signal analyzer. This also provides confirmation that the math used to calculate the Doppler waveform is correct.

## Pulse Compression: Frequency Chirp

*Overlapping returns can be separated.*

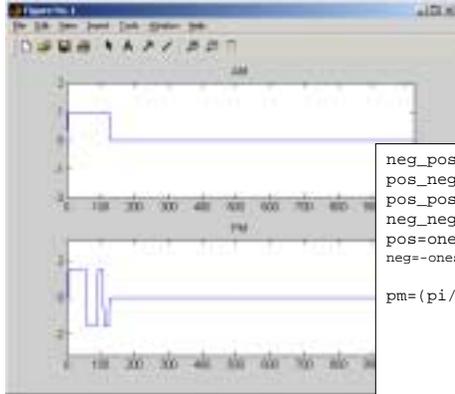


Having provided examples of simple pulsed waveforms with control of the amplitude and phase of the pulse, the next step will be to produce modulation within the pulse. Both phase and frequency modulation are used by radar systems to improve range and resolution. Radar systems that use modulation within the pulse are referred to as pulse compression radar systems.

The radar range equation points out a basic engineering trade-off between range and resolution and the need for pulse compression. To build a long range radar (or for the radar to 'see' a great distance) the radar needs high average output power. To obtain good resolution, the radar needs a narrow pulse that reduces the average output power. Pulse compression provides a path around this trade-off. Pulse compression radars will transmit a long pulse with modulation inside the pulse. The returns are processed through a filter that is matched to the characteristics of the modulation compressing the pulse in time. This compression allows the radar to separate overlapping returns while transmitting a high average power.

It is important for ELINT and radar warning receivers to correctly process these types of signals. The modulation type and deviation of the signals provide important information about the purpose and intent of an observed system. It is often difficult to obtain a signal source with the appropriate characteristics to verify the performance of the ELINT system.

## Creating a Barker Coded Pulse



- Create a sequence of waveforms that provide all possible transitions
- Concatenate the waveforms to produce the 7-bit binary phase code

```

neg_pos=(1+sin(ramp*pi/2))-1;
pos_neg=(1+sin(-ramp*pi/2))-1;
pos_pos=ones(1,4);
neg_neg=-ones(1,4);
pos=ones(1,13);
neg=-ones(1,13);

pm=(pi/2)*[0 0 0 ...
[rise pos]...           %Bit 1 high
[pos_pos pos]...       %Bit 2 high
[pos_pos pos]...       %Bit 3 high
[pos_neg neg]...       %Bit 4 low
[neg_neg neg]...       %Bit 5 low
[neg_pos pos]...       %Bit 6 high
[pos_neg neg]...       %Bit 7 low
rise-1 0 0 off];

i=.707*am.* cos(pm);
q=.707*am.* sin(pm);
    
```

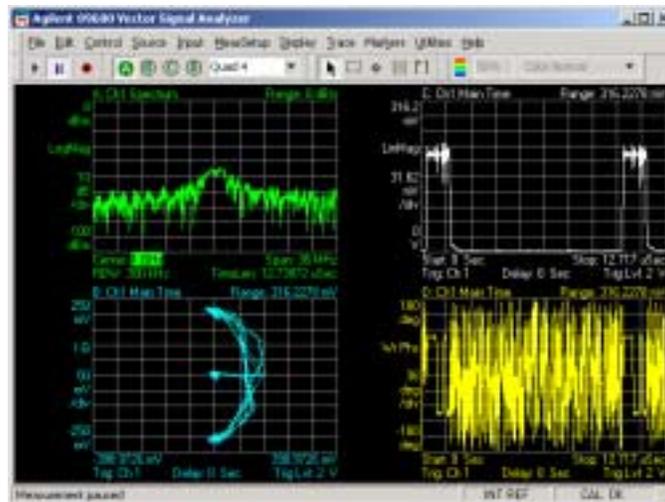
Note: Complete source code listing at [www.agilent.com/find/psg](http://www.agilent.com/find/psg)



Barker coded signals are typical in pulse compression radar systems. Barker codes are binary numbers containing between 2 and 13 bits that have a unique auto correlation functions. The points adjacent to the peak of the correlation function equal zero. This is very useful in a radar system since any spurious response can be misinterpreted as a target. A Barker coded pulse typically uses binary phase modulation. The 'chip' rate is the dwell time for each bit within the pulse. In this example, we will build a 7-bit Barker coded waveform. The 7-bit Barker code contains the bits [+1 +1 +1 -1 -1 +1 -1].

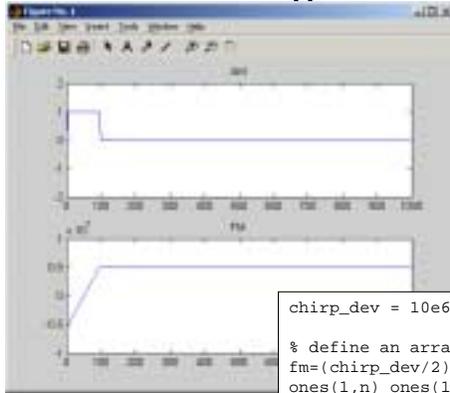
To build the phase waveform, the seven bits of information must correctly encode into a binary phase shift keyed waveform and deal with the speed of the phase transitions. The transition time between phase states will at least in part determine the occupied bandwidth of the signal. Within the program, first define the possible phase states and transitions as individual arrays, and then concatenate them into the final waveform. The two possible states the waveform can occupy are positive and negative, or +1 and -1. There are four possible transitions: negative to positive, positive to negative, negative to negative, and positive to positive. The states are built with a constant value over the 'chip' period. The transitions are built using raised cosine functions. Note that the array rise was built as part of the am array, but it is used here as a 0 to 1 phase transition. At the end of the code sequence, the function rise-1 is used to provide a -1 to 0 transition. Having constructed the array with +1 and -1 states, multiple the waveform by pi/2 to provide the appropriate phase deviation. The resulting waveform is converted to IQ and downloaded into the signal generator.

## Pulse with Barker Code (Barker.m)



Above is the demodulated signal. The lower right hand plot displays the demodulated phase versus time. Note that during the on-time of the pulse, the seven bits of the code are clearly visible. Markers may be used to verify the phase state accuracy and timing. The noise in phase between the pulses is due to the fact that during the off-time of the pulse the phase of the signal is undefined.

## Creating a Linear FM Chirp



- Create an array that contains the Linear FM Chirp
- Integrate the FM waveform to produce a PM waveform
- Convert the AM/PM waveforms into I and Q

Note: Complete source code listing at [www.agilent.com/find/psg](http://www.agilent.com/find/psg)

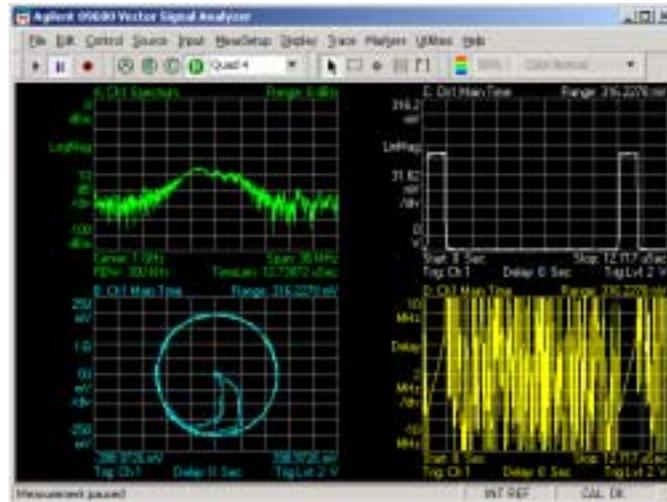
```
chirp_dev = 10e6; % total chirp deviation in Hz
% define an array which contains the chirp waveform
fm=(chirp_dev/2)*([-ones(1,n) (-1:2/(length(on)-1):1)
ones(1,n) ones(1,length(of))]);
% use an integral to translate from fm to pm
pm=(2*pi/sampclk)*cumsum(fm);
% convert am and pm to i and q and scale amplitude
i = .707*am.* cos(pm);
q = .707*am.* sin(pm);
```



The final type of signal to demonstrate is a linear FM chirp. The objective is to build an fm waveform that will linearly sweep the frequency across a know deviation. This example uses the technique introduced in the Doppler example that integrates the fm waveform to produce a pm waveform which can be converted over to IQ. The advantage of this process is it allows the user to build waveforms in frequency versus time to create arbitrary fm waveforms that can be converted and downloaded into the signal generator.

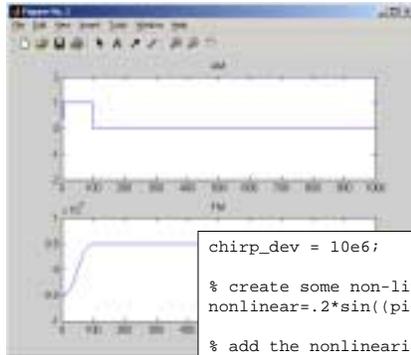
In the Matlab plot the fm chirp is built using a ramp that starts at exactly -1 and ends at exactly +1. This allows the user to easily scale the waveform by multiplying by the desired frequency deviation divided by 2. Note that the frequency will sweep both above and below the carrier frequency. Using the internal arbitrary waveform synthesizer, the signal generator can produce up to an 80MHz chirp. To eliminate an unnecessary frequency step at the beginning and end of the chirp, the fm waveform is held at the frequency of the chirp endpoints during the rise-time and fall-time of the pulse.

Pulse with Linear FM Chirp (LFM\_Chirp.m)



Note the demodulated fm waveform in the lower right hand plot. The chirp is linear and the deviation is equal to the 10MHz defined in the program (+/- 5MHz). The values in the fm demodulator are only defined during the on-time of the pulse. In the upper right hand plot, note that the amplitude of the pulse is flat during the pulse on-time. Because the signal is being swept across a 10MHz frequency span, this indicates the IQ modulator within the signal generator provides a flat frequency response across that bandwidth. The spiral effect in the IQ plot in the lower left hand corner is due to the frequency offset from the carrier during the rise and fall time of the pulse.

## Adding Non-Linear element to FM Chirp



- Create an array that contains the Linear FM Chirp and add sine wave to produce non-linear chirp
- Integrate the FM waveform to produce a PM waveform
- Convert the AM/PM into I and Q

Note: Complete source code listing at [www.agilent.com/find/psg](http://www.agilent.com/find/psg)

```

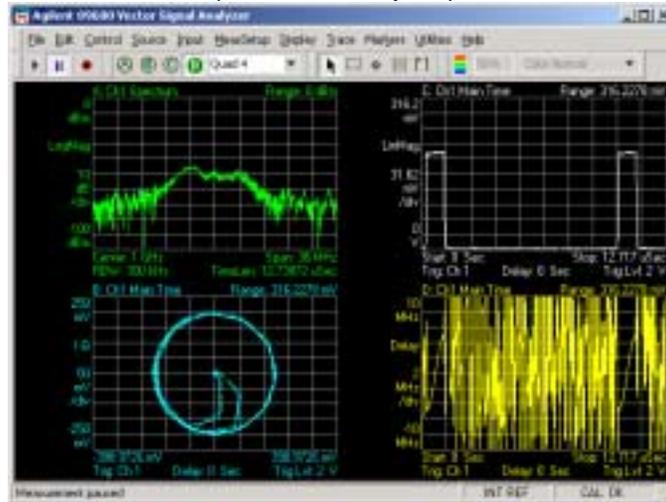
chirp_dev = 10e6; % defines the total chirp deviation in Hz
% create some non-linear distortion to add to the chirp
nonlinear=.2*sin((pi)*(-1:2/(length(on)-1):1));
% add the nonlinearity to the chirp and concatenate the sections
fm=(chirp_dev/2)*([-ones(1,n) nonlinear+(-1:2/(length(on)-1):1)
ones(1,n) ones(1,length(off))]);
% use an integral to translate from fm to pm
pm=(2*pi/sampclk)*cumsum(fm);
% convert am and pm to i and q and scale amplitude
i=.707*am.* cos(pm);
q=.707*am.* sin(pm);
    
```



The final example will demonstrate how to add a known amount of non-linear distortion to the fm chirp waveform. The non-linear distortion will be produced by the single cycle of a sine wave scaled to fit within the on-time of the pulse. The amount of non-linearity is set by scaling the amplitude of the sine wave as some percentage of the total deviation. Because the value of the sine wave is zero at its end points the maximum deviation of the chirp will not change. The resulting S-shape of the waveform is typical of the distortion seen in non-synthesized chirped signals.

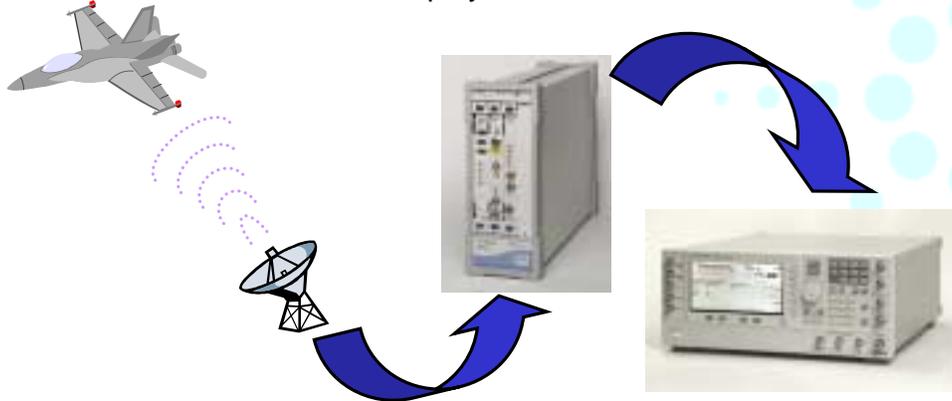
# Aerospace Defense Symposium 2003

## Pulse with Non-Linear FM Chirp (NLFM\_Chirp.m)



## Record and Download Waveforms

- Take your waveform simulation to the next level...
  - Record signals off the air with the VSA and transfer them into the PSG for playback



The cost of operating a radar test range with live targets is incredibly expensive. By recording signals and playing them back on the PSG in the lab you save time and money. The recorded signals are limited to the 36 MHz bandwidth of the VSA or the 80 MHz modulation bandwidth of the PSG if you use the Infiniium oscilloscope wide band configuration.

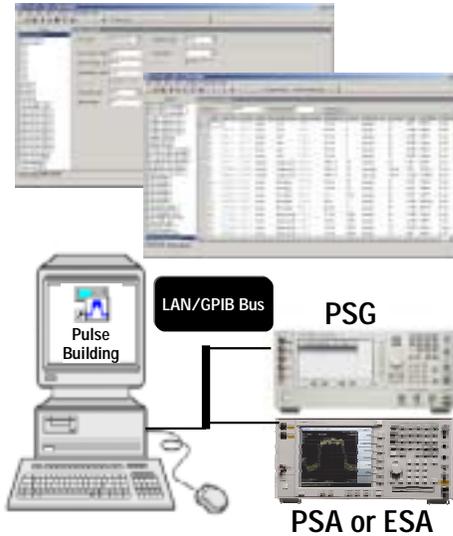
## Signal Studio for pulse building software

### Features

- Easily navigate the intuitive user interface
- Create a pulse library
  - Construct custom pulse shapes
  - Apply intra-pulse modulation
- Build a pattern library
- Apply baseband pre-distortion
  - Improve image rejection
  - Optimize RF modulation flatness
- Automate using the COM-based API
- Utilize extensive built-in Help

### Value

- Set high-level pulse parameters
- Eliminate complicated mathematics
- Simplify single-emitter test pattern generation
- Enhance signal quality

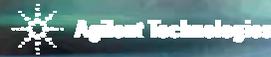
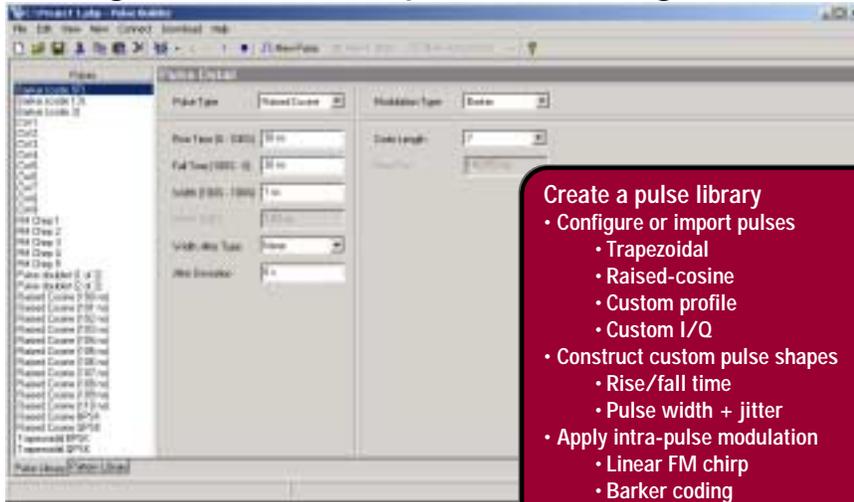


For many common applications Agilent's Signal Studio for pulse building software eliminates the need to write custom software programs. Signal Studio for pulse building provides a Graphical User Interface that makes it easy to create complex pulsed signals without detailed knowledge of the math involved in the creation of the signals. It also provides simple CW pulses with shaping and complex phase and frequency-encoded signals. The user can easily build pulses and pulse patterns to simulate complex emitters. Once created, the signals can be saved on the signal generator's internal hard disk drive and recalled later without an external pc or software.

To simplify this testing more we are offering a Signal Studio application targeted at single-emitter radar test pattern generation. Utilizing this application, the mathematics required to calculate the I/Q waveform samples will be transparent to the user. By simply setting a few high-level pulse parameters... like rise time, ramp profile, intra-pulse modulation, etc.... even the novice user can create complex radar pulses. The software will also simplify test pattern generation by providing an interface to setup custom pulse sequences.

When combined with an Agilent ESA or PSA spectrum analyzer, the Signal Studio for pulse building software applies pre-distortion to the calculated waveform to improve image rejection and RF flatness. The spectrum analyzer is required to gather the calibration data need to determine correct pre-distortion coefficients. This process is fully automated by the software. Corrections can also be applied to custom I/Q user data that has been imported to the software.

## Signal Studio for pulse building software



Using the Signal Studio for pulse building software, a library of custom pulse shapes can be defined in the time and frequency domain using four techniques:

- Select a pre-defined trapezoidal or raised cosine pulse type and set rise/fall time and pulse width
- Select custom pulse profile and enter amplitude points in the table editor to define the pulse profile
- Select custom I/Q and enter I/Q waveform samples in the table editor to define the pulse profile
- Select custom profile or custom I/Q and import ASCII user-file pulse definitions of pulse signal recordings or pulses created in other signal creation environments, such as Agilent ADS and Matlab®

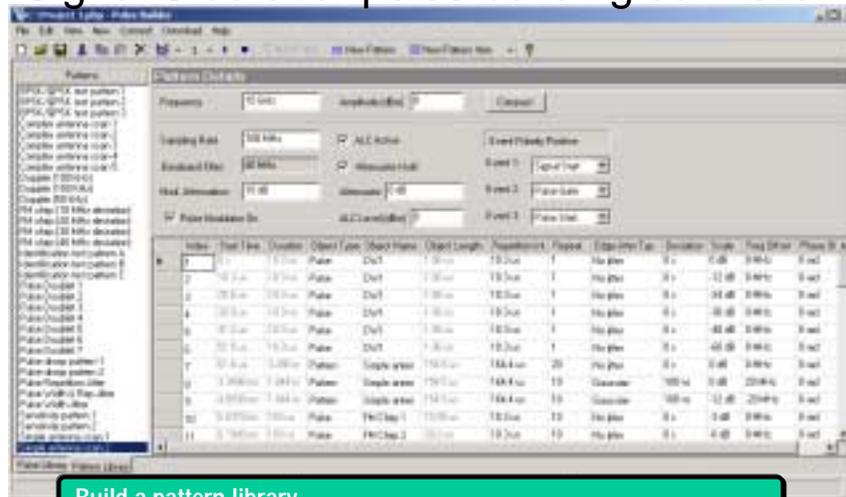
Signal Studio for pulse building utilizes the arbitrary waveform playback capability and fast sample rate of the PSG internal baseband generator to achieve flexible pulse shaping with up to 10 ns resolution between waveform sample points.

The software simplifies applying complex pulse compression, relieving the user of complicated mathematics. To apply intra-pulse modulation to any pulse type, simply select from a pull-down menu of modulation formats:

- AM step- set amplitude offset and step size
- Barker – select from seven different Barker codes (2, 3, 4, 5, 7, 11, 13)
- BPSK – alternating one-zero (0° and 180°) bit pattern with user-defined step size
- Custom BPSK – user-defined bit pattern (0° and 180°)
- FM Chirp – user-defined FM deviation up to ±40 MHz
- FM step – set frequency offset and step size
- QPSK – user-defined step size with symbols at 45°, 135°, 225°, 315°

Pulses configured using the Pulse Builder software, imported from other signal creation environments, or imported from signal recordings can be stored in the pulse library and recalled at will to creating custom pulse patterns. This provides unrivaled flexibility when constructing complex pulse patterns for receiver test.

## Signal Studio for pulse building software



### Build a pattern library

- Create and store custom pulse patterns by sequencing unique pulses
- Nest pulse patterns in other pulse patterns



Assemble custom pulse patterns using the pulses available in the pulse library. Once a pulse pattern has been created, it can be used in other pulse patterns as well. For example, a pattern that consists of a pulse doublet can be created and nested into another pattern. Additionally, pattern parameters such as repetition interval, frequency and phase offsets, amplitude scaling, etc., can all be easily applied to the pulse doublet pattern, just like configuring a single pulse. When creating a pattern, the following parameters can be set:

- Pattern item type – pulse, pattern, or off-time
- Repetition interval of pulses/patterns
- Gaussian or uniform repetition interval jitter with user-defined deviation
- Number of repetitions of pulses/patterns
- Amplitude scaling, frequency offset, and phase offset of pulses/patterns

This level of pattern definition flexibility, combined with the pattern nesting capability, simplifies the configuration of complex pulse patterns; including antenna scan patterns, frequency agile pulse pattern (within 80 MHz RF modulation bandwidth), and receiver sensitivity/selectivity test patterns.

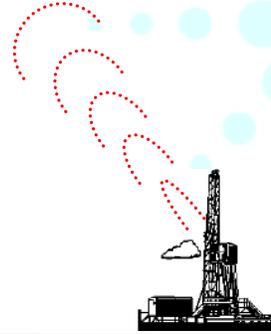
Signal start, pulse start, and pulse gate marker and trigger signals can be applied to any pulse pattern. User-defined routing allows each marker/trigger signal to be independently applied to the event 1, event 2, and event 3 ports of the PSG vector signal generator.

The software is not intended for target simulation, however, if you are working on this application, the custom I/Q pulse definition does provide the ability to download custom pulse shapes to the software for playback. Keep in mind that you will also benefit from the calibration/corrections that the software provides to improve RF flatness and image rejection even when working with custom I/Q signals. We are working on enhancing the PSG triggering capability associated with waveform playback to further enhance the software for this application and are investigating the requirements to achieve phase coherency.



## Deep memory applications

- Long radar target generation scenarios
- Evaluate radar receivers and signal processors
  - Maneuvers
  - Doppler
  - Multiple targets
- Test range recordings or develop with software (e.g. **MATLAB®**)



For many applications more memory is needed for the baseband signal than can be stored on the internal memory of the arbitrary waveform generator. For instance, field testing is a costly effort when developing radar and EW systems. If the emitter scenario encountered in the field can be simulated using software tools or better yet recorded in the field, it can be played back through a vector signal generator in the lab as often as needed no matter how long it is.

Pushing this simulation task onto off-the-shelf equipment such as a computer and an arbitrary waveform generator can cut the time and cost of developing a specialized simulation tool to a fraction of what it would take to create a custom solution.

## Hard drive streaming for waveform playback

### Typical problem

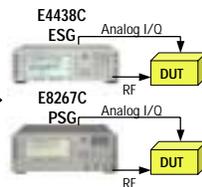
Baseband generator waveform playback memory is insufficient to play real world simulations to test advanced RF/MW systems and receivers

### Why?

- Need to playback long time records of unique test signals
  - complex simulation scenarios
  - test custom, non-repeating data sets for BER/PER/FER analysis
  - long signal recordings captured from antenna or DUT

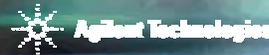
### Solution

Baseband Studio  
for waveform streaming



### Target applications

- RADAR and EW system test
- Satellite receiver test
- Software Defined Radios
- WLAN receiver test
- Secure communications transceiver test



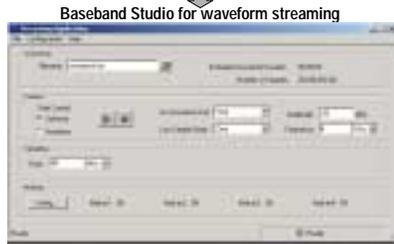
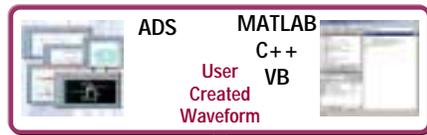
**Challenge:** The following test cases require long, unique complex modulated test signals that typically result in a waveform that is too large to fit in the largest memory available from the ESG/PSG internal baseband generator:

- Aerospace and defense customers have a need for minutes, to hours worth of unique test signals to address RADAR and EW receiver and system level tests that simulate multiple flight, threat, and target scenarios. These same customers typically need to playback long waveforms captured from antennas or design simulation software.
- The wireless communications and satellite customers have a need to playback complex design simulations and perform receiver BER analysis.
- With the development of the new digital communication systems, the use of pre-distortion, proprietary clipping, and multi-carrier signals for testing in the cellular communications industry has led to increased test time required for complex waveform calculations and repetitive file transfers. Complex waveform calculations are unavoidable; however, waveform streaming can reduce test time by eliminating load times from storage to playback memory.

In order to playback these long waveforms with any of the industry-leading arbitrary waveform generators, customers are forced to segment the waveform, then load and play one section at a time. This approach is difficult and costly to do it right.

**Solution:** Baseband Studio for waveform streaming provides a link between virtually unlimited arbitrary waveform memory space on the PC hard disk drive, and the baseband generator in the ESG/PSG. The long test scenarios I described can be streamed in real time directly to the signal generator's baseband generator and up-converted to RF or MW frequencies. In most cases this will completely eliminate the need for segmented waveforms. This tool also eliminates much of the need for expensive field testing and custom simulator development since long simulation scenarios emulating the actual receiver and system environment can be used to develop and verify your design. With virtually unlimited memory, up to hours of unique signal can be played back through the ESG/PSG and the user is free from memory restrictions associated with traditional arbitrary waveform generation.

## Baseband Studio for waveform streaming



### Key Features

- **Continuous I/Q waveform streaming**  
(direct from PC HDD to ESG/PSG)
- **Eliminates waveform download time**
- **Virtually unlimited waveform length**  
(scaleable with disk drive technology)
- **Up to 40 MSa/s & 32 MHz RF modulation BW**  
(scaleable with disk drive & PC technology)
- **Up to 25 ns resolution for pulse shaping**
- **Set playback characteristics**
  - Playback once, N times, or infinitely
  - Playback when triggered
- **Selectable waveform format**
  - 16-bit I/Q
  - 15-bit I/Q w/ 2 markers
  - 14-bit I/Q w/ 4 markers
- **Supports external HW and SW triggering**
- **Automation using the Microsoft .NET API**



The features of Baseband Studio for waveform streaming include:

**Continuous I/Q waveform streaming** – Long time records of signals created and formatted in common signal creation environments, including Agilent ADS, MATLAB, and other system design tools are easily played back using Baseband Studio for waveform streaming provided the waveform is created in the correct file format.

**Long, unique waveforms** - Waveform streaming from a PC HDD enables virtually unlimited I/Q waveform file size. The only limitation is the space available on your PC hard drive or RAID configured drive array.

**High sample rate and wide BW** - Sample rates of up to 40 MSa/s can be achieved. This translates into 16 MHz of bandwidth for each of the I and Q channels and a total of 32 MHz RF modulation bandwidth. This wide bandwidth enables simulation of multiple channels of communication signals and generation of pulsed waveforms.

**Streaming control** - Full control of the streaming process is provided from the software's GUI or API. The stream can be started, stopped, looped a specified number of times, or played continuously. When the waveform ends the user can select whether the terminal value should hold the value of the last sample or be set to zero.

**Marker functionality** - Output markers enable synchronized control and triggering at a specific point or portion of your waveform. Support for 0, 2, or 4 software markers is provided. Markers are embedded in the waveform file by using bit 0 and bit 1 of the I/Q pair.

**Utilize the Microsoft .NET-based API** - Use the Microsoft .NET-based application programming interface (API) to configure and launch hard drive streaming directly from your test executive rather than from the Baseband Studio for waveform streaming graphical user interface. The entire signal configuration and streaming process can be automated in your own programming environment. The software's built in Help system provides .Net programming examples. These examples can be easily leveraged to minimize the programming and automation learning curve.

Certainly, the configuration of the PC and associated hard drive will have a major impact on the performance of the streaming capability. Careful consideration of the end-to-end throughput of the data path is important when selecting the PC and hard drive system.

## Conclusion

- Radar emitters can be simulated using vector signals
  - Replacement of expensive custom systems
  - Waveform creation is quick and repeatable
  - Full control over pulse and modulation parameters
  - Waveform streaming allows for long emitter scenarios
- Agilent has the tools to create, generate and analyze radar waveforms
  - PSG Vector Signal Generator
  - I/Q files created using Matlab are easily downloaded into the PSG using Agilent Download Assistant
  - Signal Studio for pulse building
  - VSA Vector Signal Analyzer
  - Baseband Studio for waveform streaming



Thank You for your Attendance!!!



Thank you for attending today.

Find more information about the Agilent PSG signal generators at:  
[www.agilent.com/find/psg](http://www.agilent.com/find/psg)