Measurement Techniques for Radar and Electronic Warfare Applications

John Hansen
Senior Application Engineer
Agilent Technologies
Presentation Overview

• Pulse Compression Radar
• Time Side Lobes
• The Time Side Lobe Level Measurement
  • Windowing Functions
  • Applying the Time Sidelobe Method
  • Measurement Results
• Waveform Creation
  • SystemVue
  • Signal Studio for pulse building
  • MATLAB
• Math Functions in the Agilent Vector Signal Analyzer (VSA)
• User Defined Functions in Oscilloscopes
• Use of MATLAB (example)
Pulse Compression Radar

**Transmitter**
- VSG/AWG (Ideal Tx or Target)
- Waveform Exciter (Digital Synthesizer)
- Pulse Modulator
- Timing Sync
- CHIRP (SPREAD SPECTRUM)

**Receiver**
- Receiver Protector
- I/Q Detector
- Pulse Compression Filter (Correlation Filter)
- IFA
- LNA
- VSA (Ideal Rx)

To Signal Processor

**Anticipate __Accelerate __Achieve**

Agilent Technologies in Aerospace & Defense © 2012 Agilent Technologies
Why Pulse Compression?

- Lower power transmitter
- Smaller and lighter hardware
- Less power consumption
- Better reliability
- Better range resolution through the use of wide bandwidth pulses
- Lower probability of intercept of transmit signal
- Linear FM chirp commonly used
What are Time Side Lobes?

- Also called range side lobes
- Artifact of performing FFT on a non-periodic signal in a finite time capture
- Made worse by system anomalies and imperfections

PPI Radar Display

Time domain representation after compression
Filter Mismatch in the Receiver

- Filter Mismatch reflections in the receiver may cause ghost echo returns
- Example of how individual components can affect system performance
Windowing Functions

- Also called Weighting Functions
- Applied to suppress Time Side Lobes
- Side effects: S/N loss

Commonly used windowing functions and their attributes:

<table>
<thead>
<tr>
<th>Windowing Function</th>
<th>Peak Sidelobe Level (dB)</th>
<th>S/N Loss (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>-13.2</td>
<td>0</td>
</tr>
<tr>
<td>Hamming</td>
<td>-42.8</td>
<td>1.34</td>
</tr>
<tr>
<td>Hann</td>
<td>-32</td>
<td>1.4</td>
</tr>
<tr>
<td>Blackman</td>
<td>-58</td>
<td>2.37</td>
</tr>
<tr>
<td>Blackman-Harris (3 term)</td>
<td>-67</td>
<td>2.33</td>
</tr>
</tbody>
</table>
Hamming Windowing Function and Associated Frequency Response

- Windowing can be applied in either the time or frequency domains with equivalent results
Time Side Lobe Level (SLL)

\[ F(t) = A \cos \left( \omega_c t + \frac{1}{2} ut^2 \right) \]

Time Side Lobe Level (SLL) is a quality metric

Linear FW Chirp Pulse

Correlation Filter

Compressed Pulse

Compressed Pulse Width

SLL
Test Set Up

PC with MATLAB

M8190A 2-Channel AWG
5 GHz analog BW per channel

E8267D PSG vector signal generator
WB IQ modulation to 2 GHz BW, 44 GHz carrier

10 MHz ref

N9030A PXA signal analyzer
Up to 50 GHz
900 MHz BW IF output
89600B VSA software

90000 X series Oscilloscope
80 GSa/s; 33 GHz
MATLAB User Defined Functions
89600B VSA software

System Under Test

PCle
User Math Functions within the 89600 VSA Software

Math functions create mathematical expressions that operate on VSA trace data. Use math functions to:

- Perform mathematical operations on trace data
- Create a mathematical expression that you can apply as a filter to a waveform
- Manipulate data in the data registers
Use of VSA Math to Perform Frequency Domain Correlation of Measured vs. Ideal

For SLL:

Use time cross correlation identity:

\[ \text{Meas}(t) \times \text{Ideal}(t) = \text{ifft} \left[ \text{Meas}(f) \times \text{conj} \left[ \text{Ideal}(f) \right] \right] \]

where

\[ \text{Meas}(f) = \text{window} \times \text{fft} (\text{Meas}(t)) \]

\[ \text{Ideal}(f) = \text{window} \times \text{fft} (\text{Ideal}(t)) \]

\[ \text{Ideal} = \text{Waveform saved in data register} \]

\[ \text{window} = \text{Hanning window} \]

Ideal waveform mathematically created in MATLAB and loaded into a VSA data register
SLL Measurement Result

- **Ideal SLL**
- **Measured SLL**
- **Measured FM**
- **Ideal FM**
Time Side Lobe Level and Nonlinear FM Chirps

- Eliminates the need for time side lobe suppression and windowing
- Nonlinearity characteristics adjusted to suppress side lobes based on system anomalies and operating conditions
- No loss in S/N

Disadvantages
- Much higher level of system complexity
- Different FM chirp for each pulse used by the system
Coded Pulse Compression

Performance metrics available:

• Side Lobe Level (SLL)
• Error Vector Magnitude (EVM)
• Flexible Vector Demodulation
Barker Phase Coded Pulse Compression

- The long pulse is divided into shorter phase coded sub-pulses
- Barker codes are chosen for their optimal low side lobe levels
- Other coding schemes are available such as Frank codes and polyphase codes
- The more bits the better the side lobe suppression (-22.3 dB for a 13 bit Barker code)

13 bit Barker code:

1111100110101

Time Side Lobe Level (SLL) is a quality metric

SLL = 1/13
= -22 dB

BPSK Signal

Correlation Filter*

Compressed Pulse

*Performed in VSA
BPSK Modulation

- Constant envelope
- Two phase states (0 & 180\degree)
- One bit per symbol
- 13 symbols (13 bit Barker)
- Symbol rate = 13 / PW
EVM of a Barker Coded Pulse

- Carrier Lock
- Symbol Lock
- Filter
- Demodulate the bits
- Construct Ideal Reference from bits then use for SLL measurement
Side Lobe Level of a Barker Coded Pulse

Time cross correlation identity using VSA math function:

\[ \text{Meas}(t) \otimes \text{Ref}(t) = \text{iff}t [\text{Meas}(f) \ast \text{conj} [\text{Ref}(f)]] \]
Other Automation Tools within the 89600B Vector Signal Analysis Analysis Software
89601B VSA Software Automation
Macros

- Macros program the VSA using the C# and VB compilers that are shipped as part of the .NET framework.
- For a better programming experience, we recommend that you use Visual Studio 2010. Express editions are freely available from the Microsoft Visual Studio Express website.

What can macros be used for?

- One-button applications
- Automation of repetitive tasks
- Computation of measurement results that are beyond the scope of the basic 89600B VSA
Macro Example

3 GHz LFM Chirp Results with M8190A AWG IF and Analog PSG with option H30
SCPI Command Set for 89600B VSA Software

• Now available with the 89600B version of the VSA software

• Enables SCPI programming for easy automation of the measurement features included in the VSA software

89600B SCPI interface configuration tool
Pulse Signal Creation Alternatives
Three Tools for Pulsed Waveform Creation

**SystemVue**
- System modeling tool with Radar library
- Enables scenario modeling by adding targets, clutter, fading, noise and interferers

**Signal Studio for pulse building**
- Custom pulse shaping, modulation, antenna patterns, and user-defined pulse patterns
- Straightforward graphical user interface or with your own test executive using the COM-based API

**MATLAB**
- Extends the capabilities of Agilent signal analyzers and generators to create arbitrary waveforms, control instruments, make custom measurements, visualize data and build test systems
- Provides interactive tools and command-line functions for data analysis tasks
SystemVue

**Open Modeling**
- Existing Modeling Templates
- Custom Models: C++, .m, HDL
- Model Import: MATLAB, ADS, SignalStudio, VSA, STK
- Recorded Data

**HW Implementation**
- Existing Modeling Templates
- DSP Algorithm Creation
- Fixed Point Simulation
- HDL Code Generation
- FPGA Synthesis

**Simulation**
- Integration capability
- Connection to leading RF EDA flows
- Performance Evaluation for RF & BB
- Advanced Measurement
  - Detection Rate, False Alarm Rate
  - Dynamic Range
  - Parameter Estimation

**SystemVue**
- Advanced Dataflow engine
- Co-Simulation
- Model Libraries
- Integration of SW, HW
- HDL Simulation
- FPGA Implementation

**HW Test**
- Link to VSG/VSA/Scope/LA
- Integration/Controlling/Automation
- Custom Waveform Generation
- Advanced RF & BB Measurements
- Parameter Estimation
- Troubleshooting
SystemVue and the radar library (W1905) capabilities and features

**Signal Sources**
- LFM
- NLFM
- Barker/Frank Coded
- UWB Source
- FMCW

**T/R Modules**
- DAC, DUC
- DDS
- LNA
- DDC
- ADC
- Digital T/R

**Antenna**
- Antenna Models
- Antenna Array
- Antenna Propagation

**RF/IF Modules**
- Transmitters
- Receivers
- Filters, PA, Oscillators

**Radar Environment**
- Target
- RCS
- Clutter (1D & 2D)
- Jammer
- Interference

**Signal Processing**
- Digital Pulse Compression
- Moving Target Indication (MTI)
- Moving Target Detection (MTD)
- Constant False Alarm Rate (CFAR)
- Digital Beamforming
- Space-Time Adaptive Processing (STAP)

**Measurements**
- Waveform
- Spectrum
- Sensitivity
- Selectivity
- Dynamic Range
- Detection Rate
- False Alarm Rate

**Target Environments**
- RCS, Clutter, Jamming, Interference
Multi Radar Emitter Environment with SystemVue

8 Barker-coded Radar Signals + 8 LFM Chirp Radar Signals

Download Waveform to M8190A AWG

Multi Radar Environment

DSA91304A 13 GHz Oscilloscope

Preliminary work-in-progress
Demo: Measured Waveform on DSA91304A and VSA

8 Barker-Coded Radar Waveforms

8 LFM Chirped Radar Waveforms

Preliminary work-in-progress
Signal Studio for Pulse Building
Now Supporting M8190A and X-Series Signal Generators

Features
• Create a pulse library
  • Construct custom pulse shapes & Modulation
• Build complex pattern libraries
• Takes advantage of waveform sequencing to enable long scenario simulation
• Apply baseband pre-distortion
  • Improve image rejection
  • Optimize RF modulation flatness
• Automate using the COM-based API or import/export to threat database
• Implement Antenna Scan Modulation

Hardware Support
• E8267D – PSG Signal Generator
• E4438C – ESG Signal Generator
• N5182B & N5172B - X-Series Signal Generator
• M9330A, N8241A Arbitrary Waveform Generators
• M8190 Arbitrary Waveform Generator
Generating Radar Signals with MATLAB and Downloading to Agilent Hardware

- Waveforms defined in MATLAB can be easily downloaded to Agilent instruments
  - Agilent arbitrary waveform generators, such as the M8190A, can generate high bandwidth signals
  - Instrument Control Toolbox provides functions to download the waveforms

- Waveforms can also be downloaded to Agilent vector signal generators using the free utility: [www.agilent.com/find/downloadassistant](http://www.agilent.com/find/downloadassistant)
Time Side Lobe Level Measurement using an Oscilloscope and MATLAB User Defined Functions
Creating Radar Signals from Mathematical Definition

- Mathematical definition of the ideal signal generated from the transmitter module is required to calculate time side lobe measurements
- MATLAB® provides functions to easily define the ideal signal

```matlab
% Definition of ideal LFM chirped signal
% Define parameters of an ideal chirped LFM pulse to be sent from the TX-module
sampleRate = 8e9; % Define sample rate of ideal signal in Hz
pulsewidth = 10e-6; % Define pulse width in seconds
pulseRepetitionInterval = 50e-6; % Define PRI seconds
bandwidth = 10e6; % Bandwidth of the pulse in Hz
% Calculate ideal chirped LFM signal
tVector = 0:1/sampleRate:pulsewidth; % Vector to generate ideal pulse
IComponent = chirp(tVector,-bandwidth/2,tVector(end),bandwidth/2,'linear');
QComponent = chirp(tVector,-bandwidth/2,tVector(end),bandwidth/2,'linear');
IQData = IComponent + 1i*QComponent;
% Normalize amplitude
scale = max(max(abs(real(IQData))), max(abs(imag(IQData))));
idealPulse = IQData / scale;
```

- Any waveform that can be mathematically defined can be created using MATLAB
Creating Channel and Multipath Effects in a Waveform Obtained from a Simulated Receiver Module

When designing Radar receiver processing algorithms, it is necessary to model channel and multipath effects on the waveforms. We can do this easily using the built in functions in MATLAB.

```matlab
%% Simulate a pulse received at the RX-module corresponding to the pulse transmitted from the TX module
% Create a pulse that contains overlapping receptions
shortOffset = zeros(1, 1e-6*sampleRate); % time offset of 1e-6 seconds
longOffset = zeros(1,10e-6*sampleRate); % time offset of 10e-6 seconds
offsetAttenuatedPulse1 = [10^-3/20]*idealPulse shortOffset; % reflection w/ -3dB gain
offsetAttenuatedPulse2 = [shortOffset 10^-3/20]*idealPulse; % reflection w/ -20dB gain
overlappingPulse = offsetAttenuatedPulse1 + offsetAttenuatedPulse2;

% Create a received pulse that contains direct reflections, and overlapping reflections
receivedWaveform = [longOffset ... idealPulse ... % reflection w/ 0dB gain
longOffset ...
10^-10/20]*idealPulse ... % reflection w/ -10dB gain
longOffset ...
overlappingPulse ...]
longOffset];

% Plot the simulated received waveform
figure(3); axisHandle1 = subplot(2,1); hold on
plot([1:length(receivedWaveform)]/sampleRate, real(receivedWaveform), 'b', 'linewidth',2)
plot([1:length(receivedWaveform)]/sampleRate, imag(receivedWaveform), 'g', 'linewidth',2)
plot([1:length(receivedWaveform)]/sampleRate, abs(receivedWaveform), ':r', 'linewidth',1)
title('Simulated waveform from RX module (ideal)')
xlabel('Time (s)')
ylabel('Magnitude (V)')
legend('[I component], [Q component], [Magnitude]')
axis tight; axisLimits = axis axis([axisLimits(1:2) 1.2*axisLimits(3:4)])

% Simulate the presence of high noise in the received signal
snr = 0.01; % signal to noise ratio of 0.01dB
receivedWaveform = awgn(receivedWaveform, snr);

axisHandle2 = subplot(2,1); hold on
plot([1:length(receivedWaveform)]/sampleRate, real(receivedWaveform), 'b', 'linewidth',1)
plot([1:length(receivedWaveform)]/sampleRate, imag(receivedWaveform), 'g', 'linewidth',1)
title(sprintf('Simulated waveform from RX module (ideal + noise w/ SNR = %.2fdB)',snr))
xlabel('Time (s)')
ylabel('Magnitude (V)')
```
Analyzing Radar Signals to Calculate Time Side Lobe Measurements

- Efficient time side lobe measurements can be made by operating in the frequency domain

- In addition to creating the signals, MATLAB provides a rich environment to transform, analyze and visualize signals

- To calculate time side lobe measurements, we use MATLAB to transform the signals into frequency domain and run it through a matched filter

```matlab
%% Calculate time sidelobe measurements
receivedSignalLength = length(receivedWaveform);
idealPulse = conj(fliplr((idealPulse)));  % Flip and conjugate the idealPulse to implement matched filtering
nfft = 2^nextpow2(receivedSignalLength);  % FFT is most efficient with radix-2 number of points
w = blackman(length(idealPulse)).';  % Creating a Blackman Window of length equal to length of Ideal Pulse
fftIdeal = (fft(w.*idealPulse,nfft))/length(idealPulse);  % compute FFT of ideal pulse
fftMeasured = (fft(receivedWaveform,nfft))/length(receivedWaveform);  % compute FFT of receivedWaveform
correlationCalc= fftMeasured.*(fftIdeal);  % Multiply FFTs
compressionFilter = length(receivedWaveform)*length(idealPulse)*(ifft(correlationCalc,nfft));  % Calculating IFFT of
zz1=20*log10(abs(compressionFilter)/max(abs(compressionFilter))));  % convert to dB
```
Analyzing Radar Signals to Calculate Time Side Lobe Measurements

- On transforming the frequency domain signal back to the time domain, we observe peaks where the received signal has a high correlation with the transmitted signal.

- Using the variety of different window functions in MATLAB it is easy to replace the Blackman window used here with other windowing functions to compare the effect on the time side lobe measurements.
Implementing Analysis Algorithms

- Once the algorithm has been designed and verified in simulation, Agilent oscilloscopes can implement the MATLAB algorithms in the signal path.

- Agilent oscilloscope option N8806A – User Defined Function enables you to create and apply custom analysis routines developed in MATLAB on Agilent Infiniium 90000-X, 90000, and 9000 Series oscilloscopes.
Additional Example: Implementing Oscilloscope User Defined Functions with MATLAB
Start with Oscilloscope Waveform

Oscilloscope Waveform → Custom MATLAB Function → MATLAB Applied Trace → Perform Additional Scope Measurements
Operate on Scope Waveform with Custom MATLAB Function to Extract Pulsed RF Envelope

Custom MATLAB Function to Calculate RF Pulse Envelope with a Hilbert Transform

```
% This takes the Hilbert Transform to extract the envelope and phase

I_Data = real(hilbert(SrcData));
Q_Data = imag(hilbert(SrcData));
EnvData = sqrt(I_Data .* I_Data + Q_Data .* Q_Data);
PhaseData=unwrap((atan(Q_Data ./ I_Data)* pi/180));
```
Display the RF Pulse Envelope

Oscilloscope Waveform → Custom MATLAB Function → MATLAB Applied Trace → Perform Additional Scope Measurements

RF Pulse Envelope
Extracted from Custom MATLAB Function
Perform Scope Measurements on the RF Envelope

Pre-Configured Scope Measurements:
• Pulse Rise Time
• Pulse Fall Time
• Pulse Width
• Overshoot
MATLAB N6171A Software from Agilent for Signal Generators, Signal Analyzers, and Other Hardware

- Available with all MXG, ESG, and PSG signal generators, X-series signal analyzers, PXI / AXIe modular, and other Agilent instruments

- Order with an Agilent signal generator to generate your own waveforms (custom modulation schemes, proprietary, linear chirp, multi-tone, pulsed, multi-carrier signals, etc.)

- Order with an Agilent signal analyzer to apply your custom measurement and analysis routines to live or recorded data and build GUI-based apps. to synchronize and automate measurements

- Downloadable MATLAB signal generation and analysis scripts, videos, instrument drivers, and ordering information available at: www.agilent.com/find/matlab
Benefits of Ordering MATLAB Together with Agilent Instruments

**Complete Solution:** Acquire your hardware and software needed to program your instrument on a single purchase order

**Confidence:** MATLAB software sold through Agilent has been tested and qualified by Agilent

**Support:** Contact either Agilent or MathWorks for help with installation and technical questions

**Quick start:** Acquire numerous application examples and instrument drivers directly from Agilent to get started

**Reliability:** Ensure that your MATLAB software license is always available to you when you need it
Conclusion

• Pulse compression provides a performance advantage for modern Radar systems

• Time Side Lobe level is a key metric in characterizing pulse compression Radar performance

• Hardware and Software products are available from Agilent to easily perform these and other measurements

<table>
<thead>
<tr>
<th><strong>Signal Creation</strong></th>
<th><strong>Analysis Tools</strong></th>
<th><strong>Instruments</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Studio</td>
<td>89600 VSA Software</td>
<td>M8190A AWG</td>
</tr>
<tr>
<td>SystemVue</td>
<td>SystemVue</td>
<td>90000X Scope</td>
</tr>
<tr>
<td>MATLAB</td>
<td>MATLAB</td>
<td>E8267D VSG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N9030A PXA SA</td>
</tr>
</tbody>
</table>

Many different solutions in addition to these are available. Ask your Agilent sales representative or check out www.agilent.com/find/radar
References

Agilent Radar, EW & ELINT Testing: Identifying Common Test Challenges

Agilent Radar Measurements

Improving Radar Performance by Optimizing Overall Signal-to-Noise Ratio

Using Time Sidelobe Measurements to Assess the Performance of Compressed-Pulse Radars

Agilent MATLAB Data Analysis Software Packages for Agilent Oscilloscopes