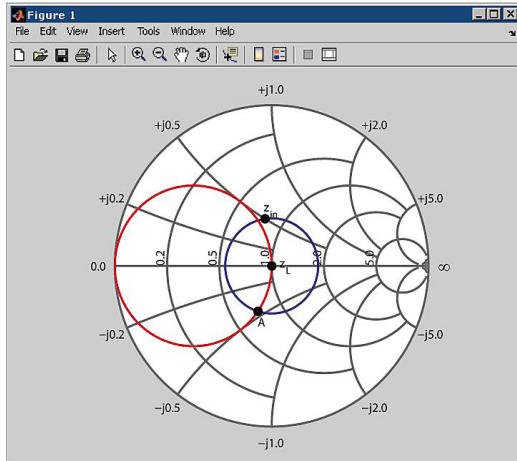# Keysight N6171A

MATLAB Data Analysis Software for Keysight Technologies Instruments

Technical Overview

– MATLAB is available directly from Keysight or its authorized sales partners with your instrument purchase

– Use MATLAB to create and generate arbitrary waveforms including multi-tone signals, pulsed radar signals, and multi-carrier modulated waveforms

– Capture, analyze, and graphically visualize IQ Data, S-Parameters, and application-specific measurements

– Develop and execute custom analysis applications directly on your instrument or on a remote PC

– Create simple or advanced GUIs to automate instrument configuration and data collection

KEYSIGHT
TECHNOLOGIES

Create and Execute Your Own Analysis Routines and Applications

Have you ever wanted to create your own custom application—or modify an existing one—for your specific testing needs? Have you ever wanted your instrument to provide tools to test unique or proprietary communications signals? Today's test conditions necessitate the ability to generate new and emerging input signals or to be able to analyze increasingly complex output data. Keysight now offers the perfect solution to meet your specific testing needs—You can now add the MATLAB software environment (Keysight Option N6171A) when you purchase an Keysight instrument. Users can obtain this high-quality instrumentation and reputable data analysis software environment from a single source—a benefit provided only by Keysight.

Generate new and emerging input signals or analyze increasingly complex output data. You can now add the MATLAB software environment when you purchase an Keysight instrument.

MATLAB Overview

Any test environment requires either stimulating a device under test, analyzing the output of a test, or performing both of these functions. MATLAB can aid in any of these aspects of testing.

MATLAB, a well-known and respected data analysis software environment and programming language created by MathWorks, is available for purchase directly from Keysight. MATLAB software can be used to mathematically generate arbitrary waveforms, make measurements, analyze and visualize data, control instruments, and build test systems. It provides interactive tools and command-line functions for a wide range of applications, including signal processing, signal modulation, digital filtering, and communications systems. MATLAB has over 1,000,000 users in diverse industries and disciplines, and it is a standard at more than 3,500 colleges and universities worldwide.

The benefits of purchasing MATLAB from Keysight

Convenience: Order MATLAB Option N6171A software together with your Keysight instrument so that it can be on a single purchase order to save you time, reduce ordering complexity, and typically add only minimal additional cost to the overall purchase

Quick start: Acquire numerous fully customizable and editable example MATLAB applications and instrument drivers when you purchase MATLAB from Keysight

Dedicated instrument license: A dedicated node-locked license for your instrument gives you confidence the license will be available when needed

Dual technical support: You may contact either Keysight or MathWorks for technical support or installation assistance

Confidence: MATLAB software sold through Keysight will be pre-downloaded on most Keysight Windows-based instruments and tested and qualified by Keysight
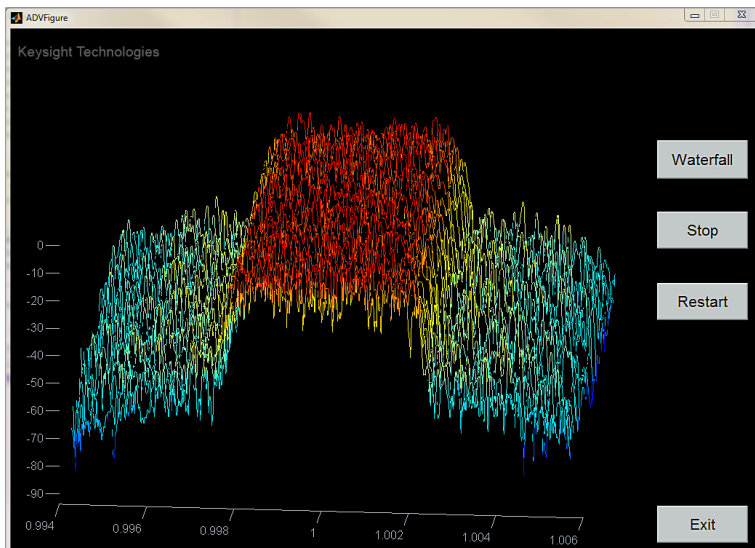

Figure 1. Visualize live wireless signals on a 3- D "waterfall plot" using an example MATLAB application available from Keysight with purchase of the N6171A MATLAB option. Execute this application directly on the signal analyzer or remotely over GPIB, LAN, or USB connectivity. Modify this application as needed to meet your specific testing needs.

With MATLAB you can:

– Test the functionality of electronic devices by making measurements with Keysight instruments and comparing them against known baselines in MATLAB

– Excite electronic devices using Keysight instruments with simple or complex waveforms created in MATLAB

– Characterize an electronic device to determine how closely it matches the design

– Verify new algorithms or measurement routines using live data from Keysight instruments

– Design custom digital filters in MATLAB and apply them to signals acquired from an Keysight instrument

– Develop a GUI or application that automates repeated data analysis or testing

Toolboxes provided in the MATLAB packages

In addition to the MATLAB software environment, the N6171A packages offer options for various MATLAB toolboxes.

Instrument control toolbox

– Enables data to be generated from MATLAB and sent to an instrument (signal analyzer, signal generator, network analyzer, PXI or AXIe modular hardware) or read into MATLAB for analysis and visualization.

– Connects to the instrument via instrument drivers, such as IVI and VXIplug&play, or via text-based SCPI commands, over commonly used communication protocols, such as GPIB, LAN (TCP/IP), USB, and VISA.
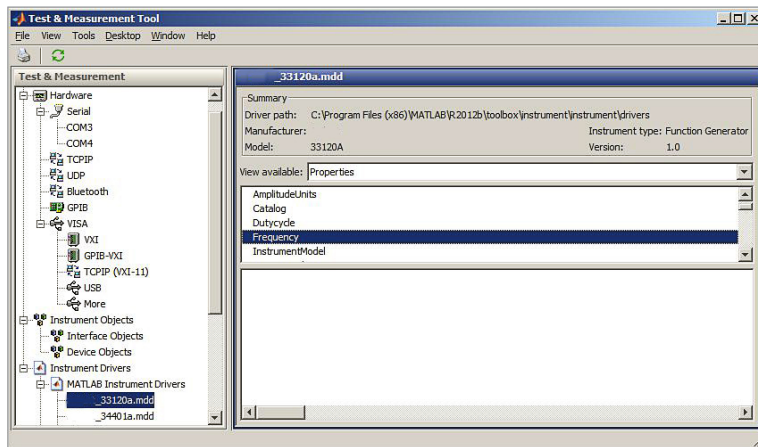
Figure 2. Configure and control Keysight instruments using GUI-based tools and command line functions provided with the Instrument Control Toolbox (available in all MATLAB N6171A packages).

Communications system toolbox

– Extends MATLAB's function, plot, and GUI capabilities for exploring, designing, analyzing, and simulating algorithms of communications systems, enabling custom and standard modulation and demodulation on instrument signals.

– Allows creation of algorithms for commercial or defense systems, such as mobile handsets and base stations, and wired and wireless local area networks.

– Ideal for research and education environments for communication systems engineering.
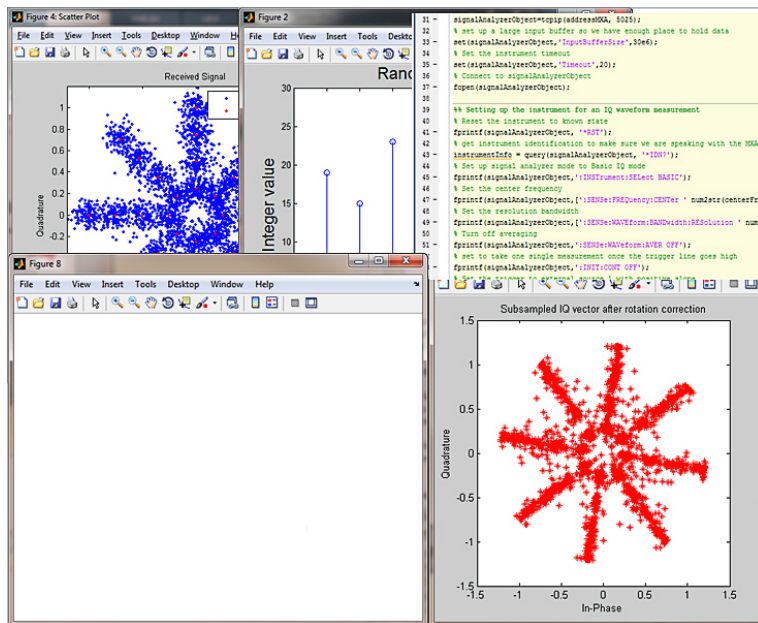

Figure 3. Analyze and test standard, proprietary, and emerging modulation schemes using GUI-based tools and command line functions provided with the Communications System Toolbox (included in MATLAB N6171A –M02 or N6171A –M03).

Signal processing toolbox

– Use MATLAB to create and apply digital filters on live or recorded instrument signals including FIR and IIR filters.

– Includes industry-standard algorithms for analog and digital signal processing (DSP).

– Provides GUIs that allow graphical creation of filters while MATLAB code is automatically being generated for later use in MATLAB scripts or applications.

DSP system toolbox

– Provides advanced filter design capabilities beyond what is offered in Signal Processing Toolbox, including extension of filter architectures and design methods for creating and applying adaptive and multi-rate filtering to live or recorded instrument signals.
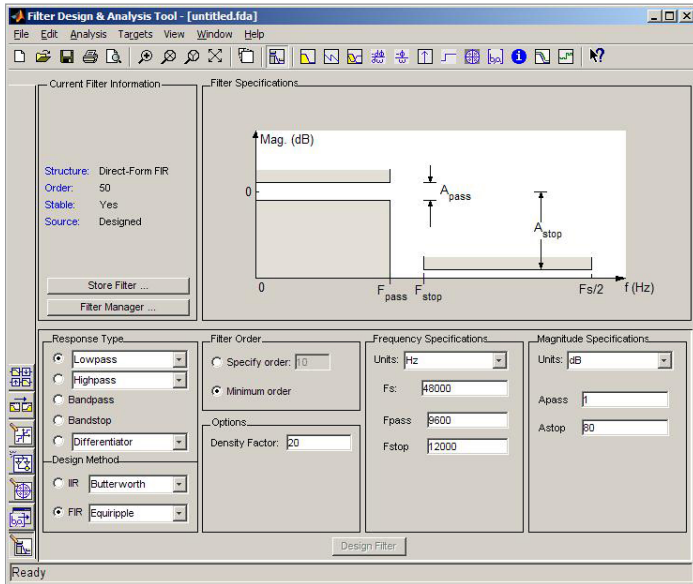

Figure 4. Create and apply digital filters on your instrument signals using GUI-based tools and command line functions provided in the Signal Processing Toolbox and the DSP System Toolbox (included in MATLAB N6171A –M02 or N6171A –M03).

RF toolbox

– Extends MATLAB functional and a GUI capabilities for designing, modeling, analyzing, and visualizing networks of RF components.

– Use RF Toolbox for working on wireless communications, radar, and signal integrity projects.

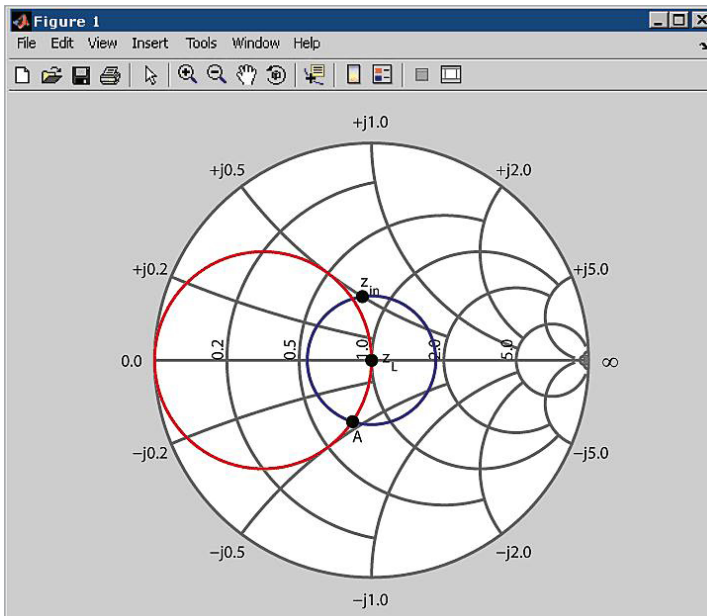– Includes rectangular, polar, and Smith chart plots for visualizing data.

Figure 5. Analyze and visualize RF data using rectangular, polar and Smith charts with GUI-based tools and command line functions provided with the RF Toolbox (included in MATLAB N6171A –M03).

www.keysight.com

Getting Started

Keysight has developed a suite of MATLAB example applications to assist customers in their development and testing needs. The example programs start with basic data acquisition and plotting. Moving beyond the basics, there are also advanced programs that allow users to create their own complete applications.

Example applications

Signal analyzer

MATLAB applications can be executed directly on Keysight signal analyzer, or on a remote PC using GPIB, LAN, or USB connectivity. Users can modify the Keysight applications developed with MATLAB to meet their specific testing needs. While running inside the instrument, the analog demodulation application can be driven with the front panel of the instrument or a remote PC operated via mouse. These are just a few of the many ways to extend the functionality of your Keysight signal analyzer using MATLAB:

– Analyze RADAR signals to calculate time side-lobe measurements
– Generate and analyze arbitrary waveforms with custom modulation schemes
– Build custom GUIs for X-Series signal analyzers
– Capture and analyze IQ Data to detect and understand signal abnormalities
– Design custom digital filters and apply them to signals acquired from an analyzer
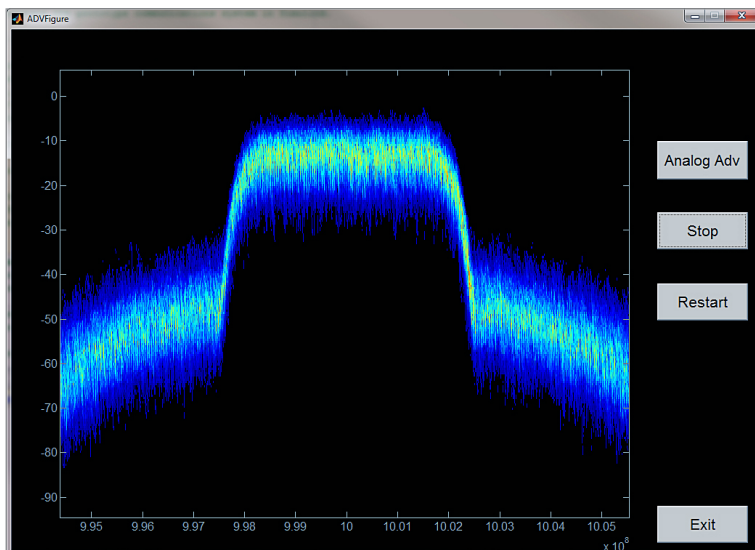

Figure 6. Example application: Persistence display developed using MATLAB software.

Order the N6171A MATLAB Technical Kit

This technical kit contains over 20 example applications and programs, tested MATLAB instrument drivers, an application note which describes how to use MATLAB with Keysight instruments and how to use MATLAB to create both beginner and advanced testing solutions, and this technical overview. Customers that purchase MATLAB from Keysight will receive access to source code for these example applications to be able to modify and extend them for their specific applications.

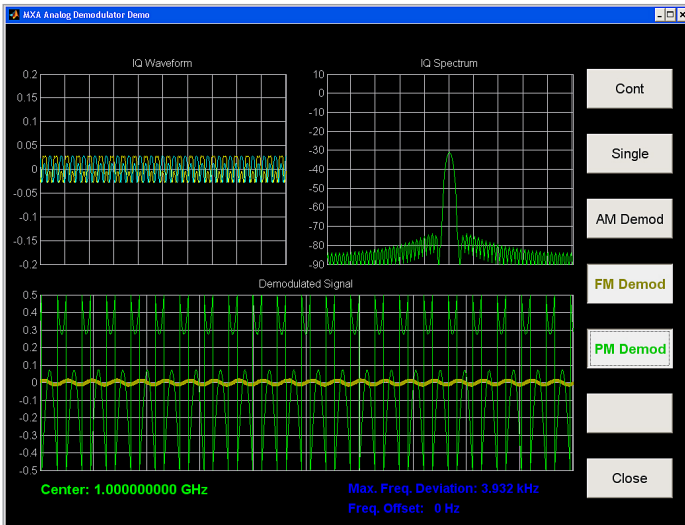To order the technical kit, visit:
www.keysight.com/find/N6171a.

Figure 7. Example application: Analog demodulation of an FM signal using a MATLAB application.

Signal generators

Keysight's wide variety of signal generators meet various testing needs and, together with MATLAB, they can be used for:

– Creating and generating IQ data

– Creating RADAR signals from mathematical definitions

– Creating channel and multipath effects in a waveform obtained from a simulated receiver module

– Create multi-tone signals, pulsed radar, and multi-carrier waveforms

– Perform digital pre-distortion to reduce effects of spectral re-growth

Keysight's tested and verified MATLAB driver can be edited for individual use cases. IQ data can be acquired with one command using the MATLAB instrument driver and the data can then be used for demodulation and analysis of complex signals.
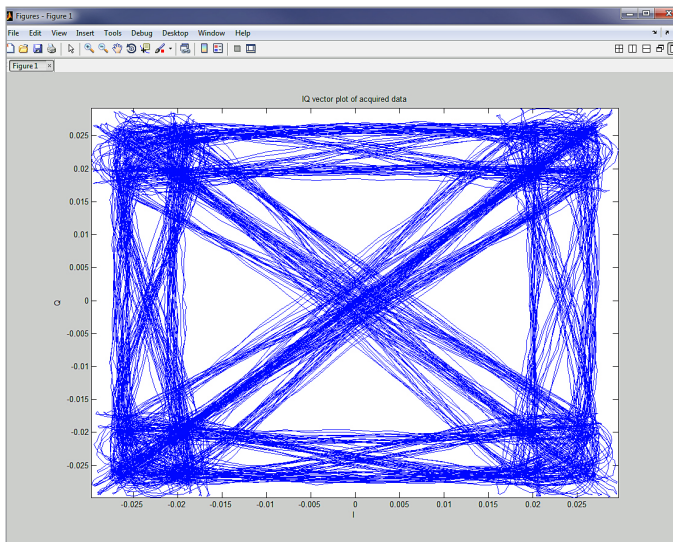

Figure 8. A MATLAB plot of a QPSK signal that used the MATLAB instrument driver for IQ data acquisition.

Stimulus and Analysis

Third order intercept (TOI) is an important parameter for characterizing RF components. To develop a 3-D visualization of TOI, MATLAB can be used to control the signal genera-tor and to acquire the results with a signal analyzer to dynamically view TOI vs. tone separation.
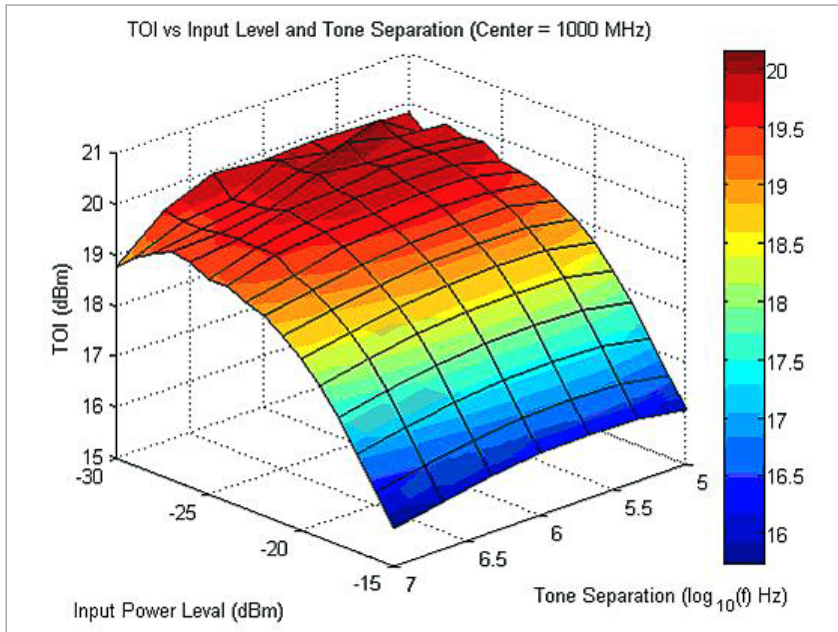

Figure 9. Example application: A MATLAB generated surface plot for analyzing third order intercept vs. input power and tone separation.

Arbitrary waveform generator

MATLAB excels at performing complex or repetitive math functions. It is especially useful for creating a signal from a mathematical definition. This capability, in conjunction with an arbitrary waveform generator (AWG) enables:

– Generation of custom UWB and SATCOM signals using MATLAB

– Development of a GUI application to create signals on traditional and AXIe form factor AWGs

– Generate arbitrary waveforms (multi-tone signals, pulsed radar signals, and multi-carrier modulated waveforms)

– Perform waveform pre-distortion and amplitude correction to generate a waveform that results in a flat amplitude response

With MATLAB you can create a multi-tone signal on an arbitrary waveform generator. To perform more extensive analysis, a simple script can be used to correct for amplitude flatness.
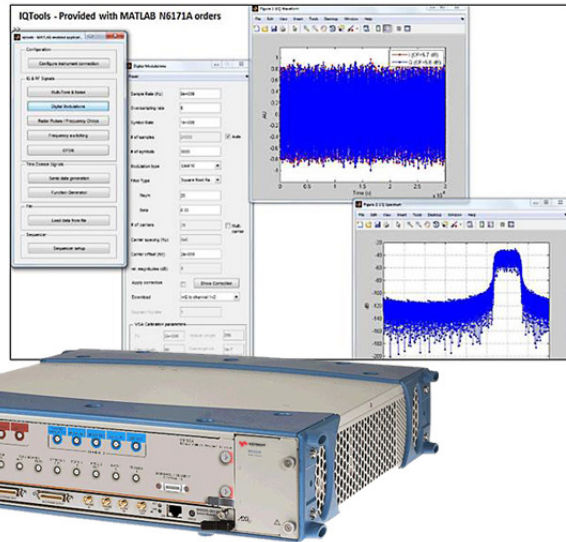
Figure 10. Example application: The IQ Tools example application is included with the purchase of any N6171A MATLAB software package (requires N6171A –M03 to operate). Use it to generate arbitrary waveforms (multi-tone signals, pulsed radar signals, and multi-carrier modulated waveforms) with the MATLAB software for the Keysight M8190A arbitrary waveform generator.
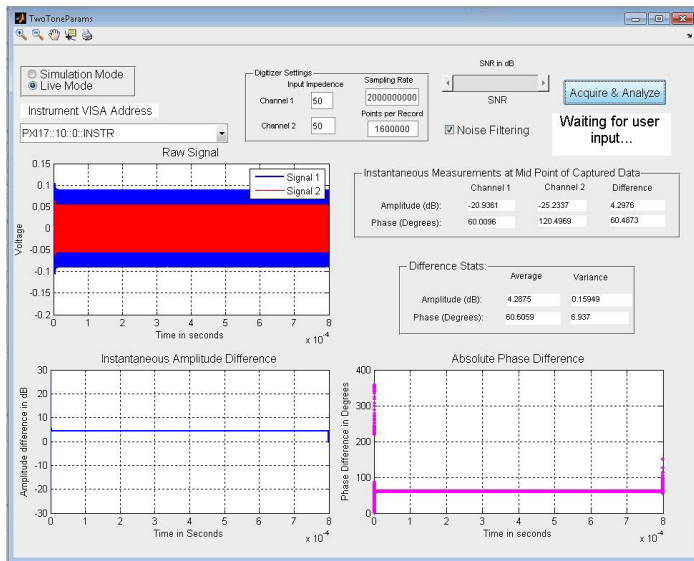
Figure 11. Example application: MATLAB example application provided with the purchase of any N6171A MATLAB software package to download phase shifted tones and measure instantaneous and statistical parameters of two tones.

PXI/AXIe modular hardware

MATLAB is an ideal choice as the software environment for controlling modular hardware. To use MATLAB with modular instruments, the software must be installed on an external or embedded PC. Doing so aids in accomplishing a wide range of testing tasks, including:

- Generate arbitrary waveforms (multi-tone signals, pulsed radar signals, and multi-carrier modulated waveforms)
- Make measurement and data analysis routines, user-defined filters, and GUI-based instrument applications
- Decoding I2C signals and testing I2C devices with your digitizer
- Generating a report characterizing a solar cell by automating measurements from multiple instruments
- Making a GUI application to demonstrate SFDR measurements

Purchase from Keysight with your PXI or AXIe controller or other modular hardware as your software environment for controlling the hardware, generating waveforms, or analyzing signals.
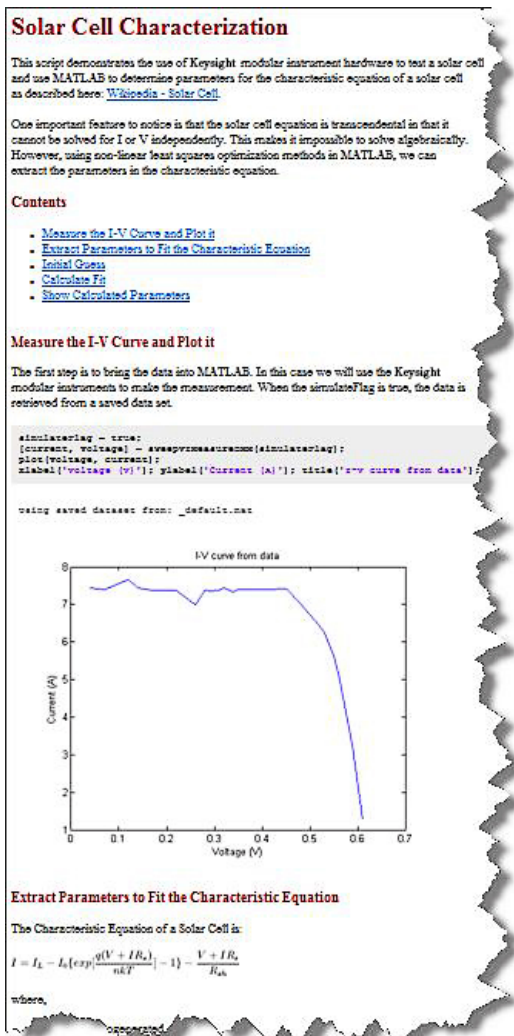

Figure 12. Example application: Use MATLAB to automatically generate reports for the Keysight M9183A DMM, M9186A VI source, and M9121A matrix switch. Calculate key characteristics of a solar cell based on measured data and automatically generate a report in MATLAB to share insights with colleagues, customers, or management.
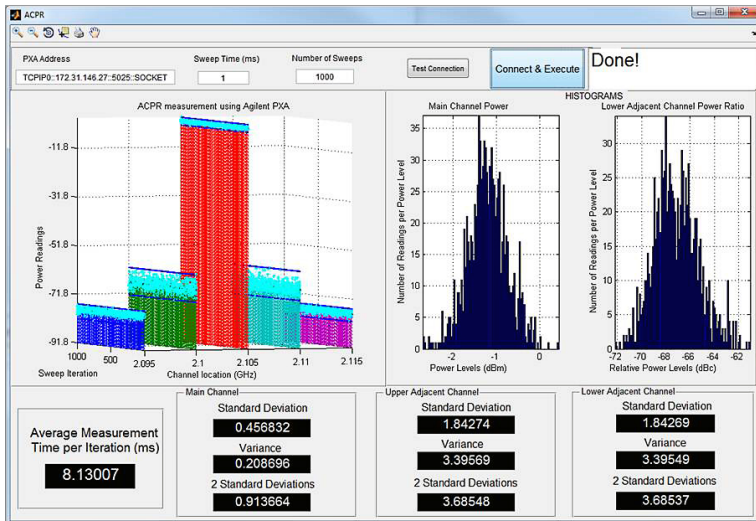
Figure 13. Example application: Automate the measurement of ACPR measurements using an example application developed in MATLAB.

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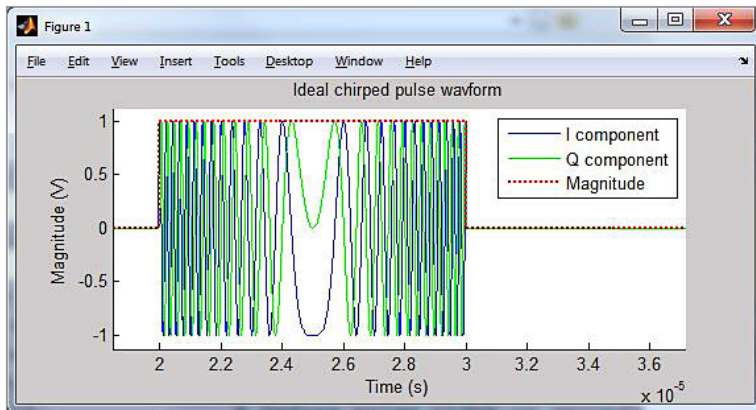

Figure 14. Example application: Create RADAR signals from mathematical definition. Mathematical definition of the ideal signal generated from the transmitter module is required to calculate time-sidelobe measurements. MATLAB provides functions to easily define the ideal signal.

```
1   %% Simulate a pulse recieved at the RX-module corresponding to the pulse transmitted from the TX module
2   % Create a pulse that contains overlapping receptions
3   shortOffset = zeros(1, 1e-6*sampleRate); % time offset of 1e-6 seconds
4   longOffset = zeros(1,10e-6*sampleRate);  % time offset of 10e-6 seconds
5   offsetAttenuatedPulse1 = [10^(-3/20)*idealPulse shortOffset]; % reflection w/ -3dB gain
6   offsetAttenuatedPulse2 = [shortOffset 10^(-25/20)*idealPulse];   % reflection w/ -20dB gain
7   overlappingPulse = offsetAttenuatedPulse1 + offsetAttenuatedPulse2;
8
9   % Create a recieved pulse that contains direct reflections, and overlapping reflections
10  recievedWaveform = [longOffset ...
11                      idealPulse ... % reflection w/ 0dB gain
12                      longOffset ...
13                      10^(-10/20)*idealPulse ... % reflection w/ -10dB gain
14                      longOffset ...
15                      overlappingPulse ...
16                      longOffset];
17
18  % Plot the simulated recived waveform
19  figure(3); axisHandle1 = subplot(211); hold on
20  plot((1:length(recievedWaveform))/sampleRate, real(recievedWaveform),'b','linewidth',2)
21  plot((1:length(recievedWaveform))/sampleRate, imag(recievedWaveform),'g','linewidth',2)
22  plot((1:length(recievedWaveform))/sampleRate, abs(recievedWaveform),':r','linewidth',1)
23  title('Simulated waveform from RX module (ideal)')
24  xlabel('Time (s)')
25  ylabel('Magnitude (V)')
26  legend('I component','Q component','Magnitude')
27  axis tight; axisLimits = axis; axis([axisLimits(1:2) 1.2*(axisLimits(3:4))])
28
29  % Simulate the presense of high noise in the recieved signal
30  snr = 0.01; % signal to noise ratio of 0.01dB
31  recievedWaveform = awgn(recievedWaveform, snr);
32
33  axisHandle2 = subplot(212); hold on
34  plot((1:length(recievedWaveform))/sampleRate, real(recievedWaveform),'b','linewidth',1)
35  plot((1:length(recievedWaveform))/sampleRate, imag(recievedWaveform),'g','linewidth',1)
36  title(sprintf('Simulated waveform from RX module (ideal + noise w/ SNR = %2.2fdB)',snr))
37  xlabel('Time (s)')
38  ylabel('Magnitude (V)')
```
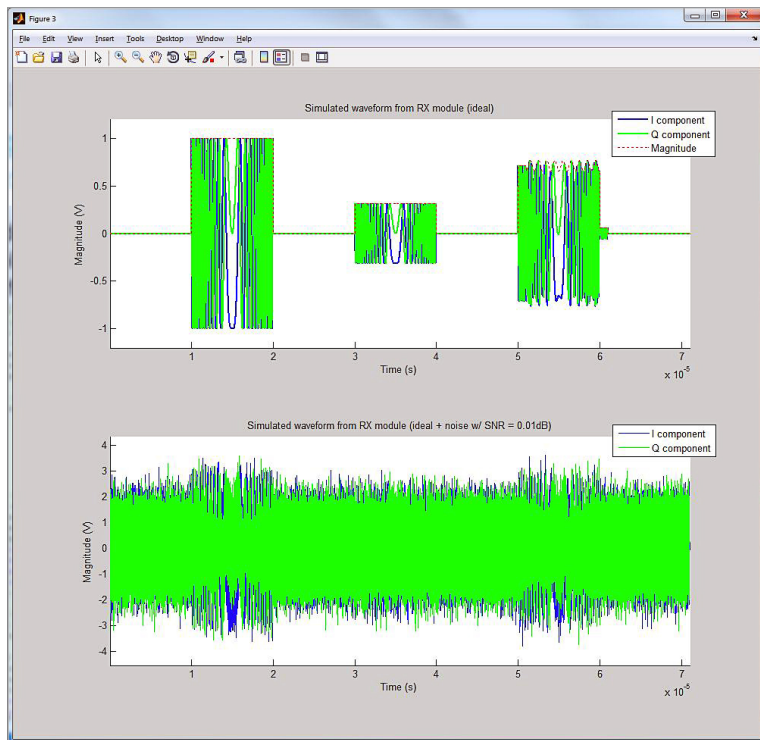

Figure 15. Example application: Create channel and multipath effects in a waveform obtained from a simulated receiver module. When designing RADAR receiver processing algorithms, it is necessary to model channel and multipath effects on the waveforms. You can easily do this using the built-in functions in the N6171A MATLAB software.

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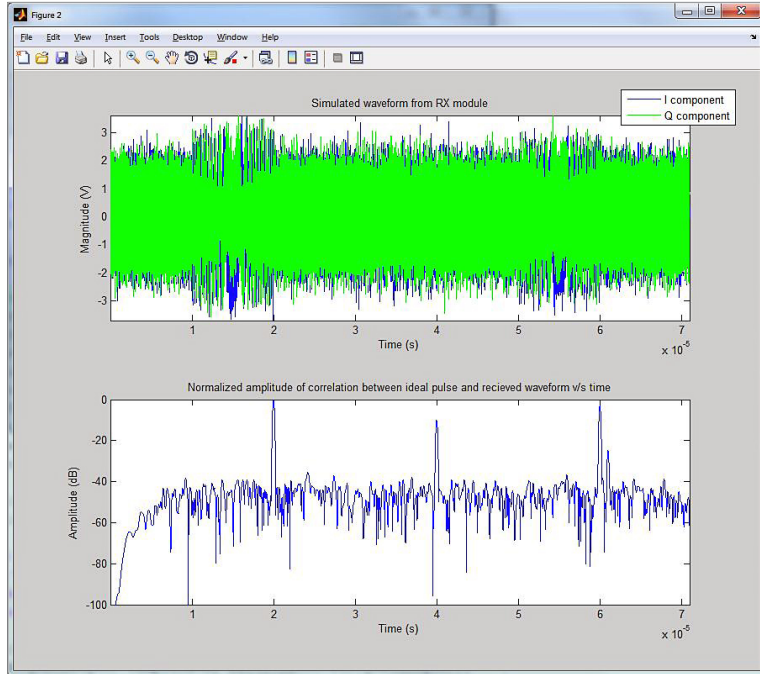

Figure 16. Example application: Use MATLAB programs to analyze RADAR signals to make time sidelobe measurements with synthesized or captured data. When the frequency domain signal is transformed back to the time domain, we observe peaks where the received signal has a high correlation with the transmitted signal. Using the variety of 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 sidelobe measurements.

Network analyzers

Network analyzers are an increasingly important measurement tool for component characterization. Data obtained from a network analyzer can be manipulated in various ways using the tools and functions provided in MATLAB. A few examples of what you can accomplish with these functions are:

– Acquire live S-parameter measurements for analysis and visualization
   Generate Touchstone format files using acquired S-parameter measurements

– Verifying signals for designing broadband matching antenna and amplifier networks, or calculating the TDR and TDT of a differential high-speed backplane channel
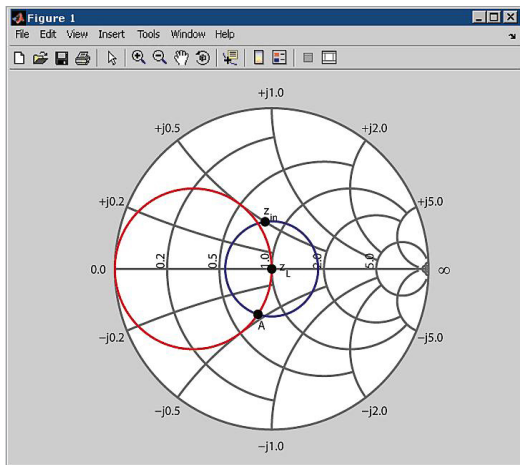

Figure 17. Analyze and visualize RF data using rectangular, polar and Smith charts with GUI-based tools and command line functions provided with the RF Toolbox (included in MATLAB N6171A –M03).

Command Expert

MATLAB owners can add Command Expert to complement instrument control. Command Expert provides integrated documentation, syntax checking, and additional interfaces for building command sequences.

Ordering Information

Modern connectivity

Choose the best connection to meet specific requirements:

– USB

– LAN – 100 based-T

– GPIB

– LXI– Class C compliant

Software licensing and configuration

Fixed, perpetual license:

This allows you to install and run MATLAB software on your Windows-based instrument or from a remote PC.

N6171A MATLAB software

Keysight offers three MATLAB N6171A software packages representing typical configurations for use with Keysight instruments. These packages range from basic MATLAB capabilities to acquire and analyze data to full support for signal processing, communications systems, filter design, and automated testing.

Description	Model-Option	Additional information	
		Products included	Notes
MATLAB – Basic Package	N6171A – M01	1. MATLAB 2. Instrument Control Toolbox	Provides an introductory MATLAB software package for controlling Keysight instruments.
MATLAB – Standard Package (recommended for most instrument purchases)	N6171A – M02	1. MATLAB – Basic Package 2. Communications System Toolbox 3. DSP System Toolbox 4. Signal Processing Toolbox	Adds toolboxes to provide a typical MATLAB software package for signal filtering, testing modulation schemes, and developing instrument apps.
MATLAB – Advanced Package (recommended when working with RF networks)	N6171A – M03	1. MATLAB – Standard Package 2. RF Toolbox	Adds RF Toolbox for designing, modeling and analyzing circuits of RF components.

Try Before You Buy!

You may request a free trial of N6171A MATLAB software at www.mathworks.com/keysight/trial

Hardware configuration

Description		Model	Additional information: Required software revision
Spectrum and signal analyzers	PSA Series spectrum analyzers	E4440A	Rev 2.07 or later
		E4443A	Rev 2.07 or later
		E4445A	Rev 2.07 or later
		E4446A	Rev 2.07 or later
		E4447A	Rev 2.07 or later
		E4448A	Rev 2.07 or later
	PXA, MXA, EXA, CXA X-Series signal analyzers	N9030A	Rev A.03.08 or later
		N9020A	Rev 1.05 or later
		N9010A	Rev 1.24 or later
		N9000A	Rev 4.01 or later
Signal generators	ESG vector signal generator	E4438C	Rev C.02.02 or later
	PSG vector signal generator	E8267D	Rev C.04.01 or later
	MXG RF vector signal generator	N5182A	Rev B.01.01 or later
	EXG and MXG X-Series RF vector signal generators	N5172B	Rev B.01.01 or later
		N5182B	Rev B.01.01 or later
Arbitrary waveform generators	12 GSa/s arbitrary waveform generator	M8190A	Rev 2.0.5.0 or later
	1.25 GSa/s arbitrary waveform generator	M9330A	Driver rev 1.2.1.0 or later
		M9331A	Driver rev 1.2.1.0 or later
Network analyzers	PNA microwave network analyzer	N5221A	Rev A.01.02 or later
		N5222A	Rev A.01.02 or later
		N5224A	Rev A.01.02 or later
		N5225A	Rev A.01.02 or later
		N5227A	Rev A.01.02 or later
	PNA-X microwave network analyzer, 10 MHz to 26.5 GHz	N5241A	Rev A.01.02 or later
		N5242A	Rev A.01.02 or later
		N5244A	Rev A.01.02 or later
		N5245A	Rev A.01.02 or later
		N5247A	Rev A.01.02 or later
PXI and AXIe modular systems	PXI embedded controller	M9036A	Windows 7
	AXIe embedded controller	M9536A	Windows 7
	PXI vector signal analyzer	M9392A	Driver rev 1.2.2404

Note: MATLAB can also be ordered with Keysight oscilloscopes using the –061, –062, and N8806A oscilloscope options. Visit www.keysight.com/find/matlab_oscilloscopes and www.keysight.com/find/udf for details.

Related Literature

Using MATLAB to Create Keysight Signal and Spectrum Analyzer Applications 5989-9377EN

Web

Product pages:
www.keysight.com/find/N6171A

To view a collection of various videos with Keysight instrumentation and N6171A MATLAB software, visit:
www.keysight.com/find/matlab_videos

Evolving Since 1939

Our unique combination of hardware, software, services, and people can help you reach your next breakthrough. We are unlocking the future of technology. From Hewlett-Packard to Agilent to Keysight.

myKeysight

myKeysight
www.keysight.com/find/mykeysight
A personalized view into the information most relevant to you.

http://www.keysight.com/find/emt_product_registration
Register your products to get up-to-date product information and find warranty information.

KEYSIGHT SERVICES
Accelerate Technology Adoption.
Lower costs.

Keysight Services
www.keysight.com/find/service
Keysight Services can help from acquisition to renewal across your instrument's lifecycle. Our comprehensive service offerings—one-stop calibration, repair, asset management, technology refresh, consulting, training and more—helps you improve product quality and lower costs.

Keysight Assurance Plans
www.keysight.com/find/AssurancePlans
Up to ten years of protection and no budgetary surprises to ensure your instruments are operating to specification, so you can rely on accurate measurements.

Keysight Channel Partners
www.keysight.com/find/channelpartners
Get the best of both worlds: Keysight's measurement expertise and product breadth, combined with channel partner convenience.

KEYSIGHT TECHNOLOGIES