Matlab Implementation
WLAN Dynamic EVM Measurement

Executive Summary:

Based on the feedback of AEs in Semiconductor test we identified a need for having an IP showing WLAN Dynamic EVM measurements in Matlab. This document is a demo script for this IP and it also contains some of the Gotchas/ care abouts for using NI .NET assemblies in matlab.

1. Getting started with NI .NET Assemblies in matlab

NI RF and Modular instruments .NET Assemblies are based on Object oriented programming. These assemblies contain different classes which contain a variety of methods and object properties. Also, for RFmx and some other complex measurements multiple .NET Assemblies interact with each other. So, the best place to get started on resources on programming of these instruments is by using Labview help documents.

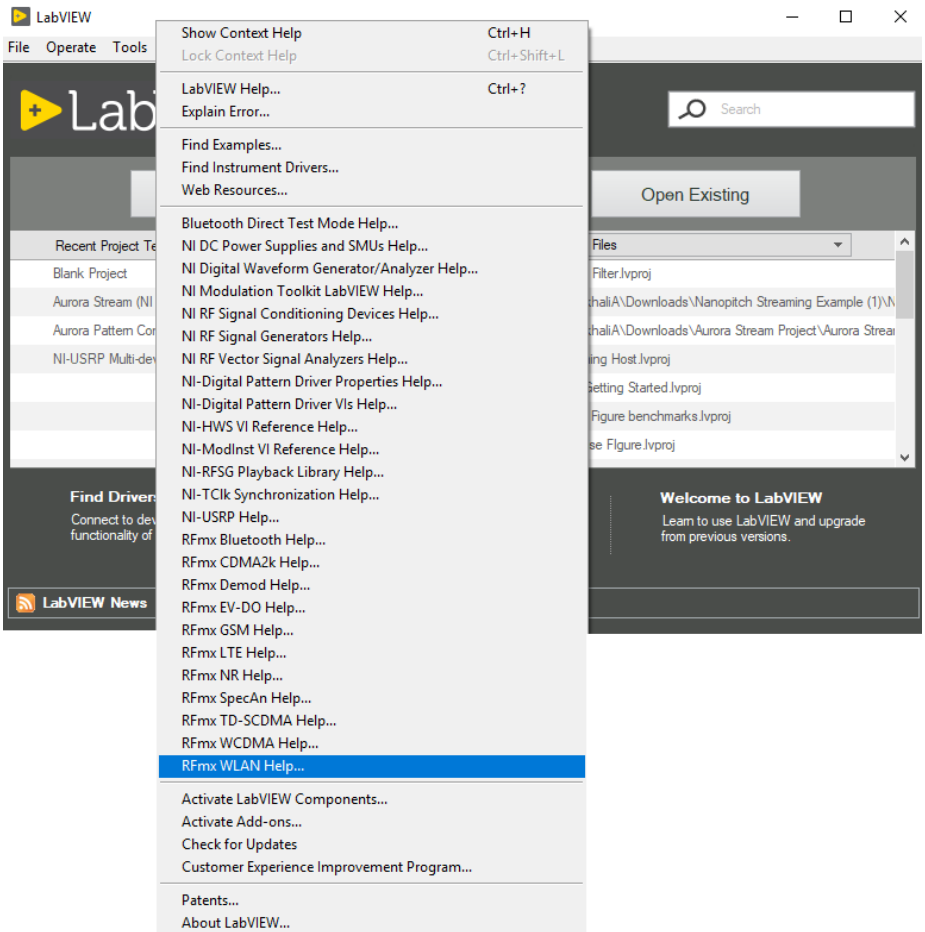

Figure 1

These help documents contain .NET References explaining the namespace of the assembly, associated object, methods, properties and data types.

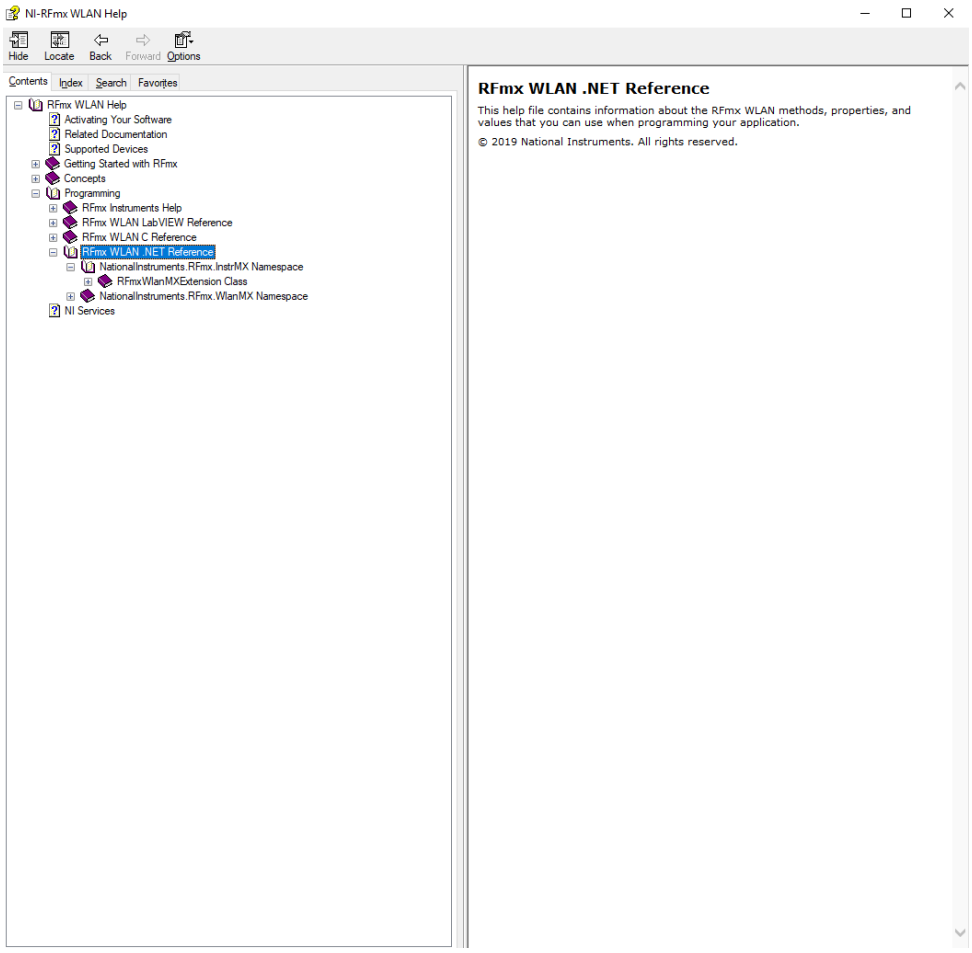

Figure 2

These resources can give insight on method and property call, helping with constructor of object and how different .NET Assemblies interact with each other.

Another resource that is helpful are visual C and C# examples for Modular Instruments and RF. All of these examples can be found in **Examples** folder of corresponding driver's folder in the root folder of *C:\Users\Public\Documents\National Instruments*. For example, the directory containing Examples for RFSG Playback is in the following folder: *C:\Users\Public\Documents\National Instruments\NI-RFSG Playback Library\Examples\DotNET*.

2. .NET Assemblies Care abouts in Matlab

The major Gotchas in matlab programming is the syntax of matlab along with some data conversions. Data conversion is required to make.NET data structure accessible for processing and display in matlab.

1. Adding Assemblies

Like other programming environments we need to add the reference to .Net assemblies and import the namespaces. Syntax and example are given below:

Syntax:

NET.addAssembly('<dll name>');

Import <namespace*>;

Example importing NIRfsg:

```
NET.addAssembly('NationalInstruments.ModularInstruments.NIRfsg.Fx40');
import 'NationalInstruments.ModularInstruments.NIRfsg'.*;
```

2. Accessing Arrays, waveforms and clusters

Data manipulation plays a key role in any measurement. For accessing components of Arrays, waveforms and clusters given below are the methods:

- ### Complex Arrays

Matlab does not allow to plot complex array using the standard plot function. So, to decompose a Complex array of singles following method is used. (same goes for complex double):

```
[real_data,imaginary_data]= ComplexSingle.DecomposeArray(ComlexArray);
```

Here complexArray is the variable/ structure that we want to decompose.

- ### Complex Waveforms

Waveform consists of three components and the methods used to extract all three are given below:

```
t0 = waveform.PrecisionTiming.TimeOffset.TotalSeconds;
dt = waveform.PrecisionTiming.SampleInterval.TotalSeconds;
real_data = waveform.GetRealDataArray(false);
imaginary_data = waveform.GetImaginaryDataArray(false);
```

Here waveform is the complex structure/ waveform data type that we want to decompose.

Also, the method to determine waveform size is as following:

```
Waveform_size = waveform.SampleCount();
```

- ### Real Waveform

```
t0 = waveform.PrecisionTiming.TimeOffset.TotalSeconds;
dt = waveform.PrecisionTiming.SampleInterval.TotalSeconds;
data = waveform.GetRawData();
```

- Spectrum Clusters

Spectrum cluster also consists of three components the methods to extract them is given below:

```
f0 = Data.StartFrequency;
df = Data.FrequencyIncrement;
spectrum_d = Data.GetData();
```

Here Data is the cluster which we want to decompose.

- System Double

There are certain numbers that are returned as System.Double by .NET methods. The method to convert them to doubles is as following:

```
Double_var = Variable.ToDouble();
```

Here Variable is System.Double data type.

3. Syntax specific care abouts

- Calling a constructor method

To create an object don't call 'new'. For example, to create an object for NIRfsg call the constructor as following:

```
rfsgsession = NIRfsg(Resource,false,false);
```

- Method outputs

Most of the .NET methods either return output by "reference" or as "out". Also, in addition to the main output variables/ structures, .NET methods mostly also return an integer which is used for error handling. We can choose to ignore this integer by indexing it as "~" (don't care).

By Ref

Let's now take an example where a method returns the "waveform" output by "reference". Also, this method returns an integer for error handling. In this case we choose to neglect the integer. Also, since output is returned by reference, we need to pass empty object in the method.

```
[~,waveform]=NIRfsgPlayback.ReadWaveformFromFileComplex(path,[]);
```

Here [] is the empty object for waveform.

Out

Now there are certain methods that pass output as "out". For such methods we do not pass anything at there place to the method and obtain the output as return parameter. For example, in this method PAPR is an out parameter for the given method. Notice that we do not pass double in its place in the method:

```
[~,Papr] = NIRfsgPlayback.RetrieveWaveformPapr(rfsgHandle,Rf_waveform_name);
```

- Object array indexing and creating arrays of objects

Matlab does not allow to access/index object arrays within a method. So, in order for us to work with those methods we need to create an intermediate variable to store indexed value of object. This is done by calling “Item” method.

Look at the function below:

```
Smu.Output[channel_name].Source.Output.Function =
DCPowerSourceOutputFunction.DCVoltage;
```

In this example we are setting output function of one of the channels of SMU as DC voltage generation. But this method cannot work in this syntax. This is because we are trying to index the Output channel by its name. So, we need to store this value in a separate variable by first using Item function. This is done below

```
Output = Item(Smu.Outputs,Channel_name);
Output.Source.Output.Function = DCPowerSourceOutputFunction.DCVoltage;
```

Now, there are also instances where we need to create array of objects. This can be easily done using NET.createArray method. Example is as following:

```
syncDevices =
NET.createArray('NationalInstruments.ModularInstruments.ITClockSynchronizable
Device', 2);
```

3. Setup for Dynamic EVM Measurement

Hardware Setup

The hardware setup for this measurement consists of following:

1. PXIe-5840
2. PXIe-4145
3. Mini circuits vat-20+
4. Qorvo RFPA5542B
5. SMA male to female

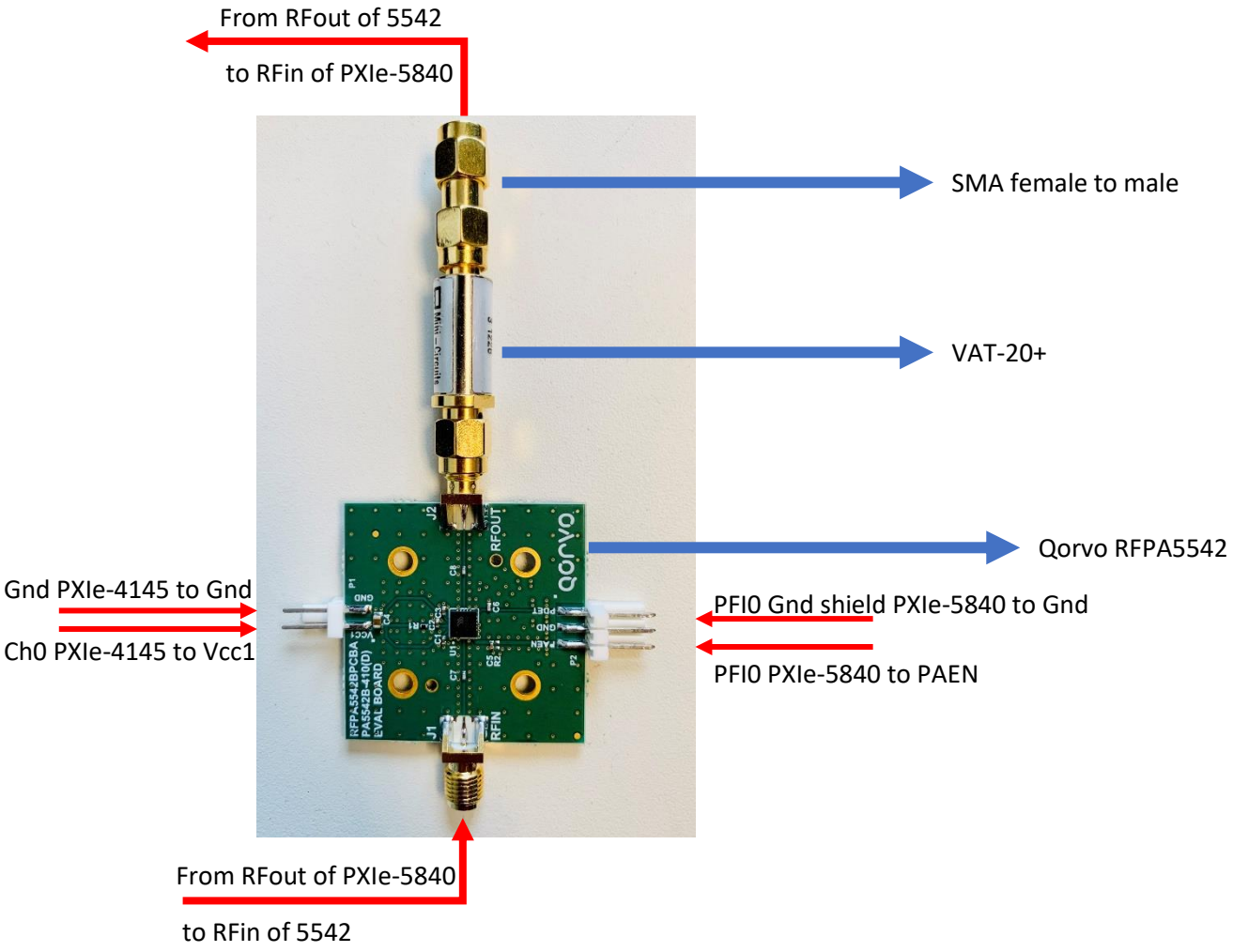

Figure 3

Software Setup

The software setup consists of following piece of software:

1. Matlab R2020a
2. RFmx 19.1.0
3. RFmx SpecAn 19.1
4. RFmx WLAN 19.1
5. RFSG 19.2
6. RFSG Playback Library 19.1
7. RFSA 19.2 (Optional)
8. NationalInstruments.Common.dll
9. LabVIEW 2019 (Optional - as the demo files has a folder performing the same measurements on LabVIEW)

4. Running Demo

The demo consists of two .m Scripts. “RficWlanGenerationDEvmPaEnabledPFI.m” is a script that runs the VSG and uses PFI0 of VSG to enable Power amplifier (UUT). This file also uses a tdms file “80211ax_80M_MCS11.tdms” to generate waveform. This tdms file must be in the same folder as the script file. To run the To Run the script use the "Run" option in matlab EDITOR. This generation script runs indefinitely till we stop the script. To stop generation, you can press on the "Stop Loop" button on "Figure 1" than pops up after running the code. Or you can just close "Figure 1" to Stop execution.

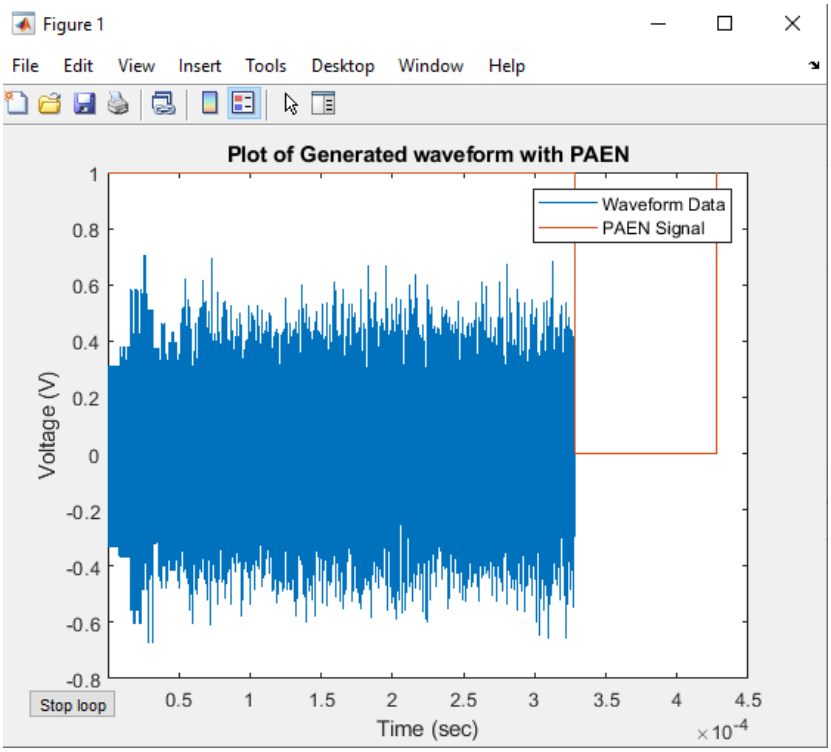

Figure 4

To keep things similar to LabVIEW implementation, we have a separate analysis script. Its name is "RFmxWLANdynamicEVM.m". To run generation and analysis in parallel, you either need parallel execution toolkit from matlab. In our case, since we don't have it, simply open a new instance of matlab and open this file in the new instance. Run this script and it will run once. This script will wait for power trigger to be received and it will wait indefinitely. Once it receives the data burst it performs that measurement and closes all hardware handles. This script also controls the SMU voltage output.

5. LabVIEW comparison

The demo also consists of LabVIEW Vis that does the generation and reception of data and perform measurements. Names of the files are "RFmxWLAN OFDMModAcc_digital trigger" and "RFIC Generation Example PA Enable (PFI)_EVM".

6. Measurement Result

This Power amplifier Qorvo 5542 has dynamic EMV between -36.5 and -35dB at 80MHz according to its datasheet : https://www.qorvo.com/products/d/da001019

11ac 80MHz DEVM		1.5	1.8	%
		-36.5	-35	dB

Figure 5

Observed measurement is as following:

```
RmsEvmMean =

  -35.4555


DataRmsEvmMean =

  -35.4384


PilotRmsEvmMean =

  -36.6464
```

Figure 6

This can be observed in the command window of matlab.