Application Note

R&S®SMW-K546 DIGITAL DOHERTY

Products:

- R&S®SMW200A
- R&S®FSW
- R&S®HMP4040
- R&S®SMW-K546
- R&S®FSW-K18

Gareth LLOYD, Dr Patrick AGYAPONG | GFM345 | Version 1e | 01.2019

https://www.rohde-schwarz.com/appnote/GFM345

ROHDE&SCHWARZ

Make ideas real

Contents

1 Overview

The R&S®SMW-K546 software option aids the optimum design of multiple path amplifiers, including Doherty Amplifiers.

This application note provides guidance on the use of the software option, both for direct control and programming via the touchscreen, and for remote control using SCPI.

The software and associated techniques may similarly be used to develop other quasi-linear, multi-path amplifiers; including Balanced, Anti-Phase (so-called Push-Pull), Distributed, Spatially Combined, Load Modulated Balanced and many more.

2 Reader's Guide

This document describes the use of the R&S®SMW-K546 Digital Doherty option for the SMW Signal Generator.

This product enables the beneficial development of both Doherty Amplifiers, and quasi-linear multiple path amplifiers in general, including amplifiers Load Modulated Balanced Amplifiers, Balanced, Anti-Phase, Spatial Combined, Distributed, etc.

More specifically, this product enables the user to arbitrarily test the amplifier as a dual-input device, thus explore the performance solution space. That is to say, that the maximum performance of the amplifying device can be directly ascertained, and an optimum or appropriate design structure identified.

Testing is divided into differentially linear and differentially non-linear phases. The former is appropriate to all amplifier types. Here, differentially refers to the relationship between the two inputs of the system.

For differentially linear, the applied signal is the same, differing only in its relative amplitude-phase-delay between the paths. For differentially non-linear, then one of the signals is derived from the other, by applying an instantaneous amplitude multiplier or gain.

This document describes both manual and automated use of the software.

For remote control, use will be made of Matlab®. Control of Rohde & Schwarz® instruments using Matlab® is made easy, using only the core package of Matlab® and the freely available tool described in "How to use Rohde & Schwarz Instruments in MATLAB" (see 7).

In the remainder of the document, R&S®SMW200A, R&S®SMW-K546 and R&S®FSW will simply be referred to as SMW200A, SMW-K546 and FSW respectively.

3 Apparatus

The basic test set-up comprises:

► SMW200A (please see 8.2 and 8.3 for the recommended minimum configuration)

► Power Supply (e.g. HMP4040)

► Signal Analyzer (e.g. FSW)

► Dual-Input DUT

► Calibration fixture (e.g. isolated, in-phase, power combiner)

A basic test platform is shown, with DUT in place, in Fig. 3-1.

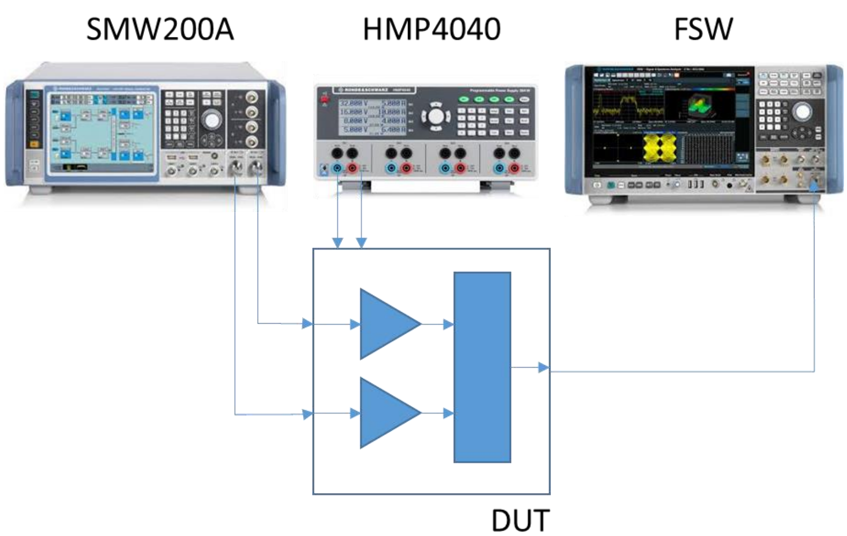


Fig. 3-1: Example dual-input amplifier testing test set-up.

In addition, depending on the specific application, the following may be needed (not shown):

► Power attenuator

► Laboratory grade Power Amplifier(s).

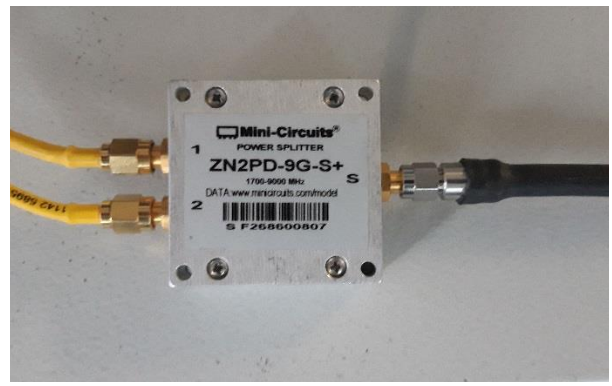
The whole test set-up can easily be automated using, for example, LAN connections between the instruments, a network router and a controller (e.g. PC) (not shown).

4 Calibration

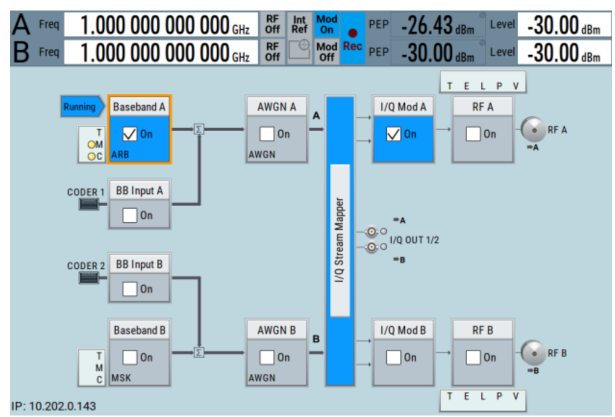
Although the SMW200A and the FSW are factory calibrated to the plane of the RF connector, this plane must be extended to accommodate the influences of test cables and driver amplifiers.

There are many methods by which this may be achieved. This application note describes one such method.

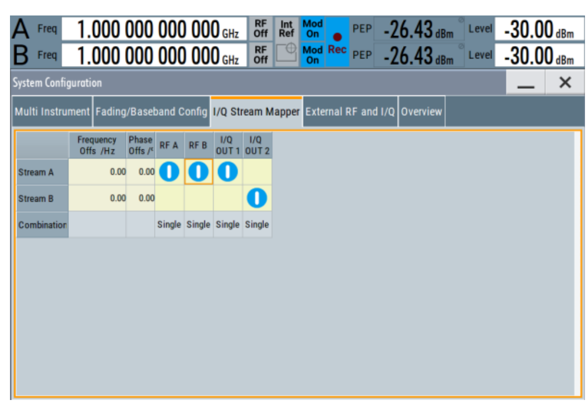
- Preset the instrument(s)
- Using SCPI:
- `*RST`
- Connect a calibration device (e.g. an off-the-shelf power splitter combiner) in lieu of the DUT.
- In this case, for a nominal 3,5GHz test, a Mini-Circuits® ZN2PD-9G-S+ is used.



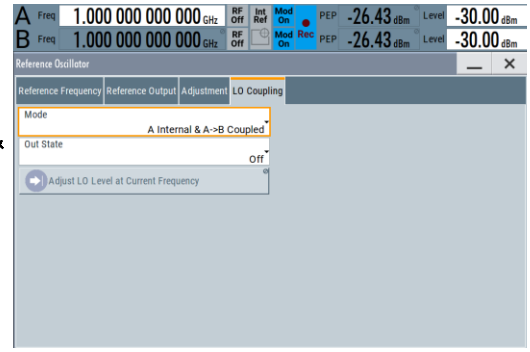
- Load the test signal into the baseband of the SMW200A. In this case, a pre-generated QPSK waveform is used.
- e.g. “Baseband A”->”ARB”->”Load Waveform…”
- The display of the SMW should appear as opposite.
- Using SCPI:
- `:SOURce1:BB:ARBitrary:WAVeform:SELect “/var/user/<filename>”`



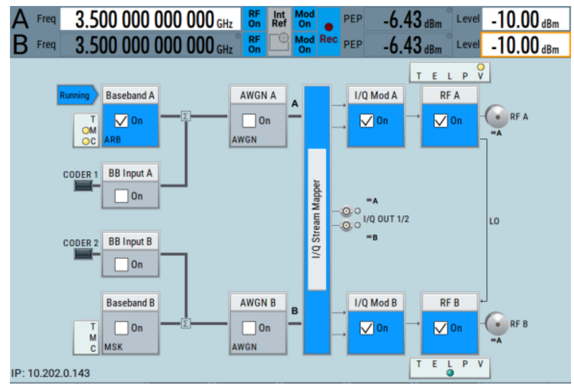
- Map the waveform to BOTH RF paths of the generator.
- e.g. “IQ Stream Mapper” and select “Stream A” + “RF B”
- Using SCPI:
- `:SCONfiguration:APPLy`
  `:SCONfiguration:OUTPut:MAPPing:RF2:STReam1:STATe 1`



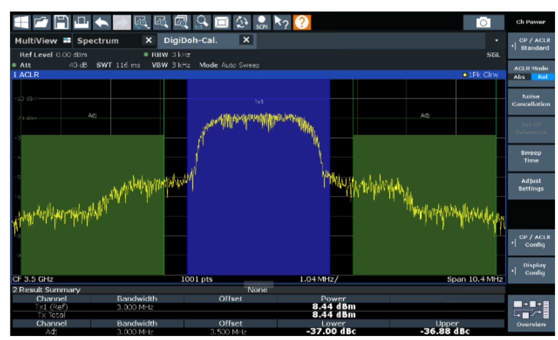
- Use a common LO for both RF Generator paths. In this example case, derived from the internal reference.
- e.g. “RF B”->”LO Coupling”->””Mode”->”A Internal & A->B Coupled”
- Using SCPI:
- :SOURce1:FREQuency:LOSCillator:MODE COUP



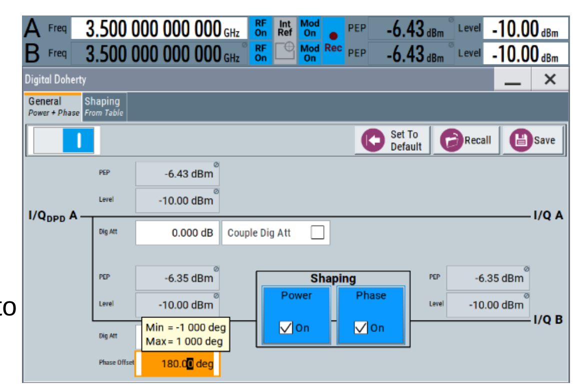
- Select the desired operating frequency and power level. Switch on the RF outputs.
- e.g. “RF A”->”Level”->”-10 dBm”
  “RF B”->”Level”->”-10 dBm”
  “RF A”->”Frequency”->”3.5 GHz”
  “RF A”->”On”
  “RF B”->”On”
- Using SCPI:
- :SOURce1:FREQuency 3500000000
  :SOURce1:POWer:POWer -10
  :SOURce2:POWer:POWer -10
  :OUTPut1:STATe 1
  :OUTPut2:STATe 1



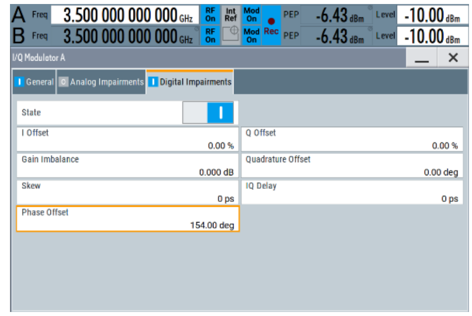
- Now, the signal from the two RF channels of the SMW, passes through the combiner and should appear on the FSW



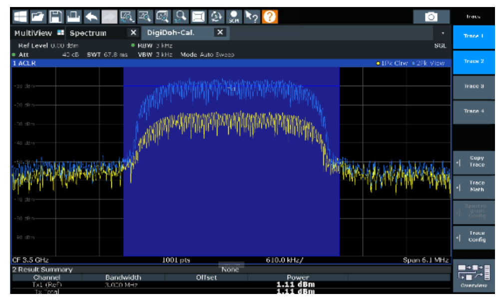
- To prepare for the calibration alignment itself, the SMW-K546 software shall be set to an anti-phase condition.
- e.g. “I/Q Mod B”->”Digital Doherty”->”Phase Offset”->”180 degrees”
- Note by default that “Shaping” tab uses the default “From Table”, shaping is switched off, and the attenuator settings are automatically set to 30 dB, to help protect the DUT.
- If it is safe to do so, the “Dig Att” settings may be adjusted and default zero shaping enabled.
- Using SCPI:
- :SOURce2:IQ:DOHerty:PHASe:OFFSet 180
  :SOURce2:IQ:DOHerty:SHAPing:POWer:STATe 1

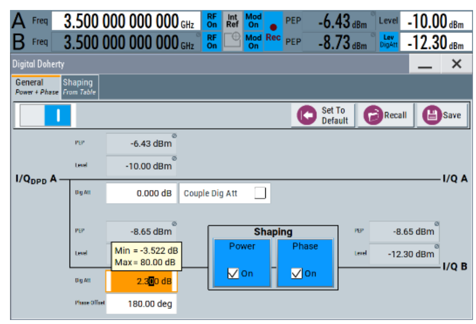


```
:SOURce2:IQ:DOHerty:SHAPing:PHASe:STAT
e 1
:SOURce:IQ:DOHerty:STATe 1
```

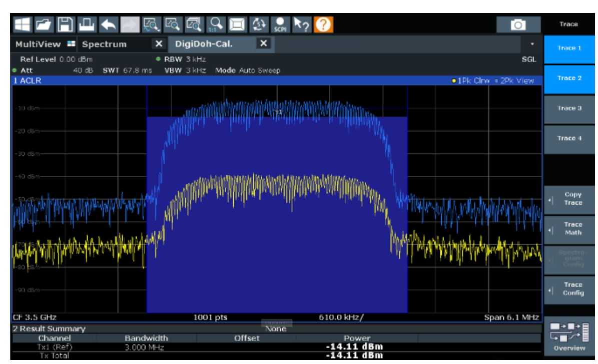


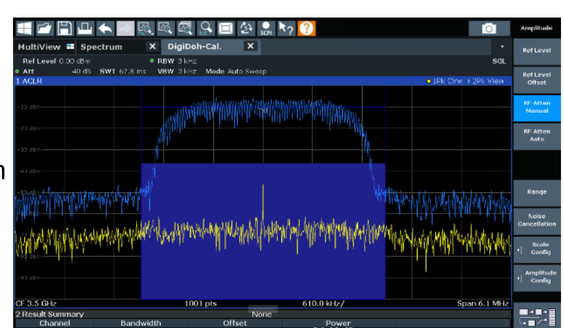
- Calibration is performed by adjusting the amplitude and phase of the two channels and minimizing the output signal level at the combiner output.
- The phase should be adjusted in Path A.
- e.g. “I/Q Mod A”->”I/Q Digital Impairments”->”State” and “Phase Offset”
- Using SCPI:
- `SOURce1:BB:IMPairment:RF1:STATe 1`
  `SOURce1:BB:IMPairment:RF1:POFFset 154`





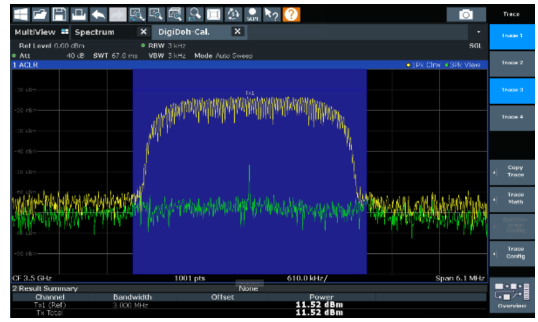
- The amplitude is best adjusted using the digital attenuation, in the SMW-K546, also in the channel which provides an attenuation (an indicated positive number), rather than a gain (a negative number)
- e.g. “I/Q Mod B”->”Digital Doherty”->”Dig Att”
- Using SCPI:
- `:SOURce2:IQ:DOHerty:POWer:ATTenuation 2.3`



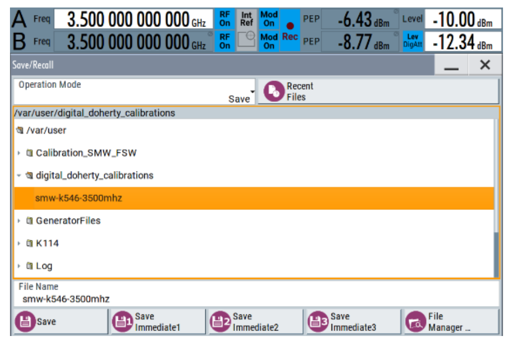


- Repeat steps 8 and 9, to ensure the best possible attenuation of the signal at the combiner output.
- Here, the result is shown after two iterations.
- Note: the tone present in the middle of the spectrum is residual LO leakage from the two signal paths. This may also be attenuated by adjusting I and Q offsets in both A and B “I/Q Modulator” -> “Digital Impairment” blocks.

- The performance of the calibration may be quickly verified by switching off RF B and comparing the signal levels before and after.
- In this case, a difference of more than 47dB can be seen between the two states, even though Laboratory Amplifiers (Mini-Circuits® ZHL-42) have been used.
- It is this difference number, the achieved cancellation, that describes the calibration accuracy.
- e.g. “RF B”->”Off”
- Using SCPI:
- `:OUTPut2:STATe 0`



- The calibrated SMW setting may now be saved.
- e.g. `“SAVE/RCL”->”SAVE”-><filename>`
- Using SCPI:
- `:SYST:SAV '/var/user/<filename>'`



- The calibration process may be repeated for as many RF and Power level points as the user wishes.
- In the test phase, these settings may simply be recalled.
- Using SCPI:
- `:SYST:RCL '/var/user/<filename>'`

The calibration fixture may now be disconnected, and the DUT inserted.

A summary of the SCPI command sequences to facilitate calibration is included in the Appendices.

Tips:

- Be sure to minimize applying mechanical stresses to the test set-up during the exchange phase.
- Switch off the RF outputs of the signal generator and de-energize the driver amplifier stages where possible.
- Keep the calibration fixture handy, for post-verification of the calibrations after the DUT measurements.

5 Testing the Amplifier DUT

In the next step, the calibration device is removed and the DUT inserted into the circuit.

The proper procedure for energizing the specific PA under test should be followed.

5.1 Linear

“Linear” means that the signal presented to the DUT second input is not “shaped”, rather that the two signals differ only in their relative amplitude and/or phase and/or delay.

This testing phase would ordinarily be done for all of the amplifier architectures.

For this test, the “Shaping” function may be either disabled or enabled, as long as the applied shaping is linear (for example, using the default table setting).

The screens presented to the user by the SMW-K546 software are shown in Fig. 5-1.

Fig. 5-1: Two GUI screens of the SMW-K546 software, (i) the "General" tab, with overview of the splitting and (ii) the "Shaping" tab, with its control of the Doherty auxiliary path characteristics.

A simple, example, measurement flow for this measurement is shown in Fig. 5-2. This kind of nested loop results in a detailed sweep within the bounds of the swept values. Optima may be discovered in the post-processing phase.

Alternatively, the designer might implement a search algorithm to optimize a specific parameter, like saturated output power or efficiency at a specific average output power. Depending on the algorithm used and the amplifier's characteristics, this might lead to a false positive result.

Typical swept (independent) parameters include:

► Frequency (`SCPI: :SOURce1:FREQuency:CW <freq>`)

► RF Level (`SCPI: :SOURce1:POWer:POWer <power>`)

and from within the SMW-K546:

► Amplitude Offset using "Dig Att" in the channel NOT used for calibration (`SCPI: :SOURce1:IQ:DOHerty:POWer:ATTenuation <value>`)

► Phase Offset (`SCPI: :SOURce2:IQ:DOHerty:PHASe:OFFSet 180`)

► Typical measured (dependent) quantities might include, from the FSW:

► Average Output Power

► Output PAR (peak to average power ratio)

► Linearity (e.g. ACLR, EVM, AM-AM and AM-PM curve width)

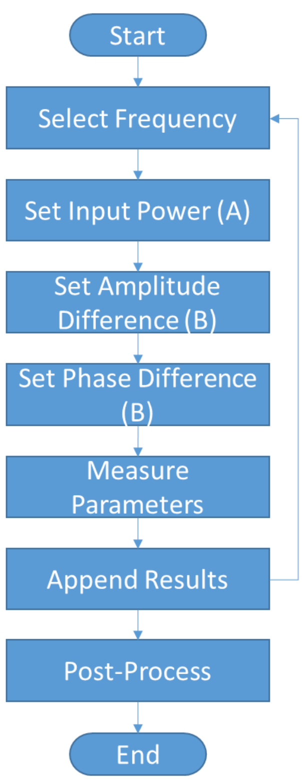


Fig. 5-2: Example measurement flow, for characterizing the dual-input amplifier.

It is usually beneficial also to record the power consumption from, e.g. the relevant power supply. This enables an energy efficacy metric to be calculated (e.g. efficiency, power wasted).

5.2 Non-Linear

The non-linear test phase is applicable generally only to the Doherty and Load Modulated Balanced amplifier types.

As the name suggests, the range of stimulus that could potentially be applied is much greater than for the linear test phase.

There are four different types of shaping profile that are catered for in SMW-K546, as shown in Fig. 5-3.

- ► Table
- ► Polynomial
- ► Normalized
- ► Classic Doherty

These are accessed from the “Digital Doherty”->”Shaping” menu.

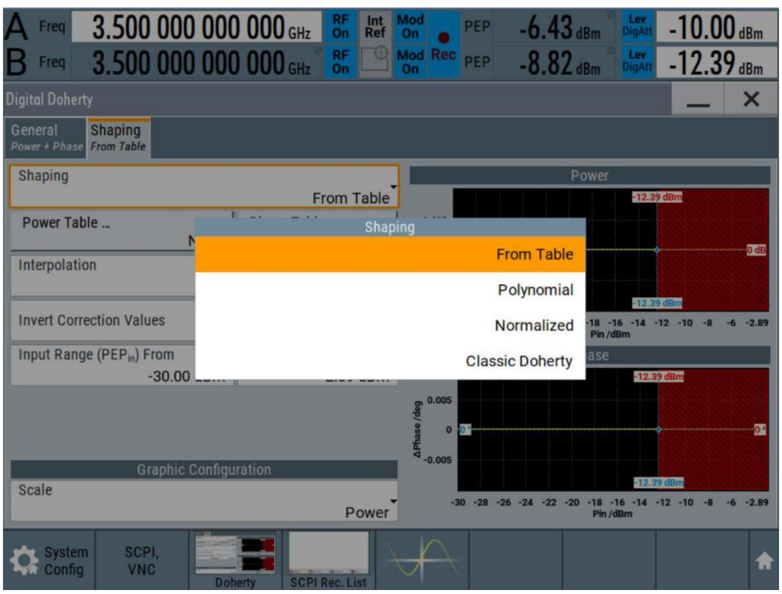


Fig. 5-3: The four types of shaping capabilities of the SMW-K546.

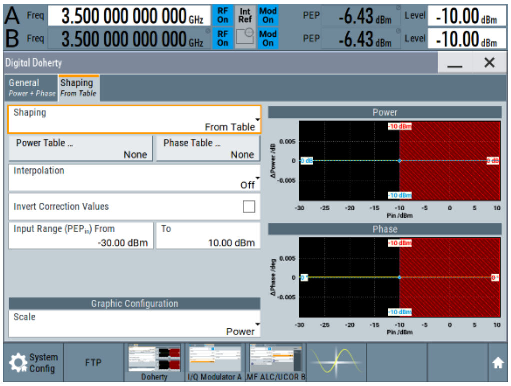
5.2.1 “From Table” Shaping

Power and/or Phase Tables for shaping may be created remotely and loaded into the SMW, or programmed via the SMW front panel.

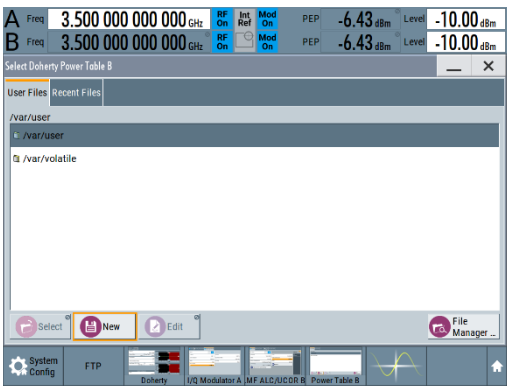
The following procedure shows how an amplitude table can be created. The same procedure applies for the creation of a phase table.

Note that the amplitude and phase values are deviations from a nominal, linear, 0 value.

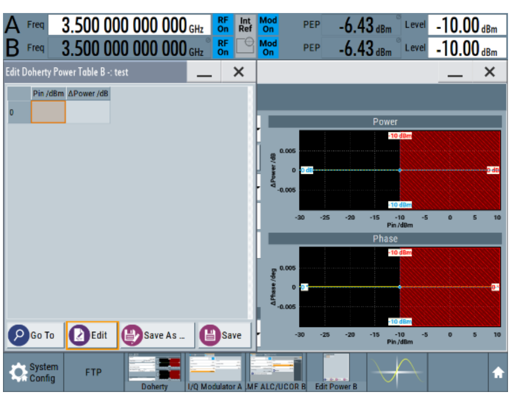
To create a table manually:



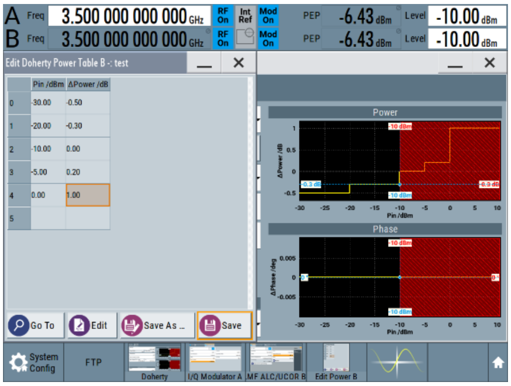
► Via the front panel, press the “Power Table…” (“Phase Table…”) key.



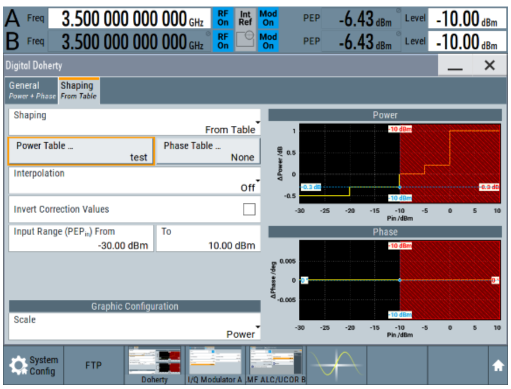
► From the dialog box, select the directory for storage and then press “New”.
Enter a filename.



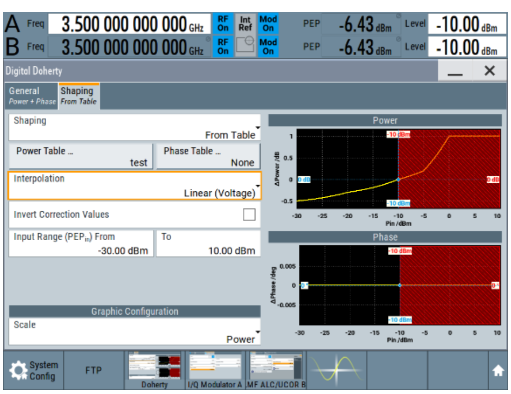
► Press “Edit” and values may be entered spreadsheet style.



► Enter values for power, and amplitude (or phase) deviation.
When the process is finished, hit “Save”.



► The filename of the created shape appears in the “Power Table” dialog box.



► Note that further options are available, especially for interpolation of the table values as well as inversion.

Commonly used SCPI commands used during remote control of the “Table” shaping parameters are shown in Table 5-1.

Action	**Command**
Invoke Table shaping operation	:SOURce2:IQ:DOHerty:SHAPing:MODE TABL
Load an existing, or create a new, amplitude shaping table	:SOURce2:IQ:Doherty:SHAPing:TABLe:AMAM:FILE:SELect <filename>
Apply interpolation to the loaded table(s)	:SOURce2:IQ:DOHerty:SHAPing:TABLe:INTerp LIN

Table 5-1: Commonly used SCPI protocol for "Table" shaping.

Example ASCII file-based definition of a “Table” shaped AM-AM characteristic looks like:

```
# Rohde & Schwarz - Digital AM/AM Predistortion Table
# Pin[dBm],deltaPower[dB]
-10,-1
-9,-0.8
-8,-0.5
-5,0
```

The filename extension should be “.dpd_magn” for amplitude shaping.

For phase shaping, the extension is “dpd_phase”, and typical file format and content, thus:

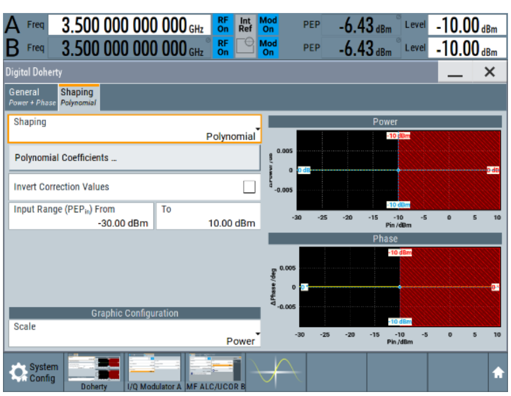
```
# Rohde & Schwarz - Digital AM/PM Predistortion Table
# Pin[dBm],deltaPhase[deg]
-30,0
-25,5
-10,25
0,25
```

In addition to easy creation through the touch screen interface, these shaping characteristics can be created in a third party software, and loaded into the instrument.

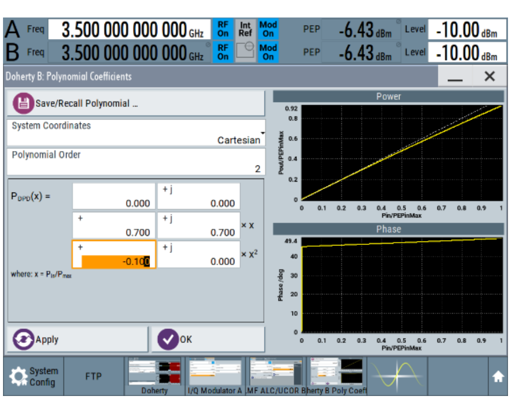
5.2.2 “Polynomial” Shaping

Polynomial programming is accessed by selecting the “Polynomial” option.

The polynomial describes the input-output characteristic of the shaping function.



► Start point for programming “Polynomial” shaping



► “System Coordinates” allows the entry of either “Cartesian” or “Cylindrical” numerology

► “Polynomial Order” allows for up to 15th degree coefficient definition

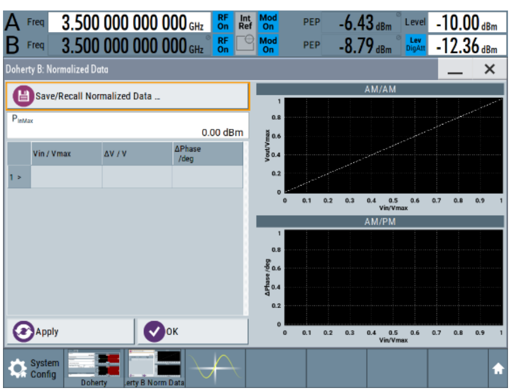
Commonly used SCPI commands used during remote control of the “Table” shaping parameters are shown in Table 5-2.

Action	**Command**
Invoke Polynomial shaping operation	:SOURce2:IQ:DOHerty:SHAPing:MODE POLY
Create a polynomial, with exemplary coefficients	:SOURce2:IQ:DOHerty:SHAPing:POLYnomial:COEFficients 0,0,0.707,0.707
Apply an inversion to the programmed characteristic	:SOURce2:IQ:DOHerty:SHAPing:TABLe:INVert 1

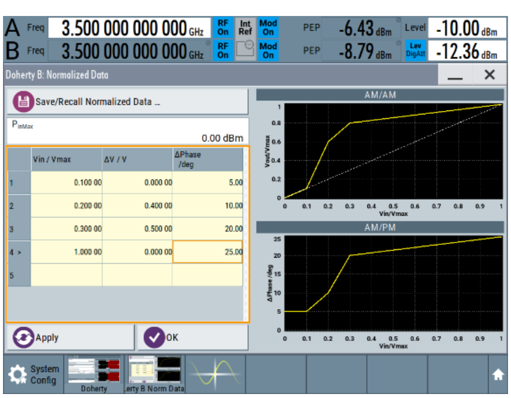
Table 5-2: Commonly used SCPI protocol for "Polynomial" shaping.

5.2.3 Normalized Shaping

Normalized, or “piecewise”, characteristic programming is accessed by selecting the “Normalized” option.

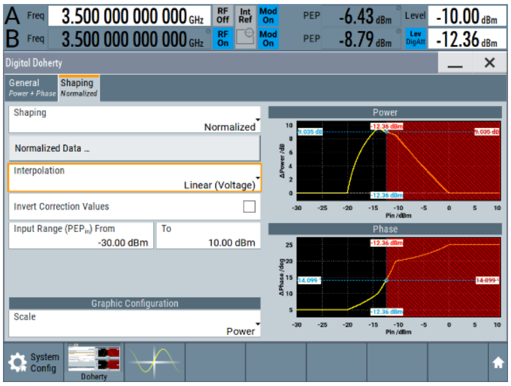


► Entering the “Normalized” shaping option brings up the options tab.
The user has the option of recalling pre-saved data, or to enter new data.
Press “Normalized Data…” to commence manual entry of shaping values



► The normalized or piecewise data is entered row by row, specifying deviation from linear values.
Note: Often, but not always, the first and last amplitude (V/V) values should be zero

► When entry is complete, hit “Apply”. If everything is in order, hit “Okay”.



► In the main “Normalized” shaping tab, the user now has the option to apply interpolation and/or to invert the characteristic, simply by selecting those options.

The following SCPI commands are commonly used for remote control of the “Normalized” or piecewise shaping parameters:

Action	Command
Invoke Normalized operation	SOURce2:IQ:DOHerty:SHAPing:MODE NORM
Load a pre-created “Normalized” or piecewise table	:SOURce2:IQ:DOHerty:NORMalized:DATA:LOAD <filename>

Table 5-3: Commonly used SCPI protocol for "Normalized" shaping.

Example ASCII file-based definition of a “Normalized” or piecewise characteristic looks like:

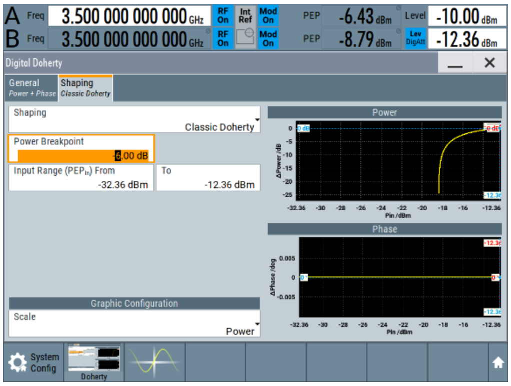
```
# Rohde & Schwarz - Digital Predistortion Normalized Table Data
# PinMax [dBm]
# number of points
# Vin/Vmax, deltaV/V, deltaPhase [deg]
0
4
0.1,0,5
0.2,0.4,10
0.3,0.5,20
1,0,25
```

The filename suffix should be “.dpd_norm”.

In addition to easy creation through the touch screen interface, these shaping characteristics can be created in a third party software, and loaded into the instrument.

5.2.4 Classic Doherty

Classical Doherty shaping, with the dogleg, or hockey stick shaped characteristic can also be generated by the software.



► With the “Classic Doherty” option selected, the user has the option to modify the breakpoint – the point at which the Doherty dogleg characteristic kicks in.
The user may also modify the range of input powers over which that range is effective.

The following SCPI commands are commonly used for remote control of the Classic Doherty parameters:

Action	Command
Invoke Classic Doherty operation	SOURce2:IQ:DOHerty:SHAPing:MODE DOH
Modify the Doherty Breakpoint power back-off	SOURce2:IQ:DOHerty:SHAPing:POWer:BREakpoint -6
Modify the upper input power level, at which the shaping characteristic is to be applied.	:SOURce2:IQ:DOHerty:PIN:MAX -12.36

Table 5-4: Commonly used SCPI protocol for "Classic Doherty" shaping.

6 Example Scripts & Command Sequences

6.1 Useful SCPI sequences

The following are SCPI sequences useful for getting started with the SMW-K546 and FSW-K18 (Amplifier Analysis) products.

A Matlab script and class, capable of parsing and sending these SCPI commands, without the need for Instrument Control Toolbox is provided later in the Chapter.

6.1.1 Prepare the SMW for calibration

The following example sequence of SCPI commands prepares the SMW, from a reset (*RST) in readiness for a calibration of amplitude, phase and delay. Note the used filename should be changed, as should frequency, etc.

```
% Phase 1 of preparation (set up general settings)
:SCONfiguration:APPLy
:SCONfiguration:OUTPut:MAPPing:RF2:STReam1:STATe 1
:SCONfiguration:OUTPut:MAPPing:IQOutput1:STReam1:STATe 0
:SCONfiguration:OUTPut:MAPPing:IQOutput2:STReam2:STATe 0
:SOURce1:FREQuency:LOSCillator:MODE COUP
:SOURce1:BB:ARBitrary:NOTCh1:APPLy
:SOURce1:BB:ARBitrary:WAVeform:SELect <filename>
:SOURce1:BB:ARBitrary:STATe 1
:SOURce1:POWer:POWer -10
:SOURce2:POWer:POWer -10
:SOURce1:FREQuency:CW 1E9
:OUTPut1:STATe 1
:OUTPut2:STATe 1
:SOURce1:BB:IMPairment:RF1:STATe 1
:SOURce1:BB:IMPairment:RF2:STATe 1
% Phase 2 of preparation (set up SMW-K546)
:SOURce2:IQ:DOHerty:POWer:ATTenuation 0
:SOURce1:IQ:DOHerty:POWer:ATTenuation 0
:SOURce2:IQ:DOHerty:PHASe:OFFSet 180
:SOURce:IQ:DOHerty:STATe 1
:SOURce2:IQ:DOHerty:SHAPing:POWer:STATe 1
:SOURce2:IQ:DOHerty:SHAPing:PHASe:STATe 1
```

6.1.2 Prepare the FSW for calibration

The following example sequence of SCPI commands prepares the FSW, from a reset (*RST) in preparedness for a calibration of amplitude, phase and delay.

```
:INST:CRE:NEW SANALYZER, 'SMW-K546-Cal.'
:INIT:CONT OFF
:CALC:MARK:FUNC:POW:SEL ACP
:SENS:POW:ACH:ACP 1
:SENS:FREQ:CENT 1E9
:SENS:POW:ACH:BWID:CHAN1 3000000
```

```
:SENS:POW:ACH:BWID:ACH 3000000
:SENS:POW:ACH:SPAC:CHAN1 3500000
:SENS:POW:ACH:SPAC:ACH 3500000
:INIT:IMM; *WAI;
:SENS:BAND:RES 3000
:SENS:WIND:DET:FUNC POS
:INP:ATT 40
```

6.1.3 Modify Phase & Amplitude to detect null condition

By adjusting the phase and amplitude difference between the signal paths, the cancellation condition can be detected.

Phase may be adjusted with the following command, and example value:

```
:SOURce1:BB:IMPairment:RF1:POFFset 151
```

Amplitude should be adjusted using the “digital attenuation” feature, ideally avoiding negative values by writing to one or other of the RF paths, using either of the pair of commands, with associated example values:

```
:SOURce1:IQ:DOHerty:POWer:ATTenuation 1.34
:SOURce2:IQ:DOHerty:POWer:ATTenuation 0
```

or

```
:SOURce2:IQ:DOHerty:POWer:ATTenuation 3.12
:SOURce1:IQ:DOHerty:POWer:ATTenuation 0
```

After each iteration or change in amplitude or phase difference, the residual, summed signal level at the output of the power combiner may be queried from the FSW, with the command:

```
CALC:MARK:FUNC:POW:RES? ACP
```

6.2 Useful Matlab® Scripts

Rohde & Schwarz® instrument communication using the Matlab® core package is made easy by the free "VISA_Instrument" class, downloadable from http://www.rohde-schwarz.com/appnote/1MA171.

Some of the functions tabulated here rely on VISA resource strings for the instruments.

An example variable set-up for this resource in the Matlab® environment looks like

```
rscSMW = 'TCPIP::10.202.1.143::INSTR';
rscFSW = 'TCPIP::10.202.1.26::INSTR';
```

6.2.1 Create A Random, “Table” based, shaping function

The MATLAB® script, (1) creates a randomized shaping function, as a complex input-output voltage transfer function, (2) converts it to the power/phase format used by the instrument (3) passes the characteristic to the instrument itself.

It makes use of the VISA_Instrument class, which enables instrument control, without the need for the Instrument Control Toolbox.

It also calls on two additional proprietary functions, “convert_vin_vout_to_K546TABL()” and “upload_K546TABL_to_SMW”, which are explicitly tabulated in this section.

```
%
% This script creates a simple, randomised shaping function for the SMW-K546
% It leverages the pchip feature of Matlab(R) for smoothing.
%
% The script family make use of additional, proprietary functions:
%     convert_vin_vout_to_K546TABL: for converting complex voltages to the
%     power/phase format used by the SMW200A instrument.
%     upload_K546TABL_to_SMW: for uploading the created shaping tables to
%     the SMW-K546 software.
%     sendSCPIcmd: which forms a SCPI command for submission to the
%     instrument
%     VISAInstrument: which enables instrument control and communication,
%     without the Instrument Control Toolbox. See
%     http://www.rohde-schwarz.com/appnote/1MA171 for more info.
%
% (C) Rohde & Schwarz 2018

% create a simple input vector, and a finer vector to which the coarse
% characteristic shall be interpolated.
vin_crude = [0.01 0.5 1];
vin = linspace(min(vin_crude),max(vin_crude),100);

% randomise a doherty amplitude- and phase- shaping
vout_amp_crude = [0.01 0.5*rand() 1];
% the phase shaping is allowed to deviate by +/- 30 degrees from reference
vout_phs_crude = [30*(rand(1)-0.5) 30*(rand(1)-0.5) 30*(rand(1)-0.5)];

% interpolate/soften/smooth the characteristic using pchip
vout_amp = pchip(vin_crude, vout_amp_crude, vin);
vout_phs = pchip(vin_crude, vout_phs_crude, vin);

% create a complex value pair amplitude series for the characteristic
vout = vout_amp .* exp(1i * (pi/180) * vout_phs); % pchip example

% plot the shaping characteristic
figure(1)
subplot(2,1,1);plot(abs(vin), abs(vout)); grid on;
ylabel('Output Voltage Amplitude');
subplot(2,1,2);plot(abs(vin), 180*angle(vout)/pi); grid on;
xlabel('Input Voltage Magnitude');
ylabel('Output Phase /deg.');

% convert voltages to K546 table format
[k546_AM,k546_PM] = convert_vin_vout_to_K546TABL(vin, vout);
% upload the table to the instrument
upload_K546TABL_to_SMW(rscSMW, 1, 'dohAmp', 'dohPhs', k546_AM, k546_PM);
```

6.2.2 Convert complex input-output voltage pairs to power-phase table format.

The function converts the conventional complex valued vin-vout pairs to the power-phase format used by the SMW-K546 product.

```
function [ioAmplitude,ioPhase] = convert_vin_vout_to_K546TABL(vin,vout)
%convert_vin_vout_to_K546TABL  Converts standard abs(input voltage) and
%complex output voltage to SMW-K546 TABL format.
%Example
%vin = linspace(0.01,1,100);
%vout = power(abs(vin),2) .* exp(1i*pi/4*power(abs(vin),2));
%[k546_AM,k546_PM] = convert_vin_vout_to_K546TABL(vin, vout);
%upload_K546TABL_to_SMW(rscSMW, 2, 'dohAmp', 'dohPhs', k546_AM, k546_PM)
% (C) Rohde & Schwarz 2018

%% Pre-process the voltages
    npts = length(vin);
    pin = 20*log10(abs(vin));
    powOut = 20*log10(abs(vout));
    powOut = powOut-pin;
    phsOut = 180*angle(vout)/pi; % SMW would like to see degrees, not radians

%% build the TABL array
    strAM1DPD = [];
    strPM1DPD = [];

    for loop = 1:npts
        strAM1DPD = [strAM1DPD, [num2str(pin(loop)), ',' num2str(powOut(loop)),
',']];
        strPM1DPD = [strPM1DPD, [num2str(pin(loop)), ',' num2str(phsOut(loop)),
',']];
    end

    ioAmplitude = strAM1DPD(1:end-1);
    ioPhase = strPM1DPD(1:end-1);

end
```

6.2.3 Upload shaping table to instrument.

```
function upload_K546TABL_to_SMW(rscSMW, numSource, nameAmpTABL, namePhsTABL,
ioAmplitude, ioPhase)
%upload_K546TABL_to_SMW Send two vectors (amplitude and phase, input and
%output values) as a table to the SMW. Usually preceeded with
%convert_vin_vout_to_K546TABL
%Example
%vin = linspace(0.01,1,100);
%vout = power(abs(vin),2) .* exp(1i*pi/4*power(abs(vin),2));
%[k546_AM,k546_PM] = convert_vin_vout_to_K546TABL(vin, vout);
%upload_K546TABL_to_SMW(rscSMW, 2, 'dohAmp', 'dohPhs', k546_AM, k546_PM)
% (C) Rohde & Schwarz 2018

%%

strInstr = [':SOURce', num2str(numSource), ':IQ:DOHerty'];
sendSCPIcmd(rscSMW, [strInstr, ':SHAPing:MODE TABL'], 1, 0);
sendSCPIcmd(rscSMW, [strInstr, ':SHAPing:TABLe:AMAM:FILE:NEW "', nameAmpTABL,'",
', ioAmplitude], 1, 0);
sendSCPIcmd(rscSMW, [strInstr, ':SHAPing:TABLe:AMPM:FILE:NEW "', namePhsTABL,'",
', ioPhase], 1, 0);

end
```

6.2.4 Send an individual SCPI command to instrument, receive a response

This script enables single SCPI commands to be sent to instruments, and instrument responses to those commands to be collated and returned.

```
function [instrumentSays] = sendSCPIcmd(visaResource, cmdToSend, withOPC,
withReset)
% sendSCPIcmd Directly sends a SCPI string to an instrument resource.
%
% Queries are detected, their collective responses are collated,
% and appended to a string, which is returned as a function output.
%
% Example: sendSCPIcmd('TCPIP::10.85.0.167::INSTR', ':SOURce1:FREQuency:CW
3600000000', 1, 0)
% Without an initial reset, instructs the signal generator with IP address
% 10.85.0.167, to change its center frequency to 3.6GHz
%
% (C) Rohde & Schwarz 2018

%% detect if the resource is already a connected... if not, create

    if (ischar(visaResource))
        objInstr = VISA_Instrument(visaResource);
        openClose = true;
    elseif (isa(visaResource, 'VISA_Instrument'))
        objInstr = visaResource;
        openClose = false;
    else
        error('"visaResource" parameter must be either ResourceName string or
"VISA_Instrument" object');
    end

%% prepare the reply string
    instrumentSays = [];

%% reset, if requested
    if withReset
        objInstr.Write('*RST'); objInstr.QueryString('*OPC?');
    end

%% prepare and write the command/query
tline = cmdToSend;
    if strfind(tline,'?')
        instrumentSays = strcat(instrumentSays, ',',
objInstr.QueryString(tline));
    else
        objInstr.Write(tline);
        if withOPC
        objInstr.QueryString('*OPC?');
        end
    end

%% if the resource was not passed already opened, then close it
    if (openClose)
```

```
        objInstr.Close();
    end

%% pre-process the instrument response
    if length(instrumentSays)>1
        instrumentSays = strrep(instrumentSays, 'NAN', 'nan');
        instrumentSays = strrep(instrumentSays(2:end),';',',');
        if isnumeric(instrumentSays)
            instrumentSays = str2num(instrumentSays);
        end
    end
end
```

6.2.5 Parse an ASCII file of SCPI commands, collate responses

The function parses an ASCII file of SCPI commands, line-by-line and sends them to an instrument. Responses to the commands are collated and returned.

```
function [instrumentSays] = parseSCPIFile(visaResource, fileToParse, withOPC,
withReset)
% PARSESCPIFILE Reads an ASCII file of SCPI commands, parses line-by-line, sends
to
% resource.
% Queries are detected, their collective responses are collated,
% and appended to a string, and returned from the function.
% The 'percent' character, as per Matlab usage, denotes a comment in the
% .scpi file.
%
% Example: parseSCPIFile('TCPIP::10.85.0.167::INSTR', 'test.scpi', 1, 0)
% Without an initial reset, parses the content of test.scpi and sends it to
% the resource with IP address 10.85.0.167

%%
% Check if visaResource is already a resource, if not, create it
    if (ischar(visaResource))
        objInstr = VISA_Instrument(visaResource);
        openClose = true;
    elseif (isa(visaResource, 'VISA_Instrument'))
        objInstr = visaResource;
        openClose = false;
    else
        error('"visaResource" parameter must be either ResourceName string or
"VISA_Instrument" object');
    end

%%
% instance the array for collating data
instrumentSays = [];

if withReset
    objInstr.Write('*RST'); objInstr.QueryString('*OPC?');
end

try
    fid = fopen(fileToParse);
    tline = fgetl(fid);
```

```
    while ischar(tline)
        %disp(tline);
        if length(tline)>0
            if (tline(1) ~= '%') && (tline(1) ~= ' ') && (tline(1) ~= '#') &&
(tline(1) ~= '/')
                if strfind(tline,'?')
                    instrumentSays = strcat(instrumentSays, ',',
objInstr.QueryString(tline));
                else
                    objInstr.Write(tline);
                    if withOPC
                        objInstr.QueryString('*OPC?');
                    end
                end
            end
        end
        tline = fgetl(fid);
    end
    fclose(fid);
    objInstr.Close();
    if instrumentSays
        instrumentSays = str2num(strrep(instrumentSays(2:end),';',','));
    end
catch
    disp('Could not open SCPI file');
end

end
```

7 Further Reading & Acknowledgements

From the Rohde & Schwarz ® website, at www.rohde-schwarz.com:

Optimizing the Perennial Doherty Power Amplifier - Reprint from the measurement-aided design article published in Microwave Journal (March 2019).

1GP67 – "Phase Adjustment of Two MIMO Signal Sources with Option B90"

1MA171 – "How to use Rohde & Schwarz Instruments in MATLAB"

SMW200A – "R&S®SMW200A Vector Signal Generator User Manual"

FSW-K18 – "R&S®FSW-K18 Amplifier Measurements User Manual"

HMP-4040 – "R&S®HMP Series Power Supplies User Manual"

From the IEEE Xplore Digital Library:

"Doherty Goes Digital" (http://ieeexplore.ieee.org/document/7506420, subscription required)

8 Ordering Information

8.1 Instruments

Designation	Type	Order No.
Vector Signal Generator	R&S®SMW200A	1412.0000.02
Signal and Spectrum Analyzer	R&S®FSW26	1331.5003.26
Power Supply	R&S®HMP4040	3629.6776.04

8.2 Recommended minimum configuration for R&S®SMW200A

Option	Designation	Material	Qty.	Remark
	SMW200A Signal Generator		1	
SMW200A	Vector signal generator, base unit; freq.opt.& BBmodule req.	1412.0000.02	1	Mandatory
SMW-B1006	Frequency range: 100 kHz to 6 GHz for RF path A (HW opt.)	1428.4800.02	1	Mandatory
SMW-B2006	Frequency range: 100 kHz to 6 GHz for RF path B (HW opt.)	1428.5807.02	1	Mandatory
SMW-B13XT	Wideband baseband main module, 2 I/Q paths to RF (HW opt.)	1413.8005.02	1	Mandatory
SMW-B9	Wideband baseband generator, 500 MHz, 256 MS (HW opt.)	1413.7350.02	2	Mandatory
SMW-B90	Phase coherence (HW opt.)	1413.5841.02	1	Mandatory
SMW-B709	Low phase noise, RF path A (HW opt.)	1428.7300.02	1	Recommended
SMW-B719	Low phase noise, RF path B (HW opt.)	1428.7500.02	1	Recommended
SMW-K541	AM/AM, AM/PM Predistortion (SL)	1413.7267.02	2	Mandatory
SMW-K546	Digital Doherty (SL)	1414.6487.02	1	Mandatory
SMW-K548	Crest factor reduction (SL)	1414.6641.02	2	Optional
SMW-K144	5G New Radio (SL)	1414.4990.02	2	Optional

8.3 Recommended minimum configuration for R&S®FSW

Option	Designation	Material	Quantity	Remark
R&S®FSW26	Signal and spectrum analyzer 2 Hz to 26.5 GHz	1331.5003.26	1	Mandatory. Up to 85 GHz upper frequency can be configured
R&S®FSW-B160	Extension to 160 MHz signal analysis bandwidth	1325.4850.14	1	Mandatory. Internal analysis bandwidth up to 2 GHz can be configured
R&S®FSW-K18	Power amplifier measurement application	1325.2170.02	1	Mandatory

R&S®FSW-K18D	Direct DPD measurements	1331.6845.02	1	Optional

Rohde & Schwarz

The Rohde & Schwarz electronics group offers innovative solutions in the following business fields: test and measurement, broadcast and media, secure communications, cybersecurity, monitoring and network testing. Founded more than 80 years ago, the independent company which is headquartered in Munich, Germany, has an extensive sales and service network with locations in more than 70 countries.

www.rohde-schwarz.com

Certified Quality Management
ISO 9001

Rohde & Schwarz training

www.training.rohde-schwarz.com

Rohde & Schwarz customer support

www.rohde-schwarz.com/support

PAD-T-M: 3572.7186.00/01.00/EN

GFM345 | Version 1e | 01.2019
Application Note | R&S®SMW-K546 Digital Doherty