R&S®NRQ6 / R&S®SGT100A Power Servoing (R&S®NRQ6-K2) Application Sheet

Application Sheet

Version 01

ROHDE&SCHWARZ

1 Your Task

For component tests, it can be required to set the DUT output power to a specific value before starting the test of a test sequence.

This application sheet describes how to use the power servoing function of an R&S NRQ6 together with an R&S SGT100A signal generator to achieve this goal. As an example of a practical test, the measurement of the 3rd harmonic is shown, using both an R&S NRQ6 and a spectrum analyzer.

2 Possible Solution

Component tests are complicated by the fact that they can be defined as a function of the DUT output power, e.g. a power amplifier. To the test engineer, this dependency poses a problem because the DUT gain is not necessarily:

- Known beforehand
- Constant over the input power
- Constant over time or temperature

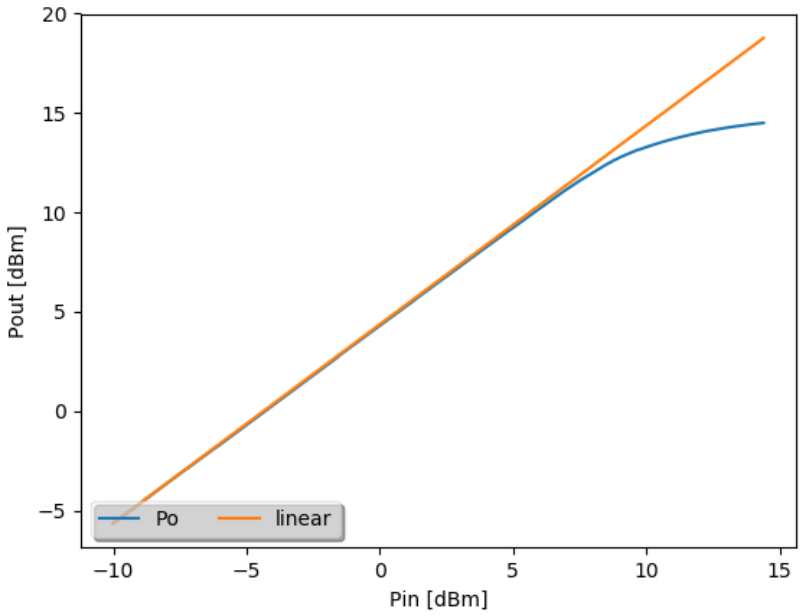

Figure 2-1: Output power vs. input power of an amplifier

This situation calls for a closed loop control that allows to measure the output power at the DUT iteratively and correct the signal generator power. Thus, in several steps, the required power at the DUT output is finally reached.

A traditional implementation of such a control loop would run on a computer, use SCPI commands to query a power sensor, compute the next value for the signal generator

power, and send it to the signal generator. These steps are repeated until the measured power is close enough to the desired value.

Depending on the start value, the allowed tolerance and the nonlinearity of the DUT, this procedure can take several 10 ms, which can be too long for automated test applications.

The power servoing functions of the R&S NRQ6 and R&S SGT100A offer a faster solution for this problem:

- The power sensor measures continuously and sends results using a dedicated serial connection to the signal generator - direct FPGA to FPGA communication, no SCPI overhead.
- The control algorithm runs on the signal generator - again, no SCPI overhead. The application only needs to set the desired level in one command.

By using the R&S NRQ6 / R&S SGT100A power servoing and the high-speed remote control of the R&S SGT100A, setting the desired power typically takes 1 ms to 1.5 ms.

3 Cabling and Configuration

Required equipment

- R&S NRQ6 power sensor with power servoing option (R&S NRQ6-K2)
- R&S SGT100A vector signal generator
- Computer
- SGMA GUI software for Windows and Red Hat / CentOS Linux operating systems. Available for download at: https://www.rohde-schwarz.com/software/sgt100a/
- Optional: spectrum analyzer, for example an R&S FSV3000 or R&S FPL1000
- Optional, if a spectrum analyzer is used: power splitter

Setup

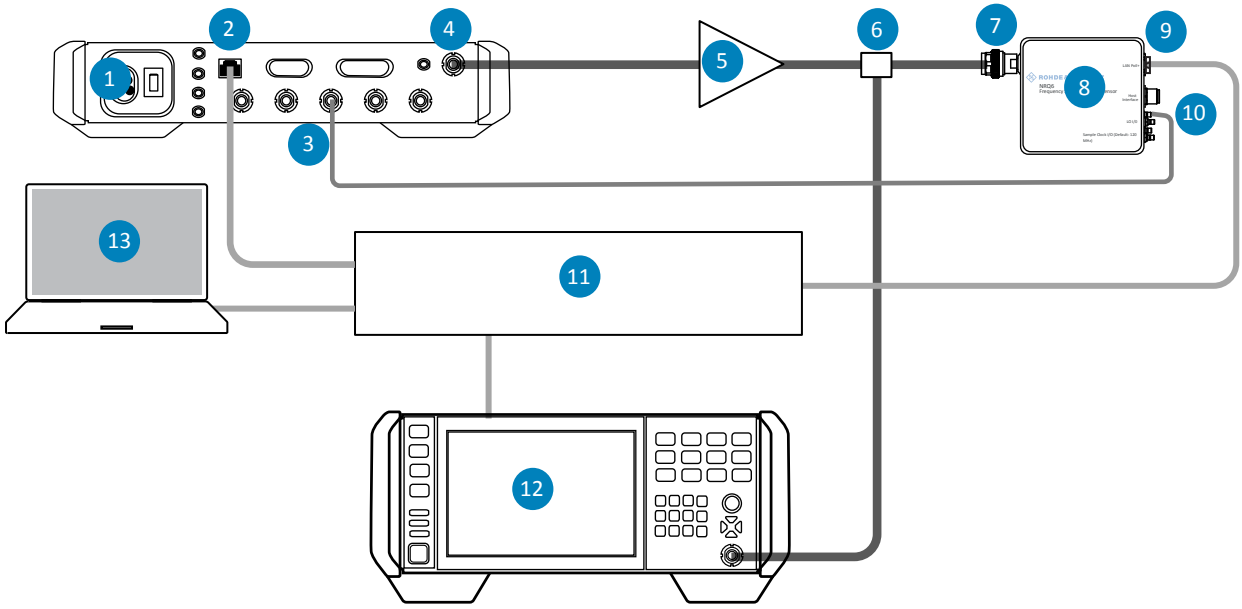

Figure 3-1: Cabling

1 = R&S SGT100A
2 = LAN interface
3 = USER 2 connector
4 = RF 50 Ω connector
5 = DUT
6 = Power splitter
7 = RF connector
8 = R&S NRQ6
9 = LAN PoE+ interface
10 = TRIG2 connector
11 = Ethernet switch supporting PoE+ power delivery, for example. Alternatively, you can use the other set-ups described in the user manual of the R&S NRQ6.
12 = Spectrum analyzer
13 = Computer

1. Connect the instruments as shown in Figure 3-1.
2. Make sure to connect the TRIG2 connector of the R&S NRQ6 to the USER 2 connector of the R&S SGT100A.
3. Connect all instruments and the computer to the local network.

Preparing the measurement

1. Switch on all instruments.
2. If the SGMA GUI software is not already available on the computer, install it.
3. Start the SGMA GUI software.
4. In the toolbar, click the blue icon to open the "Configure Instruments" dialog.

5. Click "Scan" to scan for the R&S SGT100A.

   The R&S SGT100A appears in the list of available instruments.

6. Enable the R&S SGT100A.
7. In the main dialog of the SGMA GUI, click the instrument settings button and select "RF" >"NRP Sensor Mapping".
8. Click "Scan".

   The R&S NRQ6 appears in the list.

9. Change the mapping of the R&S NRQ6 to "1".
10. In the main dialog of the SGMA GUI, click the "RF Off" to turn on the RF.
11. In the main dialog of the SGMA GUI, click the instrument settings button and select "RF" >"Level".
12. Select the "Power Servoing" tab.
13. Select the power sensor.

    You can perform steps step 4 to step 13 programmatically. To keep the procedure simple, the GUI is used here instead.

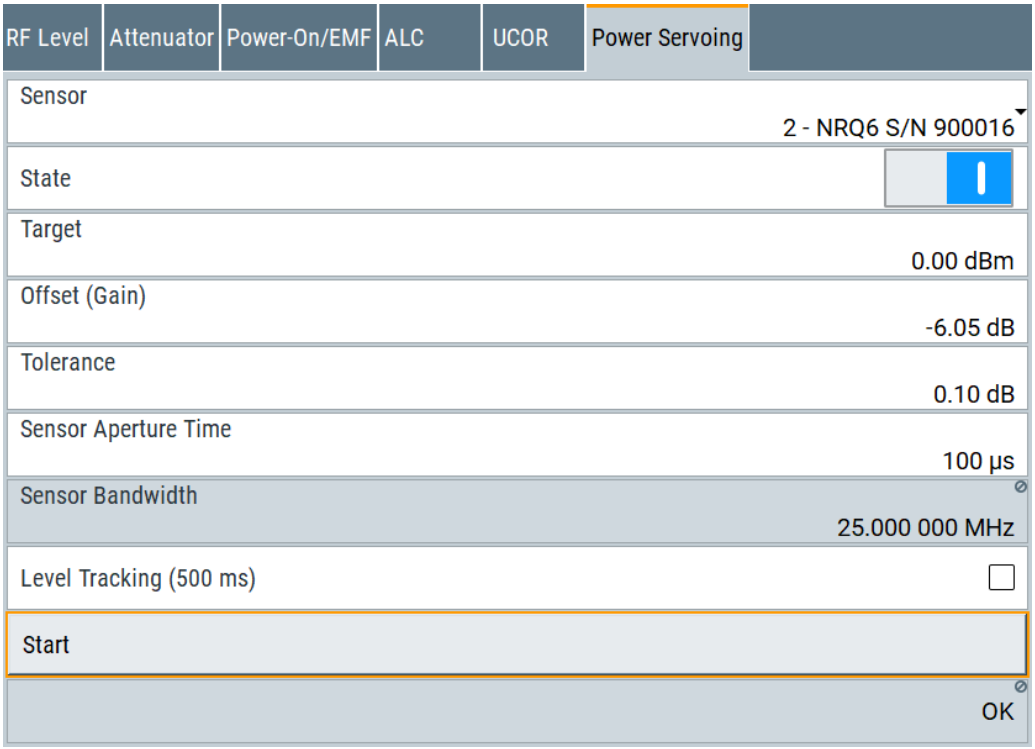

Figure 3-2: Power Servoing dialog of the SGMA GUI

Initial test

1. Enter a value in the "Target" field.
2. Set "State" to "I".
3. Click "Start".

   The result field at the bottom of the dialog shows "OK".
4. If it shows "FAIL":
   a) Click Info.
   b) Read the error history.
   c) Check your setup.
   d) Try again.

4 Application Examples

The code examples are for Python 3, but you can easily translate them into your favorite programming language. Explanations on the code are given below the code.

4.1 Measuring a Harmonic with the R&S NRQ6

For this example, the power splitter and spectrum analyzer shown in Figure 3-1 are not required.

```
def MeasureHarmonicWithNRQ(freq, harmonic, powerlist):
    '''
       Measure the nth harmonic of frequency freq at the output power
       points in powerlist using the NRQ6
    '''
    #(1) see explanations below
    NRQ.query ("*RST;*OPC?")
    NRQ.write ("UNIT:POW dbm")
    NRQ.write("SENS:BAND:RES:TYPE:AUTO:STAT OFF")
    rbw = 1e4
    aper = 1e-3
    NRQ.write("SENS:BAND:RES %f" % rbw) #(2)
    NRQ.write("FREQ %f" % (harmonic * freq)) #(3)
    NRQ.write("SENS:APER %f" % aper)

    SGT.query("*RST;*OPC?")
    SGT.write("FREQ %f" % freq)
    SGT.write("OUTP:STAT ON")
    SGT.write("POWer:SERV:TOL 0.1")
    SGT.write("POWer:SERV:SENSor:APERture 0.1e-3")
```

```
SGT.write("POWer:SERV:STATE ON") #(4)

harmoniclist = []
for pow in powerlist:
    res = SGT.query("POWer:SERV:SET? %f" % pow) #(5)
    if int(res):
        print (" %f dBm :fail" % pow)
        break
    else:
        SGT.query("POWer:SERV:STATE OFF;*OPC?") #(6)

        r = float(NRQ.query("INIT;:FETCH?")) #(7)
        harmoniclist.append(r - pow)    #(8)
        print ("%5.2f dBm\t%4.1f dBm\t%4.1f dB" %(pow, r, r - pow))

        SGT.write("POWer:SERV:STATE ON") #(9)

SGT.write("POWer:SERV:STATE OFF")
return harmoniclist
```

Explanations

(1)

Arguments of the `MeasureHarmonicWithNRQ` function are the fundamental frequency, the order of the harmonic, and a list of output power values where the measurements are performed.

(2)

To be able to measure very small levels, the resolution bandwidth is set to a small value. Note that such small resolution bandwidth values are not possible with a traditional wideband power sensor.

(3)

The sensor frequency is set to the nth harmonic. The R&S SGT100A overrides this value when enabling power servoing and restores it when disabling power servoing.

(4)

Once power servoing is enabled, the power sensor is under control of the R&S SGT100A. Make sure that the application does not access the power sensor while in this state.

(5)

When the actual power servoing operation starts, the call returns either:

- `0`: power has successfully been set to the desired value.
- `1`: signaling failure.

(6)

After power servoing has been disabled, the application can access the power sensor again and perform measurements.

(7)

The measurement is performed at the frequency of the harmonic.

(8)

`pow` is good enough as the reference, so no separate measurement is necessary to get the reference value at the fundamental. The power splitter is not considered.

(9)

Enabling and disabling power servoing incurs some delay, approx. 45 ms per pair.

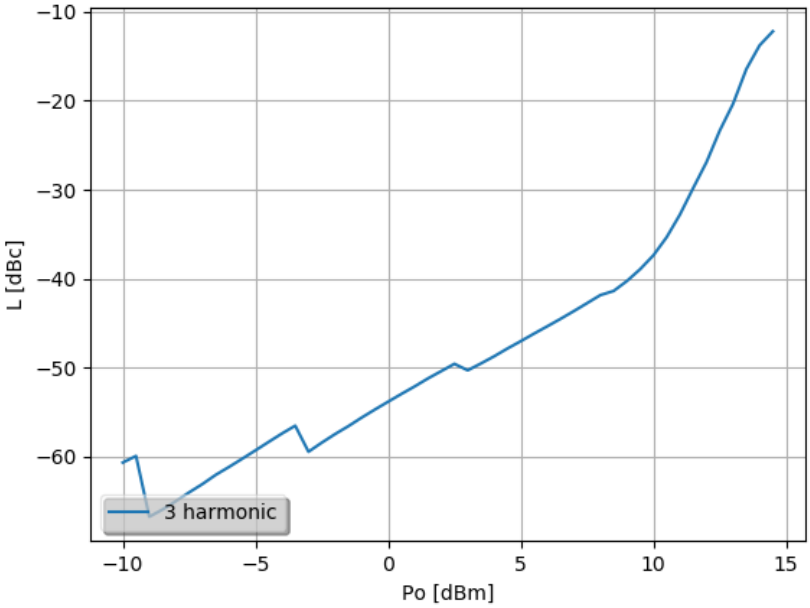

Figure 4-1: 3rd harmonic vs. output power measured with R&S NRQ6

4.2 Measuring a Harmonic with a Spectrum Analyzer

For this example, a power splitter and spectrum analyzer are required, as shown in Figure 3-1.

```
def MeasureHarmonicWithSA(freq, harmonic, powerlist):
    '''
        Measure the nth harmonic of frequency freq at the output power
        sample points in powerlist using a spectrum analyzer e.g. R&S SA
    '''
#(1) see explanations below
    span = 0
    rbw = 1e4
    sweeptime = 1e-3
    SA.query("*RST;*OPC?")
    SA.write("INIT:CONT OFF")
    SA.write("FREQ:SPAN %f" % span)
```

```
SA.write("BAND:RES %f" % rbw)
SA.write("SWE:TIME %f" % sweeptime)
SA.write("FREQUENCY:CENTER %f" % (freq * harmonic))

SGT.query("*RST;*OPC?")
SGT.write("FREQ %f" % freq)
SGT.write("OUTP:STAT ON")
SGT.write("POWer:SERV:TOL 0.1")
SGT.write("POWer:SERV:SENSor:APERture 0.1e-3")
SGT.write("POWer:SERV:STATE ON") #(2)

harmoniclist = []
for pow in powerlist:
    res = SGT.query("POWer:SERV:SET? %f" % pow) #(3)
    if int(res):
        print (" %f dBm :fail" % pow)
    else:
        SA.write("INIT:IMM;*WAI")
        r = float(SA.query ("CALC:MARK:Y?")) #(4)
        harmoniclist.append(r - pow)    #(5)
        print ("%5.2f dBm\t%4.1f dBm\t%4.1f dB" %(pow, r, r - pow))

SGT.write("POWer:SERV:STATE OFF") #(6)
return harmoniclist
```

Explanations

(1)

Arguments of `MeasureHarmonicWithSA` function are the fundamental frequency, the order of the harmonic, and a list of output power values where the measurements are performed.

(2)

Power servoing is enabled only once, and disabled only at the end of the function.

(3)

When the actual power servoing operation starts, the call returns either:

- `0`: power has successfully been set to the desired value.
- `1`: signaling failure.

(4)

The measurement is performed at the frequency of the harmonic.

(5)

`pow` is good enough as the reference, so no separate measurement is necessary to get the reference value at the fundamental. The power splitter is not considered.

(6)

Enabling and disabling power servoing is not required.

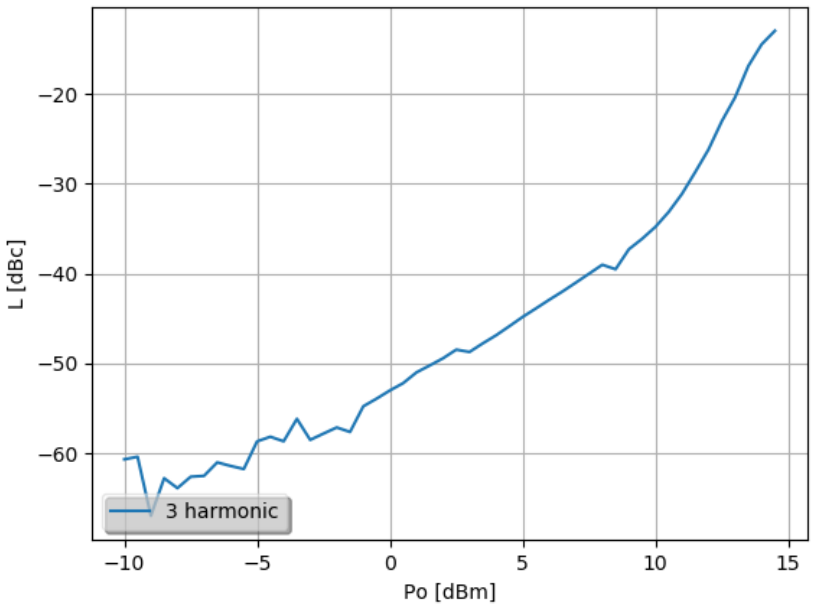

Figure 4-2: 3rd harmonic vs. output power measured with a spectrum analyzer

5 Learn More about the R&S NRQ6

For a detailed description of the capabilities of the R&S NRQ6, read its user manual. The user manual also explains all aspects of remote control features in details.

Also, you can always install our basic driver and tools package called R&S NRP Toolkit. Among various tools, this package supplies an optional SDK (software development kit), which contains many sample programs with full commented source code in various programming languages. On an MS Windows PC, you find the SDK after installation under:

`C:\ProgramData\Rohde-Schwarz\NRP-Toolkit-SDK\`

The examples especially for the R&S NRQ6 are under:

`C:\ProgramData\Rohde-Schwarz\NRP-Toolkit-SDK\NRQ\`

Download the latest version of the R&S NRP Toolkit at:

https://www.rohde-schwarz.com/software/nrp_s_sn/

6 Learn More about the R&S SGT100A

To learn more about the R&S SGT100A, download its manual from:

https://www.rohde-schwarz.com/manual/sgt100a/

A comprehensive discussion of the high-speed remote control capabilities of the R&S SGT100A is available at:

https://www.rohde-schwarz.com/applications/high-speed-remote-control-of-r-s-sgma-rf-sources-application-note_56280-241731.html