Application Note

GENERATION OF RADAR SIGNALS IN A HARDWARE IN THE LOOP (HIL) ENVIRONMENT

Products:

► R&S®SMW200A

Sebastian Kehl-Waas | 1GP124 | Version 1e | 12.2021

https://rohde-schwarz.com/appnote/1gp124

ROHDE&SCHWARZ

Make ideas real

Contents

1 Overview

This application note belongs to a series of application notes which explain how to test EW receivers at RF in the lab with commercial, off-the-shelf signal generators and software. The series will cover all relevant use cases. This application note will discuss the generation of radar signals in a hardware in the loop (HIL) environment with PDW streaming. Further application notes will address threat simulation and verification [1] and configuration of multi-channel setups for simulation of angle of arrival [2].

Development of modern EW systems is a complex and expensive process, that requires thorough testing against all requirements during all phases of development. To keep cost under control, system level testing in the lab is key and brings several advantages: test cases can be reproduced under the exact same conditions as often as necessary.

System level testing in the lab is often performed in a hardware-in-the-loop (HIL) environment. Thereby, the DUTs output is evaluated and influences its input signal. The new input signal into the DUT needs to be calculated in realtime. In EW domain, a typical test HIL setup consists of the DUT with an RF input port and some output interface, an RF signal source and a software simulator or simulation engine. The simulation engine calculates the new signals for the RF source based on the DUT's current output signal. It often uses the pulse descriptor word (PDW) format . After calculation the simulation engine streams the calculated pulse descriptor words to an RF source, e.g. over LAN. The RF source then generates the updated radar signal from the PDWs as required. The DUT, the RF source and the simulation engine are working in a loop with ideally very low latency.

Besides HIL applications, streaming PDWs into an RF Source is also important for long EW simulations lasting up to several hours. In this case, it is often not possible or practicable to pre-compute the complete scenario and hence, a streaming approach is preferred. Whereas, this application note puts the focus on HIL applications, the principles and examples also apply to long-term EW simulations.

The capabilities of modern test equipment such as wideband vector signal generators are perfect for that application and allow to bring most realistic scenarios into a well-controlled environment.

Both for generation of ultra-long scenarios and HIL applications, the R&S®SMW200A vector signal generator is the perfect choice. The R&S®SMW200A receives the streamed PDWs from the HIL scenario simulator via LAN. The powerful baseband hardware interprets the PDWs and generates the RF signal based on the pulse descriptions at the time defined in the PDWs. The R&S®SMW200A can execute and generate pulses from up to six parallel PDW streams with a maximum execution rate of up to 2 Mpulse/s or 2 MPDW/s per stream. It takes on the role of an agile signal source that generates demanding EW environments within its baseband bandwidth of 2 GHz and with a RF frequency of up to 44 GHz. It generates I/Q modulated pulsed signals, agile signals with fast switching and classical pulsed signals from streamed PDWs.

This application note provides some insight into hardware-in-the-loop testing with the R&S®SMW200A. In chapter 1, an introduction to HIL testing and realtime operation of the R&S®SMW200A is given. The hardware and software interface, the PDW format and synchronization and timing mechanisms are described in chapter 2. Chapter 3 provides various example scenarios and detailed information about intermediate calculations. Some additional hints are given in chapter 4. In chapter 5, system requirements for advanced PDW streaming with multiple emitters on multiple parallel streams are discussed. The application note closes with some notes on interesting literature and ordering information.

1.1 Introductory Note

The abbreviation "SMW" is used in this application note for the Rohde & Schwarz product R&S®SMW200A.

The SMW is a general-purpose vector signal generator with outstanding RF performance. It is capable of generating signals for all main communication, radio and avionic standards or to simulate radar systems.

1.2 Hardware-in-the-Loop (HIL)

Hardware in the loop is a test method where a device or system under test (DUT or SUT) is embedded into a simulator system that emulates the real environment of the device or system, mostly in realtime.

An electronic control unit (ECU), e.g. the flight management system of an aircraft, is a typical DUT. The DUT is connected to the simulator system (termed "HIL simulator") via its in- and outputs. In the HIL simulator, the virtual environment is computed in realtime using mathematical models. These signals from virtual sensors or other virtual system components are applied to the inputs of the DUT. The output signals of the DUT are fed to virtual actuators.

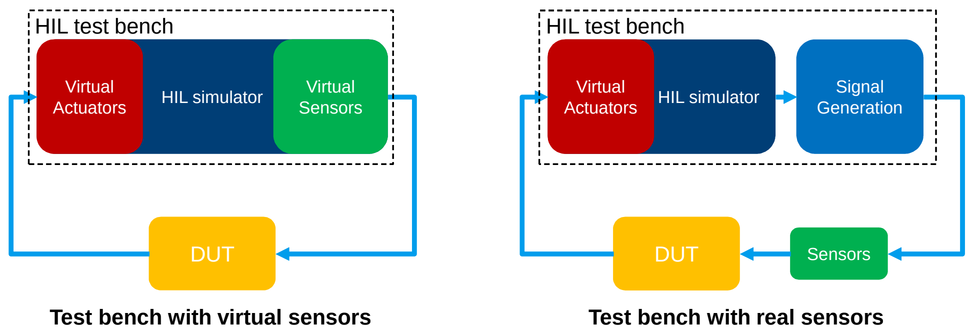

Figure 1: Block diagrams for HIL test benches with virtual and real sensors

If a larger part of a system or even a complete system is to be evaluated, the real sensors and actuators are used.

A typical customer HIL test bench for tests with radar signals (including DUT) is illustrated below. The HIL simulator calculates the (RF) environment for the DUT based on mathematical models. A description of the calculated signals is provided to the radar simulator in form of PDWs. The radar simulator generates a RF signal based on the PDWs and provides it to the DUT. The DUT reacts to the RF input signal in form of control commands, which are in turn fed back to the virtual actuators. The virtual actuators are part of the mathematical models in the HIL simulator. This closes the loop.

The application shown in Figure 2 usually is a multi-channel application, i.e. the DUT has more than one RF input. For the sake of simplicity only one RF path is shown.

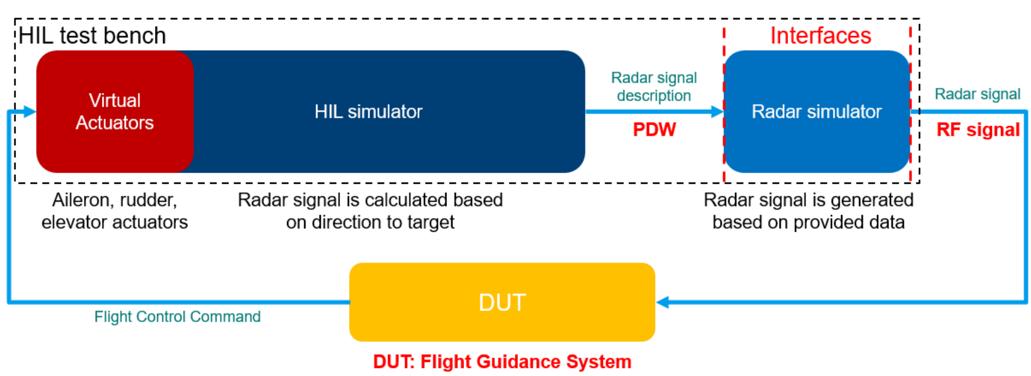

Figure 2: HIL test bench example for testing a flight guidance system

With the setup above it is possible to perform a very detailed emulation of the real DUT environment (shown below):

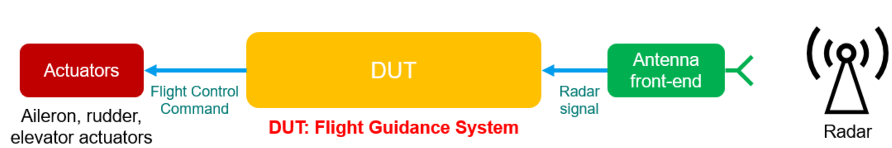

Figure 3: Real DUT environment

1.3 Open-loop Applications

As mentioned in the introduction, the PDW streaming approach is also used for the simulation of very long scenarios that are impracticable to be pre-calculated. In such test-beds the feedback path from the DUT might be missing. An example is illustrated in the figure below.

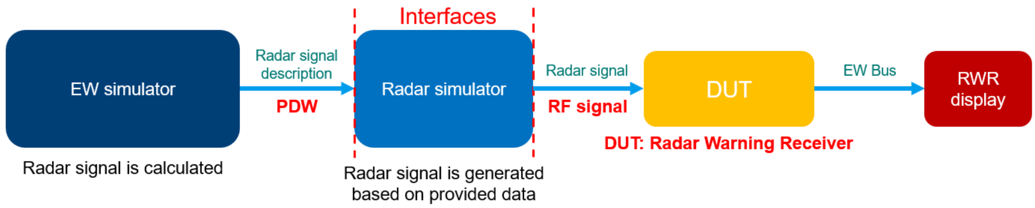

Figure 4: Open-loop testing of a radar warning receiver

The EW simulator provides the radar signal description to the radar simulator in form of PDWs. The radar simulator generates a RF signal based on the PDWs and provides it to the DUT, here a radar warning receiver (RWR). On the RWR display, it can be monitored if the RWR identifies all simulated threats correctly.

1.4 Realtime control of R&S®SMW200A

The R&S®SMW200A can be controlled in realtime using pulse descriptor words (PDWs). A PDW is, as the name suggests, a description of a pulse. The big advantage of the PDW format is its low memory size. Instead of I/Q samples a pulse can be completely characterized by the parameters in the figure below. Also, no I/Q samples are needed to fill pulse pauses. Thanks to their low memory need, PDWs can be calculated and streamed to the SMW in realtime and with low latency. This enables simulation of long-term scenarios and closed-loop testing.

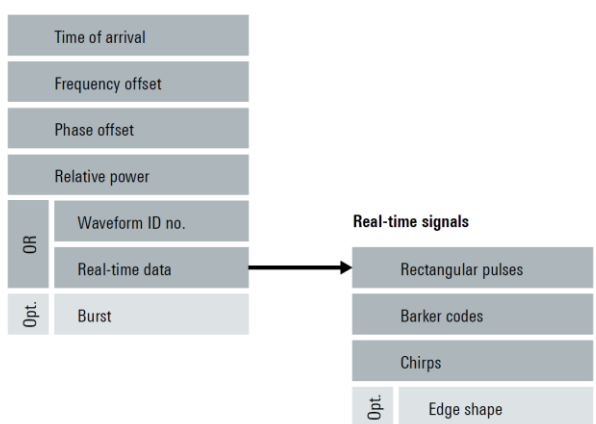

Figure 5: R&S®PDW structure

Figure 5 shows, which information is packed in a PDW. The 'Time of arrival' indicates the point in time, at which the PDW shall be played in the generator. It is given in numbers of clock cycles relative to the trigger of the Extended Sequencer firmware application. Frequency, phase and power offset provide a relative offset to the RF frequency, the phase and the power level. The PDW can contain the address of a pre-calculated waveform segment that is stored in the memory of the signal generator or data describing a pulse, which is generated in realtime. This can be an unmodulated pulse or also an I/Q modulated pulse. By default, the SMW generates a rectangular pulse based on the provided information. If an optional edge shape is specified in the PDW it is also possible to generate pulses with linear or raised cosine edges without the need to calculate an I/Q segment. Instead of a single pulse also a pulse burst consisting of a defined number of equal pulses can be defined with a PDW [3]. This allows to reduce network traffic.

The processing chain from sending the PDW to the modulated RF signal is illustrated below. The PDW is provided to the SMW via LAN. There it is loaded to the RAM and played at the specified time-of-arrival (TOA) that is specified in each PDW. Depending on the PDW, either the signal is calculated in realtime or a pre-calculated ARB file is addressed. The frequency, phase and power offsets are applied to the digital baseband signal, which is D/A converted and modulated onto the RF carrier. The frequency, phase and power offsets are relative to the RF frequency and level configured in the SMW (usage of offsets becomes clearer with the examples provided in chapter 3). The analog signal is provided at the RF output.

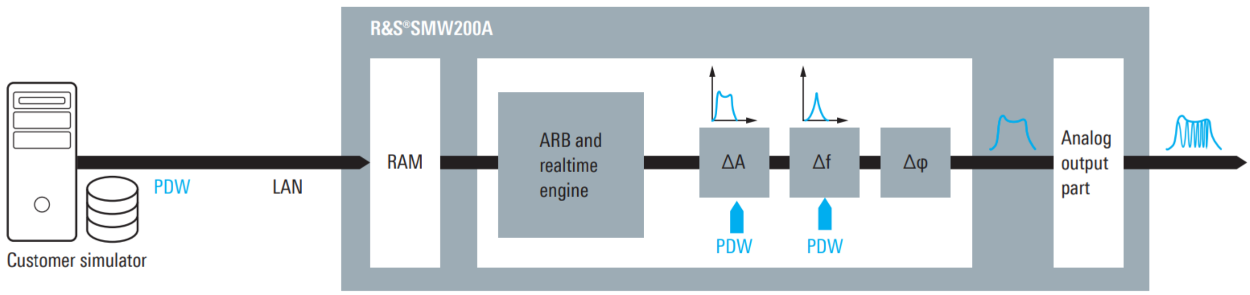

Figure 6: Concept of PDW streaming with the SMW200A

1.5 Required Options for Realtime Operation

Instrument options for realtime radar signal generation		
Option	Name	Remark
Minimum configuration		
SMW200A	Vector signal Generator	
SMW-B10xx	Frequency range 100 kHz to xx GHz for RF path A	
SMW-B13XT	Wideband baseband main module, 2 I/Q paths to RF	
SMW-B9	Wideband baseband generator, 500 MHz, 256 MS	Can be installed twice to add second PDW stream
SMW-K502	Wideband Ext. Sequencing for R&S®SMW-B9	Prerequisite for PDW streaming
SMW-K503	Realtime Control Interface	Enables realtime control interface, up to 1MPDW/s per PDW stream
Optional		
SMW-B20xx	Frequency range 100 kHz to xx GHz for RF path B	Add second RF path
SMW-B15	Fading simulator and signal processor	Add two or four additional PDW streams (requires SMW-K315)
SMW-K315	Pulse-on-Pulse simulation	Allows up to 6 individual PDW streams
SMW-K504	High speed upgrade of PDW rate	Up to 2MPDW/s per PDW stream

2 Instrument Control

2.1 Hardware Interface

The R&S®SMW200A can be equipped with up to two wideband baseband generators (SMW-B9), also called coder boards, and two or four wideband fading simulators (SMW-B15), also called fader boards. Each coder- or fader-board provides a dedicated 1Gbit/s LAN interface (RJ45) for PDW streaming (see Figure 7).

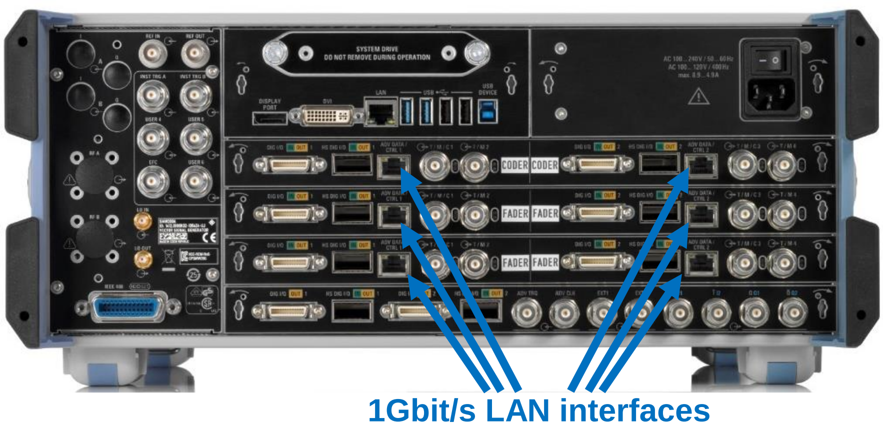

Figure 7: 1Gbit/s LAN interfaces on the SMW200A

Based on PDWs, each board generates a baseband signal with a modulation bandwidth of up to 2 GHz. With six boards a maximum of six independent baseband signals is generated. These signals can be distributed among a maximum of two independent RF paths per instrument: Either all signals are summed up and routed to a single output (common RF frequency for all baseband signals) or they are distributed equally (two different RF frequencies). This is illustrated in Figure 8.

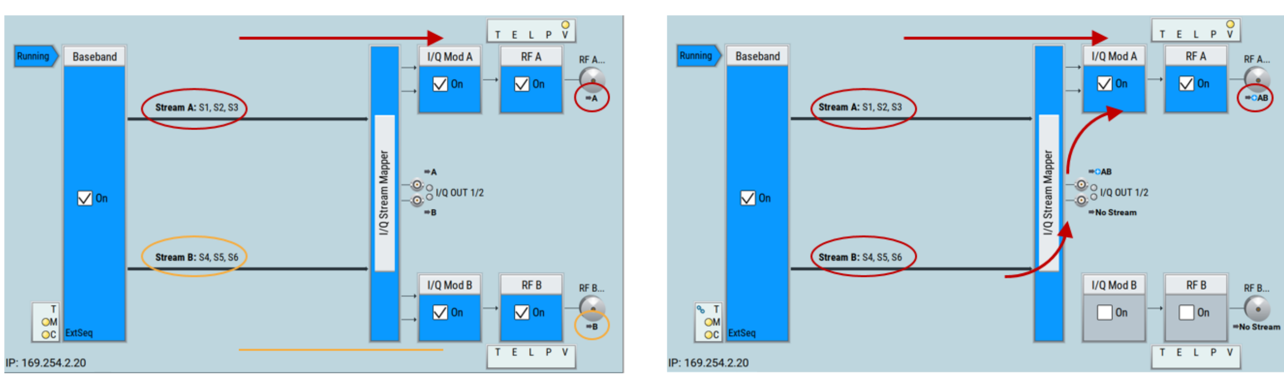

Figure 8: Possible routings of six independent PDW streams

2.2 Extended Sequencer

The Extended Sequencer (R&S®SMW-K501/-K502/-K503/-K504) is a firmware application that allows the SMW to generate complex signal sequences in realtime.

PDWs are streamed via an external LAN interface to control the signal generation. Each PDW can either refer to one of the waveform segments that have been loaded into the Extended Sequencer application in advance, or start generation of various available realtime pulses (among which are unmodulated pulses, chirps and Barker coded pulses with different pulse shapes [3]).

The SMW provides different system configuration modes. For PDW streaming, one can either use the ‘Standard’ mode or the ‘Extended Sequencer Advanced’ mode. In ‘Standard’ mode only two PDW streams can be configured. To use the maximum of six independent PDW streams, the ‘Extended Sequencer Advanced’ mode has to be selected (see section 5.3. for details).

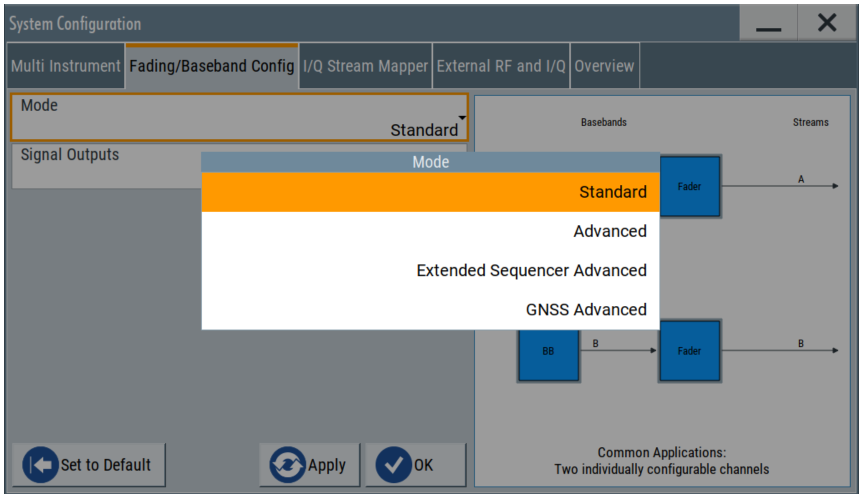

Figure 9. System configuration dialog in the SMW200A

The ‘Extended Sequencer’ provides different modes. In the ‘Pulse Sequencer’ or ‘Direction Finding’ mode the SMW can replay scenarios that are precalculated with the R&S®Pulse Sequencer software. This is normally used together with the R&S®Pulse Sequencer Software and automatically configured. To prepare the R&S SMW200A for PDW reception, the ‘Extended Sequencer’ Mode has to be set to ‘Real Time Control Interface’.

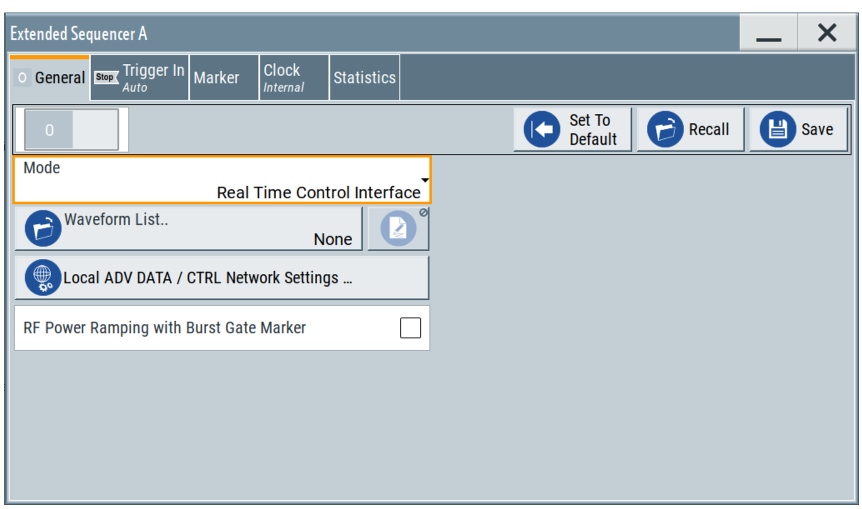

Figure 10: Extended Sequencer configuration dialog in the SMW200A (requires SMW-K503)

The network settings of the dedicated 1Gbit/s LAN interfaces can be configured by clicking on ‘Local ADV DATA/CTRL Network Settings …’. The IP address can either be assigned automatically using DHCP or a static IP address can be set.

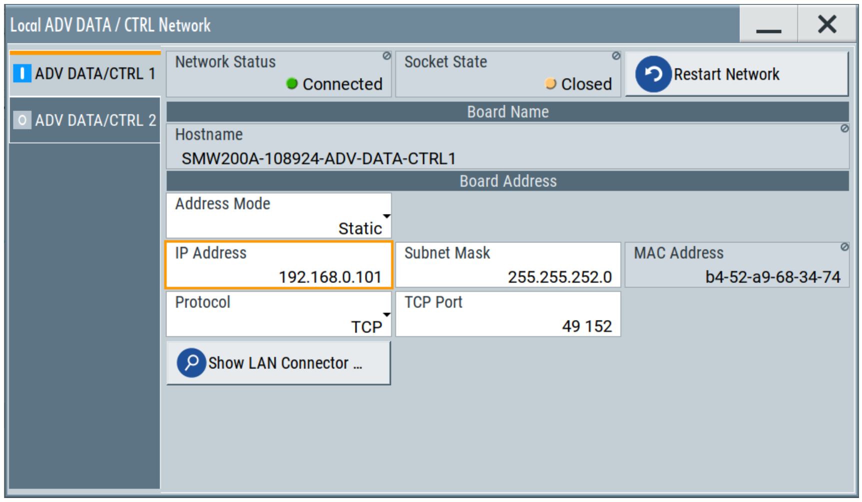

Figure 11: PDW interface configuration dialog in the SMW200A

Once the realtime operation is activated and an IP address is assigned, it remains to switch on the ‘Extended Sequencer’ and the SMW is ready for operation with PDWs.

To use pre-calculated ARB segments, the segments have to be loaded into the ARB memory with the help of a waveform list. By clicking on ‘Waveform List…’, one can either load an existing waveform list, create a new list or edit an existing list.

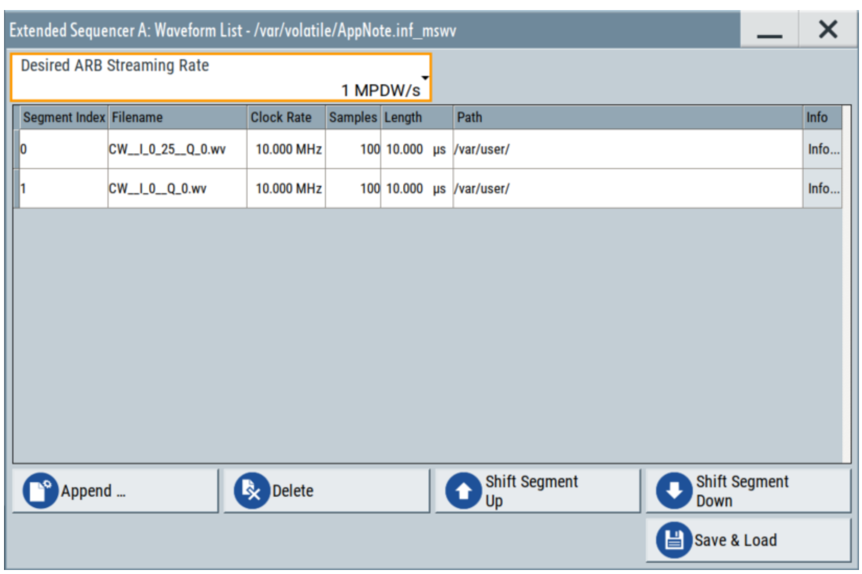

Figure 12: Waveform list configuration dialog in the SMW200A

All segments specified in the waveform list are loaded to the ARB memory. Each segment can be addressed by the respective segment index in the first column. More information on the handling of waveform lists is given in [4].

2.3 R&S Descriptor Word Structure

2.3.1 General Descriptor Word Format Specification

The SMW provides a dedicated interface to receive and process R&S descriptor words. R&S Pulse Descriptor Words (PDW) can be used to generate pulsed signals in real-time or replay pre-calculated waveform segments. R&S Timed Control Descriptor Words (TCDW) can be used to change instrument RF frequency and/or level or re-arm the Extended Sequencer.

Descriptor Word Type	Purpose
PDW	Generate pulsed signals in real-time. Replay pre-calculated waveform segments.
TCDW	Control RF parameters. Re-Arm Extended Sequencer.

Descriptor words are transmitted as sequence of bytes. For all descriptor words, their type is determined by flags in the header and flags section. The descriptor word size and content depend on the type.

2.3.2 Pulse Descriptor Word (PDW)

Each pulse descriptor word consists of header, flags, body and payload. Depending on the operating mode (basic/expert) and whether a signal is generated in real-time based on the signal description or a pre-calculated waveform segment, also called ARB segment, is replayed, the content of each part is different. The expert pulse descriptor word format additionally can contain a parameter section or extensions. A mix of formats during a running simulation is not possible. The selection of the desired format can be done locally via the SMW GUI or remotely (SCPI commands).

2.3.2.1 Header

The PDW header section contains the time of arrival (TOA) and flags which define the content of the PDW.

PDW header			**48 Bit**	**56 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
TOA	unsigned int	Timestamp relative to scenario start trigger event TOA = (timestamp in seconds) * 2.4e9	44 Bit	52 Bit
SEG	Boolean	Flag that indicates if the PDW is used to address pre-calculated waveform segments or the PDW is used to generate pulsed signals in real-time without a waveform segment 0 = ARB segment 1 = real-time signal	1 Bit	1 Bit
USE_EXTENSION	Boolean	0 = PDW extension block is not used 1 = PDW extension block is used	-	1 Bit
PARAMS	unsigned int	Additional parameters 0 = No params 1 = Use basic edge shaping 2 = RSVD 3 = RSVD	-	2 Bit
RSVD	-	Reserved for future use	3 Bit	-

2.3.2.2 Flags

The PDW flags section contains information about the xDW type, phase mode and marker settings.

PDW flags			**8 Bit**	**8 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
CTRL	Boolean	Indicates whether the descriptor word is a PDW or a TCDW 0 = PDW 1 = TCDW	1 Bit	1 Bit
RSVD	-	Reserved for future use	1 Bit	1 Bit
PHASE_MOD	Boolean	Indicates phase mode 0 = value in PHS field inside PDW body is absolute 1 = value in PHS field inside PDW body is relative to the phase value of the last sample of the previous signal	1 Bit	1 Bit
IGNORE_PDW	Boolean	PDW is ignored (no signal output)	1 Bit	1 Bit
M4	Boolean	Reserved for future use	1 Bit	1 Bit
M3	Boolean	Set Marker 3	1 Bit	1 Bit
M2	Boolean	Set Marker 2	1 Bit	1 Bit
M1	Boolean	Set Marker 1	1 Bit	1 Bit

2.3.2.3 Body

The PDW body section contains offset values for frequency, level and phase relative to the instrument RF settings.

PDW Body			**64 Bit**	**64 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
FREQ_OFFSET	int	Frequency offset added to instrument RF frequency. -1 GHz <= frequency_offset <= 1 GHz FREQ_OFFSET= (frequency_offset / 2.4e9) * 2^{32}	32 Bit	32 Bit
LEVEL_OFFSET	unsigned int	Level offset subtracted from instrument RF level. level_offset >= 0 dB LEVEL_OFFSET = $10^{(-\text{level\_offset} / 20)}$ * (2^{15} - 1)	16 Bit	16 Bit
PHASE_OFFSET	unsigned int	Phase offset 0° <= phase_offset < 360° PHASE_OFFSET = phase_offset/360° * 2^{16}	16 Bit	16 Bit

2.3.2.4 Params (only in expert mode)

Depending on the PARAMS bits in the PDW header, this 4 Byte block can be used for basic edge shaping. If USE_EXTENSION is set (USE_EXTENSION = 1), this block has to be omitted completely.

PDW params [PARAMS = 0]			**32 Bit**
Parameter	**Data type**	**Description**	**Size**
STUFFING	-	Params block unused. Set to 0.	32 Bit

PDW params [PARAMS = 1] (only supported for real-time signals: SEG=0)			32 Bit
Parameter	**Data type**	**Description**	**Size**
EDGE_TYPE	unsigned int	Describes rising/falling edge type 0 = Linear 1 = Cosine	3 Bit
MULTIPLIER	Boolean	Multiplier of Rise/Fall time to increase setting range. 0 = x1 1 = x8	1 Bit
RSVD	-	Reserved for future use	6 Bit
RISE_FALL_TIME	unsigned int	Rise/Fall time (first sample to last sample; is added to pulse width) RISE_FALL_TIME = ((Rise/Fall time in seconds) / Multiplier) * 2.4e9	22 Bit

2.3.2.5 Payload

For waveform playback the SEG flag in the PDW HEADER has to be set to ‘1’.

ARB Segment [SEG = 1]			136 Bit	96 Bit
Parameter	**Data type**	**Description**	**Basic**	**Expert**
SEGMENT_IDX	unsigned int	Index of the pre-calculated waveform, which was loaded into the SMW memory in advance	24 Bit	24 Bit
RSVD	-	Reserved for future use	112 Bit	72 Bit

For a real-time signal the SEG flag in the PDW HEADER has to be set to '0'.

Real-Time Signal Unmod [SEG = 0 and MOD=0]			136 Bit	96 Bit
Parameter	**Data type**	**Description**	**Basic**	**Expert**
MOD	unsigned int	Type of modulation 0 = Rectangular pulse	4 Bit	4 Bit
TON	unsigned int	Time on = Pulse width (first sample to last sample) TON = (Time on in seconds) * 2.4e9	44 Bit	44 Bit
RSVD	-	Reserved for future use	88 Bit	48 Bit

Real-Time Signal Linear Chirp [SEG = 0 and MOD=1]			136 Bit	96 Bit
Parameter	**Data type**	**Description**	**Basic**	**Expert**
MOD	unsigned int	Type of modulation 1 = Linear Chirp	4 Bit	4 Bit
RSVD	-	Reserved for future use	19 Bit	3 Bit
TON	unsigned int	Time on = Pulse width (first sample to last sample) TON = (Time on in seconds) * 2.4e9	25 Bit	25 Bit
FREQ_INC	int	Frequency increment in Hz/Sample freq_step = bandwidth/ (N_samples – 1) When using edge shaping, N_samples includes rising and falling edges.	64 Bit	64 Bit

		Without edge shaping, N_samples = TON. FREQ_INC = (freq_step/2.4e9) * 2^{64}		
RSVD	-	Reserved for future use	24 Bit	-

Real-Time Signal Triangular Chirp [SEG = 0 and MOD=2]			**136 Bit**	**96 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
MOD	unsigned int	Type of modulation 2 = Triangular Chirp	4 Bit	4 Bit
RSVD	-	Reserved for future use	19 Bit	3 Bit
TON	unsigned int	Time on = Pulse width (first sample to last sample) TON = (Time on in seconds) * 2.4e9	25 Bit	25 Bit
FREQ_INC	int	Frequency increment in Hz/Sample freq_step = bandwidth/ (N_samples - 1) When using edge shaping, N_samples includes rising and falling edges. Without edge shaping, N_samples = TON. FREQ_INC = (freq_step/2.4e9) * 2^{64}	64 Bit	64 Bit
RSVD	-	Reserved for future use	24 Bit	-

Real-Time Signal Barker [SEG = 0 and MOD=3]			**136 Bit**	**96 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
MOD	unsigned int	Type of modulation 3 = Barker code	4 Bit	16 Bit
CHIP_WIDTH	unsigned int	Chip width of one Barker code chip CHIP_WIDTH = (chip_width in seconds) * 2.4e9 The minimum supported chip width of a barker coded pulse is currently limited to 3.75 ns (CHIP_WIDTH = 9). The total pulse width is determined by the Barker code length times the chip width.	44 Bit	44 Bit
CODE	unsigned int	Code to select the type of Barker code **CODE** / **Length** / **Sequence** **0** / 2 / +- **1** / 2 / ++ **2** / 3 / ++- **3** / 4 / +-++ **4** / 4 / +--- **5** / 5 / +++-+ **6** / 7 / +++--+- **7** / 11 / +++---+--+- **8** / 13 / +++++--++-+-+	4 Bit	4 Bit
RSVD	-	Reserved for future use	4 Bit	4 Bit
STUFFING	-	Stuffing bits	16 Bit	16 Bit
RSVD	-	Reserved for future use	64 Bit	24 Bit

2.3.2.6 Extension (only in expert mode)

The 20 Byte extension is evaluated if the USE_EXTENSION bit in the header is set to 1. This Block has three extension fields which can be used by setting the corresponding extension flags.

2.3.2.6.1 Extension Flags

The extension flags section allows the user to define the type of extension. Up to three extension fields can be used.

PDW extension flags			**16 Bit**
Parameter	**Data type**	**Description**	**Size**
FIELD_1_TYPE	unsigned int	Describes type of field 1 0 = Unused 1 = Edge field 2 = Burst field 3-7 = RSVD	3 Bit
FIELD_2_TYPE	unsigned int	Describes type of field 2 0 = Unused 1 = Edge field 2 = Burst field 3-7 = RSVD	3 Bit
FIELD_3_TYPE	unsigned int	Describes type of field 3 0 = Unused 1 = Edge field 2 = Burst field 3-7 = RSVD	3 Bit
RSVD	-	Reserved for future use	7 Bit

2.3.2.6.2 Extension Fields

Unused Field [FIELD_x_TYPE = 0]			**48 Bit**
Parameter	**Data type**	**Description**	**Size**
STUFFING	-	Fill with 0	48 Bit

Edge Field [FIELD_x_TYPE = 1] (only supported for real-time signals: SEG=0)			**48 Bit**
Parameter	**Data type**	**Description**	**Size**
EDGE_TYPE	unsigned int	Describes rising/falling edge type 0 = Linear 1 = Cosine	3 Bit
MULTIPLIER	Boolean	Multiplier of rise/fall time to increase setting range. 0 = x1 1 = x8	1 Bit
RISE_TIME	unsigned int	Rise time (first sample to last sample; is added to pulse width) RISE_TIME = ((Rise time in seconds) / Multiplier) *2.4e9	22 Bit
FALL_TIME	unsigned int	Fall time (first sample to last sample; is added to pulse width) FALL_TIME = ((Fall time in seconds) / Multiplier) *2.4e9	22 Bit

Burst Field [FIELD_x_TYPE = 2]			48 Bit
Parameter	**Data type**	**Description**	**Size**
BURST_PRI	unsigned int	Pulse repetition interval (PRI) BURST_PRI = (PRI in seconds) * 2.4e9	32 Bit
BURST_ADD_PULSES	unsigned int	Number of repetitions in addition to the initial signal (real-time signal or ARB segment)	16 Bit

2.3.3 Timed Control Descriptor Word (TCDW)

By setting the CTRL flag in the xDW flags section, the user can issue commands such as changing the instrument RF frequency and/or amplitude of the signal generator directly from the descriptor word stream or arm the Extended Sequencer, where otherwise a SCPI command would have been necessary.

By embedding the control commands directly in the descriptor word stream, the start of the frequency or amplitude change procedure can be exactly determined by the TOA.

2.3.3.1 Header

The TCDW header section contains the TOA and flags which define the command type.

TCDW header			48 Bit	56 Bit
Parameter	**Data type**	**Description**	**Basic**	**Expert**
TOA	unsigned int	Timestamp relative to scenario start trigger event	44 Bit	52 Bit
PATH	Boolean	Specifies RF path which is affected by TCDW 0 = Path A 1 = Path B	1 Bit	1 Bit
CMD	unsigned int	Specifies command type 0 = Frequency change 1 = Amplitude change 2 = Frequency and amplitude change 3 = Arm Extended Sequencer in Trigger Mode 'Armed Auto' (Stop internal counter and set to zero, xDWs in buffer are still available)	3 Bit	3 Bit

2.3.3.2 Flags

The TCDW flags section contains information about the xDW type.

TCDW flags			16 Bit	8 Bit
Parameter	**Data type**	**Description**	**Basic**	**Expert**
CTRL	Boolean	Indicates whether descriptor word is a PDW or a TCDW 0 = PDW 1 = TCDW	1 Bit	1 Bit
RSVD	-	Reserved for future use	15 Bit	7 Bit

2.3.3.3 Body

The TCDW body section contains values for instrument RF frequency and level.

<table>
<tr><td colspan="3">TCDW Body</td><td>64 Bit</td><td>64 Bit</td></tr>
<tr><td>Parameter</td><td>Data type</td><td>Description</td><td>Basic</td><td>Expert</td></tr>
<tr><td>FVAL</td><td>unsigned int</td><td>RF frequency setting of signal generator in Hz</td><td>40 Bit</td><td>40 Bit</td></tr>
<tr><td>LVAL</td><td>signed fixed point BCD</td><td>RF level setting of signal generator in dBm
<table>
<tr><td>Bit</td><td>0</td><td>1-7</td><td>8-11</td><td>12-15</td><td>16-23</td></tr>
<tr><td>Value</td><td>Sign
0=pos
1=neg</td><td>Integer part</td><td>Tenth part</td><td>Hundredth part</td><td>Unused (Set to 0)</td></tr>
</table></td><td>24 Bit</td><td>24 Bit</td></tr>
</table>

2.4 Timing and Synchronization

2.4.1 Timing and Processing of PDWs

Each PDW has a TOA field that determines at what time the PDW shall be played back. PDWs that are received by the SMW are stored in a FIFO (first-in-first-out) buffer. The SMW has an internal counter for each PDW stream that is increased by 1 at a clock rate of 2.4 GHz (resolution of $1/(2.4\cdot 10^9)$ s). This counter serves as clock for PDW processing. The TOA parameter of a PDW is given in clock ticks.

In each clock cycle the current counter state t is compared to the TOA of the first PDW in the FIFO buffer. The following applies:

- ► If the number of clock ticks on the counter is equal to the TOA parameter of the next PDW in the FIFO buffer, the PDW is processed. Example: t =10 and TOA = 10
- ► If the counter state is smaller than the TOA of the next PDW, nothing happens. Example: t =10 and TOA = 140
- ► If the counter state is larger than the TOA of the next PDW, the PDW is ignored and removed from the buffer. This can happen, when a PDW arrives at the SMW too late, e.g. due to increased network latency. Example: t =10 and TOA = 5.

In order to processed at its dedicated TOA, a PDW has to arrive at least 100us in advance at the SMW ADV DATA / CTRL interface. After the PDW was processed in the FPGA, there is a small additional delay τ_{RF} in the order of a few µs. The delay τ_{RF} is different in each individual RF path. However, it is deterministic and can therefore be compensated.

The following timing diagram provides more details.

Figure 13: Timing diagram for PDW processing in the SMW200A

If a PDW shall be played back at TOA = 0, it can be sent to the SMW before the Extended Sequencer is triggered. The minimum processing time still applies, i.e. it has to be sent at least 100µs before the trigger.

In order to provide PDWs sufficiently in advance and at the same time optimize the update latency it is mandatory to know the value of the internal counter. Theoretically, the current counter value can be queried with a SCPI-command. In practice, this method is way too inaccurate due to the round-trip-time of SCPI command and response. Therefore, the HIL scenario simulator and the SMW have to be synchronized. A possible synchronization concept is described in the next section.

2.4.2 Synchronization of HIL simulator and R&S®SMW200A

For accurate synchronization we require

► a common trigger event, e.g. a 1 PPS signal and

► a common frequency reference, e.g. a 10 MHz reference signal[1]

for HIL simulator and SMW. A commercial-of-the-shelf (COTS) GPS disciplined clock provides both signals synchronized to a worldwide available clock signal (GPS time).

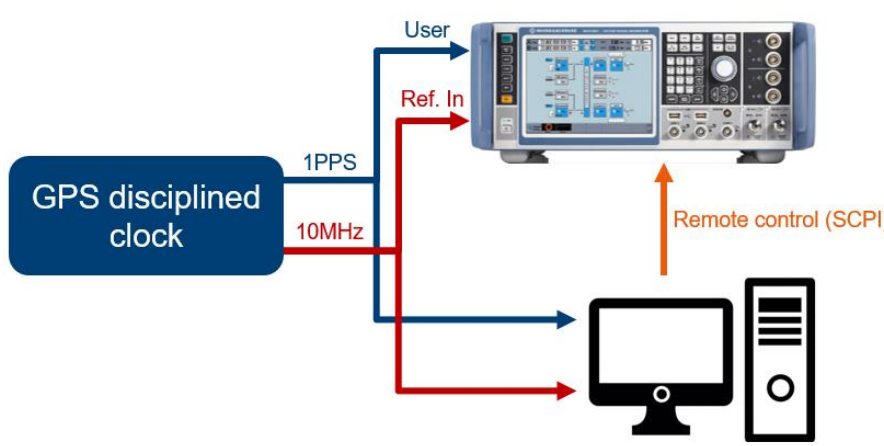

Figure 14: GPS based synchronization of HIL simulator and SMW200A

The 10MHz reference signal and the 1PPS signal from the GPS disciplined clock are provided to the reference input and a user connector of the SMW and to the HIL simulator. The SMW is remotely controlled by the HIL simulator. Cabling for PDW streaming is omitted in this figure for the sake of clarity.

[1] If the 10 MHz reference signal is missing, HIL simulator and SMW can still be triggered simultaneously with a 1 PPS signal. However, the clocks of HIL simulator and SMW will drift relative to one another.

After we configured the SMW to use the external reference, we set the trigger source of the SMW to an external trigger, e.g. External Global Trigger 1, and assign the signal to the user port that is connected to the 1PPS signal in the ‘Global Connectors…’ settings, e.g. User 1. Next, we select the option ‘Disable External Trigger’ to make sure, that the external trigger is ignored until the moment, we actually want to start our simulation.

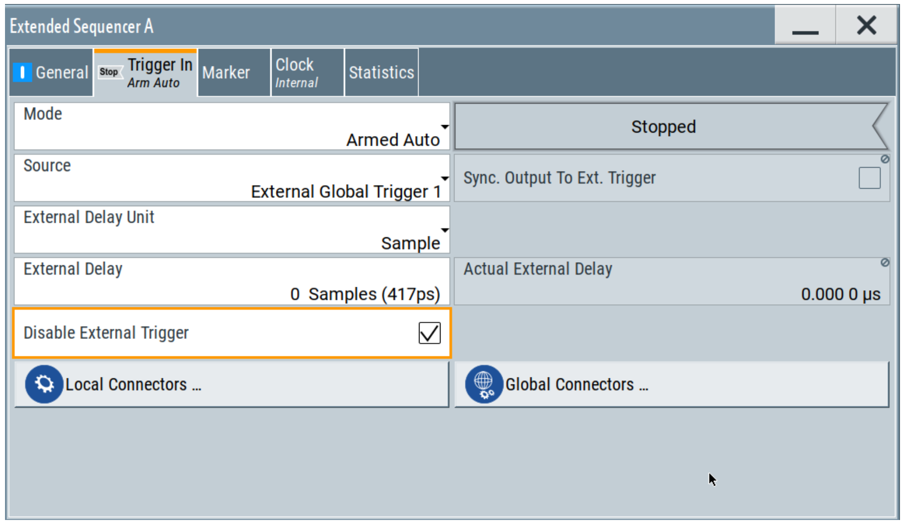

Figure 15: Trigger configuration dialog in the SMW200A

Within the second before the desired trigger pulse of the 1PPS signal, we enable the external trigger remotely with the SCPI command ‘`:Source<hw>:BB:ESEQuencer:TRIGger:EXTernal:DISable 0`’. The simulation will be triggered on the next pulse. The following timing diagram shall illustrate the concept in more detail.

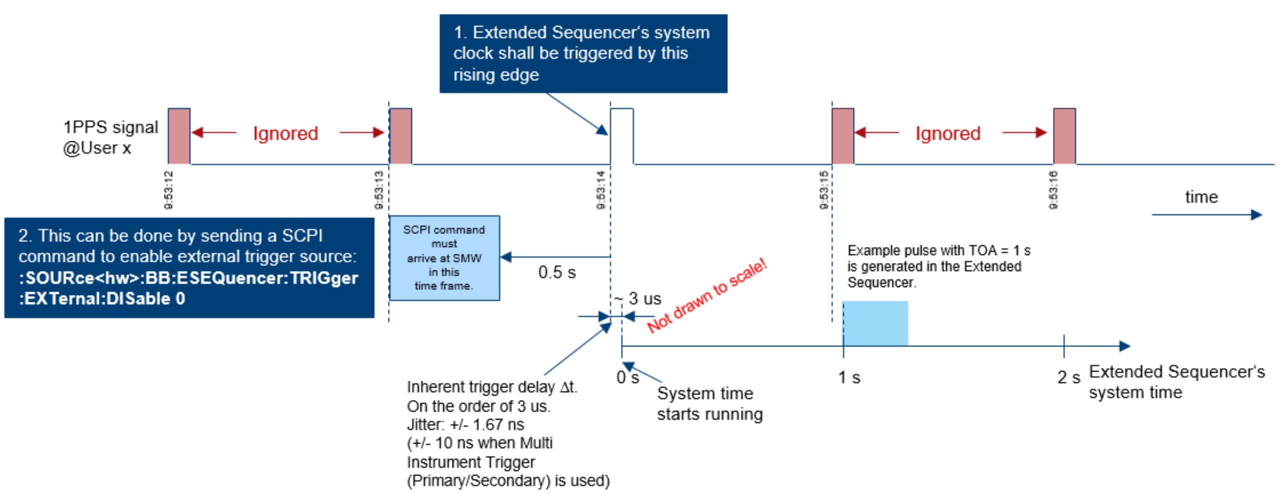

Figure 16: Timing diagram for triggering with a 1PPS signal

On the top, we see the 1PPS signal from our GPS disciplined clock with the respective timestamp at each pulse as it is provided to SMW and HIL simulator. Let’s assume, that we want to trigger our simulation on the pulse issued at 9:53:14. Therefore, we issue the SCPI command to enable the external trigger right after the pulse at 9.53.13. Now, the SMW listens on the user port for an external trigger. With the next pulse, the one at 9:53:14, the Extended Sequencer is triggered. As long as the Extended Sequencer is not stopped, all subsequent 1PPS signals are ignored.

The SMW has an inherent trigger delay of approx. 3 µs with a jitter ±1.67ns (±10ns when the Multi Instrument Trigger is used) (SMW with SMW-B9). The delay (without the jitter) is deterministic and can be considered for

TOA calculation in the HIL scenario simulator. As the values for delay and jitter may change with future FW versions, it is recommended to measure the value and consider the actual value.

In a setup with multiple SMWs, we use the Multi Instrument Trigger (Primary/Secondary mode). The 1PPS signal is connected and the SCPI command is sent to the Primary SMW only.

2.4.3 Contributions to loop latency

A very important parameter in the design of a HIL test bench is the loop latency or loop time, i.e. the time it takes until the input signal to the DUT is updated based on the output of the DUT. The inverse of the loop latency is the update rate of the HIL test bench. In the following, we will have a look on a typical HIL setup and analyze the contributions of all components to loop latency.

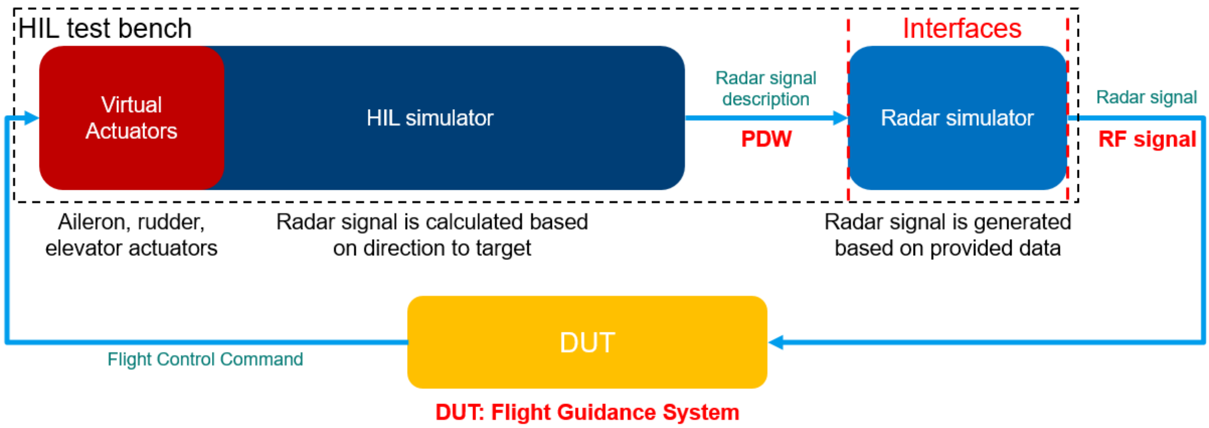

Figure 17: HIL test bench example for testing a flight guidance system

Let's take our initial example, unfold the loop at the output of the DUT and apply a timescale to analyze the timing behavior of the test environment.

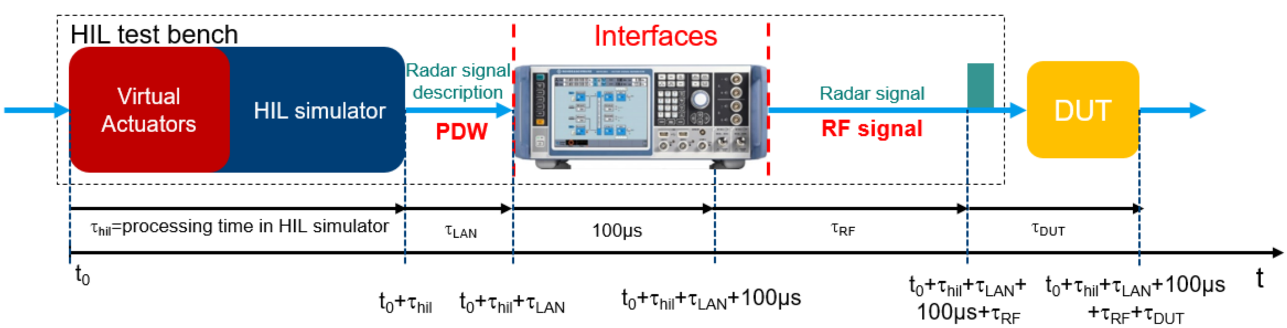

Figure 18: Contributions to loop latency in a HIL setup

The output signal of the DUT is available at t_0. It is processed by the HIL simulator and the corresponding PDWs are generated at $t_0+\tau_{hil}$. The transmission via LAN takes τ_{LAN}. The PDW has to be at the SMW 100µs in advance. After a small additional delay τ_{RF} the signal is available at the output of the SMW200A respectively at the input of the DUT. One loop cycle is completed, when the DUT has processed the new input signal within the interval τ_{DUT}.

The total loop latency is $\tau_{hil}+\tau_{LAN}+100\ \mu s+\tau_{RF}+\tau_{DUT}$ and the max. update rate for the HIL system described above is given by its inverse.

For the TOA parameter one has to consider that the clocks of HIL simulator and SMW differ by the inherent trigger delay Δt (SMW is slow by approx. 3µs). So, the TOA parameter is $t_0+\tau_{hil}+\tau_{LAN}+100\mu s-\Delta t$.

3 Generate PDWs for realistic scenarios

In a HIL system, the feedback of the DUT influences its input signal. For example, a RWR aboard an aircraft commands an evasive maneuver based on a detected threat. The evasive maneuver leads to a changing distance, direction and relative velocity among some other parameters. The new scenario parameters need to be mapped to the available PDW parameters. Apart from the scenario parameters also the emitter configuration must be represented in the PDW. This covers pulse width, antenna pattern, antenna scan, modulation-on-pulse (MOP) and many more.

The following figure gives an overview of selected scenario, emitter and receiver related parameters and the affected PDW parameters in the same colors.

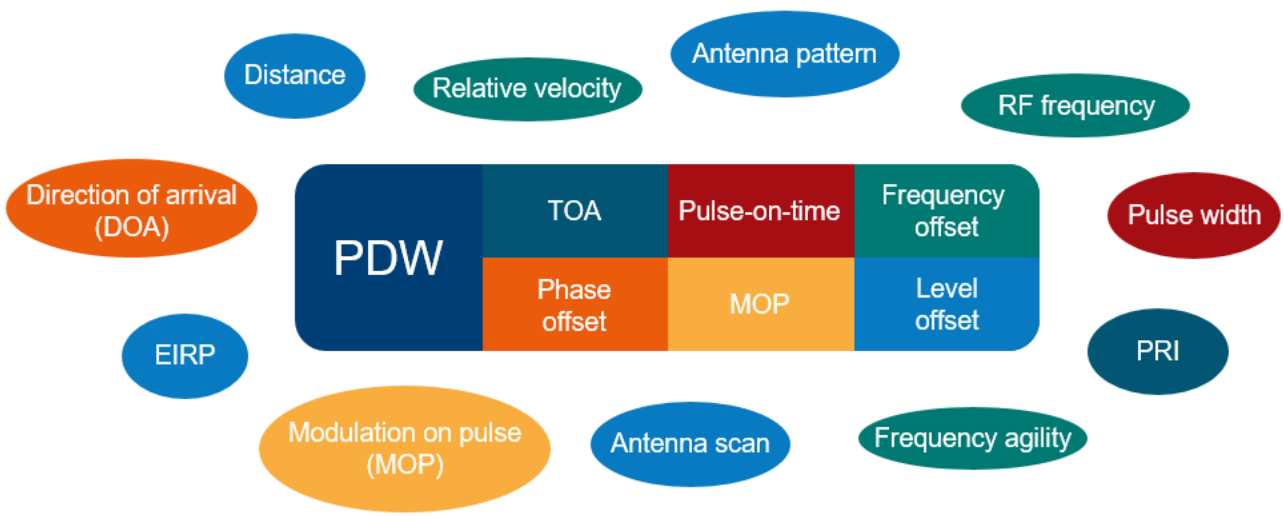

Figure 19: Overview of selected scenario, emitter and receiver parameters and their PDW parameter counterpart

In the following, different scenarios and emitters are considered and the PDW parameter calculation is discussed. As each PDW contains offsets relative to the RF frequency and level selected in the SMW, we will also discuss how to find the correct setting for a scenario and how the settings can be changed during a simulation.

Unless stated otherwise, we use a SMW200A with a single baseband (SMW-B9, modulation bandwidth: 2000MHz) and a single RF path.

In the next chapters, different examples show how real parameters and geometry map into the PDWs. Normally the customer simulators make exactly these calculations before streaming the PDW data to the signal source.

3.1 Static threat / static receiver

In the first scenario, we consider a static threat with a gaussian antenna pattern (HPBW = 2°) and no antenna scan. The antenna beam of the threat points towards the receiver.

The receiver is also static and attached to a single omnidirectional antenna.

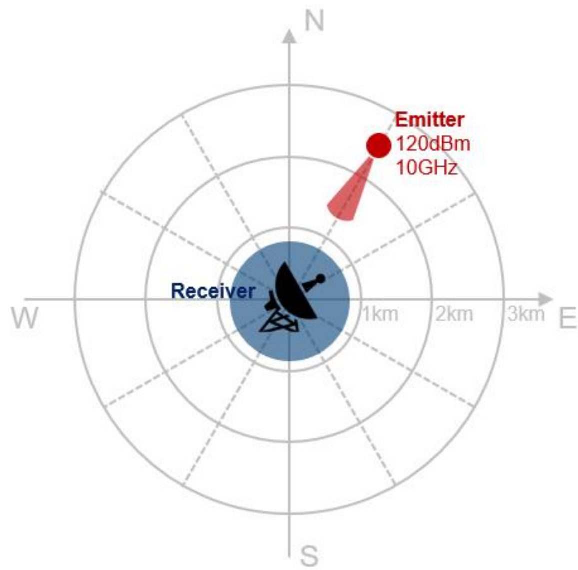

Figure 20: Scenario with static threat and static receiver

The relevant parameters for this scenario are the distance between emitter and receiver, transmit and receive antenna gain and EIRP of the transmitter.

Let's assume the following parameters:

- ► Distance R = 2.5 km
- ► EIRP = 120 dBm
- ► RF frequency f = 10 GHz
- ► Pulse width = 10 µs
- ► PRI = 50 µs
- ► No frequency hopping, no MOP
- ► RX antenna gain G_r = 0 dBi

We use the one-way radar equation to calculate the receiver input power P_r:

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi R)^2}\,, \tag{1}$$

where P_t is the transmit power, G_t is the antenna gain at the emitter and λ is the wavelength. EIRP, the product of transmit power P_t and antenna gain G_t, is given. So is the distance R. The wavelength λ can be calculated as

$$\lambda = \frac{c}{f}\,, \tag{2}$$

where c is the speed of light.

With the numbers above, we get a receiver input power of

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi R)^2} = \frac{P_t G_t G_r c^2}{(4\pi f R)^2} = 0.911 \text{ mW} = -0.41 \text{ dBm}\,.$$

In this scenario, only one emitter is simulated. Both, the emitter and the receiver are static and without antenna scan. Therefore, the received signal is constant over time.

With the results above, we set an RF frequency of 10 GHz and an RF level of -0.41 dBm in the SMW. The PDW parameters are:

- Pulse on-time: 10 µs
- Frequency offset: 0 Hz
- Level offset: 0 dB
- Phase offset: 0°
- No MOP

The only thing changing for each PDW is the TOA parameter, whereas the TOA value of the first PDW is determined by the distance between emitter and receiver and the propagation speed of the RF signal. It can be calculated as

$$TOA = \frac{R}{c} = \frac{2500\ \mathrm{m}}{c} = 8.339102\ \mu\mathrm{s}\,.$$

The TOA can be provided with a granularity of 416.67 ps (which is the inverse of the sample rate 2.4 GHz). So, the first PDW is issued after 8.339167 µs. In every succeeding PDW, the TOA parameter is increased by 50 µs (PRI).

3.2 Static, frequency agile threat with circular antenna scan

Next, we want to consider the scenario from 3.1 with two changes:

- The emitter performs a clockwise circular antenna scan with 15 RPM. The scan starts at the 12 o'clock position.
- The emitter performs pulse-to-pulse hopping through frequencies 9.9 GHz → 10 GHz → 10.1 GHz → 9.95 GHz → 10.05 GHz →10.15 GHz

First, let's have a look on the RF settings in the SMW. The level offset specified in the PDW is applied in the digital baseband. As, per default, the dynamic range in the digital domain is fully exploited, only a negative level offset is possible. Therefore, the RF level is set to the maximum power level of the scenario, which corresponds to the situation where the emitter points towards the receiver.

Looking at the one-way radar equation, we see, that the receive power is frequency dependent. The lower the frequency, the higher the receive power. We get the maximum power $P_{r,max}$= -0.32 dBm for f = 9.9 GHz, which we set as RF level in the SMW.

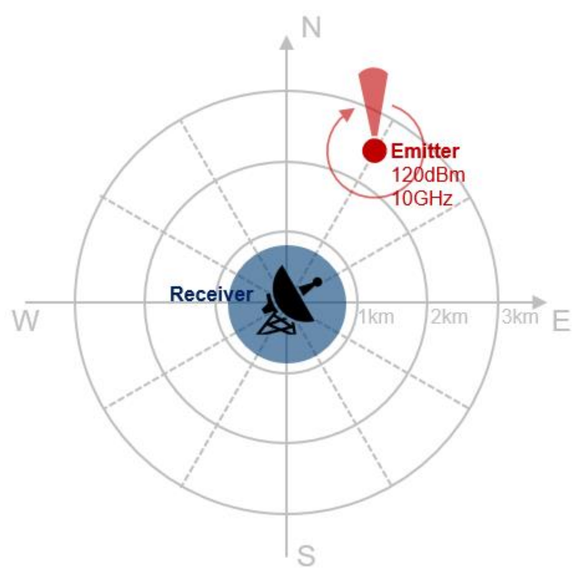

Figure 21: Scenario with static threat with circular antenna scan and static receiver

The RF frequency could be set as (9.9 GHz + 10.15 GHz)/2 = 10.025 GHz. The frequency offsets then are -125 MHz, -25 MHz, 75 MHz, -75 MHz, 25 MHz and 125 MHz. Alternatively, we could also use 10 GHz as RF frequency in the SMW, which gives us frequency offsets of -100 MHz, 0 MHz, 100 MHz, -50 MHz, 50 MHz and 150 MHz. Other choices are possible, as long as the frequency offsets are within the maximum modulation bandwidth of the SMW. Here, we use the second variant.

Next, we take the antenna scan and the pattern into account. In Figure 22, the antenna pattern is shown over azimuth angle in a cartesian coordinate system.

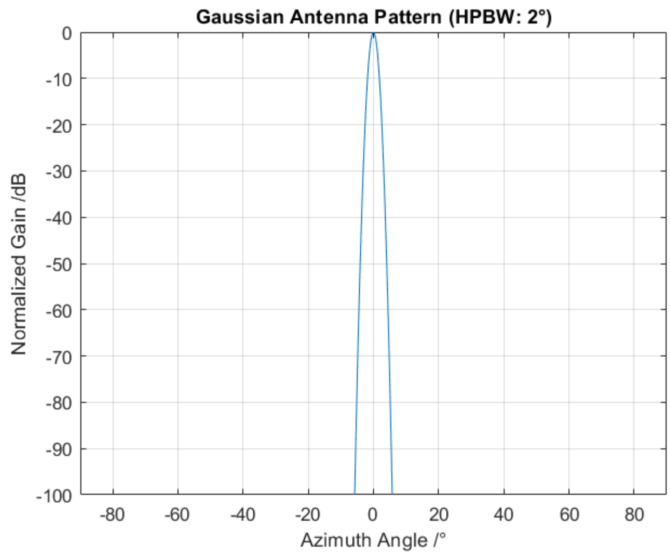

Figure 22: Normalized gain vs. azimuth angle of gaussian antenna

The gaussian transmit pattern a_t can be described as

$$a_t(\theta) = \exp\left(\frac{-\theta^2}{2\sigma^2}\right), \tag{3}$$

where θ describes the azimuth angle and σ the standard deviation of the gaussian distribution. It can be derived from the half-power beamwidth as

$$\sigma = \frac{HPBW}{2\sqrt{2\ln 2}} = \frac{2°}{2\sqrt{2\ln 2}} = 0.849°. \tag{4}$$

The azimuth angle of the receiver in relation to the emitter over time can be described as

$$\theta(t) = mod\left(210° - \frac{15 \cdot 360°}{60}t\,,360°\right), \tag{5}$$

where t is given in seconds. For example, after 6.3 seconds the antenna completed one scan and points 3° to the left of the receiver, looking from emitter to receiver. The corresponding normalized antenna gain is

$$a_t(3) = \exp\left(\frac{-3^2}{2 \cdot 0.849^2}\right) = 1.943 \cdot 10^{-3} = -27.1\text{dB}\,,$$

Assuming a dynamic range of 80 dB for the DUT, we specify a threshold of P_{min}= -80 dBm for further calculation. Signals that are received with a power level below this threshold are omitted.

The receiver is at a bearing of 210° relative to the emitter. As the circular antenna scan of the emitter is clockwise and starts pointing north, the antenna has to be turned by almost 210° until the threshold is overcome for the first time.

To determine the according time, we first calculate the angle θ_{min} relative to antenna main lobe, which corresponds to an attenuation of

$$a_t(\theta_{min}) = P_{min} - P_{r,max} = -80dBm - (-0.32dBm) = -79.68dB = 1.07628857 \cdot 10^{-8} \tag{6}$$

by solving Eqs. (3) and (4) for θ.

$$\theta_{min} = \sqrt{-2\sigma^2 \ln(a_t(\theta_{min}))} = 5.1429°$$

So, the antenna turns by $210° - 5.1429° = 204.8571°$, before the first pulse is received. This corresponds to a time of 2.27619038 seconds. As the emitter has a pulse repetition interval of 50us, the first pulse hitting the receiver is transmitted after 2.276200 seconds. Adding the time, it takes the signal to propagate from emitter to receiver, i.e. 8.339166µs, we get the TOA parameter of the first PDW. The corresponding level offset is calculated according to Eqs. (3) and (5) as

$$\theta(2.276200) = mod\left(210° - \frac{15 \cdot 360°}{60} 2.276200, 360°\right) = 5.142°$$

and

$$a_t(5.142) = \exp\left(\frac{-5.142^2}{2 \cdot 0.849^2}\right) = -79.65\, dB.$$

To determine the frequency offset of the first pulse, we divide the first TOA by the PRI to get the total number of (omitted) pulses and use modulo-6 operation, as the frequency offsets repeat every six pulses.

$$mod\left(\frac{2.276200}{50 \cdot 10^{-6}}, 6\right) + 1 = 3$$

The first pulse in our scenario has a frequency offset of 100 Mhz.

For a period of $2 \cdot \theta_{min} \cdot \frac{60}{15 \cdot 360°} = 0.1143s$ the receiver is now illuminated by the emitter. Within this period each 50us a new PDW is issued.

For the second PDW, the TOA parameter is incremented by 50 us. TON is constant with 10 us. As frequency offset we take the next entry in the list, which is -50 MHz. The phase offset is 0° and we still don't have a MOP. The only parameter remaining is the level offset, which is calculated as shown above.

An excerpt of the PDW list for one illumination period is given below:

No.	TOA /s	TON /µs	Frequency Offset /MHz	Level Offset /dB	Phase Offset /°	MOP
1	2.276208339167	10	100	-79.77	0	-
2	2.276258339167	10	-50	-79.50	0	-
3	2.276308339167	10	50	-79.44	0	-
…						
1142	2.333258339167	10	-50	-0.04	0	-
1143	2.333308339167	10	50	-0.13	0	-
1144	2.333358339167	10	150	-0.22	0	-
…						
2284	2.390358339167	10	150	-79.48	0	-
2285	2.390408339167	10	-100	-79.40	0	-
2286	2.390458339167	10	0	-79.63	0	-

3.3 Moving Receiver

So far, we only considered static emitters and receivers. Normally, when hardware-in-the-loop testing is performed, the DUT is moved through a virtual environment. Therefore, we want to add some motion to our scenario. The receiver from scenario 3.2 shall move north along a straight line with a velocity of 100m/s and pass the emitter on the west.

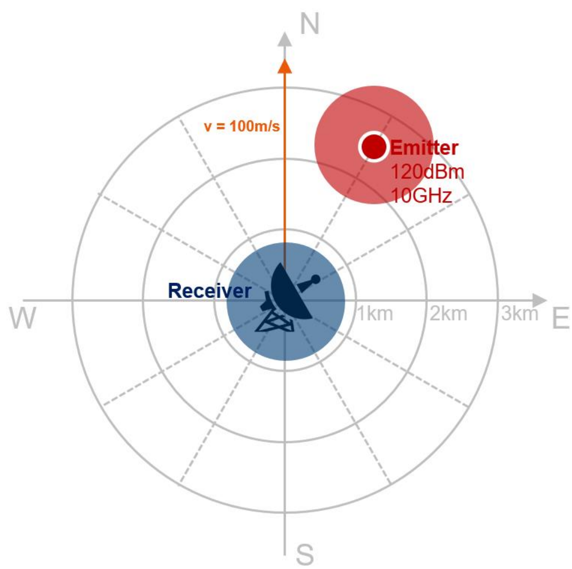

Figure 23: Scenario with static threat and moving receiver

Compared to scenario 3.2, the emitter also has an omnidirectional antenna and no scan. The power level at the receiver changes due to the varying distance between emitter and receiver. So does the Doppler frequency due to the varying relative velocity.

The receiver starts with a distance of 2500 m and a bearing of 30° to the emitter. It comes as close as 1250 m at a bearing of 90° and increases its distance back to 2500 m at a bearing of 150°. The total distance is

$$2 \cdot \cos 30° \cdot 2500 \text{ m} = \ 4330 \text{ m}.$$

At a velocity of 100 m/s, the movement takes 43.3 seconds. Distance and relative velocity over time is shown below.

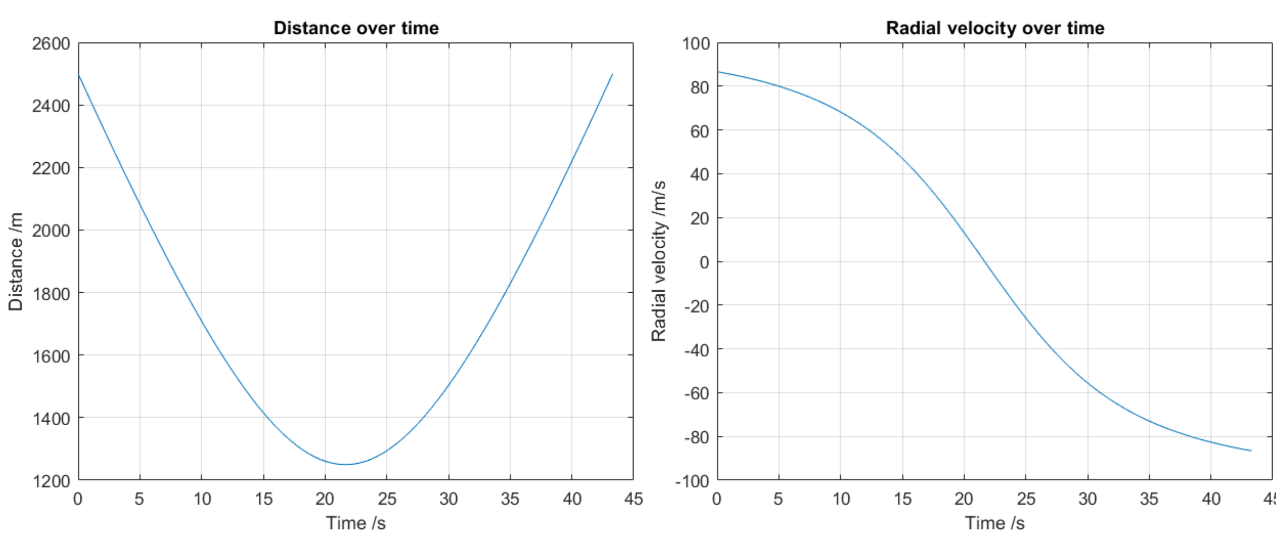

Figure 24: Relative distance and radial velocity between moving receiver and static emitter over time

The first step is to calculate the maximum received power

$$P_{r,max} = \frac{P_t G_t G_r c^2}{(4\pi f R_{min})^2} = \frac{10^{\left(\frac{120\text{ dBm}}{10\text{ dBm}}\right)} \cdot 10^{\left(\frac{0\text{ dB}}{10\text{ dB}}\right)} \cdot c^2}{(4\pi \cdot 9.9\text{ GHz} \cdot 1250\text{ m})^2} = 5.70\text{ dBm}$$

that is set as RF power level in the SMW. Relative to this level, level offsets are provided in the PDWs.

Note: In HIL testing, the minimum distance between emitter and receiver is not always known in advance. In this case the RF level has to be set sufficiently high in order to avoid clipping.

Changing distance between emitter and receiver over time affects four of the PDW parameters:

► Level offset

► Frequency

► TOA

The level offset resulting from the changing distance is calculated using the one-way radar equation. At the beginning of the scenario it is

$$P_{r,max} - P_r = P_{r,max} - \frac{P_t G_t G_r c^2}{(4\pi f R)^2} = 5.70\text{ dBm} - \frac{10^{\left(\frac{120\text{ dBm}}{10\text{ dBm}}\right)} \cdot 10^{\left(\frac{0\text{ dB}}{10\text{ dB}}\right)} \cdot c^2}{(4\pi \cdot 9.9\text{ GHz} \cdot 2500\text{ m})^2} = 6.02\text{ dB}.$$

The frequency offset in the PDW consists of two components:

► Doppler frequency shift

► Frequency hopping

The offset due to frequency hopping is -100 MHz for the first PDW. The doppler frequency shift can be calculated as

$$\Delta f = \frac{\Delta v}{c} f\ . \qquad (7)$$

With the parameters for our scenario, we get an initial doppler shift of

$$\Delta f = \frac{86.60\,\frac{\text{m}}{\text{s}}}{c} \cdot 9.9\text{GHz} = 2.860\text{kHz}\,,$$

which gives us a total frequency offset of -99.99714 MHz.

In a similar manner as the RF frequency also the PRI or PRF is affected by the moving receiver. When the receiver moves towards the emitter, a smaller PRI than actually used, is observed. Once the receiver passed the emitter, it senses a higher PRI.

The observed PRI can be calculated with

$$PRI = \frac{PRI_0}{(1 + \frac{\Delta v}{c})}\,,$$

where PRI_0 is the actual PRI and PRI is the observed PRI. The first TOA is determined by the propagation time, that is 8.339166µs as in the scenarios above. An increment of

$$PRI = \frac{50\text{µs}}{(1 + \frac{86.60\,\frac{\text{m}}{\text{s}}}{c})} = 49.999986\text{µs}$$

gives us the next TOA parameter. As the time resolution is limited to 416.67ps on the SMW, the difference between actual and observed PRI becomes visible in the TOA parameter approx. every 29 PDWs. Strictly speaking, the pulse width is influenced in the same way. However, in our scenario the difference between actual and observed pulse width is well below the SMWs resolution.

An excerpt of the PDW list for this scenario is given below:

No.	TOA /s	TON /µs	Frequency Offset /MHz	Level Offset /dB	Phase Offset /°	MOP
1	0.000008339167	10	-99.997140	-6.02	0	-
2	0.000058339167	10	0.002889	-6.11	0	-
3	0.000108339167	10	100.002918	-6.19	0	-
…						
28	0.00135833875	10	-49.997126	-6.06	0	-
29	0.00140833875	10	50.002903	-6.15	0	-
30	0.00145833875	10	150.002932	-6.24	0	-
…						
432999	21. 649904169583	10	100.000000	-0.17	0	-
433000	21. 649954169583	10	-50.000000	-0.04	0	-
433001	21. 650004169583	10	50.000000	-0.13	0	-

3.4 Multiple emitters with and without pulse dropping

Up to now, we are able to calculate PDWs for a receiver moving around a static emitter. We considered emitter EIRP, range, Doppler frequency shift, frequency agility, antenna pattern and antenna scans. The next step is to bring more than one emitter into the scenario.

In a real environment, when multiple emitters are transmitting simultaneously there is a chance that pulses from different emitters collide. The receiver would then receive both pulses at the respective frequency of each emitter. Whereas some radar warning receivers can handle overlapping pulses, others are not able to analyze those signals properly.

During simulation, there are different options to realize a multiple emitter scenario and to handle pulse collisions. In this section, we will have a look at those options with their advantages and disadvantages based on the following example:

We take the scenario from section 0 and add another emitter at an initial bearing of 270° and a distance of 2 km from the receiver.

The emitter parameters are:

► EIRP = 110 dBm

► RF frequency f = 9.5 GHz

► Pulse width = 22 us

► PRI = 60 us

► No frequency hopping, no MOP

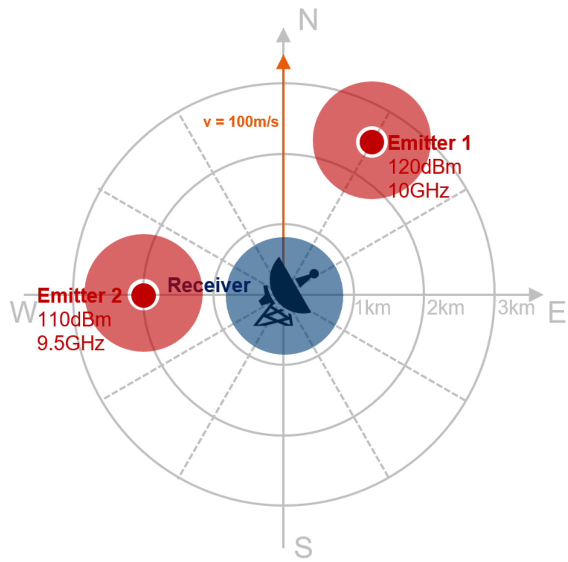

Figure 25: Scenario with two static threats and static receiver

3.4.1 Pulse dropping

The first and also easiest method is to calculate PDWs for both emitters and just send them to the SMW. By doing so, we can get two kinds of collisions:

► **Two PDWs with exact same TOA parameter are sent**

In this case, the SMW processes the PDW that is received first (red) and drops the other one (blue)

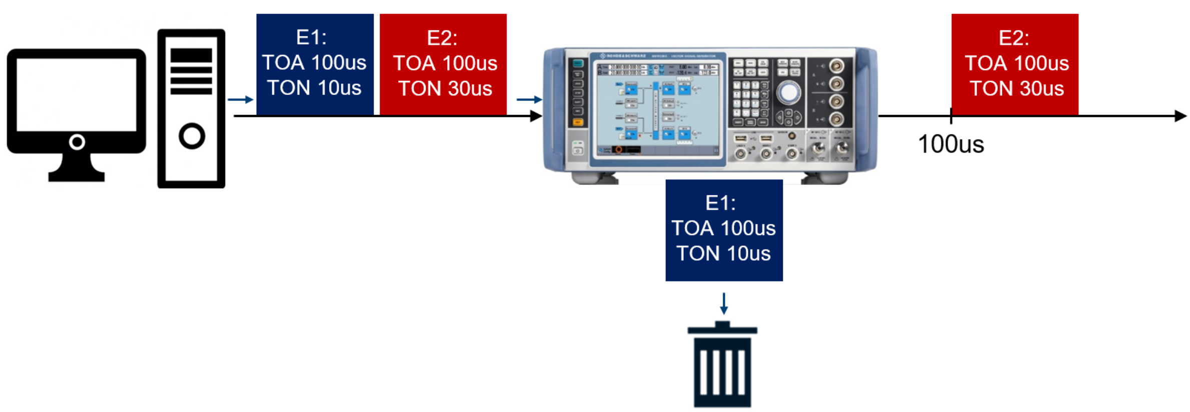

Figure 26: Pulse dropping: PDWs with same TOA parameter

► **The TOA parameter of a PDW is such, that playback should start, before the previous pulse is finished**

In this case, the SMW aborts the previous signal (red) even though not finished and starts playback of the new signal (blue) at the dedicated TOA.

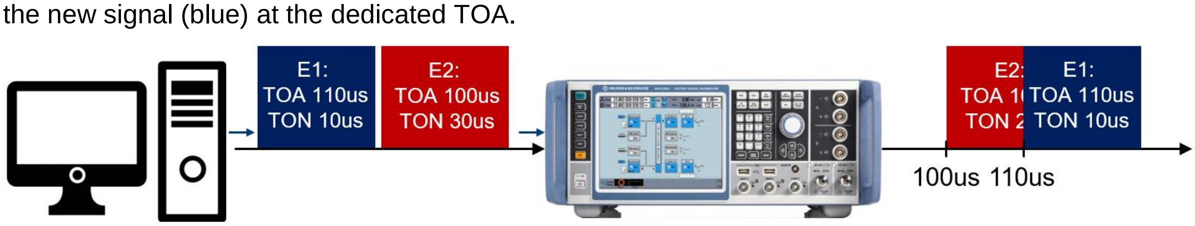

Figure 27: Pulse dropping: PDW with TOA parameter before preceeding PDW is finished

Note: PDWs have to be sent in the correct order of the TOA parameter, e.g. if a PDW with a TOA of 100us is sent after one with a TOA of 110us, it will be dropped even if there is no collision. Additionally, the minimum PRI according to the SMW-K503/K504 datasheet has to be observed.

3.4.2 Priority based pulse dropping

With the first approach, there is no rule according to that PDWs are dropped. It is random if PDWs of Emitter 1, Emitter 2 or emitter x (if more than two emitters are part of the scenario) are dropped. If we have a scenario with more and less important emitters, we probably would prefer that all PDWs of important emitters are played and only PDWs of less important emitters are dropped in case of collisions.

This can be achieved with a priority scheme in the HIL simulator. Thereby, a priority is assigned to each emitter according to which the PDWs are calculated. If for a specific TOA a PDW for a high priority emitter was generated, all emitters with lower priority will be ignored at that point in time and for the pulse-on-time indicated in the PDW. The following figure illustrates this principle

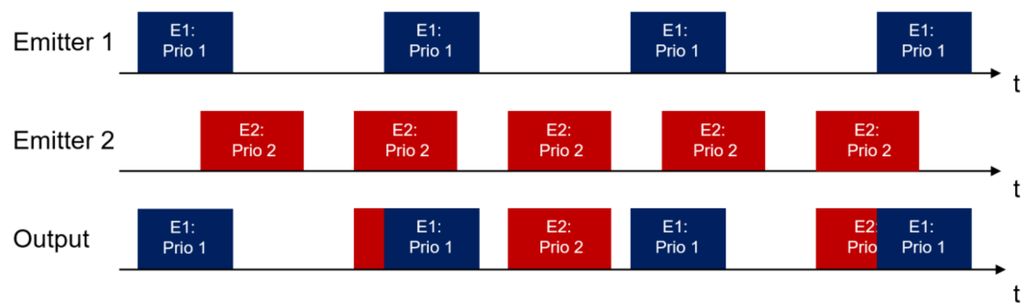

Figure 28: Priority based pulse dropping

Emitter 1 has the highest priority (Prio=1) in this scenario. For this emitter all PDWs are generated. Emitter 2 has only a priority of 2 and therefore, only some PDWs are generated. In case of collisions, PDWs for Emitter 1 are preferred.

Of course, the priority scheme in the HIL simulator could be extended to avoid interrupted PDWs as well.

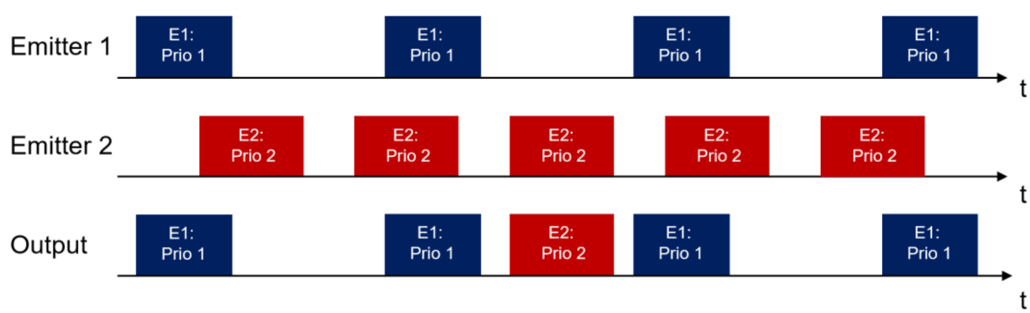

Figure 29: Priority based pulse dropping without interrupted PDWs

3.4.3 Pulse-on-pulse simulation

In section 2.1, it was already shown, that the SMW can be equipped with up to two wideband baseband generators (SMW-B9), also called coder boards, and two or four wideband fading simulators (SMW-B15), also called fader boards, each with its own 1Gbit/s LAN interface. By assigning a single emitter to each hardware board, it is possible to simultaneously simulate up to six emitters without dropping a single PDW.

For our example above, it works as follows: as in the first approach to multi emitter simulation, described in 3.4.1, PDWs are generated for each emitter. The PDWs for Emitter 1 are streamed to one SMW-B9 coder board, the PDWs for Emitter 2 are streamed to the second SMW-B9. Based on the PDWs the respective baseband signals are generated on each SMW-B9. Both baseband signals can then be routed to a common RF path in the I/Q Stream Mapper.

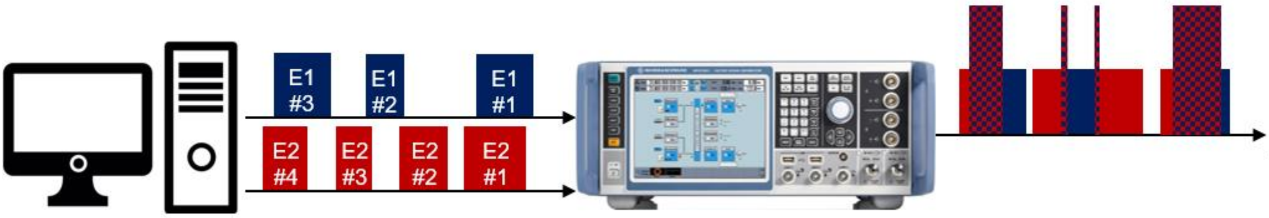

Figure 30: Pulse-on-pulse simulation on one RF output with two parallel PDW streams

Stream mapper configuration to route signals from Baseband A ("Stream A") and Baseband B ("Stream B") to RF A:

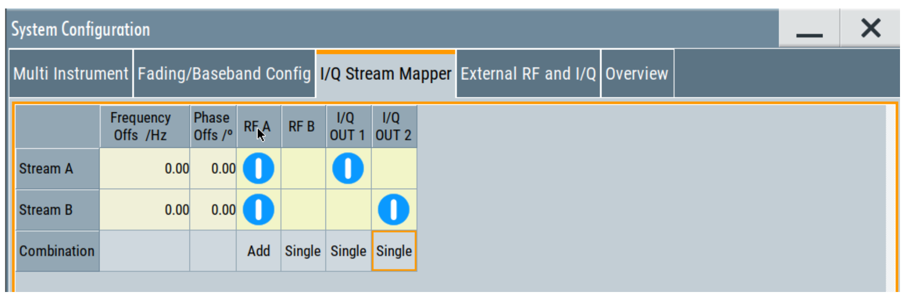

Figure 31: Stream mapper configuration dialog in the SMW200A

Block diagram with above configuration:

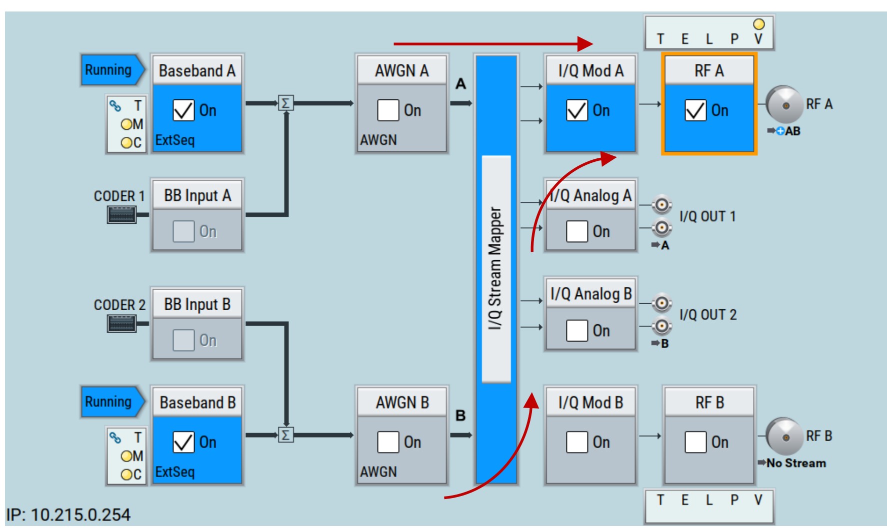

Figure 32: Block diagram showing the signal routing in the SMW200A

In this configuration two independent I/Q signal streams are added and the corresponding RF signal is provided at a single RF output. The RF frequency and level setting in the SMW is configured with respect to this RF output, i.e. with respect to the sum signal. The following example shall illustrate how the level setting in the SMW influences the power level of pulses at the RF output. For simplicity, we assume a level offset of 0 dB in the PDWs for either of the streams and a level setting of 0 dBm in the SMW (these are the two parameters that can be adjusted by the user).

In a standard single stream scenario (one I/Q signal stream routed to one RF output), the baseband I/Q signal has a level of 0 dBFS in case the level offset in the PDW is 0 dB. With a level setting of 0 dBm this results in pulses with a pulse top power of 0 dBm at the RF output.

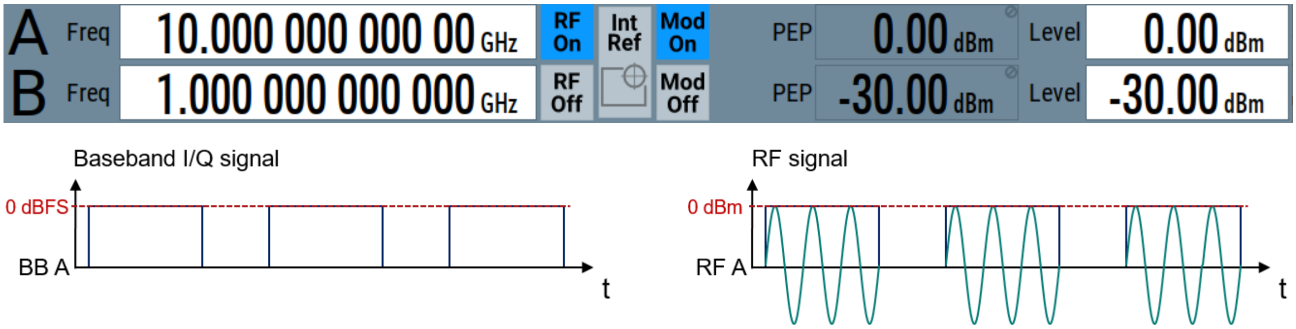

Figure 33: SMW level setting with single I/Q signal from baseband A (BB A) assigned to RF path A

When two I/Q signals are added and assigned to a single RF output, two things happen:

- Both I/Q signals are attenuated by 3.01 dB in the digital domain. As a result the top power of a pulse with 0 dB level offset in the corresponding PDW is -3.01 dBm at the RF output.
- If pulses from two different I/Q streams overlap, the level of the sum signal can increase by up to 6.02 dB. To avoid clipping of the sum signal, the attenuator chain of the SMW is set to a PEP value of 3.01 dBm (which is 6.02 dB above the level of the individual I/Q signals).

This is illustrated in Figure 34

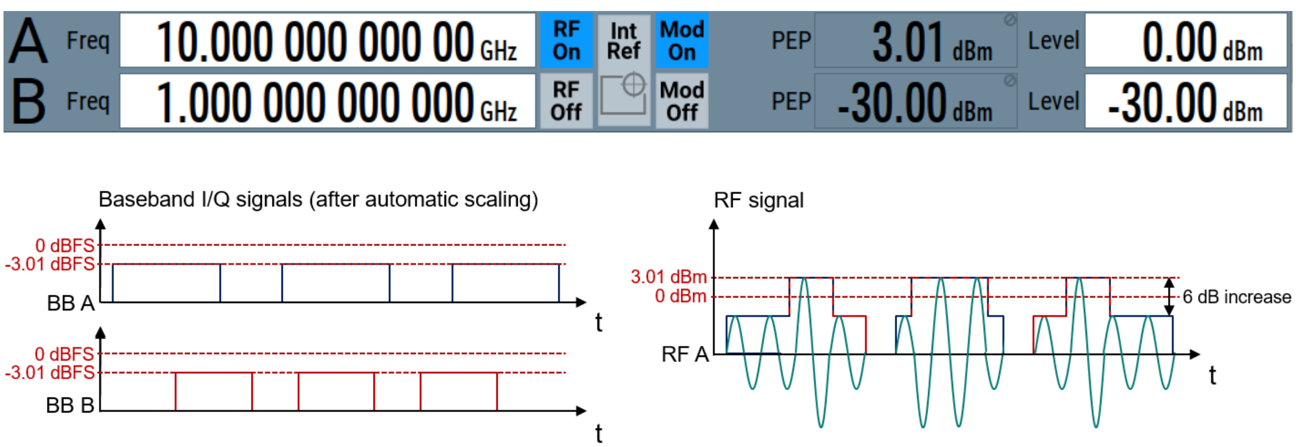

Figure 34: SMW level setting with two I/Q signals from baseband A (BB A) and B (BB B) assigned to RF path A

More information on multi emitter and pulse-on-pulse simulation is given in chapter 4.3.

3.5 Single emitter with AoA (DF)

Modern radar warning receivers use input signals from multiple antennas to determine the direction of arrival (DOA) of a signal from an emitter. Direction finding can be based on time difference of arrival (TDOA), amplitude differences or phase differences of the receive signals. To test this functionality in the lab, multiple generators are coupled, and the signals to be applied to each receiver port are calculated and played synchronously. More information on the configuration and requirements of a multi-channel signal generator setup to test direction finding capabilities are given in [2].

In this section, we only have a look on the calculation of time and carrier phase differences based on direction of arrival of the emitter signal, antenna geometry of the receiver and RF frequency to generate adequate PDWs.

Let’s assume the following scenario:

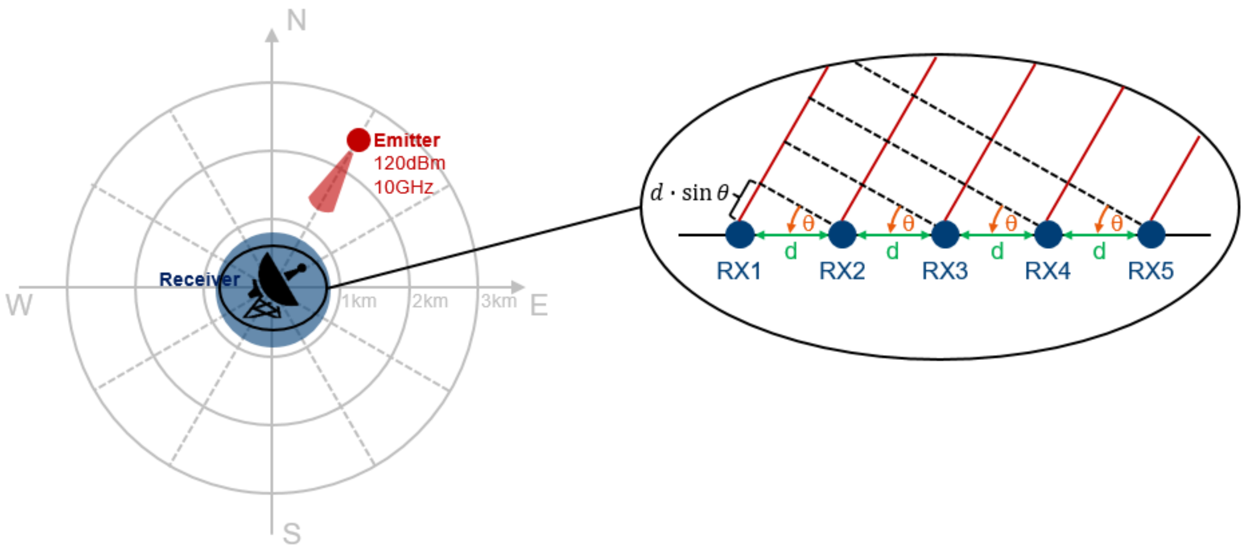

Figure 35: Scenario with static threat and static multi-antenna receiver

We have a receiver with five antenna elements that are uniformly spaced by the distance d. An emitter (pulse width = 10 µs, PRI = 50 µs) with an RF frequency of 10 GHz and an EIRP of 120 dBm is located at a bearing of θ = 30° from the receiver and range of 2.5 km. At this range, we can assume plane waves at the receiver.

The figure above shows, that the incident signal arrives at each antenna element at a different time, which can be translated into a different phase for narrowband signals.

For further calculations, we assume a spacing d of half a wavelength λ at 10 GHz:

$$d = \frac{\lambda}{2} = \frac{c}{2f}$$

The time offset between two adjacent antenna elements can be determined by dividing the range difference by the speed of propagation. We derive the range difference from the antenna geometry and the AoA as

$$\Delta R = d \cdot \sin\theta = \frac{c \cdot \sin\theta}{2f}.$$

The time offset is

$$\Delta t = \frac{d \cdot \sin\theta}{c} = \frac{\sin\theta}{2f}.$$

For our example, we get a time offset of

$$\Delta t = \frac{\sin\theta}{2f} = \frac{\sin 30°}{2 \cdot 10\ \mathrm{GHz}} = 25\ \mathrm{ps}\,,$$

which is well below the time resolution of the SMW of approx. 417ps. Even the time difference between RX1 and RX5 with 100ps is below this threshold. Therefore, we use the narrowband assumption, that allows us to represent the time distance as phase offset:

$$\Delta\varphi = \omega\Delta t = 2\pi f \Delta t = \pi \sin\theta$$

The phase offset between two adjacent antenna elements results as

$$\Delta\varphi = \pi \sin\theta = \pi \sin 30° = 90°\,.$$

Generally, RWRs only evaluate the relative phase differences between multiple input signals and not the phase information of a single input signal. Taking that into account, we can choose one antenna element as reference, e.g. RX1, and calculate all phase offsets with respect to the reference element. The PDWs for all receive paths for a certain time instance could look like this:

Antenna	No.	TOA /s	TON /µs	Frequency Offset /MHz	Level Offset /dB	Phase Offset /°	MOP
RX1	1	0.000008339166	10	0.0	0.0	0	-
RX2	1	0.000008339166	10	0.0	0.0	90	-
RX3	1	0.000008339166	10	0.0	0.0	180	-
RX4	1	0.000008339166	10	0.0	0.0	270	-
RX5	1	0.000008339166	10	0.0	0.0	0	-
RX1	2	0.000058339166	10	0.0	0.0	0	-
RX2	2	0.000058339166	10	0.0	0.0	90	-
RX3	2	0.000058339166	10	0.0	0.0	180	-
RX4	2	0.000058339166	10	0.0	0.0	270	-
RX5	2	0.000058339166	10	0.0	0.0	0	-

For the calculation above, two assumptions were used:

► Narrowband assumption

► Plane wave assumption

The narrowband assumption is necessary to represent the different arrival times as phase offsets. For other scenarios, especially with larger antenna arrays it is only partially required. Let's calculate the PDWs for the scenario above, assuming a two-element antenna array with an element spacing of 12.3 m, e.g. at the wingtips of an aircraft.

$$\Delta t = \frac{d \cdot \sin\theta}{c} = \frac{12.3 \text{ m } \cdot \sin 30°}{c} = 25.514 \text{ ns}$$

This time offset can now be separated in a time offset and a phase offset: 20.514ns equals 49 clock cycles at 417 ps. The remaining time of

$$20.514 \text{ ns} - 49 \cdot \frac{1}{2.4 \text{ GHz}} = 97.33 \text{ ps}$$

is translated into a phase offset of

$$\Delta\varphi = 2\pi f \Delta t = 2\pi \cdot 10 \text{ GHz} \cdot 97.33 \text{ ps} = 350.4° .$$

Given the granularity of 417 ps, the combination of time and phase offset provides the most realistic signal.

The PDWs would look as follows (with plane wave assumption):

Antenna	No.	TOA /s	TON /µs	Frequency Offset /MHz	Level Offset /dB	Phase Offset /°	MOP
RX2	1	0.000008339166	10	0.0	0.0	350.4	-
RX1	1	0.000008359583	10	0.0	-0.385	0	-
RX2	2	0.000058339166	10	0.0	0.0	350.4	-
RX1	2	0.000058359583	10	0.0	-0.385	0	-

Note: For this antenna geometry, the difference in receive power cannot be neglected anymore.

Last but not least, we want to have a look on the validity of the plane wave assumption. Therefore, we take our example from the beginning of this section and calculate the phase of each receiver input signal based on the range from receiver to emitter as

$$\varphi_x = \frac{R_x}{\lambda} \cdot 360° ,$$

where R_x is the distance from antenna element x to the emitter and φ_x is the absolute phase.

Antenna	No.	TOA /s	TON /µs	Frequency Offset /MHz	Level Offset /dB	Phase Offset /°	MOP
RX1	1	0.000008339166	10	0.0	0.0	351.43217	-
RX2	1	0.000008339166	10	0.0	0.0	81.43176	-
RX3	1	0.000008339166	10	0.0	0.0	171.43055	-
RX4	1	0.000008339166	10	0.0	0.0	261.42852	-
RX5	1	0.000008339166	10	0.0	0.0	351.42569	-

After referencing the phase offset to the first antenna element, we get:

Antenna	No.	TOA /s	TON /µs	Frequency Offset /MHz	Level Offset /dB	Phase Offset /°	MOP
RX1	1	0.000008339166	10	0.0	0.0	0.0	-
RX2	1	0.000008339166	10	0.0	0.0	89.99959	-
RX3	1	0.000008339166	10	0.0	0.0	179.99838	-
RX4	1	0.000008339166	10	0.0	0.0	269.99636	-
RX5	1	0.000008339166	10	0.0	0.0	359.99352	-

The results are pretty close to what we get using the plane wave assumption.

3.6 Function summary

In the previous sections 3.1 to 3.5, a detailed description of the most common intermediate calculations to generate realistic EW environments was given. Now, a short summary is given in the form of pseudo-code functions. Thereby, an object-oriented approach is taken, i.e. the functions are assigned to a receiver or an emitter class when feasible. This is just one option. Of course, other structures are possible. Python code for the functions below and some examples from the previous sections are provided with this application note.

```
Emitter:

xpos = 0
ypos = 0
zpos = 0
eirp = 120
PRI = 50e-6
PW = [10e-6]
RFfreq = 10e9
freqHopping = []
dwell = 1
scan = 'None'
initAntennaAngle = 0
scanrate = 15 #rpm
pattern = 'Omni'
HPBW = 2


setPositionAngular(radius, heading, pitch):
    (xpos, ypos, zpos) = pol2cart(radius, heading, pitch)

hasPDW(t):
    if (t % PRI) == 0:
        return True
    else:
        return False

getPulseOnTime(t):
    pdwCount = int(t/PRI)
    return PW[pdwCount%len(PW)]
```

```
getCurrentFrequency(t):
    pdwCount = int(round(t/PRI))
    return (RFfreq + getCurrentFreqHopping(t))

getCurrentFreqHopping(t):
    pdwCount = int(t/PRI)
    if freqHopping:
        return freqHopping[pdwCount%len(freqHopping)]
    else:
        return 0

getEIRP(relativeBearing, t):
    if pattern == 'Omni':
        return eirp
    else if pattern == 'Gauss':
        currentAngle = (relativeBearing - getAntennaAngle(t)) % 360
        attdB = getAntennaGain(currentAngle)
        return attdB + eirp

getAntennaGain(angle):
    if pattern == 'Gauss':
        if angle > 180:
            angle = 360 - angle
        sigma = HPBW / (2 * sqrt(2 * log(2)))
        att = exp(-(angle ** 2) / (2 * sigma ** 2))
        if att == 0.0:
            att = 1e-12
        attdB = 10 * log10(att)
        return attdB

getAntennaAngle(t):
    if scan == 'None':
        return initAntennaAngle
    else if scan == 'Circ':
        return initAntennaAngle + (scanrate * 360 / 60 * t)
```

Receiver:

```
c0 = 299792458
xpos = 0
ypos = 0
zpos = 0
vx = 0
vy = 0
vz = 0
pattern = 'Omni'

setPositionAngular(radius, heading, pitch=0):
    (xpos, ypos, zpos) = pol2cart(radius, heading, pitch)

setVelocityAngular(vmag, heading, pitch=0):
    (vx, vy, vz) = pol2cart(vmag, heading, pitch)

getCurrentPosition(t):
    return (xpos + vx*t, ypos + vy*t, zpos + vz*t)

getDistanceFromEmitter(emitter, t):
    (currentx, currenty, currentz) = getCurrentPosition(t)
    return norm([currentx-emitter.xpos, currenty-emitter.ypos, currentz-
           emitter.zpos])
```

```
getRelativeVelocityToEmitter(emitter, t):
    (currentx, currenty, currentz) = getCurrentPosition(t)
    doa = (emitter.xpos - currentx, emitter.ypos - currenty, emitter.zpos -
          currentz)
    return (dot(doa, [vx, vy, vz])/norm(doa))

getTimeOfFlight(emitter, t):
    tof = getDistanceFromEmitter(emitter, t)/c0
    return round(tof*2.4e9)/2.4e9

getObservedFrequeny(emitter, t):
    freq = emitter.getCurrentFrequency(t)
    v = getRelativeVelocityToEmitter(emitter, t)
    return (freq * (1+v/c0))

getRxLevel(emitter, t, Prmax):
    dist = getDistanceFromEmitter(emitter, t)
    levelOffset = Prmax -
   10*log10((10**(emitter.getEIRP(getRelativeBearing(emitter), t)/10) *
   getAntennaGain(emitter, t) * c0**2) / ((4*pi*emitter.getCurrentFrequency(t)*
   getDistanceFromEmitter(emitter, t))**2))
    return levelOffset

getAntennaGain(emitter, t):
    if pattern == 'Omni':
        return 1

getRelativeBearing(emitter):
    relativeBearing = arctan(xpos - emitter.xpos, ypos - emitter.ypos)
    return rad2deg(relativeBearing)
```

Helper Functions:

```
cart2pol(x, y, z):
    r = sqrt(x**2 + y**2 + z**2)
    pitch = rad2deg(arctan2(z, sqrt(x**2 + y**2)))
    heading = rad2deg(arctan2(x, y))
    return(r, heading, pitch)

pol2cart(r, heading, pitch):
    x = r * sin(deg2rad(heading)) * cos(deg2rad(pitch))
    y = r * cos(deg2rad(heading)) * cos(deg2rad(pitch))
    z = r * sin(deg2rad(pitch))
    return(x, y, z)
```

With these functions, the example from section 0 can be realized as follows:

```
pdwCount = 0
clockCycle = 50e-6
smwRfLevel = 5.7
smwRfFrequency = 10e9

E1 = Emitter(PW=[10e-6], PRI=50e-6, RFfreq=10e9, EIRP=120, freqHopping=[-100e6,
0, 100e6, -50e6, 50e6, 150e6],
             pattern='Omni')
E1.setPositionAngular(2500, 30, 0)
Rx = Receiver()
Rx.setVelocityAngular(100, 0, 0)

currentSimTime = 0.0
scenarioEndTime = 43.3
```

```
while currentSimTime < scenarioEndTime:
    if E1.hasPDW(currentSimTime):
        # intermediate calculations
        levelOffset = Rx.getRxLevel(E1, currentSimTime, smwRfLevel)
        TOA = currentSimTime + Rx.getTimeOfFlight(E1, currentSimTime)
        TON = E1.getPulseOnTime(currentSimTime)
        freqOffset = Rx.getObservedFrequeny(E1, currentSimTime) - E1.RFfreq
        pdwCount += 1
    currentSimTime += clockCycle
```

4 Additional hints for realtime threat generation

4.1 Timed control descriptor words (TCDWs) to change RF frequency or RF power level

The R&S PDW format allows to provide frequency and level offsets for each pulse. The I/Q modulation bandwidth of the SMW of 2 GHz allows frequency offsets of ±1GHz around the RF frequency. The level offset is relative to the current RF level. It is applied as attenuation of the baseband signal in the digital domain.

Sometimes the modulation bandwidth or the dynamic range of the digital baseband are not sufficient for simulation. Then it is necessary to change the RF settings in the SMW.

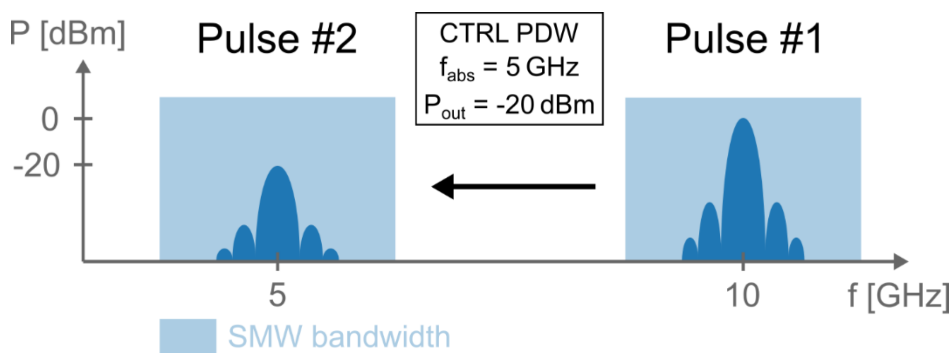

Figure 36: Switching RF level and RF frequency with TCDW

This can be done with so-called TCDWs. A TCDW is a 16Byte word which contains a TOA, the RF path in which the frequency and/or level shall be changed, a command type designator and the new RF frequency and/or RF level. More information on TCDWs is given in [5].

It is advisably to choose the TOA of the PDW following the TCDW such that the frequency and/or level have settled. Frequency and level settling times are provided in [6].

A typical use case for RF frequency changes initiated by TCDWs is, when multiple emitters separated in frequency by more than 2 GHz (max. I/Q modulation bandwidth of the SMW) shall be simulated. Let’s assume we want to simulate the following pulse sequence of two emitters, one with a center frequency of 5 GHz and the other one with a center frequency of 10 GHz.

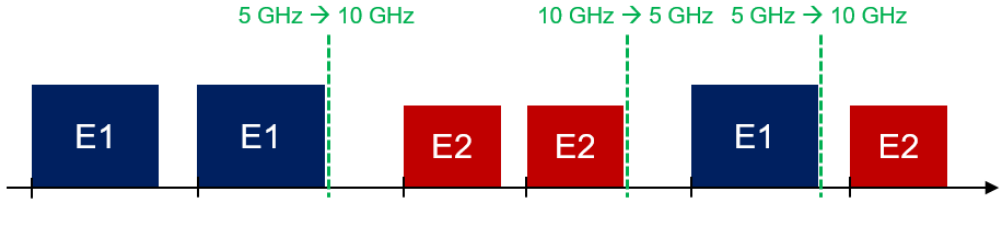

Figure 37: Simulation of two emitters at 5 GHz and 10 GHz

First, there are two pulses of emitter E1 followed by two pulses of emitter E2. Then there is another pulse of E1 and another pulse of E2. In between, the RF frequency of the SMW has to be changed from 5 to 10 GHz and vice versa. The frequency changes are initiated with TCDWs, which also have a TOA parameter to specify the time of the frequency changes. The PDW list to generate the above sequence could look like this (E1: PRI = 100 µs, pulse width = 10 µs; E2: PRI = 100 µs, pulse width = 8 µs)

Emitter	No.	TOA /s	TON /µs	Frequency Offset /MHz	Level Offset /dB	Phase Offset /°	MOP	RF frequency /GHz	RF level /dBm
E1	1	0.0000	10	0.0	0.0	0	-		
E1	2	0.0001	10	0.0	0.0	0	-		
CTRL		0.0002						10	0
E2	1	0.0020	8	0.0	-3.0	0	-		
E2	2	0.0021	8	0.0	-3.0	0	-		
CTRL		0.0022						5	0
E1	3	0.0040	10	0.0	0.0	0	-		
CTRL		0.0041						10	0
E2	3	0.0059	8	0.0	-3.0	0	-		

Note: When the RF frequency setting of the SMW is changed, i.e. the oscillator in the SMW is tuned to another frequency, to simulate multiple emitters, the phase for a single emitter is not continuous, i.e. there is no phase memory in which the previous emitter phase was stored.

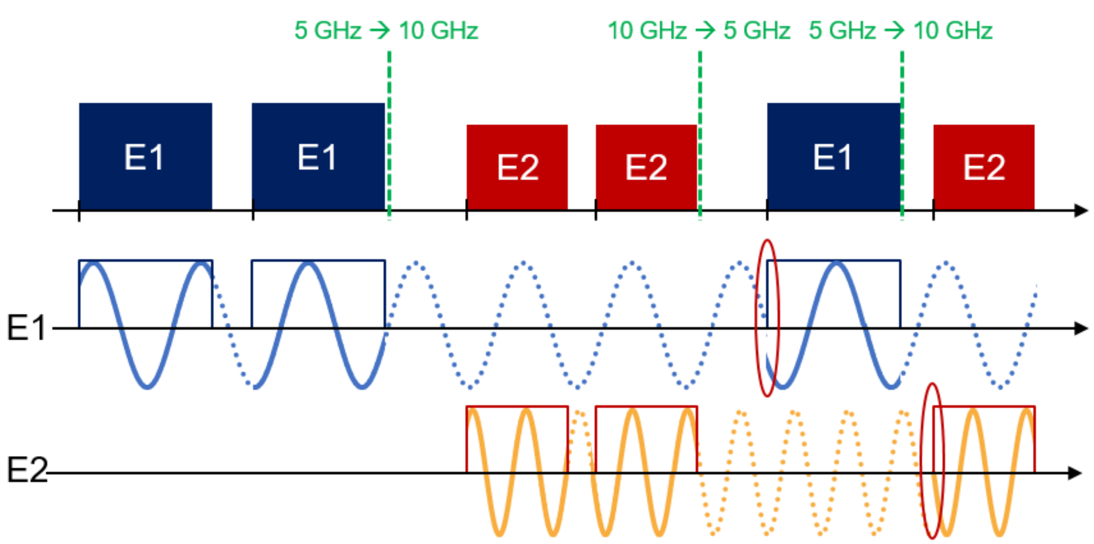

Figure 38: Phase of two emitters at 5 GHz and 10 GHz

4.2 CW interferer with PDWs

Besides pulsed signals it is also possible to simulate CW interferer signals. The easiest approach is to provide a single PDW to the SMW that contains a very long pulse on-time. The downside of this approach is, that the corresponding signal has a constant power level. In a realistic scenario, it is more likely that a CW signal is modulated, for example by an antenna pattern/scan or changing distance between emitter and receiver.

To accurately model such a scenario, we connect multiple pulses with changing amplitude. This principle is illustrated in the following figure:

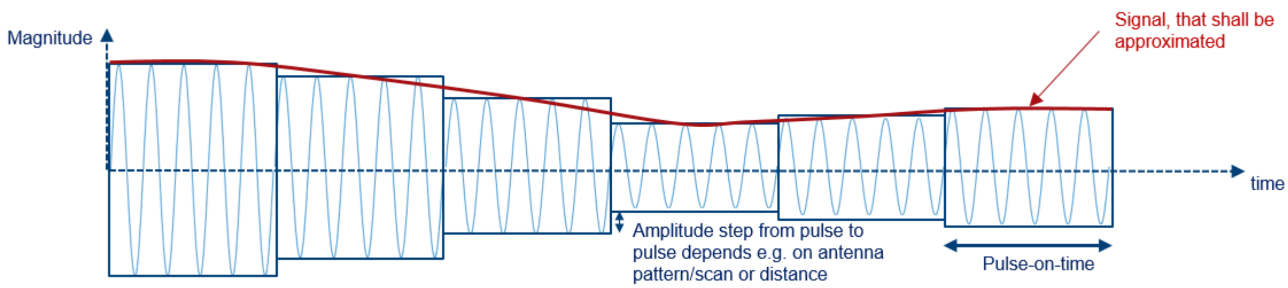

Figure 39: Concatenated PDWs to simulated CW interferer with changing signal amplitude

To avoid glitches, we have to guarantee a continuous carrier phase at the pulse transitions. Otherwise, the signal will contain higher order harmonics. A smooth phase is obtained by setting the phase mode in each PDW to relative and the phase offset to zero. Additionally, the PRI, which is equal to TON here, has to be a multiple of 8 clock cycles at a clock rate of 2.4 GHz.

Besides amplitude, changing Doppler or phase can be also considered with this approach.

The PDWs to approximate the above signal could look like this:

No.	TOA /s	TON /µs	Frequency Offset /MHz	Level Offset /dB	Phase Offset /°	Phase Mode
1	0.00000	10	0.0	0.0	0	relative
2	0.00001	10	0.0	-1.0	0	relative
3	0.00002	10	0.0	-2.0	0	relative
4	0.00003	10	0.0	-3.0	0	relative
5	0.00004	10	0.0	-2.5	0	relative
6	0.00005	10	0.0	-2.2	0	relative

4.3 Increased On/Off-Ratio

The on/off-ratio of pulsed signals can be increased by 80 dB when using the pulse modulator (option R&S®SMW-K22) in the SMW together with PDW streaming. Therefore, one has to check the box at 'RF Power Ramping with Burst Gate Marker' in the Extended Sequencer configuration dialog.

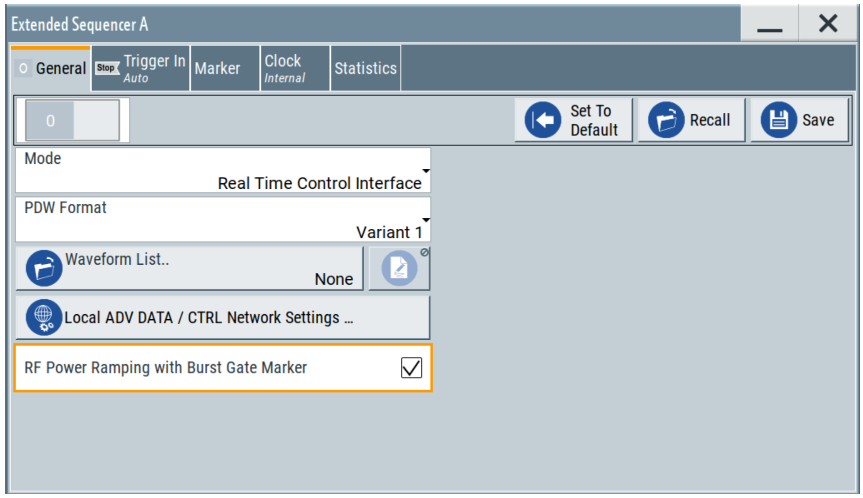

Figure 40: Configuration of the Burst Gate Marker in the Extended Sequencer configuration dialog

By activating the Burst Gate Marker an internal marker signal is generated, which controls the pulse modulator of the SMW. The marker goes high at TOA and back to low after TON (+ RISE TIME + FALL TIME when using shaped edges). This is illustrated in the following figure.

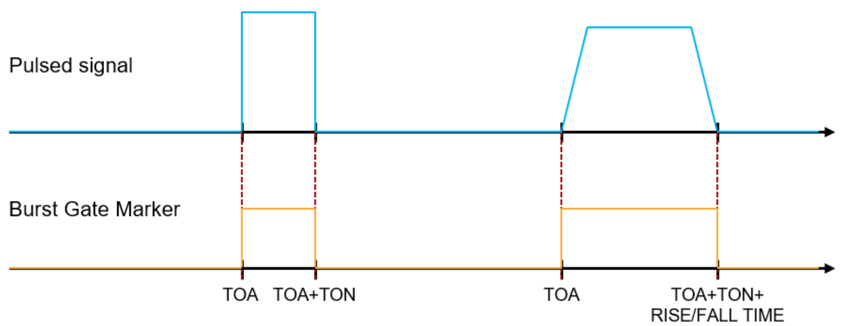

Figure 41: Illustration of Burst Gate Marker signal for pulsed signals

More information on the R&S®SMW-K22 Pulse Modulator can be found in [7].

5 Guidelines for advanced PDW streaming

Equipped with two R&S®SMW-B9 coder boards and four R&S®SMW-B15 signal processing boards, the SMW can handle up to six independent PDW streams with a maximum rate of 2 MPDW/s per stream. With a size of 32 byte per PDW, this means the HIL simulator needs to generate 64 MB/s of data per stream, in total 384 MB/s. The PDWs are received via 6 separate 1 GBit/s LAN interfaces on the individual coder and processing boards and processed in the signal generator.

In this chapter some guidelines for this kind of advanced PDW streaming are provided.

5.1 HIL simulator

The HIL simulator must be able to provide new PDWs representing the currently simulated situation at the required update rate. As we saw in section 2.4.3, the time required for PDW calculation is part of the total loop time.

Depending on the requirements, there are different options to realize a HIL simulator for radar signals.

5.1.1 Standard/High-End PC

Basic requirements on latency and update rate might be addressed with a standard/high-end PC taking on the role of a HIL simulator. Here are some tips to optimize its PDW streaming performance.

► **Work with clock cycles rather than times**

Each PDW contains a TOA that is given in number of clock cycles at a clock rate of 2.4 GHz. When using clock cycles instead of absolute times for the intermediate calculations described in chapter 3, we can save some mathematical operations for each PDW.

► **Unroll/parallelize loops**

The goal of unrolling loops is to increase PDW calculation by reducing or eliminating instructions that control the loop, such as counter increments and "end of loop" tests on each iteration.

► **Send with little overhead**

PDWs are sent using the TCP or the UDP protocol. Each protocol adds some overhead due to headers or checksums. To keep the overhead small in relation to the PDW size, it might help to send sets of multiple PDWs in one packet. The usage of the PDW burst functionality should also be considered.

5.1.2 Dedicated hardware

For low-latency and high RF-density testing, a COTS PC might not be sufficient. In that case, a GPU- or even a FPGA-based HIL simulator may be required. In comparison to CPUs, GPUs and FPGAs can often achieve better performance by parallelizing tasks.

For PDW streaming this means independent computations, like calculating Doppler frequency, power level or antenna angle can be performed in parallel tasks rather than sequentially. Apart from the intermediate calculations described in chapter 3, the assembly of a single PDW, consisting of header, flags, body and payload, can also be parallelized.

This way, GPU- or FPGA-based HIL simulators have the potential to achieve higher PDW streaming rates than PC-based HIL simulators.

5.2 Network topology

Gigabit LAN has a theoretical maximum throughput of 1 Gbit/s ≈ 118 MB/s. This is sufficient for a single PDW stream at max. PDW rate, where a data rate of up to 64 MB/s is required. However, it is not sufficient to provide the data for six parallel streams at max. PDW rate on a single line. Therefore, we can use two different network topologies:

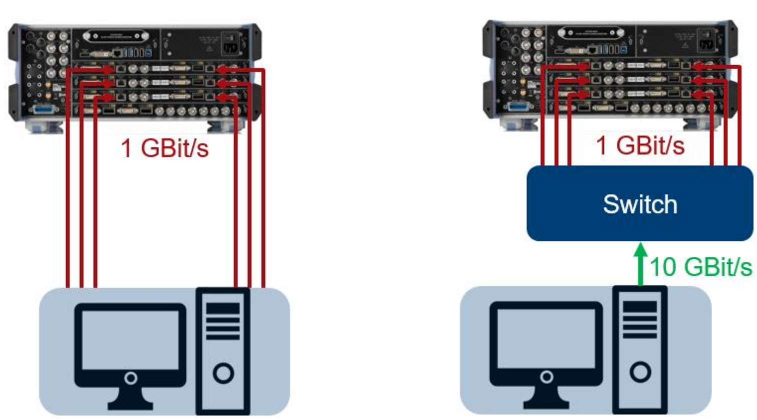

Figure 42: Possible network topologies for advanced PDW streaming

One option is to have a dedicated network adapter for each stream in the HIL simulator. This way, we have a physical network link for each PDW stream. Another option is to have a 10 GBit/s interface at the HIL simulator to provide PDWs for all parallel streams and an appropriate switch that distributes the PDWs among the coder and signal processing boards. Between switch and coder/processing board the maximum data rate is 64 MB/s, so that a standard 1 GBit/s ethernet connection is sufficient.

5.3 Radar signal generator

The SMW offers the Extended Sequencer Advanced mode for convenient configuration of six parallel PDW streams.

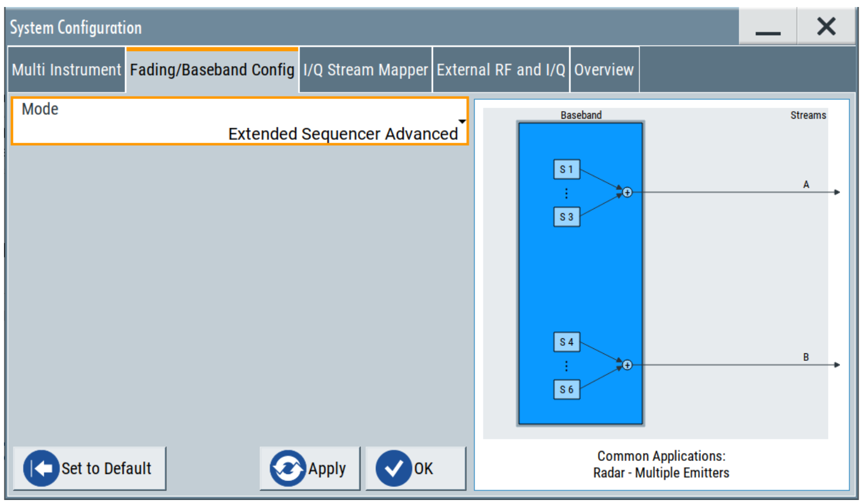

Figure 43: System configuration dialog in the SMW200A

In this advanced mode, streams can be enabled or disabled, each stream can be assigned a waveform list of pre-calculated ARB segments, a frequency-, phase- or level offset and an individual trigger delay. Waveform list handling works as described in section 2.2.

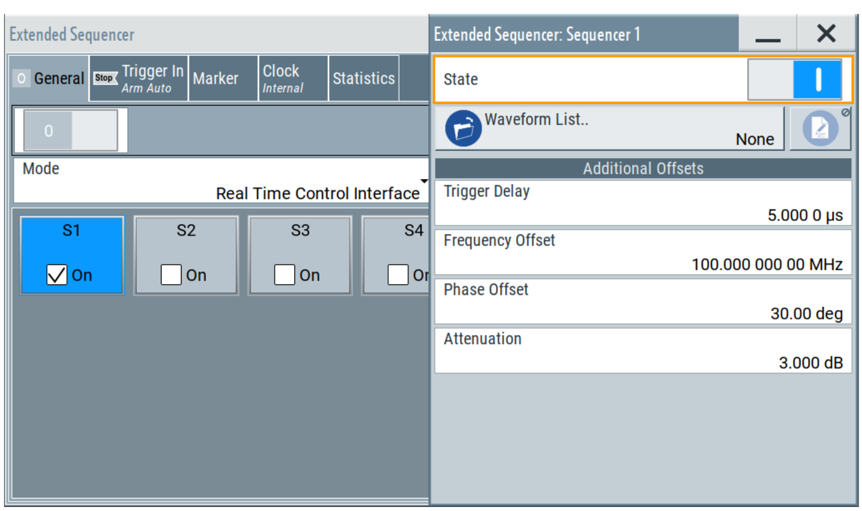

Figure 44: Extended Sequencer Advanced configuration dialog in the SMW200A

In a separate tab, the network interface for each stream can be configured. Thereby, either a static IP address can be assigned or an IP address is requested automatically from a DHCP server.

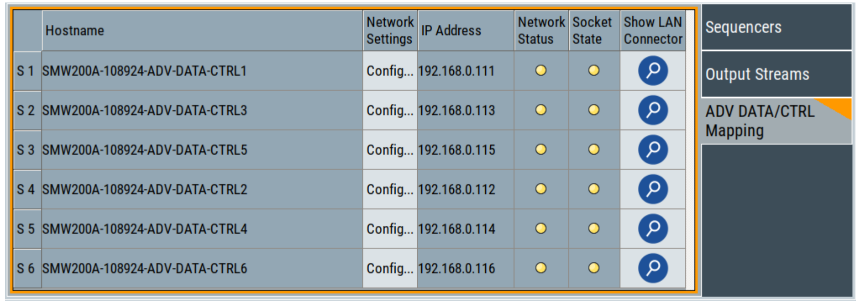

Figure 45: Extended Sequencer Advanced network configuration dialog in the SMW200A

The configured streams are assigned to the RF outputs of the SMW. For a fully equipped SMW, either all six streams are routed to a single RF output or three streams are routed to one and the other three streams to the other output.

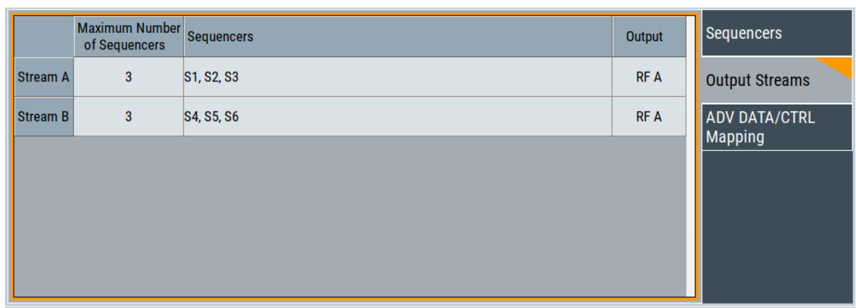

Figure 46: Extended Sequencer Advanced stream configuration dialog in the SMW200A

The level of pulses from the independent I/Q streams at the RF output is influenced by the level setting in the SMW as described in Section 3.4.3. The individual I/Q signal streams are scaled according to the number of added streams. The PEP value is adapted such that no clipping occurs.

- I/Q signals are attenuated by $10\log(n)$ dB in the digital domain, where n is the number of streams routed to the same RF output. As a result the top power of a pulse with 0 dB level offset in the corresponding PDW is $-10\log(n)$ dBm at the RF output (level setting of 0 dBm in the SMW).
- If pulses from multiple different I/Q streams overlap, the level of the sum signal can increase by up to $20\log(n)$ dB. To avoid clipping of the sum signal, the PEP value for the respective RF output in the SMW is set to $10\log(n)$ dBm (which is $20\log(n)$ dB above the level of the individual I/Q signals; level setting of 0 dBm in the SMW).

The following table provides an overview of all possible routings for a SMW with two RF paths, the corresponding attenuation of the I/Q signal and the PEP value for a level setting of 0 dBm for both RF outputs (RF A and RF B) of the SMW:

Number of streams	RF A	RF B	I/Q signal attenuation [dB]	PEP [dBm]
1[2]	1	0	0.00	0.00
2	1	1	0.00	0.00
2[2]	2	0	3.01	3.01
4	2	2	3.01	3.01
4[2]	4	0	6.02	6.02
6	3	3	4.77	4.77
6[2]	6	0	7.78	7.78

[2] Alternatively, all streams can be assigned to RF path B

6 Literature

[1] "Threat Simulation and Verification for Radar Warning Receiver Testing," Rohde & Schwarz, Application Note.

[2] "RWR Testing - Multi-channel Signal Generation (to be published)," Rohde & Schwarz Application Note.

[3] "Pulse Descriptor Word Streaming with the R&S®SMW200A," Rohde & Schwarz, Application Card.

[4] "R&S®SMW-K501/-K502/-K503/-K504/-K315 Extended and Real Time Sequencing, Pulse-on-Pulse Simulation User Manual," Rohde & Schwarz.

[5] "PDW Streaming Interface for the R&S®SMW200A Vector Signal Generator (to be published)," Rohde & Schwarz, Interface Control Document.

[6] "R&S®SMW200A Vector Signal Generator Specifications," Rohde & Schwarz.

[7] "R&S®SMW200A Vector Signal Generator User Manual," Rohde & Schwarz.

7 Ordering Information

Designation	Type	Order No.
Vector Signal Generator	R&S®SMW200A	1412.0000.02
Frequency range 100 kHz to xx GHz for RF path A	R&S®SMW-B10xx	1428.yyyy.02
Frequency range 100 kHz to xx GHz for RF path B	R&S®SMW-B20xx	1428.yyyy.02
Wideband baseband main module, 2 I/Q paths to RF	R&S®SMW-B13XT	1413.8005.02
Wideband baseband generator, 500 MHz, 256 MS	R&S®SMW-B9	1413.7350.02
Fading simulator and signal processor	R&S®SMW-B15	1414.4710.02
Wideband Extended Sequencing for R&S®SMW-B9	R&S®SMW-K502	1413.9260.02
Realtime Control Interface	R&S®SMW-K503	1414.3620.02
High speed upgrade of PDW rate	R&S®SMW-K504	1414.3665.02
Pulse-on-Pulse simulation	R&S®SMW-K315	1414.6529.02

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

1GP124 | Version 1e | 12.2021
Application Note | Generation of Radar Signals in a Hardware in the Loop (HIL) environment
Data without tolerance limits is not binding | Subject to change