Generation of GNSS Signals in a Hardware in the Loop (HIL) Environment

Application Note

Products:

- R&S®SMW200A
- R&S®SMBV100B

The vector signal generators and GNSS simulators R&S®SMW200A and R&S®SMBV100B are remote-controllable in realtime and can therefore be implemented into a dynamic HIL environment. The HIL simulator can dictate position coordinates, kinetic parameters, and vehicle attitude information based upon which the R&S® GNSS simulator updates the simulated receiver movement in realtime.

This application note presents background information and details about operating a R&S® GNSS simulator in HIL applications.

Note:

Please find the most up-to-date document on our homepage http://www.rohde-schwarz.com/appnote/1GP102.

ROHDE&SCHWARZ

Application Note
S. Kehl-Waas

4.2020 – 1GP102_3E

Table of Contents

1 Introductory Note

The abbreviations “SMW” and "SMBVB" are used in this application note for the Rohde & Schwarz products R&S®SMW200A and R&S®SMBV100B.

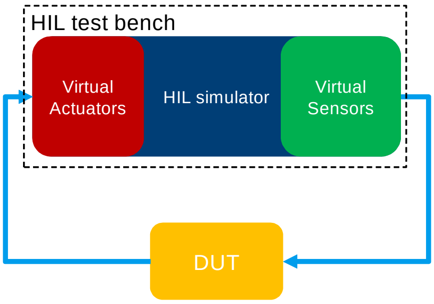
The SMW and SMBVB are general-purpose vector signal generators with outstanding RF performance. They are capable of generating signals for all main communication, radio and avionic standards or to simulate radar systems. Equipped with one or more GNSS options, the SMW and SMBVB can be turned into a full-fledged GNSS simulator for reliable and flexible testing.

2 Overview

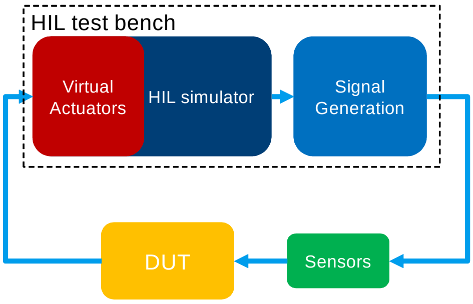
2.1 Hardware in the Loop (HIL)

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 auto-flight control 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.



Test bench with virtual sensors

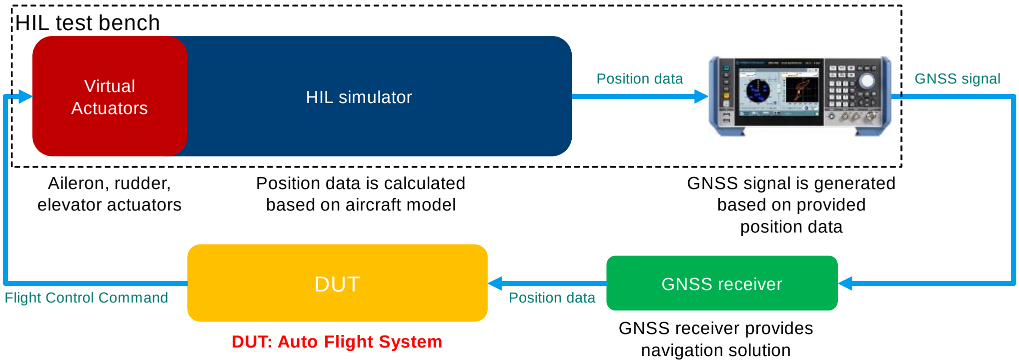


Test bench with real sensors

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

2.2 Realtime Control of R&S® GNSS Simulators

One part in the virtual world of HIL testing is the simulation of GNSS satellite signals. For this purpose a GNSS simulators such as the SMBVB or SMW are used


HIL test bench
Virtual Actuators
HIL simulator
Position data
GNSS signal
Aileron, rudder, elevator actuators
Position data is calculated based on aircraft model
GNSS signal is generated based on provided position data
Flight Control Command
DUT
Position data
GNSS receiver
DUT: Auto Flight System
GNSS receiver provides navigation solution

All R&S GNSS simulators can be remote controlled in realtime and can therefore be integrated into a dynamic HIL environment. The HIL simulator can dictate position data and kinetic parameters such as velocity, acceleration, and jerk while the GNSS simulation is running and the GNSS simulator adjusts the simulated movement accordingly – in realtime and without signal interruptions. Vehicle attitude information such as pitch, roll, and yaw values provided by the HIL simulator is also processed by both the SMBVB and the SMW.

The figure above shows an SMBVB GNSS simulator integrated in a HIL environment to validate the function of an auto flight system. Based on position data from the HIL simulator the SMBVB calculates and provides a GNSS signal to the GNSS receiver. The GNSS receiver determines the navigation solution and forwards it to the DUT, here the auto flight system. The auto flight system calculates a flight control command by comparing the current position versus the position calculated by the flight management system. The flight control commands are directed to the virtual actuators and consequently translated into a position and kinetic parameter change.

This application note describes the required commands for realtime operation in section 3. In section 4, the configuration and control of R&S GNSS simulators is described. Section 5 details synchronization and timing issues. Measures to compensate for command jitter or clock drift are also explained here. Details on the command processing of R&S GNSS simulators, like interpolation, extrapolation and prediction, are provided in section 6. In section 7, several tools and methods to debug a HIL setup are shown.

2.3 Required Options for HIL Operation

The realtime GNSS interface for R&S GNSS simulators comes as SMBVB-K109 for the SMBVB and as SMW-K109 for the SMW. The option, together with at least one GNSS option, allows to feed external vehicle motion data (position, velocity, acceleration and jerk) and attitude data (roll, pitch and yaw) for GNSS signal generation in hardware in the loop (HIL) applications.

The following table lists the GNSS options that are needed at least for HIL operation.

HIL remote control command syntax		
Option	Name	Remark
At least one of the following:		6 satellites
SMW/SMBVB-K44	GPS	
SMW/SMBVB-K66	Galileo	
SMW/SMBVB-K94	Glonass	
SMW/SMBVB-K107	Beidou	
SMW/SMBVB-K109	GNSS Real Time Interfaces	Required for HIL applications

3 Instrument Control

3.1 HIL Remote Control Commands

All R&S GNSS simulators can be controlled remotely via LAN, GPIB and USB. Typically LAN is used in HIL applications. Both, TCP and UDP, are supported

3.1.1 SCPI Position Data

The following remote commands control the R&S GNSS simulator in a HIL test setup. These commands are sent in real-time during the test. The frequency at which a new command is issued in the HIL simulator is referred to as HIL update rate. The SMBVB and the SMW support update rates up to 100Hz (or intervals down to 10ms).

Using SCPI commands, two different modes are supported. In Mode A the position data is related to the Earth Centered Earth Fixed (ECEF) coordinate system, in mode B to the World Geodetic System (WGS84). The user has the choice which notion he prefers. Generally, either mode A or mode B command notation can be used, but no mixture.

Command syntax mode A:

```
[SOUR<hw>]:BB:GNSS:RT:RECeiver[:V<st>]:HILPosition:MODE:A
<ElapsedTime>, <X>, <Y>, <Z>, <XDot>, <YDot>, <ZDot>, <XDotDot>,
<YDotDot>, <ZDotDot>, <XDotDotDot>, <YDotDotDot>, <ZDotDotDot>,
[<Yaw>, <Pitch>, <Roll>, <YawDot>, <PitchDot>, <RollDot>,
<YawDotDot>, <PitchDotDot>, <RollDotDot>, <YawDotDotDot>,
<PitchDotDotDot>, <RollDotDotDot>]
```

HIL remote control command syntax			
Mode A			
Parameter		Description	Unit
Elapsed time	Mandatory	Time elapsed since the start of the simulation	s
X, Y, Z	Mandatory	Position coordinates in the earth centered, earth fixed (ECEF) system	m
XDot, YDot, ZDot	Mandatory	(i.e. vX, vY, vZ) Velocity vector in the ECEF system	m/s
XDotDot, YDotDot, ZDotDot	Mandatory	(i.e. aX, aY, aZ) Acceleration vector in the ECEF system	m/s^2
XDotDotDot, YDotDotDot, ZDotDotDot	Mandatory	(i.e. jX, jY, jZ) Jerk vector in the ECEF system	m/s^3

HIL remote control command syntax			
Mode A			
Parameter		Description	Unit
Yaw, Pitch, Roll	Optional	(i.e. Heading, Elevation, Bank) Attitude angles (Not limited to 2π to simulate more than one cycle rotation between two updates)	rad
YawDot, PitchDot, RollDot	Optional	Attitude angular velocity (i.e. rate of change of attitude angle)	rad/s
YawDotDot, PitchDotDot, RollDotDot	Optional	Attitude angular acceleration (i.e. rate of change of angular velocity)	rad/s^2
YawDotDotDot, PitchDotDotDot, RollDotDotDot	Optional	Attitude angular jerk (i.e. rate of change of angular acceleration)	rad/s^3

Command syntax mode B:

```
[SOUR<hw>]:BB:GNSS:RT:RECeiver[:V<st>]:HILPosition:MODE:B
<ElapsedTime>, <Latitude>, <Longitude>, <Altitude>, <NDot>,
<EDot>, <DDot>, <NDotDot>, <EDotDot>, <DDotDot>, <NDotDotDot>,
<EDotDotDot>, <DDotDotDot>, [<Yaw>, <Pitch>, <Roll>, <YawDot>,
<PitchDot>, <RollDot>, <YawDotDot>, <PitchDotDot>, <RollDotDot>,
<YawDotDotDot>, <PitchDotDotDot>, <RollDotDotDot>]
```

HIL remote control command syntax			
Mode B			
Parameter		Description	Unit
Elapsed time	Mandatory	Time elapsed since the start of the simulation	s
Latitude, Longitude, Altitude	Mandatory	Position coordinates in the WGS 84 system	° and m
NDot, EDot, DDot	Mandatory	(i.e. vN, vE, vD) Velocity vector in the north east down (NED) coordinate system	m/s
NDotDot, EDotDot, DDotDot	Mandatory	(i.e. aN, aE, aD) Acceleration vector in the NED system	m/s^2

HIL remote control command syntax			
Mode B			
Parameter		Description	Unit
NDotDotDot, EDotDotDot, DDotDotDot	Mandatory	(i.e. jN, jE, jD) Jerk vector in the NED system	m/s^3
Yaw, Pitch, Roll	Optional	(i.e. Heading, Elevation, Bank) Attitude angles (Not limited to 2π to simulate more than one cycle rotation between two updates)	rad
YawDot, PitchDot, RollDot	Optional	Attitude angular velocity (i.e. rate of change of attitude angle)	rad/s
YawDotDot, PitchDotDot, RollDotDot	Optional	Attitude angular acceleration (i.e. rate of change of angular velocity)	rad/s^2
YawDotDotDot, PitchDotDotDot, RollDotDotDot	Optional	Attitude angular jerk (i.e. rate of change of angular acceleration)	rad/s^3

3.1.2 UDP Position Data

The structure of UDP position data is described below. Contrary to SCPI commands only ECEF coordinates are supported and all parameters are mandatory.

UDP packet and parameter description for HIL position data				
Data Type	Size	Parameter	Description	Unit
`integer`	4 bytes	`reserve0`	-	-
`integer`	4 bytes	`reserve1`		
`integer`	4 bytes	`reserve2`		
`integer`	4 bytes	`reserve3`		
`double`	8 bytes	`ElapsedTime`	Time elapsed since the start of the simulation	s

`double`
`double`
`double`
	8 bytes
	8 bytes
	8 bytes
		`X`
		`Y`
		`Z`
			Position coordinates in the earth centered, earth fixed (ECEF) system	m

`double`
`double`
`double`
	8 bytes
	8 bytes
	8 bytes
		`XDot`
		`YDot`
		`ZDot`
			(i.e. vX, vY, vZ) Velocity vector in the ECEF system	m/s

UDP packet and parameter description for HIL position data				
Data Type	Size	Parameter	Description	Unit

`double`
`double`
`double`
	8 bytes
	8 bytes
	8 bytes
		`XDotDot`
		`YDotDot`
		`ZDotDot`
			(i.e. aX, aY, aZ) Acceleration vector in the ECEF system	m/s^2

`double`
`double`
`double`
	8 bytes
	8 bytes
	8 bytes
		`XDotDotDot`
		`YDotDotDot`
		`ZDotDotDot`
			(i.e. jX, jY, jZ) Jerk vector in the ECEF system	m/s^3

`double`
`double`
`double`
	8 bytes
	8 bytes
	8 bytes
		`Roll`
		`Pitch`
		`Yaw`
			(i.e. Heading, Elevation, Bank) Attitude angles
			(Not limited to 2π to simulate more than one cycle rotation between two updates)
				rad

`double`
`double`
`double`
	8 bytes
	8 bytes
	8 bytes
		`YawDot`
		`PitchDot`
		`RollDot`
			Attitude angular velocity (i.e. rate of change of attitude angle)	rad/s

`double`
`double`
`double`
	8 bytes
	8 bytes
	8 bytes
		`YawDotDot`
		`PitchDotDot`
		`RollDotDot`
			Attitude angular acceleration (i.e. rate of change of angular velocity)	rad/s^2

`double`
`double`
`double`
	8 bytes
	8 bytes
	8 bytes
		`YawDotDotDot`
		`PitchDotDotDot`
		`RollDotDotDot`
			Attitude angular jerk (i.e. rate of change of angular acceleration)	rad/s^3

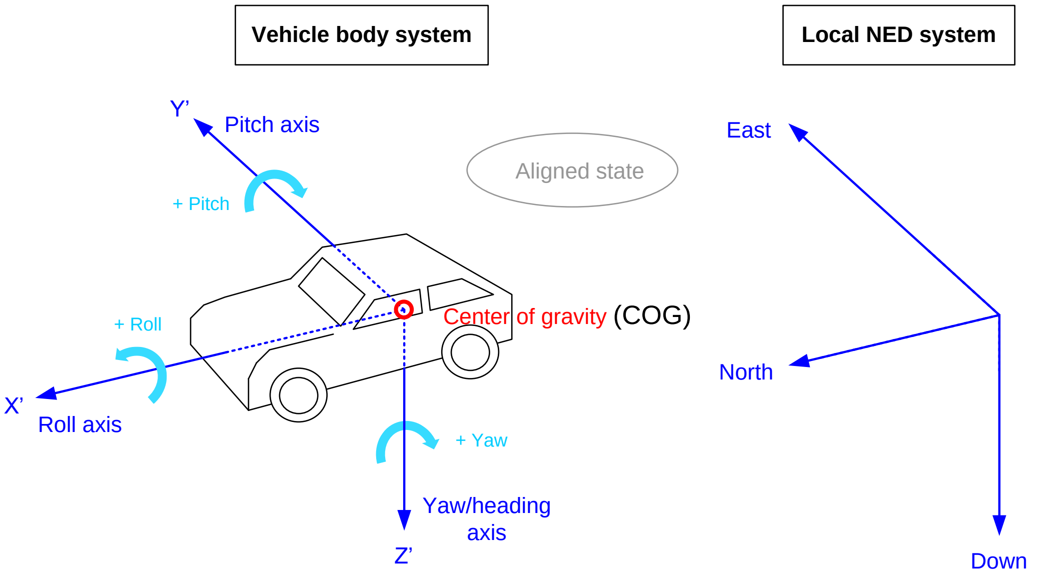
3.1.3 Parameter Description

The “Elapsed time” is the time that has elapsed since the start of the simulation. It can be queried using the command `[:SOUR<hw>]:BB:GNSS:RT:HWT?`

The kinetic parameters (velocity, acceleration, jerk) are mandatory. They are used for trajectory extrapolation and prediction as explained in detail in sections 6.

Using SCPI commands, the attitude parameters are optional. The total number of required parameters is either 13 (if no attitude information is given) or 25 (if attitude information is given). Using UDP position data, always all parameters are needed.

The attitude angles (and their time derivatives) are obtained by rotating the vehicle body system (X’ Y’ Z’) starting from an aligned state with the local NED system by a sequence of three consecutive Euler rotations around the Z’ axis, the Y’ axis and finally the X’ axis.



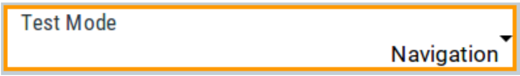
The point of origin of the NED coordinate system is always the local position, i.e. the current position sent in the command (COG).

4 Instrument Configuration

4.1 Initialize HIL Operation

The following settings configure the SMBVB and the SMW for use in a HIL test setup. These settings have to be done once before starting the HIL test, either via SCPI commands or the GNSS GUI.

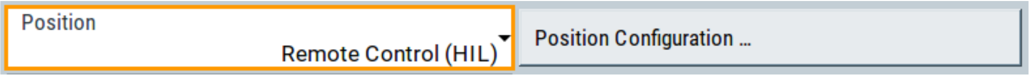
1. Select "GNSS > Test Mode > Navigation"



Test Mode
Navigation

```
[:SOURce<hw>]:BB:GNSS:TMODe NAV
```

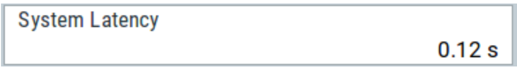
2. Select "Simulation Configuration > Receiver > Remote Control (HIL)"



Position
Remote Control (HIL)

```
[:SOURce<hw>]:BB:GNSS:RECeiver[:V<st>]:POSition HIL
```

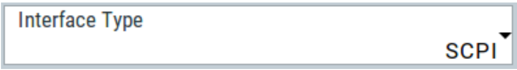
3. Select "Position Configuration" > Enter "System Latency" and "Interface Type"



System Latency
0.12 s

```
[:SOURce<hw>]:BB:GNSS:RECeiver[:V<st>]:HIL:SLATency 0.12
```

How to find the optimum system latency value is described in section 6.

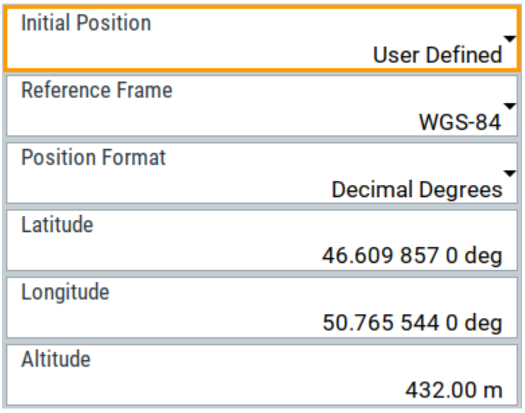


Interface Type
SCPI

```
[:SOURce<hw>]:BB:GNSS:RECeiver[:V<st>]:HIL:ITYPe SCPI
```

As described in the previous section, two interface types are provided for the HIL communication between the HIL simulator and a R&S GNSS simulator: SCPI and UDP

4. It is recommended to define a start position in the "Position Configuration" dialogue that is not too far away from the first HIL waypoint (specified by the first sent HIL command) and wait until a connected GNSS receiver provides a first position fix, before starting to stream HIL commands.



Initial Position: User Defined

Reference Frame: WGS-84

Position Format: Decimal Degrees

Latitude: 46.609 857 0 deg

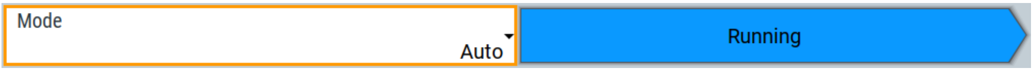
Longitude: 50.765 544 0 deg

Altitude: 432.00 m

```
[:SOURce<hw>]:BB:GNSS:RECeiver[:V<st>]:LOCation:COORdinates:DECimal[:WGS] 50.765544, 46.609857, 432.0
```

Note: Longitude is sent as first parameter in this command, different from the HIL command where latitude is sent first.

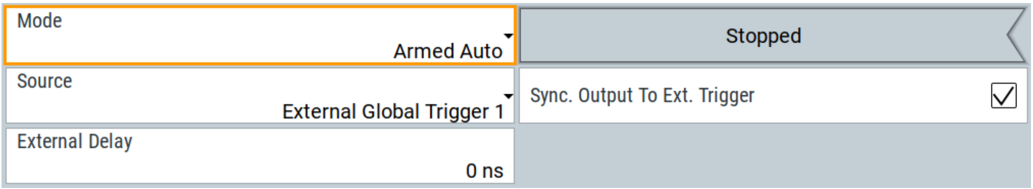
5. Per default, the trigger mode in the R&S GNSS simulator is "Auto"



Mode: Auto

```
[:SOURce<hw>]:BB:GNSS:TRIGger:SEQuence AUTO
```

In this case, the simulation is started as soon as the GNSS standard is turned on. In principle, it is not necessary to change the trigger mode, however it is possible. Instead of "Auto", the trigger modes "Armed Auto" or "Armed Retrigger" can be used for example in combination with an external trigger signal. The trigger signal, e.g. coming from the HIL simulator, is fed to the R&S GNSS simulator via one of the TRIG connectors.



Mode: Armed Auto

Source: External Global Trigger 1

External Delay: 0 ns

Sync. Output To Ext. Trigger ☑

```
[:SOURce<hw>]:BB:GNSS:TRIGger:SEQuence AAUT
[:SOURce<hw>]:BB:GNSS:TRIGger:SOURce EGT1
```

In this case, the satellite simulation is not started immediately as soon as the GNSS standard is turned on, but the R&S GNSS simulator waits for the external trigger. The first trigger event will start the simulation.

6. Switch on the GNSS simulation and the RF output.

Now, the R&S GNSS simulator is ready to process HIL remote control commands and provide the respective GNSS signal.

4.1.1 Example Code

In the previous sections 2 and 3.1 the realtime control commands and the instrument configuration were introduced. Based on this a first "Hello World"-example for the realtime operation of a R&S GNSS simulator is provided:

SCPI (MODE A):

# Reset GNSS simulator and initialize HIL Operation

```
*RST; *OPC?
:SOUR1:BB:GNSS:TMOD NAV; *OPC?
:SOUR1:BB:GNSS:REC:V1:POS HIL; *OPC?
:SOUR1:BB:GNSS:REC:V1:HIL:SLAT 0.12; *OPC?
:SOUR1:BB:GNSS:REC:V1:HIL:ITYP SCPI; *OPC?
:SOUR1:BB:GNSS:REC:V1:LOC:COOR:DEC:WGS 0.0, 0.0, 10.0; *OPC?
```

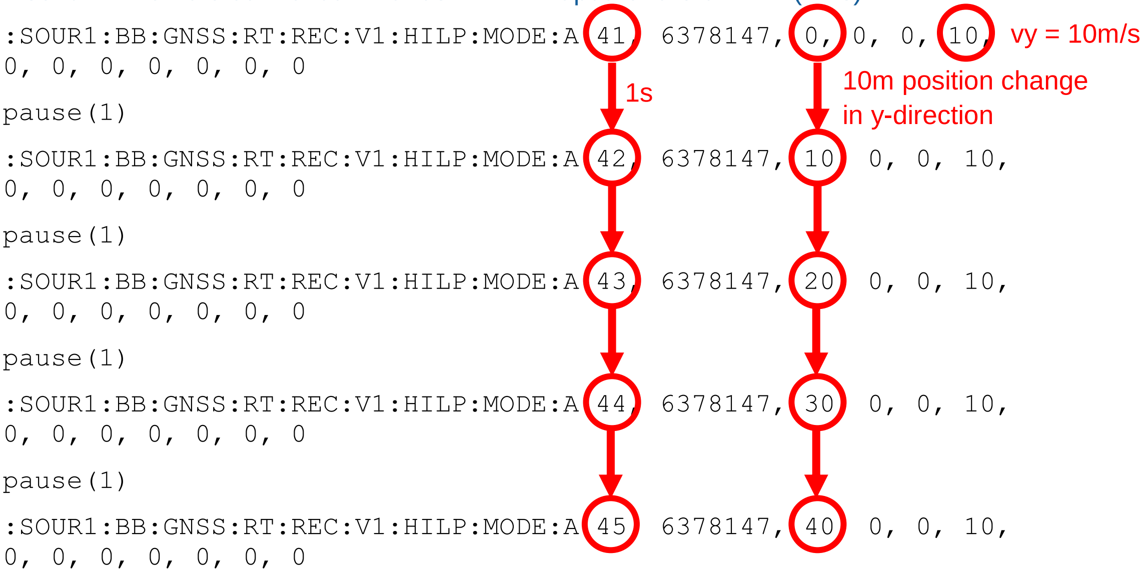
# Switch on GNSS simulation and RF output

```
:SOUR1:BB:GNSS:STAT ON; *OPC?
OUTP:STAT 1; *OPC?
```

# wait until the GNSS receiver performs its first position fix (40s)

```
pause(40)
```

# send HIL remote control commands with HIL update rate of 1 Hz (= 1s)



```
:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 41, 6378147, 0, 0, 0, 10,
0, 0, 0, 0, 0, 0, 0
pause(1)
:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 42, 6378147, 10, 0, 0, 10,
0, 0, 0, 0, 0, 0, 0
pause(1)
:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 43, 6378147, 20, 0, 0, 10,
0, 0, 0, 0, 0, 0, 0
pause(1)
:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 44, 6378147, 30, 0, 0, 10,
0, 0, 0, 0, 0, 0, 0
pause(1)
:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 45, 6378147, 40, 0, 0, 10,
0, 0, 0, 0, 0, 0, 0
```

The simulated trajectory can be observed on the simulation monitor.

Note: The trajectory provided in the GUI is only a schematic representation of the coordinates provided in the HIL remote control commands. Further processing in the signal generator, as described in Section 5, is not considered.

UDP:

```
# Reset GNSS simulator and initialize HIL Operation
*RST; *OPC?
:SOUR1:BB:GNSS:TMOD NAV; *OPC?
:SOUR1:BB:GNSS:REC:V1:POS HIL; *OPC?
:SOUR1:BB:GNSS:REC:V1:HIL:SLAT 0.12; *OPC?
:SOUR1:BB:GNSS:REC:V1:HIL:ITYP UDP; *OPC?
:SOUR1:BB:GNSS:REC:V1:LOC:COOR:DEC:WGS 0.0, 0.0, 10.0; *OPC?
# Switch on GNSS simulation and RF output
:SOUR1:BB:GNSS:STAT ON; *OPC?
OUTP:STAT 1; *OPC?
# connect to device via UDP and send the 'Go To Local (&GTL)' command, so that the
generator is still responsive to SCPI commands
device = connectUDP(ip_address, 7755)
&GTL
# wait until the GNSS receiver performs its first position fix
pause(40)# compose bytestring
reserveByte12 = 0
reserveByte34 = 0
ElapsedTime = 41
X = 6378147
Y = 0
Z = 0
XDot = 0
YDot = 0
ZDot = 0
XDotDot = 0
YDotDot = 0
ZDotDot = 0
XDotDotDot = 0
YDotDotDot = 0
ZDotDotDot = 0
Roll = 0
Pitch = 0
Yaw = 0
RollDot = 0
PitchDot = 0
YawDot = 0
RollDotDot = 0
PitchDotDot = 0
YawDotDot = 0
RollDotDotDot = 0
PitchDotDotDot = 0
YawDotDotDot = 0
```

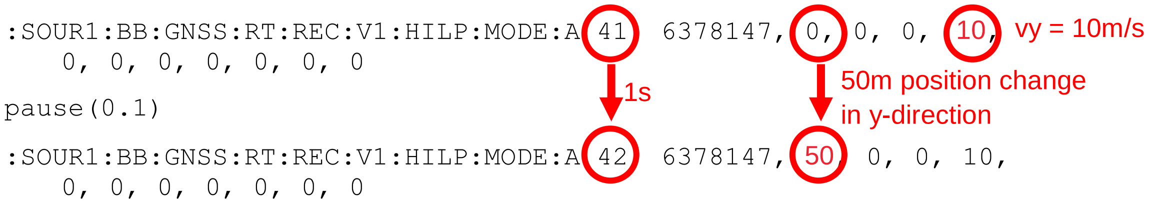
```
# UDP position data is assembled
data = [reserveByte12, reserveByte34, ElapsedTime, X, Y, Z,
XDot, YDot, ZDot, XDotDot, YDotDot, ZDotDot, XDotDotDot,
YDotDotDot, ZDotDotDot, Roll, Pitch, Yaw, RollDot, PitchDot,
YawDot, RollDotDot, PitchDotDot, YawDotDot, RollDotDotDot,
PitchDotDotDot, YawDotDotDot]
waypoint = data.tobytes()
#send UDP position data with HIL update rate of 1 Hz (= 1s)send(waypoint,
device)
pause(1)
#Only ElapsedTime and Y-coordinate change
ElapsedTime = 42
Y = 10
data[2] = ElapsedTime
data[4] = Y
waypoint = data.tobytes()
send(waypoint, device)
pause(1)

ElapsedTime = 43
Y = 20
data[2] = ElapsedTime
data[4] = Y
waypoint = data.tobytes()
send(waypoint, device)
pause(1)

ElapsedTime = 44
Y = 30
data[2] = ElapsedTime
data[4] = Y
waypoint = data.tobytes()
send(waypoint, device)
pause(1)

ElapsedTime = 45
Y = 40
data[2] = ElapsedTime
data[4] = Y
waypoint = data.tobytes()
send(waypoint, device)
pause(1)
```

In the following, examples are provided only for SCPI (Mode A). However, the introduced methods also apply when using UDP position data.

4.1.2 Potential Error Sources

1. Position, velocity, acceleration and jerk parameters are not physically correct

Example:



```
:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 41  6378147, 0, 0, 0, 10,
    0, 0, 0, 0, 0, 0, 0
pause(0.1)
:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 42  6378147, 50, 0, 0, 10,
    0, 0, 0, 0, 0, 0, 0
```

In the first command a velocity of 10 m/s along the y-axis is given. Hence, after 1s the y-coordinate should have increased by 10 m. Instead a command with a y-coordinate, that has increased by 50 m was sent next.

Although, the R&S GNSS simulator will simulate the correct signal corresponding to the HIL remote command, the output signal will show discontinuities in carrier and code phase. The reason for this is that R&S GNSS simulators have an internal update rate of 100Hz.

In the example above, the simulated positions at the internal position updates are as follows (as acceleration and jerk are zero, these values are not shown in the table):

Elapsed Time	X	Y	Z	Vx	Vy	Vz
41	**6378147**	**0**	**0**	**0**	**10**	**0**
41.01	6378147	0.1	0	0	10	0
41.02	6378147	0.2	0	0	10	0
41.03	6378147	0.3	0	0	10	0
41.04	6378147	0.4	0	0	10	0
…						
41.98	6378147	9.8	0	0	10	0
41.99	6378147	9.9	0	0	10	0
42	**6378147**	50	**0**	**0**	**10**	**0**

The parameters in bold are sent from the HIL simulator. The other position updates are calculated in the R&S GNSS simulator based on the kinetic parameter vectors of the first command.

2. Streaming rate too low

Even though R&S GNSS simulators are able to update the simulated position based on the kinetic parameter vector, this is susceptible to abrupt changes in position, velocity, acceleration or jerk, as shown above.

A method to get rid of such discontinuities is to add a time reserve to the system latency parameter. This way, a buffer can be implemented in the GNSS simulator, in which HIL commands are stored for the time specified by the system latency parameter before being applied. If a position, which shall be simulated in the future, is known in advance, the internal position updates can be derived from an interpolation rather than an extrapolation based on the kinetic parameter vector. If at least one future HIL remote control command with the next position is in the buffer, the GNSS simulator performs a linear interpolation from the current position to the future position.

This way it is guaranteed, that the next commanded waypoint is met, and no position jumps occur.

More details on how to realize a buffer are given in chapter 4.2.

3. HIL simulator and GNSS simulator are not synchronized

If the time calculated by the HIL simulator is not identical with the simulation time of the GNSS simulator, the commands are applied either too early or too late. Therefore, it is necessary to synchronize both simulators and also to compensate the latency resulting from transmitting and processing the HIL commands. These topics are covered in the next section.

5 Synchronization & Timing

The first parameter in the HIL remote control command is the elapsed time. This parameter determines the point in time at which the specified position coordinates are in effect. In other words, the GNSS signal corresponding to the specified position coordinates is output at this time point.

The elapsed time T is a relative value and refers to the starting time of the simulation. It can be queried using the command: `[:SOURce<hw>]:BB:GNSS:RT:HWTime?`

(The command `[:SOURce<hw>]:BB:GNSS:TIME:STARt:TIME?` returns the starting time of the simulation.)

Ideally, the elapsed time in the HIL command (T_{SCPI}) should be aligned to the current elapsed time in the GNSS simulator (T_{GNSS}). Ideally, their time difference is smaller than the internal update interval of 10ms. The GNSS simulator can also handle T_{SCPI} values that are less or greater than T_{GNSS}. However, if the difference between T_{SCPI} and T_{GNSS} is very large this can lead to problems in the processing (see Section 5). For this reason, T_{SCPI} and T_{GNSS} should at least be aligned closely. To do so, the HIL simulator and the GNSS simulator have to be synchronized and the latency of the HIL commands resulting from transmitting and processing the commands has to be compensated.

Many HIL simulators require an update latency as short as possible. The internal 100PPS signal limits the minimal update latency of the R&S GNSS simulator to 10 ms. This means position coordinates (waypoints) can be commanded by the HIL simulator at a rate of 100 Hz maximum, i.e. one waypoint per 10 ms cycle. This value can be achieved if the HIL simulator is synchronized to the GNSS simulator. For this purpose R&S GNSS simulators provide a 10PPS or 1PPS signal at its marker output.

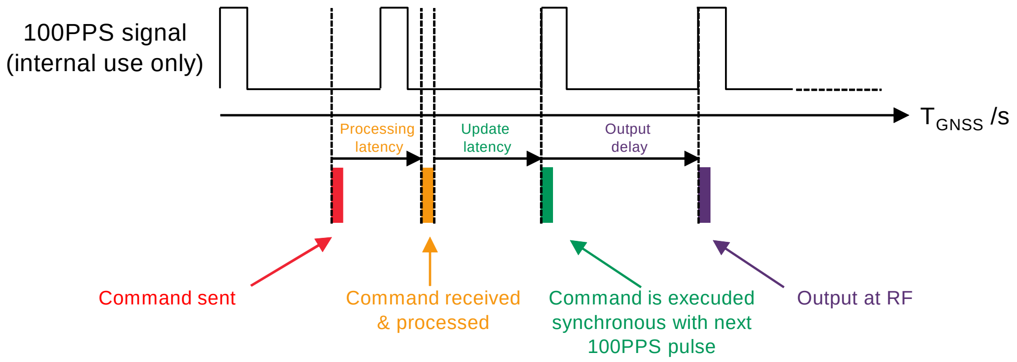
Before more details are provided on time alignment, latency compensation and synchronization in the following a definition of the term 'latency' is given.

5.1 Latency

The term “latency” describes how fast a R&S GNSS simulator can react to an incoming HIL command. Internally, R&S GNSS simulators use a 100PPS signal (100 pulses per second = 100 Hz) that gives a time resolution of 10 ms, i.e. every 10 ms a position update is executed. (The 100PPS signal is not externally available. However, synchronous 10PPS and 1PPS signals are available at the marker outputs of a R&S GNSS simulators.)

When talking about latency it makes sense to distinguish between

- Transmission latency
- Update latency
- Output delay



Processing latency

The processing latency describes the time from sending a HIL command until it is received in the SCPI command input buffer of the R&S GNSS simulator. The processing latency of all R&S GNSS simulators is less than 5 ms, approximately, including the time needed to transmit the HIL command via LAN/GPIB/USB, which is generally less than 1 ms **Error! Reference source not found.**.

Update latency

The update latency describes how fast the R&S GNSS simulator can update the simulated position coordinates according to the input from the HIL simulator. In other words, it describes the time from processing the command until execution at the next 100PPS pulse. The update latency of all R&S GNSS simulators is 10 ms at maximum.

Output delay

The output delay is a constant hardware delay. After the execution of the HIL command, it takes 10 ms until the corresponding RF signal is output at the RF connector of the GNSS simulator.

In general, after approximately 20 ms at the latest the proper position coordinates are simulated.

The latency of one or multiple HIL commands can be evaluated using the HIL latency or HIL statistics command.

The user can query the latency using the commands

```
[:SOURce<hw>]:BB:GNSS:RT:RECeiver[:V<st>]:HILPosition:LATency?
```

and

```
[:SOURce<hw>]:BB:GNSS:RT:RECeiver[:V<st>]:HILPosition:LATency:STATistics.
```

Besides the average latency, the response to the second query also contains the largest latency value and the smallest latency value since the last time a query was sent and the number of non-zero latency values.

The latency values are measured as difference between the time indicated in the SCPI command T_{SCPI} and the time this command is executed in the R&S GNSS simulator T_{GNSS}. Additionally, information on the command processing is returned.

Examples on how to use these commands for optimization and debugging of the test setup are provided in Section 6.

5.2 Time Alignment

The elapsed time in the GNSS simulator is measured relative to the simulation begin, i.e. to the moment where the GNSS simulation is either enabled or triggered. Generally, the time scale of the GNSS simulator does not match the time scale of the HIL simulator. As already mentioned, R&S GNSS simulators can handle small offsets. Nevertheless, it is recommended to keep the difference as small as possible.

A coarse alignment is performed once, in the beginning of the simulation. The alignment can be improved during the simulation by evaluating the latency of the HIL commands. In the following, the procedure is described step-by-step. Additionally, another code example is provided.

5.2.1 Alignment Procedure

1. Configure the R&S GNSS simulator as described in section 3.1
2. Wait until the GNSS receiver performs its first position fix.
3. Retrieve an initial time reference information from the R&S GNSS simulator

   Send the command `[:SOURce<hw>]:BB:GNSS:RT:HWTime?` to query the elapsed time (in seconds with a resolution of 10 ms) from the simulation begin.

   With the response value $T_{GNSS,0}$ the difference between the current time in the GNSS simulator and the HIL simulator can be calculated as

   $\Delta T = T_{GNSS,0} - T_{HIL,0}$,

   where $T_{HIL,0}$ is the current time in the HIL simulator.

   Note: The retrieved value does not consider the round-trip time of the HIL remote control commands, i.e. the latency is non-zero.

Send HIL commands as a function of the time $T_{HIL,i}$, where i stands for the ith timestep

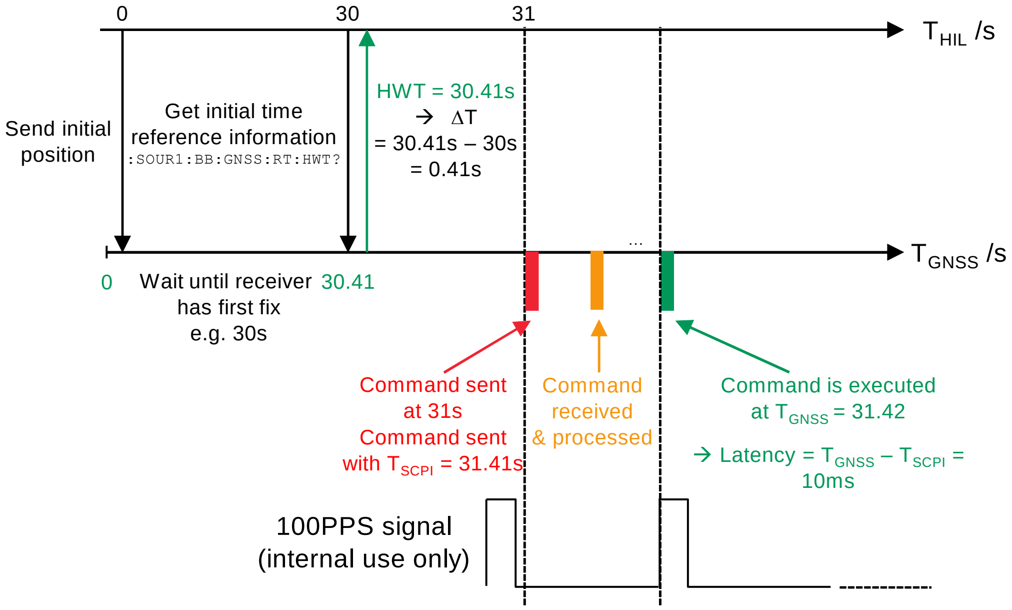
a) The time $T_{SCPI,i}$ is calculated as

   $T_{SCPI,i} = T_{HIL,i} + \Delta T$

b) The position can be used as provided by the HIL simulator

For a coarse synchronization of T_{SCPI} and T_{GNSS} one single query of the elapsed simulation time at the beginning of the HIL simulation is enough. **In fact, it is not recommended to query T_{GNSS} before each HIL command.** The alignment can be improved as part of the latency compensation, which is described in the following.

The procedure is illustrated in the figure below.


0
30
31
T_{HIL} /s
Send initial position
Get initial time reference information
:SOUR1:BB:GNSS:RT:HWT?
HWT = 30.41s
→ ΔT
= 30.41s – 30s
= 0.41s
...
T_{GNSS} /s
0
Wait until receiver has first fix e.g. 30s
30.41
Command sent at 31s
Command sent with T_{SCPI} = 31.41s
Command received & processed
Command is executed at T_{GNSS} = 31.42
→ Latency = T_{GNSS} – T_{SCPI} = 10ms
100PPS signal (internal use only)

5.2.2 Example Code

```
# Reset GNSS simulator and initialize HIL Operation
*RST; *OPC?
:SOUR1:BB:GNSS:TMOD NAV; *OPC?
:SOUR1:BB:GNSS:REC:V1:POS HIL; *OPC?
:SOUR1:BB:GNSS:REC:V1:HIL:SLAT 0.12; *OPC?
:SOUR1:BB:GNSS:REC:V1:HIL:ITYP SCPI; *OPC?
:SOUR1:BB:GNSS:REC:V1:LOC:COOR:DEC:WGS 0.0, 0.0, 10.0; *OPC?
# Switch on GNSS simulation and RF output
:SOUR1:BB:GNSS:STAT ON; *OPC?
OUTP:STAT 1; *OPC?
# wait until the GNSS receiver performs its first position fix
pause(30)
# Retrieve an initial time reference information from the R&S GNSS simulator
TGNSS0 = SOUR1:BB:GNSS:RT:HWT?
# Get HIL simulator time
THIL0 = getTime()
deltaT = TGNSS0 - THIL0
y = 0
# send HIL remote control commands with HIL update rate of 1 Hz (= 1s)
while(1)
```

```
pause(1)
THILi = getTime()TSCPIi = THILi + deltaT
:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A TSCPIi, 6378147, y, 0,
0, 10, 0, 0, 0, 0, 0, 0, 0
y = y + 10
# end while
```

5.3 Latency Compensation

5.3.1 Alignment Procedure

The figure in section 4.2.1 shows that the command with the time stamp T_{SCPI} = 31.41s is actually executed at T_{GNSS} = 31.42s. The reason for this is that the latency was not considered so far. The user can query the latency (in seconds with a resolution of 1 ms) using the commands

```
[:SOURce<hw>]:BB:GNSS:RT:RECeiver[:V<st>]:HILPosition:LATency?
```

or

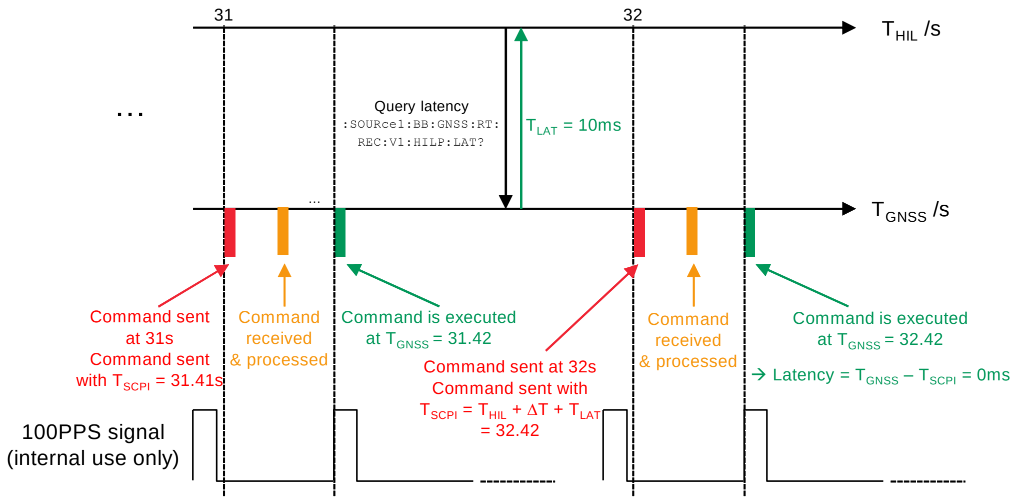
```
[:SOURce<hw>]:BB:GNSS:RT:RECeiver[:V<st>]:HILPosition:LATency:ST
ATistics? .
```

Besides the average latency, the response to the second query also contains some additional statistics on the latency of all HIL commands since the last query.

To compensate the latency, the return value T_{LAT} is added to the elapsed time (T_{SCPI}) indicated in the next HIL command.

$T_{SCPI} = T_{HIL} + \Delta T + T_{LAT}$

For example, if the GNSS simulator's response is +22 ms, T_{SCPI} can be increased by 22 ms such that the time offset is removed. Ideally, the response to the SCPI queries above should be in the range of -10 ms to 0 ms. Again, a single query at the simulation start is generally enough.



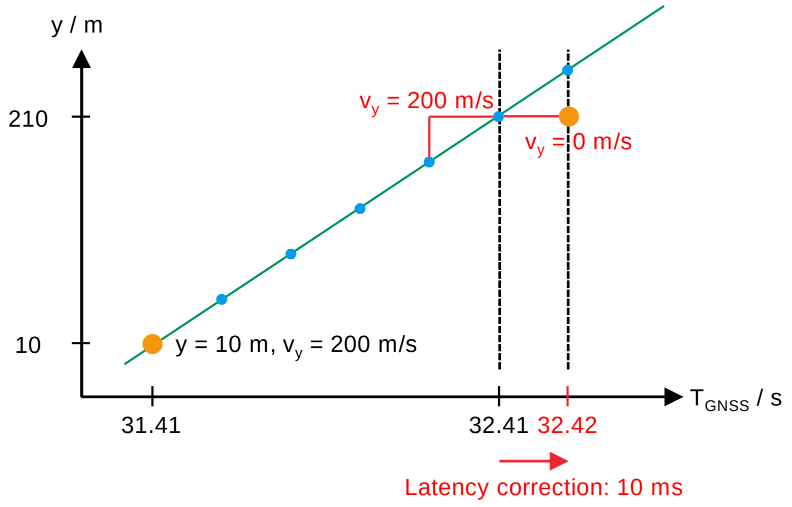
31
32
T_{HIL} /s
...
Query latency
:SOURce1:BB:GNSS:RT:
REC:V1:HILP:LAT?
T_{LAT} = 10ms
...
T_{GNSS} /s
Command sent at 31s
Command sent with T_{SCPI} = 31.41s
Command received & processed
Command is executed at T_{GNSS} = 31.42
Command sent at 32s
Command sent with $T_{SCPI} = T_{HIL} + \Delta T + T_{LAT}$ = 32.42
Command received & processed
Command is executed at T_{GNSS} = 32.42
→ Latency = $T_{GNSS} - T_{SCPI}$ = 0ms
100PPS signal (internal use only)

The commands above can also be used during simulation to cross-check the alignment between T_{SCPI} and T_{GNSS} and, if needed, to adjust T_{SCPI} in the HIL commands.

Re-alignment might be necessary due to

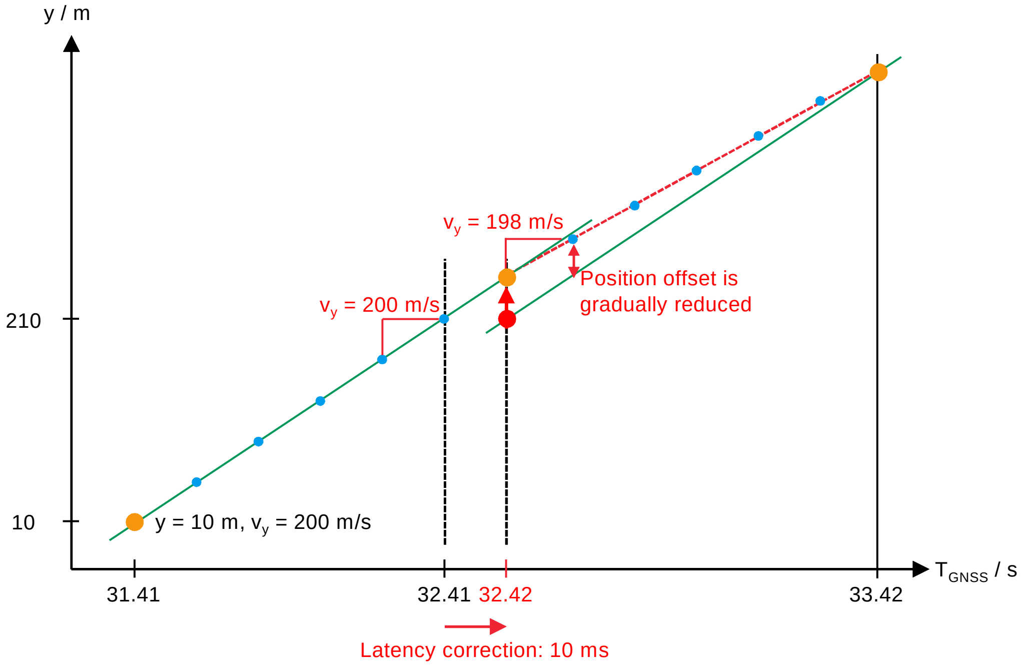
- drift between the clocks of the HIL simulator and the GNSS simulator
- varying traffic in the networks

When performing a re-alignment, in addition to correcting the timestamp in the SCPI command, the position has to be adapted accordingly. Otherwise, there are jumps in position/velocity/acceleration. This is illustrated exemplarily for the y-coordinate in the figure below.



The orange dots mark the HIL commands sent by the user. At T_{GNSS} = 31.41 s a HIL command with an y-coordinate of 10 m and a velocity in y-direction of 200 m/s is executed in the GNSS simulator. Based on this data, the GNSS simulator calculates position updates at its internal rate of 100 Hz (indicated as blue dots). As the response to the latency query returned a non-zero latency, the elapsed time in the subsequent SCPI command is corrected by 10 ms. If the position data is left unchanged, the same position is simulated at T_{GNSS} = 32.41 s and T_{GNSS} = 32.42 s. This corresponds to a velocity in y-direction of 0 m/s. That means, within one position update there is a velocity jump of 200 m/s, which might not be tolerated by the GNSS receiver.

To avoid such jumps, the position and velocity data has to be corrected by the user according to the time adjustment. The position correction should be gradually reduced over several HIL commands until the offset from the original data (generated in the HIL simulator) is zero. This technique is illustrated below.



In this example, the correction of 10ms corresponds to a correction of the y-coordinate by 2m. In order to match the output of the HIL simulator again at T_{GNSS} = 33.42 s the velocity in y-direction has to be set to 198 m/s in the HIL command.

As additional traffic is generated with the query and response, it is recommended to send it not more than twice per second.

In the following, another code example is provided.

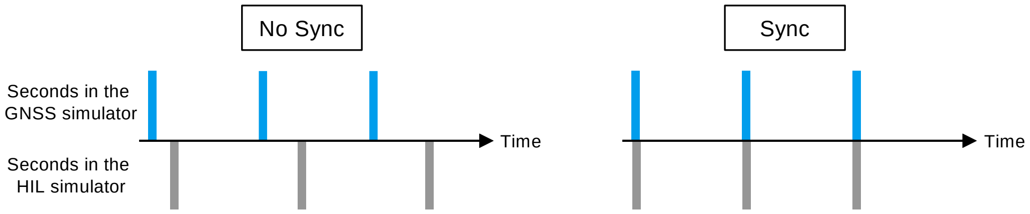
5.3.2 Example Code

```
# Reset GNSS simulator and initialize HIL Operation
*RST; *OPC?
:SOUR1:BB:GNSS:TMOD NAV; *OPC?
:SOUR1:BB:GNSS:REC:V1:POS HIL; *OPC?
:SOUR1:BB:GNSS:REC:V1:HIL:SLAT 0.12; *OPC?
:SOUR1:BB:GNSS:REC:V1:HIL:ITYP SCPI; *OPC?
:SOUR1:BB:GNSS:REC:V1:LOC:COOR:DEC:WGS 0.0, 0.0, 10.0; *OPC?
# Switch on GNSS simulation and RF output
:SOUR1:BB:GNSS:STAT ON; *OPC?
OUTP:STAT 1; *OPC?
# wait until the GNSS receiver performs its first position fix
pause(30)
# Retrieve an initial time reference information from the R&S GNSS simulator
TGNSS0 = SOUR1:BB:GNSS:RT:HWT?
THIL0 = getTime()
```

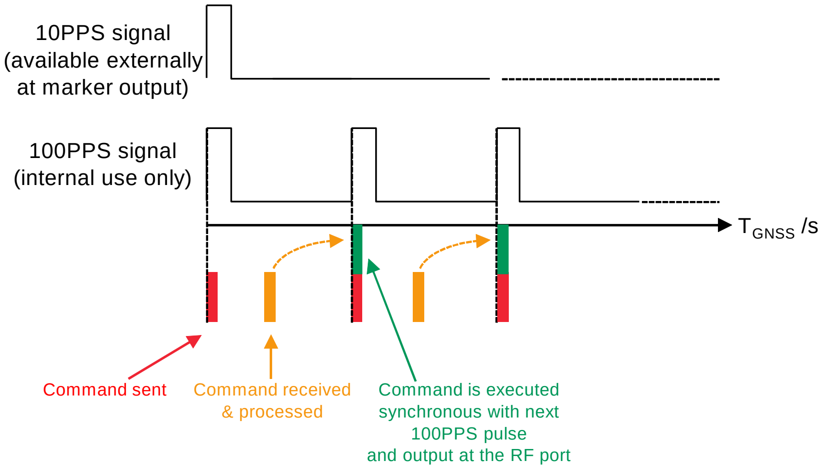
```
deltaT = TGNSS0 - THIL0
TSCPIold = THIL0 + deltaT
y = 0
vy = 200
firstSCPI = 1
counter = 0
# send HIL remote control commands with HIL update rate of 1 Hz (= 1s)
while(1)
    THILi = getTime()TSCPIi = THILi + deltaT
    # calculate new y value based on time difference between two HIL commands
    y = y + (TSCPIi - TSCPIold) * vy
    :SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A TSCPIi, 6378147, y, 0,
    0, vy, 0, 0, 0, 0, 0, 0, 0
    counter += 1
    # send latency query after first HIL command and every 10th command (10 s)
    if firstSCPI or (counter % 10 == 0)
        firstSCPI = 0
        Lat = :SOUR1:BB:GNSS:RT:REC:V1:HILP:LAT?
        # add latency to deltaT
        deltaT += Lat
        # end if
    TSCPIold = TSCPIi
    pause(1)
    # end while
```

5.4 Synchronization with Marker Signal

Some HIL simulators require a latency as short as possible. To reach the minimum of 20 ms, it is required to synchronize the clocks of the HIL simulator and the GNSS simulator. For this purpose, R&S GNSS simulators provide a 10PPS or 1PPS signal at their marker outputs. The HIL simulator can synchronize to this signal such that both units step the seconds synchronously.

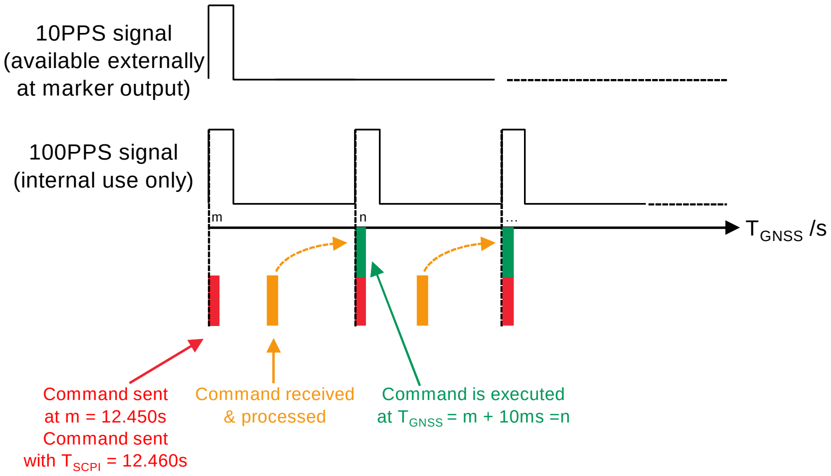


To achieve an update interval of 10 ms, the HIL simulator sends the commands synchronous with the 10PPS signal – each 10 ms a new HIL command. The GNSS simulator needs some time to process the received HIL command. Sending the command synchronous to the internal 100PPS signal gives the GNSS simulator enough time to receive and process the command. Execution takes place at the subsequent 100PPS pulse. At this pulse, the HIL simulator also issues the next command.



5.4.1 Latency Compensation

In this case, the latency compensation becomes a lot easier. The total latency, i.e. the sum of transmission and update latency, is ideally constant with a value of 10 ms. To synchronize T_{SCPI} and T_{GNSS} the HIL command sent at time point T_{GNSS} = m seconds should contain T_{SCPI} = m +10 ms = n. As an example, if m = 12.450 s, then the respective HIL command should contain T_{SCPI} = 12.460 s = n. The specified position coordinates are then simulated at the next 100PPS pulse. Please note that absolute synchronization may not be reached. A small delay (< 5 ms) is acceptable.



5.5 Compensation for Jitter

If the HIL simulator is not equipped with a real-time processor, it can happen that it sends the HIL commands with a varying update rate. This effect is often referred as a command jitter.

R&S GNSS simulators can compensate a maximum command jitter in the range of 30 ms to 120 ms, depending on the update rate. The mechanism is to add a buffer time t_{Buffer} to the "System Latency" value so that the R&S GNSS simulator has enough time to process and realign the HIL commands. The drawback of this mechanism is the adding of an extra constant delay to the system.

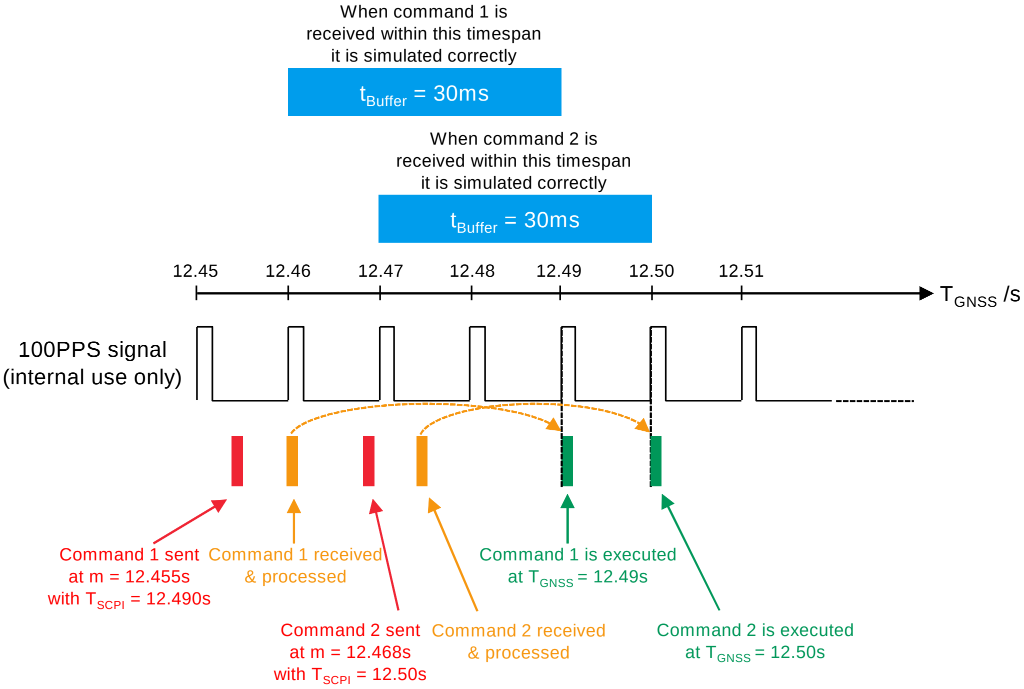
The maximum buffer time can be calculated as

$t_{Buffer,max}$ = System Latency - HIL command interval - min. system latency

E.g.: System Latency = 150 ms, HIL command interval = 20 ms (= 50 Hz), min. system latency = 20ms

$t_{Buffer,max}$ = 150 ms - 20 ms - 20 ms = 110 ms

The mechanism is shown in the figure below.


When command 1 is received within this timespan it is simulated correctly
t_{Buffer} = 30ms
When command 2 is received within this timespan it is simulated correctly
t_{Buffer} = 30ms
12.45
12.46
12.47
12.48
12.49
12.50
12.51
T_{GNSS} /s
100PPS signal (internal use only)
Command 1 sent at m = 12.455s with T_{SCPI} = 12.490s
Command 1 received & processed
Command 1 is executed at T_{GNSS} = 12.49s
Command 2 sent at m = 12.468s with T_{SCPI} = 12.50s
Command 2 received & processed
Command 2 is executed at T_{GNSS} = 12.50s

In the example above, SCPI commands are sent approximately every 10 ms. The buffer size is 30 ms. If a command is received at the R&S GNSS simulator at any time within the time span 30 ms before its designated T_{SCPI}, it is applied correctly. If the command is received earlier than 30 ms before its designated T_{SCPI} it is also applied at its dedicated timestamp, but the SMW switches from interpolation to extrapolation for the intermediate commands. What this exactly means is explained in the following section.

The buffer size can be optimized using the latency statistics command. This procedure is described in section 6.

6 Interpolation, Extrapolation, Prediction

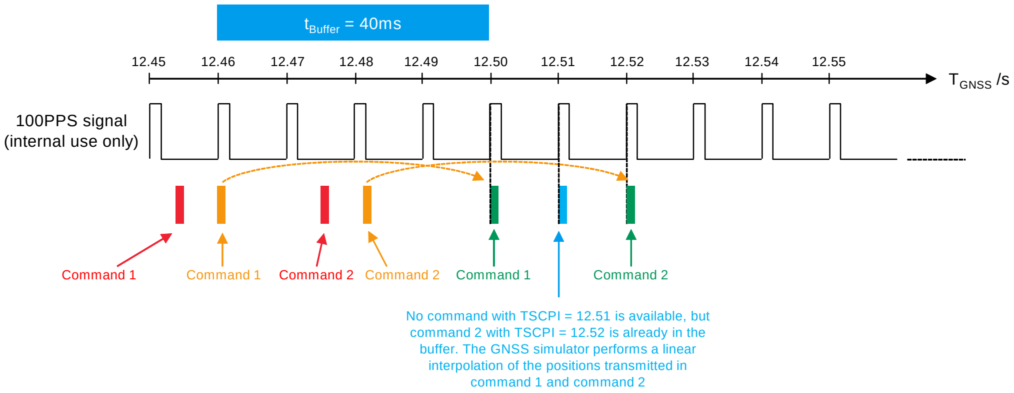
R&S GNSS simulators always work with an internal position update rate of 100Hz. Depending on the chosen HIL command streaming rate, the GNSS simulators performs a linear interpolation between two HIL commands at each internal position update step. If a HIL command is not received in time, e.g. if the command jitter is larger than the buffer, the GNSS simulator extrapolates or predicts the next simulated position from the position and the kinetic parameters of the HIL command received last. The difference between extrapolation and prediction is in the data used for calculation.

Whereas extrapolation describes the process, when the position is calculated from a received command with an old timestamp and is based on the received speed, acceleration and jerk, prediction is applied if no command was received at all. For example, if the update period is larger than 10 ms, subsequent positions are calculated based on the last known speed, acceleration and jerk.

In the following, two examples are given, which shall provide a deeper understanding of the command processing in a R&S GNSS simulator.

Interpolation:

In this example, the HIL command streaming rate is 50 Hz (= 20 ms) and a buffer of 40 ms is used. As R&S GNSS simulators work with an internal rate of 100 Hz, there is no HIL command for every second position update.



t_{Buffer} = 40ms
12.45
12.46
12.47
12.48
12.49
12.50
12.51
12.52
12.53
12.54
12.55
T_{GNSS} /s
100PPS signal (internal use only)
Command 1
Command 1
Command 2
Command 2
Command 1
Command 2
No command with TSCPI = 12.51 is available, but command 2 with TSCPI = 12.52 is already in the buffer. The GNSS simulator performs a linear interpolation of the positions transmitted in command 1 and command 2

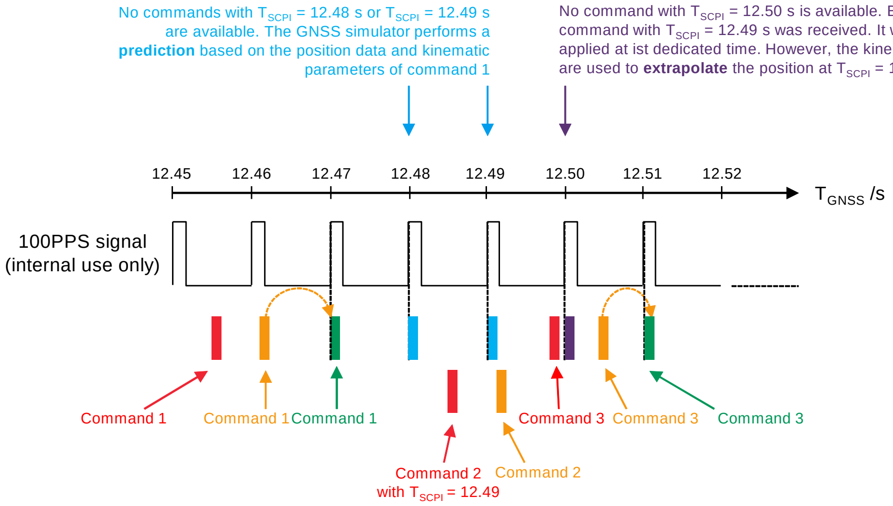
As can be seen in the figure above, a HIL command (command 1) is received and processed at T_{GNSS} = 12.46s. Due to the buffer, it is executed 40 ms later. This is considered with T_{SCPI} contained in the command. Approximately 20 ms after the first command, a second HIL command (command 2) with T_{SCPI} = 12.52s is received and processed.

Once, T_{GNSS} has progressed to T_{SCPI} = 12.50s of command 1, command 1 is executed. The next internal position update is scheduled 10 ms later, at T_{GNSS} = 12.51 s. For this time step, no HIL command with a dedicated timestamp is available. However, the HIL command for the next position update (at T_{GNSS} = 12.52 s) is already in the buffer. The GNSS simulator performs a linear interpolation between the positions transmitted in command 1 and command 2 and simulates the corresponding signal at T_{GNSS} = 12.51 s. At T_{GNSS} = 12.52 s, command 2 is executed.

For the interpolation step, kinetic parameters are not evaluated.

Prediction and extrapolation

In the next example, no additional buffer is used, i.e. t_{Buffer} = 0. The streaming rate is still 50 Hz (= 20 ms).



No commands with T_{SCPI} = 12.48 s or T_{SCPI} = 12.49 s are available. The GNSS simulator performs a **prediction** based on the position data and kinematic parameters of command 1

No command with T_{SCPI} = 12.50 s is available. But the HIL command with T_{SCPI} = 12.49 s was received. It was too late to be applied at ist dedicated time. However, the kinematic parameters are used to **extrapolate** the position at T_{SCPI} = 12.50 s

The figure above shows the reception of three consecutive HIL commands. Command 1 with T_{SCPI} = 12.47 s is received and processed in time to be applied correctly at T_{GNSS} = 12.47 s. For the position update at T_{GNSS} = 12.48 s, no HIL command is available. The GNSS simulator performs a position prediction based on the position data and kinematic parameters of the last HIL command (command 1). Command 2 with T_{SCPI} = 12.49 s is received too late to be applied at T_{GNSS} = 12.49 s. Therefore, the position is also predicted based on command 1. At T_{GNSS} = 12.50 again no HIL command with the appropriate time stamp is available. However, in the meantime a new command (command 2) was received. The position for the update at T_{GNSS} = 12.50 is extrapolated from the parameters in command 2.

The difference between prediction and extrapolation is whether a new command with an old timestamp or no command at all was received. Compared to linear interpolation for both, prediction and extrapolation, all kinetic parameters (i.e. velocity, acceleration and jerk) are used.

7 Tools for optimization and debugging

7.1 HIL Statistics

7.1.1 Structure of HIL Statistics

The response to the HIL statistics SCPI command `:SOURce1:BB:GNSS:RT:RECeiver1:HILPosition:LATency:STATistics?` provides a detailed analysis of the command processing in the GNSS simulator.

Besides information on the last, maximum and minimum latency also statistics on the number of received, interpolated, extrapolated and predicted commands are given. This information can be used to debug and also to optimize a HIL simulation system.

In total, 13 parameters describing the command processing are returned in this order:

1. `<CmdHwTime>`: Time, when response to statistics command is issued

2. `<AvgLatency>`: Average time difference between elapsed time at the GNSS simulator and elapsed time sent in the HIL commands received since the last statistics query at the time of command reception.

3. `<MaxLatency>`: Maximum time difference between elapsed times at the GNSS simulator and elapsed times sent in the HIL commands received since the last statistics query at the time of command reception.

4. `<MinLatency>`: Minimum time difference between elapsed times at the GNSS simulator and elapsed times sent in the HIL commands received since the last statistics query at the time of command reception.

5. `<NoZeroValues>`: Number of commands received, where latency was not equal to zero.

6. `<CmdReceived>`: Number of commands received since the last statistics query

7. `<CmdUsed>`: Number of commands directly used, without interpolation/prediction

8. `<CmdSync>`: Number of commands, which are applied at their specified time (Elapsed time in HiL command)

9. `<CmdExterp>`: Number of extrapolated commands (see manual for more details)

10. `<Interp>`: Number of interpolated commands (see manual for more details)

11. `<Predictions>`: Number of predicted commands (see manual for more details)

12. `<MaxUsed>`: Maximum number of commands in buffer

13. `<MinUsed>`: Minimum number of commands in buffer

A typical response for a HIL setup with a streaming rate of 10 Hz and a statistics query every five seconds, looks like this: 253.32, 0.01, 0.03, 0, 2, 50, 50, 50, 0, 450, 0, 2, 1

The statistics were sent 253.32 s after the simulation start. The latency of the last command was 10 ms, the max. and min. latency since the last statistics query was 30 ms an 0 ms. Only two commands had a latency not equal to zero. In total, 50 commands were received within the five second interval between two statistics queries. These 50 commands were used and applied at their specified time. As the internal update rate is 100Hz, 450 positions had to be interpolated. No positions had to be extrapolated or predicted, i.e. always at least one future command was in the buffer. This is also confirmed by the last parameters, which indicate a maximum of two and a minimum of one commands in the buffer.

In the following some examples on how to interpret the response and take measures to optimize or debug the HIL setup, are provided.

7.1.2 Examples

Example 1

The first simulation is run with the following parameters:

System latency: 20 ms

HIL command interval: 10 ms

Statistics are queried every 5 seconds → 500 commands per statistics interval

1	2	3	4	5	6	7	8	9	10	11	12	13
109.98	0	0.39	-0.01	194	500	374	306	68	0	86	6	0
114.98	0	0.01	-0.01	131	500	451	373	78	0	49	2	0
119.98	0	0	-0.01	8	500	500	500	0	0	0	2	0

This HIL statistics allow several observations of the system:

- The maximum latency (3) is 40ms and thus, larger than the selected system latency of 20ms. This means some commands arrive after the time specified in the time stamp and are not used. The number of commands received (6) and commands used (7) differs.
- As some commands arrive too late the buffer runs empty from time to time (13)
- When the buffer is empty, no set of future coordinates is available for interpolation. In that case, commands have to be extrapolated (9) or predicted (11).

In this scenario, the receiver might suffer from a degraded signal quality, as there may be discontinuities in code and carrier phase.

Measures to improve the HIL setup:

- Increase system latency setting

Example 2

The system latency is now increased to 150 ms. All other settings remain the same. The statistics now look as follows

1	2	3	4	5	6	7	8	9	10	11	12	13
109.99	0.01	0.02	0.01	500	500	500	500	0	0	0	13	11
114.99	0.01	0.02	0.01	500	500	500	500	0	0	0	13	11
119.99	0.01	0.02	0.01	500	500	500	500	0	0	0	13	11

- All received commands (6) are used (7) and applied at their respective time stamp (8)
- No extrapolation (9) or prediction (11) is required
- The number of future commands in the buffer is very high (13). One future command is sufficient for interpolation. The high system latency setting introduces an unnecessary delay in the HIL system.

Measures to improve the HIL setup:

- Reduce the system latency setting, such that at least one future command is in the buffer (approx. 60 ms).

Example 3

In the last example, only the HIL command interval is changed to 100 ms. The system latency is left at 150 ms. Now, within a statistics interval of 5 s, only 50 commands are sent.

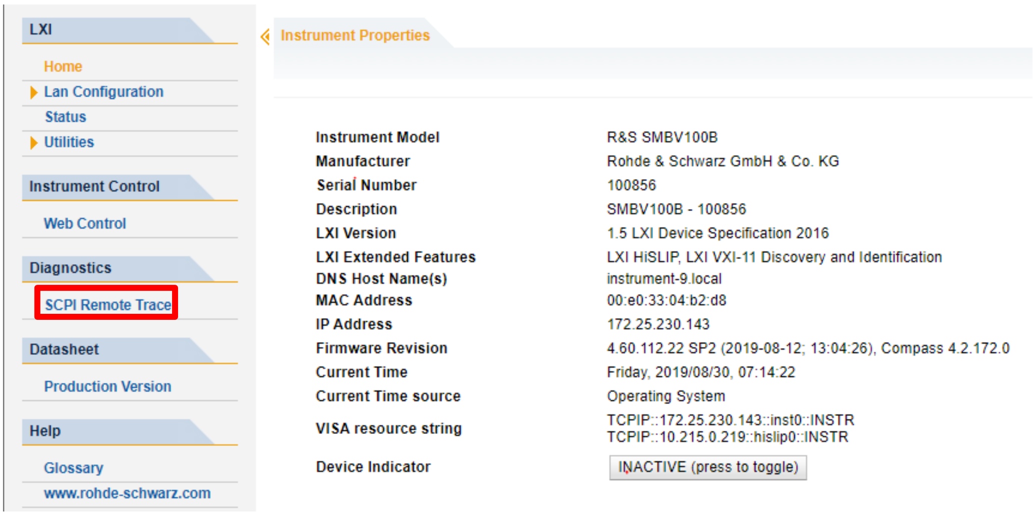
1	2	3	4	5	6	7	8	9	10	11	12	13
109.99	0	0.01	0	27	50	50	50	0	450	0	2	1
114.99	0	0.01	0	5	50	50	50	0	450	0	2	1
119.99	0	0.01	0	15	50	50	50	0	450	0	2	1

- The maximum latency (3) is smaller than the selected system latency.
- All commands are received (6) and used (7) at their respective time stamp (8)
- The HIL command interval is ten times higher than the internal update interval. As always at least one future command is in the buffer (13), the missing 450 commands are interpolated (450)

The system is running stable. No changes are required.

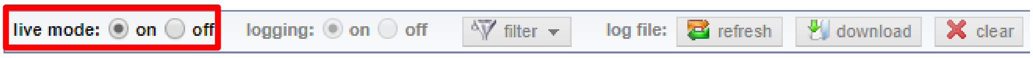
7.2 SCPI Remote Trace

On every R&S GNSS simulator a webserver is installed. With this webserver it is possible to change various settings of the generator, to remotely access the GUI and it also offers a 'SCPI Remote Trace'-tool for diagnostics.

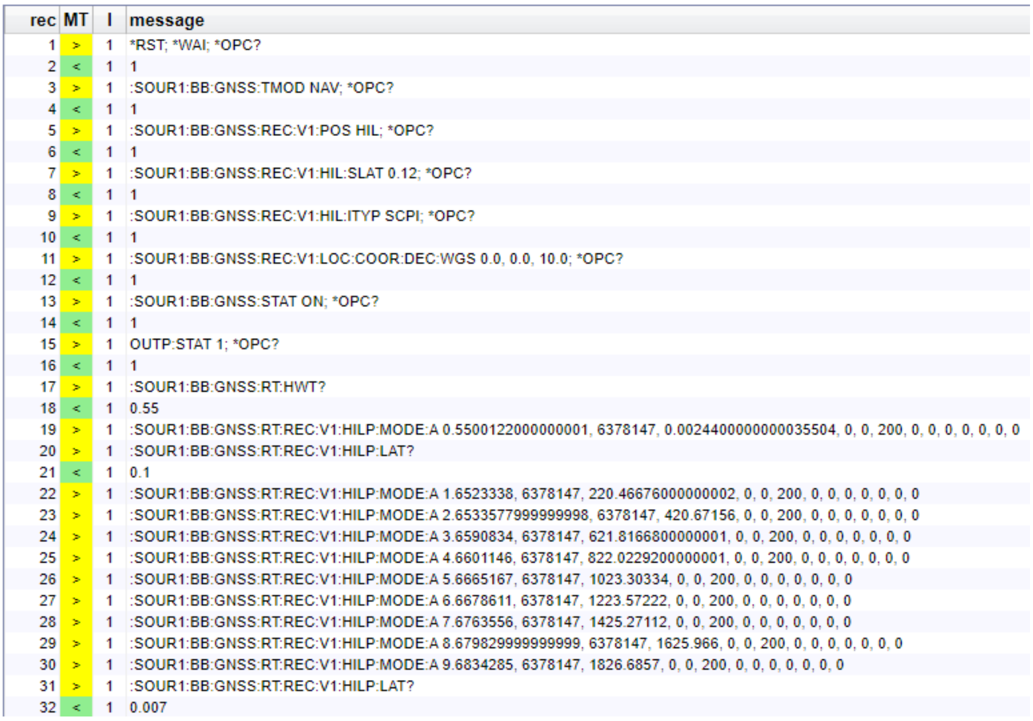


Instrument Model	R&S SMBV100B
Manufacturer	Rohde & Schwarz GmbH & Co. KG
Serial Number	100856
Description	SMBV100B - 100856
LXI Version	1.5 LXI Device Specification 2016
LXI Extended Features	LXI HiSLIP, LXI VXI-11 Discovery and Identification
DNS Host Name(s)	instrument-9.local
MAC Address	00:e0:33:04:b2:d8
IP Address	172.25.230.143
Firmware Revision	4.60.112.22 SP2 (2019-08-12; 13:04:26), Compass 4.2.172.0
Current Time	Friday, 2019/08/30, 07:14:22
Current Time source	Operating System
VISA resource string	TCPIP::172.25.230.143::inst0::INSTR TCPIP::10.215.0.219::hislip0::INSTR
Device Indicator	INACTIVE (press to toggle)

In the top line, the live mode can be enabled to observe, which SCPI commands are sent to the generator, in real-time.



A typical trace for the example in section 4.3.2. looks like this:



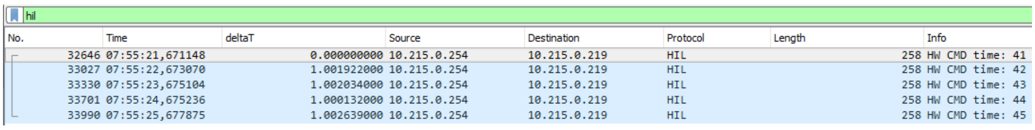
rec	MT	I	message
1	>	1	*RST; *WAI; *OPC?
2	<	1	1
3	>	1	:SOUR1:BB:GNSS:TMOD NAV; *OPC?
4	<	1	1
5	>	1	:SOUR1:BB:GNSS:REC:V1:POS HIL; *OPC?
6	<	1	1
7	>	1	:SOUR1:BB:GNSS:REC:V1:HIL:SLAT 0.12; *OPC?
8	<	1	1
9	>	1	:SOUR1:BB:GNSS:REC:V1:HIL:ITYP SCPI; *OPC?
10	<	1	1
11	>	1	:SOUR1:BB:GNSS:REC:V1:LOC:COOR:DEC:WGS 0.0, 0.0, 10.0; *OPC?
12	<	1	1
13	>	1	:SOUR1:BB:GNSS:STAT ON; *OPC?
14	<	1	1
15	>	1	OUTP:STAT 1; *OPC?
16	<	1	1
17	>	1	:SOUR1:BB:GNSS:RT:HWT?
18	<	1	0.55
19	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 0.5500122000000001, 6378147, 0.0024400000000035504, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
20	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:LAT?
21	<	1	0.1
22	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 1.6523338, 6378147, 220.46676000000002, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
23	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 2.6533577999999998, 6378147, 420.67156, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
24	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 3.6590834, 6378147, 621.8166800000001, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
25	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 4.6601146, 6378147, 822.0229200000001, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
26	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 5.6665167, 6378147, 1023.30334, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
27	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 6.6678611, 6378147, 1223.57222, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
28	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 7.6763556, 6378147, 1425.27112, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
29	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 8.679829999999999, 6378147, 1625.966, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
30	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:MODE:A 9.6834285, 6378147, 1826.6857, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0
31	>	1	:SOUR1:BB:GNSS:RT:REC:V1:HILP:LAT?
32	<	1	0.007

With the filter settings, additional information, like timestamps, can be displayed. Also, only replies of the generator can be shown. This is especially helpful for the analysis of HIL statistics.

7.3 Wireshark

Besides the SCPI Remote Trace, also a tool like Wireshark can be used to track the commands sent to the generator. An advantage of wireshark is, that also UDP commands can be tracked very comfortable. Therefore, a plugin is available together with this application note, which allows to decode the UDP commands into time, position, kinetic and attitude parameters.

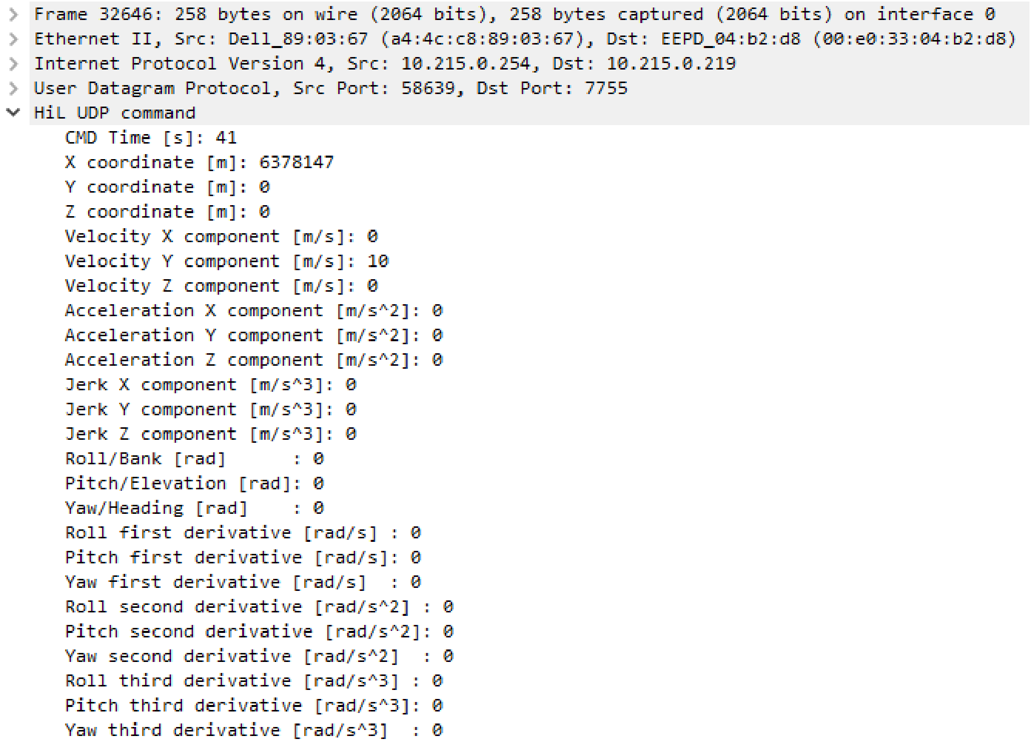
To get rid of all the other traffic, the Wireshark recording is filtered with the display filter 'hil'. Now, only commands that match the definition in the plugin are displayed. For the udp-Example in section 3, this looks as follows:



hil

No.	Time	deltaT	Source	Destination	Protocol	Length	Info
32646	07:55:21,671148	0.000000000	10.215.0.254	10.215.0.219	HIL	258	HW CMD time: 41
33027	07:55:22,673070	1.001922000	10.215.0.254	10.215.0.219	HIL	258	HW CMD time: 42
33330	07:55:23,675104	1.002034000	10.215.0.254	10.215.0.219	HIL	258	HW CMD time: 43
33701	07:55:24,675236	1.000132000	10.215.0.254	10.215.0.219	HIL	258	HW CMD time: 44
33990	07:55:25,677875	1.002639000	10.215.0.254	10.215.0.219	HIL	258	HW CMD time: 45

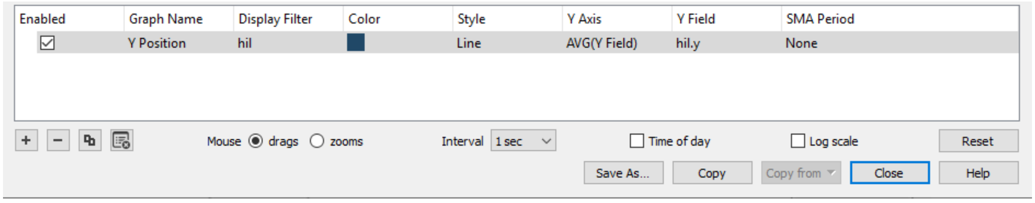
Each command can be analyzed in detail, by selecting the command and opening the layer 'HiL UDP command'.



```
> Frame 32646: 258 bytes on wire (2064 bits), 258 bytes captured (2064 bits) on interface 0
> Ethernet II, Src: Dell_89:03:67 (a4:4c:c8:89:03:67), Dst: EEPD_04:b2:d8 (00:e0:33:04:b2:d8)
> Internet Protocol Version 4, Src: 10.215.0.254, Dst: 10.215.0.219
> User Datagram Protocol, Src Port: 58639, Dst Port: 7755
v HiL UDP command
     CMD Time [s]: 41
     X coordinate [m]: 6378147
     Y coordinate [m]: 0
     Z coordinate [m]: 0
     Velocity X component [m/s]: 0
     Velocity Y component [m/s]: 10
     Velocity Z component [m/s]: 0
     Acceleration X component [m/s^2]: 0
     Acceleration Y component [m/s^2]: 0
     Acceleration Z component [m/s^2]: 0
     Jerk X component [m/s^3]: 0
     Jerk Y component [m/s^3]: 0
     Jerk Z component [m/s^3]: 0
     Roll/Bank [rad]      : 0
     Pitch/Elevation [rad]: 0
     Yaw/Heading [rad]    : 0
     Roll first derivative [rad/s] : 0
     Pitch first derivative [rad/s]: 0
     Yaw first derivative [rad/s]  : 0
     Roll second derivative [rad/s^2] : 0
     Pitch second derivative [rad/s^2]: 0
     Yaw second derivative [rad/s^2]  : 0
     Roll third derivative [rad/s^3] : 0
     Pitch third derivative [rad/s^3]: 0
     Yaw third derivative [rad/s^3]  : 0
```

All parameters transmitted in the command are displayed in plain text. It is also possible to visualize the position data over time. This is very helpful to analyze timing issues in the HIL setup, which cause position jumps. Such position jumps are immediately visible in a graph.

To create a position graph, go to 'Statistics' → 'I/O Graph'. The bottom of the window shows a list of all displayed graphs and their properties:



Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
☑	Y Position	hil	■	Line	AVG(Y Field)	hil.y	None

Mouse ● drags ○ zooms

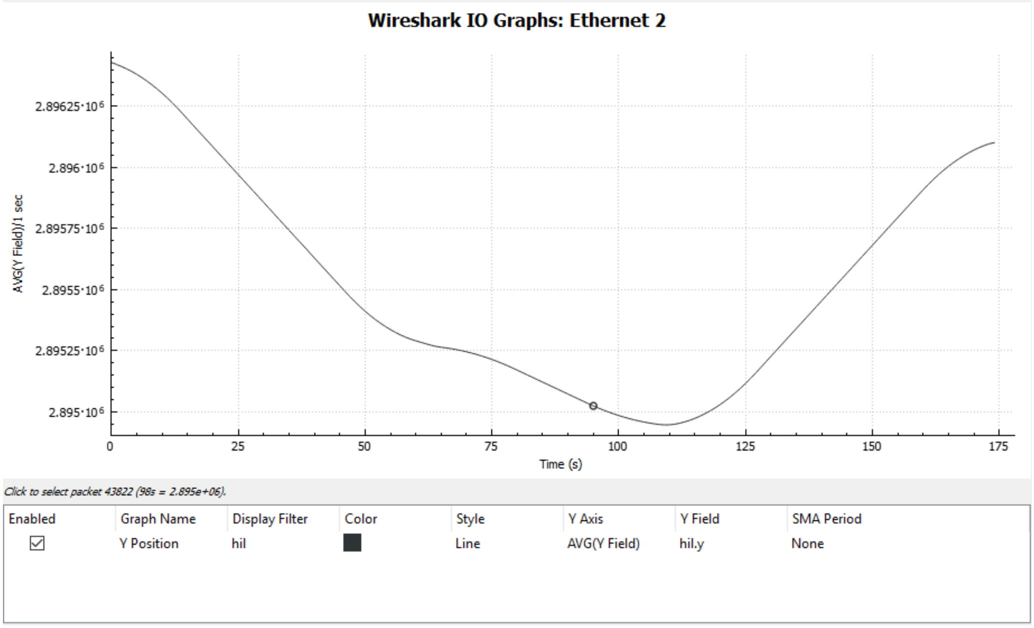
Interval 1 sec

□ Time of day

□ Log scale

Click the “+” button to add another graph. Double click on the fields to edit them. The graph name can be chosen freely, the display filter has to be 'hil'. For the y-axis, “AVG (Y-Field)” should be selected. The y-field specifies the value displayed by the graph. It can be any of the display filter field values. See the readme file of the plugin for the corresponding display filter field names. To get accurate results, the interval has to be adjusted according to your streaming rate.

A typical graph of the y-coordinate could look like this:



Click to select packet 43822 (98s = 2.895e+06).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
☑	Y Position	hil	■	Line	AVG(Y Field)	hil.y	None

8 Summary

The SMBVB and SMW are versatile general-purpose vector signal generators with outstanding RF performance. In addition, they can be turned into powerful GNSS simulators.

R&S® GNSS simulators are remote-controllable in realtime and can therefore be used for HIL testing. The HIL simulator can dictate position coordinates, kinetic parameters, and vehicle attitude information based upon the GNSS simulator updates the simulated receiver movement in realtime. The input data is provided in form of SCPI or UDP commands. An update rate of up to 100 Hz is supported.

R&S® GNSS simulators are ideal for HIL applications due to their low latency and their high command streaming rate. The built-in trajectory prediction feature minimizes or even avoids latency issues in the HIL system.

This application note explained in detail how to control a R&S® GNSS simulator in realtime with SPCI or UDP commands, how to configure it in a HIL environment and how to synchronize it with the HIL simulator. It further provided detailed insight in the command processing in the SMBVB and SMW. Finally, some tools to optimize or debug a HIL setup with built-in and external tools.

9 Abbreviations

1PPS	One pulse per second
10PPS	Ten pulses per second
100PPS	Hundred pulses per second
Beidou	Beidou (global navigation satellite system of China)
DUT	Device under test
ECEF	Earth centered, earth fixed
ECU	Electronic control unit
Galileo	Galileo (global navigation satellite system of the European Union)
Glonass	Globalnaja Nawigazionnaja Sputnikowaja Sistema (global navigation satellite system of the Russian Federation)
GNSS	Global navigation satellite system (stands for all satellite-based navigation systems)
GPS	Global positioning system (of the United States of America)
GPIB	General purpose interface bus
HIL	Hardware in the loop
LAN	Local area network
NED	North east down
RF	Radio frequency
UDP	User datagram protocol
USB	Universal serial bus
SCPI	Standard commands for programmable instruments
T	Elapsed time since simulation start
TCP	Transmission control protocol
WGS	World geodetic system

Appendix

A Interface Description

Parameter		Comment
Internal doppler update interval	10ms	
HIL Update Rate	up to 100 Hz (continuous)	It is recommended to use multiples of the internal position update interval of 10ms, e.g. 20ms = 50Hz
Min. system latency	20 ms	
System latency setting range	20 ms to 150ms	
Max. HIL Command Interval Jitter	30 ms to 120ms	Depends on system latency setting and HIL Update rate. Example: system latency setting: 140 ms HIL update interval: 40 ms Max. HIL Command interval Jitter = 140 ms - 40 ms - 20 ms (min. internal latency) = 80 ms ($t_{Buffer,max}$)
Physical interfaces	LAN, USB, GPIB	
Command format	SCPI, UDP	

Rohde & Schwarz

The Rohde & Schwarz electronics group offers innovative solutions in the following business fields: test and measurement, broadcast and media, secure communications, cybersecurity, radiomonitoring and radiolocation. Founded more than 80 years ago, this independent company has an extensive sales and service network and is present in more than 70 countries.

The electronics group is among the world market leaders in its established business fields. The company is headquartered in Munich, Germany. It also has regional headquarters in Singapore and Columbia, Maryland, USA, to manage its operations in these regions.

Regional contact

Europe, Africa, Middle East
+49 89 4129 12345
customersupport@rohde-schwarz.com

North America
1 888 TEST RSA (1 888 837 87 72)
customer.support@rsa.rohde-schwarz.com

Latin America
+1 410 910 79 88
customersupport.la@rohde-schwarz.com

Asia Pacific
+65 65 13 04 88
customersupport.asia@rohde-schwarz.com

China
+86 800 810 82 28 |+86 400 650 58 96
customersupport.china@rohde-schwarz.com

Sustainable product design

- Environmental compatibility and eco-footprint
- Energy efficiency and low emissions
- Longevity and optimized total cost of ownership

Certified Quality Management
ISO 9001

Certified Environmental Management
ISO 14001

This application note and the supplied programs may only be used subject to the conditions of use set forth in the download area of the Rohde & Schwarz website.

Version 1GP102_3E | R&S®Generation of GNSS Signals in a Hardware in the Loop (HIL) Environment

PAD-T-M: 3573.7380.02/03.00/EN

Rohde & Schwarz GmbH & Co. KG
Mühldorfstraße 15 | 81671 Munich, Germany
Phone + 49 89 4129 - 0 | Fax + 49 89 4129 – 13777

www.rohde-schwarz.com