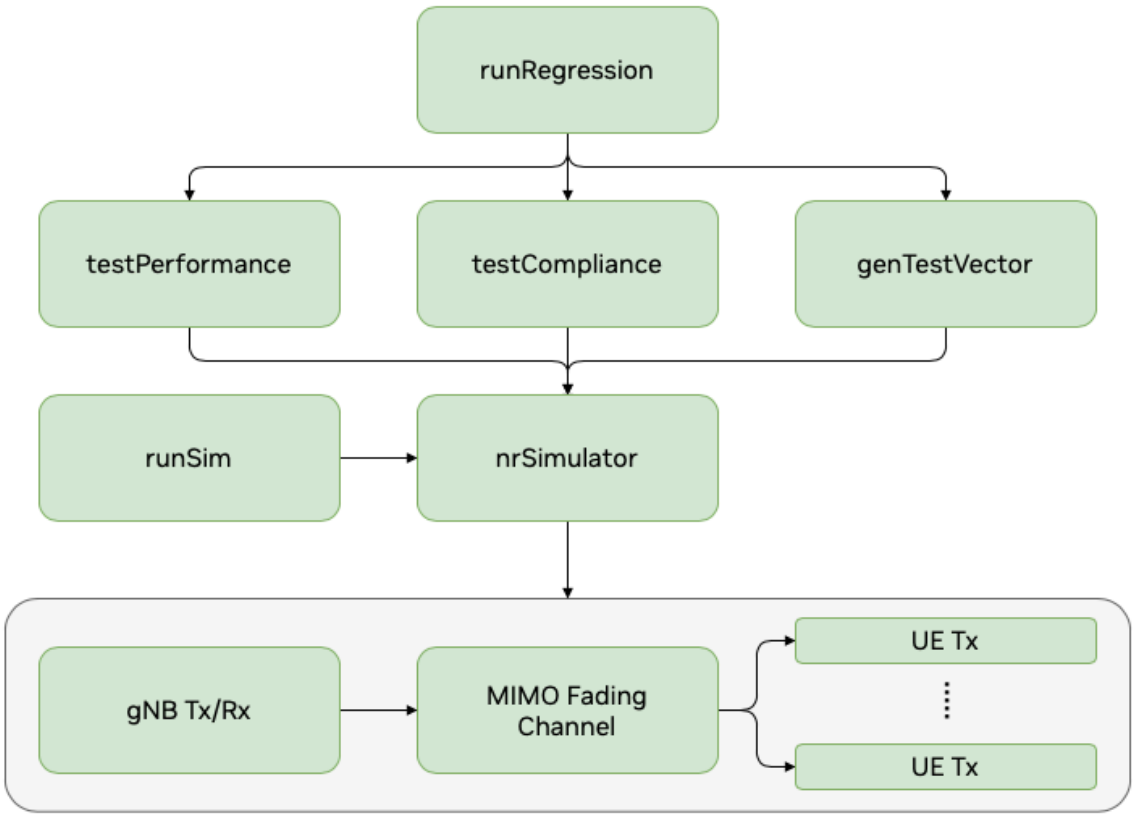
Using 5G Models for Testing and Validation

Table of contents

List of Figures

Aerial CUDA-Accelerated RAN includes a simulation model called nr_sim that is written in Matlab matching with the CUDA implementation in cuPHY library. It can be found under $cuBB_SDK/5GModel/nr_matlab. It serves as a reference model for Aerial design and verification, which covers from L1/L2 FAPI interface to O-DU/O-RU FH interface.

A high level function block diagram of the `nr_sim` is shown in the following figure. The core of nrSim is the simulation engine nrSimulator.m, which includes Matlab models for gNB transmitter and receiver, MIMO fading channel and UE transmitter. nrSimulator.m can be called by runSim.m with external configuration mode or by runRegression.m with internal configuration mode.

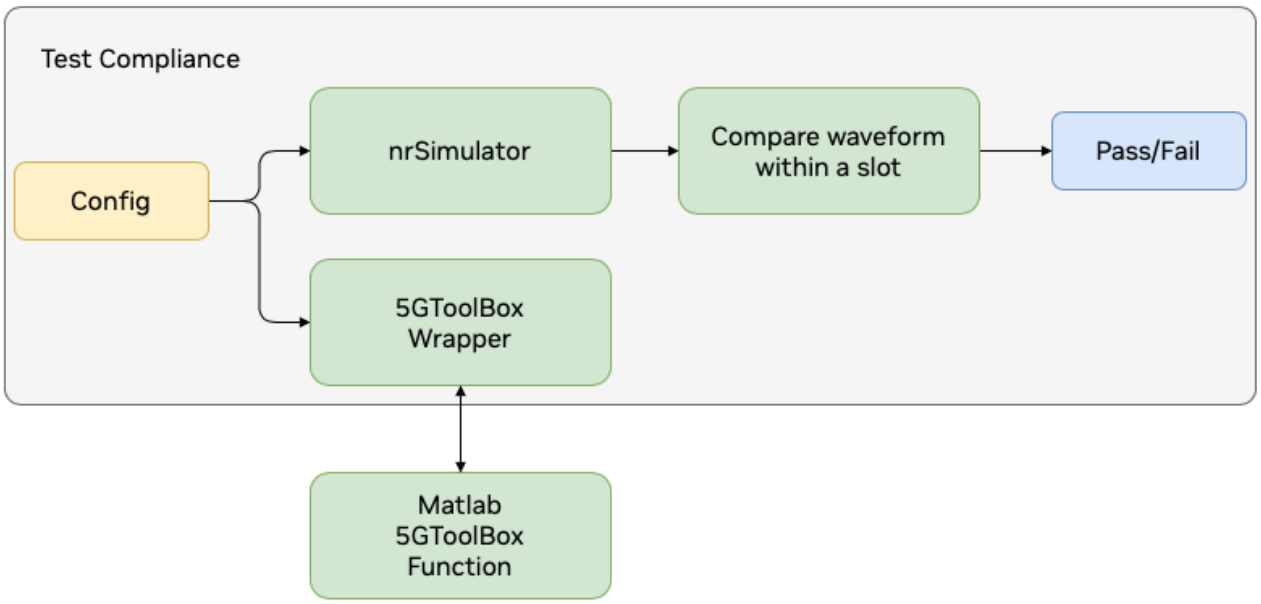


nr_sim functionality

The simulator provides three major features: waveform compliance test, test vector generation and PHY performance simulation.

Waveform compliance test

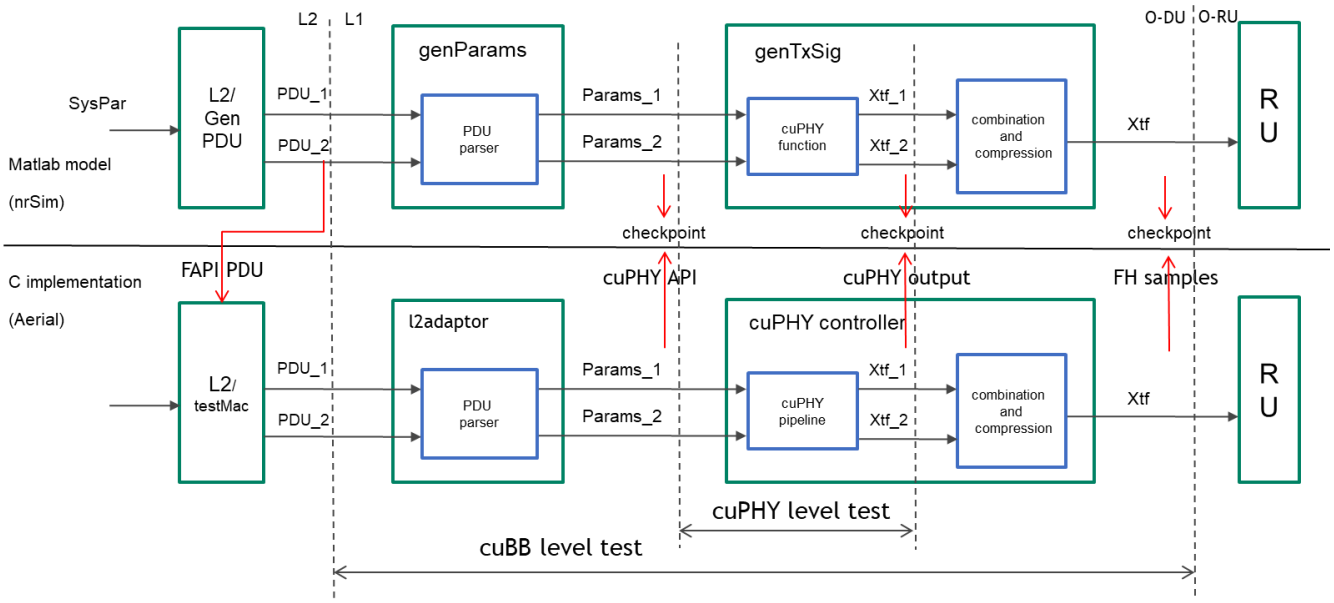
The purpose of waveform compliance test is to make sure our understanding of 3GPP standards regarding signal waveform generation is correct. It is achieved by checking nrSim generated signal against Matlab 5G Toolbox generated signal.



Waveform compliance test

Test Vector Generation

nrSim can generate test vectors for L2/L1 FAPI PDU, cuPHY channel pipeline API parameters, cuPHY channel pipeline output and the compressed samples in a slot.



Test vector generation

Two types of test vectors will be generated for each test case configuration.

- FAPI test vector including FAPI PDU for all the channels in this slot and FH compressed samples for this slot. There is only one FAPI TV per slot.
- cuPHY test vector including cuPHY parameters and input/output for a cuPHY channel pipeline call. There can be multiple cuPHY TVs per slot.

PHY Performance Simulation

The purpose of this test is to make sure that Aerial PHY performance can meet 3GPP requirement by checking nrSim performance simulation results with the same channel condition and test configuration specified by the 3GPP standard.

nrSim Configuration

The input to the simulation engine nrSimulator.m is a single data structure SysPar, which includes all the 3GPP related configurations and simulation control related configurations. The outputs of nrSimulatior include SysPar, UE (array of structures for all UEs) and gNB (structure for gNB).

[SysPar, UE, gNB] = nrSimulator(SysPar)

Matlab functions listed in the table below generate the default configuration for the parameters in SysPar.

Data s tructure	Field	Description	Matlab function for default co nfiguration
SysPar	t estAlloc	Specify DL/UL direction and the number of each type of channels allocated for the slot	initSysPar
	carrier	Specify carrier level configuration	cfgCarrier
	ssb	Specify SSB configuration	cfgSsb
	pdcch	Specify PDCCH channel configuration	cfgPdcch
	pdsch	Specify PDSCH channel configuration	cfgPdsch
	csirs	Specify CSIRS channel configuration	cfgCsirs
	prach	Specify PRACH channel configuration	cfgPrach
	pucch	Specify PUCCH channel configuration	cfgPucch
	pusch	Specify PUSCH channel configuration	cfgPusch
	srs	Specify SRS channel configuration	cfgSrs
	Chan	Specify MIMO fading channel configuration	cfgChan
	SimCtrl	Specify Simulation control parameters	cfgSimCtrl

Configuration options for the testAlloc is summarized in the table below. DL and UL fields indicate if the test is for a DL or an UL slot. The remaining fields hold the number of PHY channel allocations for the test. A given test can include multiple combinations of PHY channels, i.e. 1 SSB allocation, 4 PDCCH allocations, 4 PDSCH allocations, etc.

Data str ucture	Field	Description
tes tAlloc	Dl	Enable DL test
	Ul	Enable UL test
	Ssb	Enable SSB allocation
	Pdcch	Number of PDCCH channels in a slot
	Pdsch	Number of PDSCH channels in a slot

	Csirs	Number of CSIRS channels in a slot
	Prach	Number of PRACH channels in a slot
	Pucch	Number of PUCCH channels in a slot
	Pusch	Number of PUSCH channels in a slot
	Srs	Number of SRS channels in a slot

SysPar definition for 3GPP carrier and slot configuration with each channel is mostly based on SCF-FAPI specification.

The Chan configuration refers to MIMO fading channel model.

Data structure	Field	Description
Chan	Type	'AWGN', 'TDLx-xx-xxx' (3GPP MIMO fading channel)
	SNR	Channel SNR in dB
	Delay	Channel propagation delay in second
	CFO	Carrier frequency offset in Hz
	Use5Gtoolbox	Reserved
	gain	Reserved

The SimCtrl structure includes global configuration settings that are used in the simulation.

Data structure	Field	Sub-field	Description
Si mCtrl	N_UE		Number of UEs
	N_frame		Number of frames per run
	N_slot_run		Number of slots in a frame to run. (0: run all slots in a frame)
	tim eDomainSim		Enable time domain simulation (required for applying fading channel model, delay and CFO)

	plotFigure	tfGrid	Plot time/freq domain signal
		constellation	Plot constellation before and after equalizer
	genTV	Enable	Enable TV generation at gNB side
		enableUE	Enable TV generation at UE side
		tvDirName	Name for TV directory
		cuPHY	Enable cuPHY TV in h5 format
		FAPI	Enable FAPI TV in h5 format
		FAPIyaml	Enable FAPI TV in yaml format
		slotIdx	Indices of slots on which TV will be generated
		for ceSlotIdxFlag	Force slot index = slotIdx(1) for every slot
		bypassComp	Bypass FH sample compression
		idx	Reserved
		TVname	Prefix for the name of TVs
		fp16AlgoSel	0: Use half function (Matlab fixed point toolbox required) 1: Use vpf16 function (Matlab fixed point toolbox not required)
		CFOflag	Enable CFO correction
		e nableRssiMeas	Enable RSSI measurement
	capSamp		Reserved
	result		Reserved

nrSim Usage

For different test and simulation purpose, nrSim provides two modes to change the configurations and run the Matlab model.

- External configuration mode (runSim): This mode is to use an external configuration file in yaml format to update the parameters. nrSim will read this yaml configuration file and set the SysPar parameters accordingly. It is recommended that non matlab

developer uses this mode to generate test vectors which requires no change to the Matlab code.

- Internal configuration mode (runRegression): This mode is to change the SysPar parameters directly in the Matlab code between initSysPar and nrSimulator. Matlab developer can pre-define a set of configuration used by compliance test, test vector generation and PHY performance simulation. Multiple runs can be performed in this mode with different configurations.

Matlab Environment Preparation

Matlab version:

- R2020a or later

Matlab licenses:

- MATLAB
- Communications Toolbox
- DSP System Toolbox
- Signal Processing Toolbox
- Fixed-Point Designer (optional)
    - Call half function to accelerate testing/simulation
    - Can be disabled by setting SimCtrl.fp16AlgoSel = 1
- Parallel Computing Toolbox (optional)
    - Accelerate testing/simulation automatically
- 5G Toolbox (optional)
    - Not required for TV generation
    - Required for waveform compliance test and performance simulation

Preparation:

- After download the source code, launch Matlab on the directory of nr_matlab and run startup to add all sub-directories into Matlab search path.

External Configuration Mode (runSim)

1. Find the yaml configuration template file cfg_template.yaml under nr_matlab. If it is missing, run genCfgTemplate to generate it.

2. Use a text editor to change the parameters in the yaml file. Basically cfg_template.yaml is a yaml (text) version of SysPar data structure. Please refer to section 3 for the description of SysPar parameters. After change is done, save it to another file name, for example, cfg_test.yaml.

3. Run runSim(cfg_filename, tv_filename), for example, runSim('cfg_test.yaml', 'my_test'). nrSim will read cfg_test.yaml file, update SysPar accordingly, run nrSimulator and generate test vector files with name starting with my_test. The generated TV files are stored under the folder named by SysPar.SimCtrl. tvDirName, for example, GPU_test_input.

4. Another option is to use runSim(cfg_filename, 'test',tv_filename),

Notes:

- This mode only supports test vector generation with SimCtrl.genTV.enable set to 1. It does not support waveform compliance test and PHY performance test.

- If SimCtrl.plotFigure.tfGrid is set to 1, the time/freq signal in a frame or the specified number of slots in a frame (controlled by N_slot_run) can be plotted to provide visualized channel allocations.

- Non Matlab developer can write script in any language to modify the yaml template file and automatically generate a number of different yaml configuration files for different testing purpose.

Internal Configuration Mode (runRegression)

Instead of updating configuration through the external yaml configuration file case by case, the internal configuration mode changes SysPar parameters directly inside the Matlab code, which allows Matlab developer to define and execute a batch of test cases more efficiently. The main function for this mode is runRegression, which supports a flexible combination of testSet, channelSet and caseSet as the input arguments.

runRegression(testSet, channelSet, caseSet)

	Values	Value selection	
testSet	‘Compliance’, ‘TestVector’, ‘Performance’, ‘allTests’	Multiple	
channelSet	‘ssb’, ‘pdcch’, ‘pdsch’, ‘csirs’, ‘dlmix’, ‘allDL’, ‘prach’, ‘pucch’, ‘pusch’, ‘srs’, ‘ulmix’, ‘allUL’, ‘allChannels’	Multiple	
caseSet	‘full’, ‘compact’, ‘selected’	Single	

Here are some example commands.

- Full regression test for all channels

runRegression({‘allTests’}, {‘allChannels’}, ‘full’)

- Waveform compliance test and test vector generation for pdcch and pdsch channels with compact set

runRegression({‘Compliance’, ‘TestVector’}, {‘pdcch’, ‘pdsch’}, ‘compact’)

- PHY performance simulation for PRACH channel

runRegression({‘Performance’}, {‘prach’}, ‘full’)

The test cases for each channel are defined in the Matlab file testCompGenTV_xxxx.m, where xxxx is the channel name. Matlab developer can modify the Matlab file to create and assign test cases for full set, compact set and selected set.

- full set includes all the test cases which can be generated by nrSim and pass waveform compliance test against 5G Toolbox.
- compact set includes a subset of full set test cases which are supported by cuPHY implementation. TVs from Compact set can be used for nightly CICD regression test.

- selected set includes a subset of compact set test cases which are essential for cuPHY verification. TVs from Selected set can be used for merge request (MR) CICD regression test.

Notes:

- testCompGenTV_dlmix and testCompGenTV_ulmix supports multi-channel multi-slot TV generation without waveform compliance check.

- testPerformance_prach, testPerformance_pusch and testPerformance_pucch support PHY performance test for PRACH (format 0/B4), PUSCH (non-UCI) and PUCCH (format 0/1).

Below is an example of full regression test summary with Matlab command

runRegression({'allTests'}, {'allChannels'}, 'full')

```
Channel  Compliance_Test  Error   Test_Vector  Error  Performance_Test   Fail
-------------------------------------------------------------------------------
SSB            15             0          15          0             0             0
PDCCH          47             0          47          0             0             0
PDSCH         222             0         222          0             0             0
CSIRS          55             0          55          0             0             0
DLMIX           0             0          16          0             0             0
PRACH          20             0          20          0            48             0
PUCCH         110             0         110          0            60             0
PUSCH         199             0         199          0            32             0
SRS             2             0           2          0             0             0
ULMIX           0             0           6          0             0             0
-------------------------------------------------------------------------------
Total         670             0         692          0           140             0
Elapsed time is 1221.657852 seconds.
```

An example output of a full regression test summary