5G NR Radar

Steve Blandino

National Institute of
Standards and Technology
U.S. Department of Commerce

WIRELESS NETWORKS DIVISION, COMMUNICATION TECHNOLOGY LAB

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY,

GAITHERSBURG, MD, USA

JUNE, 2025

Licensing Statement

5G NR Radar Documentation

5G New Radio Radar (5G NRad) is a MATLAB-based simulation framework that models the radar sensing capabilities of a 5G New Radio (NR) base station, as presented in:

> *"Detecting Airborne Objects with 5G NR Radars"*, S. Blandino et al., (submitted to) IEEE MILCOM 2025.

5G NRad implements a monostatic sensing configuration in which a single 5G NR base station (BS) functions as both transmitter and receiver to detect and localize airborne Sensing Targets (STs) within its coverage area. The BS transmits standardized 5G NR communication signals and processes the received echoes scattered by targets in the scene. Target position is estimated by extracting the angle of arrival (AoA) and round-trip delay from the received signals.

To generate sensing observations, 5G NRad uses the Positioning Reference Signal (PRS) defined in the 5G NR standard TS 38.211. The platform supports realistic deployment scenarios, including Urban Micro (UMi) and Urban Macro (UMa), and integrates 3GPP-compliant models for path loss, shadow fading, and line-of-sight (LoS) probability based on TR 38.901 and TR 36.777. In addition, 5G NRad incorporates recent 3GPP developments related to Release 19, including updated radar cross-section (RCS) models for airborne targets and monostatic background clutter synthesis.

Key Features:

- End-to-end radar processing chain: PRS generation, Orthogonal Frequency-Division Multiplexing (OFDM) transmission, analog beamforming, clutter suppression, and target detection.
- 3GPP-based environmental modeling: supports Line-of-Sight (LoS) / Non-Line-of-Sight (NLos) propagation, shadowing, fading, and multipath.
- Range-Doppler processing based detection.
- Evaluation support for both UMi and UMa deployment scenarios with configurable Unmanned Aerial Vehicle (UAV) altitudes.

5G NRad is developed in MATLAB and tested on MATLAB R2024b. It requires the MATLAB 5G Toolbox for full functionality.

1 Folder Structure

```
./
|- main.m
|- examples/
   |- UMi-Av25/
      |- Input/
         |- bsConfig.txt
         |- prsConfig.txt
         |- sensConfig.txt
         |- simulationConfig.txt
         |- targetConfig.txt
|- channel/
   |- backgroundChannel_6GHz_3GPP_38.901_UMa_NLOS.json
   |- targetChannel_6GHz_UMaAV.json
|- src/
```

The repository is structured as follows:

- `main.m`: Main script and entry point to run the simulation.
- `src/`: Contains all MATLAB source code files implementing the 5G NR radar simulation functionalities, including PRS generation, channel modeling, beamforming, and target detection algorithms.
- `backgroundChannel/`: Includes pre-generated JavaScript Object Notation (JSON) files representing background channel environments. These files are generated offline according to channel modeling methodologies developed by the 3GPP Release 19 working group.
- `examples/`: Contains predefined example scenarios illustrating how to configure and run the software.
  - `UMi-Av25/Input/`: Example input configuration files for the Urban Micro (UMi) scenario with UAV altitude of 25 m. The configuration includes:
    - `bsConfig.txt`: Base station position.
    - `prsConfig.txt`: Positioning Reference Signal (PRS) parameters.
    - `sensConfig.txt`: Sensing and detection algorithm parameters.
    - `simulationConfig.txt`: General simulation scenario settings.
    - `targetConfig.txt`: Parameters defining sensing target (ST) position and velocity

2 Configuration Files

5G NRad configuration is specified through five configuration files located within each scenario's `Input` directory. These files define the radar system parameters, scenario settings, sensing configuration, and PRS settings. Each parameter's allowable range and default values (if applicable) are described below.

2.1 Configuration Files Description

5G NRad configuration is specified through five configuration files located within each scenario's `Input` directory. These files define the radar system parameters, scenario settings, sensing configuration, and PRS settings. Each parameter's allowable range and default values (if applicable) are described below.

2.1.1 Target Configuration (`targetConfig.txt`)

This file defines the ST's position and velocity as 6D coordinates as [x, y, z, vx, vy, vz] where x,y,z are in m and vx, vy, vz are in m/s. There are no strict limits, but values should correspond to the simulation environment.

2.1.2 Base Station Configuration (`bsConfig.txt`)

This file defines the base station's position as 3D coordinates in meters, specified as [x, y, z]. There are no strict limits, but values should correspond to the simulation environment.

2.1.3 PRS Configuration (`prsConfig.txt`)

Defines the Positioning Reference Signal (PRS) parameters:

- `PRSResourceSetPeriod`: Slot periodicity and offset, specified as [TPRSPeriod, TPRSOffset]. TPRSPeriod must be one of {4, 5, 8, 10, 16, 20, 32, 40, 64, 80, 160, 320, 640, 1280, 2560, 5120, 10240}, and TPRSOffset must be in the range [0, TPRSPeriod – 1]. Default: [1, 0].
- `PRSResourceOffset`: Slot offset of each PRS resource relative to the PRS resource set offset. Range: 0–511; default: 0.
- `PRSResourceRepetition`: Number of times each PRS resource is repeated. Allowed values: {1, 2, 4, 6, 8, 16, 32}; default: 1.
- `PRSResourceTimeGap`: Slot offset between consecutive repeated instances of a PRS resource. Allowed values: {1, 2, 4, 8, 16, 32}; default: 1.

- `NumRB`: Number of Physical Resource Blocks (PRBs) allocated for PRS. Range: 0–275; default: 66.
- `RBOffset`: Starting PRB index for PRS allocation. Range: 0–274; default: 0.
- `CombSize`: PRS comb size, indicating the spacing between resource elements within a PRS. Allowed values: {2, 4, 6, 12}; default: 4.
- `REOffset`: Resource Element offset within the PRS comb. Range: 0 to CombSize−1; default: 0.
- `NPRSID`: PRS sequence identity, used to generate different PRS sequences. Range: 0–4095; default: 4.
- `NumPRSSymbols`: Number of OFDM symbols allocated for each PRS resource. Range: 0–12; default: 4.
- `SymbolStart`: Starting OFDM symbol index for PRS within a slot. Range: 0–13; default: 1.

2.1.4 Sensing Configuration (`sensConfig.txt`)

Defines sensing-related processing parameters:

- `pri`: Pulse Repetition Interval (PRI) in seconds, defining the time between consecutive pulses. Range: >0 to 10; default: 0.0005.
- `dopplerFftLen`: Length of the Fast Fourier Transform (FFT) used for Doppler processing. Range: 1–2048; default: 64.
- `window`: Type of window function applied in Doppler processing. Choices: {'rect', 'hamming', 'blackmanharris', 'gaussian'}; default: 'blackmanharris'.
- `windowLen`: Length of the window function. Range: 1 to dopplerFftLen; default: equal to dopplerFftLen.
- `windowOverlap`: Overlap ratio between consecutive windows in Doppler processing. Range: 0–1; default: 0.5.
- `pulsesCpi`: Number of pulses per Coherent Processing Interval (CPI). Range: 0–1000; default: 16.
- Constant false alarm rate (CFAR) parameters (optional):
  - `cfarGrdCellRange`, `cfarGrdCellVelocity`: Number of guard cells in range and velocity dimensions. Range: 0–10000; default: 0.
  - `cfarTrnCellRange`, `cfarTrnCellVelocity`: Number of training cells in range and velocity dimensions. Range: 0–10000; default: 0.

2.1.5 Simulation Configuration (`simulationConfig.txt`)

This configuration file sets general simulation parameters. The software loads and validates the following parameters from the file, ensuring they are within allowed ranges, or otherwise assigns default values:

- **Carrier Frequency** (`systemFc`): Defines the operating frequency. Allowed range: 1 GHz – 60 GHz; default: 30 GHz.
- **Noise Figure** (`systemNF`): Sets the system noise figure. Allowed range: $0 - 20$ dB; default: 7 dB.
- **System Bandwidth** (`systemBw`): Determines the simulation bandwidth in MHz. Allowed values: {20, 40, 60, 80, 100, 200, 400}; default: 100.
- **Channel Scenario** (`channelScenario`): Specifies the propagation environment according to standardized scenarios. Allowed values: {`UMiAV`, `UMaAV`, `RMaAV`}; default: `UMiAV`.
- **Horizontal Antenna Elements** (`antennaNumH`): Number of antenna elements horizontally. Allowed range: 1 – 1024; default: 32.
- **Vertical Antenna Elements** (`antennaNumV`): Number of antenna elements vertically. Allowed range: 1 – 1024; default: 32.
- **Antenna Coupling Efficiency** (`antennaCouplingEfficiency`): Efficiency factor of the antenna coupling. Allowed range: 0 – 1; default: 0.7.
- **Carrier Subcarrier Spacing** (`carrierSubcarrierSpacing`): Spacing between OFDM subcarriers in kHz. Allowed values: {15, 30, 60, 120, 240} kHz; default: 120 kHz.
- **Carrier Grid Size** (`carrierNSizeGrid`): Size of the carrier resource grid in Physical Resource Blocks (PRBs). Allowed range: 1 – 275; default: 66.

Each parameter can be adjusted in `simulationConfig.txt` to tailor the simulation scenario to specific research requirements or experiments.

2.2 Channel File Specification (`.json`)

This section describes the JSON schema used by 5GNrad to load wideband multipath channels (e.g., 3GPP 38.901 UMa NLOS). Channel files are stored in a scenario's `/Input/` folder and are loaded automatically by `loadBackgroundChannel` and `loadTargetChannel` when you call `configScenario`.

2.2.1 What this file contains

A channel file encodes a set of multipath "drops" (or clusters), each comprising parallel arrays for delays, powers, angles, and per-ray phases. Keys observed in the provided file include: `PathDelays`, `AveragePathGains`, `AnglesAoD`, `AnglesAoA`, `AnglesZoD`, `AnglesZoA`, `InitialPhases`, `CPM`, and `HasLOSCluster`.

2.2.2 File structure

- **Container**: a JSON *array*. Each element corresponds to one multipath drop (or cluster list) at a given TX–RX pairing or time instant.
- **drop object**: each element is a JSON *object* with the fields listed below. Arrays within a drop must have the *same length* (number of rays).

2.2.3 Fields (per drop)

`PathDelays` Array of real numbers (seconds). Delays of each ray.

`AveragePathGains` Array of non-negative real numbers (linear power). One value per ray, aligned with `PathDelays`.

`AnglesAoD` Array of azimuth angles of departure in degrees (-180 to $+180$). One value per ray.

`AnglesAoA` Array of azimuth angles of arrival in degrees (-180 to $+180$). One value per ray.

`AnglesZoD` Array of zenith angles of departure in degrees (3GPP convention; negative offsets from boresight). One value per ray.

`AnglesZoA` Array of zenith angles of arrival in degrees. One value per ray.

`InitialPhases` Array of per-ray phases in radians (range $-\pi$ to π).

`CPM` Optional array holding 2×2 complex cross-polarization matrices per ray. If unused, keep as `[]`.

`HasLOSCluster` Boolean flag indicating whether a LOS cluster is explicitly included in this drop.

2.2.4 Generating your own channels

1. **Decide the drop granularity**: one file can carry one or many drops (array length S). Use multiple drops for time evolution.

2. **Assemble per-ray arrays**: for each drop, compute N_{rays} values for delay, power (linear), azimuth/zenith AoD/AoA, and phase. Keep array lengths equal and rays aligned by index.
3. **Polarization**: if you model dual-pol coupling, encode a 2×2 `CPM` per ray. Otherwise leave `CPM` empty `[]`.
4. **LOS handling**: set `HasLOSCluster=true` and include a dedicated LOS ray if your scenario requires it (UMa NLOS typically uses `false`).
5. **Place the file**: save under `<channel>/` and point your scenario to it (5GNrad picks it up when `configScenario` calls the background/target channel loaders).(loader call) :contentReference[oaicite:13]index=13

2.2.5 Channel File Naming Conventions

5GNrad distinguishes between *background* channels and *target* channels, each stored as a JSON file in the `channel/` subfolder of the root path. The filenames are automatically constructed from two elements:

- the carrier frequency expressed in GHz (`fcGHz`),
- the scenario type, inferred from the scenario path.

2.2.5.1 Background Channel The background channel file is named according to the pattern:

`backgroundChannel_<fcGHz>GHz_<scenarioType>.json`

where the scenario type matches the 3GPP 38.901 nomenclature:

- `3GPP_38.901_UMa_NLOS` if the scenario path contains "uma",
- `3GPP_38.901_UMi_NLOS` if the scenario path contains "umi",
- `3GPP_38.901_RMa_NLOS` if the scenario path contains "rma".

2.2.5.2 Target Channel The target channel file follows a similar convention, with filenames of the form:

`targetChannel_<fcGHz>GHz_<scenarioType>.json`

but here the scenario type uses the suffix:

- `UMaAV` if the scenario path contains "uma",

- `UMiAV` if the scenario path contains “umi”,
- `RMaAV` if the scenario path contains “rma”.

2.2.5.3 Example. For a UMa scenario at 6 GHz, the expected filenames are:

- `backgroundChannel_6GHz_3GPP_38.901_UMa_NLOS.json`
- `targetChannel_6GHz_UMaAV.json`

3 Output Description

Upon completion, the simulation generates an output file named `error.csv`, saved in the following path relative to the selected scenario:

```
./examples/UMi-Av25/Output/error.csv
```

This file contains tabular evaluation metrics for each sensing realization or time step. The columns include:

- `positionError`: Total 3D position error in meters between the estimated and true target location.
- `rangeError`: Error in estimated round-trip delay converted to meters.
- `velocityError`: Absolute error in estimated radial velocity (m/s).
- `azimuthError`: Azimuth angular error in degrees.
- `elevationError`: Elevation angular error in degrees.
- `dopplerPeakPower`: Maximum power observed in the Doppler spectrum for the detected target.
- `dopplerpeakToAverage`: Peak-to-average power ratio in the Doppler domain.
- `isLos`: Line-of-sight condition (1 if LoS, 0 if non-LoS).