How QuaDRiGa Generates Channel Coefficients

(Technical Documentation: section 3 of QuaDRiGa v2.0.0 documentation)

3GPP-SCM, WINNER, COST, and 3GPP-3D channel models focused on 2-D modelling. WINNER+ came with parameter tables with the additional elevation component. Polarization was also a key to increasing spatial degrees of freedom, which was adopted by WINNER, but was still missing some of the geometry-based stochastic modelling and was strictly incorporated statistically. All the previous models had limited continuous time evolution capabilities. The only supported scope was a few milliseconds -> restricted mobile terminal (MT) mobility.

These notes (QuaDRiGa v2.0.0 chapter 3) describe how QuaDRiGa extended WINNER to include time evolution, geometric polarization, and 3-D propagation effects. Modelling approach consists of 2 steps: stochastically generated large-scale parameters (LSPs: delay/angular spreads) and random 3-D positions of scattering clusters.

Inputs to the model: network layout (qd\_layout), terminal trajectories (qd\_track), propagation scenario (qd\_track), antenna patterns (qd\_arrayant), simulation parameters (qd\_simulation\_parameters)

Possible to extract channel coefficients from qd\_layout with get\_channels

qd\_builder generates the actual LSPs, SSFs, and functions that are needed to generate the channel coefficients. Handles most of the steps below

Output of model should be of type “qd\_channel” which contains coefficients and delays

For detailed explanation, continue reading. Main summary/takeaways from what I saw with using QuaDRiGa can be seen at the end of this document at “**Takeaways**”

Step 1: Calculation of correlated large-scale parameter maps

Ensures LSPs are consistent. **Delay**, **angular** **spreads**, K-**factor**, and **shadow** **fading** do not change rapidly

Step 2: Calculation of initial delays and path powers

LSPs known -> fast-fading channels calculated separately for each mobile terminal. Takes specific values of delay spread and K-factor from step 1 -> translated to multipath components, each with specific **power** and **delay** value

Step 3: Calculation of departure and arrival angles

Each multipath component gets assigned specific departure direction at TX and arrival direct ion at RX. **Power** and **delay** values from step 2 remain unchanged/ Directions are chosen s.t. **angular** **spreads** from step 1 are maintained.

Step 4: Drifting of delays, angles, and phases **over a short segment**

Incorporates mobility/spherical waves at mobile terminal. Given outputs of steps 2 and 3, calculates position of **last-bounce scatterer** (LBS). When mobile terminal moves, LBS positions are kept fixed, **delays** and **directions** are updated. Update of **phases** from multipath components (Doppler shift). Possible because propagation path changes in a deterministic matter (mobile terminal path/speed known).

Step 5: Calculation of polarized channel coefficients

Takes care of antenna and polarization effects. Includes method to change orientation of antennas to match mobile terminals/base station orientations (defined in inputs). Effects due to scattering of multipath components handled by method (Jones calculus) with successive linear transformations -> **calculates polarization state of a multipath component**

Step 6: Application of path gain, shadow fading, and K-Factor

Distance-dependent path gains and shadow fading are applied to channel coefficients. Also considers differences caused by step 4.

Step 7: Transitions between segments

For long sequences. Some scattering clusters change. Some scenarios may change due to mobile terminal path. Terminal’s path split into **segments** where LSPs do not change (WSS properties). Channel traces generated independently for each **segment**. All traces/segments combined in a longer sequence in last step.

Channel coefficients are calculated at a constant sample rate that fulfills sampling theorem

Appropriate sampling rate is proportional to the maximum speed of the MT. Channel coefficients are calculated at fixed positions with sampling rate fs. Aka sample density.

Step 1 deals with LSP->aka Large-scale fading model. Steps 2-6 deal with small-scale-fading (SSF). The 7 LSPs are:

1. RMS delay spread
2. Ricean K-factor
3. Shadow fading
4. Azimuth spread of departure
5. Azimuth spread of arrival
6. Elevation spread of departure
7. Elevation spread of arrival

Distance dependent correlation is modeled by 2-D maps. Maps initialized from i.i.d. zero-mean Gaussian with desired variance. Once maps are generated, initial LSPs for each **segment** are obtained. Granularity of each LSP described on three levels: propagation scenario level, link level, and path level.

Once LSP maps are generated, SSF part of the model generates individual scattering clusters for each mobile terminal. They read **delay spread** and **K-factor** -> delays are drawn randomly from a scenario-dependent delay distribution. Delays are adjusted s.t. first delay is set to zero and then they are sorted.

NLOS path **powers** are drawn from single slope exponential power delay profile (PDP) depending on the delay spread and random component from scenario-dependent coefficients. Power from first path is scaled according to initial K-Factor from the map and powers are normalized so that their sum power is 1 Watt. Scaling by K-factor changes delay spread. Delays are scaled such that the correct delay spread can be calculated from the generated path-delay and path-powers.

WINNER and 3GPP-3D models scale delays with an empiric formula that corrects the delays/reduce effect of high K-Factor. Delay spread is always different from the value in the map with QuaDRiGa. New method ensures that scattering clusters are distributed in a way that the delay spread calculated from multipath components is same as that in the map. Departure/arrival directions of multipath components discussed next, and those are combined with delays in order to calculate 3-D positions of scattering clusters.

**Departure/Arrival angles**

Four angles are calculated for each propagation path. Azimuth angle of departure (AoD), azimuth angle of arrival (AoA), elevation angle of departure (EoD), elevation angle of arrival (EoA). Delay spreads for each corresponding angle obtained by reading values from LSP map. Multipath component angles are generated by assigning random angles to already known path powers from previous step. Different approach than WINNER or 3GPP-3D where angles are mapped to already known powers.

Azimuth angles: LOS angle defined as “0,” and random list for NLOS generated from Gaussian normal with zero-mean and variance given by LSP map. Mapped on an interval between [-pi, pi]. Angular spreads updated until actual spreads converge (most processing done for NLOS).

Elevation angles: somewhat similarly generated

**Antennas and Polarization**

Geometry-based stochastic channel models (GSCM) allow separations of propagation and antenna effects. Antennas do not radiate equally in all directions. Radiated power is a function of the angle/polarization. Quadriga handles this.

**Combining Sup-Paths into Paths**

Due to random initial phases, a simple sum of subpaths will result in a random path power. This issue left open by WINNER and 3GPP-3D. “Solved” by QuaDRiGa by definig an average power around which the path power is allowed to fluctuate.

**Path Gain, Shadow Fading, and K-Factor**

Last step of small-scale-fading model. Complex-valued amplituge *g* for each multipath components are described for all antenna pairs.

**Transitions between segments**

Used for mobile terminals that move far enough for LSPs to change… may not be relevant for our current testing needs

**Takeaways**

QuaDRiGa extends the LSF and SSF parts of previous models. Especially in areas of spatial consistency with LSPs, LSF, SSF consistency, mobility of terminals, and antenna polarization

Now, the output of the channel model can be directly compared to the output of a measurement campaign. It is possible to generate **channel coefficients** with the same spatial and temporal resolution as **measured data**. -> new and improved way to generate channel state information/channel coefficients than previously used models.

The qd\_channel object that is outputted from QuaDRiGa model contains the channel coefficients.

Each coefficient is indexed by [number of receive element, number of transmit element, number of path, number of snapshot]

Receive element corresponds to an Rx/part of Rx antenna array, likewise for transmit element. Number of path corresponds to either the LOS or NLOS multipath. Snapshot corresponds to (…?) channel being merged from longer time/changing position/path of rx terminals.

Example of how coefficients are generated:

A close up of a logo

Description automatically generated

s becomes the qd\_simulation\_parameter object which will store parameters such as sample density, samples per meter, center frequency, and some other parameters that are binary (on/off).

A screenshot of a cell phone

Description automatically generated

a and a2 become qd\_arrayant objects where the user can manipulate the antenna parameters at the receiver/transmitter. Using the visualize function allows one to see a plot showing the element configurations.

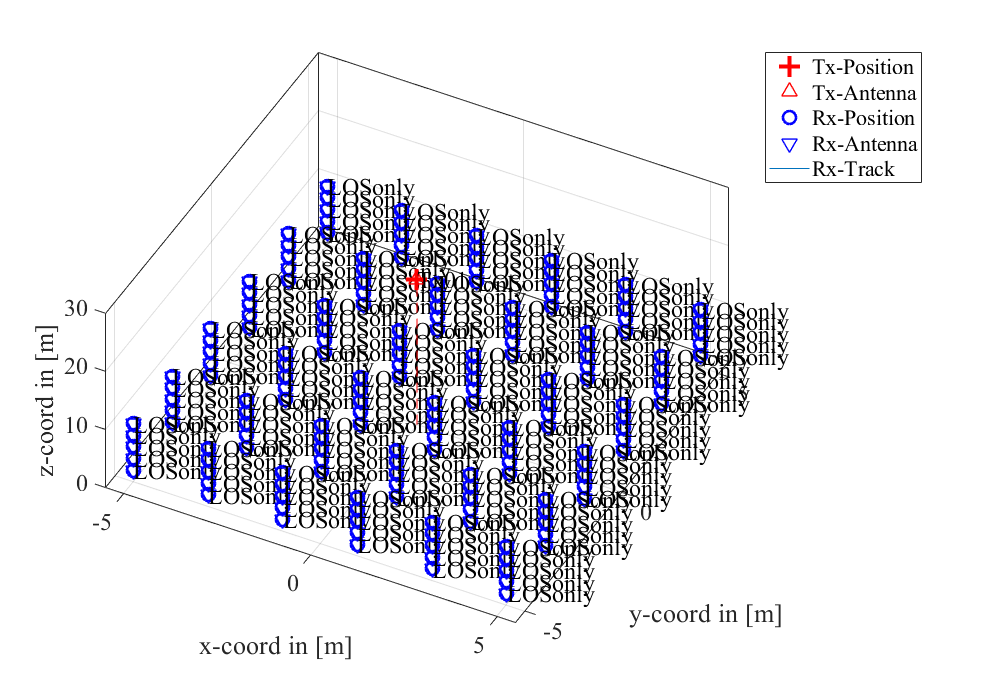
A close up of a map

Description automatically generated

A screenshot of a cell phone

Description automatically generated

l = qd\_layout(s) initializes a new layout (more user input into how the tx/rx’s are dispersed). l.track initializes how the MT’s will move (this example has them staying still). Scenarios are available to QuaDriGa for channels with different properties (inside, outside, some are pulled directly from measurements made from shared projects-Berlin).



Above image shows the grid of receivers equally spaced away from a centralized antenna to extract channel information

A screenshot of a social media post

Description automatically generated

p becomes a qd\_builder object which implements the previously mentioned steps for LSF generation etc. Have to manually generate ssf parameters using the qd\_builder once it finishes its initial LSF generation. Once that’s done, channel state information can be extracted.

A screenshot of a cell phone

Description automatically generatedA screenshot of a social media post

Description automatically generated

The line of sight coefficients can be seen above. The data ranges from (:,:,1,1) to (:,:,1,180) since the grid I made has 180 receivers (5x5x6). The delay vector corresponds to each channel coefficient, but they all have a value of 0 with the current implementation. I do not think this is correct, but it may be due to using a LOS only scenario and delays are meant to be used for multipath/NLOS delays from the LOS propagation time.

Code used can be found under this Github repository:

<https://github.com/JamesMcNaney/Summer19_MIMO/blob/master/Quadriga/tutorials/test03.m>

This code was modified from one of the QuaDRiGa tutorials, and there are still a few functions I don’t understand (or understand why they are breaking), but I believe what I know so far should be able to extract channel coefficients from the QuaDRiGa model (which are hopefully more accurate) in your simulation files.

Questions? Comments? Areas to explore/develop? How do you suggest I move forward?