

PyWiCh: Python Wireless Channel Simulator

Pablo Belzarena

Facultad de Ingeniería - UdelaR

Montevideo, Uruguay

email - belza@fing.edu.uy

Abstract—This work presents PyWiCh, an open source wireless channel simulator which can be used both as a stand-alone software through its graphical interface or integrated into other applications. Wireless channel simulators are an essential tool for the research and development of next-generation networks based on massive MIMO and millimeter waves. PyWiCh is the first simulator developed in Python that implements the 3GPP simulation model for 5G networks. PyWiCh also proposes solutions to two current research problems in wireless channel simulation: spatial consistency and scatters movement. Through its use, adoption, and enhancement, this project intends to build a community that continues work and research on this topic so as to improve the simulator and develop new features and new models which can be integrated into PyWiCh.

Index Terms—Python, wireless, channel, simulator, mMIMO, millimeter waves, 3GPP.

I. INTRODUCTION

New wireless technologies like 5G and beyond introduce new challenges in both theoretical and applied research. In particular, the use of massive multiple input multiple output (MIMO) and millimeter waves, although they significantly improve the performance of these networks, generates the need to understand better how a wireless channel behaves in these new conditions. A realistic simulation model is fundamental for evaluating how the wireless channel behaves in these new conditions. The 3rd Generation Partnership Project (3GPP) is in charge of carrying out the standardization process of cellular technologies, particularly 5G. This organization has produced various technical reports over the years with models for simulating different cellular technologies. The latest technical report, which applies to 5G technology, presents a channel model valid from 0.5 to 100 GHz that supports massive MIMO [1]. Different models and tools have been developed over the years to simulate a wireless channel. Given the impossibility of solving Maxwell's equations in complex scenarios where there are hills, forests, buildings, moving vehicles, etc., essentially two types of models have been proposed.

On the one hand, deterministic models have been proposed based on thinking of electromagnetic waves as reflected or scattered rays on different surfaces. These rays arrive from the transmitter to the receiver after being scattered on one or more surfaces. Although the approximation achieved with these models is very good, they generally have a high computational cost. They are also not easy to use since they need a good representation of the terrain, the relief, and the scenario. There

are many works on ray tracing for wireless channel simulation, starting with the seminal paper of Schaubach et al. [2].

Given the difficulties of the deterministic ray tracing method, different authors propose stochastic channel models (SCM). These models are simpler but often achieve less precise approximations. The spatial statistical channel model (SSCM) is an evolution of these models. The seminal paper for SSCM models is Saleh and Valenzuela [3]. These models assume that the scenario has several scattering surfaces that reflect the rays from the transmitter to the receiver. These scatters, also called clusters, are composed of a set of reflectors. In each cluster, a set of rays are reflected. Each ray is characterized by a power, a delay, and departure and arrival angles. These parameters (power, delay, and angles) behave according to a statistical model. These statistical models have a set of parameters that are adjusted according to the scenario (indoor, outdoor, etc.). A good review of wireless channel models can be found in [4], and a comparison between ray tracing models and SSCM models can be found in [5].

The 3GPP model is a 3D SSCM model. The transmitting and receiving devices move in a three-dimensional space, and the angles will have an azimuth and an elevation. The 3GPP model has different scenarios where large measurement campaigns estimate the model parameters for each scenario.

Different software proposals implement simulators of wireless channels with ray tracing or SSCM models. The vast majority of these simulators are proprietary software. Of those that implement SSCM models, we remark NYUSIM [6] and QUADRIGA [7]. QUADRIGA implements its own SSCM model, although it has several points in common with the 3GPP model. NYUSIM implements the 3GPP model, although it has some differences from this model. A comparison of both simulators can be seen at [8]. Both systems are widely used. They are intended to be used primarily as end-user software and not to be integrated with other systems. Their implementation in both cases is in the proprietary software MATLAB. Another implementation of 3GPP that should be highlighted is achieved within the ns-3 software [9]. The ns-3 software, among its channel models, has a partial implementation of the 3GPP. In this case, it is a C++ implementation, and is not intended as stand-alone channel simulation software nor to be integrated with other systems, but rather to be used as part of ns-3.

In this work, we propose a wireless channel simulation

system that implements the 3GPP standard, developed entirely in Python, intended to be used both as stand-alone software with a graphical interface as well as to be integrated into a broader system. This software, as far as we know, is the only Python implementation of the standard 3GPP. PyWiCh is an essential tool for the research and development of next-generation networks based on massive MIMO and millimeter waves. In addition, PyWiCh proposes solutions to two current research problems in wireless channel simulation: spatial consistency and scatters movement.

PyWiCh is published on GitHub under GNU General Public License v3.0. PyWiCh source code, user manual and software documentation can be found in <https://github.com/PyWiCh/PyWiCh>. The previous link has the software to download and also how to cite PyWiCh [10].

The rest of the paper is organized as follows. We explain the architecture of PyWiCh in Section II, its main features in Section III and the graphical user interface in Section IV. Simulation results and discussion are presented in Section V and conclusions in Section VI.

II. PYWICH SOFTWARE DESIGN

PyWiCh software can be used as an API from another software or can be used stand alone. PyWiCh has a graphical user interface to facilitate simulations. Though the GUI provides the main features, the PyWiCh classes have more capabilities when called directly.

PyWiCh has three main software packages. The simulator package, which is the system's core, contains the main classes to configure and run the simulation. The GUI package has classes that implement the graphical user interface. The GUI package classes allow the creation and configuration of the simulator's main objects and run the simulation. The simulator stores the simulation results in a filesystem. The third main package, the Graph package, reads the simulation results from the files system and shows the main results graphically. Figure 1 shows the main packages of PyWiCh.

The simulator package of PyWiCh has four main Classes:

- The first one is Antenna class where the user can configure the transmit and receive antenna arrays.
- The second one is the Scenario class where the user can configure the size of the scenario, the Base Station position, the parameters of the system, the loss model, the shadowing model, etc.
- The third one is the FastFading class, that has the methods and procedures to build the wireless channel matrix and other fast fading parameters.
- The fourth one is the FrequencyBand class which has the information about the frequency band, the OFDM system and the noise characteristics.

PyWiCh has a fifth important class that is the ChannelPerformance class. This class is used to run the simulation and to obtain some performance metrics of the system.

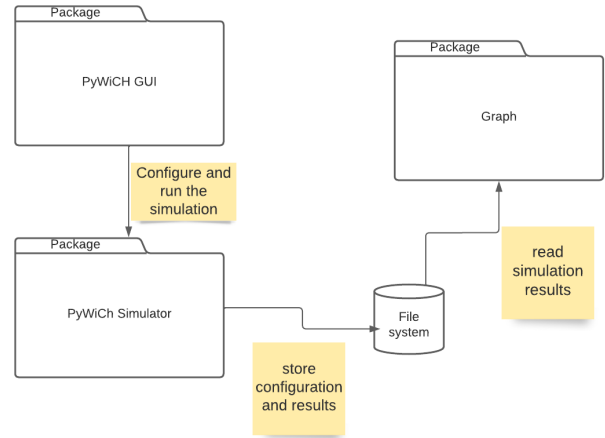


Figure 1: PyWiCh package Diagram. The graphical interface is cleanly decoupled from the simulator's core.

Before running a simulation, the five previous objects must be built and configured. The configuration parameters of each object are explained in the software documentation and depend on the type of object. For example an Antenna can be an isotropic antenna or a 3GPP antenna model. Both types of antennas have different parameters. Figure 2 shows a diagram with the main PyWiCh classes and their relationships. As can be seen in Figure 2, the AntennaIsotropic and Antenna3GPP classes inherit from the base class Antenna. If the user wants to add another Antenna model, he has to create a new child class with the corresponding parameters and override the method that calculates the antenna gain. The same happens with the base class Scenario. Currently, Scenario3GPP inherits from Scenario. Scenario3GPP class has all the attributes and methods common to the scenarios proposed in 3GPP. From Scenario3GPP, inherit Scenario3GPPIndoor, Scenario3GPPUMA, and Scenario3GPPUMI that contain the specific parameters and methods of each scenario. The user can add a new scenario that uses the same parameters of the 3GPP model by writing a child class of Scenario3GPP.

III. PYWICH MAIN FEATURES

In this version, PyWiCh implements the 3GPP channel model with some additional optional features. PyWiCh has an implementation of the following three scenarios of the 3GPP model:

- UMi scenario. Street canyon, open area. The Base Stations are mounted below rooftop levels of surrounding buildings. UMi is intended to capture real-life scenarios such as a city or station square.
- UMa scenario. The Base Stations are mounted above rooftop levels of surrounding buildings.
- Indoor scenario. This scenario is intended to capture various typical indoor deployment scenarios, including office environments, and shopping malls.

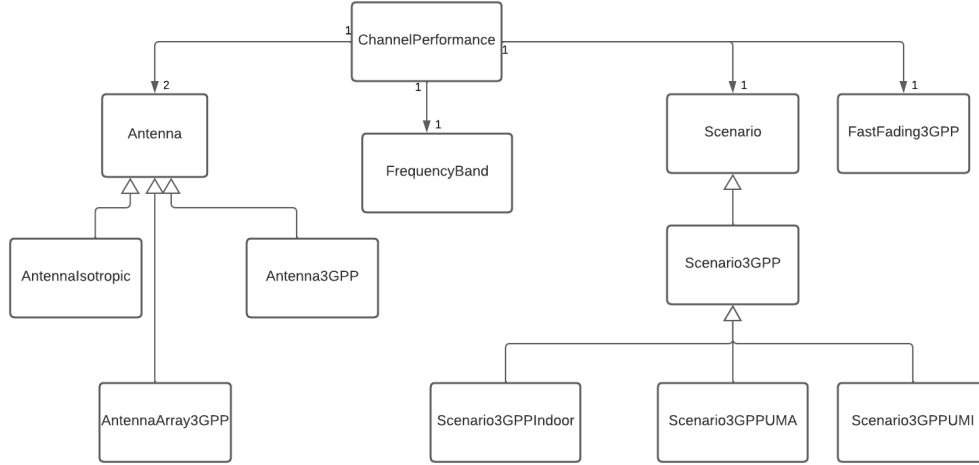


Figure 2: Diagram of PyWiCh main classes

If the user wants to build a new scenario, it is only necessary to write a class that inherits from the Scenario3GPP class and set the parameters for that class. For example, a user can take a measurement campaign, adjust the parameters to her measurements and enter these parameters into its class. In this way, the user has a model of his scenario. The main features of the 3GPP scenarios are:

- Path Loss. The path losses typically depend on the distance between the receiver and transmitter and operation frequency. Different scenarios have different expressions for path loss.
- Shadowing. The shadowing is the attenuation caused by different obstacles between the transmitter and the receiver. This variable is typically modeled with a log-normal random variable.
- LOS model. Each scenario has a Line of Sight probability model that typically depends on the distance between the transmitter and the receiver.
- Large-scale parameters. The 3GPP defines two types of parameters to build the fast-fading model of the scenario. The large-scale parameters (LSPs) are a parameter set that defines the behavior of the short-scale parameters (SSPs) that define the MIMO channel matrix. The LSPs parameters are the delay spread of the clusters delays, the angles spread (arrival and departure) of the cluster angles, the Ricean K factor, and the shadowing.
- Short-scale parameters. The SSPs characterize the behavior of each cluster and define the MIMO channel matrix. The main SSPs are:
 - The delay of each cluster.
 - The power delay profile.
 - The azimuth and inclination angles (arrival and departure) of each cluster and each ray of the cluster.
 - The antenna cross polarization.

In the following subsections, we remark two important

features of PyWiCh that can optionally complement the 3GPP channel model.

A. Spatial consistency

When two mobiles are close, their channel responses with the Base Station are correlated. The same occurs with the positions of one mobile route. These correlations imply that the LSPs and SSPs parameters must have spatial consistency. The 3GPP channel model defines in the first technical reports only the LSP spatial consistency. Each LSP is spatially correlated by a correlation distance specified in the 3GPP document. However, the SSPs in these first reports are independent random variables at different points of the scenario. This problem of the spatial consistency of the channel parameters is an interesting and current research topic [11] [12] [13]. NYUSIM and QUADRIGA propose methods for the spatial consistency of the SSPs parameters. In their last technical reports, the 3GPP proposes a method for the spatial consistency of the SSPs parameters [1]. This method has two parts. First, the method defines the SSP spatial correlation as a 2D random process based on correlation distance for each SSP. However, as is said in [12] what this 2D random process represents is not clearly defined. Second, the method proposes two different procedures to give spatial consistency during mobility simulations. PyWiCh uses the following approach for spatial consistency:

- 1) Using the correlation distance of each LSP, PyWiCh generates a grid of correlated LSPs values.
- 2) The route of each mobile user is divided into sections where the SSP parameters are correlated. A section of the route has the same set of LSP parameters.
- 3) Using the correlation distance of each SSP, PyWiCh generates a grid of correlated SSP parameters.

Two different methods in PyWiCh can implement spatial consistency in the route section. In the first one, PyWiCh

uses procedure A of section 7.6.3.2 of 3GPP document. In the second one, PyWiCh computes the mobile position and interpolates the four close corners of the SSPs grid.

B. Scattering movement

The scatters movement is a complex feature to implement in SSCM models. The scatters in this model are defined by their angles and the delay between the transmitter and receiver. Therefore their physical place in the scenario is not explicitly defined, and the movement is not easy to implement. The 3GPP model does not implement the scatters movement. PyWiCh optionally implements the scatters movement by an approximation. PyWiCh implements the scatters movement only when the SSPs spatial correlation is obtained using procedure A of section 7.6.3.2 of 3GPP document. During the SSPs update equations, PyWiCh computes the velocity of the mobile as the relative velocity vector of the difference between the scatter velocity vector and the mobile velocity vector.

IV. PYWICH GUI

The PyWiCh GUI helps to define, configure, run and analyze the simulation. Figure 3 shows the user interface of PyWiCh. With the first four buttons of the first row of buttons,

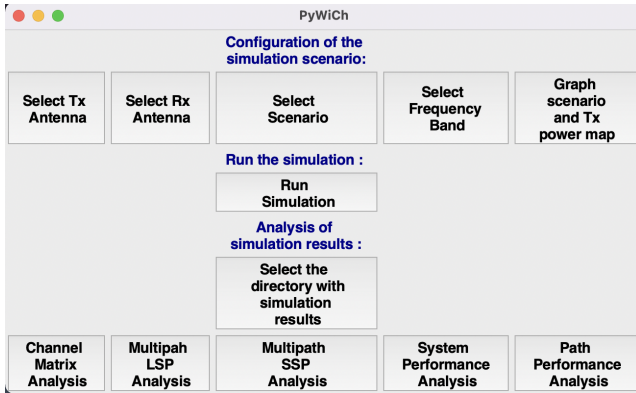


Figure 3: PyWiCh GUI

the user can configure the transmit antenna array, the receive antenna array, the scenario and the OFDM frequency band. For example, Figure 4 shows the configuration of the antenna array.

In addition to the configuration of each object, the user can visualize the behavior of the configured object. In the case of the antenna array configuration, the user can analyze the antenna elements' radiation pattern and the array's radiation pattern. Figure 5 shows the power radiation pattern of an antenna element that follows the 3GPP antenna model, and Figure 6 shows the 3D radiation pattern of an array of 16 antenna elements where each element follows the 3GPP antenna model.

After the user configures the antenna arrays, the scenario and the OFDM frequency band can plot the scenario. This plot shows the power radiation map for each scenario point

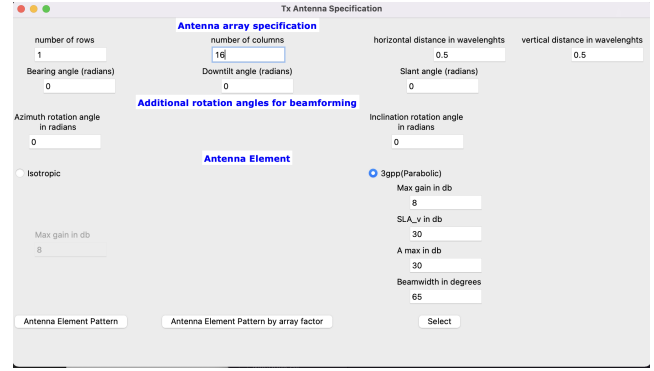


Figure 4: PyWiCh Tx antenna array configuration.

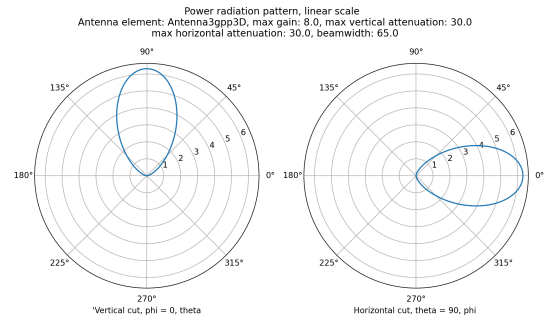


Figure 5: PyWiCh Tx antenna element power radiation pattern.

and the device's mobile route. Figure 7 plots an example of the radiation pattern of a base station placed in the position $(x=0, y=0)$ and the routes of two mobile devices.

V. MAIN SIMULATION RESULTS

After the user runs a simulation, PyWiCh saves the scenario configuration and the simulation results to files. The simulation results' files include the LSPs of the scenario, the SSPs of each mobile device route point, the path loss and shadowing for each point of each mobile route, and the channel matrix H_{usn} . H_{usn} is the channel matrix for each Tx antenna element u , each Rx antenna element s and each scatter n . These files have the raw results data of the simulation. In addition, during the simulation, PyWiCh computes some performance indicators. These main performance indicators are:

- The Fourier Transform of the channel matrix for each physical resource block of the OFDM grid.
- The received power at each point of each mobile device route and for each physical resource block of the OFDM grid.
- The SNR for each point of each mobile device route and for each physical resource block of the OFDM grid.

The PyWiCh GUI has several predefined graphical analysis of the simulation results. In the following Figures, we show only some of these graphical analyses. We remark that the user has all the raw simulation results in files, and she can

3D Power radiation pattern and power radiation pattern multiplied by array factor
Antenna array : 1x16 dh: 0.5 dv: 0.5

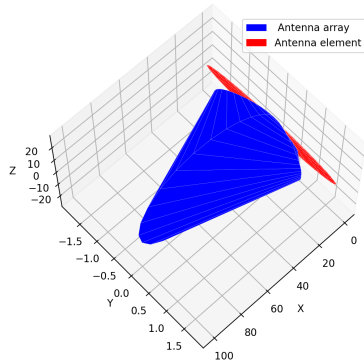


Figure 6: Radiation pattern of antenna array and antenna array element.

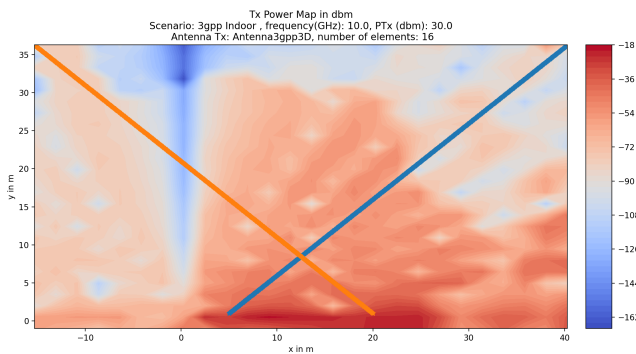


Figure 7: Power map and mobiles routes in the scenario.

build other performance analyses. The SSCM models have parameters (LSPs and SSPs) that are random variables. These random variable are correlated in space and between different parameters. PyWiCh GUI allows the analysis of the realization of these random variables and its correlations for the current simulation. For example, Figure 8 shows one of the LSP parameters (the Delay Spread) and its space variations in this simulation scenario. In addition, Figure 9 shows the delay of

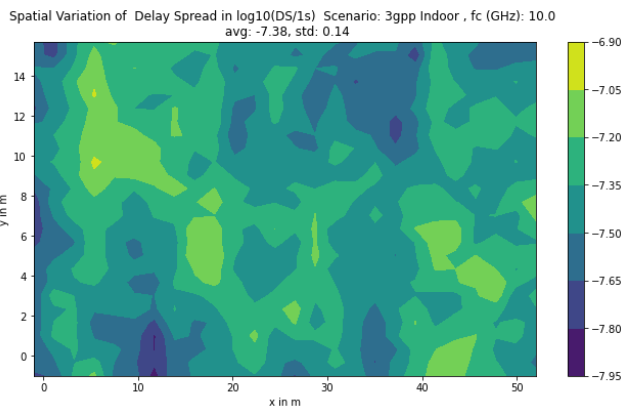


Figure 8: Delay Spread

each cluster for one point in the route of a mobile device.

In MIMO systems, it is essential to know the condition

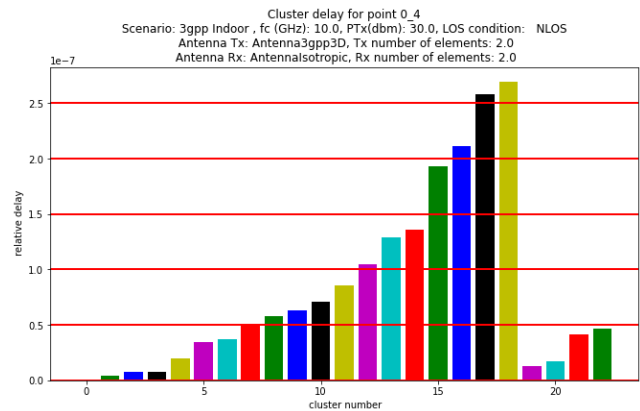


Figure 9: Cluster delays.

number of the Fourier Transform of the channel matrix. The condition number is the ratio of the maximum singular value to the minimum singular value of a matrix. The condition number gives information about the capacity gain of the MIMO system. When the condition number is close to 1, the MIMO system has a significant multiplexing gain, and the gain decreases when the condition number grows [14]. Figure 10 shows the condition number and the plot of the eigenvalues of the channel matrix for one physical resource block at one point of the mobile device route.

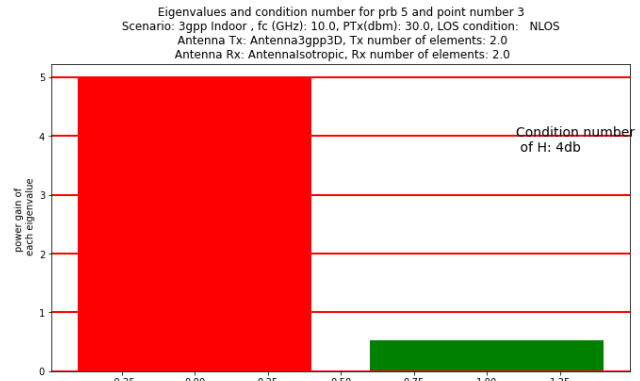


Figure 10: Condition number and eigenvalues plot

Figure 11 shows the received power in each physical resource block of the OFDM grid. The received power is the transmitted power in each physical resource block minus the path loss and shadowing at this point of the route plus the beamforming gain of the MIMO system. This analysis gives information about the fading in the OFDM band. For example, if it shows strong variations, the system has selective frequency fading, or if it has no variations, the system has flat fading in this band.

Finally, Figure 12 shows the last example of the many possibles analysis of simulation results. It shows the SNR of the mobile device along its route. The authors validated

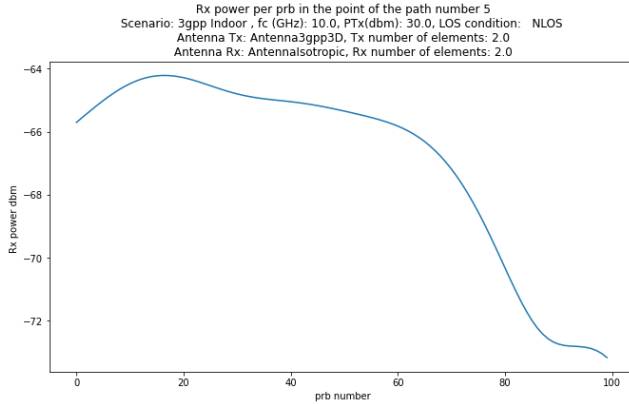


Figure 11: Received power per physical resource block.

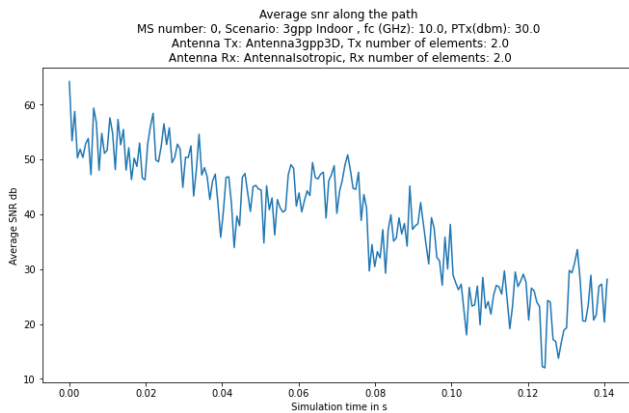


Figure 12: SNR along the mobile device route.

the software against the standard 3GPP model in different components (antenna, scenario, fast fading, etc.). Along with the software, the test and validation code used is distributed.

VI. CONCLUSIONS

In this work, we have presented PyWiCh. This software simulates a wireless channel in the frequency range from 0.5 to 100 GHz. PyWiCh is the first simulator developed in Python that implements the 3GPP channel model for 5G. Furthermore, it presents other important additional features to simulate a wireless channel. In this paper, we have seen the general architecture of PyWiCh, where the graphical interface is cleanly decoupled from the simulator's core. This separation enables the software to be used as a stand-alone tool, or its engine can be integrated into other applications. In addition, we show examples of PyWiCh graphic tools that allow analyzing the behavior of mobile devices when moving in a simulation scenario. This first version is available on Github, and we invite all those interested in developing and researching this topic to join the project as collaborators to continue the development of this tool.

ACKNOWLEDGEMENTS

This work has been partially supported by the Agencia Nacional de Investigación e Innovación, Uruguay (Project FMV_1_2019_1_155700, "Artificial Intelligence applied to 5G networks") and Fondo Carlos Vaz Ferreira, Convocatoria 2021, Dirección Nacional de Innovación, Ciencia y Tecnología, Ministerio de Educación y Cultura, Uruguay (Project FVF-2021-128- DICYT). The author would like to thank Víctor González Barbone and Alicia Fernandez for helpful discussions.

REFERENCES

- [1] 3GPP TR 38.901 version 16.1.0 Release 16. 5G; Study on channel model for frequencies from 0.5 to 100 GHz. 2020.
- [2] K. R. Schaubach and N. J. Davis and T. S. Rappaport, "A ray tracing method for predicting path loss and delay spread in microcellular environments," [1992 Proceedings] Vehicular Technology Society 42nd VTS Conference - Frontiers of Technology, 1992, pp. 932-935 vol.2, doi: 10.1109/VETEC.1992.245274.
- [3] A. A. M. Saleh and R. Valenzuela, "A Statistical Model for Indoor Multipath Propagation," in IEEE Journal on Selected Areas in Communications, vol. 5, no. 2, pp. 128-137, February 1987, doi: 10.1109/JSAC.1987.1146527.
- [4] Kan Zheng and Suling Ou and Xuefeng Yin, "Massive MIMO Channel Models: A Survey", International Journal of Antennas and Propagation, vol. 2014, Article ID 848071, 10 pages, 2014. <https://doi.org/10.1155/2014/848071>.
- [5] Tariq, S. and Al-Rizzo, H. and Hasan, M. N. and Kunju, N. and Abushamleh, 2021, 'Stochastic versus Ray Tracing Wireless Channel Modeling for 5G and V2X Applications: Opportunities and Challenges', in H. Al-Rizzo, S. Abushamleh (eds.), Antenna Systems, IntechOpen, London. 10.5772/intechopen.101625.
- [6] S. Ju and O. Kanhere and Y. Xing and T. S. Rappaport, "A Millimeter-Wave Channel Simulator NYUSIM with Spatial Consistency and Human Blockage," 2019 IEEE Global Communications Conference (GLOBECOM), Hawaii, USA, Dec. 2019, pp. 1-6.
- [7] F. Burkhardt and S. Jaeckel and E. Eberlein and R. Prieto-Cerdeira, "QuaDRiGa: A MIMO channel model for land mobile satellite," The 8th European Conference on Antennas and Propagation (EuCAP 2014), 2014, pp. 1274-1278, doi: 10.1109/EuCAP.2014.6902008.
- [8] Y. He and Y. Zhang and J. Zhang and L. Pang and Y. Chen and G. Ren, "Investigation and Comparison of QuaDRiGa, NYUSIM and MG5G Channel Models for 5G Wireless Communications," 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), 2020, pp. 1-5, doi: 10.1109/VTC2020-Fall49728.2020.9348775.
- [9] Riley, G.F. and Henderson, T.R. (2010). The ns-3 Network Simulator. In: Wehrle, K., Güneş, M., Gross, J. (eds) Modeling and Tools for Network Simulation. Springer, Berlin, Heidelberg.
- [10] Pablo Belzarena, PyWiCh-Python Wireless Channel Simulator, version 1.0, release date 29/7/2022, DOI:10.5281/zenodo.6941434, <https://github.com/PyWiCh/PyWiCh>.
- [11] Y. Wang and Z. Shi and M. Du and W. Tong, "A millimeter wave spatial channel model with variant angles and variant path loss," 2016 IEEE Wireless Communications and Networking Conference, 2016, pp. 1-6, doi: 10.1109/WCNC.2016.7564894.
- [12] F. Ademaj and S. Schwarz and T. Berisha and M. Rupp, "A Spatial Consistency Model for Geometry-Based Stochastic Channels," in IEEE Access, vol. 7, pp. 183414-183427, 2019, doi: 10.1109/ACCESS.2019.2958154.
- [13] J. Bian and C. -X. Wang and X. Gao and X. You and M. Zhang, "A General 3D Non-Stationary Wireless Channel Model for 5G and Beyond," in IEEE Transactions on Wireless Communications, vol. 20, no. 5, pp. 3211-3224, May 2021, doi: 10.1109/TWC.2020.3047973.
- [14] David Tse and Pramod Viswanath. 2005. Fundamentals of wireless communication. Cambridge University Press, USA.