Generation of 5G/6G Wireless Channels Using Raymobtime with Sionna's Ray-Tracing

Sávio Bastos, Ailton Oliveira, Daniel Suzuki, Luan Gonçalves, Ilan Sousa and Aldebaro Klautau

***Abstract*— It is challenging to generate wireless channels for 5G/6G scenarios with sophisticated mobility patterns. Ray-tracing (RT) is one of the most adopted techniques, especially when 5G/6G millimeter frequency bands are considered. However, most ray tracers, such as Wireless InSite and Sionna's, provide limited support to mobility: only the radio transceivers move over time, while all other objects are held in fixed positions. The Raymobtime methodology was proposed to fill this gap. Raymobtime supports repositioning all mobile objects, which enables simulating advanced mobility patterns. When the Raymobtime methodology was first implemented, its RT was solely based on the commercial Wireless InSite ray tracer. This work expands Raymobtime software to also support the recent NVIDIA Sionna's ray tracer, which is open source and mostly written in Python. The preliminary results indicate that Sionna's brings a fast RT tool, but it still needs to be systematically validated against measured channels.**

***Keywords*— Wireless, channels, ray-tracing, 5G/6G.**

I. INTRODUCTION

High mobility scenarios have proven to be a challenge for 5th generation (5G) and the 6th generation (6G) of wireless systems to achieve the strict proposal requirements regarding, low latency and high reliability, leveraging a diverse range of frequency bands, spanning from below 6 GHz to 1 THz [1]. The ray-tracing (RT) simulations have proven to be accurate and effective tools for modeling wave propagation in the high-frequency domain, comprehensively and precisely representing radio behavior in complex environments, considering reflections, diffraction, and scattering factors [2]. By modeling these interactions, RT simulations enable deep insights into the complex propagation characteristics of wireless signals, thereby aiding in the efficient design and optimization of mobile communication networks.

Raymobtime [3], presents a comprehensive approach for generating realistic datasets to simulate wireless communications. Several channel generation software provides support to mobility, but this is limited to moving the radio transceivers. In contrast, Raymobtime simulates the mobility of all mobile objects in a scene, including scatters. Using ray-tracing to obtain the channels, even with sophisticated mobility simulations, Raymobtime achieves temporal, spatial, and spectral consistency. This facilitates researchers to investigate machine learning and other techniques in the context of wireless communication simulations.

S. Bastos, A. Oliveira, D. Suzuki, L. Gonçalves, I. Sousa, A. Klautau, are with LASSE UFPA, Belém-Pará, Brasil; E-mails: [savio.bastos, ailton.pinto, daniel.suzuki, luan.gonçalves] @itec.ufpa.br, [ilan, aldebaro] @ufpa.br. The authors thank RNP/MCTI for the financial assistance to the project Brasil 6G (01245.010604/2020-14).

When Raymobtime was proposed, ray-tracing was solely based in Wireless InSite (WI), with real world scenarios and mobile scatters thereby pursuing a heightened accuracy. WI is a robust software commercialized by Remcom, which has its results validated by several works [4].

NVIDIA has recently made available the Sionna™ wireless communications simulator [5]. It is a link-level simulator to prototype communications systems. It has support to the integration of neural networks and machine learning strategies to communication systems.

This paper expands the software that implements the Raymobtime methodology, to support using the Sionna RT module. This combination of open source projects allows generating large datasets of wireless channels for research and development of 5G and 6G network applications. The contributions of this paper are: a) two datasets with small scale parameters taken from RT simulations; b) and source code to reproduce the dataset, and generate new ones.[1]

II. SYSTEM MODEL

This section presents the dynamics of the Raymobtime methodology implemented with Sionna's ray tracer. Two fundamental concepts of this methodology are *scenes* and *episodes*. Scenes are snapshots of the environment and the psychical events related to the propagation of radio-frequency waves in a given instant of time. Episodes are composed of a set of scenes equally spaced by a sampling period T_s, small enough to preserve the temporal correlation between scenes, in which a new set of mobile objects (car, bus, truck) are chosen to be the receivers and are kept along the scenes of the episode. Besides, all episodes are separated by a period T_e large enough so each episode is independent of one another. The following two subsections presents how the mobility pattern (*placement*) were inserted in each episode and how RT simulations were conducted.

The placement process is described in Fig. 1. The python orchestrator will start the placement process by calling Simulation of Urban Mobility (SUMO)[2] and using its information (positions, orientation, etc.) to write the samples of the simulation. Then, an XML file is created containing every object of the scene (buildings, cars, buses, and trucks) and the meshes indicators of each object. In addition, it chooses a set of vehicles to be the receiver (Rx) and creates a JSON file with the coordinates of each Rx and transmitter (Tx) in the sample. When the last scene of an episode is completed, a time period T_e is elapsed, and a new episode starts.

[1]The code will be published to Github in the case of the paper's approval.
[2]https://www.eclipse.org/sumo/

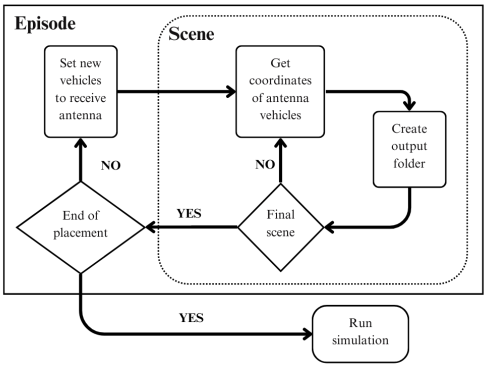

Fig. 1: Simulation dynamics fluxogram.

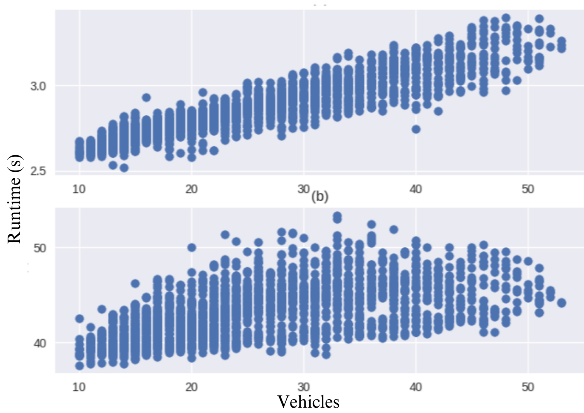

Fig. 2: Runtime: Simulation (a) Sionna, (b) Wireless InSite.

With the conclusion of the placement, the script executes the RT simulation of each sample with their respective scenario file. Initially, the scenario is loaded, using JSON file to load the Tx and Rx antennas, then the Sionna simulates the RT of the sample, and by the end of one simulation, the rays data is saved in an npz file. That process is repeated until the total number of samples. The Raymobtime dataset organization has been adapted for Sionna RT output files. The dataset is organized in a 4th dimensional (4D) structure: number of scenes, number of Tx and Rx pairs, the maximum number of rays, and rays parameters. The rays parameters, in order, are the: received power (dBm), time of arrival (seconds), the elevation angle of departure (degrees), the azimuth angle of departure (degrees), the elevation angle of arrival (degrees), the azimuth angle of arrival (degrees). Unlike the number parameters of [3], some information has not been obtained, such as phase angle and line of sight (LOS) indicator.

III. Results

With the proposed method, a new dataset was created. It uses the Paulista Avenue from the city of São Paulo, Brazil as scenario. The Paulista Avenue region was chosen due to its urban canyon organization. Table I presents the simulation parameters, including T_s, the number of scenes N_s, and the number of episodes N_e. The simulation involved ten mobile Rx's in each scenery, and the transceivers were set to transmit/receive a carrier at 60 GHz.

TABLE I: Configuration of the new dataset

Local	Frequency	T_s	N_s	Ep	Samples	Mobile
São Paulo	60 GHz	1 s	1	2300	2300	True
São Paulo	60 GHz	10 ms	100	25	2500	True

Futhermore, the simulation's runtime was compared between WI and Sionna simulations, using the same scenario and arrangement of elements in scenes. The Fig. 2 results show that Sionna's performance is ten times faster than WI simulations. Collecting the info from both WI and Sionna and calculates the Multiple-Input Multiple-Output (MIMO) channels, then it was calculated the Normalized Mean Square Error (NMSE) from the channels, such as in [4], using the WI as channel reference. Since Sionna does not provide the channel phase, we adopted a random uniform phase. The values of NMSE in Fig. 3 show sometimes large discrepancy between the channels generated by WI and Sionna.

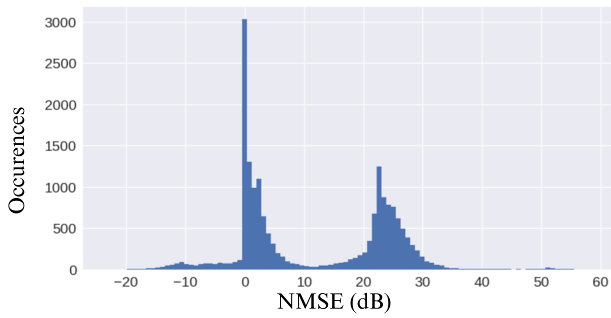

Fig. 3: NMSE histogram between Sionna and Wireless InSite channels.

IV. Conclusions

WI is a well-established RT simulator, already validated with experiments that compare its results with measurement channels. Sionna's RT is open-source software and presents relatively fast execution time. However, as shown in this paper, the channels obtained with Sionna's RT present sometimes a huge discrepancy when compared to the ones generated by WI. This suggests that Sionna's RT stills requires a thorough validation. While research advances in this area, adding advanced Raymobtime mobility to Sionna increases its usability for different applications, such as vehicular networks.

References

[1] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6G wireless systems: Vision, requirements, challenges, insights, and opportunities," *IEEE Proceedings*, vol. 109, no. 7, pp. 1166–1199, 2021.

[2] M. Rumney, "The critical importance of accurate channel modelling for the success of mmWave 5G," in *2017 11th European Conference on Antennas and Propagation (EUCAP)*. IEEE, 2017, pp. 3688–3691.

[3] A. Klautau, A. de Oliveira, I. P. Trindade, and W. Alves, "Generating MIMO Channels For 6G Virtual Worlds Using Ray-tracing Simulations," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2021.

[4] D. Suzuki, A. Oliveira, L. Gonçalves, I. Correa, A. Klautau, S. Lins, and P. Batista, "Ray-Tracing MIMO Channel Dataset for Machine Learning Applied to V2V Communication," in *2022 IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2022, pp. 1–6.

[5] J. Hoydis, F. A. Aoudia, S. Cammerer, M. Nimier-David, N. Binder, G. Marcus, and A. Keller, "Sionna RT: Differentiable Ray Tracing for Radio Propagation Modeling," *arXiv preprint arXiv:2303.11103*, 2023.