



Uavnoma: A UAV-NOMA Network Model under Non-Ideal Conditions

SOFTWARE METAPER

BRENA LIMA

NUNO FACHADA

RUI DINIS

DANIEL B. COSTA

MARKO BEKO

*Author affiliations can be found in the back matter of this article

]u[ubiquity press

ABSTRACT

Uavnoma is a set of Python functions and a front-end script for modeling, studying, and analyzing the communication system composed by an unmanned aerial vehicle (UAV) and two ground users. We assume that the UAV acts as an aerial base station to serve the users according to non-orthogonal multiple access (NOMA) principles. For more practical insights, residual hardware impairments (RHI) and imperfect successive interference cancellation (ipSIC) are considered. More specifically, **uavnoma** allows the modelers to study and visualize the system performance in terms of achievable rate and outage probability. Additionally, the package produces figures and tables showcasing results from the specified performance metrics.

CORRESPONDING AUTHOR:

Brena Lima

Lusófona University,
COPELABS, Lisbon, Portugal

brenaklima@gmail.com

KEYWORDS:

Non-orthogonal multiple access (NOMA); Non-ideal conditions; Numerical simulation; Performance analysis; Unmanned aerial vehicle (UAV)

TO CITE THIS ARTICLE:

Lima B, Fachada N, Dinis R, Costa DB, Beko M 2022 Uavnoma: A UAV-NOMA Network Model under Non-Ideal Conditions. *Journal of Open Research Software*, 10: 9. DOI: <https://doi.org/10.5334/jors.397>

(1) OVERVIEW

INTRODUCTION

The massive adherence of society to wireless communication technologies has enabled significant technological development in recent years, generating new scenarios and new challenges. Based on this, different network structures are needed to meet the heterogeneous requirements of the applications foreseen for the next generation of mobile communication. These different network infrastructures will impose stringent requirements to system designers, such as reliable communications, massive connectivity, low latency, low energy consumption, seamless connectivity, and high spectral efficiency. To achieve these heterogeneous requisites, numerous strategies and transmission technologies have been proposed. In this context, it is necessary to explore and understand the principles of the techniques and tools expected for beyond fifth-generation (B5G) networks, so that new solutions can be proposed. This work provides a package to auxiliary the study and formulation of system models related to two potential tools for B5G: non-orthogonal multiple access (NOMA) and unmanned aerial vehicles (UAV).

Since the design of appropriate multiple access techniques is crucial for supporting the massive number of devices foreseen for future networks, the NOMA technique is investigated in this work. NOMA is a crucial multiple access technique for supporting the massive number of devices foreseen for future networks, due to promising gains when compared to the orthogonal counterpart [1, 2]. The main idea of NOMA is to perform superposition coding (SC) at the transmitter and successive interference cancellation (SIC) at the receiver. Then, the users share the same resource block of time/frequency with distinct power levels.

In turn, the demand for UAV in communication networks has increased in the last few years [3]. UAVs working as aerial base stations (BSs) have also been identified as enablers of future-generation wireless networks due to the ability to perform complex tasks in diverse scenarios and flexible configurations. Due to these features, UAVs can significantly improve the connectivity and coverage range of communication networks. By combining the intrinsic features of NOMA and UAV, it is possible to achieve enormous gains for the overall network in terms of coverage, latency, fairness, capacity, spectral efficiency, and energy efficiency [4].

In the investigated scenario, a UAV acts as a BS which employs NOMA to communicate with two ground users. In order to provide practical insights, this work presents the **uavnoma** package, a Python simulation model of a downlink UAV-NOMA network with two users under the effect of the residual hardware impairments (RHI) and imperfect successive interference cancellation (ipSIC) on the network. In addition, Rician fading is taken into

account to model the line-of-sight (LoS) component to the Air-to-Ground (A2G) communication link between the UAV and ground users.

In summary, the **uavnoma** package provides a set of functions to generate the simulation parameters and to calculate the performance of the UAV-NOMA framework. The package also provides a front-end script to run simulations. The functions provided by **uavnoma** allow the user to:

- Study the UAV-NOMA network and use it as a basis for implementing other technologies.
- Modify the parameters and analyze the system's performance, considering the error propagation due to the RHI and the ipSIC in terms of outage probability and achievable rate.
- Visualize simulation output and produce tables and figures to evaluate the performance of the system.

These utilities were developed to study the modeling and performance of UAVNOMA networks under non-ideal conditions.

The remainder of this paper is organized as follows. The Section "Theoretical background" describes the system model under consideration. In the next Section, "Implementation and architecture", the architecture and organization of the package are presented. A usage example is discussed in Section "Usage Example", while the package's quality assurances are described in Section "Quality control". Section "Availability" provides information about the package's availability for different architectures, as well as its dependencies. Finally, the Section "Reuse potential" describes the potential extensions of the package.

THEORETICAL BACKGROUND

Scenario Let us consider a downlink UAV-aided NOMA network, as illustrated in Figure 1. In this scenario, a UAV is deployed as an aerial base station that communicates with $M = 2$ users with different channel conditions. We consider that each node is equipped with a single antenna and both UAV and users operate in the half-duplex mode. Furthermore, we assume that users subscribe to different traffic services that require a minimum data rate to be provided acceptably.

We assume that the UAV continuously flies with a constant velocity in a circular trajectory of radius r_1 and altitude H , and the users are randomly positioned in a region of radius r_2 . Based on three-dimensional Cartesian coordinates, the location of the UAV (x,y,z) and of the m -th user $(x_m,y_m,0)$, with $m \in \{1, 2\}$, can be represented respectively by $(r_1 \cos\theta_u, r_1 \sin\theta_u, h)$ and $(r_2 \cos\theta_m, r_2 \sin\theta_m, 0)$. The variables θ_u and θ_m are defined as the azimuth angle of the UAV and of the m -th user along their respective circles. The Euclidian distance between the UAV and the m -th user can be obtained by

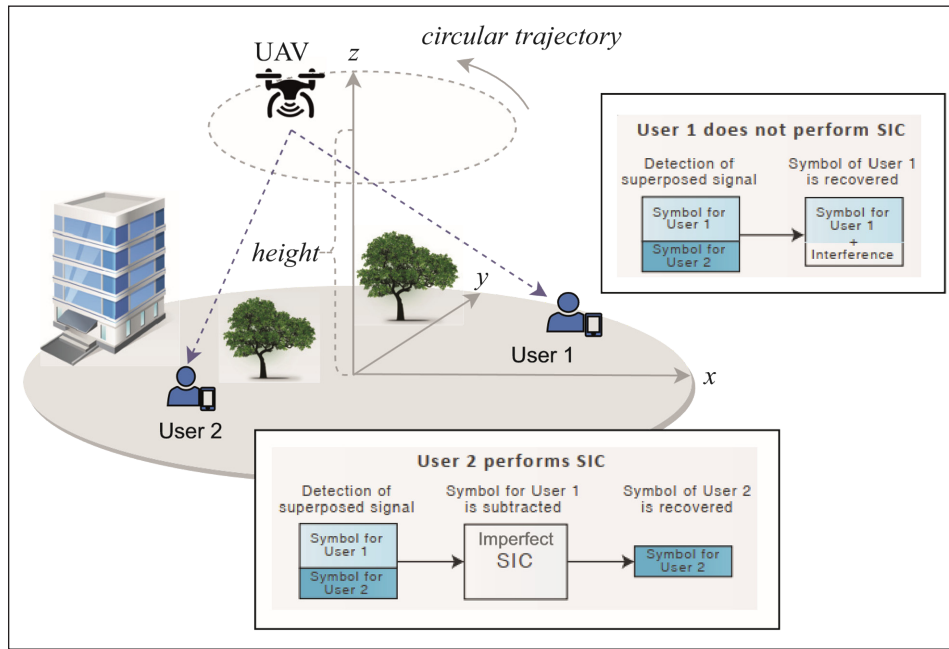


Figure 1 System model: simplified representation of UAV-NOMA transmission.

$$d_m = \sqrt{(x_m - x)^2 + (y_m - y)^2 + H^2}. \quad (1)$$

Channel Model Since the LoS path typically exists in our scenario, Rician fading is considered, and the channels between the UAV and the users are assumed to follow a Rician distribution.

A Rician fading channel is described using two parameters: 1) Rician factor (K), and, 2) the total power of the direct path and the scattered paths, $P_{LoS} = \mu^2 + 2\sigma^2$, where μ denotes the mean and σ denotes the standard deviation. Considering a given K and P_{LoS} , the μ and σ can be calculated, respectively, as follows

$$\mu = \sqrt{\frac{K}{K+1}} P_{LoS}, \quad (2)$$

and

$$\sigma = \frac{P_{LoS}}{\sqrt{2(K+1)}}. \quad (3)$$

The channel coefficients h_m from the UAV to the m -th user, modeled as complex Gaussian variables, are expressed as

$$h_m = \frac{g_m}{\sqrt{d_m^\psi}}, \quad (4)$$

where g_m denotes the deterministic LoS channel component modeled by complex Gaussian random variables, $g_m \sim \text{CN}(\mu, 2\sigma^2)$. The parameter ψ denotes the path loss exponent. Without loss of generality, it is assumed that users are ordered by the channel gain, i.e., $|h_1|^2 < |h_2|^2$, primary user < secondary user.

NOMA transmission According to the NOMA protocol, the transmission process starts with the UAV transmitting a superposed message $x = \sqrt{\alpha_1 P} s_1 + \sqrt{\alpha_2 P} s_2$ to all users by

superposition coding (SC), where P is the average transmit power of the UAV, s_1 is the message of the primary user, α_1 denotes the power allocation factor for the primary user message, s_2 is the message of the secondary user, and α_2 denotes the power allocation factor for the secondary user message, with $\alpha_1 + \alpha_2 \leq 1$ and $\alpha_1, \alpha_2 > 0$. In real deployments, due to many impairments, the network can experience residual hardware degradation.

In the receiver, the users employ the SIC technique. The SIC technique consists in decoding the strongest signal component first, which will be the signal of the primary user due to the power factor (α_1) allocated by the UAV. By contrast, the remaining signals are treated as interference. Specifically, the primary user tries to decode its message by treating the secondary user's message as noise. The more realistic signal-to-interference-plus-noise ratios (SINRs) at the primary user when we consider non-ideal conditions, such as RHI, can be expressed as

$$\gamma_1 = \frac{\rho |h_1|^2 \alpha_1}{\rho |h_1|^2 (\alpha_2 + \kappa^2) + 1}, \quad (5)$$

where $\rho = P/\lambda^2$ denotes the signal-to-noise ratio (SNR) at the UAV, and κ ($0 \leq \kappa \leq 1$) denotes the error propagation factor that models the impact of RHI. When $\kappa = 0$, perfect SIC is assumed. On the other hand, $\kappa = 1$ is the worst case, corresponding to a fully failed hardware scenario.

When SIC is employed by the secondary user, the process occurs successively until the secondary user decodes its respective message. First, the secondary user tries to decode the primary user's message by treating its message as noise and then removes the message from his observation successively. It is worth noting that perfect SIC is difficult to happen in practice. Then, as a result of the ipSIC at the users, a minimal error in the first detection will cause large errors due to the recursive characteristics

of the process. Based on the above, the more realistic SINR at the secondary user when we consider non-ideal conditions, such as RHI and ipSIC, can be expressed as

$$\gamma_2 = \frac{\rho |h_2|^2 \alpha_2}{\rho |h_2|^2 (\alpha_1 \epsilon + \kappa^2) + 1}, \quad (6)$$

where ϵ ($0 \leq \epsilon \leq 1$) denotes the error propagation factor that models the impact of ipSIC. When $\epsilon = 0$, perfect SIC is assumed, while $\epsilon = 1$ reflects the worst-case scenario.

Under this consideration, the instantaneous data rate achieved by the primary user can be written as

$$R_1 = \log_2(1 + \gamma_1). \quad (7)$$

The achievable rate by the secondary user can be expressed as

$$R_2 = \log_2(1 + \gamma_2). \quad (8)$$

By definition, the outage event can occur when the rate provided to any user is not enough to decode their data flow or the data flow of the weaker users. Based on this, we investigate the outage probability of received messages to characterize the link-level performance for the users in UAV-NOMA networks with non-ideal conditions, such as the effect of RHI and ipSIC, as described below.

The outage probability achieved by the primary user and secondary user can be expressed, respectively, as

$$P_1 = \Pr[R_1 < r], \quad (9)$$

and

$$P_2 = \Pr[R_2 < r], \quad (10)$$

where r denotes the target rate.

In turn, substituting (5) and (7) into (9), the outage probability of the primary user can now be expressed as

$$P_1 = \Pr \left[\log_2 \left(1 + \frac{\rho |h_1|^2 a_1}{\rho |h_1|^2 (\alpha_2 + \kappa^2) + 1} \right) < r \right], \quad (11)$$

Substituting (6) and (8) into (10), the outage probability of the secondary user can be rewritten as

$$P_2 = \Pr \left[\log_2 \left(1 + \frac{\rho |h_2|^2 a_2}{\rho |h_2|^2 (\alpha_1 \epsilon + \kappa^2) + 1} \right) < r \right], \quad (12)$$

IMPLEMENTATION AND ARCHITECTURE

The **uavnoma** package is implemented in Python, containing a front-end script and functions to model the UAV-NOMA system under non-ideal conditions. The script provides a command-line user interface, accepting model parameters, running the simulation, and displaying results using plots and a table, optionally exporting them to a CSV file. The functions are shown in [Figure 2](#).

- Design the initial parameters.
- Calculate the position of the UAV and users.
- Model the fading based on Rician Distribution.
- Generate the channel gain between UAV and users.
- Analyze system performance using as metrics the instantaneous achievable rate and outage probability.

The included front-end script allows anyone to experiment with the model without a deep knowledge of Python. The package provides the capability to exploit the main characteristics of the transmission model in which the UAV is used as an aerial base station. In the system, data is forwarded to users using the NOMA technique. To simulate a more realistic scenario, the effects of hardware impairments and imperfect SIC are assumed. In addition, it allows us to explore new scenarios, such as cooperative communication, full-duplex communication, and propose methodologies to improve system performance, such as energy allocation policies, trajectory optimization, energy harvesting, decoding order, and channel estimation.

As shown in [Figure 2](#), the package is organized in three modules, two of them containing library functions (**generate_values.py** and **performance_metrics.py**), and another representing the front-end script (**command_line.py**). The library functions allow the user to programmatically configure and run the UAV-NOMA system model. Besides allowing users to run simulations from the command-line, the front-end script is fully commented, also acting as a tutorial on how to use the library functions.

Specifically, the **generate_values.py** module has functions to generate data related to the position of the users and the UAV, fading distribution, and channel coefficient gain. More specifically, the function group is described as follows:

random_position_uav: this function provides a UAV position based on a 3D Cartesian plane due to the mobility and height of the UAV.

random_position_users: this function generates the position of the users based on a 2D Cartesian plane.

fading_rician: this function generates the parameters to model the fading based on Rician distribution to model the fading considering LoS effects.

generate_channel: this function calculates the specific distance between the users and the UAV, the small-scale fading, large-scale fading, and finally, it generates the channel coefficients. The channel gains are sorted to identify the primary user and secondary user.

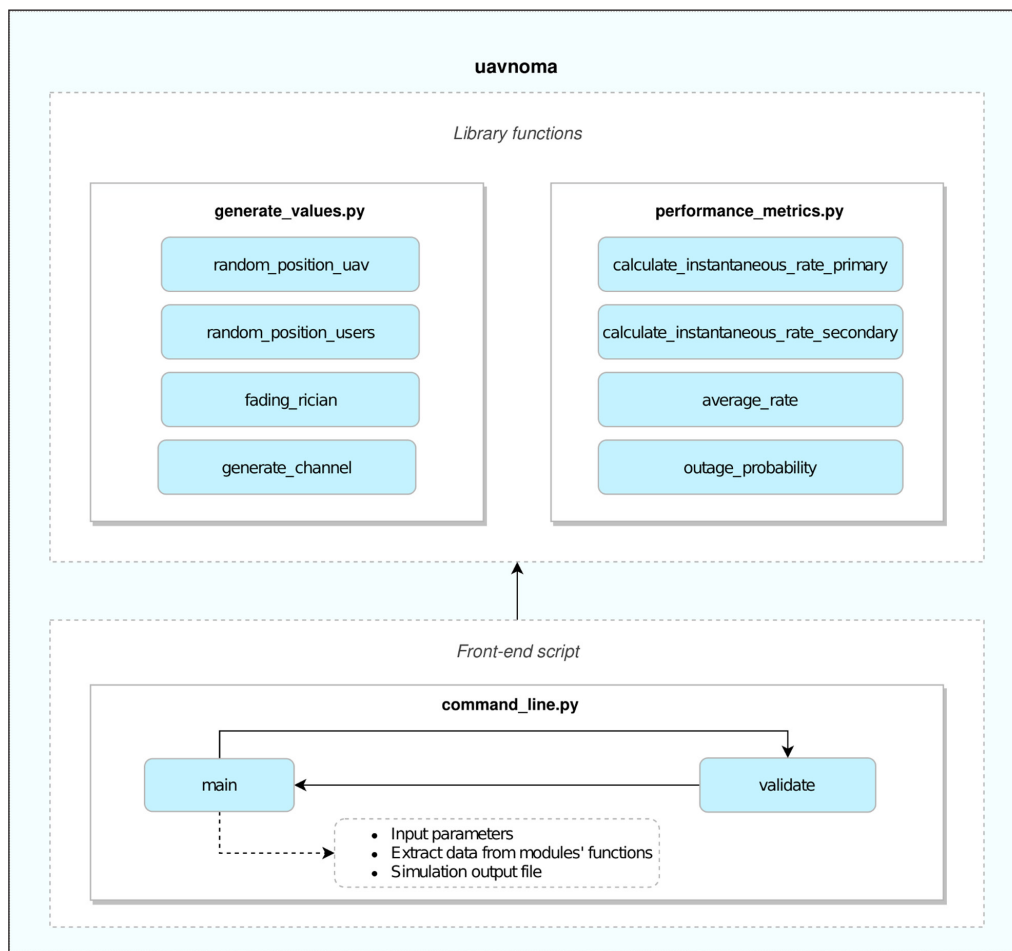


Figure 2 Structure of the **uavnoma** package. The package is organized into two main components, library functions and a front-end script. The library functions component is composed of two modules: `generate_values.py` and `performance_metrics.py`. The front-end script component is composed of only one module: `command_line.py`.

The second module, **performance_metrics.py**, extracts information about channel coefficients and uses it to calculate performance metrics. Specifically, the module is composed of the following functions:

`calculate_instantaneous_rate_primary`: this function calculates the achievable rate of the primary user. It calculates the SINR experienced by the primary user and based on the result, generates the achievable rate under non-ideal conditions of RHI.

`calculate_instantaneous_rate_secondary`: following the NOMA principles, this function returns the achievable rate of the secondary user under non-ideal conditions of RHI and ipSIC.

`average_rate`: this function provides the average achievable rate of the system.

`outage_probability`: from the obtained data about the user's rate, this function returns the system outage probability, the outage probability of the primary user, and the outage probability of the secondary user.

The **command_line.py** module contains the front-end script, which automates the analysis and presentation of the data obtained by the two library modules. The main function is called when the script is invoked with the **uavnoma** command. The validate function provides a verification of the user-specified parameters. Users can manipulate the behavior of the system by modifying the script's parameters. The script uses the library functions to perform a simulation and is able to produce plots and tables as shown in [Figures 3–5](#), as well as output results to a CSV file. The type of generated output is also controlled via the script's parameters, as discussed in the next section.

USAGE EXAMPLE

An example simulation was executed with the parameters shown in [Table 1](#), according to the following command:

```
$ uavnoma --seed 123 -s 50000 -p 2 -f 15 -l 2.2 -r
2 -ur 15 -uh 20 -t1 0.5 -t2 0.5 -hi 0.05 -si 0.05 -p1
0.8 -p2 0.2 --snr-min 10 --snr-max 60
```

The **uavnoma** command outputs the results displayed in [Figure 3](#) to the terminal, generating the plots shown in [Figures 4 and 5](#).

SIMULATION PARAMETERS	VALUE	SIMULATION PARAMETERS	VALUE
Monte Carlo samples	50000	Target Rate Secondary User	0.5
Power LoS	2.0	Residual Hardware Impairments	0.05
Rician Factor	15.0	Residual Imperfect SIC	0.05
Pathloss Exponent	2.2	Power coefficient of the Primary User	0.8
Radius UAV	2.0	Power coefficient of the Secondary User	0.2
Radius Users	15.0	Starting SNR in dB	10
UAV Height	20.0	Finishing SNR in dB	60
Target Rate Primary User	0.5	Seed for pseudo-random number generator	123

Table 1 Simulations parameters for the results shown in Figures 3–5.

SNR (dB)	Outage probability System	Outage probability Primary user	Outage probability Secondary user	Average achievable rate System	Average achievable rate Primary User	Average achievable rate Secondary User
10	1	1	1	0.00913646	0.0125802	0.00569266
12	1	1	1	0.0143964	0.0197969	0.00899597
14	1	1	1	0.0226109	0.0310296	0.0141923
16	1	1	1	0.035337	0.048342	0.0223319
18	1	1	1	0.0548176	0.0746368	0.0349985
20	1	1	1	0.0841256	0.113737	0.0545146
22	1	0.99914	1	0.127163	0.170183	0.0841419
24	1	0.97866	1	0.188345	0.248526	0.128165
26	0.99916	0.84808	0.99846	0.271809	0.351967	0.191651
28	0.97206	0.55078	0.96532	0.380123	0.480619	0.279626
30	0.8086	0.2544	0.78904	0.512824	0.63011	0.395537
32	0.45572	0.09226	0.42908	0.665391	0.791509	0.539274
34	0.14762	0.03054	0.12914	0.829331	0.953029	0.705632
36	0.03106	0.01074	0.02214	0.99364	1.10298	0.884297
38	0.00576	0.00374	0.0021	1.14724	1.23263	1.06185
40	0.00164	0.00146	0.00018	1.20143	1.33767	1.2252
42	0.00072	0.00072	0	1.39144	1.4181	1.36478
44	0.0003	0.0003	0	1.47656	1.47682	1.47629
46	0.00012	0.00012	0	1.5392	1.51809	1.5603
48	8e-05	8e-05	0	1.58342	1.54625	1.62059
50	2e-05	2e-05	0	1.61363	1.56503	1.66223
52	2e-05	2e-05	0	1.63378	1.57736	1.69019
54	0	0	0	1.64697	1.58536	1.70858
56	0	0	0	1.6555	1.5905	1.7205
58	0	0	0	1.66097	1.59379	1.72816
60	0	0	0	1.66446	1.59589	1.73304

Figure 3 Table provided by the front-end script when using the parameters shown in Table 1.

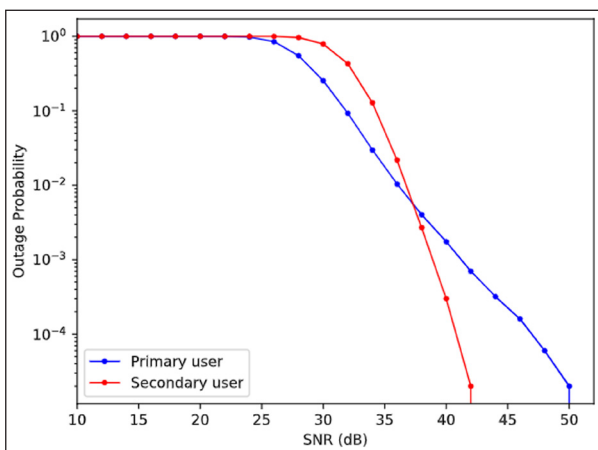


Figure 4 Plot of outage probability provided by the front-end script when using the parameters shown in Table 1.

To evaluate the impact of the outage probability, Figure 4 shows the outage probability performance of the primary and secondary users. From this figure, one can observe that for intermediate SNR values, the secondary user achieves more outage probability values. This effect is the result of the accumulation of inter-user interference during the successive decoding process on each user. Since the secondary user fails to decode and remove the primary user’s signal from their data, then interfering components will accumulate

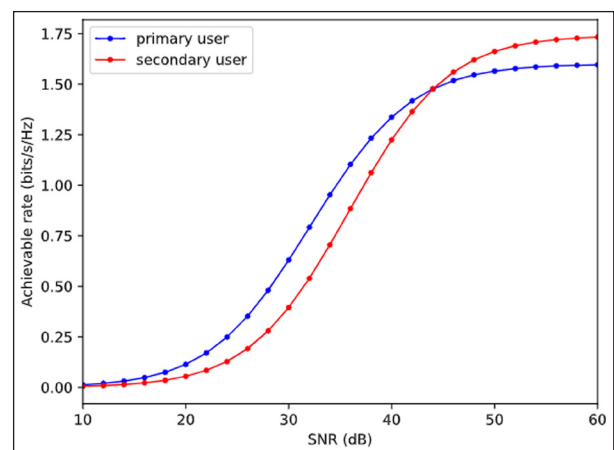


Figure 5 Plot of achievable rate provided by the front-end script when using the parameters shown in Table 1.

successively throughout the communication process. Thus, the level of interference experienced by the stronger user will be higher, which directly impacts his performance. In order to present more insights.

Figure 5 shows the achievable rate of the primary and secondary users. As expected, the rate curves corroborate the results obtained based on the outage probability figure. For high SNR values, the rate achieved by the secondary user becomes better than that of the

primary user, even with hardware imperfection and SIC. The parameters can be adjusted to investigate different behaviors of the achievable rate.

QUALITY CONTROL

All functions have been unit-tested for functionality and usability. Each of the simulation parameters was tested by providing a range of valid and invalid inputs and examining the results obtained. Additionally, a regression test is performed on the *uavnoma* command against previously validated results. These tests are integrated into the GitHub Actions continuous integration pipeline. That way, tests are automatically run each time new commits are pushed to GitHub.

(2) AVAILABILITY

OPERATING SYSTEM

Any system capable of running Python ≥ 3.8 .

PROGRAMMING LANGUAGE

Python 3.8, or higher.

DEPENDENCIES

- numpy
- matplotlib
- pandas
- tabulate
- argparse

When the package is installed in development mode, the required dependencies/plugins are also installed.

LIST OF CONTRIBUTORS

The software was created by Brena Lima and Nuno Fachada.

SOFTWARE LOCATION

Archive

Name: uavnoma

Persistent identifier: <https://doi.org/10.5281/zenodo.5467595>

License: MIT License

Publisher: Zenodo

Date published: 06/09/2021

Code repository

Name: uavnoma

Persistent identifier: <https://github.com/limabrena/uavnoma>

License: MIT License.

Date published: 06/09/2021

LANGUAGE

English.

(3) REUSE POTENTIAL

The functions in **uavnoma** package are suitable for future extensions of the system model. One of the main utilities of the model is the possibility for users to manually modify the parameters to analyze other scenarios, such as an urban micro or macrocell. Other benchmark parameters formats can be specified in `random_position_users` and `random_position_uav`. The functions can be integrated into other scenarios in order to investigate specific issues of UAV-NOMA communication. The `fading_rician` and `generate_channel` functions can be used for the investigation of power allocation, energy efficiency, LoS designing, and control of residual impairments. Once the modelers calculate the users' rate, the `outage_probability` function can be called to calculate the failure rate of the system for two users independently of the scenario.

Users can request support by opening an issue on GitHub.

FUNDING INFORMATION

This work is supported by Fundação para a Ciência e a Tecnologia under Grant No.: UIDB/04111/2020 (COPELABS).

COMPETING INTERESTS

The authors have no competing interests to declare.

AUTHOR AFFILIATIONS

Brena Lima  orcid.org/0000-0002-2804-1125

Lusófona University, COPELABS, Lisbon, Portugal

Nuno Fachada  orcid.org/0000-0002-8487-5837

Lusófona University, COPELABS, Lisbon, Portugal

Rui Dinis  orcid.org/0000-0002-8520-7267

Department of Electrical Engineering, NOVA University of Lisbon, Caparica, Portugal; Department of Electrical Engineering, NOVA University of Lisbon, Caparica, Portugal

Daniel B. Costa  orcid.org/0000-0002-5439-7475

Technology Innovation Institute, 9639 Masdar City, Abu Dhabi, United Arab Emirates

Marko Beko  orcid.org/0000-0001-7315-8739

Instituto de Telecomunicações, Instituto Superior Técnico, University of Lisbon, Portugal; COPELABS, Lusófona University, Lisbon, Portugal

REFERENCES

1. **Saito Y, Kishiyama Y, Benjebbour A, Nakamura T, Li A, Higuchi K.** Non-orthogonal multiple access(noma) for cellular future radio access. *IEEE 77th Vehicular Technology Conference*. 2013; 1–5. DOI: <https://doi.org/10.1109/VTCSpring.2013.6692652>
2. **Ding Z, Lei X, Karagiannidis GK, Schober R, Yuan J, Bhargava VK.** A survey on non-orthogonal multiple access for 5 g networks: Research challenges and future trends. *IEEE Journal on Selected Areas in Communications*. 2017; 2181–2195. DOI: <https://doi.org/10.1109/JSAC.2017.2725519>
3. **Zeng Y, Zhang R, Lim T.** Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Communication Magazine*. 2016; 54. DOI: <https://doi.org/10.1109/MCOM.2016.7470933>
4. **Liu Y, Qin Z, Cai Y, Gao Y, Li GY, Nallanathan A.** Uav communications based on non-orthogonal multiple access. *IEEE Wireless Communications*. 2019; 26(1): 52–57. DOI: <https://doi.org/10.1109/MWC.2018.1800196>

TO CITE THIS ARTICLE:

Lima B, Fachada N, Dinis R, Costa DB, Beko M 2022 Uavnoma: A UAV-NOMA Network Model under Non-Ideal Conditions. *Journal of Open Research Software*, 10: 9. DOI: <https://doi.org/10.5334/jors.397>

Submitted: 22 September 2021 **Accepted:** 09 September 2022 **Published:** 19 October 2022

COPYRIGHT:

© 2022 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

Journal of Open Research Software is a peer-reviewed open access journal published by Ubiquity Press.