Open Source Python Package for 5G MIMO Processing

Lucas Shibata, Francisco Müller and Aldebaro Klautau

***Abstract*— MIMO (multiple-input multiple-output) wireless communication systems play a fundamental role in 5G communications and beyond. This work describes the development of an open source Python package aimed to assist students and researchers in the study of 5G MIMO systems. Its current version includes the ability to estimate the capacity of a MIMO channel for different levels of CSI (channel-state information) available to the transmitter and receiver and the calculation of directivity patterns for antenna arrays. A brief discussion on the conversion of code developed originally in Matlab to Python is included. The major objective is to allow the community to explore, improve and expand this Python package for 5G MIMO processing.**

***Keywords*— MIMO, 5G, Python, capacity, antenna arrays.**

I. INTRODUCTION

Fifth generation (5G) mobile communication systems are already being deployed around the world. One of the main enabling technologies for 5G is the use of MIMO (multiple-input, multiple-output) technology [1].

Python is now one of the most used programming languages, due to its relative easy of use. In the communications field, Matlab is still used by many companies and research groups. One of the main reasons is the availability of decades of legacy code developed by said groups. But licensing costs for Matlab and its associated toolboxes can be too taxing. Many areas in engineering, like data mining, deep learning and image processing, already have a large amount of open source libraries written in Python. Libraries like NumPy [2], [3] turn the conversion of Matlab code to Python almost seamless. A logical next step is to expand the use of Python in the context of the new 5G communication systems. More specifically, in this work, for MIMO processing, like capacity estimation and antenna arrays characterization.

This works presents an open source Python package implementing important aspects of 5G communication systems, such as the calculation of capacity of a MIMO channel, uniform linear antenna (ULA) and uniform planar antenna (UPA) arrays characterization. The work is organized as follows: Section II presents an overview of the package development, organization and functionalities. Section III shows some results obtained using the Python package, with concluding remarks on Section IV.

II. PACKAGE OVERVIEW

Most of the code currently included in the package was originally developed using Matlab by members of the Telecommunications, Automation and Electronics Research and Development Center (LASSE), based on MIMO techniques available in the literature [4]–[7]. Matlab is proprietary and closed-source software, developed by Mathworks [8]. On the other hand, Python is free software, open source and more used than Matlab [9], allowing the community to contribute and innovate. This makes it easier for researchers to have codes shared and executed by anyone. In this paper, we leverage these advantages of the Python language to develop our package.

To port the codes originally written in Matlab, as well as developing new code, the authors used Python libraries such as NumPy [2] and Matplotlib [10]. These packages widely used in the language ecosystem, are stable and have well organized documentation, allowing more accessibility to developers and users. The version currently used is Python 3.7.7 [11]. NumPy is used to read and manipulate arrays and external files, in addition to their math library. The Matplotlib library is mainly used for plotting graphs, as illustrated in Section III.

A. Package Functionalities

Currently, the proposed Python package allows for the calculation of some fundamental systems parameters for MIMO communications, including the capacity estimate for a given MIMO channel, which can be read from an external file, and the radiation properties for uniform antenna arrays.

Regarding the radiation properties, the package include functions capable of generating the radiation patterns for different uniform antenna arrays. This mathematical tool represents the radiation properties in function of the spatial coordinates, allowing the user to calculate power flux density, radiation intensity, field strength, directivity, phase or polarization of the antenna array [5], [6]. Support for uniform linear antenna arrays (ULA), uniform planar antenna arrays (UPA) and uniform circular antenna arrays (UCA) is implemented in our package. In Matlab, the same functionality is included in a separate paid toolbox [12].

In addition, the Python package includes functions to estimate the capacity of MIMO channels. These channels can be manually entered by the user or read from files. For instance, HDF5 format support is available. Three capacity estimation techniques are currently implemented in the package: equal-power, single-mode and eigenbeamforming [4]. The equal power method assumes the transmitter has no knowledge of the MIMO channel matrix H, therefore all the transmitting antennas have the same input power. Its capacity is given by:

$$C_{equal-power} = \log_2(|I_{N_r} + \frac{\rho}{N_t} HH^*|) \quad (1)$$

Lucas Shibata, Francisco Müller and Aldebaro Klautau are with LASSE - Telecommunications, Automation and Electronics Research and Development Center, Belém-PA, Brazil. E-mails: lucas.shibata@itec.ufpa.br, and {fmuller, aldebaro}@ufpa.br.

where N_t and N_r are the numbers of transmit and receive antennas, respectively; I_{N_r} the identity matrix of dimension $N_r \times N_r$, ρ is the signal-to-noise (SNR) power ratio, $*$ is the transpose conjugate operator, λ_i is the i-th eigenvalue of the HH^H, where i is the i-th transmit antenna index and $\mathrm{E}\left\{|s_i|^2\right\}$ is the variance of the transmitted symbols.

The other two MIMO methods assume both transmit and receive ends know H. The MIMO capacity when using the eigenbeamforming method can be calculated as

$$C_{eig} = \sum_{i=1}^{r} \log_2(1 + \rho \mathrm{E}\left\{|s_i|^2\right\} \lambda_i) \qquad (2)$$

where r is the rank of H. The capacity for the single-mode MIMO method is given by

$$C_{single\_mode} = \log_2(1 + \rho\lambda_1). \qquad (3)$$

The Python Package for 5G MIMO Processing described in this paper and its associated documentation can be accessed at [13].

III. Results

In order to demonstrate the functionalities of the proposed Python library, some numerical results are explored in this section. Fig. 1 shows the estimated channel capacities assuming $N_t = 4$ and $N_r = 3$ using the methods described in Section II for a given MIMO channel H and SNR values varying from -20 to 20 dB.

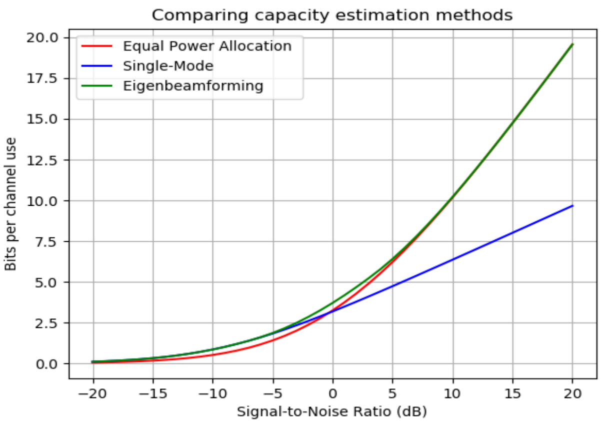

Fig. 1. Channel capacity estimates for a MIMO channel representing a communication system with 4 transmit antennas and 3 receive antennas.

The next result is the directivity pattern of an ULA, as shown in Fig. 2. The antenna array itself was calculated through the use of the geometrical channel model described in [6]. In this example, the wireless channel was estimated using a ray-tracing modeling software called Wireless InSite [14] and imported into the simulation environment, an additional functionality provided by the proposed Python package.

IV. Conclusions

The research and development of MIMO systems for 5G networks is an hot topic currently. This work presents a Python library seeking to accelerate and optimize this study, providing

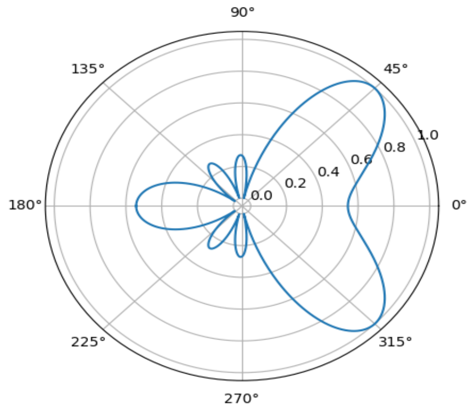

Fig. 2. Directivity pattern for an uniform linear antenna array with 4 transmit antenna elements spaced half-wavelength apart.

an open and free tool for anyone to use and contribute, making it easier to share results with the community. The current features include MIMO channel capacity estimation and antenna array characterization. The move from Matlab to Python was smooth due to the libraries developed by the Python community, which include many of the functionalities provided by Matlab. The open source Python package for 5G MIMO processing can be freely accessed and the authors plan to keep supporting it, with the inclusion of new code developed by our research group and by the community.

Acknowledgements

This work was supported by the Innovation Center, Ericsson Telecomunicações S.A. and CNPq, Brazil.

References

[1] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The next generation wireless access technology*. Academic Press, 2018.
[2] “Numpy reference.” [Online]. Available: https://numpy.org/
[3] “Numpy for Matlab users.” [Online]. Available: https://numpy.org/doc/stable/user/numpy-for-matlab-users.html
[4] J. Hampton, *Introduction to MIMO Communications*. Cambridge University Press, 2014.
[5] C. Balanis, *Antenna Theory*, 3rd ed. Wiley-interscience, 2005.
[6] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
[7] A. Ali, N. Gonzalez-Prelcic, and R. W. Heath, “Millimeter wave beam-selection using out-of-band spatial information,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, p. 1038–1052, 2018.
[8] “Matlab.” [Online]. Available: https://www.mathworks.com/help/matlab/index.html
[9] B. Weber, “Matlab vs Python: Why and how to make the switch,” 2020. [Online]. Available: https://realpython.com/matlab-vs-python/
[10] “User’s guide - Matplotlib 3.2.1 documentation.” [Online]. Available: https://matplotlib.org/users/index.html
[11] “Python 3.7.7 documentation.” [Online]. Available: https://docs.python.org/3.7/
[12] “Phased array system toolbox.” [Online]. Available: https://www.mathworks.com/products/phased-array.html
[13] “5G MIMO Processing Python Package repository,” 2020. [Online]. Available: https://gitlab.lasse.ufpa.br/software/mimo-python/-/tree/suggestion
[14] “Wireless InSite: 3D Wireless Prediction Software,” 2020. [Online]. Available: https://www.remcom.com/wireless-insite-em-propagation-software