



## **Examples of Using pyAerial**

# Table of contents

Running a PUSCH link simulation

---

LDPC encoding-decoding chain

---

Dataset generation by simulation

---

Dataset generation for LLRNet

---

LLRNet model training

---

Channel estimation on transmissions captured using Aerial Data Lake

---

Decoding PUSCH transmissions captured using Aerial Data Lake

---

We provide a number of examples of using NVIDIA pyAerial for GPU-accelerated 5G NR signal processing, and for machine learning experiments. The examples are in Jupyter notebook format. The notebooks here are pre-executed, but they can be also interactively run following the instructions in [Getting Started with pyAerial](#).

## Running a PUSCH link simulation

The first example shows how to use pyAerial for modeling 5G NR compliant PUSCH transmission and reception. In this example, the whole PUSCH pipeline is modeled within pyAerial, using the cuPHY library as a backend for GPU acceleration.

The notebook shows two ways of running the PUSCH receiver pipeline: In the first, the user only needs to make a single call using the Python API, and the whole PUSCH receiver is run. In the other, the PUSCH receiver pipeline is split into its different receiver components, each called separately using the Python API. This approach enables replacing any of the PUSCH receiver components for example by an AI/ML model, and benchmarking that against the conventional receiver.

NVIDIA Sionna is used in the example for radio channel modeling.

- [Using pyAerial to run a PUSCH link simulation](#)

## LDPC encoding-decoding chain

The second example gives an example of using cuPHY's GPU accelerated 5G NR LDPC encoding and decoding chain (including also rate matching) modules through the pyAerial Python API. The encoding/decoding modules are expected to be useful for example in AI/ML model validation when implementing some parts of the receiver using machine learning.

- [Using pyAerial for LDPC encoding-decoding chain](#)

## Dataset generation by simulation

- [Using pyAerial for data generation by simulation](#)

This notebook generates a fully 5G NR compliant PUSCH/PDSCH dataset using pyAerial. The cuPHY library is used through its Python bindings in pyAerial for PUSCH/PDSCH slot generation, and NVIDIA Sionna is used for radio channel modeling. The PUSCH/PDSCH slots get generated and transmitted through different radio channels.

The example stores the dataset for use in the consequent LLRNet examples. Equally well the data could be generated on the fly during simulation.

## Dataset generation for LLRNet

In this example, pyAerial is used to generate a log-likelihood ratio dataset based on the PUSCH/PDSCH dataset generated in the previous example. Using pyAerial, the complete PUSCH receiver chain is formed, and LLR data is collected after the channel equalizer. The log-likelihood ratio data is used to train an LLRNet model in the next example. LLRNet, published in

Shental, J. Hoydis, *“Machine LLRning: Learning to Softly Demodulate”*,  
<https://arxiv.org/abs/1907.01512>

is a simple neural network model that takes equalizer outputs, i.e. the complex-valued equalized symbols, as its input and outputs the corresponding log-likelihood ratios (LLRs) for each bit, basically replacing the conventional soft demapper in the receiver chain.

**Note:** This notebook requires that the former example in [Dataset generation by simulation](#) has been run first.

- [LLRNet: Dataset generation](#)

## LLRNet model training

In this example, the LLR data from the previous example is used to train and validate an LLRNet model for computing log-likelihood ratios. The trained LLRNet is plugged in the PUSCH receiver chain, replacing the conventional soft demapper, and its performance is validated. The model also gets exported into ONNX format consumed by the NVIDIA TensorRT inference engine.

The example shows essentially how to use pyAerial for validating a component of the PUSCH receiver chain, and how to export a model in a format that is ready to be integrated in a real system.

**Note:** This notebook requires that the former example in [Dataset generation for LLRNet](#) has been run first - that generates the data for this notebook.

- [LLRNet: Model training and testing](#)

## Channel estimation on transmissions captured using Aerial Data Lake

This example shows how to query PUSCH data from an Aerial Data Lake database and perform channel estimation on that PUSCH data using pyAerial.

**Note:** This notebook requires that the clickhouse server used by Aerial Data Lake is running, and that the example data has been imported into a database. Refer to the Aerial Data Lake docs on how to do this.

- [Using pyAerial for channel estimation on Aerial Data Lake data](#)
- For more information, refer to the [Aerial Data Lake section](#)

## Decoding PUSCH transmissions captured using Aerial Data Lake

- [Using pyAerial for PUSCH decoding on Aerial Data Lake data](#)

Similarly to the previous example, this example illustrates the use of pyAerial in context of Aerial Data Lake. In this example, the PUSCH data queried from the database is run through a full PUSCH receiver pipeline implemented using the pyAerial API. The example also illustrates how the pyAerial PUSCH components enable fetching intermediate results from the receiver pipeline.

**Note:** Similarly to the previous notebook, this notebook requires that the clickhouse server used by Aerial Data Lake is running, and that the example data has been imported into a database. Refer to the Aerial Data Lake docs on how to do this.

- For more information, refer to the [Aerial Data Lake section](#)

© Copyright 2024, NVIDIA.. PDF Generated on 06/06/2024