



PUSCH

This module contains classes related to the Physical Uplink Shared Channel, PUSCH.

```
class aerial.phy5g.pusch.pusch_rx.PuschRx
```

PUSCH receiver pipeline.

This class implements the whole PUSCH reception pipeline from the received OFDM post-FFT symbols to the received transport block (along with CRC check).

```
__init__(cell_id, num_rx_ant, num_tx_ant, num_ul_bwp=273, num_dl_bwp=273, mu=1, enable_cfo_correction=0, enable_to_estimation=0, enable_pusch_tdi=0, eq_coeff_algo=1, ldpc_kernel_launch=aerial.pycuphy.PuschLdpcKernelLaunch.PUSCH_RX_ENABLE_DRIVER_L
```

Initialize PuschRx.

Parameters

- **cell_id** (*int*) – Physical cell ID.
- **num_rx_ant** (*int*) – Number of receive antennas.
- **num_tx_ant** (*int*) – Number of transmit antennas.
- **num_ul_bwp** (*int*) – Number of PRBs in a uplink bandwidth part. Default: 273.
- **num_dl_bwp** (*int*) – Number of PRBs in a downlink bandwidth part. Default: 273.
- **mu** (*int*) – Numerology. Values in [0, 3]. Default: 1.
- **enable_cfo_correction** (*int*) –
Enable/disable CFO correction:
 - 0: Disable (default).
 - 1: Enable.
- **enable_to_estimation** (*int*) –

Enable/disable time offset estimation:

- 0: Disable (default).
- 1: Enable.

- **enable_pusch_tdi** (*int*) –

Time domain interpolation on PUSCH.

- 0: Disable (default).
- 1: Enable.

- **eq_coeff_algo** (*int*) –

Algorithm for equalizer coefficient computation.

- 0 - ZF.
- 1 - MMSE (default).
- 2 - MMSE-IRC.

- **ldpc_kernel_launch** (*PuschLdpcKernelLaunch*) – LDPC kernel launch method.

Return type

None

```
run(rx_slot, num_ues, slot=0, num_dmrs_cdm_grps_no_data=2, dmrs_scrm_id=41,
start_prb=0, num_prbs=273, dmrs_syms=None, dmrs_max_len=2, dmrs_add_in_pos=1,
start_sym=2, num_symbols=12, scids=None, layers=None, dmrs_ports=None,
rntis=None, data_scids=None, mcs_tables=None, mcs_indices=None, code_rates=None,
mod_orders=None, tb_sizes=None, rvs=None, ndis=None)
```

Run PUSCH Rx.

This runs the cuPHY PUSCH receiver pipeline for a single UE group sharing the same time-frequency resources, i.e. having the same PRB allocation, and the same start symbol and number of allocated symbols. Default values get filled for the parameters that are not given.

Parameters

- **rx_slot** (*numpy.ndarray*) – A tensor representing the receive slot buffer of the cell.
- **num_ues** (*int*) – Number of UEs in the UE group.
- **slot** (*int*) – Slot number.
- **num_dmrs_cdm_grps_no_data** (*int*) – Number of DMRS CDM groups without data [3GPP TS 38.212, sec 7.3.1.1]. Value: 1->3.
- **dmrs_scrm_id** (*int*) – DMRS scrambling ID.
- **start_prb** (*int*) – Start PRB index of the UE group allocation.
- **num_prbs** (*int*) – Number of allocated PRBs for the UE group.
- **dmrs_syms** (*List[int]*) – For the UE group, a list of binary numbers each indicating whether the corresponding symbol is a DMRS symbol.
- **dmrs_max_len** (*int*) – The *maxLength* parameter, value 1 or 2, meaning that DMRS are single-symbol DMRS or single- or double-symbol DMRS.
- **dmrs_add_in_pos** (*int*) – Number of additional DMRS positions.
- **start_sym** (*int*) – Start OFDM symbol index for the UE group allocation.
- **num_symbols** (*int*) – Number of symbols in the UE group allocation.
- **scids** (*List[int]*) – DMRS sequence initialization for each UE [TS38.211, sec 7.4.1.1.2].
- **layers** (*List[int]*) – Number of layers for each UE.
- **dmrs_ports** (*List[int]*) – DMRS ports for each UE. The format of each entry is in the SCF FAPI format as follows: A bitmap (mask) starting from the LSB where each bit indicates whether the corresponding DMRS port index is used.
- **rntis** (*List[int]*) –

- **data_scids** (*List[int]*) – Data scrambling IDs for each UE, more precisely *dataScramblingIdentityPdsch* [TS38.211, sec 7.3.1.1].
- **mcs_tables** (*List[int]*) – MCS table to use for each UE (see TS 38.214).
- **mcs_indices** (*List[int]*) – MCS indices for each UE.
- **code_rates** (*List[float]*) – Code rate for each UE. This is the number of information bits per 1024 coded bits.
- **mod_orders** (*List[int]*) – Modulation order for each UE.
- **tb_sizes** (*List[int]*) – TB size in bytes for each UE.
- **rvs** (*List[int]*) – Redundancy versions for each UE.
- **ndis** (*List[int]*) – New data indicator per UE.

Returns

A tuple containing:

- *np.ndarray*: Transport block CRCs.
- *List[np.ndarray]*: Transport blocks, one per UE.

Return type

np.ndarray, *List[np.ndarray]*

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