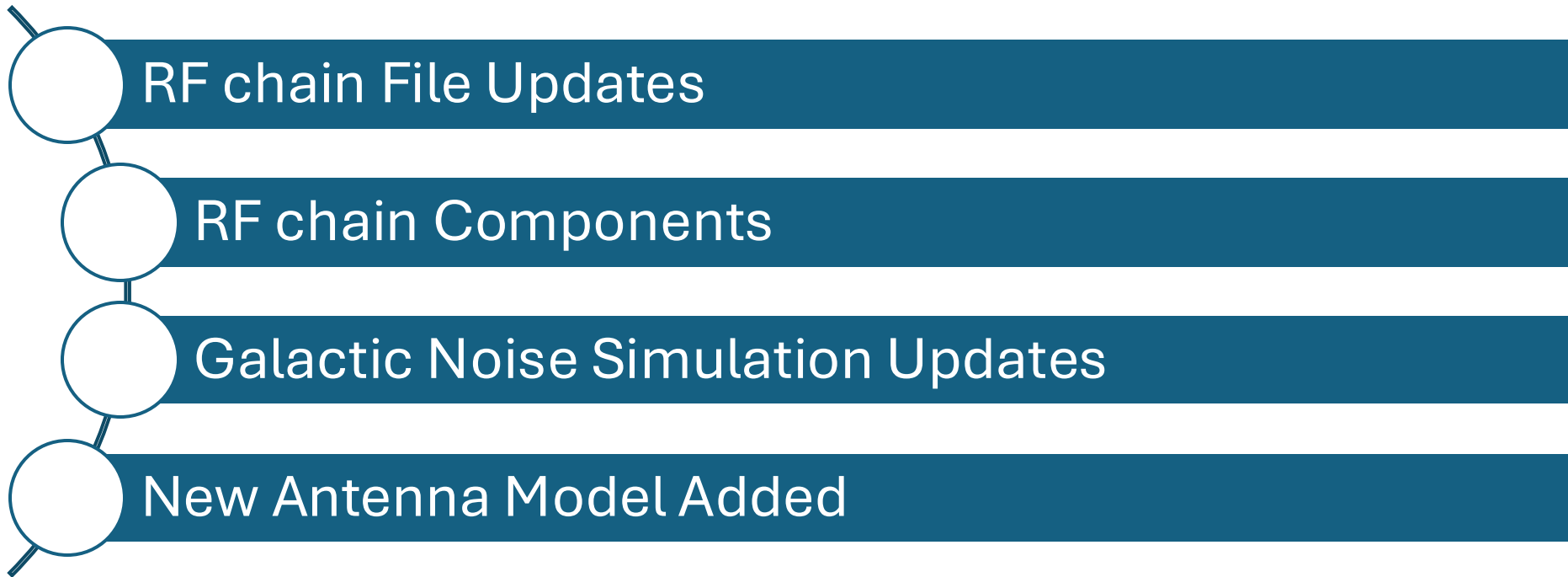


RFChain Updates

George Vittakis, Stavros Nonis, Antonios Leisos, Kostas Papageorgiou

GRAND Meeting, 02-06 June/2025 Warsaw

OVERVIEW



New Data Files

- RF chain v2

- /sps/grand/RFChain_XD/Rfchain_20250302_updated/

- Balun before Matcing Network: balun13in20230612.s2p

- Matching_network: MatchingNetworkX.s2p, MatchingNetworkY.s2p, MatchingNetworkZ.s2p

- LNA: LNA-X.s2p, LNA-Y.s2p, LNA-Z.s2p

- Cable_connector: cable+Connector.s2p

- Amfilter+Feb: feb+amfitler+biast.s2p

- Balun before ADC : balun_before_ad.s2p

- Zin_antenna : Z_ant_3.2m.csv

- Effective-length: EFL_data_3.2m_XYZ_20241220.mat

- Galactic Noise Calculations

- /sps/grand/RFChain_XD/RFchain_20250302_updated/PG_ALL_jifen.mat {file.shape(221x3x24)}

Files provided by Pengfei, XinXu

Updated RF chain Scripts

- Path /home/grand/grand/sim/detector/rf_chain2.py
- In the same format as of the existing file rf_chain.py but calculates the transfer functions using rf chain version 2 files which means
 - Balun before Matching Network: balun13in20230612.s2p
 - Matching_network: MatchingNetworkX.s2p, MatchingNetworkY.s2p, MatchingNetworkZ.s2p
 - LNA: LNA-X.s2p, LNA-Y.s2p, LNA-Z.s2p
 - Cable_connector: cable+Connector.s2p
 - Amfilter+Feb: feb+amfilter+biast.s2p
 - Balun before ADC : balun_before_ad.s2p
 - Zin_antenna : Z_ant_3.2m.csv
 - Effective-length: EFL_data_3.2m_XYZ_20241220.mat
- The new arrangement of the elements in the RF chain v2 is:
Balun1, Matching Network, LNA, Cables+Connector, Amfilter+Feb, balun2

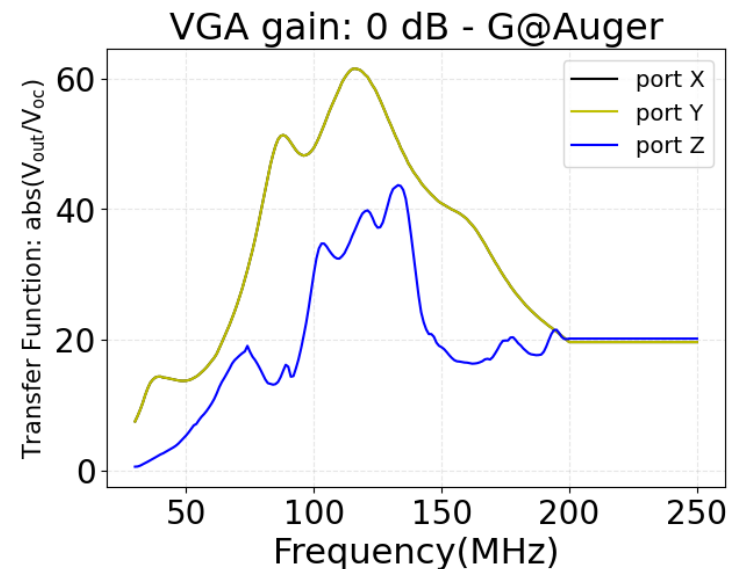
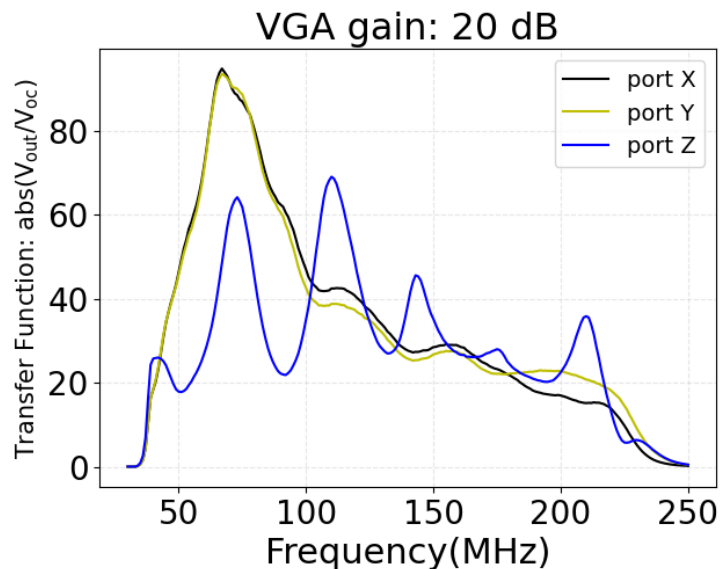
Updated RF chain Scripts

The scripts `plot_rf_chain.py` (</home/grand/scripts/>) and `convert_efield2voltage.py` (</home/grand/scripts/>) use the RF chain v2.

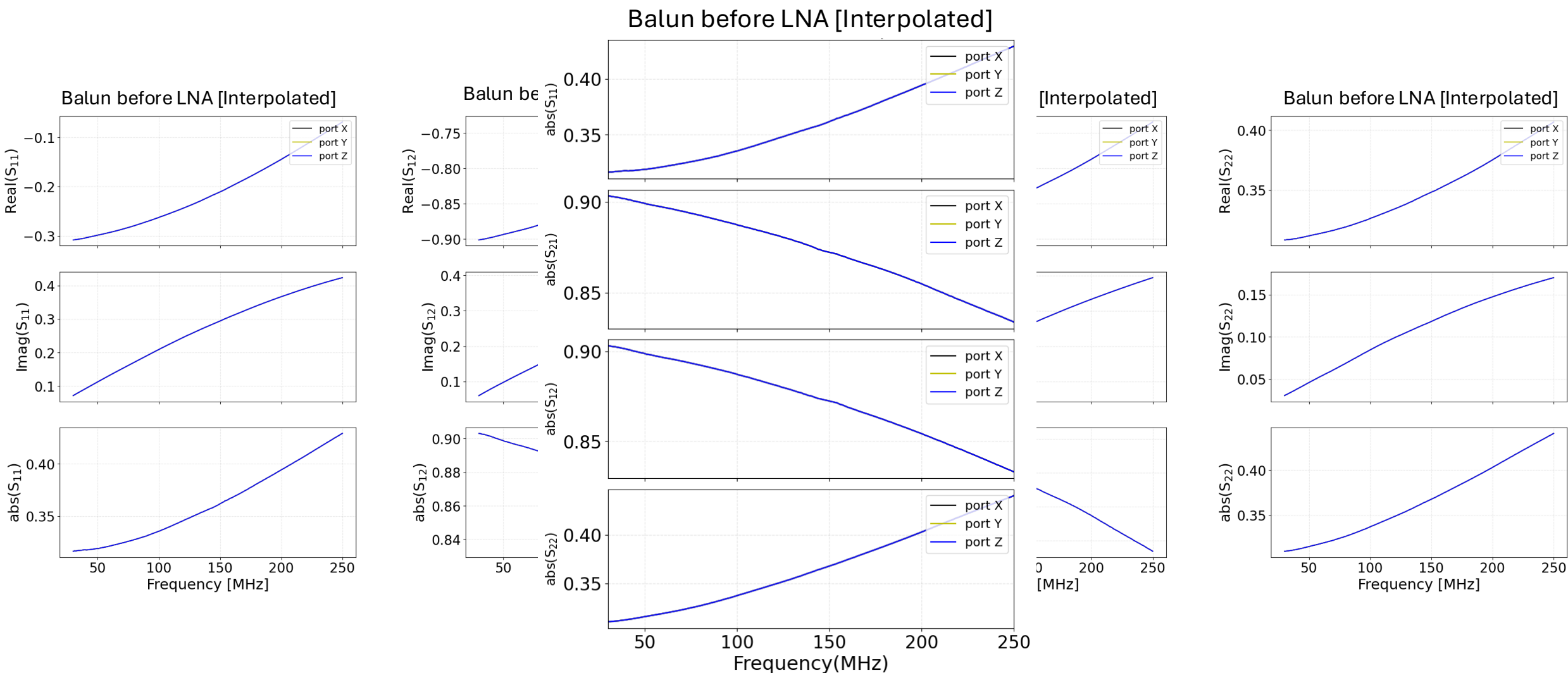
The scripts can be used with the Grand@Auger transfer functions.

Example: `$ python3 plot_rf_chain.py rf_chain_gaa --savefig (default=False)`

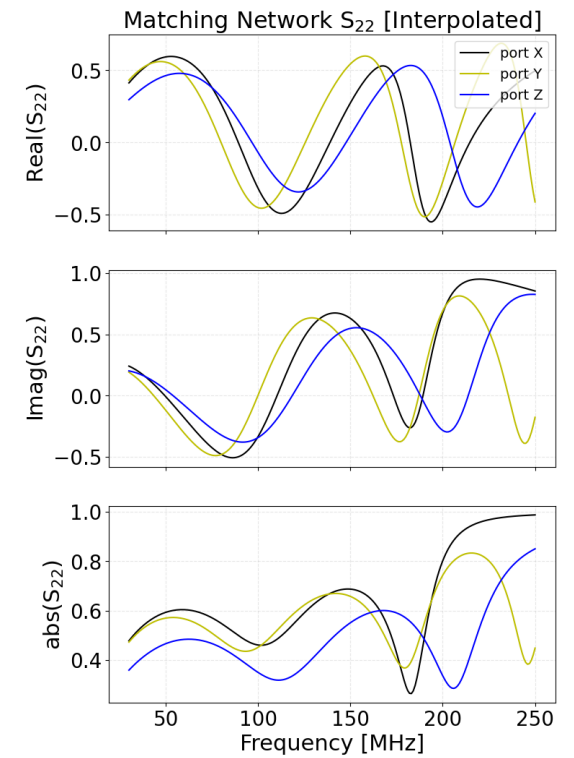
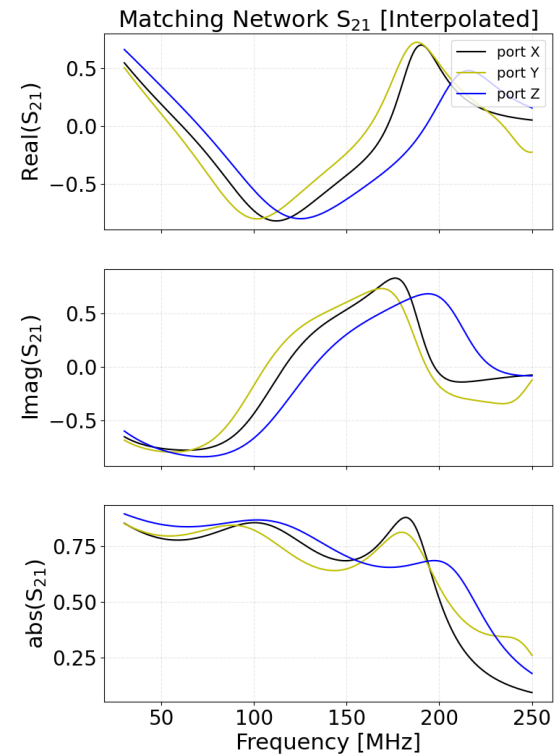
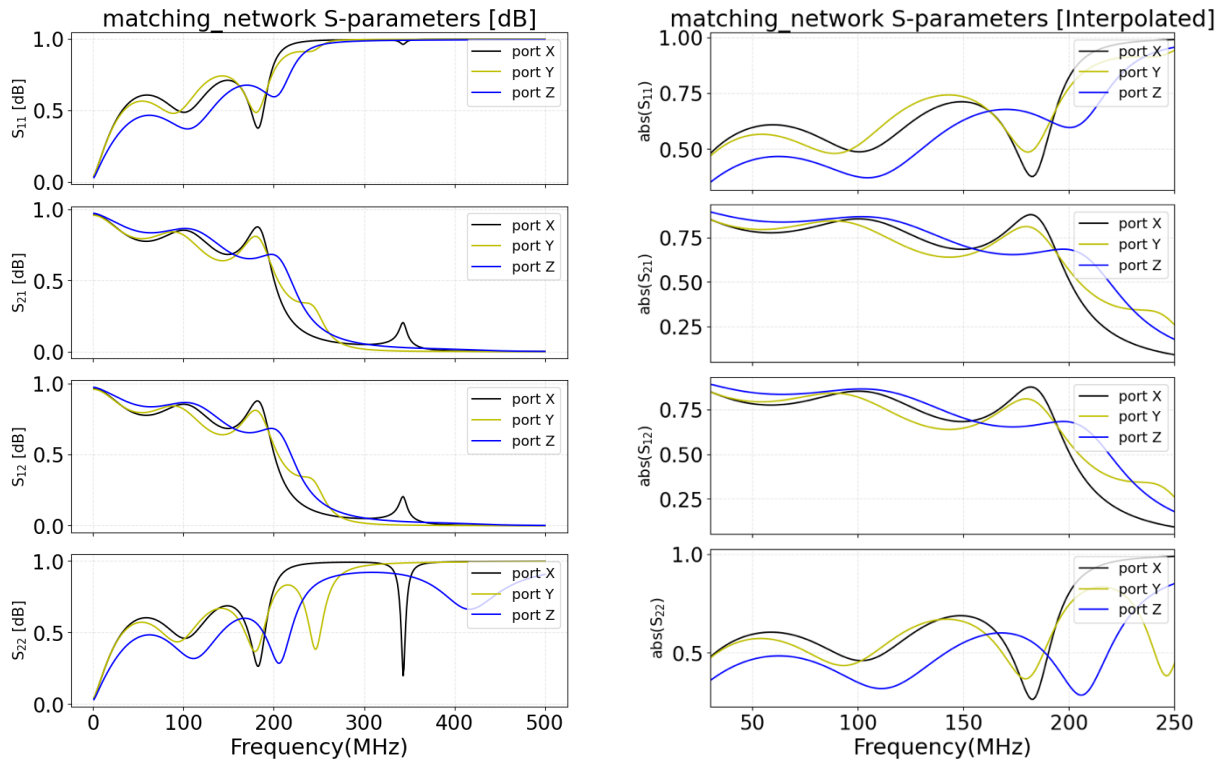
Example: `$ python3 convert_efield2voltage.py test_efield.root -o test_voltage.root --no_rf_chain - rf_chain_gaa`



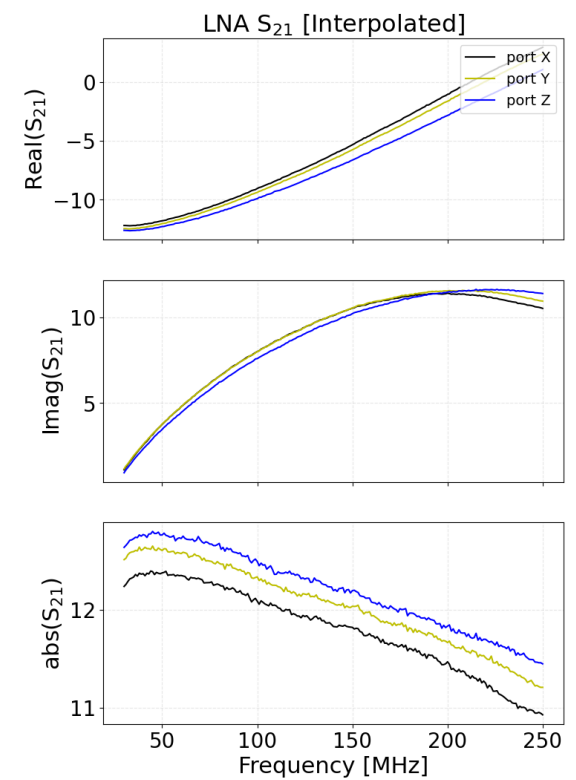
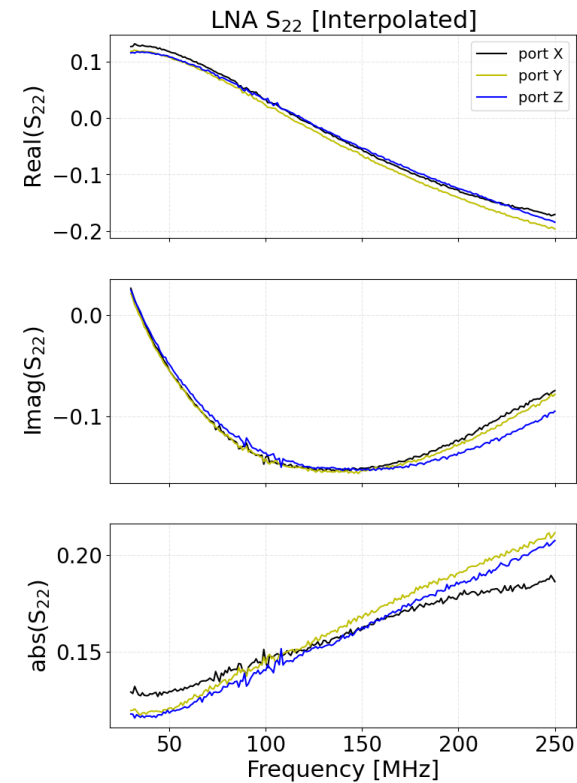
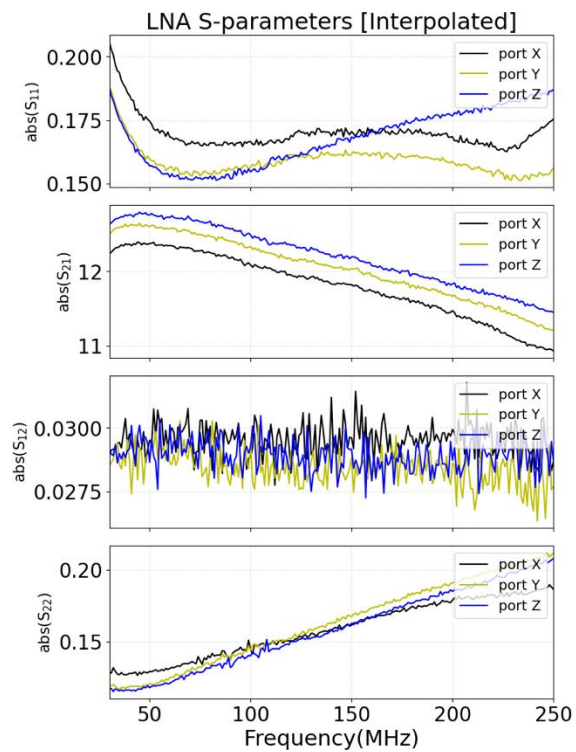
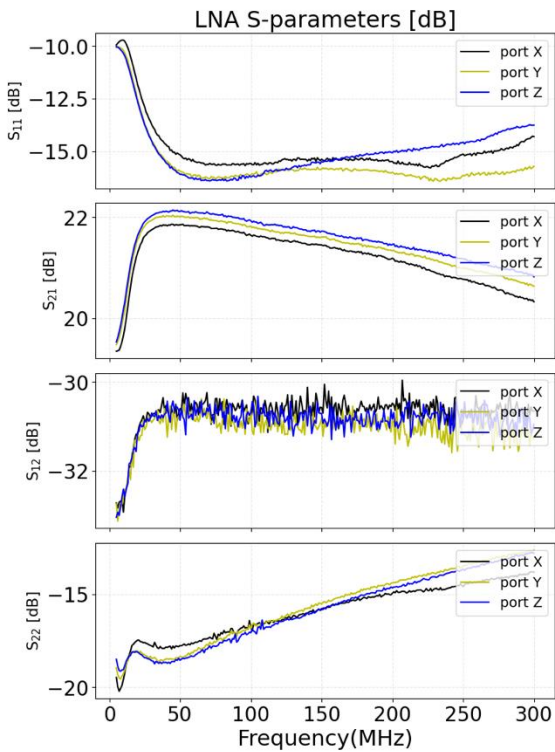
RF chain Components



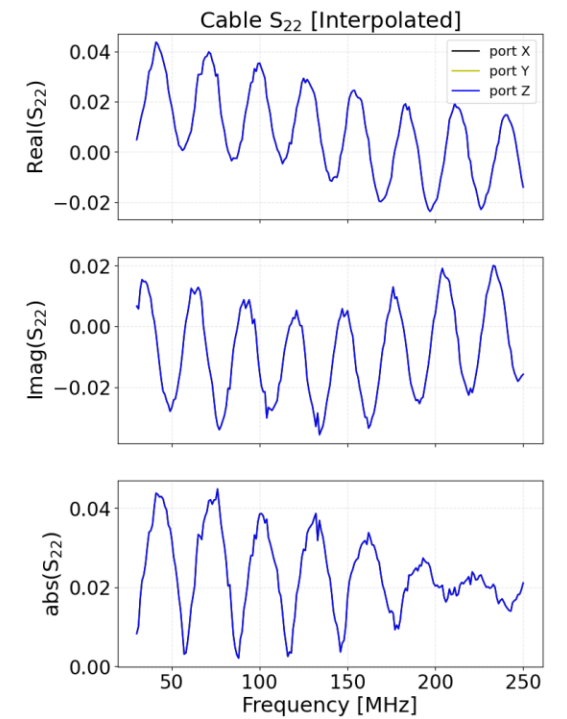
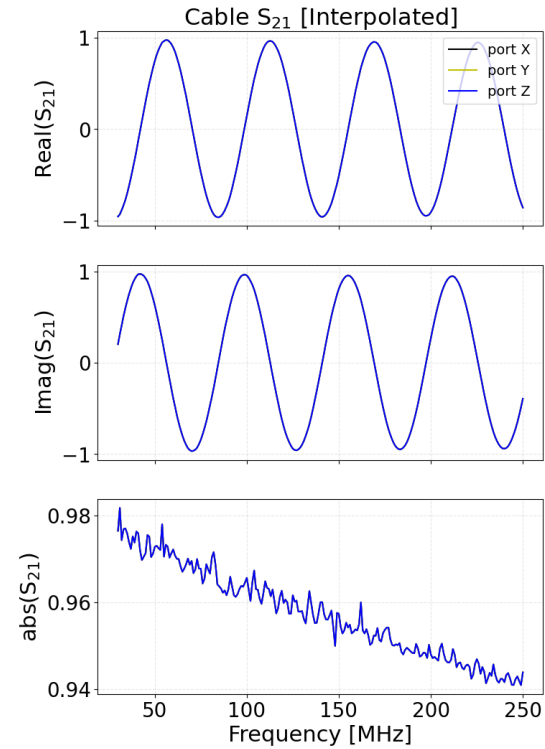
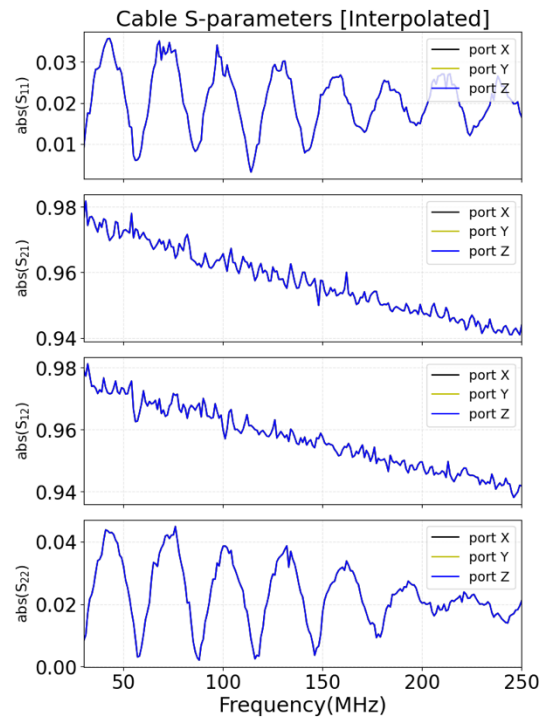
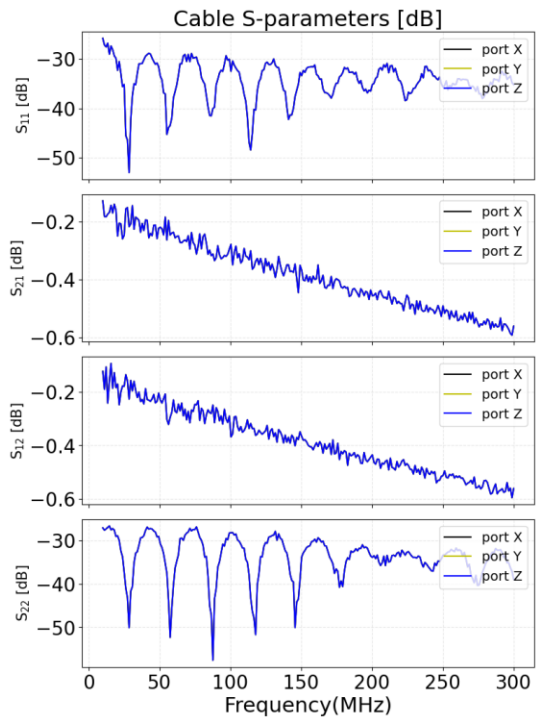
RF chain Components



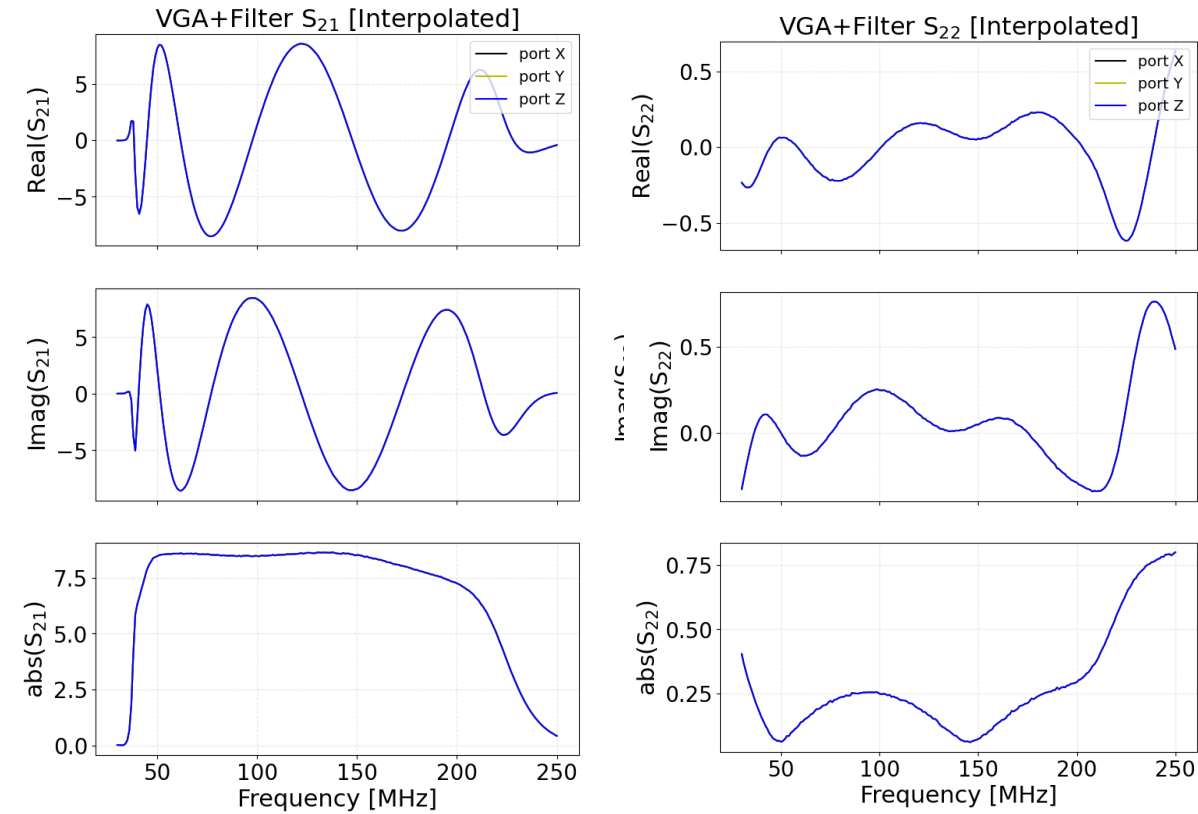
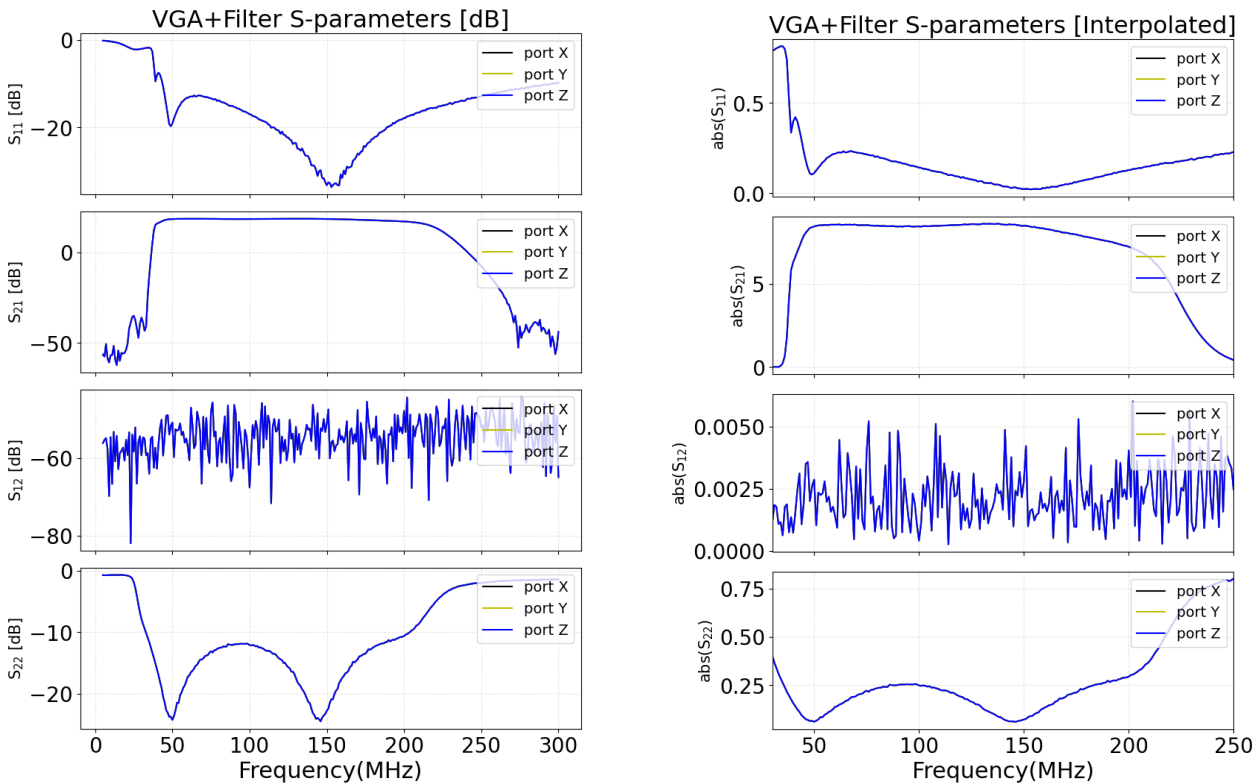
RF chain Components



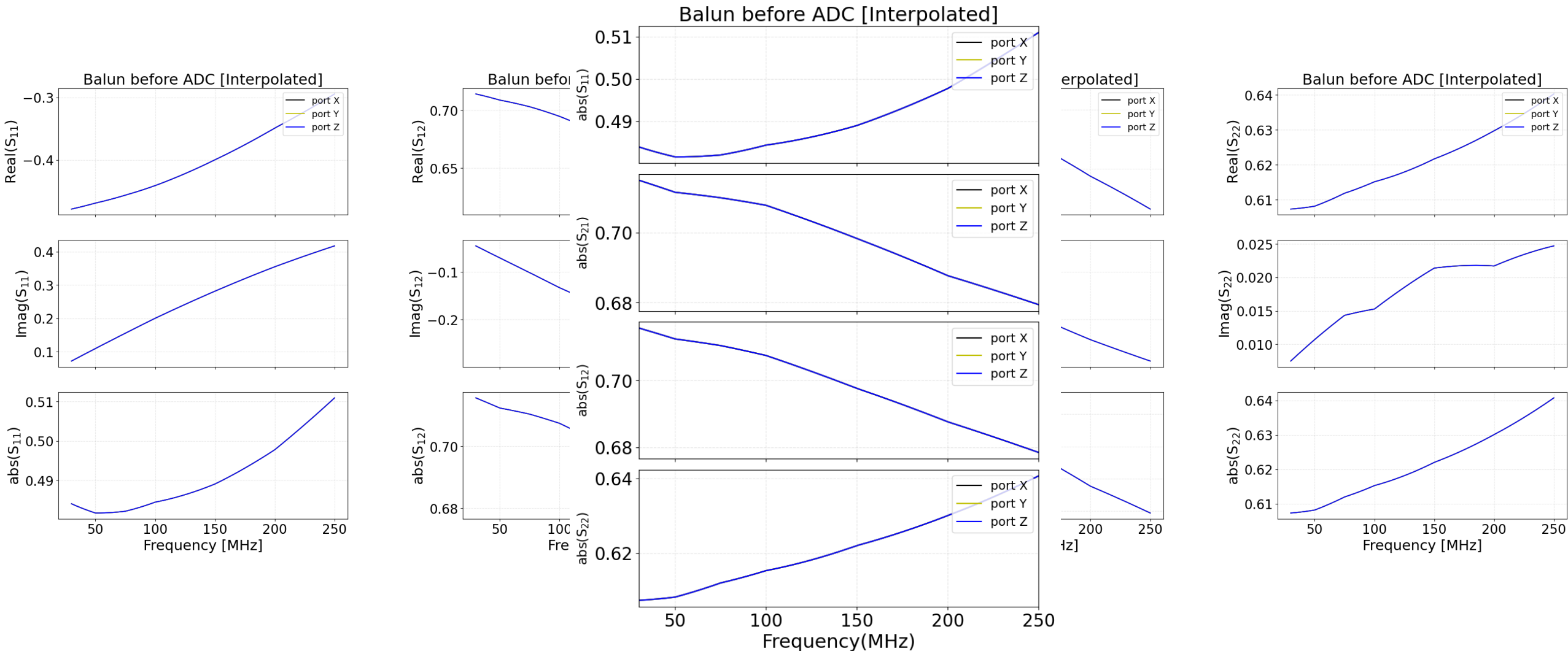
RF chain Components



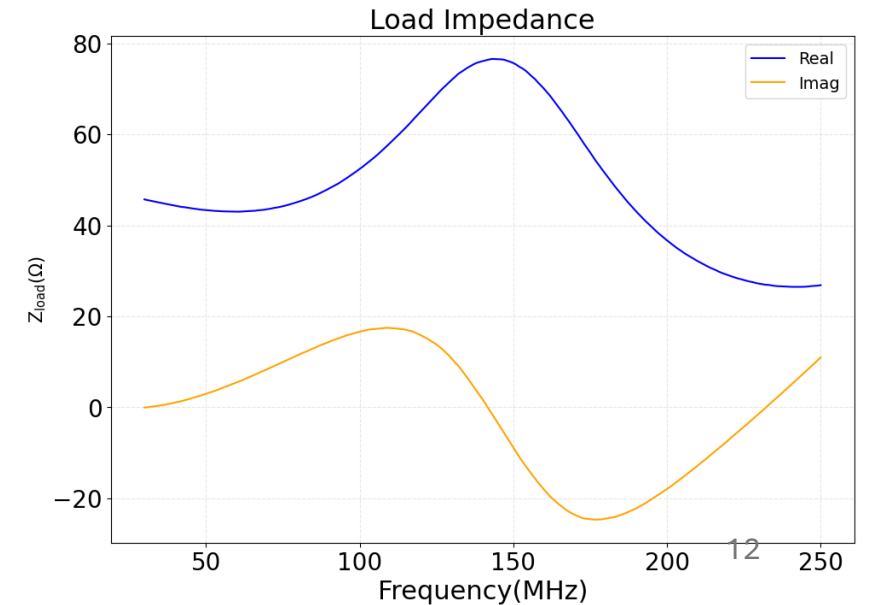
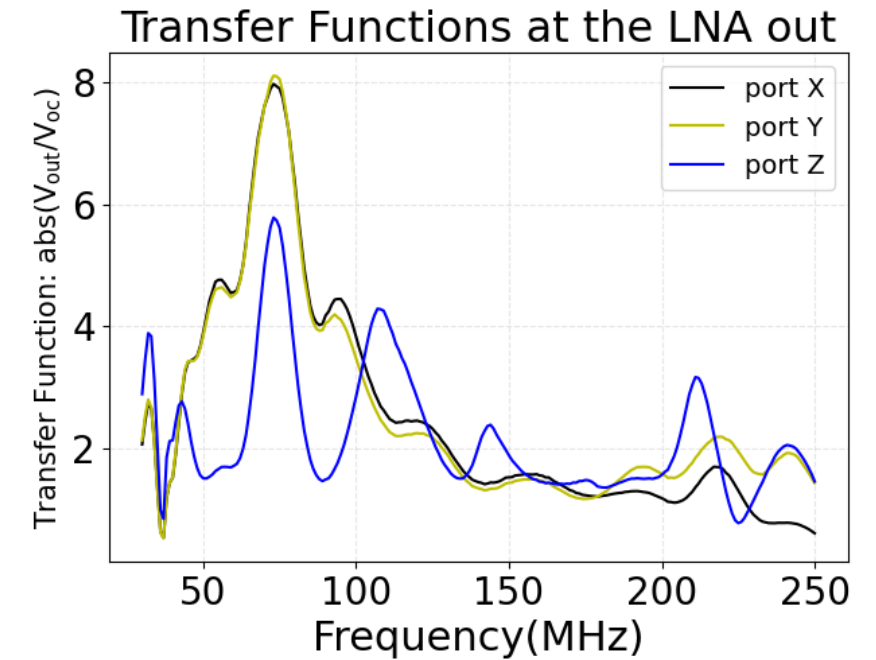
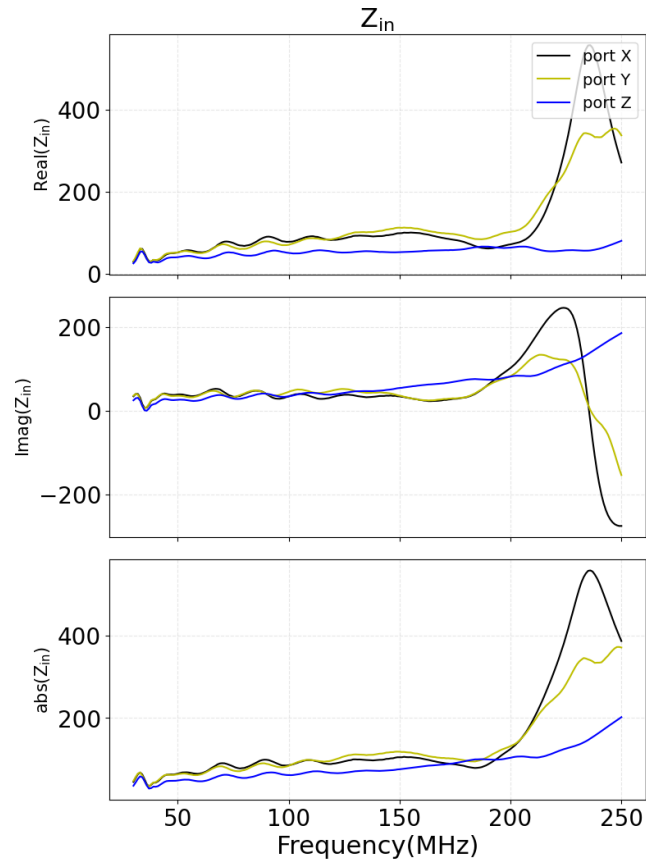
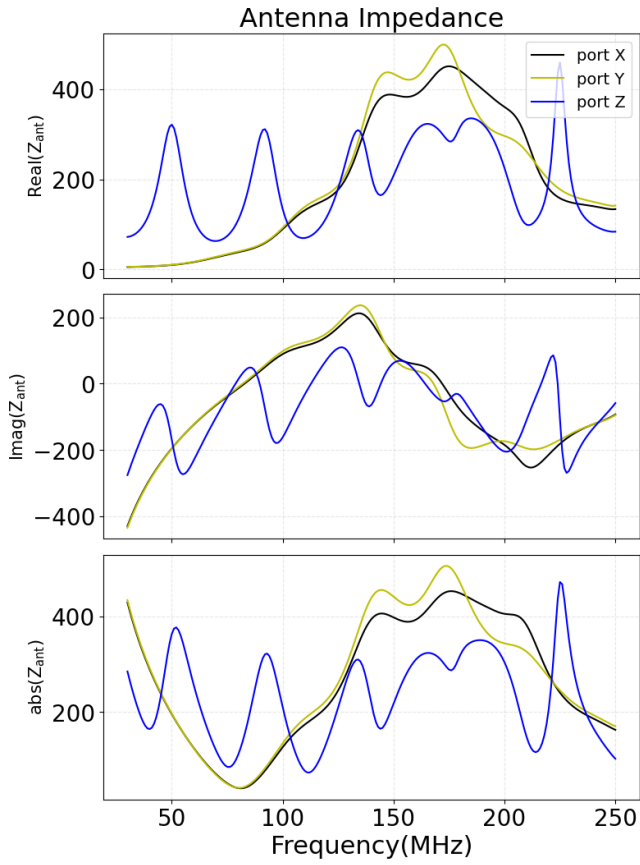
RF chain Components



RF chain Components



RF chain Components



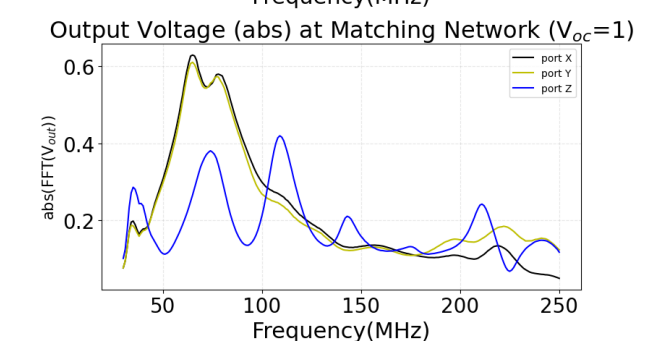
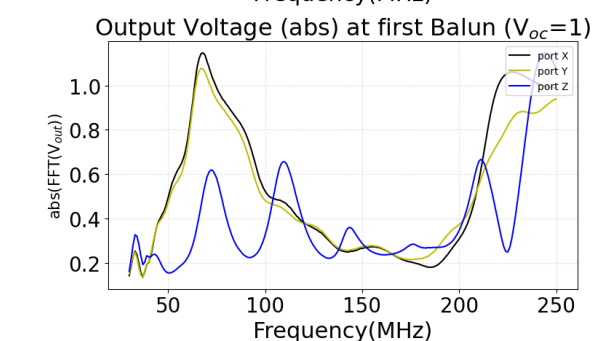
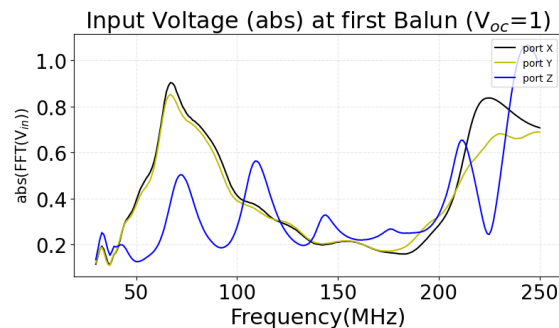
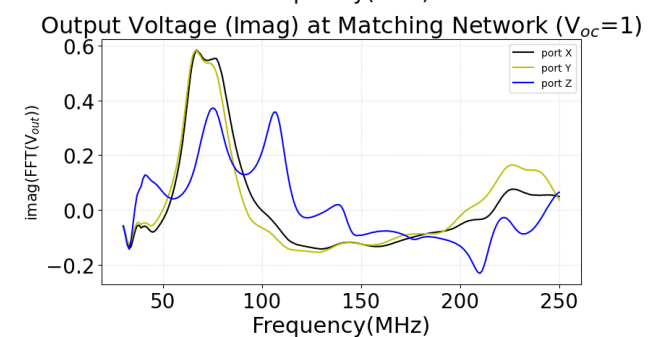
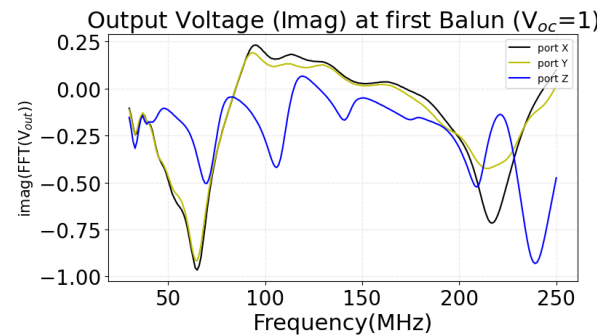
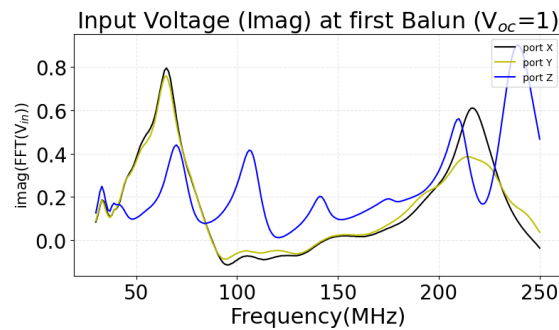
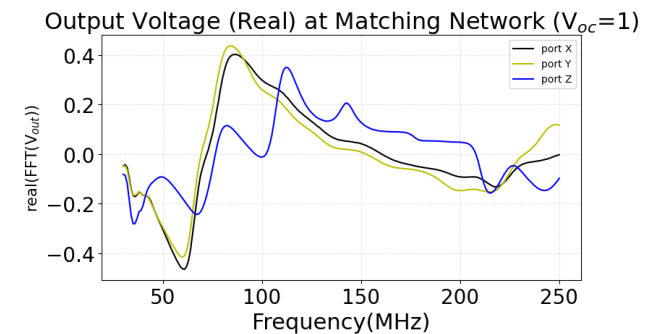
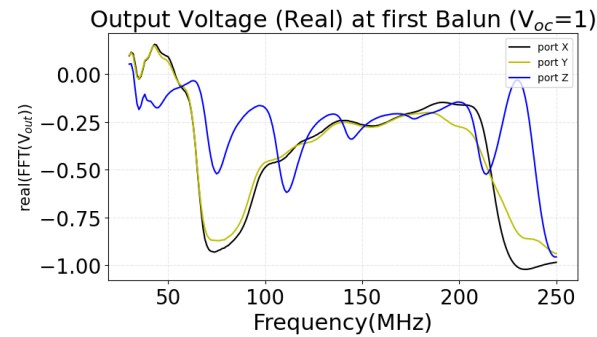
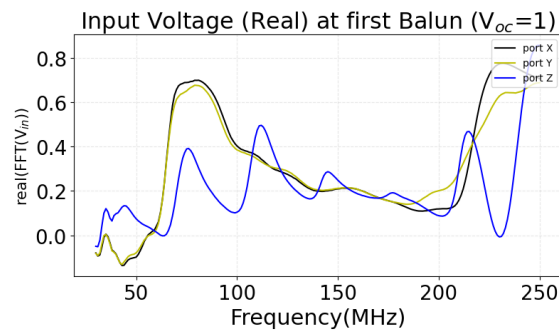
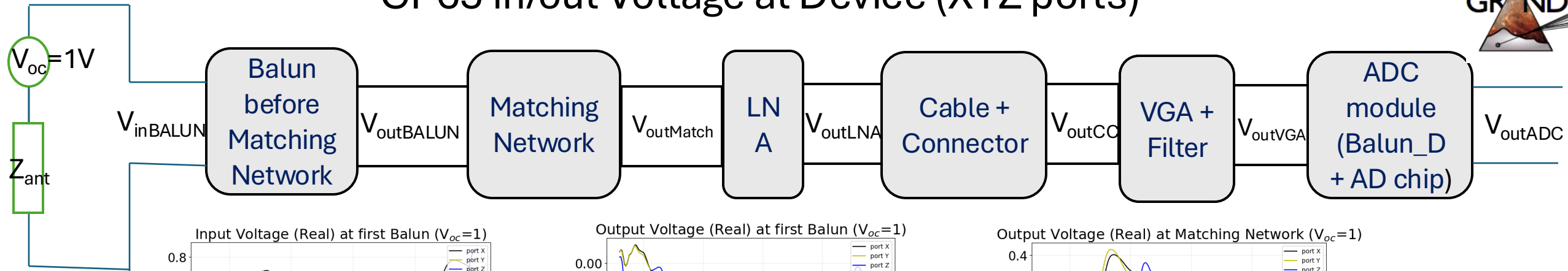
Updated RF chain Scripts

Two scripts `Compute_Vout_AT_Device_save.py` (</home/grand/scripts/>) and `plot_Vout_AT_Device.py` (</home/grand/scripts/>) were created with which one can calculate and plot the output voltage at each element of the chain as well as the ratio $V_{\text{out}}/V_{\text{in}}$.

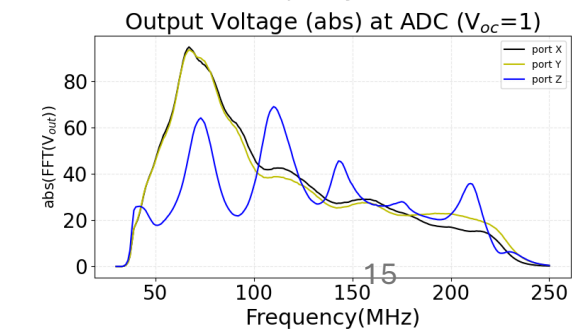
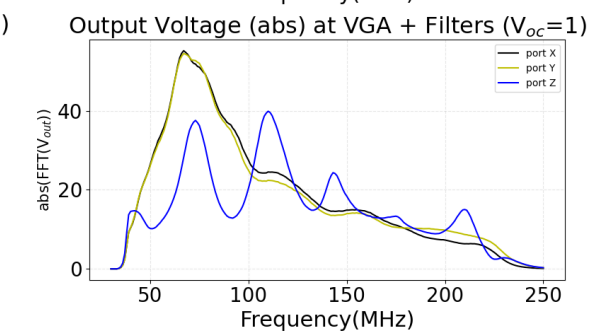
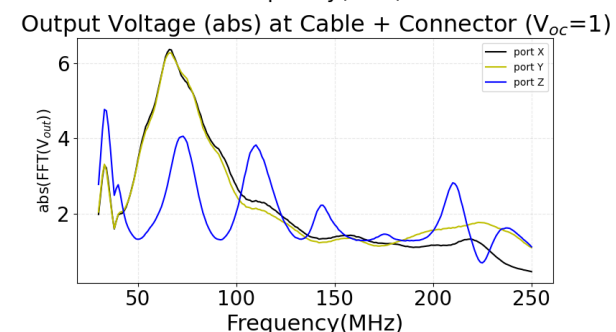
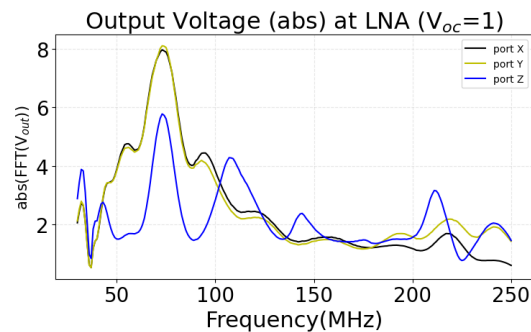
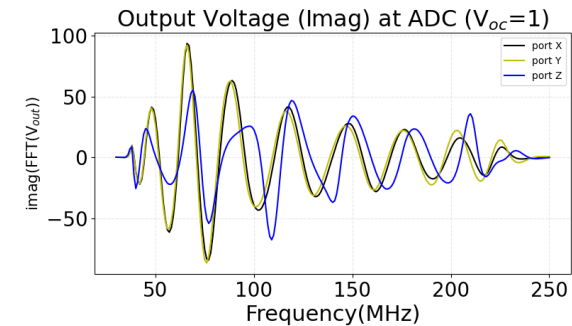
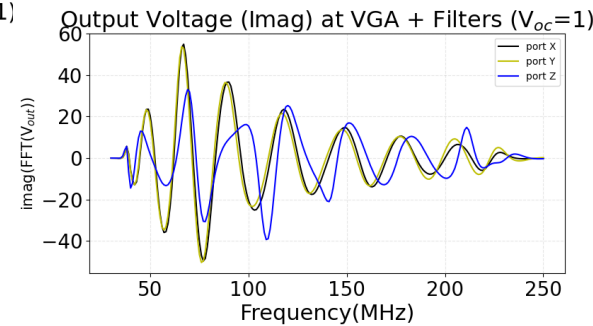
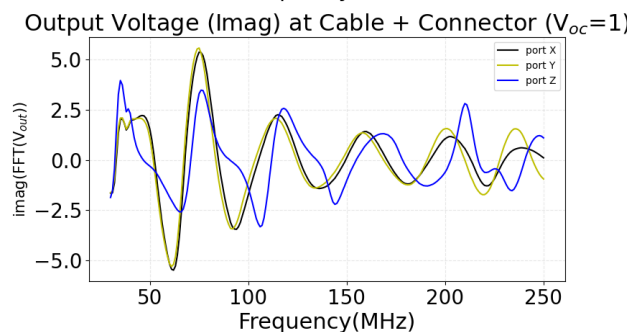
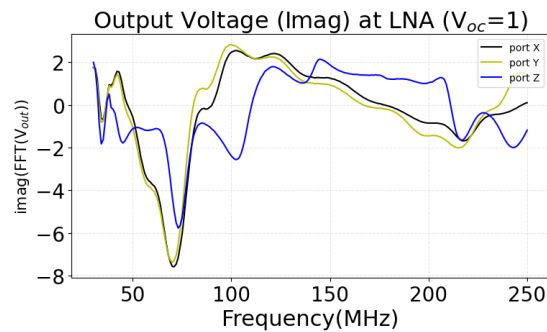
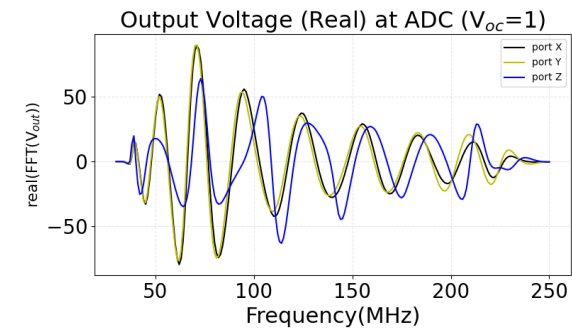
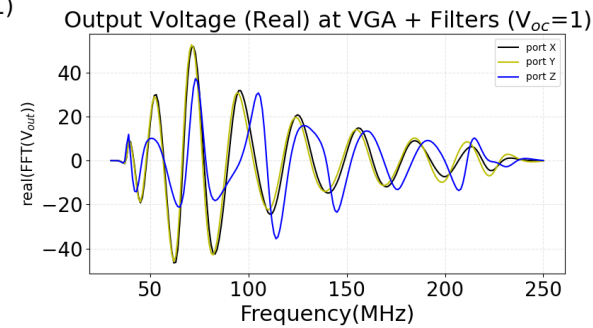
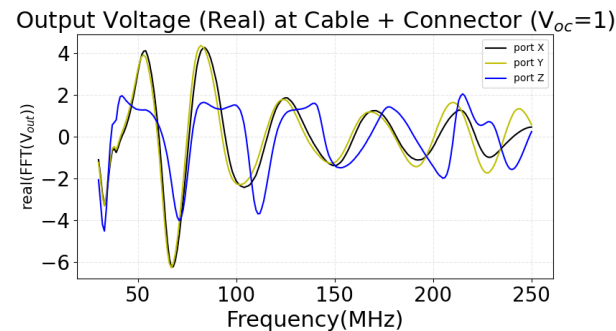
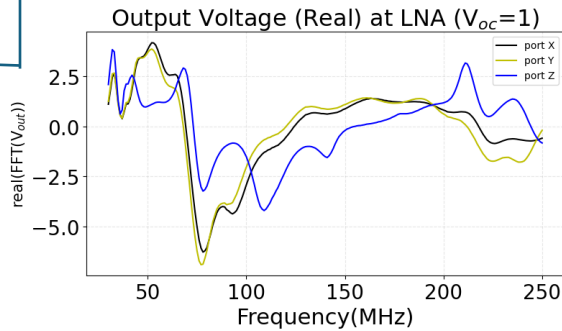
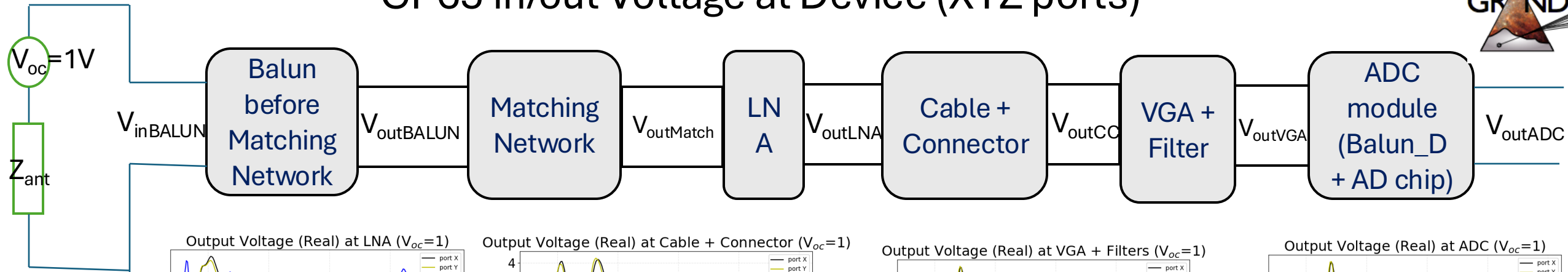
Example: *`$ python3 plot_Vout_AT_Device.py Vout_Ina --savefig`*
Using `--savefig` saves the image in the directory `/grand/scripts/figures/`

Other Options = ["Vin_balun1", "Vout_balun1", "Vout_match_net", "Vout_Ina", "Vout_cable_connector", "Vout_VGA", "Vout_tot", "Vratio_Balun1", "Vratio_match_net", "Vratio_Ina", "Vratio_cable_connector", "Vratio_vga", "Vratio_adc"]

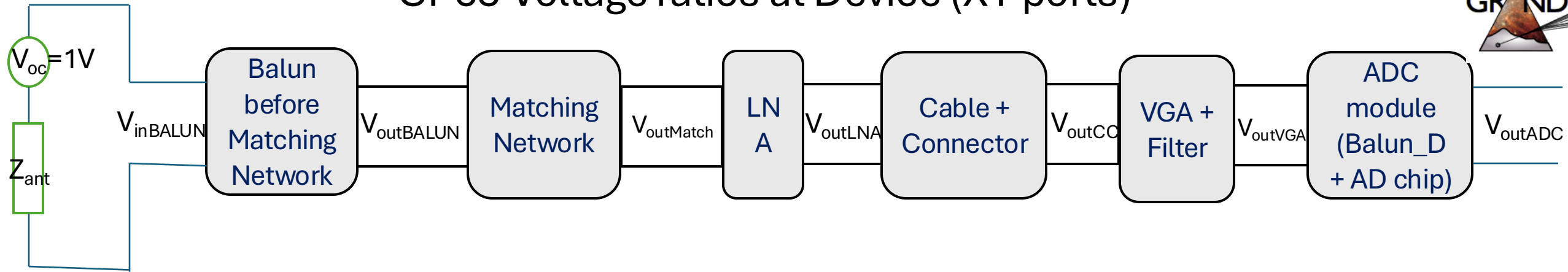
GP65 in/out Voltage at Device (XYZ ports)



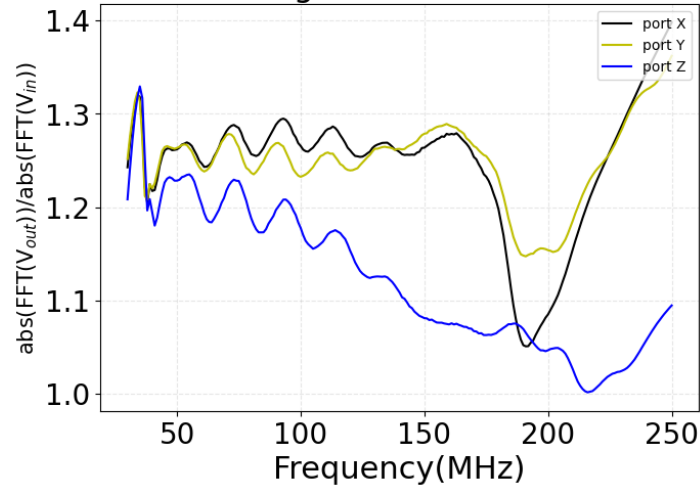
GP65 in/out Voltage at Device (XYZ ports)



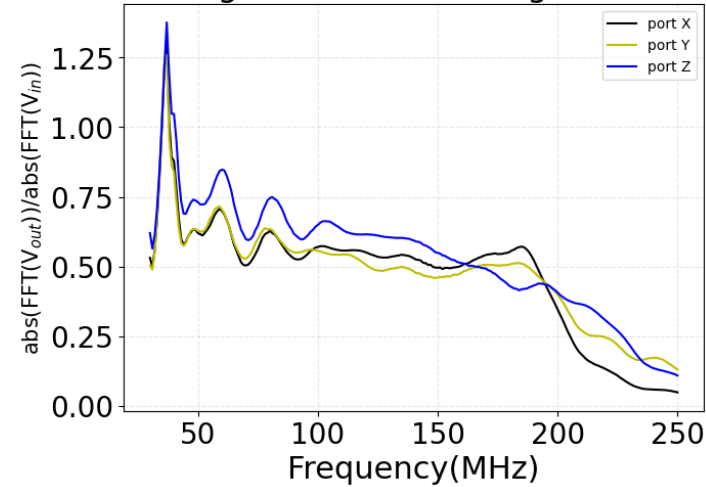
GP65 Voltage ratios at Device (XY ports)



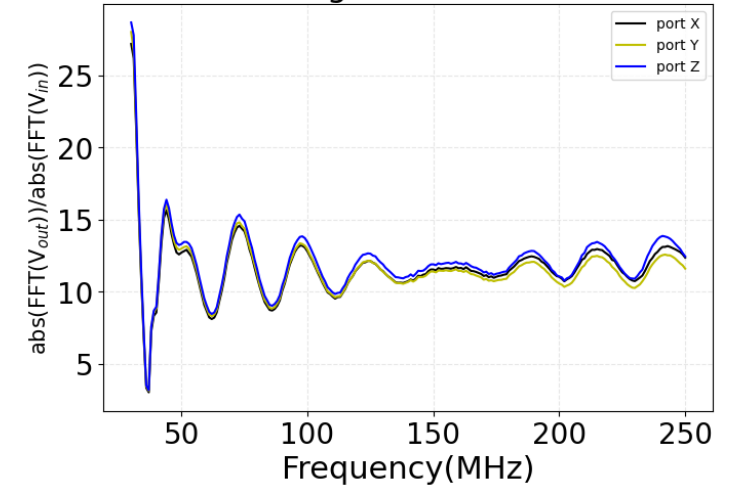
Voltage ratio at Balun1



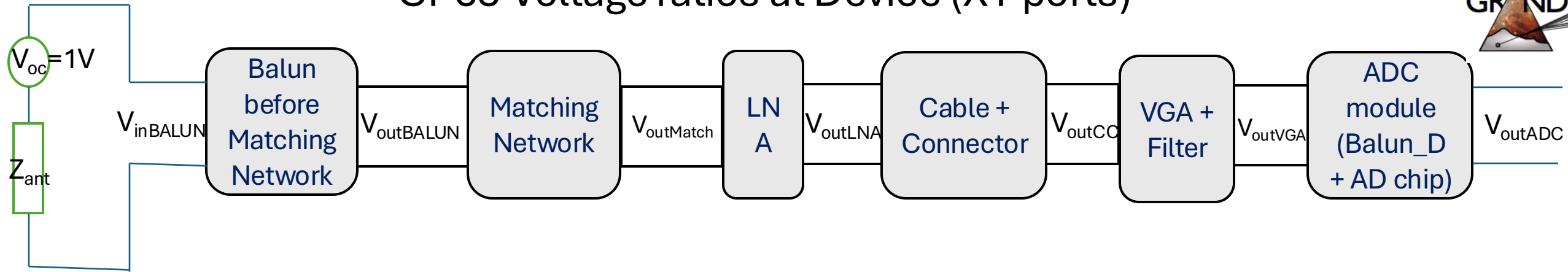
Voltage ratio at Matching network



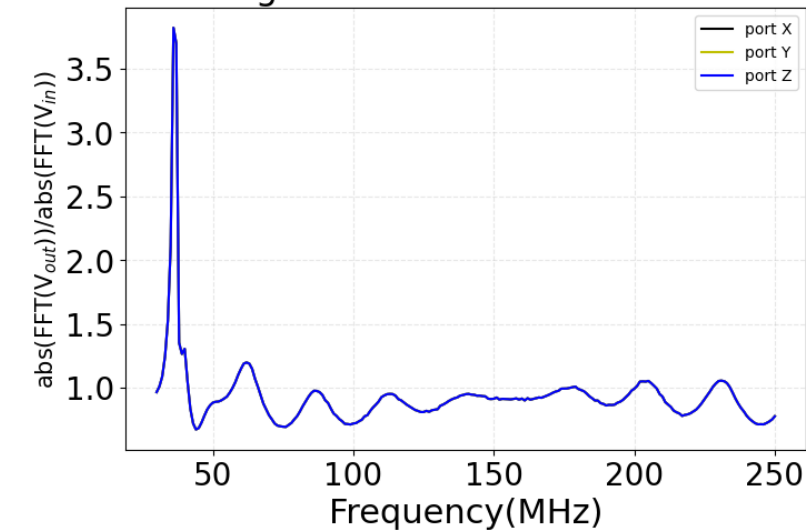
Voltage ratio at LNA



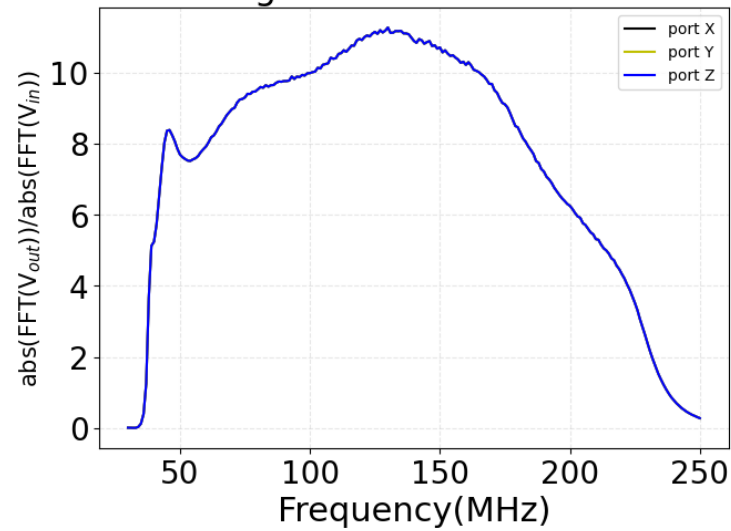
GP65 Voltage ratios at Device (XY ports)



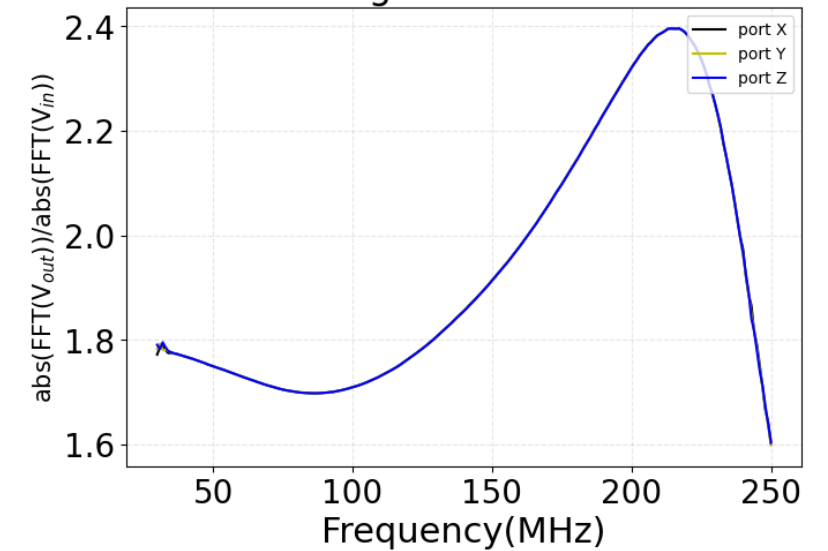
Voltage ratio at Cable+Connector



Voltage ratio at VGA+Filters

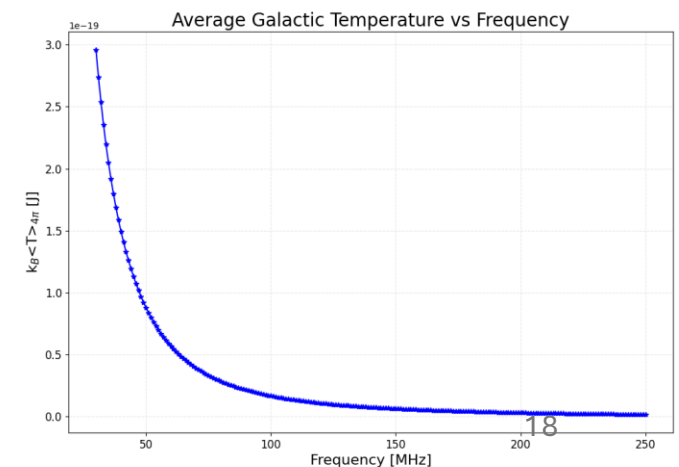
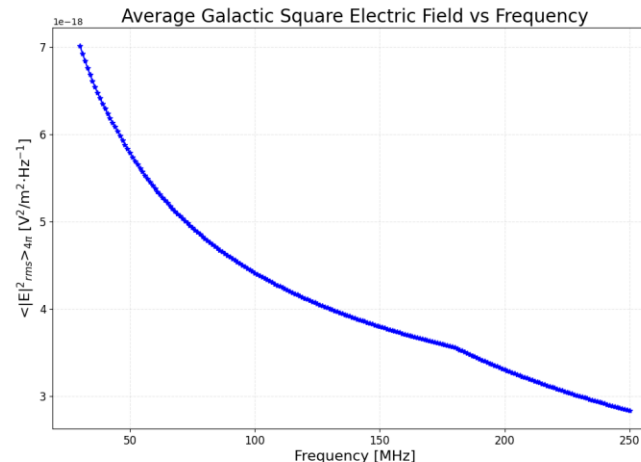
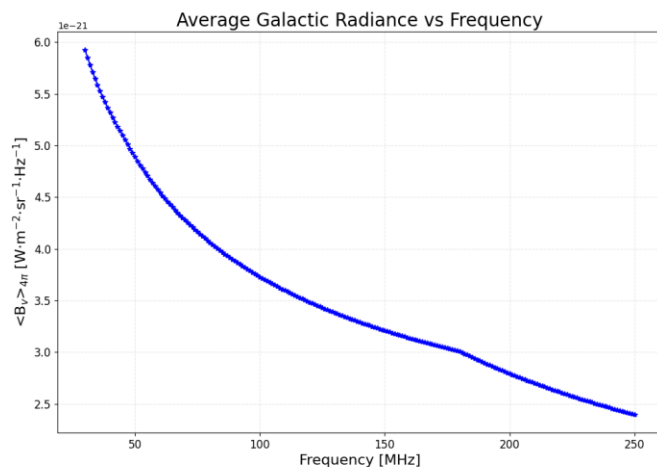


Voltage ratio at ADC

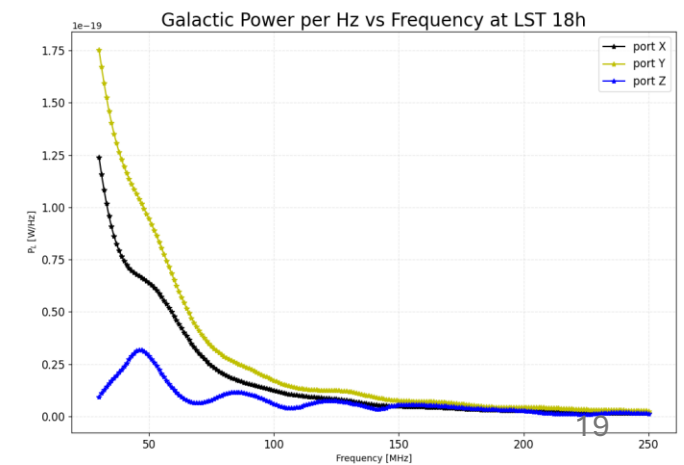
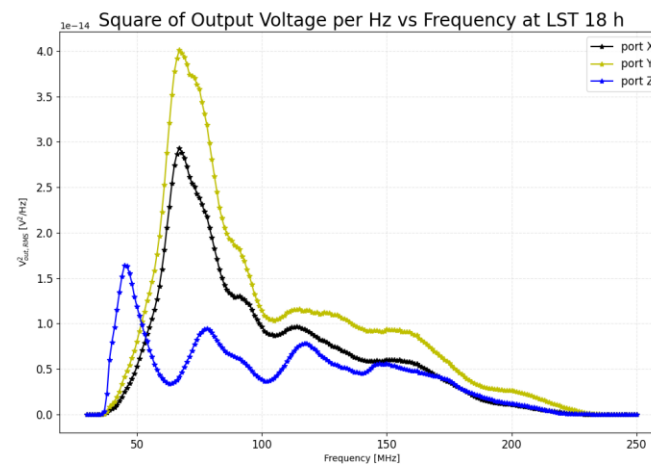
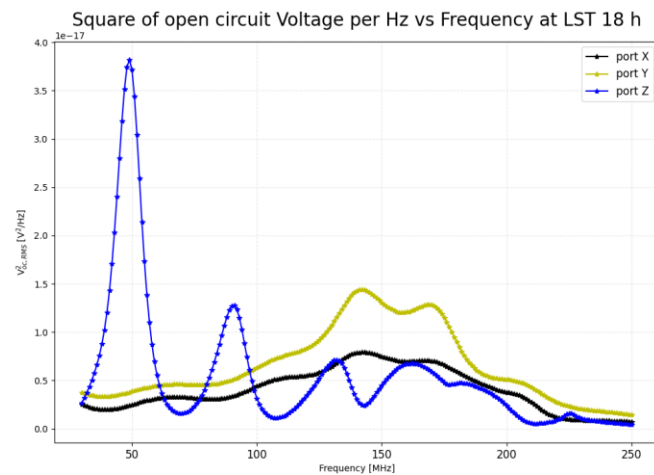
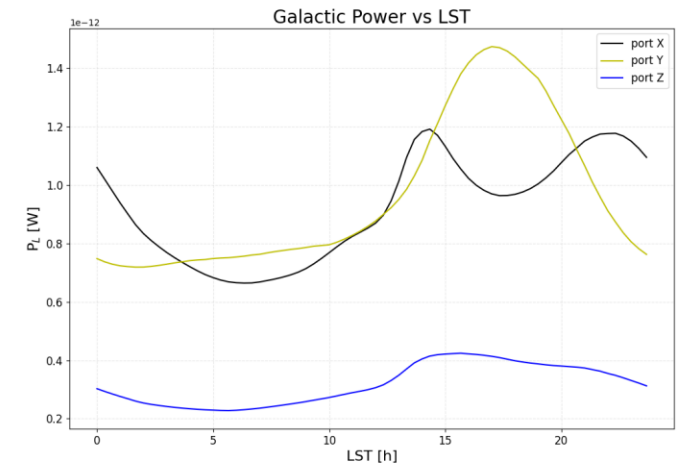
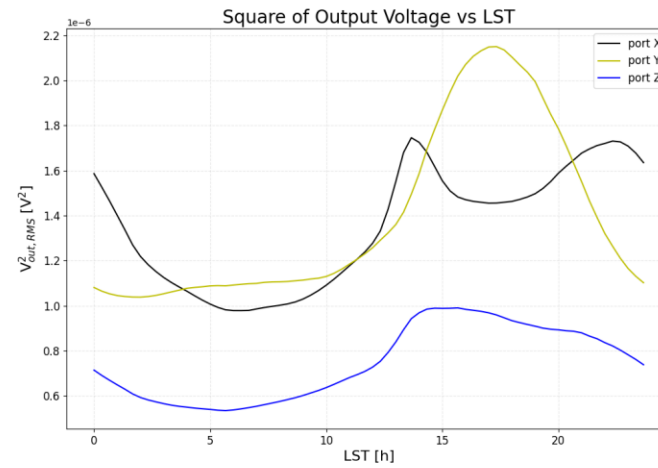
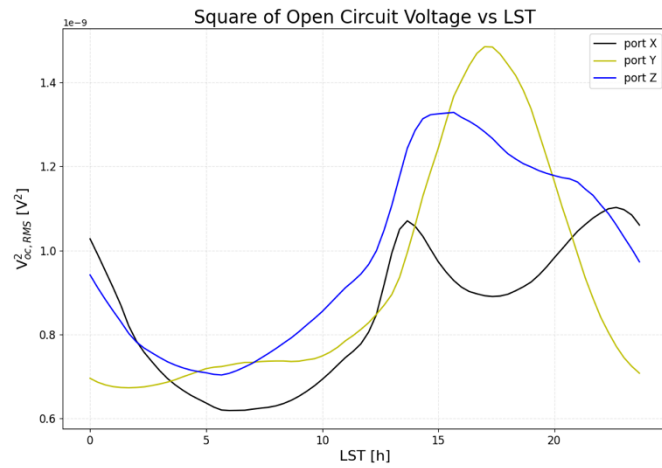


Galactic Noise Simulations

- The script **Compute_(Plot_)Galactic_noise.py** (path: `/home/grand/grand/sim/noise/`) computes the Galactic noise for 221 frequency bins (30-250 MHz with 1 MHz step) and 72 Local Sidereal Time (LST) bins (0:00 – 23:40 with 20 min step) through sky integration.
- How to use: **Compute_Plot_Galactic_noise.py** [-h]
[`--site {gp13,gaa}`]
[`--du_type {GP300,GP300_nec,GP300_mat}`]
[`--run_mode {Voc, Vout, PL, Efield}`]
[`--lst_value {0, 1, 2, ... 23}`]



Galactic Noise Outputs



New Antenna Model

Antenna Effective Length

➤ /sps/grand/RFChain_XD/RFchain_20250302_updated/E
FL_data_3.2m_XYZ_20250313.mat
{file.shape(221x3x3x181x361)}

