Manual for using the Spectral Analysis Tool

Luca Rigon, TU Munich, Summer 2023/Winter 2024

Satellite Geodesy Research Facility of the Technical University of Munich

The tool is using a various combinations of command argument sets. Some sample execution sets are given in the following paragraphs, as well as some explanations for the required input parameters and for the code, since there are some adaptation points when the working station is changed, such as folder paths.

<u>Required input parameters:</u>

python main.py -s 2019.05.28.18:00:00 -session vt9148

The required parameters for the execution are the **-s** parameter, which indicates the start time for the interested session output, and the **-session** parameter, which indicates the information of the session which is to be analyzed (i.e. the filename).

<u>Additional time parameters:</u>

python main.py -s 2018.06.25.22:24:00 -d 1h -session vt8176

python main.py -s 2019.05.13.23:56:00 -e 2019.05.14.17:00:00 -session vt9133ws

The **-e** and the **-d** parameters indicate the end time of the session to be considered, and the duration of the session, respectively. The duration can only be in hours (format: xxh). If these are not given, the whole duration of the session will be taken for the analysis.

NB: executing the script without setting any optional parameter (listed below) will result to an empty run of the code.

<u>Optional parameters:</u>

python main.py -s 2019.05.28.18:00:00 -session vt9148 -GNUplot 2019.05.28.18:14:00

python main.py -s 2019.05.13.23:56:00 -d 02h -session vt9133ws -removeCal 0 -spectrogram 1

python main.py -s 2019.05.13.18:00:00 -session vt9133 -skyplot 1

python main.py -s 2019.05.13.18:00:00 -session vt9133 -skyplot all

python main.py -s 2019.05.13.23:56:00 -d 02h -session vt9133ws -spectrogram 1 -method mean

python main.py -s 2019.05.28.18:00:00 -session vt9148 -e 2019.05.28.22:46:00 -removeCal 0 -GNUplot 2019.05.28.18:00:00 -spectrogram 1 - skyplot 9 -method median

The **-removeCal** parameter determines whether the registered calibration signals (peaks in the measurements) are to be considered for the analysis or not. Its default is set to True, meaning that the peaks will always be removed unless the input **-removeCal 0** is provided. The frequencies of these calibration signals (hence their location within the datasets) are already provided and shall be adapted to the receiving antenna's calibration signal frequencies and channels which are being considered for the analysis (these can be found within the function *remove_peaks()* in the *utils.py* - file).

The **-method** parameter allows to choose on a method for analyzing the data: **-method max** returns the maximum values measured in each channel and is the default parameter in case it is not provided. For a further smoothening of the power measurements **-method median** and **-method mean** can be used; any 'numpy' – function is allowed.

The **-GNUplot** parameter indicates the time at which the spectral data must be visualized as a GNU plot. This data will only be plotted if the indicated time is in the required format (see below), corresponds to the time of an existing scan-file, and is situated within the chosen analysis time of the session. Another allowed input is **-GNUplot all**, which will then generate the plot for every scan within the chosen time range. If not provided, no GNUplots will be returned.

The **-spectrogram** parameter determines whether to perform the spectral analysis (and plot the spectrogram) of the session. Its default is set to False, meaning that the spectral analysis will only run if the input **-spectrogram 1** is provided.

The **-skyplot** parameter determines whether to visualize the RFI power levels spatially as a polar plot or not, for the Azimuth and Elevation values provided in the summary file. Its default is set to False; the skyplot will only be generated for the provided inputs **-skyplot n** or **-skyplot all**, where **n** can be a number between 1 and 16, representing the channel to be visualized. When choosing **all**, a skyplot for each channel is created and saved, as well as a skyplot for the values taken over all bands contemporaneously. The channel numbers and they respective frequency ranges are shown in Fig. 1 (channels 9-16 have the same frequencies again, but opposite polarization).

The **-sky_clip** parameter specifies whether to clip the power intensities on a specific range, allowing to visualize better strong disturbances on the skyplot, or to filter them out instead (to be implemented). Input is -**sky_clip [clip_min, clip_max]**, where these two values specify the range of values on which to clip it. If not provided, no clipping is performed.

	1	2	3	4	5	6	7	8
Band A	3032.4	3064.4	3096.4	3224.4	3320.4	3384.4	3448.4	3480.4
Band B	5272.4	5304.4	5336.4	5464.4	5560.4	5624.4	5688.4	5720.4
Band C	6392.4	6424.4	6456.4	6584.4	6680.4	6744.4	6808.4	6840.4
Band D	10232.4	10264.4	10296.4	10424.4	10520.4	10584.4	10648.4	10680.4

Fig. 1

Multiple optional parameters can be used simultaneously.

The **start**, **end** and the **GNUplot** parameters should have the following format:

%Y.%m.%d.%H:%M:%S (YYYY.mm.dd.HH:MM:SS)

For each session folder, there might be different day of year files. This is handled by listing all the available files in the particular folder of the session and analyzing them regardless checking the time, and in the spec files, this can be observed. For example, the following execution shows this case: **python main_edit.py -s 2019.05.28.23:54:00 -e 2019.05.29.00:12:00 -session vt9148 -spectrogram 1**, and is shown in Fig.2:

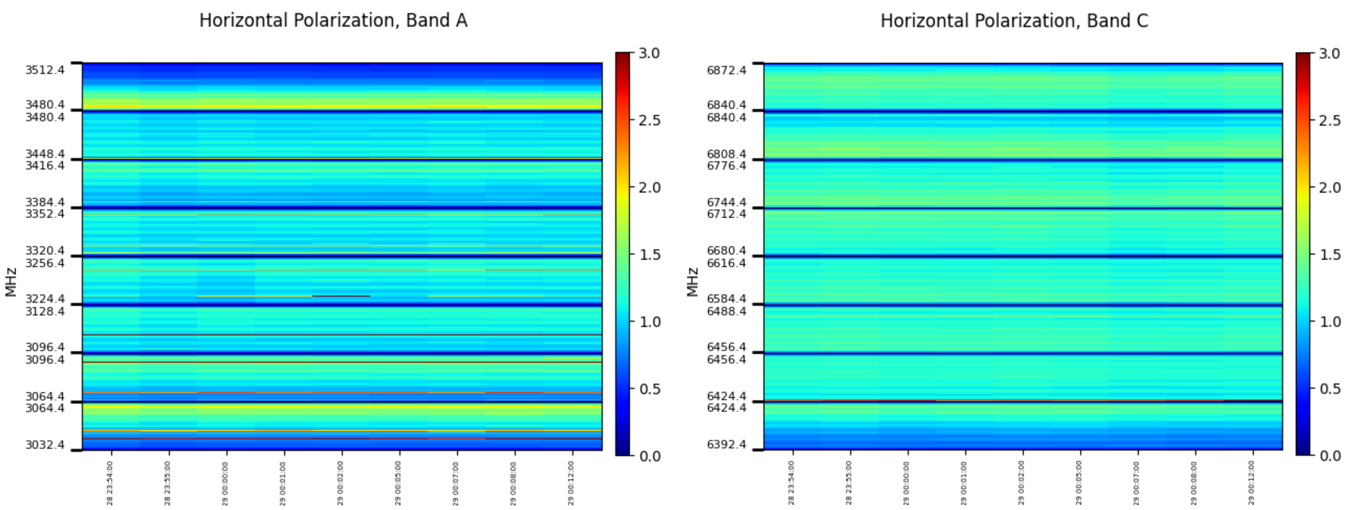

Fig. 2

As can be seen here, if there is another day in the folder, the spectrum results show that in the desired format.

For each spectral band, the scans of the chosen session are extracted and appended into one large dataset, going from the defined beginning time until the end time.

These are then further processed based on the input parameters, returning different types of images. The possible outputs are:

- One image, or n_s images (where n_s is the number of samples found in the session), containing 16 GNU - power plots measured at the 8 different frequency ranges, for both 2 polarizations (Horizontal/ Vertical), for each spectral band A, B, C, D (4x1 or 4xn_s images, looking like Fig.3)
- A spectrogram which shows the measured (maximum value, mean or median, …) power for all 8 frequency ranges over the whole selected time span, for each polarization, and each spectral band A, B, C, D (4x2 images, see Fig.s2,4)
- One skyplot, or 17 skyplots which visualize the spatial distribution of the measured intensities (maximum value, mean or median, …), according to the corresponding Azimuth and Elevation data provided in the summary file of the session. An arrow is

displayed pointing towards the highest disturbance within the selected spectral channel. The frequency, as well as the intensity, of the highest disturbance over all bands are returned as well. This is generated for the chosen frequency channels (skyplot input parameter 1-16) and 4 spectral bands (1x4 images, see Fig.5a), or for every single channel (skyplot input parameter “all”, returning 17x4 images).

- A map of the observatory (using the OpenStreetMap API) with the same arrow pointing towards the returned highest disturbance over all bands. The observatory coordinates can be manually fixed at the beginning of the main script, and map visualization can manually be removed at line 261 of the main script, within the *sky_plot()* function (Fig.5b).

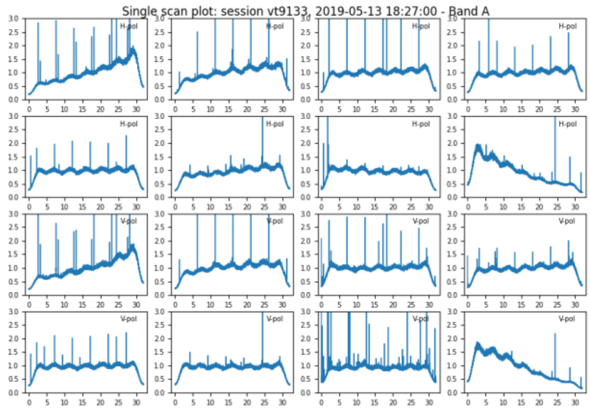

Fig. 3

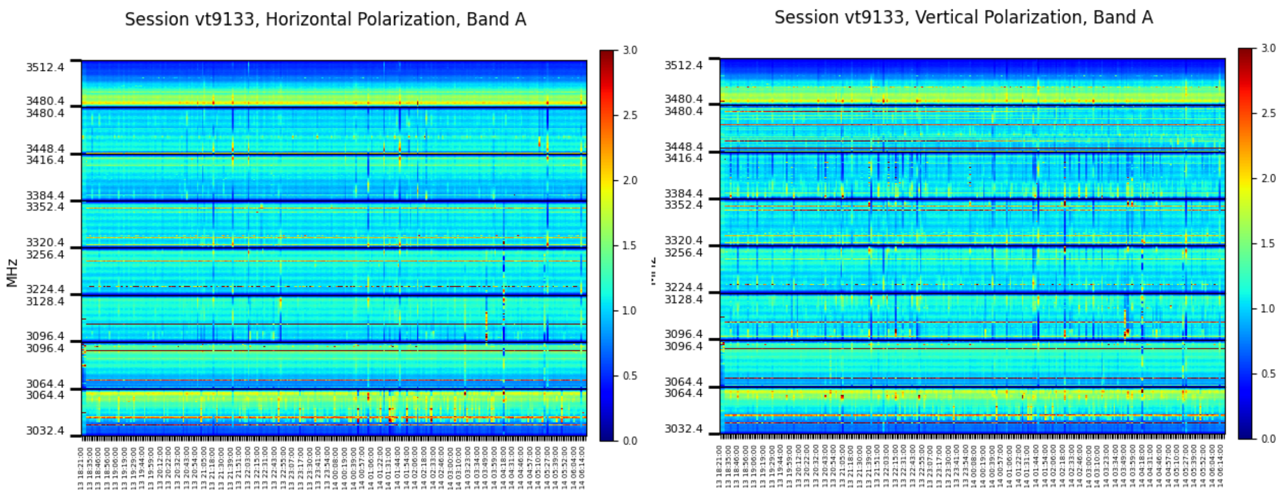

Fig. 4

For entering for the interested session folder, the following code selection should be updated (found int *utils.py*):

```
def get_session_path(session):
    # Update here for the correct session path where the path is different.
    path = f'Datasets/{session}/'
    return path
```

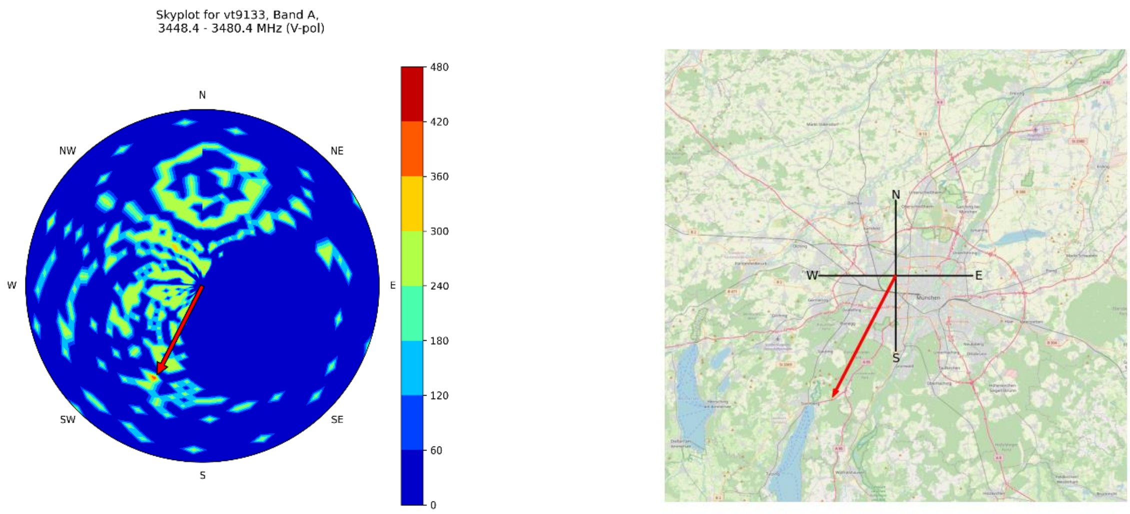

Fig. 5

This code snippet assumes that the given session parameter in the command argument has the same name as the folder containing all the scans to be analyzed. For example, for the above execution scenarios, the folders are appearing like following:

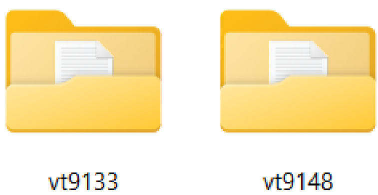

Fig. 6

This path has to be the same for all input files, hence also the session summary file has to be included in this folder.

A results folder ("Figures/*sessionname*") for the selected session is automatically generated within the current workspace, where the output images are automatically saved. The image names contain the selected parameters, as well as their timestamps.

Here, another important note is the name of the files to be analyzed. From the command arguments, the session parameter is given. For searching the names, the first 6 characters are searched in the dedicated session folder, so that the 'ws' or another set of letters attached at the end, indicating different antennas, aren't creating problems.

For the spectrograms, it was observed that some of the executions are taking a little longer than expected with some sessions when there are many files to be analyzed. At this point, a quick solution can be the adaptation of the split numbers. If there are a lot of records inside a search of interest, the split number can be decreased. (Variable *split_num* at line 72 of the main script). The default value is 160.