

tpa_frq_extraction

April 3, 2025

1 Frequency Range extraction

obtain the frequency range for the TPA1 satellite from the ITU database

```
[1]: import pandas as pd
      from IPython.display import display
      from ITUtils import itu_to_bandwidth, channel_appender
```

```
[2]: filename = './databases/NZquery.csv'
      # import the table using pandas
      NZdataframe = pd.read_csv(filename)
      # check the table header
      print(NZdataframe.columns)
      # get the TPA table only
      TPAdf = NZdataframe[NZdataframe[' com_el.sat_name'] == 'TPA-1']
      # check the content
      display(TPAdf)
```

```
Index(['com_el.ntc_id', ' com_el.tgt_ntc_id', ' com_el.adm',
      ' com_el.ntwk_org', ' com_el.sat_name', ' com_el.long_nom',
      ' com_el.ntc_type', ' com_el.prov', ' com_el.ntf_rsn', ' com_el.d_rcv',
      ' orbit.orb_id', ' orbit.nbr_sat_pl', ' orbit.apog_km',
      ' orbit.perig_km', ' orbit.op_ht_km', ' s_beam.emi_rcp',
      ' s_beam.beam_name', ' grp.grp_id', ' grp.freq_min', ' grp.freq_max',
      ' grp.bdwidth', ' emiss.seq_no', ' emiss.pwr_ds_max',
      ' emiss.design_emi', ' carrier_fr.freq_carr'],
      dtype='object')
```

	com_el.ntc_id	com_el.tgt_ntc_id	com_el.adm	com_el.ntwk_org \
0	124545322	NaN	NZL	NaN
1	124545322	NaN	NZL	NaN
2	124545322	NaN	NZL	NaN
3	124545322	NaN	NZL	NaN
4	124545322	NaN	NZL	NaN
5	124545322	NaN	NZL	NaN
6	124545322	NaN	NZL	NaN
7	124545322	NaN	NZL	NaN
8	124545322	NaN	NZL	NaN
9	124545322	NaN	NZL	NaN

10	124545322	NaN	NZL	NaN
11	124545322	NaN	NZL	NaN

	com_el.sat_name	com_el.long_nom	com_el.ntc_type	com_el.prov	\
0	TPA-1	NaN	N	9.1/IA	
1	TPA-1	NaN	N	9.1/IA	
2	TPA-1	NaN	N	9.1/IA	
3	TPA-1	NaN	N	9.1/IA	
4	TPA-1	NaN	N	9.1/IA	
5	TPA-1	NaN	N	9.1/IA	
6	TPA-1	NaN	N	9.1/IA	
7	TPA-1	NaN	N	9.1/IA	
8	TPA-1	NaN	N	9.1/IA	
9	TPA-1	NaN	N	9.1/IA	
10	TPA-1	NaN	N	9.1/IA	
11	TPA-1	NaN	N	9.1/IA	

	com_el.ntf_rsn	com_el.d_rcv	...	s_beam.emi_rcp	s_beam.beam_name	\
0	A	04.10.2024	...	R	UHFUP	
1	A	04.10.2024	...	R	UHFUP	
2	A	04.10.2024	...	R	SUP	
3	A	04.10.2024	...	R	UHFUP	
4	A	04.10.2024	...	E	SDN	
5	A	04.10.2024	...	R	SUP	
6	A	04.10.2024	...	E	UHFDN	
7	A	04.10.2024	...	R	UHFUP	
8	A	04.10.2024	...	E	UHFDN	
9	A	04.10.2024	...	E	UHFDN	
10	A	04.10.2024	...	E	UHFDN	
11	A	04.10.2024	...	E	SDN	

	grp.grp_id	grp.freq_min	grp.freq_max	grp.bdwidth	emiss.seq_no	\
0	124694579	401.825	402.0	NaN	2	
1	124694579	401.825	402.0	NaN	1	
2	124694581	2055.000	2075.0	NaN	1	
3	124694579	401.825	402.0	NaN	2	
4	124694582	2202.000	2239.0	NaN	1	
5	124694581	2055.000	2075.0	NaN	1	
6	124694580	401.825	402.0	NaN	1	
7	124694579	401.825	402.0	NaN	1	
8	124694580	401.825	402.0	NaN	2	
9	124694580	401.825	402.0	NaN	1	
10	124694580	401.825	402.0	NaN	2	
11	124694582	2202.000	2239.0	NaN	1	

	emiss.pwr_ds_max	emiss.design_emi	carrier_fr.freq_carr
0	-55.0	9K50F1DAN	401.96
1	-55.0	19K8F1DAN	401.96

2	-41.0	250KM1DAN	2055.60
3	-55.0	9K50F1DAN	401.90
4	-55.8	1M00M1DAN	2237.50
5	-41.0	250KM1DAN	2065.70
6	-39.8	19K8F1DAN	401.96
7	-55.0	19K8F1DAN	401.90
8	-39.8	9K50F1DAN	401.96
9	-39.8	19K8F1DAN	401.90
10	-39.8	9K50F1DAN	401.90
11	-55.8	1M00M1DAN	2202.90

[12 rows x 25 columns]

```
[3]: # filter by fmin, fmax and emission direction
columns_to_keep = [' grp.grp_id', ' emiss.seq_no', ' s_beam.beam_name', ' grp.
    ↳freq_min', ' grp.freq_max', ' s_beam.emi_rcp', ' carrier_fr.freq_carr',
    ↳' emiss.design_emi']
TPAfreqs = TPAdf[columns_to_keep]

# sort by carrier
TPAfreqs = TPAfreqs.sort_values(by=[ ' grp.grp_id', ' carrier_fr.freq_carr', '↳
    ↳s_beam.emi_rcp', ' emiss.design_emi'])
display(TPAfreqs)
fmins = TPAfreqs[' grp.freq_min'].unique()
fmaxs = TPAfreqs[' grp.freq_max'].unique()

# Create a DataFrame with the unique values
unique_freqs_df = pd.DataFrame({
    ↳' grp.freq_min': fmins,
    ↳' grp.freq_max': fmaxs
})

# Display the DataFrame as a nicely formatted table
display(unique_freqs_df)
```

	grp.grp_id	emiss.seq_no	s_beam.beam_name	grp.freq_min	grp.freq_max	\
7	124694579	1	UHFUP	401.825	402.0	
3	124694579	2	UHFUP	401.825	402.0	
1	124694579	1	UHFUP	401.825	402.0	
0	124694579	2	UHFUP	401.825	402.0	
9	124694580	1	UHFDN	401.825	402.0	
10	124694580	2	UHFDN	401.825	402.0	
6	124694580	1	UHFDN	401.825	402.0	
8	124694580	2	UHFDN	401.825	402.0	
2	124694581	1	SUP	2055.000	2075.0	
5	124694581	1	SUP	2055.000	2075.0	
11	124694582	1	SDN	2202.000	2239.0	
4	124694582	1	SDN	2202.000	2239.0	

	s_beam.emi_rcp	carrier_fr.freq_carr	emiss.design_emi
7	R	401.90	19K8F1DAN
3	R	401.90	9K50F1DAN
1	R	401.96	19K8F1DAN
0	R	401.96	9K50F1DAN
9	E	401.90	19K8F1DAN
10	E	401.90	9K50F1DAN
6	E	401.96	19K8F1DAN
8	E	401.96	9K50F1DAN
2	R	2055.60	250KM1DAN
5	R	2065.70	250KM1DAN
11	E	2202.90	1M00M1DAN
4	E	2237.50	1M00M1DAN

	grp.freq_min	grp.freq_max
0	401.825	402.0
1	2055.000	2075.0
2	2202.000	2239.0

2 Append Channel information and create TPA1 reference table

```
[4]: # Example usage for itu_to_bandwidth in file ITUtils.py
itu_designation = "1M00M1DAN"
bandwidth = itu_to_bandwidth(itu_designation) / 1000000
print(f"The bandwidth for {itu_designation} is {bandwidth} MHz")
```

The bandwidth for 1M00M1DAN is 1.0 MHz

```
[8]: # Add columns for channel bandwidth, channel frequency minimum, and channel
      ↪ frequency maximum
TPAfreqs = channel_appender(TPAfreqs)

# Display the updated DataFrame REMOVE ORDERING
display(TPAfreqs.sort_index())
```

	grp.grp_id	emiss.seq_no	s_beam.beam_name	grp.freq_min	grp.freq_max	\
0	124694579	2	UHFUP	401.825	402.0	
1	124694579	1	UHFUP	401.825	402.0	
2	124694581	1	SUP	2055.000	2075.0	
3	124694579	2	UHFUP	401.825	402.0	
4	124694582	1	SDN	2202.000	2239.0	
5	124694581	1	SUP	2055.000	2075.0	
6	124694580	1	UHFDN	401.825	402.0	
7	124694579	1	UHFUP	401.825	402.0	
8	124694580	2	UHFDN	401.825	402.0	
9	124694580	1	UHFDN	401.825	402.0	
10	124694580	2	UHFDN	401.825	402.0	

11	124694582	1	SDN	2202.000	2239.0
----	-----------	---	-----	----------	--------

	s_beam.emi_rcp	carrier_fr.freq_carr	emiss.design_emi	channel.bandwidth	\
0	R	401.96	9K50F1DAN	9500.0	
1	R	401.96	19K8F1DAN	19800.0	
2	R	2055.60	250KM1DAN	250000.0	
3	R	401.90	9K50F1DAN	9500.0	
4	E	2237.50	1M00M1DAN	1000000.0	
5	R	2065.70	250KM1DAN	250000.0	
6	E	401.96	19K8F1DAN	19800.0	
7	R	401.90	19K8F1DAN	19800.0	
8	E	401.96	9K50F1DAN	9500.0	
9	E	401.90	19K8F1DAN	19800.0	
10	E	401.90	9K50F1DAN	9500.0	
11	E	2202.90	1M00M1DAN	1000000.0	

	channel.freq_min	channel.freq_max
0	401.95525	401.96475
1	401.95010	401.96990
2	2055.47500	2055.72500
3	401.89525	401.90475
4	2237.00000	2238.00000
5	2065.57500	2065.82500
6	401.95010	401.96990
7	401.89010	401.90990
8	401.95525	401.96475
9	401.89010	401.90990
10	401.89525	401.90475
11	2202.40000	2203.40000

```
[7]: # create reference database of tpa1 for conflicts determination
TPAdf = channel_appender(TPAdf)
outname = './databases/TPAtable.csv'

# Save the DataFrame to a CSV file
TPAdf.to_csv(outname, index=False)

print('DataFrame saved to' ,outname)
```

DataFrame saved to ./databases/TPAtable.csv