In [ ]:

```
import h5py
import math
import numpy as np
import scipy
import matplotlib.pyplot as plt
from scipy import signal
import pandas as pd
import matplotlib.dates as mdates
import scipy.sparse as sparse
import datetime as dt
from datetime import datetime
from datetime import date
import matplotlib.ticker as ticker
from mpl_toolkits.axes_grid1 import make_axes_locatable
from datetime import date, timedelta
import matplotlib.ticker as plticker
from matplotlib.colors import LogNorm,Normalize
```

In [ ]:

```
dir_name=""
file_name= dir_name+"C:/CSES/CSES_01_EFD_3_L02_A1_028281_20180807_070512_20180807_074400_000.h5"

OrbitNumber=file_name.split("_")[6];
with h5py.File(file_name, "r") as f:
    UTC_TIME=f["UTC_TIME"][()][:,0]
    GEO_LAT=f["GEO_LAT"][()][:,0]
    GEO_LON=f["GEO_LON"][()][:,0]
    ALT=f["ALTITUDE"][()][:,0]
    Workmode=f["WORKMODE"][()][:,0]
    MAG_LAT=f["MAG_LAT"][()][:,0]
    MAG_LON=f["MAG_LON"][()][:,0]
    VERSE_TIME=f["VERSE_TIME"][()][:,0]
    A131_W=f["A131_W"][()]
    A132_W=f["A132_W"][()]
    A133_W=f["A133_W"][()]
    A131_P=f["A131_P"][()]
    A132_P=f["A132_P"][()]
    A133_P=f["A133_P"][()]
```

In [ ]:

```
f = h5py.File(dir_name+file_name, 'r')
columns = list(f.keys())

df = pd.DataFrame([])
for column in columns:
    try:
        data = np.array(f[column])
        if data.shape[1] == 1:
            df[column] = data.flatten()
        elif column.endswith('_P' ) :
            mat = sparse.coo_matrix(data, shape=data.shape)
            df[column] = mat.toarray().tolist()
        elif column =="A131_W":
            selected_data=np.array(data[0:1077,:])
            mat = sparse.coo_matrix(selected_data, shape=selected_data.shape)
            df[column] = mat.toarray().tolist()
        elif column =="A132_W":
            selected_data=np.array(data[0:1077,:])
            mat = sparse.coo_matrix(selected_data, shape=selected_data.shape)
            df[column] = mat.toarray().tolist()
        elif column =="A133_W":
            selected_data=np.array(data[0:1077,:])
            mat = sparse.coo_matrix(selected_data, shape=selected_data.shape)
            df[column] = mat.toarray().tolist()
        else:
            print(column+ ' skipped')
    except Exception as e:
        print(column+ ' parse error: ' + str(e))
S_burst = df[df.WORKMODE == 2]
df['DATE_TIME'] = pd.to_datetime(df.UTC_TIME, format='%Y%m%d%H%M%S%f')
```

In [ ]:

```
import sys
sys.path.insert(0,"C:/CSES/")
from efd import EFD
efd = EFD()
vlf = efd.read("C:/CSES/", "CSES_01_EFD_3_L02_A1_028281_20180807_070512_20180807_074400_000.h5")
```

In [ ]:

```
vlf.vlf_signal[['UTC_TIME']]=df[['UTC_TIME']]
vlf.vlf_signal[['A131_P']]=df[['A131_P']]
vlf.vlf_signal[['A132_P']]=df[['A132_P']]
vlf.vlf_signal[['A133_P']]=df[['A133_P']]
vlf.vlf_signal[['A131_W']]=df[['A131_W']]
vlf.vlf_signal[['A132_W']]=df[['A132_W']]
vlf.vlf_signal[['A133_W']]=df[['A133_W']]
vlf.vlf_signal[['VERSE_TIME']]=df[['VERSE_TIME']]

df_burst = vlf.vlf_signal[vlf.vlf_signal.WORKMODE == 2]
vlf.vlf_signal['Frequency'] = vlf.vlf_signal.apply(lambda M: M[0])
DATE=vlf.vlf_signal.DateTime.map(lambda x: x.strftime('%Y-%m-%d'))
TIME=vlf.vlf_signal.DateTime.map(lambda x: x.strftime('%H-%M-%S'))
```

In [ ]:

```
sampling_frequency=51200
n_lines=1024
FFT_samples=2048
frequency_bin=(sampling_frequency/2)/n_lines
time_bin_survey=(vlf.vlf_signal.VERSE_TIME[191]-vlf.vlf_signal.VERSE_TIME[190])/10**3
time_bin_burst=time_bin_survey/50
Bw=51200*2/1024
print("sampling_frequency [Hz]=",sampling_frequency)
print("num_lines=",n_lines)
print("FFT_samples=", FFT_samples)
print("frequency_bin [Hz]=",frequency_bin)
print("time_bin_survey [sec]=",time_bin_survey)
print("time_bin_burst [sec]=",time_bin_burst)
print("frequency range tot=0-1024 kHz")
print("frequency range selected=0-25 kHz")
print("Bandwidth=" ,Bw)
```

In [ ]:

```
start={}
stop={}
idx=0
lastWorkmode=1
for i in range (0,vlf.vlf_signal.WORKMODE.shape[0]):
    if vlf.vlf_signal.WORKMODE[i]==2 and lastWorkmode!=2:
        start[idx]=i
    if vlf.vlf_signal.WORKMODE[i]==1 and lastWorkmode==2:
        stop[idx]=i-1
        idx=idx+1
    lastWorkmode=vlf.vlf_signal.WORKMODE[i]
minStart=start[0]
```

```
maxStop=stop[idx-1]
print(f"STARTposition={start} - STOPposition={stop}")

print("Workmode.shape=",vlf.vlf_signal.WORKMODE.shape)
print("start.len=", len(start))
```

In [ ]:
```
def power(array):
  powerX=180+20*np.log10(array)
  return powerX
```

In [ ]:
```
powerX=power(A131_P)
powerY=power(A132_P)
powerZ=power(A133_P)
```

In [ ]:
```
def inter(array):
  data=np.asarray([array[i]  for i in range (0,len(array)) if  isinstance(array[i], np.ndarray)])
  data_t = np.empty(shape=(data.shape[0], data.shape[1]))
  for i in range (0, data.shape[0]):
    meanX_b = np.mean(data[i])
    data_t[i] = data[i] - meanX_b

  zero=np.zeros_like(data_t)
  s=np.array([[i,j]for i,j in zip(data_t,zero)]).reshape(2*data_t.shape[0],data_t.shape[1])
  outX_b2=s.ravel().reshape(-1,2048)

  M_b = outX_b2.shape[1]
  hamming_b = signal.get_window("hamming", M_b)

  FFT=np.array([scipy.fft.fft(outX_b2[i]*hamming_b) for i in range(0,outX_b2.shape[0])])
  inter= np.abs(FFT.T[:1024])**2
  inter2X = 200+20*np.log10(inter/Bw, where=0<inter, out=np.nan*inter)

  return inter2X
```

In [ ]:
```
inter2X=inter(vlf.vlf_signal.X)
inter2Y=inter(vlf.vlf_signal.Signal)
inter2Z=inter(vlf.vlf_signal.Z)
```

In [ ]:
```
def amplitude(array):
  data=np.asarray([array[i]  for i in range (0,len(array)) if  isinstance(array[i], np.ndarray)])
  data_t = np.empty(shape=(data.shape[0], data.shape[1]))
  for i in range (0, data.shape[0]):
    meanX_b = np.mean(data[i])
    data_t[i] = data[i] - meanX_b
  outX=data_t.ravel().reshape(-1,2048)

  M_b = outX.shape[1]
  hamming_b = signal.get_window("hamming", M_b)

  FFT_low=np.array([scipy.fft.fft(outX[i]*hamming_b) for i in range(0,outX.shape[0])])
  out= np.abs(FFT_low.T[:1024])**2
  outX_b=200+20*np.log10(out/Bw)
  return outX_b
```

In [ ]:
```
outX_b=amplitude(vlf.vlf_signal.X)
outY_b=amplitude(vlf.vlf_signal.Signal)
outZ_b=amplitude(vlf.vlf_signal.Z)
```

In [ ]:
```
def minMax(array):
  survey_min=round(np.nanmin(array),2)
  survey_max=round(np.nanmax(array),2)

  return survey_min,survey_max
```

In [ ]:
```
survey_minX,survey_maxX=minMax(powerX)
survey_minY,survey_maxY=minMax(powerY)
survey_minZ,survey_maxZ=minMax(powerZ)
burst_high_minX,burst_high_maxX=minMax(inter2X)
burst_high_minY,burst_high_maxY=minMax(inter2Y)
burst_high_minZ,burst_high_maxZ=minMax(inter2Z)
burst_low_minX,burst_low_maxX=minMax(outX_b)
burst_low_minY,burst_low_maxY=minMax(outY_b)
burst_low_minZ,burst_low_maxZ=minMax(outZ_b)

print("survey_minX=", survey_minX)
print("survey_maxX=", survey_maxX)
print("burst_high_minX=", burst_high_minX)
print("burst_high_maxX=", burst_high_maxX)
print("burst_low_minX=", burst_low_minX)
print("burst_low_maxX=", burst_low_maxX)
print("survey_minY=", survey_minY)
print("survey_maxY=", survey_maxY)
print("burst_high_minY=", burst_high_minY)
print("burst_high_maxY=", burst_high_maxY)
print("burst_low_minY=", burst_low_minY)
print("burst_low_maxY=", burst_low_maxY)
print("survey_minZ=", survey_minZ)
print("survey_maxZ=", survey_maxZ)
print("burst_high_minZ=", burst_high_minZ)
print("burst_high_maxZ=", burst_high_maxZ)
print("burst_low_minZ=", burst_low_minZ)
print("burst_low_maxZ=", burst_low_maxZ)
```

In [ ]:
```
def extPlot(survey_minX,survey_maxX,burst_high_minX,burst_high_maxX,burst_low_minX,burst_low_maxX):
  minimo=[survey_minX,burst_high_minX,burst_low_minX]
  massimo=[survey_maxX,burst_high_maxX,burst_low_maxX]
  min_3plots=np.min(minimo)
  max_3plots=np.max(massimo)

  m=[survey_minX,burst_low_minX]
  mas=[survey_maxX,burst_low_maxX]
  min_2plots=np.min(m)
  max_2plots=np.max(mas)
  return min_3plots,max_3plots,min_2plots,max_2plots
```

In [ ]:
```
min_3plotsX,max_3plotsX,min_2plotsX,max_2plotsX=extPlot(survey_minX,survey_maxX,burst_high_minX,burst_high_maxX,burst_low_minX,burst_low_maxX)
min_3plotsY,max_3plotsY,min_2plotsY,max_2plotsY=extPlot(survey_minY,survey_maxY,burst_high_minY,burst_high_maxY,burst_low_minY,burst_low_maxY)
min_3plotsZ,max_3plotsZ,min_2plotsZ,max_2plotsZ=extPlot(survey_minZ,survey_maxZ,burst_high_minZ,burst_high_maxZ,burst_low_minZ,burst_low_maxZ)
```

In [ ]:
```
minimo_tot=[survey_minX,burst_high_minX,burst_low_minX,survey_minY,burst_high_minY,burst_low_minY,survey_minZ,burst_high_minZ,burst_low_minZ]
massimo_tot=[survey_maxX,burst_high_maxX,burst_low_maxX,survey_maxY,burst_high_maxY,burst_low_maxY,survey_maxZ,burst_high_maxZ,burst_low_maxZ]
min_9plots=np.min(minimo_tot)
max_9plots=np.max(massimo_tot)
```

```
minimo_tot6=[survey_minX,burst_high_minX,survey_minZ,burst_high_minZ,survey_minY,burst_high_minY]
massimo_tot6=[survey_maxX,burst_high_maxX,survey_maxZ,burst_high_maxZ,survey_maxY,burst_high_maxY]
min_6plots=np.min(minimo_tot6)
max_6plots=np.max(massimo_tot6)
```

In [ ]:

```
print('val min_3plotX=', min_3plotsX)
print('val max_3plotX=',max_3plotsX)
print('val min_3plotY=', min_3plotsY)
print('val max_3plotY=',max_3plotsY)
print('val min_3plotZ=', min_3plotsZ)
print('val max_3plotZ=',max_3plotsZ)
print('val min_9plot=', min_9plots)
print('val max_9plot=',max_9plots)
print('val min_6plot=', min_6plots)
print('val max_6plot=',max_6plots)
```

In [ ]:

```
def threePlot(EFD,AmpP,AmpW,min_3plotsX,max_3plotsX,survey_minX,survey_maxX,burst_low_minX,burst_low_maxX,burst_high_minX,burst_high_maxX,powerX,outX_b,inter2X,output):
  fig, axs = plt.subplots(3,1, sharex=True,figsize=(20,10) )

  axs[0].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + f"-Survey mode EFD VLF {EFD}-{AmpP} power spectrum [mV/Hz^0.5]-"+ f"time_bin_survey={time_bin_survey}"
            + f"-vminTot= {min_3plotsX}" + f"-vmaxTot= {max_3plotsX}"+ f"-vminSurvey= {survey_minX}" + f"-vmaxSurvey= {survey_maxX}")


  axs[1].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + f"-Burst mode low resolution EFD VLF {EFD} [mV/m]- FFT {AmpW} wave form "+f"- time_bin_burst_low={time_bin_survey}"
            + f"-vminTot= {min_3plotsX}" + f"-vmaxTot= {max_3plotsX}"+
            f"-vminBurst_low resolution= {burst_low_minX}" + f"-vmaxBurst_low resolution= {burst_low_maxX}" )


  axs[2].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + f"-Burst mode high resolution EFD VLF {EFD} [mV/m]- from FFT {AmpW} wave form"
           +f"- time_bin_burst_high={time_bin_burst}"  + f"-vminTot= {min_3plotsX}" + f"-vmaxTot= {max_3plotsX}"
            + f"-vminBurst_high resolution= {burst_high_minX}" + f"-vmaxBurst_high resolution= {burst_high_maxX}" )


  x_lims = list(map( lambda x:x,vlf.vlf_signal.DateTime))
  x_lims = mdates.date2num(x_lims)
  ext=[x_lims.min(),x_lims.max(),0,1024]
  im = [0]*3
  im[0] = axs[0].imshow(np.rot90(powerX),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_3plotsX,vmax=max_3plotsX)

  im[1] = axs[1].imshow(np.fliplr(np.rot90(outX_b,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_3plotsX,vmax=max_3plotsX)

  im[2] = axs[2].imshow(np.fliplr(np.rot90(inter2X,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_3plotsX,vmax=max_3plotsX)


  axs[2].set_xlabel('DATE_TIME')
  axs[0].set_ylabel('f [kHz]')
  axs[1].set_ylabel('f [kHz]')
  axs[2].set_ylabel('f [kHz]')

  plt.ylim(0,1024)



  x1=vlf.vlf_signal.iloc[12]['DateTime']
  x2=vlf.vlf_signal.iloc[189]['DateTime']
  y=np.arange(0,1025)
  axs[1].fill_betweenx(y,x1,x2,facecolor="white")

  x1=vlf.vlf_signal.iloc[657]['DateTime']
  x2=vlf.vlf_signal.iloc[1076]['DateTime']
  y=np.arange(0,1025)
  axs[1].fill_betweenx(y,x1,x2,facecolor="white")

  x1=vlf.vlf_signal.iloc[0]['DateTime']
  x2=vlf.vlf_signal.iloc[10]['DateTime']
  y=np.arange(0,1025)
  axs[1].fill_betweenx(y,x1,x2,facecolor="white")

  x1=vlf.vlf_signal.iloc[12]['DateTime']
  x2=vlf.vlf_signal.iloc[189]['DateTime']
  y=np.arange(0,1025)
  axs[2].fill_betweenx(y,x1,x2,facecolor="white")

  x1=vlf.vlf_signal.iloc[657]['DateTime']
  x2=vlf.vlf_signal.iloc[1076]['DateTime']
  y=np.arange(0,1025)
  axs[2].fill_betweenx(y,x1,x2,facecolor="white")

  x1=vlf.vlf_signal.iloc[0]['DateTime']
  x2=vlf.vlf_signal.iloc[10]['DateTime']
  y=np.arange(0,1025)
  axs[2].fill_betweenx(y,x1,x2,facecolor="white")

  x1=vlf.vlf_signal.iloc[12]['DateTime']
  x2=vlf.vlf_signal.iloc[189]['DateTime']
  y=np.arange(0,1025)
  axs[0].fill_betweenx(y,x1,x2,facecolor="white")

  x1=vlf.vlf_signal.iloc[657]['DateTime']
  x2=vlf.vlf_signal.iloc[1076]['DateTime']
  y=np.arange(0,1025)
  axs[0].fill_betweenx(y,x1,x2,facecolor="white")

  x1=vlf.vlf_signal.iloc[0]['DateTime']
  x2=vlf.vlf_signal.iloc[10]['DateTime']
  y=np.arange(0,1025)
  axs[0].fill_betweenx(y,x1,x2,facecolor="white")



  divider = make_axes_locatable(axs[0])
  cax = divider.append_axes('right',size="1%")
  plt.colorbar(im[0], cax=cax, orientation='vertical',label='dB')
  divider = make_axes_locatable(axs[1])
  cax = divider.append_axes('right',size="1%")
  plt.colorbar(im[1], cax=cax, orientation='vertical',label='dB')
  divider = make_axes_locatable(axs[2])
  cax = divider.append_axes('right',size="1%")
  plt.colorbar(im[2], cax=cax, orientation='vertical',label='dB')


  axs[2].xaxis.set_ticks_position("bottom")
  axs[2].xaxis.set_label_position("bottom")
  axs[2].xaxis_date()
  axs[2].xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
  axs[2].xaxis.set_major_locator(mdates.MinuteLocator(interval=1))
  vlf.vlf_signal.plot( x='DateTime', y='Frequency', ax=axs[2], alpha=0)

  ax0=axs[2].twiny()
  ax1=axs[2].twiny()


  ax0.set_xlabel('GEO_LAT')
  ax0.xaxis.set_ticks_position("bottom")
  ax0.xaxis.set_label_position("bottom")
```

```
  ax0.spines["bottom"].set_position(("axes", -0.3))
  ax0.xaxis.grid(False)
  ax0.set_xlim(GEO_LAT.min(), GEO_LAT.max())
  ax0.xaxis.set_major_locator(ticker.FixedLocator(np.arange(GEO_LAT.min(), GEO_LAT.max(), 5)))
  vlf.vlf_signal.plot(x='GEO_LAT', y='Frequency', ax=ax0,alpha=0)

  ax1.set_xlabel('MAG_LAT')
  ax1.xaxis.set_ticks_position("bottom")
  ax1.xaxis.set_label_position("bottom")
  ax1.spines["bottom"].set_position(("axes", -0.6))
  ax1.xaxis.grid(False)
  ax1.set_xlim(MAG_LAT.min(), MAG_LAT.max())
  ax1.xaxis.set_major_locator(ticker.FixedLocator(np.arange(MAG_LAT.min(), MAG_LAT.max(), 0.5)))
  vlf.vlf_signal.plot( x='MAG_LAT', y='Frequency', ax=ax1, alpha=0)


  plt.savefig(f'/content/drive/MyDrive/Colab Notebooks/Plot/{output}.png',bbox_inches = 'tight')


  plt.show()
```

In [ ]:
```
threePlot('EFDX','A131P','A131W',min_3plotsX,max_3plotsX,survey_minX,survey_maxX,burst_low_minX,burst_low_maxX,burst_high_minX,burst_high_maxX,powerX,outX_b,inter2X,'3plotX')
```

In [ ]:
```
threePlot('EFDY','A132P','A132W',min_3plotsY,max_3plotsY,survey_minY,survey_maxY,burst_low_minY,burst_low_maxY,burst_high_minY,burst_high_maxY,powerY,outY_b,inter2Y,'3plotY')
```

In [ ]:
```
threePlot('EFDZ','A133P','A133W',min_3plotsZ,max_3plotsZ,survey_minZ,survey_maxZ,burst_low_minZ,burst_low_maxZ,burst_high_minZ,burst_high_maxZ,powerZ,outZ_b,inter2Z,'3plotZ')
```

In [ ]:
```
fig, axs = plt.subplots(9,1, sharex=True,figsize=(50,40) )

axs[6].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Survey mode EFD VLF Ex-A131P power spectrum [mV/Hz^0.5]-"+ f"time_bin_survey={time_bin_survey}"
            + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+ f"-vminSurvey= {survey_minX}" + f"-vmaxSurvey= {survey_maxX}")
axs[7].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Survey mode EFD VLF Ey-A132P power spectrum [mV/Hz^0.5]-"+ f"time_bin_survey={time_bin_survey}"
            + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+ f"-vminSurvey= {survey_minY}" + f"-vmaxSurvey= {survey_maxY}")
axs[8].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Survey mode EFD VLF Ez-A133P power spectrum [mV/Hz^0.5]-"+ f"time_bin_survey={time_bin_survey}"
            + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+ f"-vminSurvey= {survey_minZ}" + f"-vmaxSurvey= {survey_maxZ}")

axs[3].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode low resolution EFD VLF Ex [mV/m]- FFT A131_W wave form "+f"- time_bin_burst_low={time_bin_survey}"
            + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+
            f"-vminBurst_low resolution= {burst_low_minX}" + f"-vmaxBurst_low resolution= {burst_low_maxX}" )
axs[4].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode low resolution EFD VLF Ey [mV/m]- FFT A132_W wave form "+f"- time_bin_burst_low={time_bin_survey}"
            + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+
            f"-vminBurst_low resolution= {burst_low_minY}" + f"-vmaxBurst_low resolution= {burst_low_maxY}" )
axs[5].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode low resolution EFD VLF Ez [mV/m]- FFT A133_W wave form "+f"- time_bin_burst_low={time_bin_survey}"
            + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+
            f"-vminBurst_low resolution= {burst_low_minZ}" + f"-vmaxBurst_low resolution= {burst_low_maxZ}" )

axs[0].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode high resolution EFD VLF Ex [mV/m]- from FFT A131W wave form"
+f"- time_bin_burst_high={time_bin_burst}"  + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"
            + f"-vminBurst_high resolution= {burst_high_minX}" + f"-vmaxBurst_high resolution= {burst_high_maxX}" )
axs[1].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode high resolution EFD VLF Ey [mV/m]- from FFT A132W wave form"
+f"- time_bin_burst_high={time_bin_burst}"  + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"
            + f"-vminBurst_high resolution= {burst_high_minY}" + f"-vmaxBurst_high resolution= {burst_high_maxY}" )
axs[2].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode high resolution EFD VLF Ez [mV/m]- from FFT A133W wave form"
+f"- time_bin_burst_high={time_bin_burst}"  + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"
            + f"-vminBurst_high resolution= {burst_high_minZ}" + f"-vmaxBurst_high resolution= {burst_high_maxZ}" )

x_lims = list(map( lambda x:x,vlf.vlf_signal.DateTime))
x_lims = mdates.date2num(x_lims)
ext=[x_lims.min(),x_lims.max(),0,1024]
im = [0]*9
im[0] = axs[0].imshow(np.fliplr(np.rot90(inter2X,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)
im[1] = axs[1].imshow(np.fliplr(np.rot90(inter2Y,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)
im[2] = axs[2].imshow(np.fliplr(np.rot90(inter2Z,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)
im[6] = axs[6].imshow(np.rot90(powerX),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)
im[7] = axs[7].imshow(np.rot90(powerY),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)
im[8] = axs[8].imshow(np.rot90(powerZ),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)
im[3] = axs[3].imshow(np.fliplr(np.rot90(outX_b,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)
im[4] = axs[4].imshow(np.fliplr(np.rot90(outY_b,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)
im[5] = axs[5].imshow(np.fliplr(np.rot90(outZ_b,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_9plots,vmax=max_9plots)


axs[8].set_xlabel('DATE_TIME')
axs[0].set_ylabel('f [kHz]')
axs[1].set_ylabel('f [kHz]')
axs[2].set_ylabel('f [kHz]')
axs[3].set_ylabel('f [kHz]')
axs[4].set_ylabel('f [kHz]')
axs[5].set_ylabel('f [kHz]')
axs[6].set_ylabel('f [kHz]')
axs[7].set_ylabel('f [kHz]')
axs[8].set_ylabel('f [kHz]')
plt.ylim(0,1024)

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[0].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[0].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[0].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[1].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[1].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[1].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[2].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[2].fill_betweenx(y,x1,x2,facecolor="white")
```

```
x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[2].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[3].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[3].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[3].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[4].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[4].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[4].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[5].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[5].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[5].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[6].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[6].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[6].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[7].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[7].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[7].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[12]['DateTime']
x2=vlf.vlf_signal.iloc[189]['DateTime']
y=np.arange(0,1025)
axs[8].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[657]['DateTime']
x2=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[8].fill_betweenx(y,x1,x2,facecolor="white")

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
y=np.arange(0,1025)
axs[8].fill_betweenx(y,x1,x2,facecolor="white")


divider = make_axes_locatable(axs[0])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[0], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[1])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[1], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[2])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[2], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[3])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[3], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[4])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[4], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[5])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[5], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[6])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[6], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[7])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[7], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[8])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[8], cax=cax, orientation='vertical',label='dB')

axs[8].xaxis.set_ticks_position("bottom")
axs[8].xaxis.set_label_position("bottom")
axs[8].xaxis_date()
axs[8].xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
```

```
axs[8].xaxis.set_major_locator(mdates.MinuteLocator(interval=1))
vlf.vlf_signal.plot( x='DateTime', y='Frequency', ax=axs[8], alpha=0)

ax0=axs[8].twiny()
ax1=axs[8].twiny()


ax0.set_xlabel('GEO_LAT')
ax0.xaxis.set_ticks_position("bottom")
ax0.xaxis.set_label_position("bottom")
ax0.spines["bottom"].set_position(("axes", -0.3))
ax0.xaxis.grid(False)
ax0.set_xlim(GEO_LAT.min(), GEO_LAT.max())
ax0.xaxis.set_major_locator(ticker.FixedLocator(np.arange(GEO_LAT.min(), GEO_LAT.max(), 5)))
vlf.vlf_signal.plot(x='GEO_LAT', y='Frequency', ax=ax0,alpha=0)

ax1.set_xlabel('MAG_LAT')
ax1.xaxis.set_ticks_position("bottom")
ax1.xaxis.set_label_position("bottom")
ax1.spines["bottom"].set_position(("axes", -0.6))
ax1.xaxis.grid(False)
ax1.set_xlim(MAG_LAT.min(), MAG_LAT.max())
ax1.xaxis.set_major_locator(ticker.FixedLocator(np.arange(MAG_LAT.min(), MAG_LAT.max(), 0.5)))
vlf.vlf_signal.plot( x='MAG_LAT', y='Frequency', ax=ax1, alpha=0)

plt.savefig('/content/drive/MyDrive/Colab Notebooks/Plot/plot9.png',bbox_inches = 'tight')

plt.show()
```

In [ ]:

```
fig, axs = plt.subplots(6,1, sharex=True,figsize=(90,36))

axs[0].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Survey mode EFD VLF Ex-A131P power spectrum [mV/Hz^0.5]-"+ f"time_bin_survey={time_bin_survey}"
                 + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+ f"-vminSurvey= {survey_minX}" + f"-vmaxSurvey= {survey_maxX}")
axs[1].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Survey mode EFD VLF Ey-A132P power spectrum [mV/Hz^0.5]-"+ f"time_bin_survey={time_bin_survey}"
                 + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+ f"-vminSurvey= {survey_minX}" + f"-vmaxSurvey= {survey_maxY}")
axs[2].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Survey mode EFD VLF Ez-A133P power spectrum [mV/Hz^0.5]-"+ f"time_bin_survey={time_bin_survey}"
                 + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+ f"-vminSurvey= {survey_minZ}" + f"-vmaxSurvey= {survey_maxZ}")

axs[3].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode low resolution EFD VLF Ex [mV/m]- FFT A131_W wave form "+f"- time_bin_burst_low={time_bin_survey}"
                 + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+
                 f"-vminBurst_low resolution= {burst_low_minX}" + f"-vmaxBurst_low resolution= {burst_low_maxX}" )
axs[4].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode low resolution EFD VLF Ey [mV/m]- FFT A132_W wave form "+f"- time_bin_burst_low={time_bin_survey}"
                 + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+
                 f"-vminBurst_low resolution= {burst_low_minY}" + f"-vmaxBurst_low resolution= {burst_low_maxY}" )
axs[5].set_title(f"Orbit={OrbitNumber}" + f"-Date={DATE[0]}" + "-Burst mode low resolution EFD VLF Ez [mV/m]- FFT A133_W wave form "+f"- time_bin_burst_low={time_bin_survey}"
                 + f"-vminTot= {min_9plots}" + f"-vmaxTot= {max_9plots}"+
                 f"-vminBurst_low resolution= {burst_low_minZ}" + f"-vmaxBurst_low resolution= {burst_low_maxZ}" )

x_lims = list(map( lambda x:x, vlf.vlf_signal.DateTime))
x_lims = mdates.date2num(x_lims)
ext=[x_lims.min(),x_lims.max(),0,1024]
im = [0]*6
im[0] = axs[0].imshow(np.rot90(powerX),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_6plots,vmax=max_6plots)
im[1] = axs[1].imshow(np.rot90(powerY),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_6plots,vmax=max_6plots)
im[2] = axs[2].imshow(np.rot90(powerZ),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_6plots,vmax=max_6plots)

im[3] = axs[3].imshow(np.fliplr(np.rot90(inter2X,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_6plots,vmax=max_6plots)
im[4] = axs[4].imshow(np.fliplr(np.rot90(inter2Y,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_6plots,vmax=max_6plots)
im[5] = axs[5].imshow(np.fliplr(np.rot90(inter2Z,2)),interpolation='None',cmap='jet',aspect='auto', extent=ext,vmin=min_6plots,vmax=max_6plots)

axs[5].set_xlabel('DATE_TIME')
axs[0].set_ylabel('f [kHz]')
axs[1].set_ylabel('f [kHz]')
axs[2].set_ylabel('f [kHz]')
axs[3].set_ylabel('f [kHz]')
axs[4].set_ylabel('f [kHz]')
axs[5].set_ylabel('f [kHz]')
plt.ylim(0,1024)

x1=vlf.vlf_signal.iloc[0]['DateTime']
x2=vlf.vlf_signal.iloc[10]['DateTime']
x3=vlf.vlf_signal.iloc[12]['DateTime']
x4=vlf.vlf_signal.iloc[189]['DateTime']
x5=vlf.vlf_signal.iloc[657]['DateTime']
x6=vlf.vlf_signal.iloc[1076]['DateTime']
y=np.arange(0,1025)
axs[0].fill_betweenx(y,x1,x2,facecolor="white")

axs[0].fill_betweenx(y,x3,x4,facecolor="white")

axs[0].fill_betweenx(y,x5,x6,facecolor="white")
axs[1].fill_betweenx(y,x1,x2,facecolor="white")

axs[1].fill_betweenx(y,x3,x4,facecolor="white")

axs[1].fill_betweenx(y,x5,x6,facecolor="white")
axs[2].fill_betweenx(y,x1,x2,facecolor="white")

axs[2].fill_betweenx(y,x3,x4,facecolor="white")

axs[2].fill_betweenx(y,x5,x6,facecolor="white")
axs[3].fill_betweenx(y,x1,x2,facecolor="white")

axs[3].fill_betweenx(y,x3,x4,facecolor="white")

axs[3].fill_betweenx(y,x5,x6,facecolor="white")

axs[4].fill_betweenx(y,x1,x2,facecolor="white")


axs[4].fill_betweenx(y,x3,x4,facecolor="white")


axs[4].fill_betweenx(y,x5,x6,facecolor="white")


axs[5].fill_betweenx(y,x1,x2,facecolor="white")


axs[5].fill_betweenx(y,x3,x4,facecolor="white")


axs[5].fill_betweenx(y,x5,x6,facecolor="white")

divider = make_axes_locatable(axs[0])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[0], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[1])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[1], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[2])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[2], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[3])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[3], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[4])
```

```
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[4], cax=cax, orientation='vertical',label='dB')
divider = make_axes_locatable(axs[5])
cax = divider.append_axes('right',size="1%")
plt.colorbar(im[5], cax=cax, orientation='vertical',label='dB')

axs[5].xaxis.set_ticks_position("bottom")
axs[5].xaxis.set_label_position("bottom")
axs[5].xaxis_date()
axs[5].xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
axs[5].xaxis.set_major_locator(mdates.MinuteLocator(interval=1))
vlf.vlf_signal.plot( x='DateTime', y='Frequency', ax=axs[5], alpha=0)

ax0=axs[5].twiny()
ax1=axs[5].twiny()


ax0.set_xlabel('GEO_LAT')
ax0.xaxis.set_ticks_position("bottom")
ax0.xaxis.set_label_position("bottom")
ax0.spines["bottom"].set_position(("axes", -0.3))
ax0.xaxis.grid(False)
ax0.set_xlim(GEO_LAT.min(), GEO_LAT.max())
ax0.xaxis.set_major_locator(ticker.FixedLocator(np.arange(GEO_LAT.min(), GEO_LAT.max(), 5)))
vlf.vlf_signal.plot(x='GEO_LAT', y='Frequency', ax=ax0,alpha=0)

ax1.set_xlabel('MAG_LAT')
ax1.xaxis.set_ticks_position("bottom")
ax1.xaxis.set_label_position("bottom")
ax1.spines["bottom"].set_position(("axes", -0.6))
ax1.xaxis.grid(False)
ax1.set_xlim(MAG_LAT.min(), MAG_LAT.max())
ax1.xaxis.set_major_locator(ticker.FixedLocator(np.arange(MAG_LAT.min(), MAG_LAT.max(), 0.5)))
vlf.vlf_signal.plot( x='MAG_LAT', y='Frequency', ax=ax1, alpha=0)

plt.savefig('/content/drive/MyDrive/Colab Notebooks/Plot/6plots.png',bbox_inches = 'tight')

plt.show()
```

In [ ]:
```
def zoom(filter, array):
  df_1 = df_burst.query(filter)
  df_1S_A131_W = np.array(df_1[array].tolist())
  data2s=np.asarray([df_1S_A131_W[i]  for i in range (0,len(df_1S_A131_W)) if  isinstance(df_1S_A131_W[i], np.ndarray)])
  data_2s = np.empty(shape=(data2s.shape[0], data2s.shape[1]))
  for i in range (0, data2s.shape[0]):
    meanX_b = np.mean(data2s[i])
    data_2s[i] = data2s[i] - meanX_b

  outX_1S=data_2s.ravel().reshape(-1,2048)
  M_b = outX_1S.shape[1]
  hamming_b2s = signal.get_window("hamming", M_b)
  FFT2sec=np.array([scipy.fft.fft(outX_1S[i]*hamming_b2s) for i in range(0,outX_1S.shape[0])])
  inter1= np.abs(FFT2sec.T[:1024])**2

  return inter1,df_1
```

In [ ]:
```
inter1X, df1X=zoom('20180807072852815 <= UTC_TIME <= 20180807072853839','X')
inter1Y, df1Y=zoom('20180807072852815 <= UTC_TIME <= 20180807072853839','Signal')
inter1Z, df1Z=zoom('20180807072852815 <= UTC_TIME <= 20180807072853839','Z')
inter2X, df2X=zoom('20180807072851791 <= UTC_TIME <= 20180807072853839','X')
inter2Y, df2Y=zoom('20180807072851791 <= UTC_TIME <= 20180807072853839','Signal')
inter2Z, df2Z=zoom('20180807072851791 <= UTC_TIME <= 20180807072853839','Z')
inter30X, df30X=zoom('20180807072829983 <= UTC_TIME <= 20180807072859983','X')
inter30Y, df30Y=zoom('20180807072829983 <= UTC_TIME <= 20180807072859983','Signal')
inter30Z, df30Z=zoom('20180807072829983 <= UTC_TIME <= 20180807072859983','Z')
inter1mX, df1mX=zoom('20180807072800591 <= UTC_TIME <= 20180807072859983','X')
inter1mY, df1mY=zoom('20180807072800591 <= UTC_TIME <= 20180807072859983','Signal')
inter1mZ, df1mZ=zoom('20180807072800591 <= UTC_TIME <= 20180807072859983','Z')
inter5mX, df5mX=zoom('20180807072005519 <= UTC_TIME <= 20180807072512687','X')
inter5mY, df5mY=zoom('20180807072005519 <= UTC_TIME <= 20180807072512687','Signal')
inter5mZ, df5mZ=zoom('20180807072005519 <= UTC_TIME <= 20180807072512687','Z')
```

In [ ]:
```
def zoomInter(filter, array):
  df_1 = df_burst.query(filter)
  df_1S_A131_W = np.array(df_1[array].tolist())
  data2s=np.asarray([df_1S_A131_W[i]  for i in range (0,len(df_1S_A131_W)) if  isinstance(df_1S_A131_W[i], np.ndarray)])
  data_2s = np.empty(shape=(data2s.shape[0], data2s.shape[1]))
  for i in range (0, data2s.shape[0]):
    meanX_b = np.mean(data2s[i])
    data_2s[i] = data2s[i] - meanX_b

  zero=np.zeros_like(data_2s)
  s=np.array([[i,j]for i,j in zip(data_2s,zero)]).reshape(2*data_2s.shape[0],data_2s.shape[1])
  outX_1S=s.ravel().reshape(-1,2048)

  M_b = outX_1S.shape[1]
  hamming_b2s = signal.get_window("hamming", M_b)
  FFT2sec=np.array([scipy.fft.fft(outX_1S[i]*hamming_b2s) for i in range(0,outX_1S.shape[0])])
  inter1= np.abs(FFT2sec.T[:1024])**2
  inter1S = 70+20*np.log10(inter1/Bw, where=0<inter1, out=np.nan*inter1)
  return inter1S,df_1
```

In [ ]:
```
inter1X2, df1X=zoomInter('20180807072852815 <= UTC_TIME <= 20180807072853839','X')
inter1Y2, df1Y=zoomInter('20180807072852815 <= UTC_TIME <= 20180807072853839','Signal')
inter1Z2, df1Z=zoomInter('20180807072852815 <= UTC_TIME <= 20180807072853839','Z')
inter2X2, df2X=zoomInter('20180807072851791 <= UTC_TIME <= 20180807072853839','X')
inter2Y2, df2Y=zoomInter('20180807072851791 <= UTC_TIME <= 20180807072853839','Signal')
inter2Z2, df2Z=zoomInter('20180807072851791 <= UTC_TIME <= 20180807072853839','Z')
inter30X2, df30X=zoomInter('20180807072829983 <= UTC_TIME <= 20180807072859983','X')
inter30Y2, df30Y=zoomInter('20180807072829983 <= UTC_TIME <= 20180807072859983','Signal')
inter30Z2, df30Z=zoomInter('20180807072829983 <= UTC_TIME <= 20180807072859983','Z')
inter1mX2, df1mX=zoomInter('20180807072800591 <= UTC_TIME <= 20180807072859983','X')
inter1mY2, df1mY=zoomInter('20180807072800591 <= UTC_TIME <= 20180807072859983','Signal')
inter1mZ2, df1mZ=zoomInter('20180807072800591 <= UTC_TIME <= 20180807072859983','Z')
inter5mX2, df5mX=zoomInter('20180807072005519 <= UTC_TIME <= 20180807072512687','X')
inter5mY2, df5mY=zoomInter('20180807072005519 <= UTC_TIME <= 20180807072512687','Signal')
inter5mZ2, df5mZ=zoomInter('20180807072005519 <= UTC_TIME <= 20180807072512687','Z')
```

In [ ]:
```
def zoomPlot(array, filter,output, title):
  fig, ax = plt.subplots()
  x_lims = list(map( lambda x:x, filter.DateTime))
  x_lims = mdates.date2num(x_lims)
  ext=[x_lims.min(),x_lims.max(),0,1024]

  plt.rcParams["figure.figsize"] = (20,3)
  plt.ylim(0,1024)
  ax.set_title(OrbitNumber+": focus on Burst mode high resolution " +DATE[0] +title)
  im=plt.gca().imshow(70+20*np.log10(np.fliplr(np.rot90(array,2))), interpolation='none', cmap='jet',aspect='auto',extent=ext)
  #im=plt.gca().imshow(70*np.log10(array), interpolation='none', cmap='jet',aspect='auto')
  plt.gca()

  ax.set_ylabel('f [kHz]')
```

```
  ax.set_xlabel('DATE_TIME')

  ax.xaxis_date()
  ax.xaxis.set_ticks_position("bottom")
  ax.xaxis.set_label_position("bottom")
  ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M:%S'))
  ax.xaxis.set_major_locator(mdates.SecondLocator( interval=  5))
  ax.set_xlim(filter.DateTime.min(),filter.DateTime.max())
  plt.xticks(rotation = 45)

  fig.colorbar(im, ax=ax, pad=0.05)
  plt.savefig(f'/content/drive/MyDrive/Colab Notebooks/Plot/{output}.png',bbox_inches = 'tight')
  plt.show()
```

In [ ]:
```
def zoomPlotInter(array, filter,output, title):
  fig, ax = plt.subplots()
  x_lims = list(map( lambda x:x, filter.DateTime))
  x_lims = mdates.date2num(x_lims)
  ext=[x_lims.min(),x_lims.max(),0,1024]

  plt.rcParams["figure.figsize"] = (20,3)
  plt.ylim(0,1024)
  ax.set_title(OrbitNumber+": focus on Burst mode high resolution " +DATE[0] +title)
  im=plt.gca().imshow(np.fliplr(np.rot90(array,2)), interpolation='none', cmap='jet',aspect='auto',extent=ext)
  #im=plt.gca().imshow(array, interpolation='none', cmap='jet',aspect='auto')
  plt.gca()

  ax.set_ylabel('f [kHz]')
  ax.set_xlabel('DATE_TIME')

  ax.xaxis_date()
  ax.xaxis.set_ticks_position("bottom")
  ax.xaxis.set_label_position("bottom")
  ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M:%S'))
  ax.xaxis.set_major_locator(mdates.SecondLocator( interval=  5))
  ax.set_xlim(filter.DateTime.min(),filter.DateTime.max())
  plt.xticks(rotation = 45)

  fig.colorbar(im, ax=ax, pad=0.05)
  plt.savefig(f'/content/drive/MyDrive/Colab Notebooks/Plot/{output}.png',bbox_inches = 'tight')
  plt.show()
```

In [ ]:
```
zoomPlot(inter1X,df1X,'1secXinter',"-EFDX-1second-7.28.52815-7.28.53839 ")
```

In [ ]:
```
zoomPlot(inter1Y,df1Y,'1secYinter',"-EFDY-1second-7.28.52815-7.28.53839 ")
```

In [ ]:
```
zoomPlot(inter1Z,df1Z,'1secZinter',"-EFDZ-1second-7.28.52815-7.28.53839 ")
```

In [ ]:
```
zoomPlotInter(inter2X2,df2X,'2secXiter',"-EFDX-2seconds-7.28.51791-7.28.53839 ")
```

In [ ]:
```
zoomPlotInter(inter2Y2,df2Y,'2secYiter',"-EFDY-2seconds-7.28.51791-7.28.53839 ")
```

In [ ]:
```
zoomPlotInter(inter2Z2,df2Z,'2secZiter',"-EFDZ-2seconds-7.28.51791-7.28.53839 ")
```

In [ ]:
```
zoomPlotInter(inter30X2,df30X,'30secXinter',"-EFDX-30seconds-7.2829983-7.2859983 ")
```

In [ ]:
```
zoomPlotInter(inter30Y2,df30Y,'30secYinter',"-EFDY-30seconds-7.2829983-7.2859983 ")
```

In [ ]:
```
zoomPlotInter(inter30Z2,df30Z,'30secZinter',"-EFDZ-30seconds-7.2829983-7.2859983 ")
```

In [ ]:
```
zoomPlotInter(inter1mX2,df1mX,'1minXinter',"-EFDX-1minute-7.2800591-7.2859983 ")
```

In [ ]:
```
zoomPlotInter(inter1mY2,df1mY,'1minYinter',"-EFDY-1minute-7.2800591-7.2859983 ")
```

In [ ]:
```
zoomPlotInter(inter1mZ2,df1mZ,'1minZinter',"-EFDZ-1minute-7.2800591-7.2859983 ")
```

In [ ]:
```
zoomPlotInter(inter5mX2,df5mX,'5minXinter',"-EFDX-5minutes-7.2005519-7.2512687 ")
```

In [ ]:
```
zoomPlotInter(inter5mY2,df5mY,'5minYinter',"-EFDy-5minutes-7.2005519-7.2512687 ")
```

In [ ]:
```
zoomPlotInter(inter5mZ2,df5mZ,'5minZinter',"-EFDZ-5minutes-7.2005519-7.2512687 ")
```

In [ ]:
```
zoomPlot(inter1X,df1X,'1secX',"-EFDX-1second-7.28.52815-7.28.53839 ")
```

In [ ]:
```
zoomPlot(inter1Y,df1Y,'1secY',"-EFDY-1second-7.28.52815-7.28.53839 ")
```

In [ ]:
```
zoomPlot(inter1Z,df1Z,'1secZ',"-EFDZ-1second-7.28.52815-7.28.53839 ")
```

In [ ]:
```
zoomPlot(inter2X,df2X,'2secX',"-EFDX-2seconds-7.28.51791-7.28.53839 ")
```

In [ ]:
```
zoomPlot(inter2Y,df2Y,'2secY',"-EFDY-2seconds-7.28.51791-7.28.53839 ")
```

In [ ]:
```
zoomPlot(inter2Z,df2Z,'2secZ',"-EFDZ-2seconds-7.28.51791-7.28.53839 ")
```

In [ ]:
```
zoomPlot(inter30X,df30X,'30secX',"-EFDX-30seconds-7.2829983-7.2859983 ")
```

In [ ]:
```
zoomPlot(inter30Y,df30Y,'30secY',"-EFDY-30seconds-7.2829983-7.2859983 ")
```

In [ ]:
```
zoomPlot(inter30Z,df30Z,'30secZ',"-EFDZ-30seconds-7.2829983-7.2859983 ")
```

In [ ]:
```
zoomPlot(inter1mX,df1mX,'1minX',"-EFDX-1minute-7.2800591-7.2859983 ")
```

In [ ]:
```

```

```
zoomPlot(inter1mY,df1mY,'1minY',"-EFDY-1minute-7.2800591-7.2859983 ")
```

In [ ]:
```
zoomPlot(inter1mZ,df1mZ,'1minZ',"-EFDZ-1minute-7.2800591-7.2859983 ")
```

In [ ]:
```
zoomPlot(inter5mX,df5mX,'5minX',"-EFDX-5minutes-7.2005519-7.2512687 ")
```

In [ ]:
```
zoomPlot(inter5mY,df5mY,'5minY',"-EFDY-5minutes-7.2005519-7.2512687 ")
```

In [ ]:
```
zoomPlot(inter5mZ,df5mZ,'5minZ',"-EFDZ-5minutes-7.2005519-7.2512687 ")
```