Test_Flight_20151110

November 10, 2015

In this report, we'll review the test flight of the ER-2 on 11/10/2015. This was a short, ~2-h flight that spent some time off the CA coast. It was relatively similar to the previous day's flight, just a bit shorter. The ER-2 overflew several buoys. First let's import all the needed modules and ingest and process the raw data.

```
In [1]: from __future__ import print_function
        import numpy as np
        import matplotlib.pyplot as plt
        import os
        import glob
        import rawpyampr
        import tempfile
        import pyampr
        from IPython.display import Image
        %matplotlib inline
```

```
In [3]: import warnings
        warnings.filterwarnings('ignore')
        def delete_file(fname):
            try:
                os.remove(fname)
            except:
                pass
```

```
In [4]: datadir = './'
        files = glob.glob(datadir + '*.dat')
        print(files)
        fname = os.path.basename(files[0])[:-4]
```

```
['./AMPR-20151110-164454.dat']
```

```
In [5]: payload = rawpyampr.ampr_payload.AMPR_Payload(files[0])
        l1file = fname + '_L1.nc'
        l2file = fname + '_L2.nc'
        delete_file(l1file)
        payload.writeLevel1B(l1file)
        L1B = rawpyampr.ampr_level1b.AMPR_QC(l1file)
        delete_file(l2file)
        L1B.writeLevel2B(l2file)
```

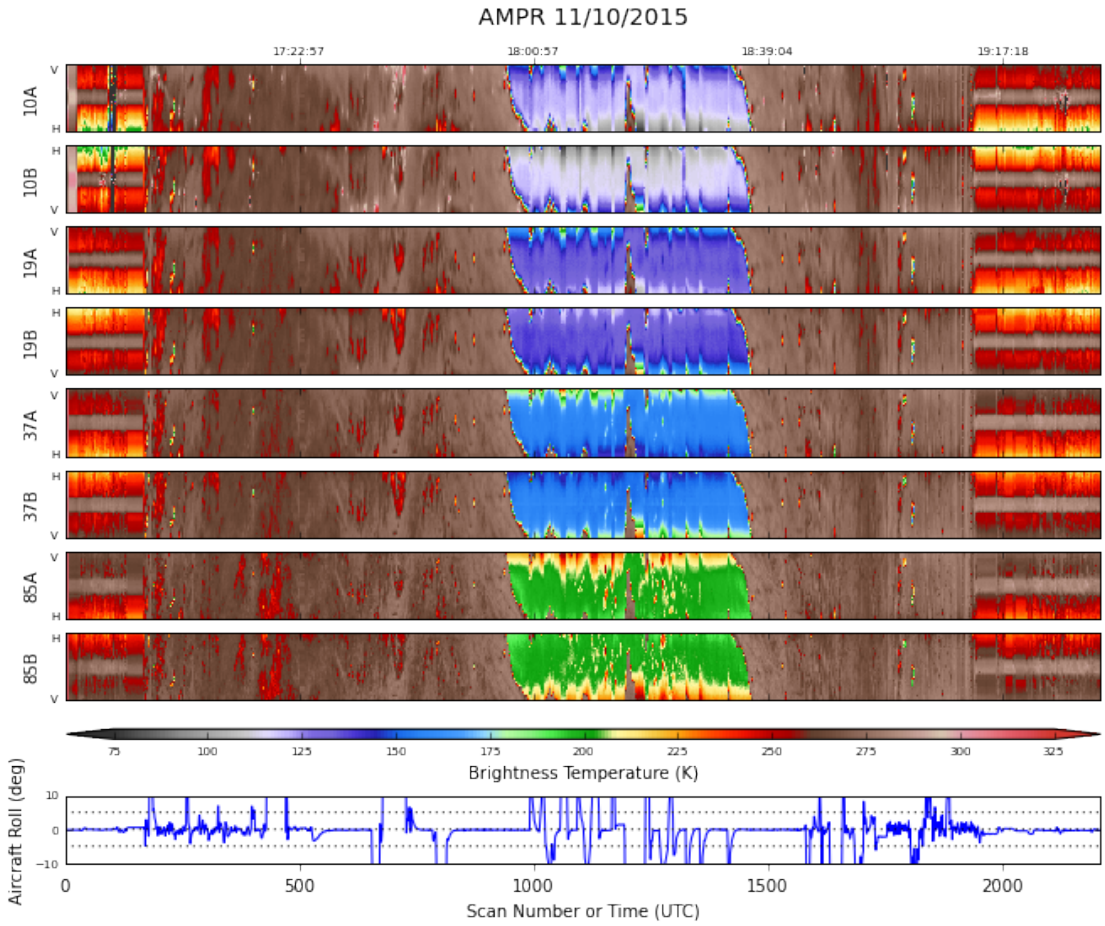
```
All of file:  ./AMPR-20151110-164454.dat  : Read Successfully
End of data stream reached
Interpreting Navigation Records as: IWG1
 No navigation file found
Navigating pixels using internal recording of nav data.
```

```
Number points to converge: 4
Writing to output file:  AMPR-20151110-164454_L1.nc
Found Navigation Data!
Writing to output file: AMPR-20151110-164454_L2.nc
File containing water fraction not on path
```

Now we are ready to read in and display the L2 geolocated brightness temperatures.

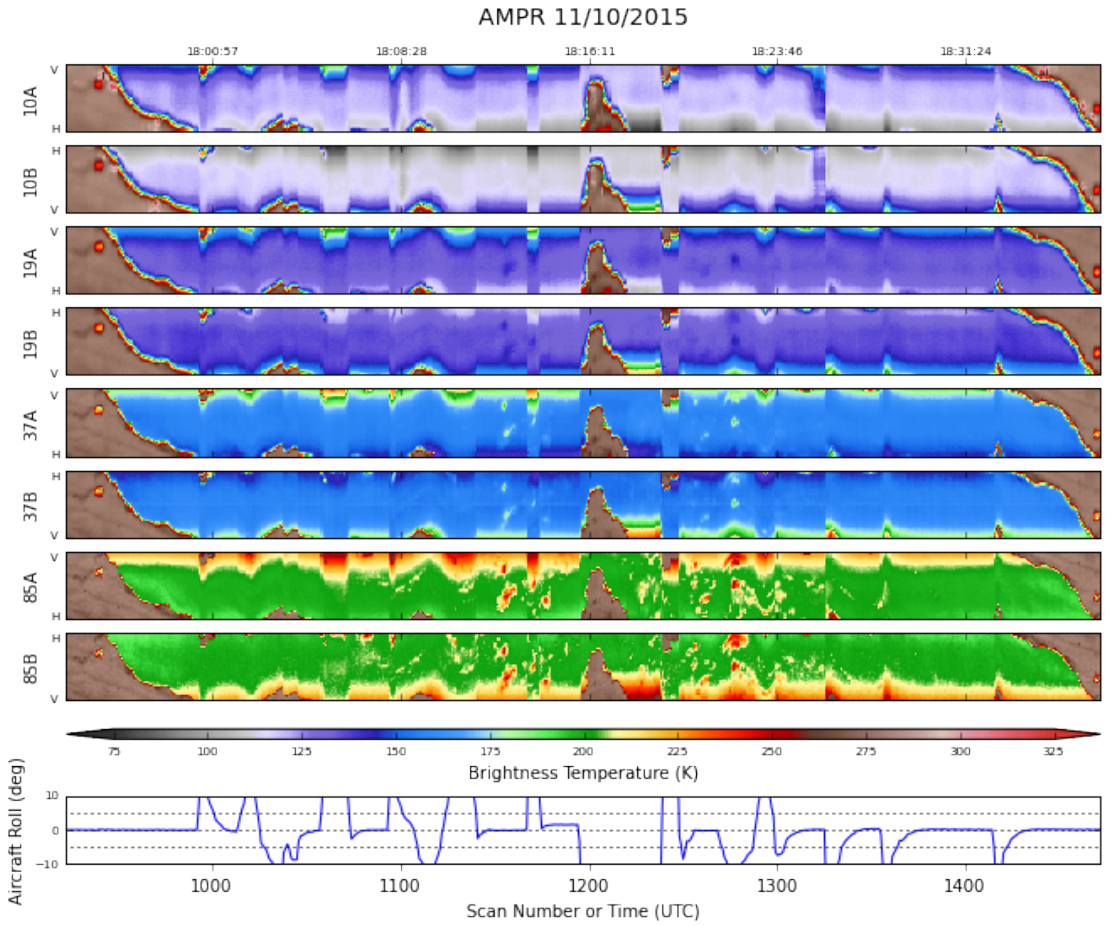
```
In [6]: data = pyampr.AmprTb(l2file)
        data.plot_ampr_channels()
```

```
********************
read_ampr_tb_level2b(): Reading AMPR-20151110-164454_L2.nc
Assuming IPHEX data structure.
Change to proper project if incorrect, otherwise errors will occur.
Currently available field projects: IPHEX, MC3E, TC4, TCSP, JAX90, COARE,
CAMEX1, CAMEX2, CAMEX3, CAMEX4, TRMMLBA, KWAJEX, TEFLUNA, FIRE3ACE, CAPE
Default: project = 'IPHEX'
Found Navigation Data!
(2208,)
********************


********************
plot_ampr_channels():
Available scans = 1 to 2208
Available times = 16:44:58 - 19:33:04
********************
```



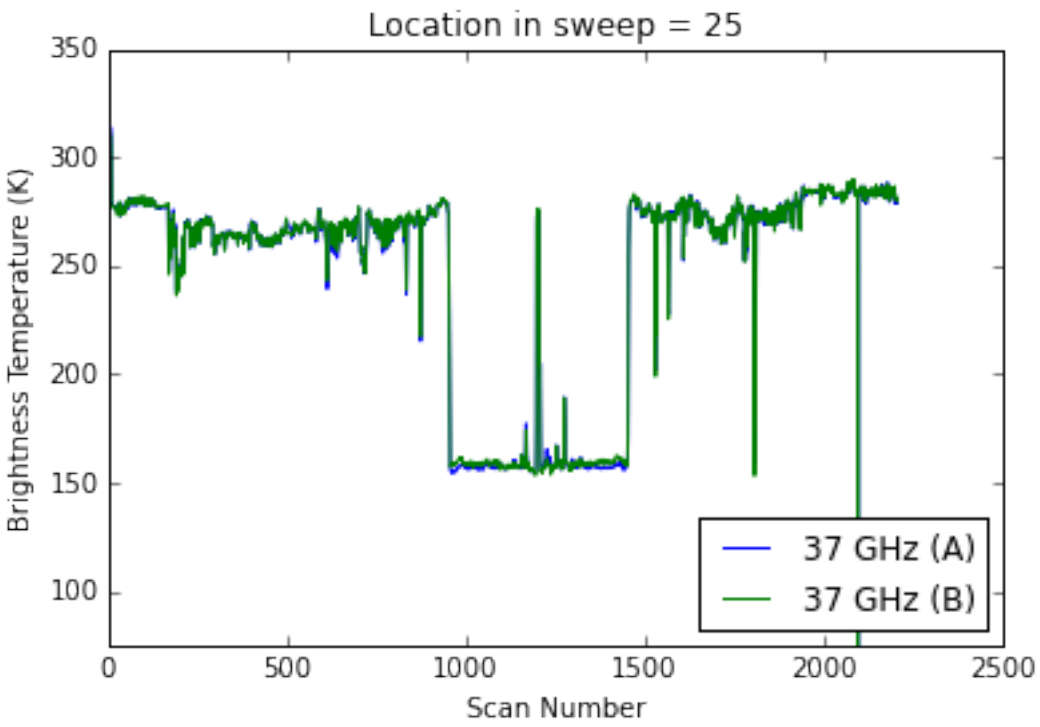
Overall, this looks pretty good. No notable outages, even short-lived ones. Let's investigate the over-water portion of the flight for now.

```
In [10]: data.plot_ampr_channels(timerange=['17:55:00', '18:37:00'])

********************
plot_ampr_channels():
Available scans = 1 to 2208
Available times = 16:44:58 - 19:33:04
********************
```



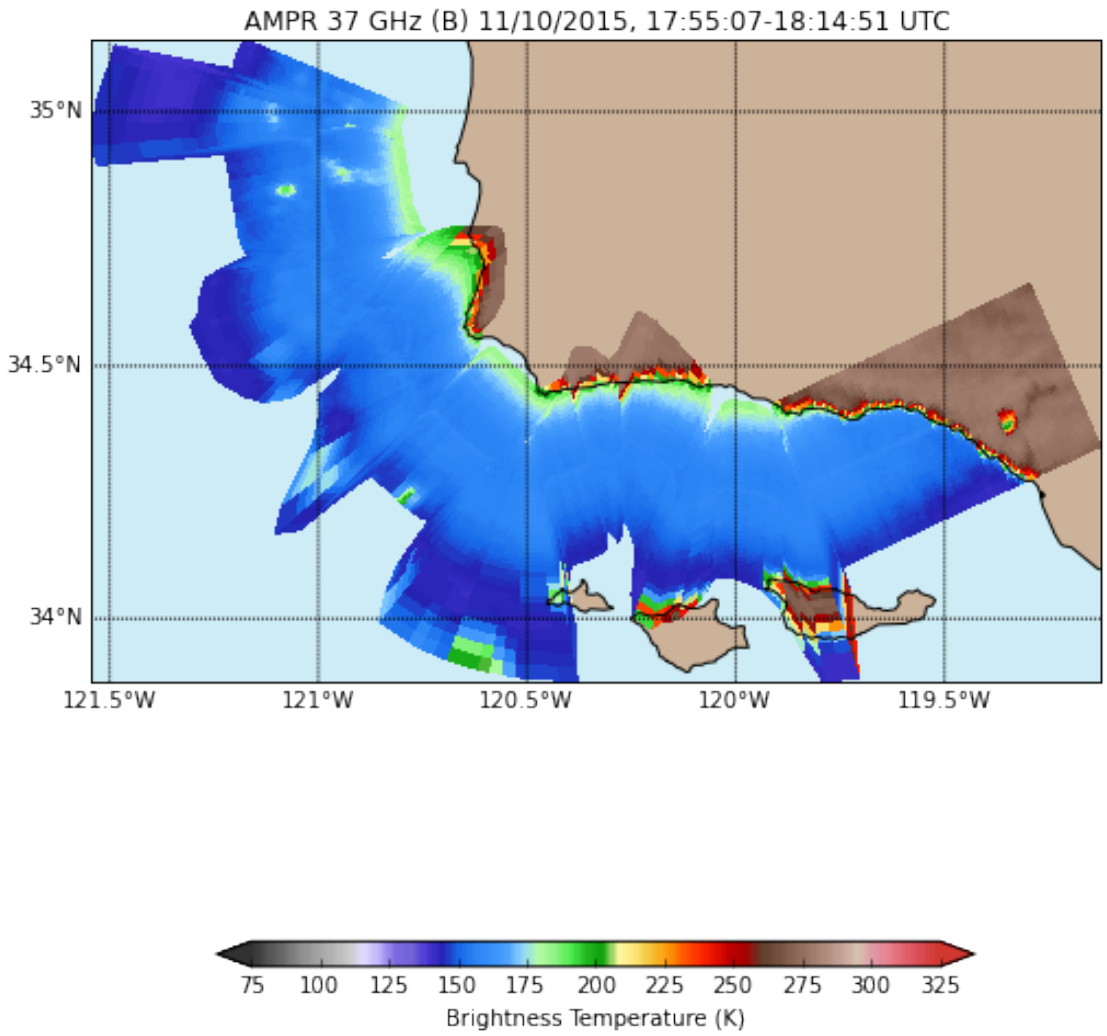
Not as much precipitation out there today, so the buoy overflights should be more useful for calibration. There is nadir striping from HIWRAP during this portion of the flight, as expected, leading to a slight difference between the nadir time series of 37 GHz A & B.

```
In [11]: chan = '37'
         ylim = [75, 350]
         index = 25
         tsa = getattr(data, 'TB'+chan+'A')[:, index]
         tsb = getattr(data, 'TB'+chan+'B')[:, index]
         cond = tsa > -1000
         plt.plot(np.arange(len(tsa))[cond], tsa[cond], label=chan+' GHz (A)')
         plt.plot(np.arange(len(tsa))[cond], tsb[cond], label=chan+' GHz (B)')
         plt.ylim(ylim)
         plt.xlabel('Scan Number')
         plt.ylabel('Brightness Temperature (K)')
         plt.legend(loc='lower right')
         plt.title('Location in sweep = '+str(index))

Out[11]: <matplotlib.text.Text at 0x1120e2b00>
```



Let's check out 37 GHz (B) channel on the outbound portion of the over-water flight . The striping occurs during straight and level flight over the water, as expected. During maneuvers (when the "fanning" of the AMPR swath occurs) the striping goes away.

```
In [20]: stuff = data.plot_ampr_track('37b', timerange=['17:55:00', '18:15:00'],
                                      show_track=False, return_flag=True,
                                      meridians=0.5, parallels=0.5, resolution='h')
         stuff[2].fillcontinents(color='#CCB299', lake_color='#CEECF5',
                                 ax=stuff[1], zorder=0)
         stuff[2].drawmapboundary(fill_color='#CEECF5', ax=stuff[1])

********************
plot_ampr_track():
Available scans = 1 to 2208
Available times = 16:44:58 - 19:33:04
********************

Out[20]: <matplotlib.patches.Rectangle at 0x114a98cf8>
```



Not much else to say about this flight. The data look good and AMPR is ready for OLYMPEX. Still no realtime AMPR data, as Ames continues to troubleshoot a firewall issue. However, we are now receiving realtime status packets, so some progress has been made

In [ ]:

In [ ]:

In [ ]: