Science_Flight_20151204

December 5, 2015

In this report, we'll review the science flight of the ER-2 starting on 12/04/2015. This was a ~7-h flight that sampled offshore and landfalling post-frontal convection. First, let's import all the needed modules and ingest and process the raw data.

```
In [32]: from __future__ import print_function
         import numpy as np
         import matplotlib.pyplot as plt
         import datetime as dt
         import os
         import glob
         import pyart
         import rawpyampr
         import pyampr
         import dualpol
         from copy import deepcopy
         from awot.graph.common import create_basemap
         from awot.graph.flight_level import FlightLevel
         from pyart_tools import (
             plot_list_of_fields, list_nexrad_files, read_nexrad_aws)
         %matplotlib inline
```

```
In [2]: import warnings
        warnings.filterwarnings('ignore')
        def delete_file(fname):
            try:
                os.remove(fname)
            except:
                pass
```

```
In [3]: datadir = './'
        files = glob.glob(datadir + '*.dat')
        print(files)
        fname = os.path.basename(files[0])[:-4]
```

```
['./AMPR-20151204-100018.dat', './AMPR-20151204-124104.dat', './AMPR-20151204-171046.dat']
```

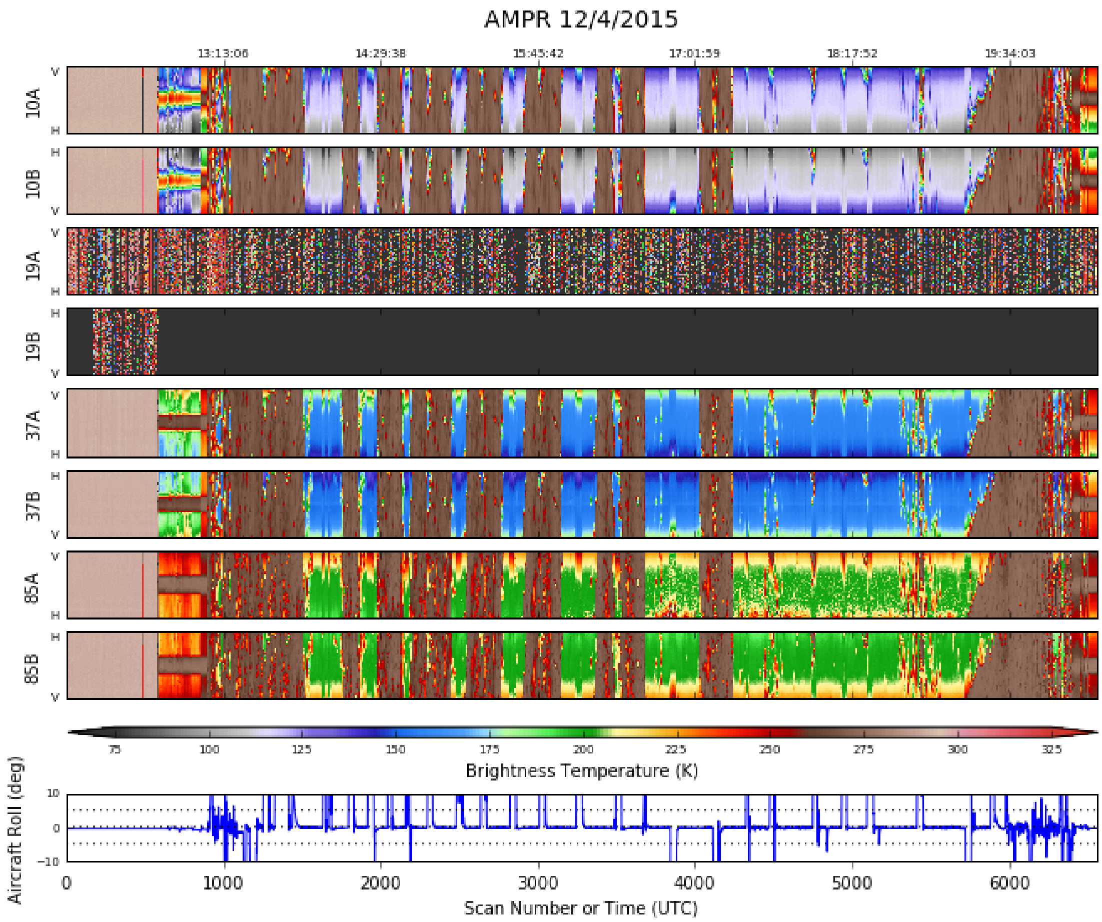
```
In [4]: payload = rawpyampr.ampr_payload.AMPR_Payload(files)
        l1file = fname + '_L1.nc'
        l2file = fname + '_L2.nc'
        delete_file(l1file)
        payload.writeLevel1B(l1file)
        L1B = rawpyampr.ampr_level1b.AMPR_QC(l1file)
        delete_file(l2file)
        L1B.writeLevel2B(l2file)
```

```
All of file: ./AMPR-20151204-100018.dat Read Successfully
End of data stream reached
All of file: ./AMPR-20151204-124104.dat Read Successfully
End of data stream reached
All of file: ./AMPR-20151204-171046.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-20151204-100018_L1.nc
Found Navigation Data!
Writing to output file: AMPR-20151204-100018_L2.nc
File containing water fraction not on path
```

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

```
In [5]: data = pyampr.AmprTb(l2file)
        data.plot_ampr_channels()
```

```
********************
read_ampr_tb_level2b(): Reading AMPR-20151204-100018_L2.nc
Assuming OLYMPEX data structure.
Change to proper project if incorrect, otherwise errors will occur.
Currently available field projects: OLYMPEX, IPHEX, MC3E, TC4, TCSP, JAX90, COARE,
CAMEX1, CAMEX2, CAMEX3, CAMEX4, TRMMLBA, KWAJEX, TEFLUNA, FIRE3ACE, CAPE
Default: project = 'OLYMPEX'
Found Navigation Data!
(6558,)
********************


********************
plot_ampr_channels():
Available scans = 1 to 6558
Available times = 10:00:22 - 20:17:07
********************
```

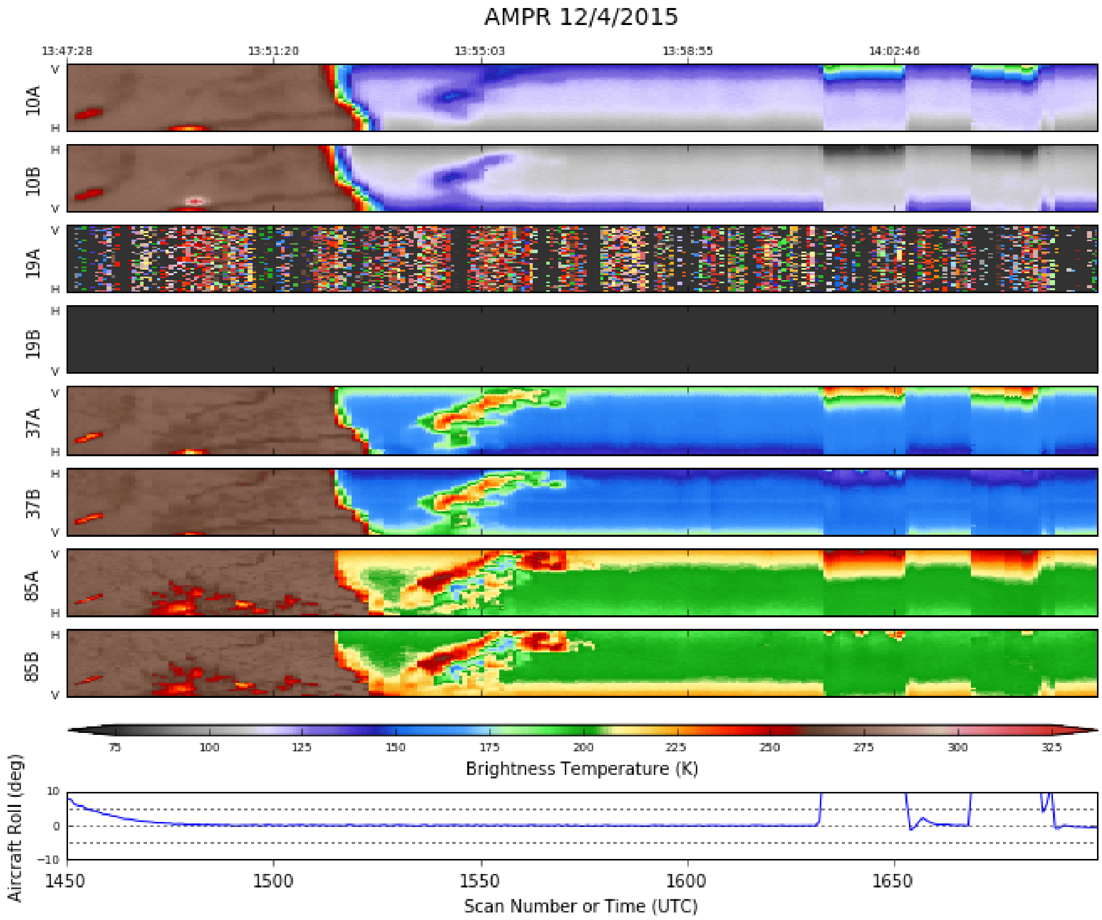


As can be seen, we did not have the 19 GHz channels again during this flight. Most other channels behaved well, although there may have been some noise issues on the 85 GHz (A) channel during the latter portions of the flight. The early part of the chart consists of engineering test data, which is why the functioning channels look so uniform at first. We continue to troubleshoot the 19 GHz issue, and are planning to pull the instrument after the 12/05/2015 flight to check it out during the planned down day on 12/06/2015.

For now, however, we will focus on the science. The ER-2, after gaining altitude, performed coordinated overflights of post-frontal convection that was moving quickly onshore. Let's examine one of these overpasses.

```
In [89]: data.plot_ampr_channels(scanrange=[1450, 1700])
```

```
********************
plot_ampr_channels():
Available scans = 1 to 6558
Available times = 10:00:22 - 20:17:07
********************
```



The precipitation system of interest is the linear feature that is just offshore. There are some interesting differences in the emission signals between 10, 37, and 85 GHz. Let's look at the geolocated data during this time. Since we're missing the 19 GHz channel, let's overplot NEXRAD data in its place.

```
In [14]: # Let's grab NEXRAD data for this case
         radlist = list_nexrad_files(2015, month='12', day='04', station='KLGX')
         radar = read_nexrad_aws('KLGX20151204_135318_V06.gz')
```

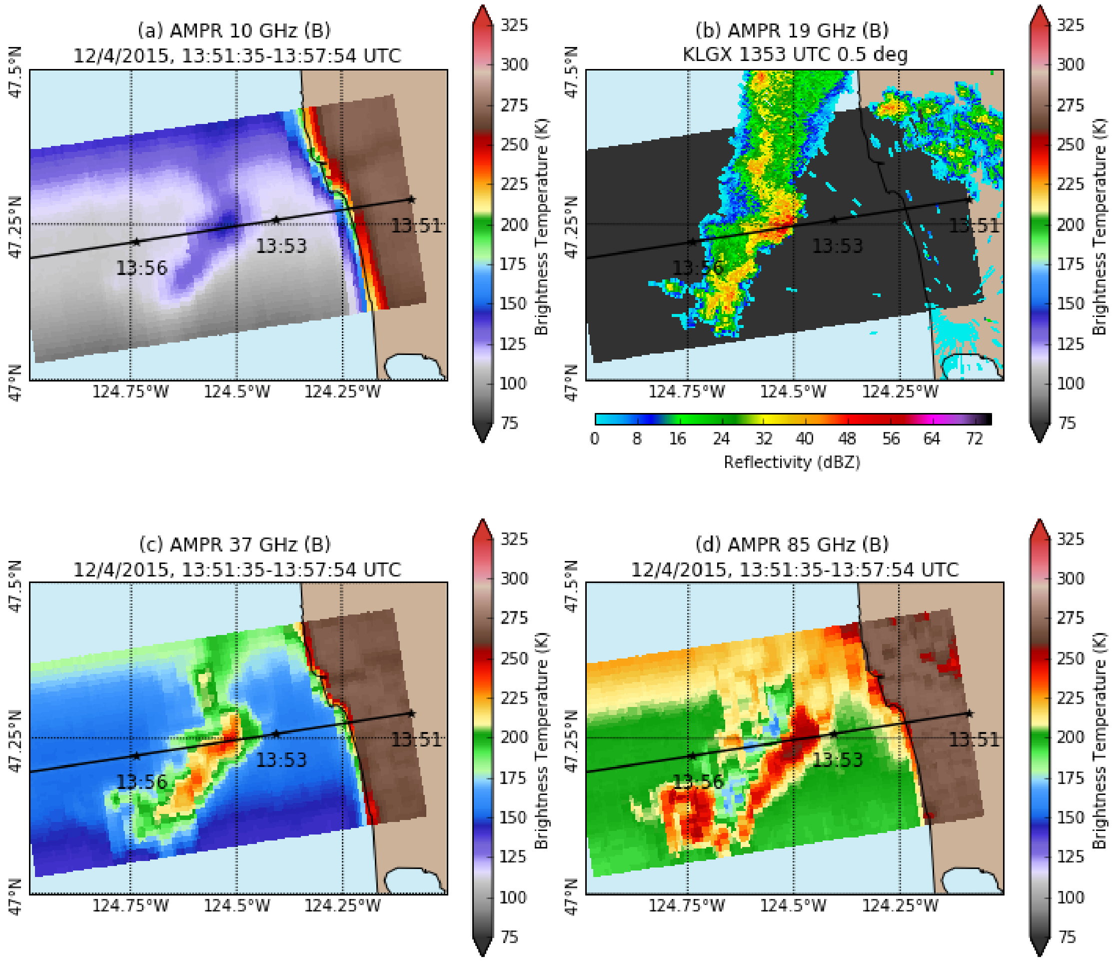
```
https://noaa-nexrad-level2.s3.amazonaws.com/2015/12/04/KLGX/KLGX20151204_135318_V06.gz
```

```
In [33]: # Do some quick QC on the NEXRAD data
         zc = deepcopy(radar.fields['REF']['data'])
         radar.add_field_like('REF', 'ZC', zc, replace_existing=True)
         retrieve = dualpol.DualPolRetrieval(
             radar, dz='ZC', dr='ZDR', dp='PHI', rh='RHO', use_temp=False,
             dsd_flag=False, fhc_flag=False, precip_flag=False,
             liquid_ice_flag=False, kdp_window=5, verbose=False,
             ice_flag=False, qc_flag=True, gs=250.0,
             thresh_sdp=20.0, speckle=3,
             thresh_dr=np.array(dualpol.DEFAULT_DR_THRESH)+1.0)
```

```
0.606287956237793 seconds to run csu_kdp
No sounding provided
```

In [70]:
```python
# Import the ER-2 nav into AWOT (https://github.com/nguy/AWOT)
# This simplifies plotting the track with time stamps
flight = pyampr.read_aircraft_nav_into_awot(data)
tst = '13:51:30'
ted = '13:58:00'
start = '2015-12-04 ' + tst
end = '2015-12-04 ' + ted
offs = (-0.05, -0.05)
```

In [72]:
```python
display = data.plot_ampr_track_4panel(
    chan='b', timerange=[tst, ted], maneuver=False, return_flag=True,
    meridians=0.25, parallels=0.25, resolution='h',
    show_grid=True, latrange=[47, 47.5], lonrange=[-124.99, -124])
display.ax2.set_title('(b) AMPR 19 GHz (B) \nKLGX 1353 UTC 0.5 deg')

# Play some tricks to overplot NEXRAD on the same basemap using Py-ART
dr = pyart.graph.RadarMapDisplay(radar)
rdata = dr._get_data('ZC', 1, None, True, None)
x, y = dr._get_x_y('ZC', 1, True, True)
_x0, _y0 = display.basemap(dr.loc[1], dr.loc[0])
pm = display.basemap.pcolormesh(
    _x0 + x * 1000., _y0 + y * 1000.,
    rdata, vmin=0, vmax=75, cmap='pyart_NWSRef', ax=display.ax2)
cax = display.fig.add_axes([0.52, 0.55, 0.35, 0.01])
plt.colorbar(pm, label='Reflectivity (dBZ)', cax=cax, orientation='horizontal')

# And plot flight times too
f2 = FlightLevel(flight, basemap=display.basemap)
for ax in [display.ax1, display.ax2, display.ax3, display.ax4]:
    f2.plot_trackmap(min_altitude=50., lw=2.5, start_time=start,
                     end_time=end, ax=ax)
    f2.time_stamps(start_time=start, end_time=end,
                   labelspacing=30, ax=ax, label_offset=offs)
plt.savefig('combo_ampr_nexrad.png')
```

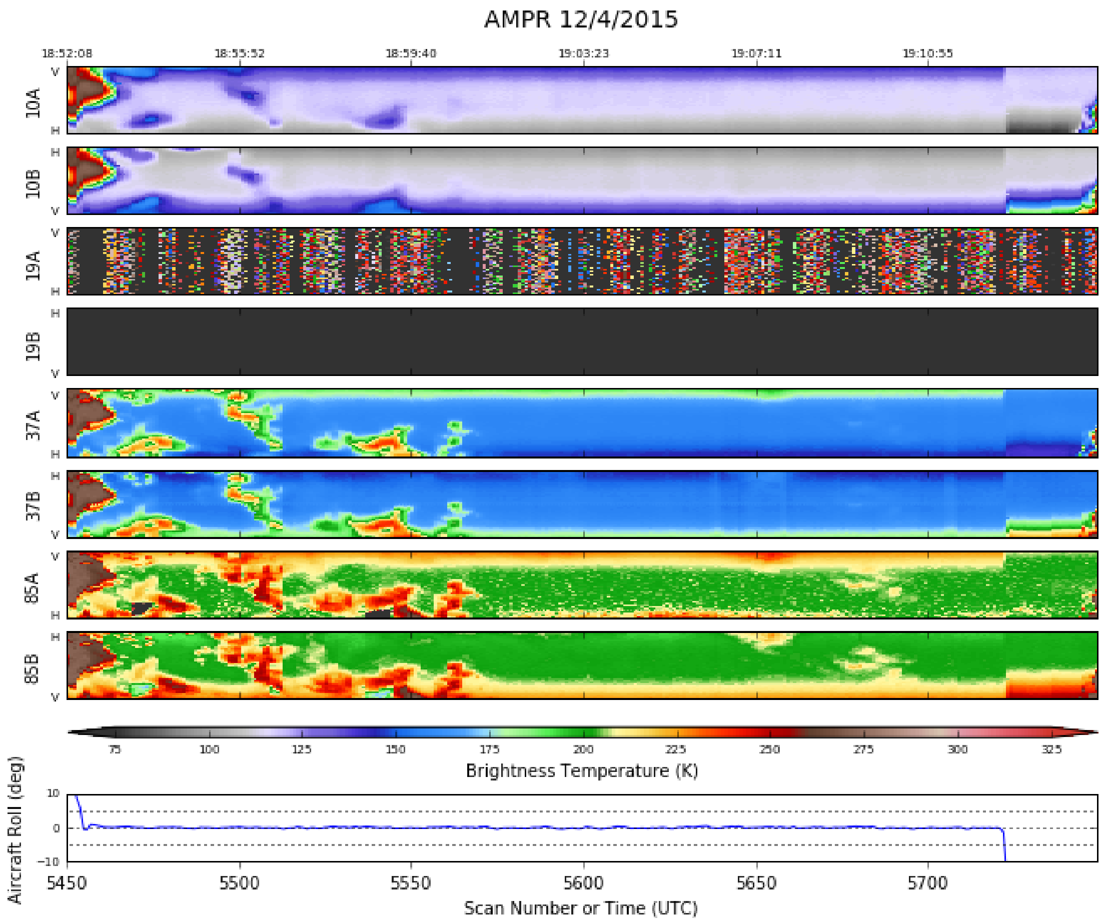


Looks like some possible ice scattering at 85 GHz, just rearward of the strong emission in the convective line. Note the brightness temperatures that are colder than the sea surface TBs in that location. Nice emission signal at 10 GHz too, especially corresponding with the highest low-level reflectivities from KLGX. This was a heavily raining core!

Later in the flight, the ER-2 overflew some developing clouds in support of the AirMSPI instrument. AMPR got some good observations from this period as well.

```
In [76]: data.plot_ampr_channels(scanrange=[5450, 5750])
```

```
********************
plot_ampr_channels():
Available scans = 1 to 6558
Available times = 10:00:22 - 20:17:07
********************
```



Note some of the speckling in the 85 GHz (A) channel over open water, after the aircraft passed over the primary precipitation system. This is likely that noise floor issue that was noticed on an earlier flight. However, it may have also corrupted some of the data near the right fringe of the scene (when the A channel is primarly H polarized). Normally the lowest TBs should be seen there over open water, since at H polarization sea surface TBs decline off nadir. There are also some pockets of missing or anomalously low 85 GHz TBs early in the run. Let's take another look at AMPR geolocated with the NEXRAD data.

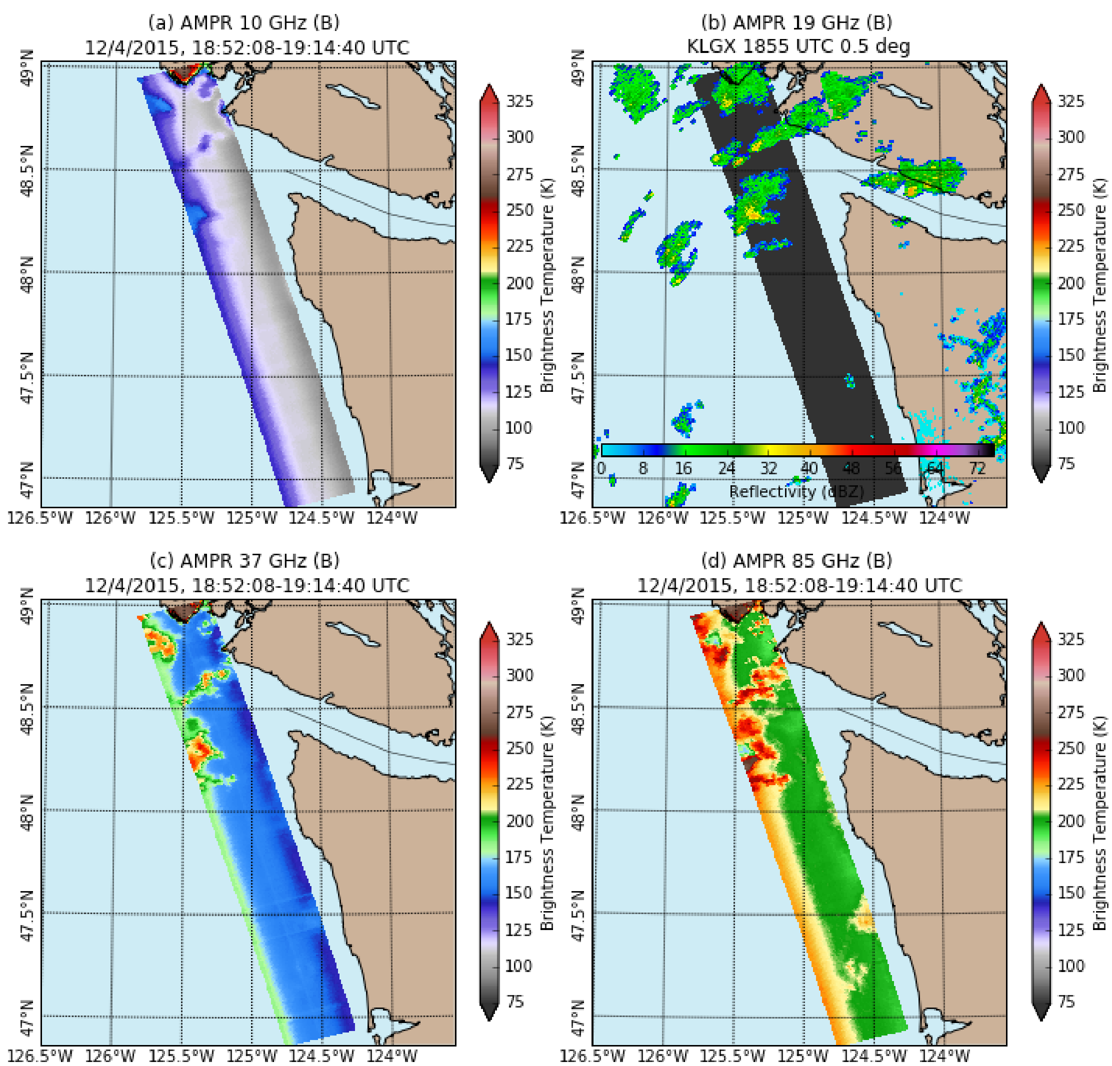
```
In [78]: # print(radlist)
         radar2 = read_nexrad_aws('KLGX20151204_185502_V06.gz')
         zc = deepcopy(radar2.fields['REF']['data'])
         radar2.add_field_like('REF', 'ZC', zc, replace_existing=True)
         retrieve = dualpol.DualPolRetrieval(
             radar2, dz='ZC', dr='ZDR', dp='PHI', rh='RHO', use_temp=False,
             dsd_flag=False, fhc_flag=False, precip_flag=False,
             liquid_ice_flag=False, kdp_window=5, verbose=False,
             ice_flag=False, qc_flag=True, gs=250.0,
             thresh_sdp=20.0, speckle=3,
             thresh_dr=np.array(dualpol.DEFAULT_DR_THRESH)+1.0)
```

```
https://noaa-nexrad-level2.s3.amazonaws.com/2015/12/04/KLGX/KLGX20151204_185502_V06.gz
0.46489715576171875 seconds to run csu_kdp
```

```
No sounding provided
```

```
In [85]: display = data.plot_ampr_track_4panel(
             chan='b', maneuver=False, return_flag=True,
             meridians=0.5, parallels=0.5, resolution='h',
             show_grid=True, scanrange=[5450, 5750], lonrange=[-126.5, -123.5])
         display.ax2.set_title('(b) AMPR 19 GHz (B) \nKLGX 1855 UTC 0.5 deg')

         # Play some tricks to overplot NEXRAD on the same basemap using Py-ART
         dr = pyart.graph.RadarMapDisplay(radar2)
         rdata = dr._get_data('ZC', 1, None, True, None)
         x, y = dr._get_x_y('ZC', 1, True, True)
         _x0, _y0 = display.basemap(dr.loc[1], dr.loc[0])
         pm = display.basemap.pcolormesh(
             _x0 + x * 1000., _y0 + y * 1000.,
             rdata, vmin=0, vmax=75, cmap='pyart_NWSRef', ax=display.ax2)
         cax = display.fig.add_axes([0.52, 0.58, 0.35, 0.01])
         plt.colorbar(pm, label='Reflectivity (dBZ)', cax=cax, orientation='horizontal')
```

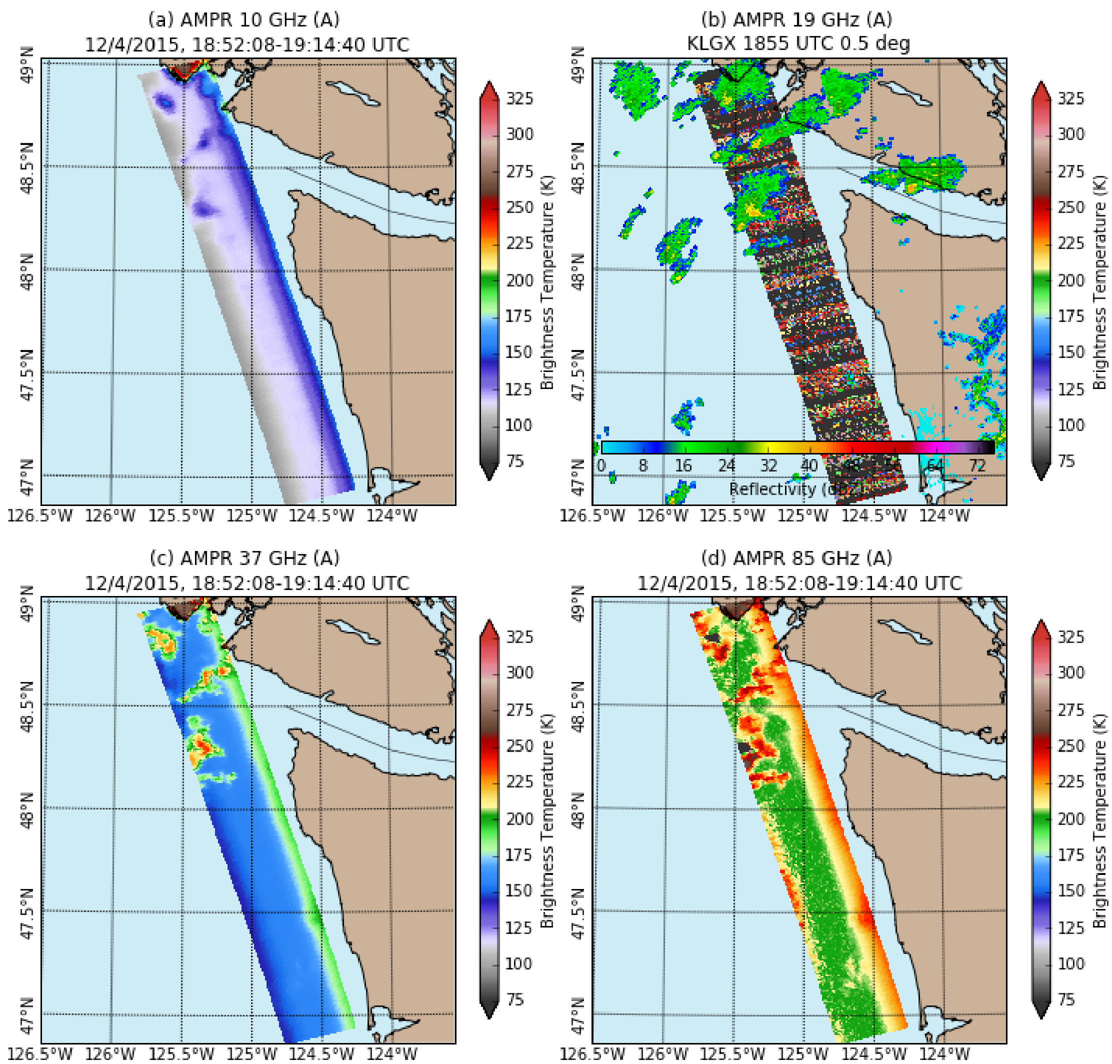
```
Out[85]: <matplotlib.colorbar.Colorbar at 0x150a3b2e8>
```



That was for the B channels. And now the corresponding look for the A channels:

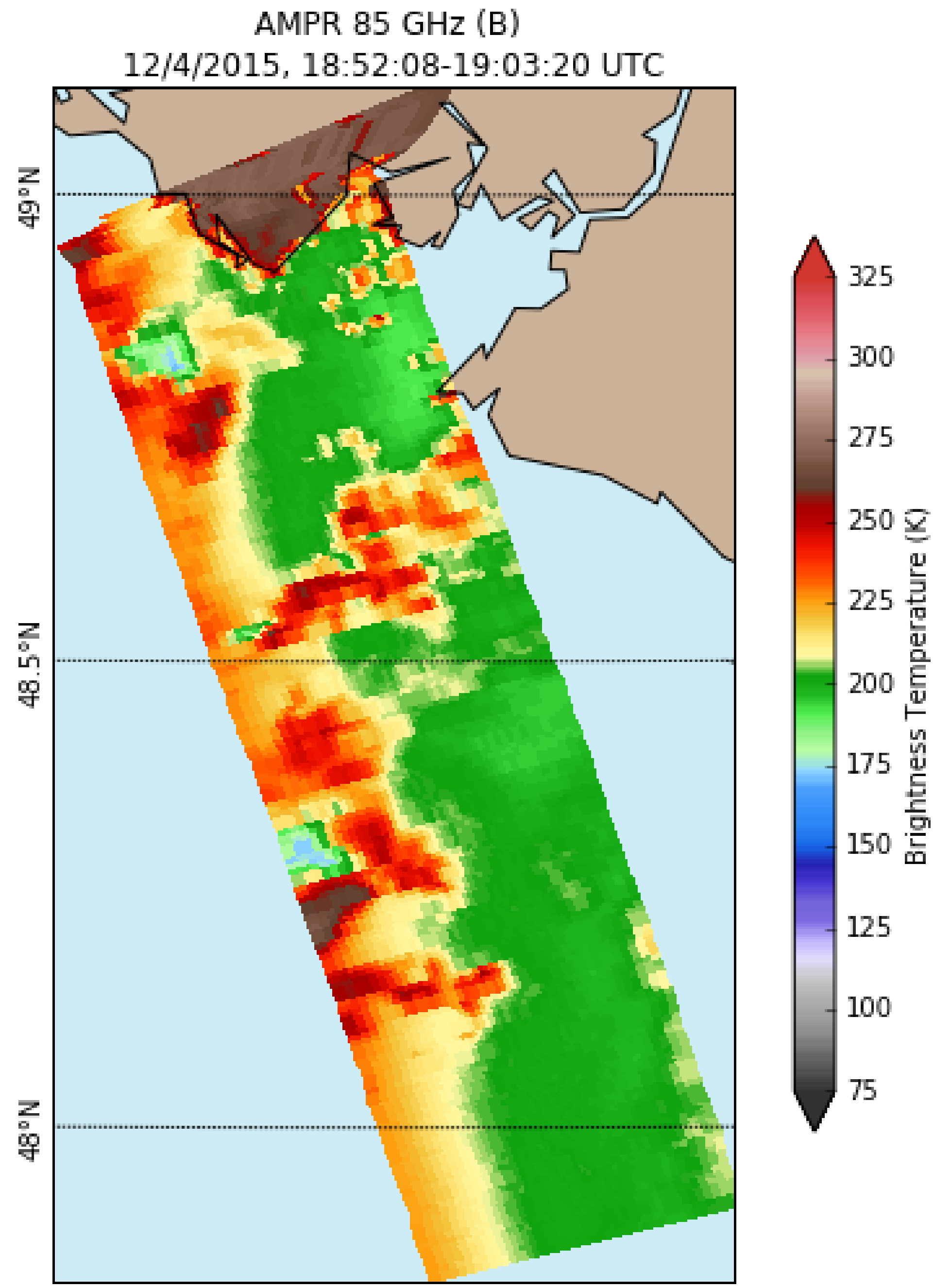
In [86]:
```
display = data.plot_ampr_track_4panel(
    chan='a', maneuver=False, return_flag=True,
    meridians=0.5, parallels=0.5, resolution='h',
    show_grid=True, scanrange=[5450, 5750], lonrange=[-126.5, -123.5])
display.ax2.set_title('(b) AMPR 19 GHz (A) \nKLGX 1855 UTC 0.5 deg')

# Play some tricks to overplot NEXRAD on the same basemap using Py-ART
dr = pyart.graph.RadarMapDisplay(radar2)
rdata = dr._get_data('ZC', 1, None, True, None)
x, y = dr._get_x_y('ZC', 1, True, True)
_x0, _y0 = display.basemap(dr.loc[1], dr.loc[0])
pm = display.basemap.pcolormesh(
    _x0 + x * 1000., _y0 + y * 1000.,
    rdata, vmin=0, vmax=75, cmap='pyart_NWSRef', ax=display.ax2)
cax = display.fig.add_axes([0.52, 0.58, 0.35, 0.01])
plt.colorbar(pm, label='Reflectivity (dBZ)', cax=cax, orientation='horizontal')
```

Out[86]:
```
<matplotlib.colorbar.Colorbar at 0x15390cd68>
```

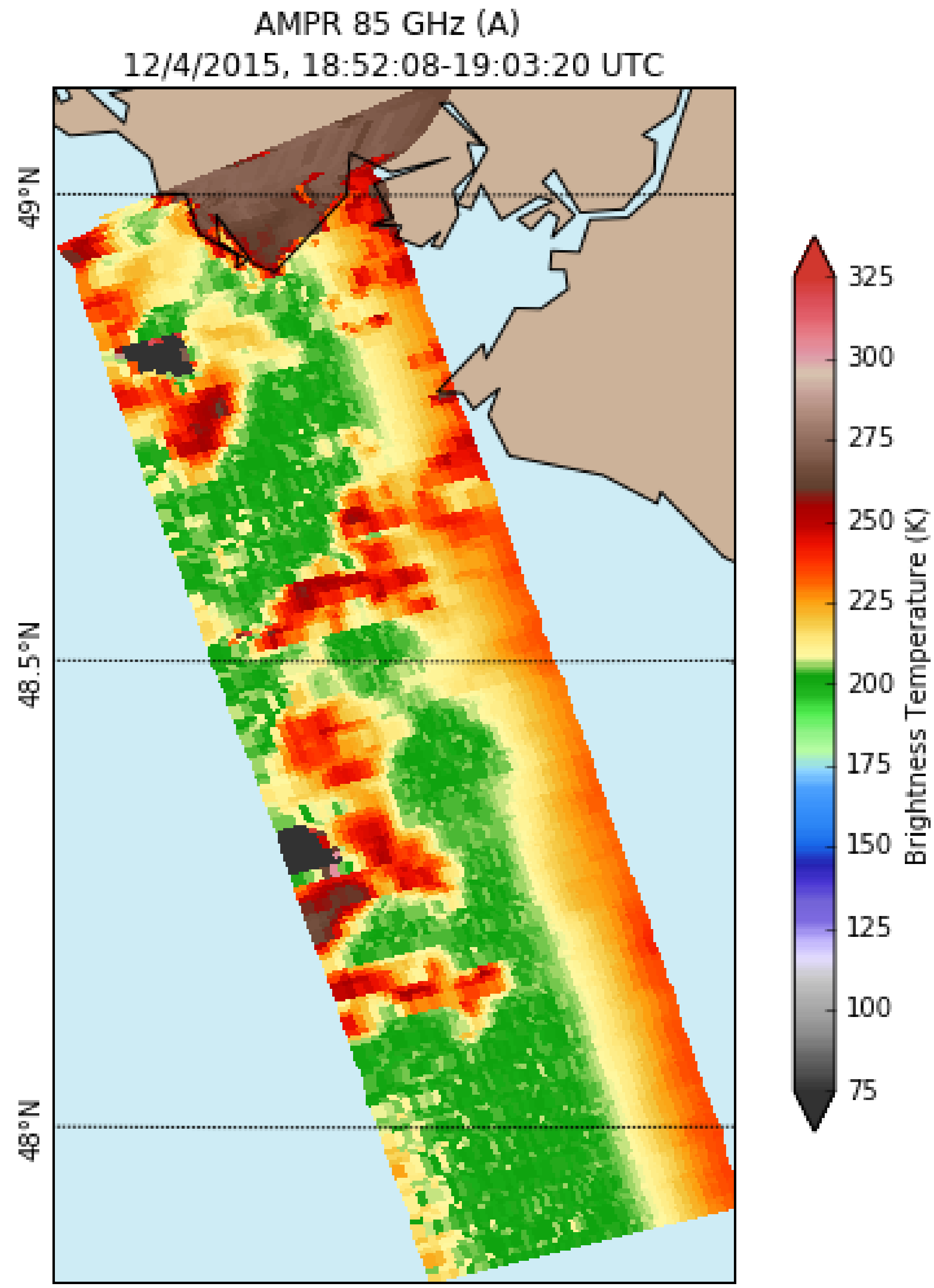


As we can see, very little echo near the west edge of the AMPR swath later in the north-to-south run, suggesting the increased 85 GHz (A) TBs there are due to noise. However, it's possible that in the regions of missing/low 85-GHz TBs near the northern precipitation systems, there may be ice scattering, which shows up at 85 GHz (B) but not A due to the increased noise floor at this time. Let's zoom in.

```
In [92]: data.plot_ampr_track('85b', scanrange=[5450, 5600], resolution='i', parallels=0.5)
```




```
In [93]: data.plot_ampr_track('85a', scanrange=[5450, 5600], resolution='i', parallels=0.5)
```



It's tough to say - we need to look at the data quality in this case in more detail. For now, during the first half of the flight 10, 37, and 85 GHz all look fine. During the latter half of the flight, 85 GHz (A) is questionable in certain regions, especially over open water and if there was the potential for ice scattering. We continue to work on the 19 GHz issues, and hope for at least a diagnosis by Sunday. The 85 GHz issue is difficult to diagnose since it is so intermittent (this is the first time in several flights that it has cropped up). However, we'll take a look at that channel during the down day too.

```
In [ ]:
```