Applanix Code Summary

1. After completing the processing using PosPAC, run Read\_vnav2.m. Then use Newdatamerge to put 1 Hz data in the netcdf file.
2. Matlab Code

* Read\_vnvav2.m:
  + This code reads in the output files from the POSPac processing, and converts them into ASCII files. It runs as a function, with the desired inputs being the file that you would like to read, and the file that you would like created. Both should be input as strings. This code is where the calculation for true heading is done (ie, platform heading – wander angle). The tricky part about this code is the conversion between Applanix time and seconds after midnight UTC. First a correction for the difference between UTC and GPS time of 16 seconds is applied. Then the Applanix time (seconds after midnight UTC Sunday) is converted into seconds after midnight UTC. This involves the manual input into line 92 of the number of the day of the week (eg, Tuesday is day number 3).
* Applanix\_varadd
  + This is the original data mere code that is very difficult to work with. The start index that corresponds to the Applanix start time must be entered manually. Also, if the Applanix continued to collect data after NIMBUS was turned off for the flight. This code is also very, very slow to run in terms of computation time – I believe because of putting the NetCDF file into redefine repeatedly which is what I addressed in “Newdatamerge.m”. The premise of this code is to copy a variable already in the data system, then rename it and reassign it the corresponding values for the appropriate Applanix data. This must be can separately after varadd\_run. If the code does not execute properly it is best to clear all the variables so the NetCDF file really closes- otherwise strange errors and definitions arise.
* Varadd\_run
  + This code runs both Read\_vnav2.m and Applanixfrequencyconvert.m from one call. However, it is important to edit the file names in the call to read\_vnav2.m, and the date for the time conversion in the actual Read\_vnav2.m code.
* Applanixfrequencyconvert
  + This code reads in the text file generated from Read\_vnav2.m and returns a matrix that has taken the input data from 10 Hz to 1 Hz using a linear interpolation. This code is designed to be run as part of Varadd\_run and as a precursor to Applanix\_varadd.
* Newdatamerge
  + This code uses a different method of creating the new variables without the NetCDF file being in redefine mode. This makes the process dramatically faster. It still requires some fine tuning to adjust the start time properly.
  + However, this code only defines the variables it does not put the data into the appropriate variables.
  + To add the data to the variables, the command is:
    - ncwrite('DEEPWAVErf26.nc','LAT\_APP',onehertztest(1838:22058,2));
    - where 1838:22058 indicate the starting and ending indices of the valid data, and 2 represents the variable column position.
    - This must be done for every variable, typically I do this from the command line.
* Notes on running any of this code:
  + If it stops during the execution of it, delete all of the variables before trying again.

1. R code

* New\_app\_merge.R is adaptation of Al Cooper’s gust pod processer. I backed out the relevant parts of the code and had great hope for this code as it already works properly when adding 25 Hz data. However, I still had some array size matching issues. This may have been due to the fact that I was accidently using 200 Hz data instead of 10 Hz data. However, even with that error fixed Al expressed concerns about how the code would function with 10 Hz data as opposed to 25 Hz data. I have not been yet been able to attempt this code with actual 10 Hz data, or to pursue Al’s suggestion of interpolating the data up to 25 Hz from 10 Hz.

Where the difficulty arises is in the merging of the sample rate data. Due to the way I copy the definition of a variable, and then put data into the copy my array sizes do no match. I have not yet been able to discover how to do this manually (although I have some theories in R that have also not yet worked). The command I use to extract the variable size (netcdf.inqDim) fails when using data collected at a higher frequency than 1 Hz. Since this is a function that is defined in the Matlab netcdf package, I wonder if it is a limitation of the language more than the data (or myself). It is possible that this could be done in IDL, but I am unconvinced that the same problems do not exist there.