These notes explain some of the steps in OTRECakrd.Rnw and may be of interest to anyone duplicating or extending the work.

The underlying document generating OTRECakrd.Rnw and OTRECakrd.pdf is a LyX file, OTRECakrd.lyx, which was exported to produce those files. LyX makes it easy to generate the LaTeX commands that produced the text, but it is hard to use for development and debugging, so repeatedly exporting to the .Rnw format and executing that in RStudio is my normal working process.

1. The "initialization" code chunk: I make extensive use of Ranadu, and in connection with that some packages including ggplot2 are loaded but aren't used in this particular study. Some other packages are just loaded by convention but aren't used here. I do use:

   (a) library(tidyverse): This loads in particular dplyr::filter and dplyr::select that I use often when constructing plots.

   (b) library(magrittr): This enables pipes via the '%>%' command, used frequently here. I favor this construction because it makes clear the specific steps leading to the plot. This usage is discussed with a specific example below.

2. "The Problem with AKRD" section:

   The following example, from the chunk 'AKRDplot', illustrates how plots are generated with the "pipe" function:

   ```
   blankNA(Data, Data$TASX < 105, c('AKRD', 'AOAREF')) %>%
     select(Time, AKRD, AOAREF)  %>%
     selectTime(160000, 190000) %>%
     plotWAC(legend.position='top')
   ```

   The first line generates a new data.frame where the variables AKRD and AOAREF are set missing where the restriction (TASX < 105) is satisfied. The resulting data.frame is piped to the next line, where it becomes the first argument to the "select" function (from the package dplyr, here included by the "library(tidyverse)" call earlier. "select" generates a data.frame containing only the listed variables. That data.frame is then piped to "selectTime()", a Ranadu function, where it becomes the first argument. That function generates a new data.frame containing only the specified time interval. That data.frame is then piped to "Ranadu::plotWAC()" where it becomes the first argument.

3. The "Revised Approach" section:

   (a) It would probably be preferable to process using the KalmanFilter processor, but the corrections are so minor in these flights that I have used "CorrectPitch()" instead.

   (b) An earlier description of the complementary-filter approach to representing angle-of-attack, and the justification, can be found at this URL:
   https://drive.google.com/open?id=0B1kIUH45ca5AaFg4Sk9UWDY5OFE.

(c) The discussion includes the possibility of introducing time shifts, but cursory exploration of these shifts indicated that they were not necessary. This was done with 1-Hz data files; it might be worth considering again with 25-Hz data.

(d) The Ranadu function "getProjectData()" was used to load available data from OTREC. This function searches for all available netCDF files, including test flights if the ".Test" argument is set TRUE, and constructs a composite data.frame containing all flights. It adds a variable "RF" to the data.frame that is the flight number, with numbers 51, 52, 53 for test flights 1, 2, and 3, and that can be used to select individual flights. Once research flights are present also in the OTREC directory, they will also be loaded, so a change will be needed then if only test flights are desired.

(e) The time-in-air (TIA) variable was calculated from the time past the first time when the variable WOW_A (weight on wheels) was zero. This was done for each flight separately. Note the "as.double()" function in the definition, needed to avoid problems later where the result otherwise is treated as a POSIXct or diff.time variable.

(f) A same loop over flights was used to correct the pitch. This needs to be done for individual flights to avoid the carryover effects that otherwise could occure between flights.

(g) There is a switch, "needData", that when TRUE causes the flights to be read for the project. This is used to save repeated reading and construction of the data.frame that is used, which is saved in SaveRData ("OTRECakrd.Rdata") and, for "needData" FALSE, is loaded from disk. It is safe to leave needData TRUE.

(h) Versions of this program have been applied to SOCRATES and WECAN, and there is some inactive code left from those uses. There are some comments where changes are needed for another project. Without those changes, this is not suited to application to another project. In particular, full-project reviews were used to identify areas where there were problems with some measurements, and there is code to exclude those regions. A section used to construct the review plots for this purpose is suppressed in the code via an "if (FALSE) { }" section.

(i) For OTREC, a separate data.frame was constructed that included only pitch and speed-run maneuvers. These maneuvers were included:

flight	type	start	end	altitude (ft)	comment
tf01	speed run	185418	185642	11,500	one-way only, not increasing and decreasing
"	"	201201	201349	20,000	" "
tf02	"	153918	154810	16,000	
"	pitch	154828	160002	16,000	
tf03	speed run	182956	183423	35,500	
"	pitch	185327	190540	26,500	
"	speed run	190100	190646	26,500	overlapping with previous pitch
"	pitch	192947	194158	7,500	

These were used in separate fits, as discussed in the text, to isolate the times where the measurements are particularly suited to this use.

4. The "Results" section:

   (a) The fits are the result of "lm()" calls where the variables used are in composite data.frames spanning, for OTREC, the three test flights. In addition, the variables are restricted to times when the best vertical wind measurements are expected: airspeed above 105 (to eliminate periods near takeoff and landing), absolute value of roll smaller than 2°, and no flags set suggesting that the measurements should be excluded (recorded by missing-value WIC). In the case of OTREC, only a section of tf01 is excluded in that way to eliminate the section of the flight with missed approaches. In addition, flight periods with rates of climb or descent exceeding 5 m/s were excluded because such periods often produce outlier vertical wind (for reasons I don't understand but may arise from airflow changes when power settings are particularly low or high).

   (b) The coefficients for the fits are extracted from the result of "lm()" using "coef()", and those coefficients are used in subsequent processing to find the empirical representation of AKRD (here called AKY). They are also placed into the text using the "knitr" command "\Sexpr{}" so that there will be no errors transcribing and no obsolete results left in the text. More information about these fits can be obtained using the "summary()" command applied to the fit result; e.g., "summary(fs)" or "summary(fssr)" for, respectively, the full-data-set fit (with the exclusions in the previous item) or the fit to the maneuvers data-set.

   (c) The calculation of the new value of vertical wind, here WIY, uses this code:

```
Data$AKY <- cffx * Data$QRF + cfs[1] + cfs[2] * Data$QRS +
  cfs[3] * Data$QCFS + cfs[4] * Data$TIA
DataW <- Data
DataW$ATTACK <- Data$AKY
DataW <- Ranadu::WindProcessor (DataW)
Data$WIY <- DataW$WIN
```

The first line uses the fit coefficients and fit variables to calculate the new angle of attack. The data.frame is then copied to DataW for use calculating the wind. The variable "ATTACK" in that data.frame is replaced by the new angle-of-attack, but all other variables used to calculate wind are unchanged. The Ranadu function "WindProcessor ()" then is called; this uses the usual wind-dependent variables to duplicate the standard calculation of the vector wind, using appropriate reference-frame transformations, rotation rates, etc., as in the normal "gust" calculation. The resulting wind is placed into new variables, one of which (WIN) is the new updraft, so that is copied into the Data data.frame.

(d) The calculations are then repeated with the new vertical wind, using Equation (4) from the main document. This can be iterated, but in the version used here that is only done once. The difference between (4) and the exact relationship represented by "WindProcessor()" will cause a gradual drift if this is iterated further.

(e) A final section of code was used to plot the difference between AKY produced in this way and AOAREF produced by (4). This was reviewed to check for outlier values that might bias the fit, but it was decided not to exclude any other regions. These plots are now suppressed (via "EVAL=FALSE " in the header of the "BAD" code chunk).

– End of Memo –