18 November 2015

TO: AMTD-AAC file, manuscript for AMT
FROM: William Cooper
SUBJECT: workflow for generating this manuscript

1 Purpose

This section describes the workflow for the code and text in "AMTD-AAC.Rnw" that generates the manuscript for the proposed journal article “Algorithms for Correcting Measurements of Attitude Angles.” AMTD-AAC.Rnw contains both text (in LaTeX format) and R processing script for the analysis in this Technical Note. The description of workflow provided here includes the process of collecting the observations and processing them to data files, the data archives used, the steps required to generate the plots and other results including the instances where manual intervention is required to identify appropriate subsets of the data, the relevant R code and LaTeX documents, and all the steps leading to the generation of the text in the manuscript. "AMTD-AAC.Rnw" is the authoritative reference, but this overview and these diagrams will help explain the workflow at a general level and so should substitute for reading the R and LaTeX code in most cases. The intent is to describe the workflow in sufficient detail to support replication of the analysis and figures presented in this manuscript, to facilitate changes based on new data or new analysis approaches, and to make it practical to apply the proposed algorithms to data collected by research aircraft used in atmospheric studies.

This document is proposed to be a journal article, with two purposes: (1) to describe a method for correcting attitude angles measured by an INS, and (2) to call attention to the NCAR Technical Note “Uncertainty in Measurements of Wind from the NSF/NCAR Gulfstream Aircraft” Cooper et al. (2016). An ancillary goal is to provide extensive documentation that can lead to full reproducibility of this research using archived and preserved information, and this description of the workflow is part of that documentation.

2 Acquisition of the primary data

The measurements used in this report were collected using the NSF/NCAR GV research aircraft during the DEEPWAVE project of 2014. The onboard data-acquisition program ’aeros’ recorded the data in digital format, and those data files were then processed by the program ’nimbus’ to produce an archive in NetCDF format. The software management group of NCAR/EOL maintains a version-controlled archive of these programs, so if they are of interest they can be obtained by contacting the data-management group of EOL (at this or this address). The data files available from NCAR/EOL can be found at links on this URL. The details of the processing algorithms are documented here: Processing Algorithms. These procedures as they pertain to the measurement of wind are also documented in Cooper et al. (2016). The resulting data files contain measurements in scientific units and brief descriptions of each measurement, included as netCDF attributes for the files and for each variable..

3 The AMTD-AAC.Rnw file

The structure of the .Rnw file is that it is basically LaTeX, generated for simplicity using LyX and exported to .Rnw format and then processed in RStudio (RStudio (2009)). The .lyx file will run equivalently and produce a PDF-format version of the manuscript, but it proved easier to tailor the file to Copernicus requirements within RStudio. Within that file or within the .lyx file there are "chunks" of R code (R Core Team (2013)), delineated by a header having this format:

```
<<title, var=setting, ...>>=
...R code
@
```

These generate plots and other results of analysis that are incorporated into the manuscript using 'knitr' (Xie (2013, 2014)) In RStudio, the chunks appear as gray sections in the file when it is edited. Where tasks involve execution of R code, the chunk containing the code is referenced in the discussion below. Any results from the processing can be incorporated into the LaTeX text via "\Sexpr{}" calls embedded in the LaTeX portion of the file.

The file and the manuscript are organized into two main sections where the two correction algorithms, one for pitch and roll and the other for heading, are developed. A third section discusses the results and presents some checks and estimates of the magnitude of the typical corrections, and a final section summarizes the results and also discusses some aspects of how reproducibility is documented. The latter topic is discussed in much more detail in this workflow description. Each of those sections is discussed in some detail in this memo.

4 Required R packages including Ranadu

The R code used for analysis reported in this paper relies heavily on a package of routines for R called "Ranadu." This is a set of R scripts for working with the NetCDF archive data files produced by NCAR/EOL/RAF, and it includes some convenience routines for generating plots and performing other data-analysis tasks with the archived data. The Ranadu package is available at this GitHub address and is also included in the supplementary material accompanying this manuscript. To run the R code referenced here, that package should be included in the R installation. The AMTD-AAC.Rnw routine requires that package and also some others referenced in the file, including "knitr", "ggplot2", "grid", "ggthemes", "zoo" and "signal". In addition, Ranadu requires "ncdf4", "tcltk", "stats", and "reshape2". Some parts of "Ranadu" reference additional packages as needed, but they are not used in AMTD-AAC.Rnw so do not need to be available for this routine to run.

The data processing for this manuscript involved revising some parts of the Ranadu package, as listed below. The relative timing among measurements is particularly important in this study, so

some development of utilities to aid in studies of timing was useful. The netCDF files sometimes have mixed rates, e.g., 5_Hz for GPS-provided measurements and 25 Hz for some others including interpolation to 25 Hz from 13 Hz for IRS-provided measurements. In "Ranadu", this required some modifications to getNetCDF(), which generates an R data.frame by reading the netCDF file, and the addition of a new function ShiftInTime() to shift particular variables forward or backward in time, as described below.

4.1 getNetCDF () modifications

To handle data files with mixed-rate variables (e.g., 5-Hz GPS but 25-Hz IRS and others at 1 Hz), this script for reading the netCDF data files incorporates code to produce a single-rate R data.frame. For this purpose, there is a function "IntFilter()" in that script that interpolates and filters variables. The RAF data archives follow the convention that each sample is tagged with a time that is the *beginning* of the time interval, not the center. For example, for 50-Hz samples averaged to one second, the sample represents the average of 50 samples beginning at the specified time and so should be interpreted as a sample average at 0.5 s past that specified time. The "IntFilter()" routine observes this convention by interpolating a low-rate sample to the specified higher rate and then shifting the resulting time series forward in time by half the original time interval, with duplication of the first measurement to fill the initial 1/2-period slots and also replication of the trailing 1/2-period slots that are not filled by the linear interpolation routine. Tests verified that this provided an appropriate representation of the measurement as interpolated to the higher rate and shifted forward to match other higher-rate measurements.

4.2 ShiftInTime ()

To shift time series variables forward or backward, this new function was added to the "Ranadu" package. The function interpolates the supplied time series to a higher rate (125 Hz), uses the R function "stats::approx()" for linear interpolation, shifts the interpolated series an appropriate number of bins forward or backward (duplicating or truncating the end values as necessary), optionally applies a smoothing filter, and then returns an appropriate subset to represent the time series at the original sample rate.

4.3 XformLA ()

The algorithm for heading correction requires that the accelerations measured by an inertial navigation system (IRS) be transformed from the aircraft or a-frame reference coordinates to a local-level Earth-relative frame or l-frame in which the {x, y, z} coordinates are respectively east, north, and up. This transformation was coded as a new "Ranadu" function. This function takes as input a data.frame containing the attitude angles measured by the INS is variables named PITCH, ROLL, and THDG (true heading), all in units of degrees. It transforms an a-frame vector to an l-frame

vector and returns the result. It would be preferable to do this in vectorized form, but such a representation hasn't yet been found, so the function uses a loop. As noted in the routine, the approach using R 'apply()' functions was slower than the loop, but that is because the specific approach tried here is likely unnecessarily convoluted. This is an area of needed improvement, but the loop as coded is practical and adequate for this study.

The transformation coded in this routine was obtained in standard ways using rotations ϕ about the roll axis, then θ about the pitch axis, and then ψ about the heading axis. The text lists sources for this transformation, but they differ from the matrix used here in that this transformation starts from what is called here the a-frame or aircraft reference frame with x forward, y starboard, and z downward and transforms to the local frame or l-frame with x east, y north, and z upward. The transformation matrix $\mathbf{R}_a^l$ is obtained as follows, where the first matrix changes the axis definitions:

$$\mathbf{R}_a^l = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}$$

$$R_a^l = \begin{bmatrix} \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\psi\sin\phi - \sin\psi\sin\theta\cos\phi \\ \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & -\cos\psi\sin\theta\cos\phi - \sin\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & -\cos\theta\cos\phi \end{bmatrix} \quad (1)$$

4.4 CorrectPitch () and CorrectHeading ()

Two scripts were also included in the "Ranadu" package to calculate the corrections to pitch, roll, and heading as described in the manuscript. They are discussed below in the respective sections for those two corrections.

5 The pitch-correction algorithm

Manuscript Sect. 2.1: The basis for the correction

After an introduction, Sect. 2 of the manuscript develops and applies the algorithm for correcting pitch and roll. Section 2.1 is an outline of basic theory, as also developed in more detail in Cooper et al. (2016). This is based on standard analysis such as is available in Noureldin et al. (2013). One subtlety may be worth mention here. The Schuler oscillation involves coupling among the errors in Earth-relative velocity, pitch or roll angle, and latitude or longitude. These are strongly coupled through the INS in order to maintain bounds on the growth of the errors. The choice developed here, represented in Equations (2) and (3), is to relate the pitch or roll errors to the acceleration errors. An acceleration error also leads to a position error and so to errors in latitude and longitude, and it might appear that those errors should also enter the error in pitch, but the IRS includes the measured acceleration and so the measured position in its integration and so already compensates

the measured pitch for this error. Other representations of the Schuler coupling among these errors could be used, but they are equivalent so it is most straightforward to represent the error in pitch or roll in terms of the errors in acceleration.

Manuscript Sect._2.2: An example

In general, it is necessary to transform the errors in pitch obtained from (2) and (3) to the reference frame of the aircraft, but there is one flight of the NSF/NCAR GV that was almost directly southbound for the full flight so for that flight the angle transformations are not necessary except for the need to reverse the signs of the pitch and roll errors to account for the flight direction. For the purpose of illustrating how the pitch error is coupled to the ground-speed errors through the Earth-relative acceleration, measurements from that flight (from 1 June 2014, flying from Kona, Hawaii to Pago-Pago) were used. Figure 1, top, shows the ground-speed errors estimated by the difference between IRS-provided measurements and those from a GPS receiver. In the data file for this flight, these are variables {VEW, VNS} from the INS and {GGVEW, GGVNS} from the GPS receiver. The netCDF file was called DEEPWAVEff02.nc, and the plot was generated in R chunk "v-errors-straight-leg" using "plotWAC()" and "lineWAC()", convenience functions in Ranadu that call the standard R graphics routines with particular options. For the bottom plot, the derivatives of the error components shown in the top plot were calculated using signal::sgolayfilt(), which has the option of calculating the derivative while using a Savitzky-Golay filter. A length of 1013 points was chosen because that spans about 1/5 of a Schuler oscillation, and a third-order filter was chosen. There are about 15 s of missing values in the time series for VEW and VNS, and the filter call fails if the time series includes missing values, so before calling the filter the routine zoo::na.approx() was used to interpolate across the gap. The steps in the correction algorithm will be discussed again in the subsequent section where the pitch-correction function is described.

Manuscript Sect. 2.3: Transformation of the attitude angles

This section develops the transformation needed to obtain the pitch and roll errors in the aircraft-body frame or a-frame from the l-frame errors provided by (2) and (3). The transformation is a simple rotation by the negative of the heading angle. The transformation is developed by using a unit vector representing the platform orientation and considering how the components of that unit vector transform under rotation. The the definitions of the pitch and roll angles are used in Eqs. (6) to obtain the pitch and roll errors in the a-frame. The signs can be checked as follows: For $\psi = \pi/2$, $\delta\theta^{(a)} = -\delta\phi^{(l)}$ and $\delta\phi^{(a)} = \delta\theta^{(l)}$; for $\psi = -\pi/2$, $\delta\theta^{(a)} = \delta\phi^{(l)}$ and $\delta\phi^{(a)} = -\delta\theta^{(l)}$; and for $\psi = \pi$ both components change sign. This conforms to expected behavior from visualization of the misalignment vector under these conditions.

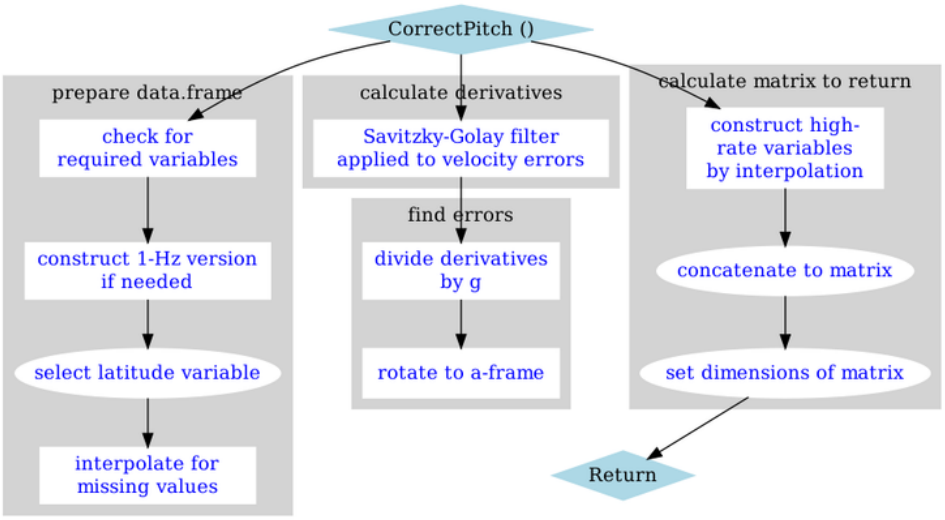

Figure 1: Workflow diagram for the Ranadu function CorrectPitch(), which returns estimates of errors in measurements of pitch and roll. The returned values, with units of degrees, should be subtracted from the measurements to obtain corrected values.

The Ranadu::CorrectPitch() function:

A function to calculate the errors in pitch and roll was developed and incorporated into the Ranadu package on the basis of the above steps, to facilitate correction of the measurements. That function is described in detail here, and the authoritative reference is the code contained in "PitchCorrection.R" at the Ranadu GitHub site referenced earlier. The function takes as input a data.frame containing the variables VNS, VEW, GGVNS, GGVEW, LAT or LATC, GGALT, PITCH, ROLL, and THDG, which are respectively the north and east component of the ground speed from the IRS, the corresponding components from the GPS, the latitude, the geometric altitude, and the pitch, roll, and heading angles, in units of $\mathrm{m\,s^{-1}}$ or m or $^\circ$ as appropriate. A workflow diagram for the function is shown in Fig. 1. The steps in the algorithm are these:

1. Prepare the data.frame:
   (a) Test to see if all required variables are present in the input data.frame; return 0 otherwise.
   (b) Determine the data rate of the input data.frame from the time difference between samples. If it is higher than 1 Hz, extract a working 1-Hz data frame to use for the calculations. This greatly improves performance for high-rate files.
   (c) Allow for either LAT or LATC to provide the latitude required for the gravity routine.
   (d) Interpolate to fill in missing values for VNS, VEW, GGVNS, GGVEW, using the zoo::na.approx() function. If there are remaining missing-value elements in the time

series (which can occur if the gaps exceed 1000 s), set the values to zero for those regions to avoid failure at the next step. The variable MaxGap in the function determines the maximum gap for interpolation; it can be changed only by changing the code.

2. Calculate the derivatives:

   Determine the derivatives of the error terms (VNS-GGVNS) and (VEW-GGVEW) using third-order Savitzky-Golay filters, specifically the R function signal::sgolayfilt(). Use a span of 1013 points (changeable via an argument to CorrectPitch()) and calculate the first derivative by supplying an appropriate argument (m=1) to the filter routine.

3. Find the errors in pitch and roll:

   (a) Calculate the acceleration of gravity as a function of latitude and altitude using the Ranadu function Gravity() and divide the derivatives (with appropriate sign) by this value to get the *l*-frame errors in pitch and roll.

   (b) Transform the result to the *a*-frame using manuscript equation (6).

4. Construct the matrix to return, which has rows matching the input data.frame and two columns, the first representing the pitch error and the second the roll error.

   (a) If the data rate of the original data.frame is higher than 1 Hz, interpolate the working 1-Hz data.frame to the original rate, again using zoo::na.approx() for linear interpolation.

   (b) Concatenate the two vectors representing the pitch and roll errors to one vector.

   (c) Convert to an appropriate matrix with dimensions [DL,2] where DL is the length of the original data.frame. This conversion uses "dim()" with the concatenated vector.

5. Return a two-dimensional array with the two components being the *a*-frame errors in pitch and roll, so that the result can be used to obtain corrected values of the pitch and roll via

```
PITCHC <- D$PITCH - CorrectPitch (D)[,1]
ROLLC  <- D$ROLL - CorrectPitch (D)[,2]
```

All steps in this function are vectorized, to operate on the entire time series in vectorized function calls. Even for high-rate data.frames, the processing involves little delay: On a modest linux desktop system, processing a 7-h 25-Hz file takes about 2 s.

Manuscript Sect. 2.5: Discussion

This section is mostly opinion and assessment and did not require calculations except for order-of-magnitude estimates explained in the text.

Manuscript Sect. 2.6: Some details regarding smoothing

The comments in this section have already been discussed in regard to the "CorrectPitch()" function. This section makes some of those aspects of the algorithm more specific in the manuscript.

6 The heading-correction algorithm

Manuscript Sect. 3.1: The basis for the correction

This section is theoretical, with equations developed as described, so additional description of the workflow is not useful. The details will be provided later with the description of the heading-correction algorithm.

Manuscript Sect.3.2: The transformation from *a*-frame to *l*-frame

The key to implementation of the proposed algorithm is transformation of the measured accelerations to a reference frame where they can be compared to the observed accelerations via GPS. The transformation itself was described in detail in Sect. 4.3, and two references for this transformation are listed in the manuscript. As checks, the transformations given in those two references (Eq. 2.6 in Lenschow and Spyers-Duran (1989) and Eq. 2-82 in Noureldin et al. (2013)) were further transformed to give directly the *a*-frame-to-*l*-frame transformation, with signs of rotation angles as required, and each led to (1) as quoted in Sect. 4.3.

It seemed useful to check that this transformation gives *l*-frame accelerations in reasonable agreement with those measured by GPS, so several checks were performed. One, reported in the manuscript, used a circle maneuver flown during DEEPWAVE research flight 15. The circle maneuvers are discussed in considerable detail in Sect. 7.1 of Cooper et al. (2016), where plots of the flight track in a coordinate frame drifting with the wind show that the Lagrangian tracks are close approximations to circles. They were flown under control of the flight management system of the aircraft, which was able to maintain constant roll angle to a tolerance of a fraction of a degree. Two circles were flown in left turns, then two additional circles were flown in right turns. The ground-speed components measured by GPS were differentiated to obtain reference measurements of the corresponding acceleration components, again using the approach of replacing missing values by interpolation and then using fitted Savitzgy-Golay polynomials to find the derivatives. In this case, because the conditions changed too rapidly for the long extents used in the pitch-correction algorithm, the length of the polynomials was reduced to 21 1-Hz measurements for the horizontal components and 7 1-Hz measurements for the vertical component of acceleration. The results are shown in Fig. 3 of the manuscript, as the dotted cyan and orange symbols. This plot was generated in R chunks "reinitialization" (where the data were read and shifted in time using the "ShiftInTime()" function described in Sect. 4.2), "sg-poly-smoothing" (where the interpolation for missing values was included and where derivatives of the GPS-measured ground-speed components were

calculated using fitted Savigzky-Golay polynomials), and "get-b-vector-and-transform" (where the accelerations were transformed to the l-frame). To match the GPS-derived accelerations, the transformed accelerations were also filtered using Savitzgy-Golay polynomials with the same characteristics (but returning the filtered values instead of the derivatives). The result of this transformation to the l-frame is shown by the blue and green lines in Fig. 3. They match well, confirming the approximate validity of the transformation.

Several other checks were performed also but have not been described in the manuscript. One important case was to consider a pitch maneuver in which the pitch was varied cyclically with about a 20-s period, with wings level but with airspeed and altitude varying. The maneuver flown on flight 15 during the period of about 4:25:00–4:28:30 UTC showed strong oscillations in the vertical acceleration, and again the body accelerations transformed to the l-frame matched will the accelerations deduced from the changes in vertical speed measured by the GPS receiver. There is commented code in AMTD-AAC.Rnw, at the bottom of chunk "get-b-vector-and-transform", that would generate this plot if uncommented.

Some details of the procedure used to generate this plot will be discussed more fully in the next subsection, where the heading-correction algorithm is discussed. One additional aspect needing further discussion here is need to account for some rotations relative to an inertial frame. The same sources used for the coordinate transformation (Lenschow and Spyers-Duran (1989) and Noureldin et al. (2013)) provide equations for this correction. Here, Eqs. 5.55–5.57 from Noureldin et al. (2013) have been used. When these corrections were included, the heading errors deduced as in the following subsection changed by typically about 0.01° while the magnitude of the heading correction was about 0.08° (for NSF/NCAR GV flight at about 220 $\mathrm{m\,s^{-1}}$ and at the latitude of New Zealand), so it appears useful to include the correction. Some approximations appeared acceptable, though, including using a mean radius of the Earth, neglecting the height of the aircraft in comparison to that radius, and neglecting the small correction that arises from the INS being displaced a small distance from the center of gravity of the aircraft and therefore experiencing false accelerations during rotation of the aircraft.

Manuscript Sect. 3.3: A proposed correction algorithm for heading

Here, the Ranadu function "CorrectHeading()" will be discussed to illustrate how the requirements and sequence of steps can be implemented. This function has a structure similar to that of CorrectPitch(), as shown in Fig. 2, with four sets of tasks related to initial handling of the data, calculation of the required accelerations, calculation of the heading correction, and formation of the output that the function returns. Before calling this function, time shifts should be applied to the measurements as needed to obtain coincidence between values reported by the INS and the GPS receiver, and if desired corrections to pitch and roll should be made and saved in the data.frame as variables PITCHC and ROLLC. These will be used in place of PITCH and ROLL if they are present. The task description for the function call is as follows:

1. Prepare the data.frame: This section is the same as for the pitch/roll correction, except the required variables are different (including the body accelerations but not needing the INS-

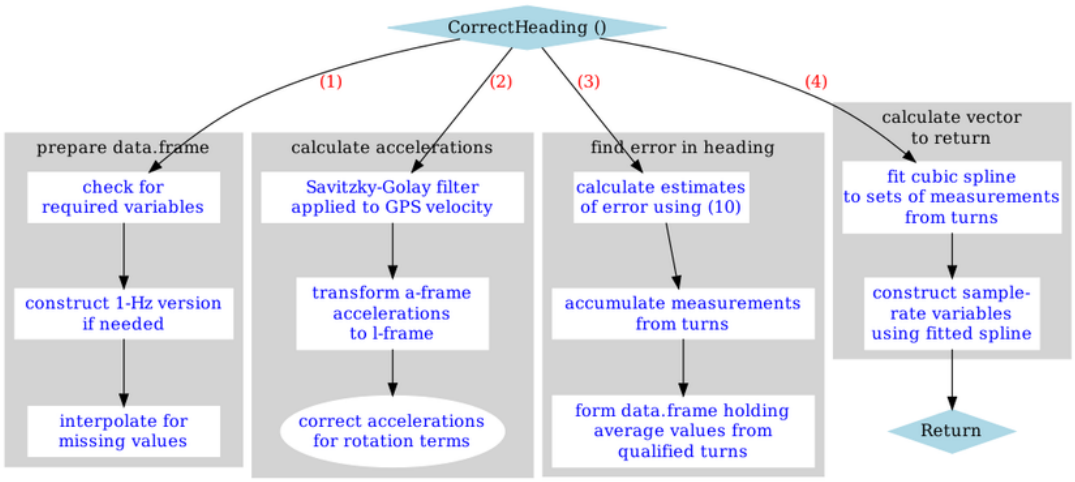

Figure 2: Workflow diagram for the Ranadu function CorrectHeading(), which returns estimates of the error in heading. The returned values, with units of degrees, should be subtracted from the measurements to obtain corrected values. Timing adjustments if needed should be made before calling this function.

provided ground-speed components) and the interpolation for missing values applies to the revised set of variables (BLATA, BLONGA, BNORMA, GGVNS, GGVEW, GGALT, LAT, PITCH, ROLL, THDG). A 1-Hz data.frame is generated if the input data.frame is higher rate, to speed the calculations and because it is not expected that the heading error will fluctuate at high frequency (or at least cannot be corrected for such fluctuations).

2. Calculate the accelerations:

   (a) Determine the derivatives of the GPS-measured ground-speed components GGVNS and GGVEW using third-order Savitzky-Golay filters, specifically the R function signal::sgolayfilt(). In this case, a much shorter span of 21 1-Hz points is used (changeable via an argument to CorrectHeading()) because the accelerations change in turns much faster than do the velocity errors used in the pitch-correction function. Again, the first derivative was found by supplying an appropriate argument (m=1) to the filter routine.

   (b) Calculate the acceleration of gravity as a function of latitude and altitude using the Ranadu function Gravity() and add this to the normal component of acceleration (BNORMA) provided by the INS, because this is subtracted before the normal acceleration is reported. Then form a matrix of the a-frame accelerations that has three columns and a number of rows matching the length of the working 1-Hz data.frame.

   (c) Transform the resulting matrix to the a-frame using the Ranadu transformation routine "XformLA ()", described in Sect. 4.3 above, which returns a matrix containing the corresponding l-frame accelerations. After transformation, again add the acceleration

of gravity to the third component (with positive sign because the direction of the z-axis is now upward).

(d) Adjust the resulting accelerations for the rotation terms given by (11)–(13).

(e) Apply the same-span and same-order Savitzky-Golay filter to the transformed l-frame accelerations from the INS, to match the filtering applied to the GPS measurements.

3. Find the error in heading:

(a) Use (10) from the manuscript to estimate the heading error at each point. Qualify these values by requiring that the magnitude of the horizontal acceleration be greater than $1\,\mathrm{m\,s^{-2}}$ for the estimate to be considered valid; replace others with the missing-value flag NA.

(b) Search for turns, separately assembling data from left turns (ROLL<-10°) and from right turns (ROLL>10°). End each segment when there are 5 min of flight without meeting either turn criterion. If the accumulated measurements have at least 25 1-Hz measurements in each turn direction, consider this to be valid data for estimating the heading correction and save the mean values and standard deviations for each turn direction, along with the mean time, in accumulation arrays. If the accumulated measurements do not meet this test, discard. In either case, reinitialize to look for the next segment. The internal parameters TGAP and NSL respectively set the 5-min and 25-sample tests in the function; they can only be changed at present by changing the code.

(c) Test the number of qualifying turns:

i. If less than three, return a default value (-0.08). This can be provided as an optional argument (.default) to the function.

ii. Otherwise, form the accumulated characteristics from the turns, including the mean error and the standard deviation, into a data.frame named EH.

4. Calculate the vector of error estimates to return from the function:

(a) If the default value has not already been returned, fit a cubic spline to the accumulated measurements from qualifying turns. The fit is determined using the R function "smooth.spline ()" with a number of degrees of freedom equal to one less than twice the number of qualifying turns (because each qualifying turn provides two data points, one for each turn direction), with weights for each point equal to the inverse of the variance of the contributing measurements of heading error, and with a parameter "spar" set to 0.7. The latter determines the degree of smoothing, and this choice was the result of exploration with various parameters. This appears to provide a reasonable representation of the variation in the measurements while maintaining a slowly varying result over the course of research flights. Exploration with a variety of research flights led to this choice, but there is no further backup for this choice except that it seemed to produce

reasonable results. The R documentation for the basic graphics package includes more discussion of the fit routine and the effect of the "spar" parameter, and it also provides literature references.

(b) Construct the vector to return using the fitted cubic spline to find the value of the estimated heading error for each time in the input data.frame. The heading can then be corrected via
THDGC <- D$THDG - CorrectHeading (D)

The routine also includes optional code, enabled by setting "GeneratePlot <- TRUE" at the beginning of the routine, to create a plot showing the results. This code is normally inactive but can be copied or activated when it is useful to see the results in plotted form.

Manuscript Sect. 3.4: Examples

This manuscript section presents the result of applying the preceding algorithm to two research flights.

For the first, a plot of the individual "qualified" measurements (primarily, with magnitude of the horizontal acceleration greater than 1 m s^{-2}), each representing a measurement over 1 s, is included as Fig. 4. This is generated in R chunk "error-components" using standard R graphics via the Ranadu::plotWAC() function. To generate the included smoothed line, a variable was introduced that initially had value equal to the mean heading error for the full set of measurements shown, then these values were replaced by the qualified measurements of heading error where they existed, and the result was filtered using a be-directional Butterworth low-pass filter with a cutoff period of 5 min.

A plot of the result of applying the heading correction algorithm discussed above was also included, as Fig. 5. This was generated using the R "ggplot2" graphics package using the data.frame "HC" generated as described in the discussion of the heading-correction algorithm. This plot is generated in R chunk "plot-heading-correction-rf15" in the AMTD-AAC.Rnw program file. The plot superimposes results from calls to "geom_errorbar()", "geom_point()" and "geom_line", the latter showing results of the spline fit.

The error bars need clarification here. These are estimates of the standard deviation of the mean that applies to each point determined for a qualifying turn. However, the points are not independent samples because smoothing has been applied to the accelerations before determining the estimated errors and because there is some inherent distance over which the estimates are correlated. A Savitzky-Golay smoothing polynomial gives an effective reduction in variance of about $9/(4M)$ where M is the span of the polynomial, so for $M = 21$ the effective reduction is about a factor of 0.11 in variance, as would result from averaging about 9 points. Therefore, it is assumed that the effect of averaging points is to reduce the variance by $(N/9)^{-1}$ rather than N^{-1}, so the standard deviation of the mean σ_m is at least $\sigma/\sqrt{N/9}$ rather than $\sigma/\sqrt{N}$ where σ is the sample standard deviation. These standard deviations in the mean values still appear unrealistically small, and smoothed splines typically intersect only about 40% of the error bars. To obtain error bars for

which about 63% of the error bars are intersected by reasonable spline fits, it would be necessary to increase the standard deviations by an additional factor of about 2. Because the error bars otherwise look so small, the plots show two-standard-deviation error bars for the errors in the mean values, as noted in the captions. Weight factors as used to fit the cubic spline are $1/\sigma_m^2$ so they remain proportional to N and are not affected by this assumed correlation, but this choice affects the size of the plotted error bars in Figs. 5 and 6.

The plot for the second example, Fig. 6, was constructed in the same way. The plot is generated in R chunk "spline-plot", where the weighted mean and weighted standard deviation for the heading correction are also calculated. They are incorporated into the text via "\Sexpr()" calls.

7 Some properties of the corrections (Manuscript Sect. 4)

The pitch correction

The intent of this section is to illustrate typical magnitudes of the deduced corrections and so to support assessment of their importance.

1. Figure 7 shows values of the heading correction for a representative flight. This was generated in R chunk "typical-errors" using the R ggplot2 routine "geom_density()".

2. The reverse-heading legs are discussed in more detail in Cooper et al. (2016), Sect. 6.4.7. The workflow included identifying the reverse-heading times, but that had already been done for inclusion in that technical note, so the same times are included here. More discussion of this selection is included with a workflow statement that accompanies that reference. The plots are generated in R chunk "reverse-course-w-comparison" using the R routine "hist()", part of standard R graphics. The pitch correction was calculated using the "CorrectPitch()" function described earlier, in Sect. 5.

3. Manuscript Sect. 4.1.3 provides some estimates of the uncertainty in the pitch correction. One estimate quoted without justification is the estimate that the uncertainty in the pitch correction arising from the uncertainty in determining the derivatives is smaller than 0.0001°. Some elaboration leading to that estimate is presented here. Modern GPS receivers, especially if augmented by special signals or special processing, produce 1-Hz measurements with uncertainty of around $0.03\,\mathrm{m\,s^{-1}}$(Cooper et al. (2016)). The consistency of the Schuler oscillation, as illustrated by the top panel in Fig. 1, suggests that derivatives in velocity can be determined by averaging over periods of at least 10 min or more, so if the ground-speed measurements from the INS have uncertainty of about $0.03\,\mathrm{m\,s^{-1}}$ (where variance spectra for the 1-Hz measurements begin to show noise), the uncertainty in differences between these two signals might be expected to be $0.03\sqrt{2} \approx 0.04\,\mathrm{m\,s^{-1}}$, and averaged over 10 min or perhaps 60 autocorrelation times the resulting difference could be resolved to $0.04/\sqrt{60} \approx 0.005\,\mathrm{m\,s^{-1}}$. Over intervals separated by 10 min, the derivative then might be

determined to $0.005 \times 1/600 \approx 10^{-5}\,\mathrm{m\,s^{-2}}$, leading to an uncertainty in the pitch correction from manuscript Eqn. (2) of about 10^{-6} or 0.00005°. S-G polynomials reduce noise in an average by a factor of $\sqrt{(3(3m^2-7)/4m(m^2-4)} \simeq 0.05$ for m=1013 or 0.06 for m=601. Then for $0.04 * 0.06 = 0.0024$m/s, 600-s separation gives acceleration uncertainty of about $0.0024/600 = 4e-6$ or pitch uncertainty of 2e-5°. That is the reason that this potential contribution to uncertainty in the correction has been neglected.

The heading correction

This subsection just references earlier results and notes a relevant result from the Technical Note.

Applanix studies

Mention is included here of an attempt to use an INS with incorporated Kalman filter as a reference, even though that did not prove useful. During the DEEPWAVE project, files from an Applanix INS were recorded at 10 Hz, so those can be compared to the measurements from the standard Honeywell IRS without Kalman filtering but adjusted using the procedures developed in this note. Several problems were encountered when attempting to use the measurements from the Applanix, so this line of investigation was abandoned, although it may be worth further investigation, especially because there are post-processing steps available for the Applanix unit that improve the measurements. Without such post-processing, these problems led to difficulties:

1. The Applanix and Honeywell units were evidently installed with mean differences in orientation of about 0.1° in all three attitude angles. Because the heading algorithm applied to the Honeywell measurements indicated a net offset, it was unresolved if the difference arose from installation differences or from IRS differences. The corrected heading from the Honeywell IRS differed from the Applanix heading more than the uncorrected heading from the Honeywell IRS, but it could be the case that this indicates an installation-angle difference. The standard deviation of the differences of the corrected or uncorrected Honeywell heading vs the Applanix heading were essentially the same.

2. The available measurements included body accelerations and components labeled as x, y, and z velocity and x, y, and z body acceleration, but these don't match any of the velocities or accelerations (body or local) determined from the Honeywell system, so it is unclear what these represent. For example, they can't be used to determine the orientation of the Applanix system by comparing ground-speed components to those determined from the GPS because the measurements do not appear to be valid representations of the ground-speed components.

References

Cooper, W. A., Friesen, R. B., Hayman, M., Jensen, J. B., Lenschow, D. H., Romashkin, P. A., Schanot, A. J., Spuler, S. M., Stith, J. L., and Wolff, C.: Characterization of uncertainty in measurements of wind from the NSF/NCAR Gulfstream V research aircraft, proposed NCAR technical note, undergoing review NCAR/TN-XXX+STR, Earth Observing Laboratory, NCAR, Boulder, CO, USA, URL https://drive.google.com/open?id=0B1kIUH45ca5AT2NwVFpqRFNOZ0U, 2016.

Lenschow, D. H. and Spyers-Duran, P.: Measurement techniques: air motion sensing, Tech. rep., National Center for Atmospheric Research, URL https://www.eol.ucar.edu/raf/Bulletins/bulletin23.html, 1989.

Noureldin, A., Karamat, T., and Georgy, J.: Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration, SpringerLink : Bücher, Springer Berlin Heidelberg, doi: 10.1007/978-3-642-30466-8, URL https://books.google.com/books?id=qGbsW6sDr7YC, 2013.

R Core Team: R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, URL http://www.R-project.org, 2013.

RStudio: RStudio: Integrated development environment for R (Version 0.98.879), URL http://www.rstudio.org, 2009.

Xie, Y.: Dynamic Documents with R and knitr, Chapman and Hall/CRC, Boca Raton, Florida, URL http://yihui.name/knitr/, iSBN 978-1482203530, 2013.

Xie, Y.: knitr: A general-purpose package for dynamic report generation in R, URL http://yihui.name/knitr/, r package version 1.6, 2014.

– End of Memo –

Reproducibility information for the manuscript:

PROJECT:	AMTD-AAC
ARCHIVE PACKAGE:	AMTD-AAC.zip
CONTAINS:	attachment list below
PROGRAM:	AMTD-AAC.Rnw
ORIGINAL DATA:	/scr/raf/Prod_Data/DEEPWAVE
SUBSET DATA:	NCAR HPSS; contact NCAR/EOL/RAF data management
WORKFLOW:	WorkflowAttitudeAngleNote.pdf
GIT:	https://github.com/WilliamCooper/AMTD-AAC.git

Attachments: AMTD-AAC.Rnw
AMTD-AAC.pdf
SessionInfo