Earth Observing Laboratory

Memorandum:

23 August 2019

TO: Sensible Heat Flux File
FROM: Al Cooper
SUBJECT: Suggested filter for dynamic heating

Overview

I argued during my RSIG presentation on 19 August (cf. this URL) that we are introducing noise by applying the dynamic-heating correction at frequencies where the probe does not respond. It would be desirable to avoid that contamination of our signal, which we could do by filtering the dynamic-heating correction to match the time response of the sensors. This memo records the results of some exploration of such filtering and incorporates the R code to implement it.

The earlier studies were somewhat ambiguous regarding the time constant for the Rosemount "fast" probe, with results favoring 875 ms but not ruling out about 500 ms. For the HARCO, the evidence was convincing that the time constant was about 2.2 s. To be conservative, I will use 500 ms and 1.0 s, respectively. To match existing data processing, I will assume respective recovery factors of 0.9854 and 0.975 because, for the selected flight leg, those gave the best match to the existing measurements (ATF1 and ATH1) of air temperature.

The filter applied here is a Butterworth 3rd-order filter with cutoff frequency specified to match the assumed time constants. It is a centered filter, implemented using "signal::filtfilt()" in R, and to avoid end effects the dynamic-heating term is filtered from 5-min before to 5-min after the desired analysis period and then the time series is truncated to the desired time (SOCRATES research flight #15, 5:52:00 – 6:20:00. The dynamic-heating term, labeled DH here, was calculated without the recovery factor from TASX$^2/(2C_p)$ where C_p is the specific heat of air at constant pressure.

The problem is illustrated in Fig. 1, where the two contributions to ATH1, the measured recovery temperature RTH1 and the dynamic-heating term DH=TASX$^2/(2C_p)$, are shown separately. At low frequency, below about 0.01 Hz, fluctuations in the air temperature (blue line) arise primarily from fluctuations in the measured recovery temperature (green line) and fluctuations in dynamic heating (black line) make only a minor contribution. Around 0.03 to 0.05 Hz, fluctuations in the measured recovery temperature are significantly greater than those in the calculated air temperature because fluctuations in airspeed make a significant contribution to the measured recovery temperature but data processing removes part of those fluctuations and the resulting calculated air temperature is less variable than either of the contributing measurements. This is just how the measurement and calculation should work. However, a problem arises for frequency above about 0.2 Hz because the measured recovery temperature does not respond properly to fluctuations at those frequencies. The data-processing calculation, however, assumes that it does respond and removes the expected contribution from dynamic heating fluctuations, thereby introducing erroneous variance into the calculated air temperature (ATH1). The result of standard processing is that the spectral variance in dynamic heating is transferred to ATH1, but it is spurious because RTH1 is not responding to these fluctuations.

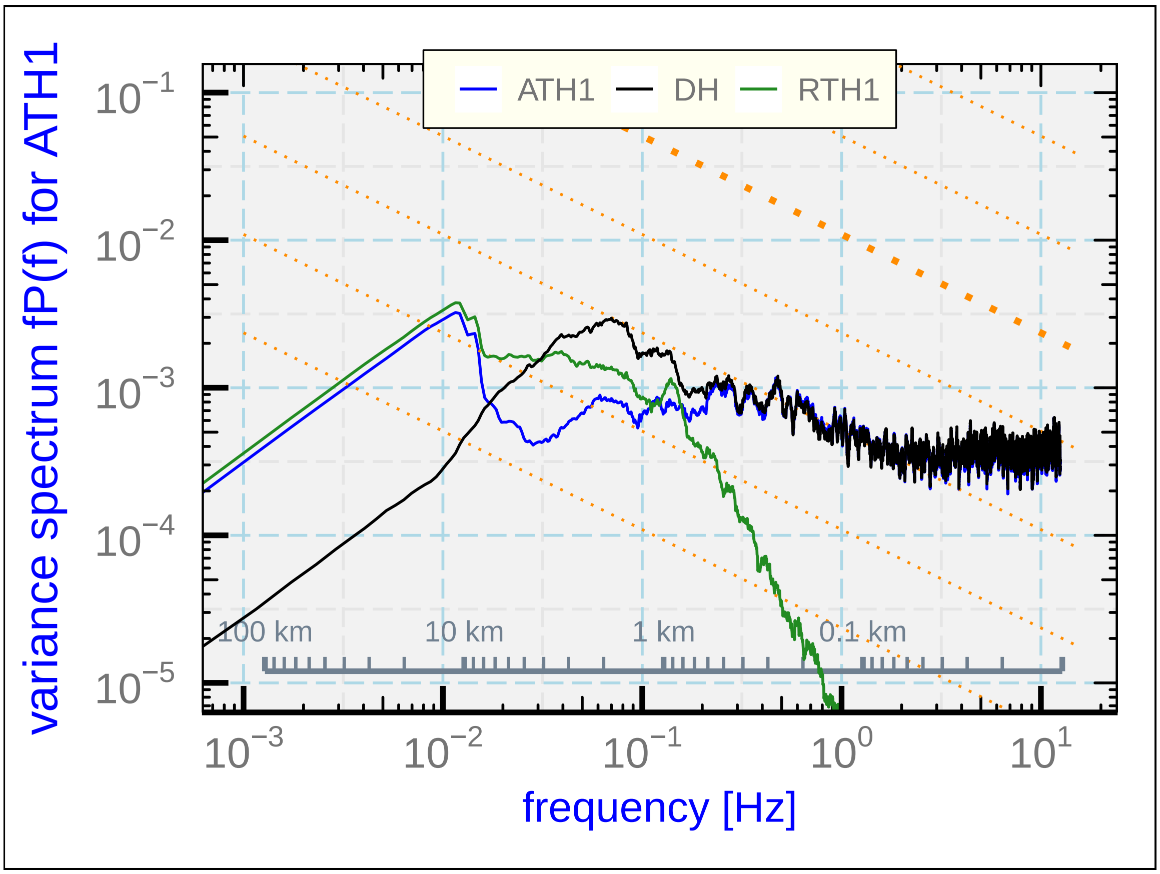

Figure 1: The measured temperature ATH1 from the HARCO sensor on the GV during SOCRATES research flight 15, 5:52:00 – 6:20:00 UTC. Also shown are the two components that combine to produce this measurement, the recovery temperature (RTF1) and the dynamic-heating correction (DH) using the formula given in the text.

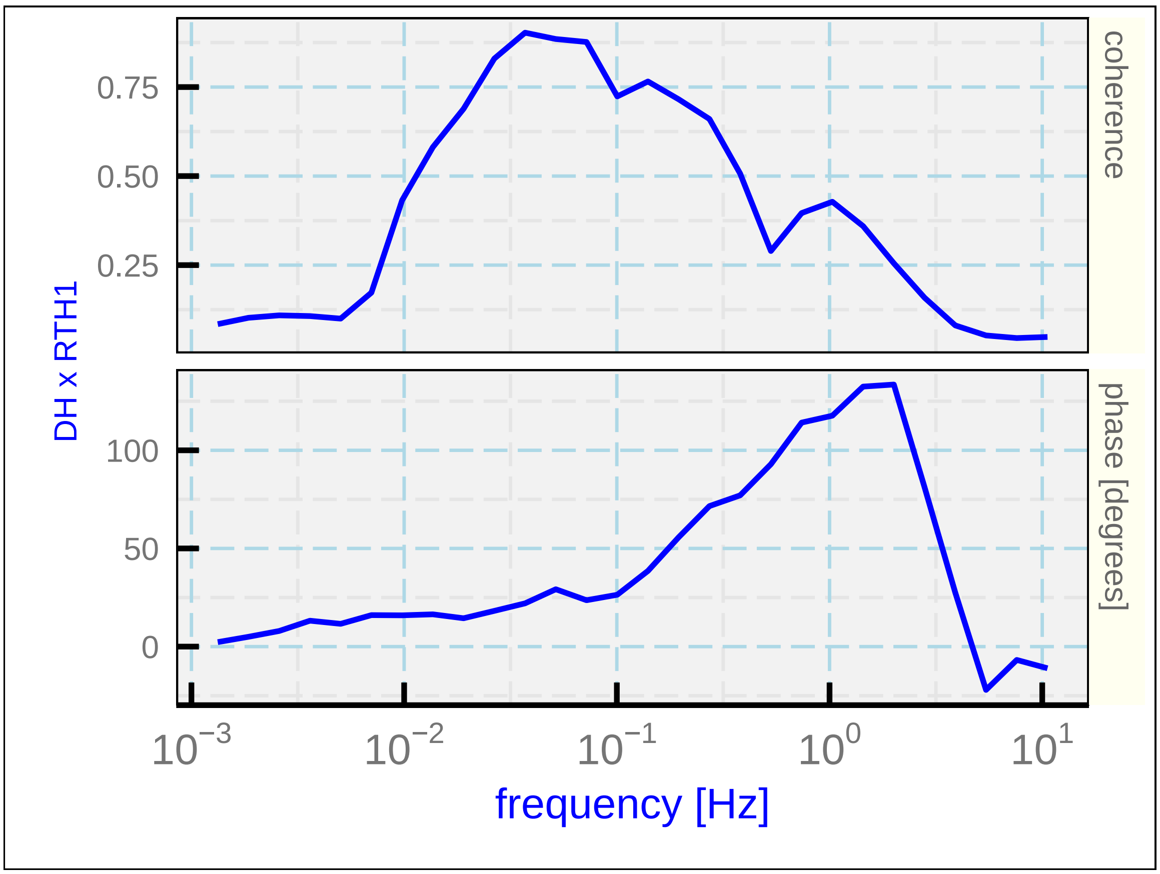

Figure 2: Coherence (squared) and phase for the recovery-temperature and dynamic-heating variables RTH1 and DH, for measurements from the HARCO sensor on the GV during SOCRATES (research flight 15, 5:52:00 - 6:20:00 UTC.

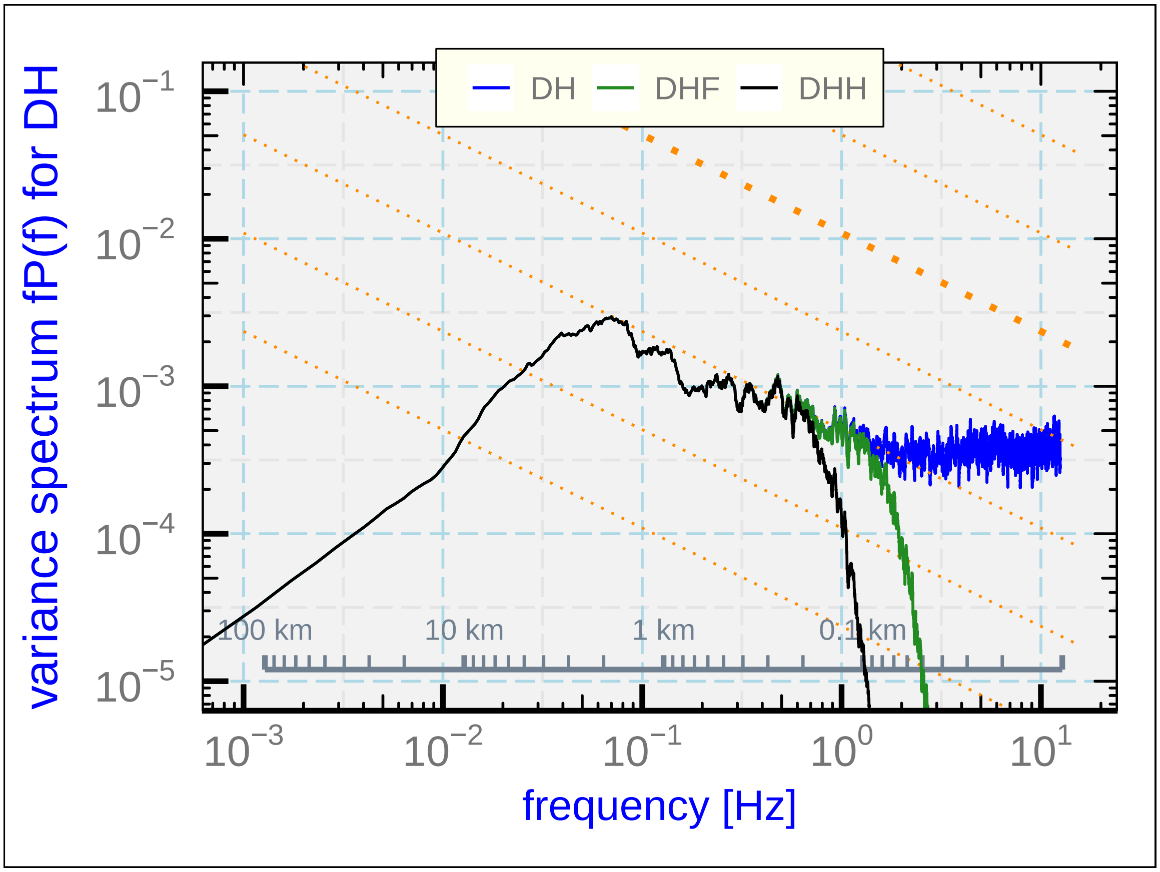

Figure 3: Variance spectra for the dynamic-heating correction (DH) and for that correction filtered using Butterworth 3rd-order low-pass filters with cutoff periods of 0.5 s (DHF) and 1.0 s (DHH).

A particularly clear indication that the dynamic heating is not reflected in the recovery temperature can be seen in a plot of the coherence and phase between those two, shown in Fig. 2. Around 0.05 Hz the signals have coherence aboved 0.8 and are almost in phase, as expected if dynamic-heating fluctuations are driving the measurement RTH1. However, with increasing frequency from this point the coherence drops and the phase shift increases, becoming 90° at about 0.5 Hz, as expected if the time constant for RTH1 is 2 s. At that point the fluctuations in RTH1 and WIC would not detect a real contribution to a measurement of sensible heat flux, and beyond that the contribution would even be of the wrong sign. It therefore seems justified to remove these spurious contributions to variance in ATH1.

Filtering the Dynamic Heating Correction

Figure 3 shows the result of filtering. In this plot, DH is the unfiltered dynamic-heating term, DHF is the result of filtering with an assumed time constant of 0.5 s as is appropriate for ATF1, and DHH is the corresponding result for ATH1 where a 2.0 s time constant is assumed. The result is effective

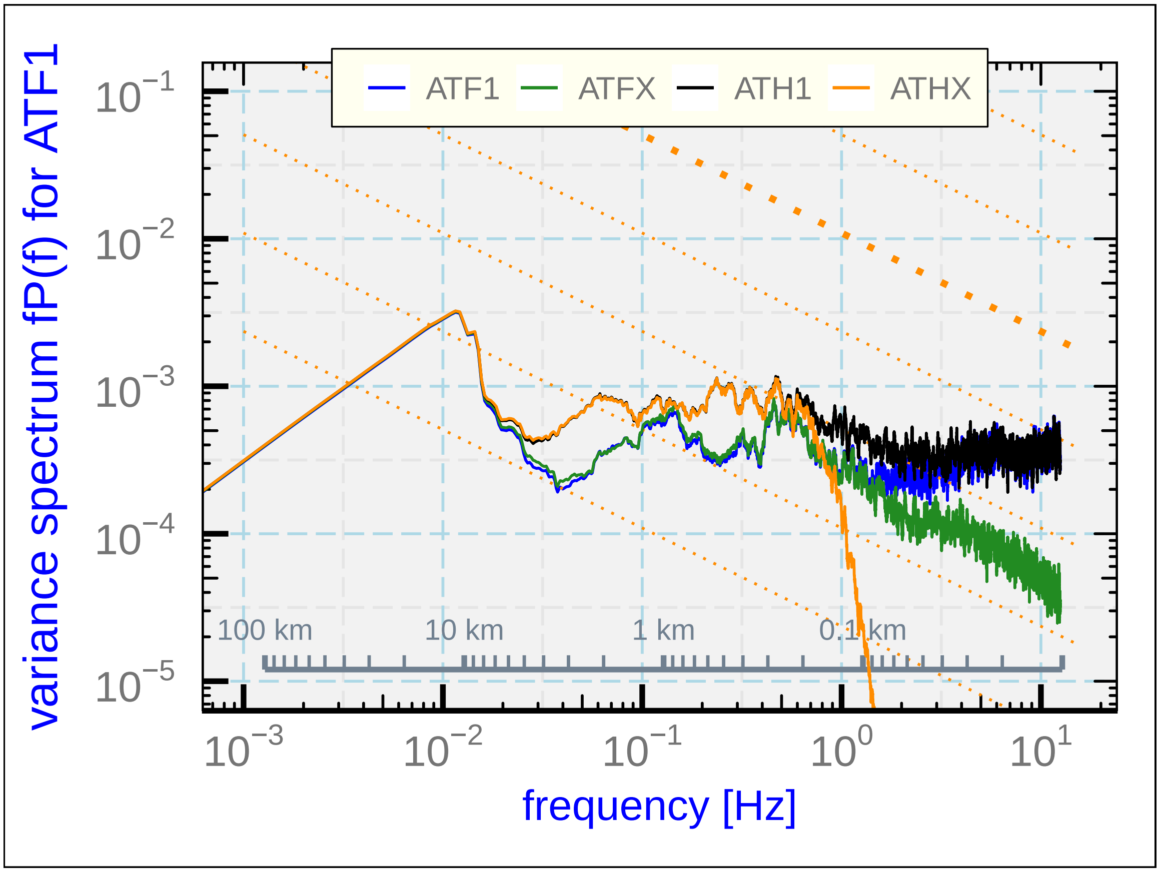

Figure 4: Variance spectra for the air-temperature measurements ATF1 and ATH1, and those modified by replacing the dynamic-heating terms by those shown in the previous plot (respectively, ATFX and ATHX). At low frequency, the black line (ATH1) is obscured by the orange line (ATHX) because those variables are nearly identical below about 5 Hz.

attenuation of the variability for frequencies where it is thought that the probes will not respond to a real signal and where application of the dynamic-heating correction produces erroneous spectral variance.

The variance spectra for the temperature measurements that result from using these filtered dynamic-heating corrections are shown in Fig. 4, as the black and orange lines. As originally processed, the spectra show flattening in the frequency region above 1 Hz and both probes show almost identical variance spectra despite their very different time constants. With filtering, the spectrum for ATFX (modified ATF1) appears reasonable, with the high-frequency variance arising from the recovery temperature, while the spectrum for ATHX (modified ATH1) is affected more significantly and even shows some attenuation at a frequency of 0.1 Hz. It is likely that this approach leaves some significant contribution from dynamic heating that is spurious because of the conservative cutoff frequencies selected, but the modified variance spectra are not so obviously bad as the originals. It is likely that the variance above 1 Hz in ATFX, arising dominantly from variations in RTF1, is mostly spurious also. However, it seems preferable to leave that extra variance in the result because

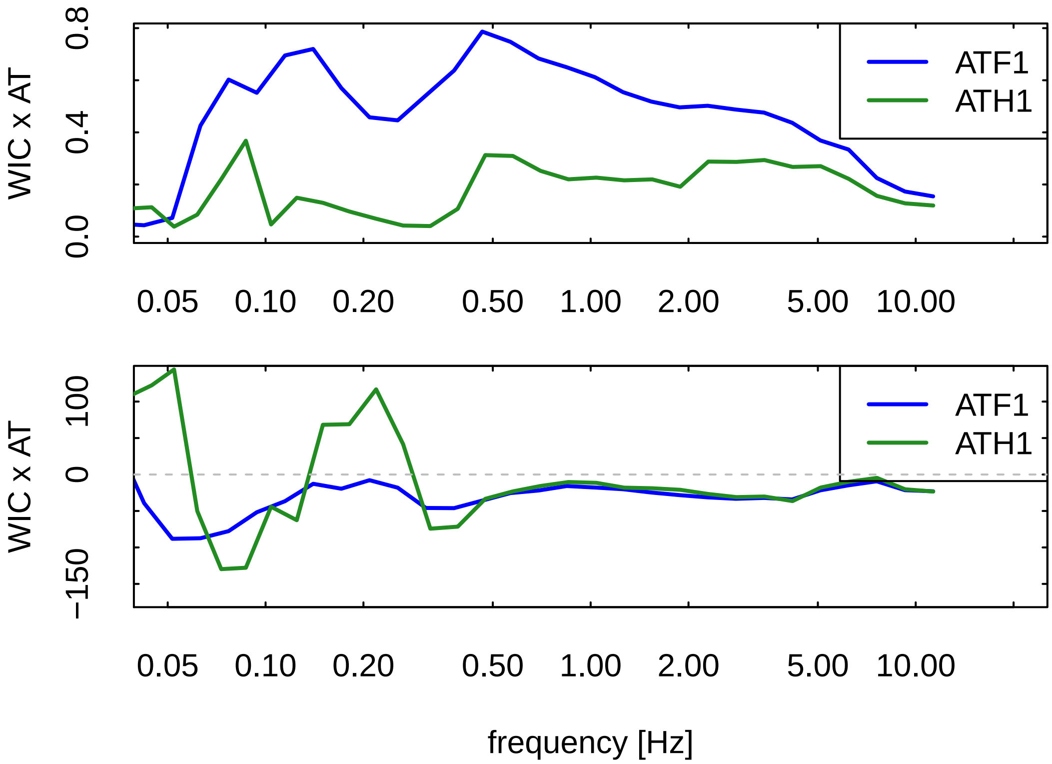

Figure 5: Coherence squared (top) and phase (bottom, degrees) for the same flight segment used in previous plots. These are the results for the unmodified original measurements of temperature.

its source is not clear and further correction could always be applied.

Effect on Measurement of Sensible Heat Flux

It is useful to consider how the modified variables would change a calculation of sensible heat flux. Figure 5 shows the squared coherence and phase calculated from the cross-spectra of vertical wind and temperature for the two temperature measurements ATH1 and ATF1. Coherence is markedly higher for ATF1 than for ATH1 even at low frequency, and both are in phase with vertical wind and significantly non-zero even above about 1 Hz where poor response from both sensors is expected. This will produce a significant contribution to measured sensible heat flux that appears to be spurious.

In contrast, Fig. 6 shows that filtering reduces the high-frequency coherence that is likely spurious while having no significant effect on the low-frequency spectra. Figure 7 shows a sample calculation of the sensible heat flux. It shows that, for the unmodified measurement ATF1, there is a

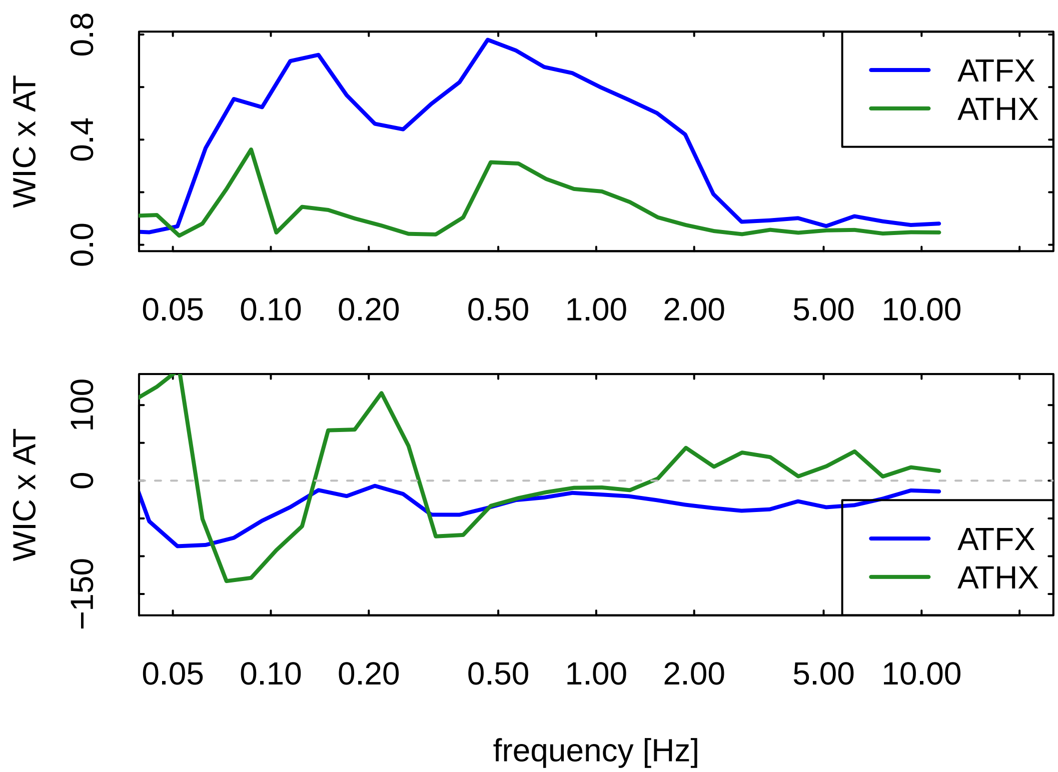

Figure 6: Coherence squared (top) and phase (bottom, degrees) for the same flight segment used in previous plots. These are the results for the measurements of temperature as modified by filtering of the dynamic-heating correction.

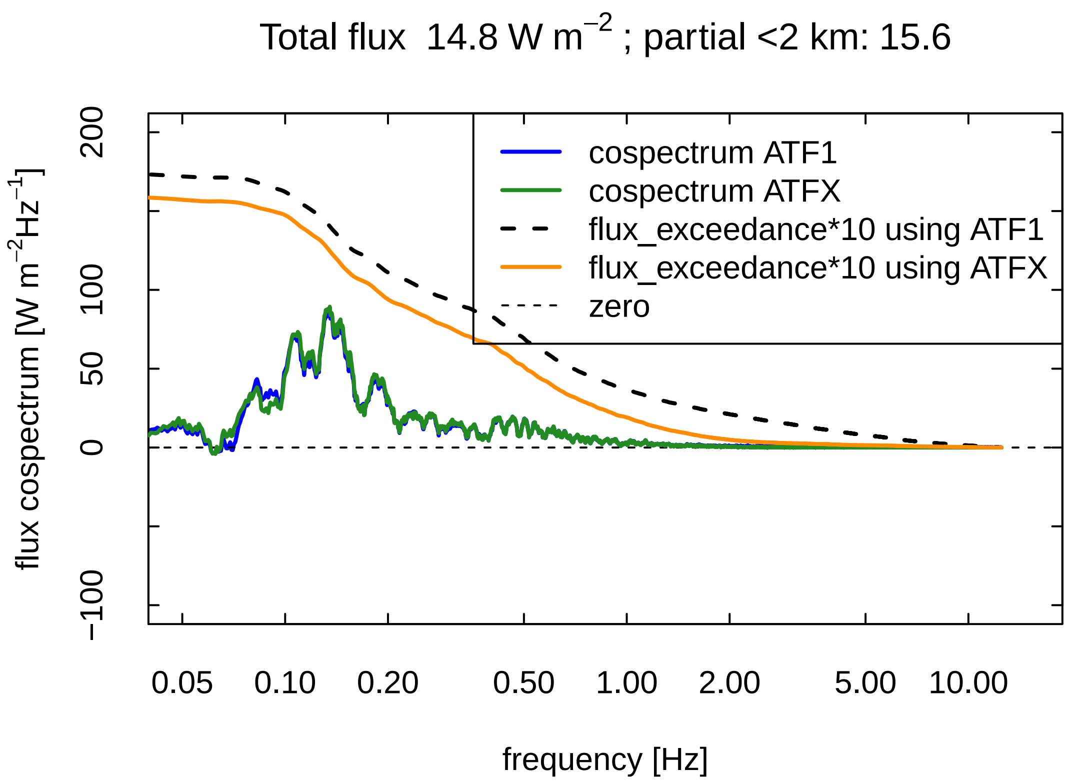

Figure 7: Cospectrum for WIC x ATF1 or ATFX, scaled by specific heat and air density to yield units of W m^{-2} per Hz. Also shown is the flux exceedance for each, scaled by a factor of 10, that is obtained by summing the cospectrum from the highest frequency downward.

non-negligible contribution from frequencies above 2 Hz that appears to be spurious,[1] and that spurious contribution is mostly eliminated when the filtered dynamic-heating correction is used.

The important question left unresolved by this plot is: Is a significant contribution from higher frequencies missed because of the response of the temperature senor? There are significant contributions from the highest frequency range where the sensor should repond, for example from 0.5 to 2 Hz, so it is possible that there is an additional non-negligible contribution to sensible heat flux that is missed because the temperature sensor is too slow.

A Second-Pass Correction Script

As implemented here, the filtering of the dynamic-heating correction is applied forward and backward to avoid introducing a phase shift. A recursive implementation could be used in first-pass processing, but there is danger of introducing an undesirable phase shift so it seems better for now to consider this as a processing step to be applied to netCDF files that will introduce additional variables (here ATFY) from which the spurious dynamic-heating contribution has been removed or attenuated.

Often the recovery temperature RTF1 is not retained in the production high-rate netCDF files. However, it is not necessary for that variable to be present to implement a filtering process because all that is needed is to calculate the original dynamic-heating correction, filter that, and add the difference to ATF1 to obtain ATFY. A complication, however, involves the recovery factor. The attributes of recent GV netCDF files include this attribute for ATF1:

```
ATF1:RecoveryFactor = "0.9959 + 0.0283 log10(mach) +
                       0.0374 (log10(mach))^2 +
                       0.0762 (log10(mach))^3" ;
```

It should then be the case that the recovery factor, retrieved from the ratio

$$\alpha = \frac{2C_p(\mathrm{RTF1} - \mathrm{ATF1})}{\mathrm{TASX}^2}$$

should match that formula and give a unique relationship to MACHX. That appears to be approximately but not exactly matched to the variables, as shown in Fig. 8. The departures from the formula are puzzling, but I'll use that formula anyway. The dynamic-heating correction that has already been applied to RTF1 is:

$$\mathrm{ATF1} = \mathrm{RTF1} - \alpha \mathrm{DH} = \mathrm{RTF1} - \alpha \frac{\mathrm{TASX}^2}{2C_p}$$

[1]This high-frequency contribution arises from small contributions from many different frequencies in the high-frequency range, because as plotted the discrete frequencies occur at an interval of 6×10^{-4} Hz.

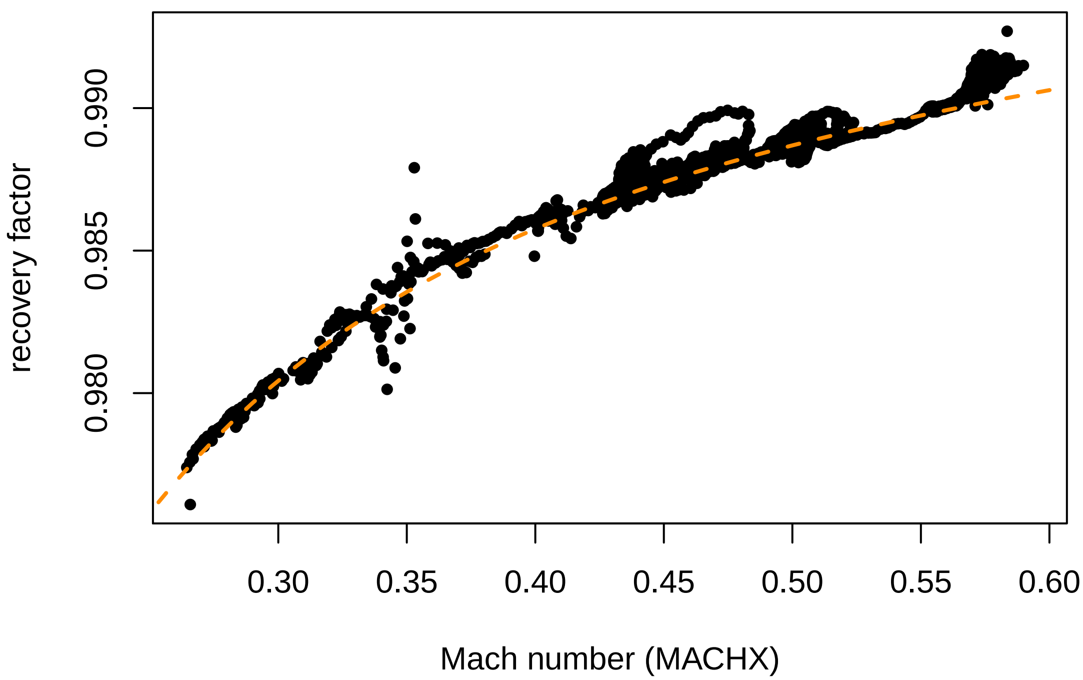

Figure 8: Required recovery factor for agreement between the air temperature and the measurements of recovery temperature and airspeed, for the same data used for previous plots. The dashed orange line is the relationship expected from the formula given in the attributes. The calculation uses a humidity-dependent value of the specific heat for Cp. This plot was calculated using the data from SOCRATES research flight 9.

so if the desired filtered correction is $\alpha \mathrm{DH}^{\dagger}$ then the appropriate correction to apply to ATF1 is

$$\mathrm{ATFY} = \mathrm{ATF1} + \alpha\left(\mathrm{DH} - \mathrm{DH}^{\dagger}\right)$$

An advantage of this formula is that the measured recovery temperature RTF1, often not preserved in processed and especially production netCDF files, is not needed.

The following then is a straightforward segment of R code that can modify an existing netCDF file to add the variables ATFY and ATHY and write them to a new netCDF file with the suffix T (e.g., "SOCRATESrf09T.nc"). It should be used on a high-rate file only. The R code is in the "addATFY" chunk in FilterDynamicHeating.Rnw and could readily be extracted to a script, perhaps with added user interaction or batch transfer for the project and flight, that could be used to process existing high-rate netCDF files.

```
# needed packages
library(Ranadu)
library(zoo)

##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

require(signal)

## Loading required package: signal
##
## Attaching package: 'signal'
## The following object is masked from 'package:dplyr':
##
##     filter
## The following objects are masked from 'package:stats':
##
##     filter, poly

## A function to transfer attributes:
copy_attributes <- function (atv, v, nfile) {
  for (i in 1:length(atv)) {
    aname <- names(atv[i])
    if (grepl ('name', aname)) {next}  # skips long and standard names
    if (grepl ('units', aname)) {next}
```

```
    if (grepl ('Dependencies', aname)) {next}
    if (grepl ('actual_range', aname)) {next}
    if (is.numeric (atv[[i]])) {
      ncatt_put (nfile, v, attname=aname, attval=as.numeric(atv[[i]]))
    } else {
      ncatt_put (nfile, v, attname=aname, attval=as.character (atv[[i]]))
    }
  }
}

## get the old netCDF variables needed to calculate ATFY
VarList <- standardVariables(c('ATF1', 'ATH1'))
## Specify the Project and Flight:
Project <- 'SOCRATES'
Flight <- 'rf15h'  # should be high-rate
fname <- file.path(DataDirectory(), sprintf('%s/%s%s.nc',
                Project, Project, Flight))
D <- getNetCDF (fname, VarList)
Rate <- attr(D, 'Rate')

## Calculate the new variables:
D$Cp <- SpecificHeats (D$EWX / D$PSXC)[, 1]
D$DH <- D$TASX^2 / (2 * D$Cp)
# filter DH:
if (Rate == 25) {
  CutoffPeriod <- Rate * 0.5  # assumed time constant for ATF1
  CutoffPeriod2 <- Rate * 1.0 # assumed time constant for ATH1
} else {  # This is mostly to avoid program failure for a LRT file
  CutoffPeriod <- Rate * 2  # assumed time constant for ATF1
  CutoffPeriod2 <- Rate * 2.2 # assumed time constant for ATH1
}

D$DHF <- zoo::na.approx (as.vector(D$DH), maxgap=1000*Rate,
                          na.rm = FALSE, rule = 2)
D$DHF[is.na(D$DHF)] <- 0
D$DHF <- signal::filtfilt (signal::butter (3,
                            2/CutoffPeriod), D$DHF)
D$DHH <- zoo::na.approx (as.vector(D$DH), maxgap=1000*Rate,
                          na.rm = FALSE, rule = 2)
D$DHH[is.na(D$DHH)] <- 0
D$DHH <- signal::filtfilt (signal::butter (3,
                            2/CutoffPeriod2), D$DHH)
```

```
D$ATFY <- D$ATF1 + (D$DH - D$DHF) *
          RecoveryFactor(D$MACHX, probe='UNHEATED')
D$ATHY <- D$ATH1 + (D$DH - D$DHH) * RecoveryFactor(D$MACHX)

## create and open a copy of the old file for writing:
fnew <- sub ('.nc', 'T.nc', fname)
## beware: overwrites without warning!!
Z <- file.copy (fname, fnew, overwrite=TRUE)
netCDFfile <- nc_open (fnew, write=TRUE)
## retrieve dimension info from the old file
Dimensions <- attr (D, "Dimensions")
Dim <- Dimensions[["Time"]]
if ("sps25" %in% names (Dimensions)) {
  Rate <- 25
  Dim <- list(Dimensions[["sps25"]], Dimensions[["Time"]])
}
DATT <- D

## variables to add to the netCDF file:
VarNew <- c("ATFY", "ATHY")
VarOld <- c("ATF1", "ATH1")
VarUnits <- rep("deg_C", 2)
VarLongName <- rep("Ambient Temperature, Unheated filtered", 2)
VarStdName <- rep("air_temperature", 2)

## create the new variables
varCDF <- list ()
for (i in 1:length(VarNew)) {
  ## create the new variable and add it to the netCDF file
  varCDF[[i]] <- ncvar_def (VarNew[i],
                  units=VarUnits[i],
                  dim=Dim,
                  missval=as.single(-32767.), prec="float",
                  longname=VarLongName[i])
  if (i == 1) {
    newfile <- ncvar_add (netCDFfile, varCDF[[i]])
  } else {
    newfile <- ncvar_add (newfile, varCDF[[i]])
  }
  ## transfer attributes from the old variable and add new ones
  ATV <- ncatt_get (netCDFfile, VarOld[i])
  copy_attributes (ATV, VarNew[i], newfile)
```

```
  ncatt_put (newfile, VarNew[i], attname="standard_name",
             attval=VarStdName[i])
  ## add the measurements for the new variable
  if (Rate == 1) {
    ncvar_put (newfile, varCDF[[i]], D[, VarNew[i]])
  } else if (Rate == 25) {
    ncvar_put (newfile, varCDF[[i]], D[, VarNew[i]],
               count=c(25, nrow(D)/25))
  }
}
## then close to write the new file
nc_close (newfile)
```

– End of Memo –

Reproducibility:

PROJECT:	FilterDynamicHeating
ARCHIVE PACKAGE:	FilterDynamicHeating.zip
CONTAINS:	attachment list below
PROGRAM:	FilterDynamicHeating.Rnw
ORIGINAL DATA:	/scr/raf_data/SOCRATES/rf15h.nc
WORKFLOW:	WorkflowFilterDynamicHeating.pdf
GIT:	https://github.com/WilliamCooper/Reprocessing/FilterDynamicHeating.git

Attachments: FilterDynamicHeating.Rnw
FilterDynamicHeating.pdf
WorkflowFilterDynamicHeating
SessionInfo