

UV Hygrometer

This note records what I found in a quick look at the UV hygrometer processing and recommends how this can be updated to current vapor-pressure processing.

Processing in use:

The processing chain in use is this:

1. The voltage from the UVH is converted to water vapor density via cal coefficients. For IDEAS-4, the netCDF header says they are {38,-11,0}. Incidentally, the units in that header are wrong; it says “deg_C”, should be g/m³
2. Then, after validity checks, a 300-s running average of (rhodp-rhouv) is calculated.
3. rhouv is then corrected by adding this difference, *multiplied by the first cal coefficient*, to rhouv.
4. rhouv is then used to calculate the water vapor pressure. This is based on the Magnus-Tetens equation for vapor pressure:

$$e = e_0 \exp\left(\frac{aT_{DP}}{b + T_{DP}}\right) \quad (1)$$

where $e_0 = 6.105\text{mb}$ and $a=17.27$, $b=237.7$.¹ Because the vapor pressure is related to the vapor density via $e = \rho R_w T$, this can be rewritten as

$$\frac{aT_{DP}}{b + T_{DP}} = \ln \frac{e}{e_0} = \ln \left(\frac{\rho R_w T}{e_0} \right) \quad (2)$$

The processing code that implements this uses “ela” for e/e_0 .²

$$ela = \frac{rhouv * T_k}{1322.3}$$

5. This vapor pressure (ela) is then converted from (1):

$$T_{DP} = \frac{237.3 \log(ela)}{17.27 - \log(ela)}$$

which gives the dew point in Celsius units.

¹I think this number is correct and perhaps somewhere a typo has changed the value being used to 237.3

²I calculate 1322.8 instead of 1322.3

Comments:

1. I suspect an error has entered the processing chain in step 3, where the first calibration coefficient multiplies the feedback term. It may be a clue that the default value for this coefficient is 1 and other coefficients are zero; as interpreted now, that would suppress all response to the measurements, so that is not a meaningful default and suggests that the use of these coefficients has perhaps changed from their original intent. The first calibration coefficient is the constant term in the polynomial calibration, so I can't see why this should enter here. It leads to two problems:
 - (a) The feedback has a time constant much faster than 300 s, but 300 s or longer might be a reasonable time for this feedback. The first coefficient in use now is 38, so I think the feedback is 38 times more than is desired. This risks instability in the feedback to have the feedback term this large.
 - (b) As implemented, 38/300 of the signal is lost at 1 Hz from having the feedback this large. This means that measureThis note records what I found in a quick look at the UV hygrometer processing and recommends how this can be updated to current vapor-pressure processing.ments like fluxes would have only 87% the amplitude that is measured from a 1-Hz fluctuation to which the dewpoint sensor does not respond.
2. Would it be better, in the data-quality checks, to compare diff to some fraction of rhov , like “if ($\text{fabs}(\text{diff}) > \text{rhov} * 0.5$) { $\text{diff} = \text{copysign}(\text{rhov} * 0.5, \text{diff})$;}”? Using an absolute value here seems puzzling for a variable with as large a range as is normal for rhov .
3. The Tetens or Magnus-Tetens equation (1) is not a good choice for general use because, as originally developed, it applies above the freezing point. It departs significantly from the dew point at low temperature. That is a good reason for changing the algorithm in use. Note that this may be important if, as Chris thought, the UVH was used as the primary humidity reference for ICE-T: The sub-zero values there are quite important for the project.
4. The processing chain is analogous to what is used for the chilled-mirror sensors: From the original measurement, determine e , and from that determine T_{DP} . In the case of the UVH, the determination of e is via the perfect gas law (the last equality is (2)) while for the chilled-mirror instruments it is via either Goff-Gratch or Murphy-Koop applied to the mirror temperature, with correction for the enhancement factor. Then, for the chilled-mirror instruments, the determination of T_{DP} is via an interpolation table based on Murphy-Koop (or, in the past, an equation based on Goff-Gratch), while for the UVH the conversion is based on the Magnus-Tetens equation.
5. The running average is implemented in awkward code and could be made much simpler and clearer. Still better, in my opinion, would be an exponential update; this has better spectral characteristics, is easier to adjust, and can be implemented in simple and obvious code. See “Recommendations.”

Suggestions / Recommendations

1. Consider making the code consistent with the processing chain for the chilled-mirror hygrometers, as follows:

- (a) Calculate rhouv as at present, perhaps with substitution of exponential updating for block averaging as suggested below.
- (b) Calculate vapor pressure from the perfect gas equation, using rhouv and ATX:

$$\{\text{EDP}\} = \{\text{RHOUV}\} * (\{\text{ATX}\} + 273.15) * R_w / (C_{mb2Pa} * C_{kg2g})$$

where $R_w = 8.314472 \times 10^3 \text{ J kmol}^{-1} \text{ K}^{-1} / 18.0153 \text{ kg kmole}^{-1} = 461.5228 \text{ J kg}^{-1} \text{ K}^{-1}$, $C_{mb2Pa} = 100 \text{ Pa mb}^{-1}$, and $C_{kg2g} = 10^3 \text{ g kg}^{-1}$, so that the net result is

$$\{\text{EDP}\} = 4.615228 \times 10^{-3} \{\text{RHOUV}\} * (\{\text{ATX}\} + 273.15)$$

- (c) If a dew point is desired, use the Murphy-Koop interpolation table
2. Fix the updating code. Options are:
 - (a) remove the cal coefficient term from the update term. In addition, if you keep block averaging: this is very awkwardly coded now, so Chris could clean it up. There is value in that beyond esthetics: Someone else reading this will have an easier time grasping quickly what is happening. For example, I like to do these something like this, or even combine the two statements into one line with a comma so that the entire circular-buffer handling is isolated to that one line.

```
uv_sum += diff-uv_buffer[buffIndex = ++buffIndex%300] ;
uv_buffer[buffIndex] = diff;
```

- (b) change to exponential updating, as follows (perhaps with Tau=300 to give a 300-s time constant for 1 Hz processing). Maybe something like:

```
offset += (rhodt-rhouv-offset)/Tau;
rhouv += offset;
```

These statements can be separated, as in the present code, so that the update only occurs when the validity checks are true but the offset is applied to all measurements.

NOTE: I HAVEN'T TESTED ANY OF THIS! VALUES LIKE 300 S OR 0.5 WOULD NEED RE-TESTING IF REDEFINED AS SUGGESTED HERE.