**Instruction Guide for the AERIoe Retrieval Algorithm**

Dave Turner, NOAA

7 March 2016

AERIoe is a 1-d variational retrieval that attempts to retrieve profiles of temperature and humidity, as well as several other parameters (if selected) from AERI observed radiances. The observation vector could include much more than only the AERI observations, depending on the options set in the Variable Input Parameter (VIP) file. For example, lidar profiles of water vapor, or NWP model output, can also be used as an input observation to the retrieval, as can time-interpolated radiosonde profiles, microwave radiometer brightness temperatures, surface meteorology observations, and more.

The retrieval uses an optimal estimation framework to propagate the uncertainties from the observations (and the sensitivity of the forward model) to provide a full error characterization of the retrieved solution (i.e., a posterior covariance matrix is provided for every retrieval). The forward model used to simulate the AERI observations is the LBLRTM, and other forward models have been written to simulate the other “observations” used in the retrieval.

The LBLRTM needs to be installed in a directory with a specific subdirectory structure so that the code is able to find all of the needed pieces. Furthermore, the AERIoe uses the “lblrun” script that was written by Paul van Delst decades ago to facilitate the LBLRTM runs. If MWR observations are also included in the retrieval, then the MonoRTM needs to be installed including the “wrapper” routine that I’ve written to facilitate the running of that model.

This code is pretty stable and is able to provide quality thermodynamic retrievals over 95% of the time for a well-calibrated AERI system. However, this is a research algorithm and thus is always in some state of development. The details of an earlier version (pre Release\_1\_0) are described in the paper by Turner and Löhnert JAMC 2014.

**Understanding the Variable Input Parameter (VIP) file** is important to change the behavior of the retrieval code, as virtually all of the important parameters come in via this mechanism. The command line is only used to pass in the date / times to execute, the name of the VIP file, and the name of the file with the a priori information in it. All of the VIP keywords have default values, so the user’s VIP file does not need to define them. However, *I recommend that the user define any keywords that are in green below*. Keywords that are in gray are in active development, and should not be used (please).

* **Most common VIP entries used are highlighted in Green**
* **Other VIP entries that are used by some are highlighted in cyan**
* **VIP entries that are experimental are in highlighted in gray**

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Type** | **Value and Meaning** |
| tres | Integer | 0 – run retrieval at native resolution |
| X > 0 is the temporal resolution [minutes]. Note that the sample time will be associated with the middle of the averaging period. |
| avg\_instant | Integer | 0 – average radiance over the temporal resolution |
| 1 –an instantaneous sample (i.e., only one AERI spectrum closest to the gridded time) per averaging temporal resolution interval |
| tag | String | Name for a temporary subdirectory that will be created in the working directory (the latter will be specified below) |
| **AERI Information** | | |
| aeri\_type | Integer | 0 – output the various options and exit |
| 1 – ARM netcdf files (ch1 or ch2, summary, and engineering data needed) |
| 2 – DMVtoCDF conversion with names “C1\_rnc.cdf” and “\_sum.cdf” |
| 3 – DMVtoCDF conversion with names “C1.RNC.cdf” and “.SUM.cdf” |
| 4 – ARM-style data stream names, but all engineering data found in summary file |
| aeri\_pca\_nf  (not yet fully implemented) | Integer | 0 – AERI data was not PCA noise filtered |
| 1 – AERI data was PCA noise filtered |
| aerich1\_path | String | Path to the AERI radiance data |
| aerisum\_path | String | Path to the AERI summary/sum data |
| aerieng\_path | String | Path to the AERI engineering/sum data |
| aeri\_smooth\_noise | Integer | Temporal window [minutes] to smooth the AERI noise with time |
| aeri\_calib\_pres | 2 element vector of float | Two-element vector [intercept, slope] that has the linear correction to apply to calibrate the internal AERI pressure sensor. newP = intercept + slope \* oldP |
| aeri\_use\_missingDataFlag | Integer | 0 – process all AERI data |
| 1 – use the “missingDataFlag” in ch1 file to remove bad AERI spectra before any averaging/retrieval. Note that the AERI only rarely sets this flag. However, if you would like to pre-screen your AERI data to identify bad spectra, then just reset this flag for those bad samples |
| aeri\_hatch\_switch | Integer | 1 – only include “hatchOpen” == 1 when averaging the data (all other spectra are removed |
| 2 – include all AERI samples when averaging (assumption is hatchOpen flag is not working) |
| aeri\_fv | Float | X > 0 implies that a foreoptics correction for obscuration should be applied |
| aeri\_fa | Float | X > 0 implies that an AERI aftoptics correction should be applied |
| aeri\_lat | Float | If “aeri\_alt” (below) > 0, then use this as the latitude for the AERI instrument [degN] |
| aeri\_lon | Float | If “aeri\_alt” (below) > 0, then use this as the longitude for the AERI instrument [degE] |
| aeri\_alt | Float | < 0 implies that the latitude / longitude / altitude stored in the output netCDF file should come from the AERI data file |
| >= 0 implies that the VIP file values should be used as the instrument location in the output netCDF file. This value is the altitude of the instrument [m MSL] |
| psfc\_min | Float | QC for the minimum surface pressure from the AERI [mb] |
| psfc\_max | Float | QC for the maximum surface pressure from the AERI [mb] |
| aeri\_min\_675\_bt | Float | Minimum brightness temperature [K] in the 675-680 cm-1 window (QC screen) |
| aeri\_max\_675\_bt | Float | Maximum brightness temperature [K] in the 675-680 cm-1 window (QC screen) |
| **External profile information** | | |
| ext\_wv\_prof\_type | Integer | 0 – no external WV profile used in the retrieval |
| 1 – ARM radiosonde used |
| 2 – ARM Raman lidar (rlprofmr) used |
| 3 – NCAR WV DIAL used |
| 4 – NWP model output (GB format) used. While possible, please use “mod\_wv\_prof\_type” below instead! |
| 99 – (Test) AER’s GVRP water vapor profile from RHUBC-2 used |
| ext\_wv\_prof\_path | String | Location of the data for the external WV profile data, if needed |
| ext\_wv\_prof\_minht | Float | Minimum height to use the external WV profile data in the retrieval [km AGL] |
| ext\_wv\_prof\_maxht | Float | Maximum height to use the external WV profile data in the retrieval [km AGL] |
| ext\_wv\_noise\_mult\_val | 3-element array of Float | Multipliers to apply to the noise profile of the external WV profiler. Must be > 0. |
| ext\_wv\_noise\_mult\_hts | 3-element array of Float | Heights [km AGL] for the noise multipliers above (i.e., a profile of multipliers) |
| ext\_wv\_add\_rel\_error | Float | If > 0, then add this relative error [%] (computed as this value/100 \* profile) to the WV noise profile |
| ext\_wv\_time\_delta | Float | Maximum amount of time from endpoints of the external WV dataset to extrapolate [hours] |
| ext\_temp\_prof\_type | Integer | 0 – no external temperature profile used in the retrieval |
| 1 – ARM radiosonde used |
| 2 – ARM Raman lidar (rlproftemp) used |
| 4 – NWP model output (GB format) used. While possible, please use “mod\_temp\_prof\_type” below instead! |
| 99 – (Test) AER’s profile from RHUBC-2 used |
| ext\_temp\_prof\_path | String | Location of the data for the external temperature profile data, if needed |
| ext\_temp\_prof\_minht | Float | Minimum height to use the external temperature profile data in the retrieval [km AGL] |
| ext\_temp\_prof\_maxht | Float | Maximum height to use the external temperature profile data in the retrieval [km AGL] |
| ext\_temp\_noise\_adder\_val | 3-element array of Float | Factors to add to the noise profile of the external temperature profiler. Must be >= 0. |
| ext\_temp\_noise\_adder\_hts | 3-element array of Float | Heights [km AGL] for the noise adder above (i.e., a profile of values to add to the noise of the external noise profile) |
| ext\_temp\_time\_delta | Float | Maximum amount of time from endpoints of the external temperature dataset to extrapolate [hours] |
| **NWP model profile information** | | |
| mod\_wv\_prof\_type | Integer | 0 – no NWP WV profile used in the retrieval |
| 4 – NWP model output (GB format) used |
| mod\_wv\_prof\_path | String | Location of the data for the NWP WV profile data, if needed |
| mod\_wv\_prof\_minht | Float | Minimum height to use the NWP WV profile data in the retrieval [km AGL] |
| mod\_wv\_prof\_maxht | Float | Maximum height to use the NWP WV profile data in the retrieval [km AGL] |
| mod\_wv\_noise\_mult\_val | 3-element array of Float | Multipliers to apply to the noise profile of the NWP WV profiler. Must be > 0. |
| mod\_wv\_noise\_mult\_hts | 3-element array of Float | Heights [km AGL] for the noise multipliers above (i.e., a profile of multipliers) |
| mod\_wv\_time\_delta | Float | Maximum amount of time from endpoints of the NWP WV dataset to extrapolate [hours] |
| mod\_temp\_prof\_type | Integer | 0 – no NWP temperature profile used in the retrieval |
| 4 – NWP model output (GB format) used |
| mod\_temp\_prof\_path | String | Location of the data for the NWP temperature profile data, if needed |
| mod\_temp\_prof\_minht | Float | Minimum height to use the NWP temperature profile data in the retrieval [km AGL] |
| mod\_temp\_prof\_maxht | Float | Maximum height to use the NWP temperature profile data in the retrieval [km AGL] |
| mod\_temp\_noise\_adder\_val | 3-element array of Float | Factors to add to the noise profile of the NWP temperature profiler. Must be >= 0. |
| mod\_temp\_noise\_adder\_hts | 3-element array of Float | Heights [km AGL] for the noise adder above (i.e., a profile of values to add to the noise of the external noise profile) |
| mod\_temp\_time\_delta | Float | Maximum amount of time from endpoints of the NWP temperature dataset to extrapolate [hours] |
| **External surface meteorology information** | | |
| ext\_sfc\_temp\_type | Integer | 0 – No external surface temp data used |
| 1 – ARM met data used |
| 2 – NCAR ISFS data used |
| 3 – CLAMPS met data used |
| ext\_sfc\_temp\_rep\_error | Float | Noise value to add [degC] to add to the surface temperature obs uncertainty to account for representativeness error |
| ext\_sfc\_wv\_type | Integer | 0 – no external surface WVMR data used |
| 1 – ARM met data used |
| 2 – NCAR ISFS data used |
| 3 – CLAMPS met data used |
| ext\_sfc\_wv\_mult\_error | Float | Multiplier to use to scale the surface WVMR uncertainty to account for representativeness error |
| ext\_sfc\_path | String | Location of the data for the external surface met data, if needed |
| ext\_sfc\_time\_delta | Float | Maximum amount of time from endpoints of the external temperature dataset to extrapolate [hours] |
| ext\_sfc\_relative\_height | Integer | Height of met station relative to the AERI sky port [m] |
| **Microwave radiometer information** | | |
| mwr\_type | Integer | 0 – no MWR obs included |
| 1 – Tb fields are individual time series |
| 2 – Tb fields are in a 2-dim array |
| mwr\_path | String | Location of the data for the external MWR data, if needed |
| mwr\_rootname | String | Rootname of the input MWR netcdf file |
| mwr\_elev\_field | String | Name of the elevation field in the netcdf file. The assumption is that data in this field that are 90 deg are zenith. If field is not found, then all obs are assumed to be zenith and will be used |
| mwr\_n\_tb\_fields | Integer | Number of Tb fields to use in retrieval. |
| mwr\_tb\_field\_names | String | Single string, comma delimited, that has the names of the Tb fields in the netCDF file. If mwr\_type == 2 then this would have a single field name; if mwr\_type == 1 then there should be mwr\_n\_tb\_fields in this list |
| mwr\_tb\_field1\_tbmax | Float | Maximum value allowed in the first Tb field to be used for QC [degK]. If the check fails, then all MWR obs at that time will be considered bad (e.g., due to rain, etc) |
| mwr\_tb\_freqs | String | Single string, comma delimited, that has the frequencies to use in the forward calculation for each channel [GHz] |
| mwr\_tb\_noise | String | Single string, comma delimited, that has the noise values to use for each frequency [degK] |
| mwr\_tb\_bias | String | Single string, comma delimited, that bias in each MWR channel [degK]; these values will be ADDED to the MWR Tb obs |
| mwr\_pwv\_field | String | Name of the PWV field in the MWR field; this will be “passed-through” to the output of the AERIoe output file |
| mwr\_pwv\_scalar | Float | Multiplier to use to convert the MWR-provided PWV into units of [cm] |
| mwr\_lwp\_field | String | Name of the LWP field in the MWR field; this will be “passed-through” to the output of the AERIoe output file |
| mwr\_lwp\_scalar | Float | Multiplier to use to convert the MWR-provided LWP into units of [g/m2] |
| **Ceilometer (cloud base height) information** | | |
| cbh\_type | Integer | 0 – output options and stop |
| 1 – ARM ceilometer |
| 2 – ASOS data (GB “cbh” format) |
| 3 – CLAMPS DLfp format |
| 4 – ARM dlprofwstats format |
| cbh\_path | String | Location of the cloud base height files |
| cbh\_window\_in | Float | Inner temporal window, full-sized centered upon the output time to look for cloud [minutes]. This sets an uncertainty value |
| cbh\_window\_out | Float | Outer temporal window, full-sized centered upon the output time to look for cloud [minutes]. This sets a larger uncertainty value. |
| cbh\_default\_ht | Float | The default CBH [km AGL] to use if the above data does not find a cloud within the outer window (or data does not exist) |
| **Output file information** | | |
| output\_rootname | String | Rootname (also called the ‘platform’ name in ARM) for the output data file |
| output\_path | String | Location that the output data should be written |
| output\_clobber | Integer | 0 – do not clobber any pre-existing file with the same name. Instead, output an error condition and stop |
| 1 – Clobber any pre-existing output file with the same name |
| 2 – Append to an output file that has the same date, as long as the structure of that netCDF file matches the one that would be created (e.g., same dimensions, same basic VIP parameters) |
| output\_keep\_file\_small | Integer | 0 – all fields written out (default) |
| 1 – do not write Sop, Akernal, other fields to keep output file smaller |
| **Radiative transfer model information** | | |
| lbl\_home | String | Path to the LBLRTM distribution. The subdirectories here should include “bin” and “hitran”, and the “bin” subdirectory should contain the PvD’s “lblrun” script |
| lbl\_version | String | A string that describes the version of the LBLRTM being used |
| lbl\_temp\_dir | String | A working directory that will be used by lblrun script. It should be on a fast disk (i.e., do not use an NFS mounted disk) |
| lbl\_std\_atmos | Integer | The background atmosphere to use for the other absorbing gases and above the atmosphere being retrieved. The values are: |
| 1 – Tropical Atmosphere |
| 2 – Mid-latitude summer atmosphere |
| 3 – Mid-latitude winter atmosphere |
| 4 – Sub-arctic summer atmosphere |
| 5 – Sub-arctic winter atmosphere |
| 6 – US Standard atmosphere |
| lbl\_tape3 | String | Name of the tape3 file (which contains the HITRAN spectroscopic info) to use. Default is ‘tape3.data’. This should be a filename, and should be found in the directory lbl\_home/HITRAN. |
| monortm\_wrapper | String | Executable path/name that runs the wrapper for the MonoRTM |
| monortm\_version | String | String with the version information on the MonoRTM being used |
| monortm\_exec | String | Path/Name of the executable for the AER model MonoRTM. This is used to build the config file for the Monortm wrapper |
| monortm\_spec | String | Path/Name of the spectral line file (TAPE3) for the MonoRTM. This is used to build the config file for the Monortm wrapper |
| lblrtm\_jac\_option | Integer | 1 – Use LBLRTM finite differences. (This is very slow and does not allow clouds) |
| 2 – “3 calc” method |
| 3 – “deltaOD 3calc” method (recommended and default) |
| lblrtm\_forward\_threshold | Float | Upper LWP threshold [g/m2]; use the LBLRTM direct radiance calculation below this value and the Radxfer calculation above this value |
| monortm\_jac\_option | Integer | 1 – MonoRTM finite differences (this is very slow) |
| 2 – 3calc method (recommended and default) |
| jac\_max\_ht | Float | Compute the jacobian from the surface to this maximum altitude [km AGL] |
| **Retrieval options** | | |
| max\_iterations | Integer | Maximum number of iterations to perform (default is 10) |
| first\_guess | Integer | 1 – use Prior as FG |
| 2 – use 7 deg/km lapse rate and 60% RH profile as FG |
| 3 – use solution for previous sample as FG (or prior if no previous sample) |
| superadiabatic\_maxht | Float | Maximum height above the surface where a superadiabatic layer can exist [km AGL] |
| spectral\_bands | 2 x N matrix of float | The spectral band(s) of the AERI data to use in the retrieval (units of cm-1), where the limits of the band are indicated with a dash and each band is separated with a comma. The bands do not have to be in a sorted order, but the each band should go from minimum to maximum value. Example with 3 spectral regions is:  675-710, 538-588, 895-905  (Maximum allowed is 200 bands) |
| retrieve\_ch4 | Integer | 0 – do not retrieve methane |
| 1 – do retrieve methane using exponential model |
| 2 – do retrieve methane using stair step model (not fully implemented yet) |
| retrieve\_n2o | Integer | 0 – do not retrieve nitrous oxide |
| 1 – do retrieve nitrous oxide using exponential model |
| 2 – do retrieve nitrous oxide using stair step model (not fully implemented yet) |
| retrieve\_co2 | Integer | 0 – do not retrieve carbon dioxide |
| 1 – do retrieve carbon dioxide using exponential model |
| 2 – do retrieve carbon dioxide using stair step model (not fully implemented yet) |
| retrieve\_lcloud | Integer | 0 – do not retrieve liquid cloud properties |
| 1 – do retrieve liquid cloud properties (LWP and effective radius) |
| retrieve\_icloud | Integer | 0 – do not retrieve ice cloud properties |
| 1 – do retrieve ice cloud properties (optical depth and effective radius) |
| lcloud\_ssp | String | Path/Name of the single scattering property database for liquid cloud properties integrated over an assumed size distribution (of the SSP\_DB format) |
| icloud\_ssp | String | Path/Name of the single scattering property database for ice cloud properties integrated over an assumed size distribution (of the SSP\_DB format) |
| qc\_rms\_value | Float | The RMS threshold value (for obs\_vector minus forward\_calc), where values less than this threshold have the QC flag in the output file set to “success” (if they converged that is) |
| **Prior mean and covariance information** | | |
| prior\_t\_ival | Float | The multiplicative inflation factor (>= 1) to apply at the surface in the prior’s covariance matrix for temperature |
| prior\_t\_iht | Float | The height [km AGL] where the prior\_t\_ival reaches 1.0 for temperature (i.e., this is a linear interpolation between prior\_t\_ival at the surface and 1.0 at this height) |
| prior\_q\_ival | Float | The multiplicative inflation factor (>= 1) to apply at the surface in the prior’s covariance matrix for water vapor |
| prior\_q\_iht | Float | The height [km AGL] where the prior\_q\_ival reaches 1.0 for water vapor (i.e., this is a linear interpolation between prior\_q\_ival at the surface and 1.0 at this height) |
| prior\_tq\_cov\_val | Float | A multiplicative factor (0 < val <= 1) that is used to decrease the correlation between temperature and water vapor at all heights in the prior covariance matrix |
| prior\_chimney | Float | Height [km AGL] of the chimney; prior levels below this level are decorrelated with prior data above this level |
| prior\_co2\_mn | 3-element array of Float | First element: background CO2 level [ppm] in Xa. If set to -1, then use DDT’s simple model to predict CO2 concentration given the year and month |
| Second element: difference between surface and background values [ppm] |
| Third element depends on value of retrieve\_co2: If ==1, then this is the coefficient of the exponential, if ==2, then this is the BL depth [km AGL] |
| prior\_co2\_sd | 3-element array of Float | Uncertainty (1-sigma) of the three elements in prior\_co2\_mn to use in Sa [ppm] |
| prior\_ch4\_mn | 3-element array of Float | First element: background CH4 level [ppm] in Xa |
| Second element: difference between surface and background values [ppm] |
| Third element depends on value of retrieve\_ch4: If ==1, then this is the coefficient of the exponential, if ==2, then this is the BL depth [km AGL] |
| prior\_ch4\_sd | 3-element array of Float | Uncertainty (1-sigma) of the three elements in prior\_ch4\_mn to use in Sa [ppm] |
| prior\_n2o\_mn | 3-element array of Float | First element: background N2O level [ppm] in Xa |
| Second element: difference between surface and background values [ppm] |
| Third element depends on value of retrieve\_n2o: If ==1, then this is the coefficient of the exponential, if ==2, then this is the BL depth [km AGL] |
| prior\_n2o\_sd | 3-element array of Float | Uncertainty (1-sigma) of the three elements in prior\_n2o\_mn to use in Sa [ppm] |
| prior\_lwp\_mn | Float | This is the assumed mean LWP to use in the Xa [g/m2] |
| prior\_lwp\_sd | Float | Uncertainty (1-sigma) to assume in the LWP for prior covariance matrix [g/m2] |
| prior\_lReff\_mn | Float | This is the assumed mean liquid cloud effective radius to use in Xa [microns] |
| prior\_lReff\_sd | Float | Uncertainty (1-sigma) to assume in the liquid cloud effective radius for prior covariance matrix [microns] |
| prior\_itau\_mn | Float | This is the assumed mean ice cloud optical depth to use in Xa [unitless] |
| prior\_itau\_sd | Float | Uncertainty (1-sigma) to assume in the ice cloud optical depth for prior covariance matrix [unitless] |
| prior\_iReff\_mn | Float | This is the assumed mean ice cloud effective radius to use in Xa [microns] |
| prior\_iReff\_sd | Float | Uncertainty (1-sigma) to assume in the ice cloud effective radius for prior covariance matrix [microns] |
| **Global attribute information**  Any number of global attributes can be included in the VIP, as long as they start “globatt\_X” and the “X” is different for each attribute. For example: | | |
| globatt\_Site | String | A string to indicate the site where the instruments are located (i.e., retrieval is being performed) |
| globatt\_Instrument | String | A string to indicate the serial number or other identification for the AERI |
| globatt\_Dataset\_contact | String | Person responsible for collecting / distributing these data |