WindCube software package

version 1.4 (parallel)

Jana Preißler, C-CAPS, NUIG, jana.preissler@nuigalway.ie

May 9, 2016

1 Introduction

This is a documentation for scripts written in Python 2.7 to convert the ASCII files exported from the internal MySQL data base on the WindCube instrument to netCDF file format. This software package can also be used to plot the radial wind, carrier-to-noise ratio and relative backscatter, and calculate horizontal wind speed and direction, and vertical wind speed from VAD scans. This all can be done on an operational basis (run as cron job).

The programme package has been tested on Windows 7 and Ubuntu 12.04 (server). Adjusting the data paths in the configuration file should be the only changes necessary to run it on either of the platforms once all required Python modules are installed.

The Python scripts are explained in the following sections. A list of required Python modules is given at the end of the document.

2 The input

The ASCII input files are exported from the internal MySQL data base on the WindCube using the scripts provided by the manufacturer. Relative input and output paths and input file names with the date as only variable can be specified in the configuration file `config_lidar.py` ('DataPath', 'OutPath', 'txtInput', 'ncInput'). The programme has been tested with file names following the convention `YYYYmmdd-HH_ending.txt`, with `YYYY`, year in four digits; `mm`, month in two digits; `dd` day of the month in two digits; `HH`, hour of the day in two digits. The file endings (`ending`) can be adjusted in the configuration file `config_lidar.py` ('fend' in 'VarDict'). They are pre-set to the following:

- whole_radial: contains the whole radial wind and carrier-to-noise ratio (CNR),
- radial_beta: contains the attenuated relative backscatter,
- spectra: contains the Doppler spectra,

- dbs_wind: contains the wind components from the DBS scans.

The folder `YYYY\` is appended to the output path. If the year folder is missing, it will be added to the output path.

3 The configuration file

All variable parameters are specified in the configuration file `config_lidar.py`. The user needs to modify the input and output paths, the scan ID numbers, the global attributes and general variables for netCDF files (location, latitude, longitude, etc.; `GloDict` and `GenDict['val']`). Also, the date of the run needs to be set. The script can be run as cron job in near real time operation. Alternatively, as string can be provided as `sDate`.

The list `proplist` includes all runs, the software should go through: ['beta', 'wind', 'dbs', 'spectra']. The run 'spectra' reads in the spectra files and stores them as netCDF. No plots are done in this run. The run 'beta' reads in the beta files, stores them as netCDF and plots the different scans depending on `SWITCH_MODE, SWITCH_PLOT, SWITCH_OUTNC` (see below). The run 'wind' reads in the radial wind files, stores them as netCDF and plots the different scans depending on `SWITCH_MODE, SWITCH_PLOT, SWITCH_OUTNC` (see below). It also calculates the horizontal wind and direction as well as the vertical wind velocity and plots them, if 'VAD' is included in `SWITCH_MODE`. The run 'dbs' is for testing purposes only. It can be used to compare the wind components from a DBS to VAD results.

The switches can be used to customise the output. The user has the choice:

- to plot the background, or use the confidence index provided by the instrument to remove the background (`SWITCH_REMOVE_BG`);
- to zoom in on the background noise by reducing the colour scale limits (`SWITCH_ZOOM`);
- to plot results or not (`SWITCH_PLOT`);
- to store results to netcdf or not (`SWITCH_OUTNC`);
- to print status messages on screen if run from command line (`SWITCH_OUTPUT`);
- of timing of the main processes while running the script and printing time elapsed since start of script if output is activated (`SWITCH_TIMER`);
- of the scan types that will be processed and plotted (either all types (['all']) or specific types provided in a list: ['VAD', 'LOW', 'LOS', 'LOS90']) (`SWITCH_MODE`);
- to create additional output in HDCP2-like netCDF format (requires, in run 'wind', existing beta netCDF file, or in run 'beta', existing wind and VAD netCDF file)(`SWITCH_HDCP2`);
- to delete text files after converting them to netcdf format (`SWITCH_CLEANUP`, *in testing phase*);

- of the number of parallel processing pools to use to read input and fit VAD (0 for no parallel processing)(`SWITCH_POOL`).

The `SWITCH_NC` should remain at False, as the other two options are faulty or not properly tested. At this setting all text files of the specified date in data path are used as input.

The dictionary `CompDict` includes the scan IDs of the composite scans used for VADs (instead of PPI). The dictionary keys are scan IDs of the composites and the values are a list of the scan ID's of the individual LOS measurements. This is used to combine all relevant LOS scans for the VAD fitting and the netcdf output.

For hard-target scans, a zoom in to specific LOS scans can be set up. The range limits are adjusted in dictionary `LOSzoom`, where the dictionary keys are scan IDs and the values are a list of plot limits in meter [min_range, max_range].

The large dictionary `VarDict` contains all variables for reading the text files, plotting and netcdf creation. There is one entry in the dictionary for each output parameter (level 0: spectra; level 1: wind, cnr, beta, dbs; level 2: VAD). The dictionary `AttDict` contains variable information for the HDCP2-like output. *This needs cleaning up / combining VarDict and AttDict.*

4 The windcube tools

This script includes all tools for data reading, output, and the wind fit (VAD).

4.1 Input

get_data
Reads in data from text files returned from the MySQL data base on the instrument. Returns a pandas data frame including all data.

open_existing_nc
Opens existing netCDF file, created by `export_to_netcdf`. Returns a pandas data frame.

4.2 NetCDF output

export_to_netcdf
Prepares pandas data frame for export to netcdf file.

create_xray_dataset
Exports xray data set to netCDF file, including global attributes, long names and units.

all_att_to_df
Adds all attributes to a dictionary, which is then added to the data frame; calls `add_attribute`.

add_attribute
Adds attribute to netcdf variable, if field is not empty.

4.3 Wind fit

wind_fit
Runs a loop over all VAD scans and fits a sine curve to each range bin (least square fit). The loop over all VAD scans is executed in parallel pools, if `SWITCH_POOL`¿1 in the configuration file. For optimal performance, the number of pools should be equal the number of different VAD scans performed by the instrument (`plot_ts`). This function also calls the plotting function to plot the horizontal wind speed and direction and the vertical wind. Exports results to a separate netCDF file for each VAD scan type. Additionally, it produces combined VAD plots using the results from a 15 degree VAD up to 150 m above ground and the results from a 75 degree VAD above that.

run_fit
The actual fit function, applied during the loop over all VAD scans and range bins.

change_scan_IDs
Changes single LOS scan IDs of VAD composites to one ID.

set_outliers_to_nan
Removes outliers before the wind fit.

4.4 Others

printif
Prints message if output option is set in configuration file (`SWITCH_OUTPUT`).

timer
Calculates time since start of the script. Prints the difference, if output option is set to True.

5 The windcube plotting

This script contains all functions necessary for plotting the different scan patterns and results.

prepare_plotting
Brings the pandas data frame to a 2-dimensional grid. Returns also axes limits and color bar properties.

get_lims
Determines the plotting limits and returns them. Called by `prepare_plotting`.

plot_ts
Plots a time-height contour plot. Stores it in the output directory.

plot_low_scan
Plots low level scan data on a polar grid.

plot_los
Plots LOS (line-of-sight) scans. Calls `plot_ts`.

plot_correlation
Scatter plot of two cloumns in a pandas data frame.

6 The windcube extras

This script contains extra functions, mostly used for testing.

compare_dbs

Compares DBS wind to VAD scan fit results.

create_hdcp2_output

Brings the data in correct format to be processed by `export_to_netcdf` and output in HDCP2-like netCDF format.

7 The run file

The file `run.py` should be used to start the programme. It is importing all information from the configuration file as well as all functions from the other files. It then calls the functions according to the specifications in the configuration file, especially the switches. The function calls `get_data` separately for each input file in a loop, which is executed in parallel pools if `SWITCH_POOL`¿1 in the configuration file.

8 Python modules

It is recommended to install Python 2.7 from `https://www.python.org/downloads/` and Anaconda from `https://www.continuum.io/downloads` and use the Anaconda terminal to install additional packages and run the scripts.

datetime date and time conversions

time process timing

os file and path handling

glob handling files

pdb de-bugging

numpy numerical operations

scipy least square fit of wind data (namely scipy.optimize)

matplotlib plotting (namely matplotlib.pyplot and matplotlib.dates)

pandas easier handling of large data sets

seaborn plotting

xray conversion to netcdf

multiprocessing parallel processing