

WindCube software package

version 1.0

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

October 29, 2015

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.

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 change necessary.

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

2 The configuration file

All variable parameters are specified in the configuration file `config_lidar.py`. The user needs to modify the data (input) and output paths, the scan ID numbers, the global attributes for netCDF files (location, latitude, longitude, etc.). 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 switches can be used to customise the output. The user has the choice to plot the background, or use the confidence index to remove the background (`SWITCH_REMOVE_BG`); zoom in on the background noise by reducing the colour scale limits (`SWITCH_ZOOM`); use existing netCDF files if they are available in data path (True), or uses all text files in data path as input (False), or appends latest text file in data path to existing netcdf file in data path and removes this text file ('append') (`SWITCH_NC`); print status messages on screen if run from command line (`SWITCH_OUTPUT`); 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: 'VAD', 'LOW', 'LOS', 'LOS90') (`SWITCH_MODE`).

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; level 2: VAD). *The reading and storing of spectra is in place but needs improvement considering the output.*

3 The windcube tools

This script includes all functions used for data conversion and plotting.

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.

export_to_netcdf

Exports pandas data frame to netCDF file, including global attributes, long names and units.

open_existing_nc

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

wind_fit

Runs a loop over all VAD scans and fits a sine curve to each range bin (least square fit). 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.

set_outliers_to_nan

Removes outliers before the wind fit.

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`.

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.

4 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 tools file. It then calls the functions according to the specifications in the configuration file, especially the switches.

5 Python modules

datetime date and time conversions

glob handling files

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

numpy numerical operations

os file and path handling

pandas easier handling of large data sets

pdb de-bugging

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

seaborn plotting

time process timing

xray conversion to netcdf