Supplement of Atmos. Meas. Tech., 15, 1983–2000, 2022
https://doi.org/10.5194/amt-15-1983-2022-supplement

Supplement of

A relaxed eddy accumulation (REA) LOPAP system for flux measurements of nitrous acid (HONO)

Lisa von der Heyden et al.

Correspondence to: Jörg Kleffmann (kleffman@uni-wuppertal.de)

Section S1: External sampling unit with REA-inlet

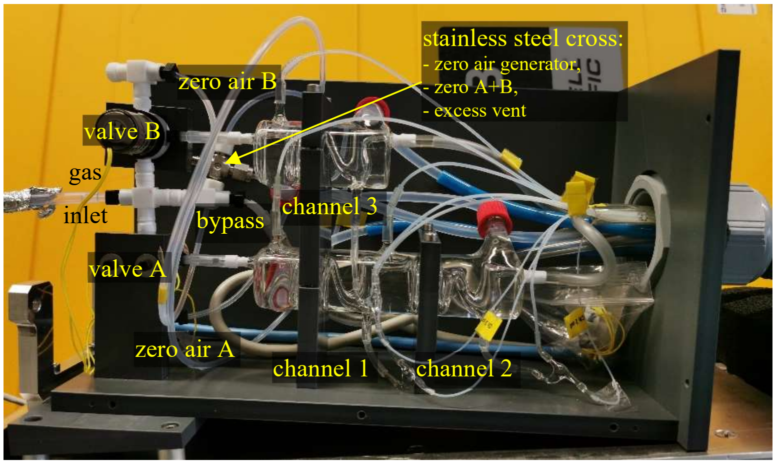

Figure S1: External sampling unit of the REA-LOPAP, here opened.

Section S2: Liquid flow scheme of the REA-LOPAP

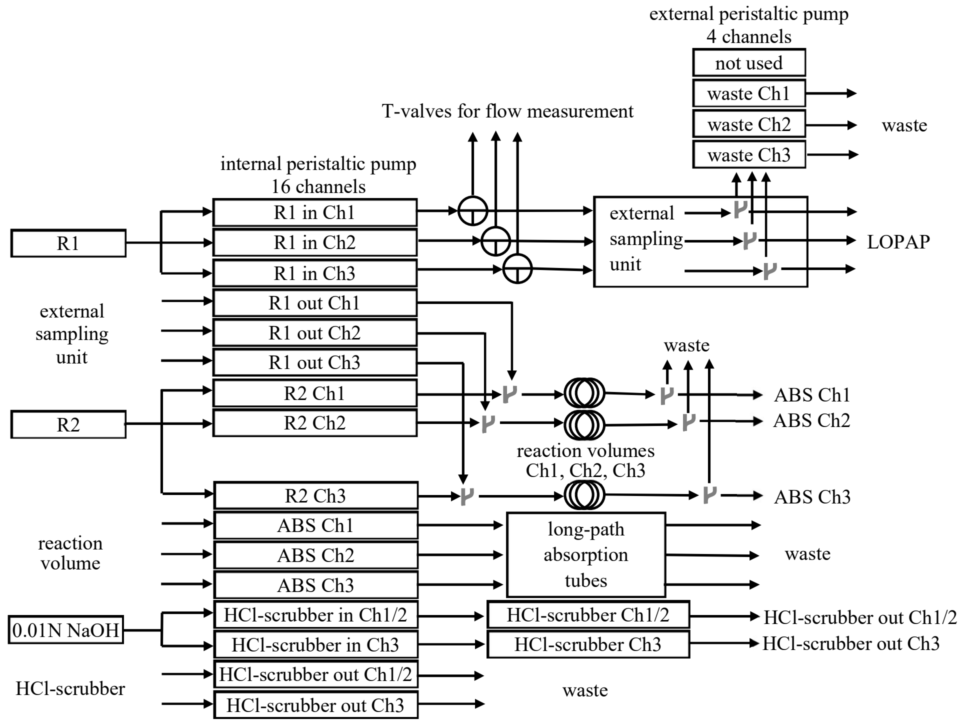

Figure S2: Liquid flow scheme of the 3-channel LOPAP.

Section S3: REA data logging and valve controlling software – PyREA

PyREA is a custom developed program, which controls the REA system. It acquires raw data from the ultrasonic anemometer, processes the raw samples, switches the magnetic valves and writes required result files for data analysis / post processing. In Figure S3 the processing of the anemometer data by the PyREA software is schematically shown. PyREA has a basic browser based graphical user interface (GUI), which displays monitoring information (graphical representations of the current anemometer data, log messages etc.), allows to control the primary configuration parameters of the REA system (e.g. measurement cycle switching times) and enables manual direct control of the valves.

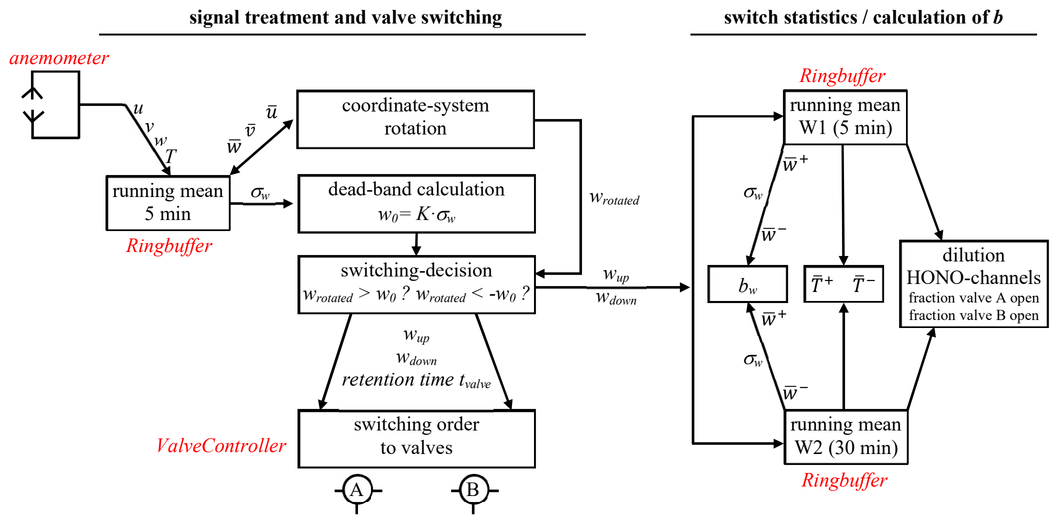

Figure S3: Processing of the anemometer data by the PyREA software.

PyREA is primarily implemented in the Python programming language and is designed for the operation on modern single board computer systems, which feature general-purpose input-output (GPIO) connectors to be used as low-level control interface to the valves. PyREA uses a small C++ hardware abstraction layer module for the GPIO access.

The browser-based GUI uses a simple client/server structure: A HTML5 / Java Script frontend presentation layer connects asynchronously to a small backend server, which provides structured access to functions to change the state of the REA processing and to retrieve monitoring data for the frontend display. Since the system is operated in a protected network environment, the backend access is currently not protected against unwanted access or encrypted.

PyREA has a modular structure. The individual modules exchange messages and data with a simple listener / observer pattern. The primary modules of the system are:

Anemometer

The anemometer module communicates with the ultrasonic anemometer hardware. It listens to incoming raw data samples from the anemometer (10 Hz), annotates them with a timestamp and distributes the samples to the other system components (which are registered observers for “new anemometer data” events).

Ringbuffer

Data is accumulated in ringbuffer structures, which are queue-like data structures with a fixed length. Data samples (scalars and vector samples with multiple components) can be put into the ringbuffer structures at the front. If the structure is full, the oldest sample is removed from the list of samples, leading to data samples, which are “pushed through” the ring buffer. The ringbuffer implementation allows to perform a set of mathematical operations (e.g. average, sum) on the stored samples with reduced computing time. The temporal averaging operations (5 min, 30 min averages) described in section 4.1 of the article are performed with the ringbuffer module.

Valve / Valve Controller

The ValveController module is the core of the REA system: It takes the raw anemometer samples, performs the operations of the REA method described in section 2 of the article and generates valve switch decisions from the measured anemometer data and the configuration of the REA system. The switch decisions are sent to the instances of the valve class, which is the interface to the actual valve hardware.

Filewriter / Logger

Tabular data (e.g. averaged wind data, REA parameters) have to be retained by the REA system for the data post processing and data analysis steps required to calculate actual HONO fluxes. Such tabular data are written to structured text files (CSV), which also can be compressed with a data compression utility. System events (e.g. parameter changes, manual valve switching) are written to a log file.

GUI / GUI Backend

The frontend component of the GUI is written in Java Script with *Jquery* (https://jquery.com/). It uses *Bootstrap* (https://getbootstrap.com/) for layout and graphical user interface widgets (buttons etc.), and *Plotly* (https://plotly.com/) components for the graphical representation of data. The GUI backend uses the *CherryPy* (https://cherrypy.github.io/) framework as webserver and to provide the connectivity to the frontend and as a webserver.

Operation Modes

The primary operation mode of PyREA is the *REA* mode, which controls the valves according to the vertical wind and the deadband as described in the article. Commonly, the operation modes are alternated with customizable periods (*auto-REA* mode) for long-term measurements. Besides this automated scheme, the operation mode and the valve state can be also switched manually. Additionally, multiple test modes are available: The internal data processing and switching logic can be tested with simulated / synthetic anemometer data and with pre-recorded anemometer data provided as CSV file. The valves can also be switched in a simple pattern with configurable opening and closing periods to test for deviations between the LOPAP channels, effects of switching frequencies and dilution as described in the article.

Section S4: Laboratory set-up

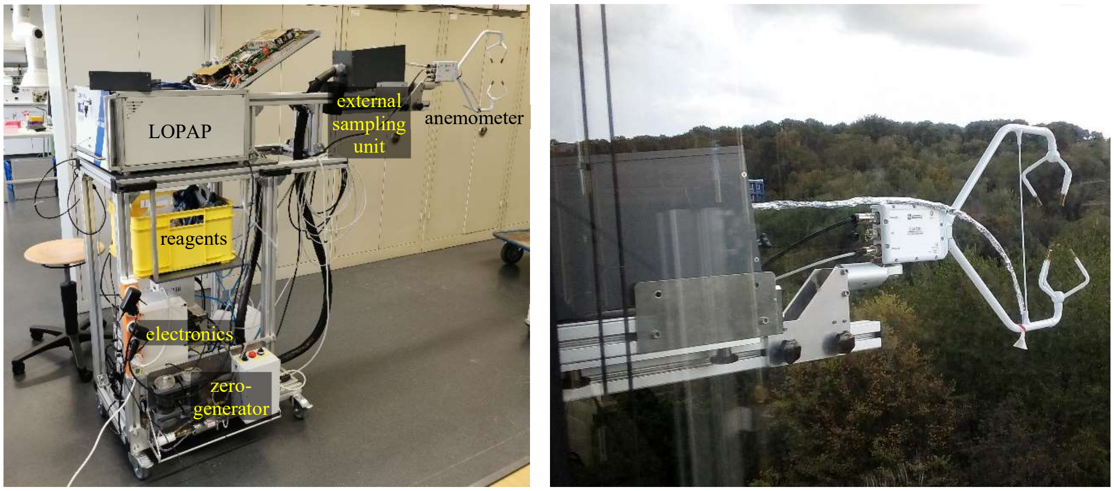

Figure S4: Laboratory set-up: left: moveable laboratory rack; right: external sampling unit and anemometer in the atmosphere in front of the facade of the laboratory.

Section S5: Field set-up

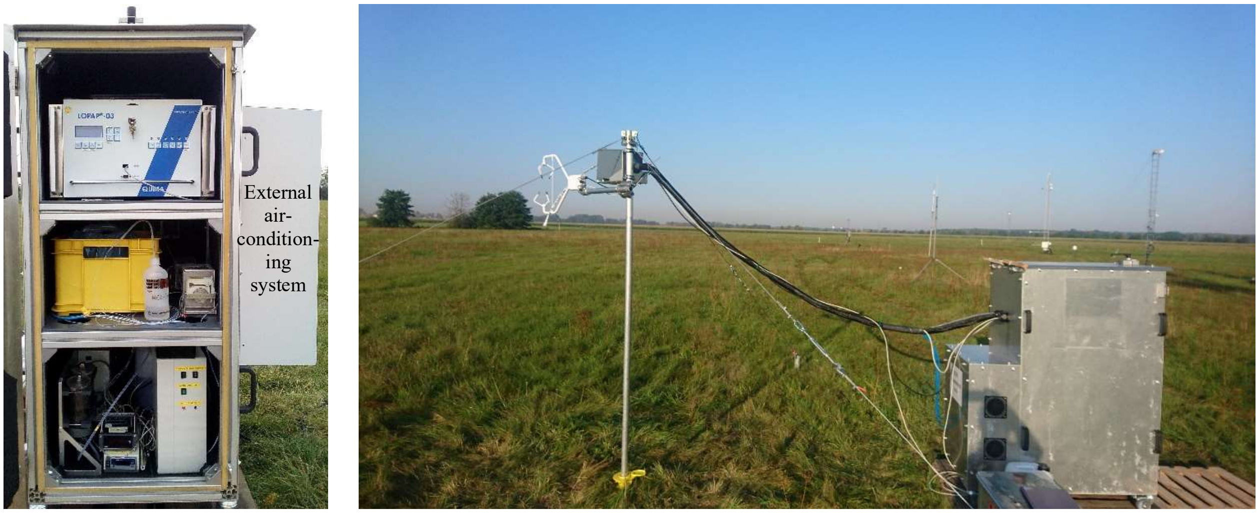

Figure S5: Left: opened field rack with LOPAP (upper level), reagents and external peristaltic pump (middle level), electronics and zero-generator (lower level) and external air conditioning system (side); right: field rack with mast for the external sampling unit and the anemometer.

Section S6: Experimental data of the field campaign in Melpitz

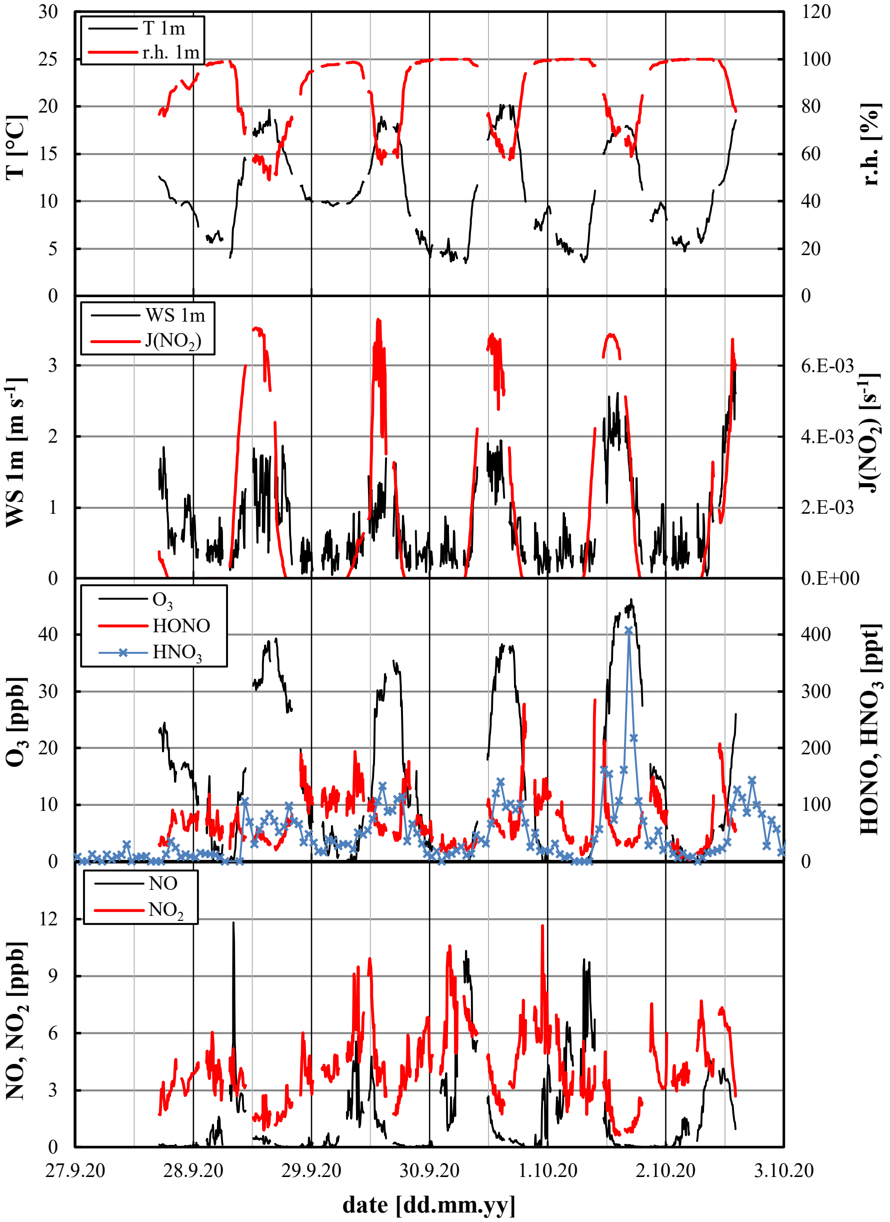

Figure S6: Simultaneous experimental data (5 min averages) of the campaign in Melpitz. For HNO3 all experimental data (1 h) are shown. 80

Section S7: Assessment of turbulence production

Similar to other micrometeorological flux measurements, the quality of the calculated fluxes in the REA method depends on the fulfilment of the underlying assumptions, which are for example, homogeneity of the surface and well-developed turbulence. The state of turbulence in the atmospheric boundary layer is characterized by the friction velocity and the buoyancy flux. The friction velocity describes the turbulent transport of momentum, whereas the buoyancy flux describes the turbulent transport of heat from the surface by convection, both of which represent mechanisms of turbulence production. For the concepts and equations of both measures, see for example Stull (1988).

During the field campaign in Melpitz, wind speeds and friction velocities were often quite low, which indicates low mechanical turbulence production by wind shear. The maximum of the friction velocity occurred in the afternoon, whereas during night, the friction velocity was typically below 0.05 m s^{-1} (Figure S7). The buoyancy flux showed a maximum around noon, started to decrease already at 15:00 h and often became negative already in the late afternoon. Negative buoyancy fluxes represent negative gradients of temperature and a stable atmospheric layering, caused by the radiative cooling of the surface or extraction of heat from the air by evaporation. When the buoyancy flux is negative, turbulence can only be produced mechanically.

Taking these two processes of turbulence production into account, well developed turbulence can be assumed only from about 08:30 h to 16:30 h during the field campaign in Melpitz. In the evenings and during the nights, turbulence intensity was much weaker and higher relative errors of the calculated HONO fluxes are expected, which are not quantifiable though.

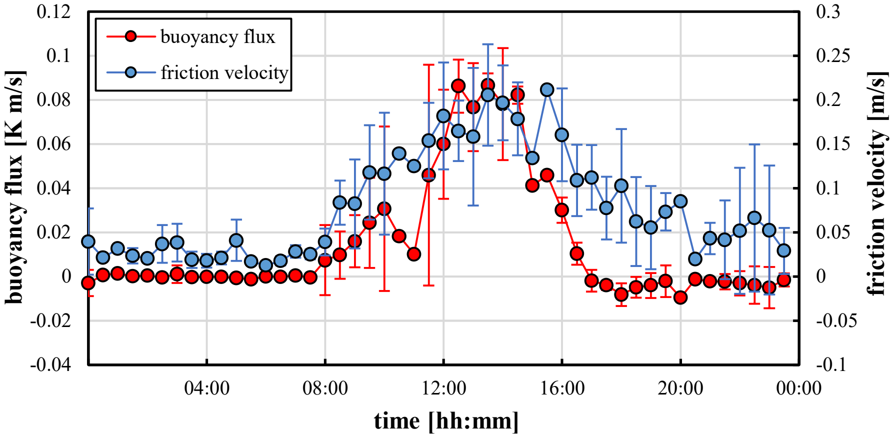

Figure S7: Mean buoyancy flux and friction velocity during the last five days of the campaign. Error bars represent standard deviations.

References Supplement:

Stull, R. B.: An introduction to boundary layer meteorology, Kluwer Academic Publishers, Dordrecht/Boston/London, 1988.