UNIVERSITÄT BAYREUTH

Abt. Mikrometeorologie

Application of a multi-step error filter for post-processing of atmospheric flux and meteorological basic data

Johannes Lüers

Florian Detsch

Peng Zhao

Work Report

No. 58

Bayreuth, June 2014

Arbeitsergebnisse, Universität Bayreuth, Abt. Mikrometeorologie
Work Reports, University of Bayreuth, Dept. of Micrometeorology
Print, ISSN 1614-8916; Internet, ISSN 1614-8924

http://www.bayceer.uni-bayreuth.de/mm/
https://epub.uni-bayreuth.de/view/series/Arbeitsergebnisse_=2F_Universit=E4t_Bayreuth,_Abteilung_Mikrometeorologie.html

Eigenverlag: Universität Bayreuth, Abt. Mikrometeorologie
Vervielfältigung: Druckerei der Universität Bayreuth
Herausgeber: Prof. Dr. Thomas Foken

Universität Bayreuth, Abteilung Mikrometeorologie
D-95440 Bayreuth

Content

1 Introduction

The R software package described in this working report includes a new multi-step error filter, called *MSEF*, in addition to the quality checks implemented in the TK3 software or any other eddy-covariance software package. Optionally, a gap-filling routine can be applied after having checked the data in order to fill small measuring gaps by linear interpolation.

The TK3 eddy-covariance software package (Mauder and Foken 2011) as a subsequent version of 'Turbulenzknecht' (Foken 1999) and the TK2 software (Mauder and Foken 2004) calculates high-resolution eddy-covariance data. These, in turn, reflect micrometeorological energy exchange processes at the surface which are quality assured by applying a set of corrections and quality tests. The binary output data set contains 30-minute means of the high-frequency measured and calculated parameters.

MSEF uses these output files as input data in order to apply further quality checks including reasonable consistency limits, status-or-threshold values (*STV*) like the diagnostic *AGC* value, and a set of statistical tests. Note that all the included checking parameters can be interactively adjusted by the user according to individual needs. After the facultative filling of small gaps, the software generates various output files that are corrected for low-quality data and outliers.

Figure 1.1 illustrates the whole procedure from raw data measurements and calculations via post-processing by *MSEF* to adjusted and gap-filled high-quality data sets. Approaches to fill longer gaps are currently being tested and will be discussed in a subsequent working report.

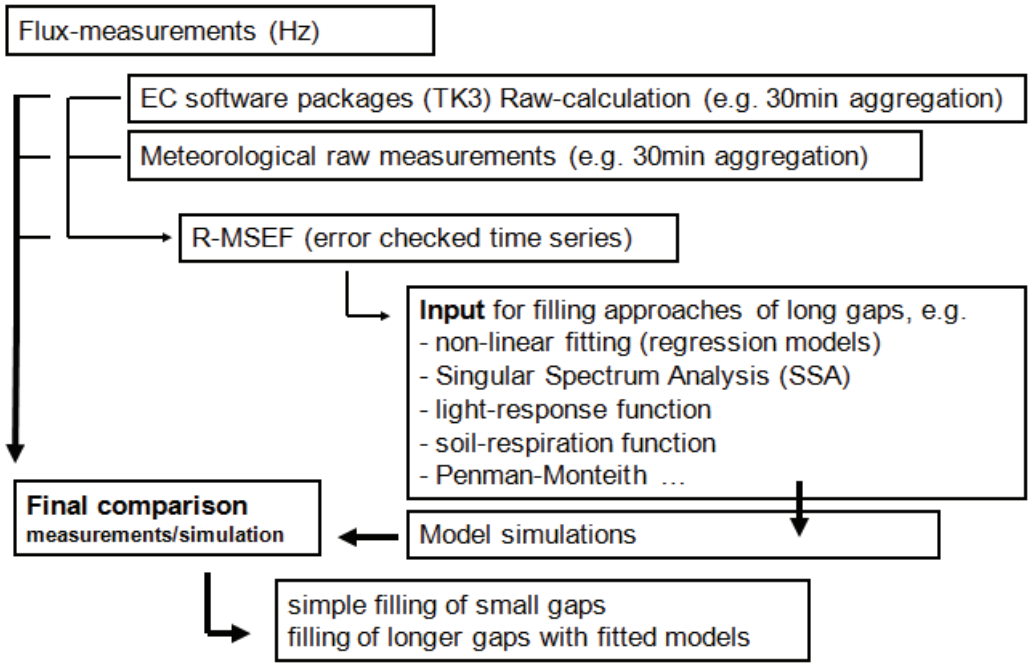

Figure 1.1: Processing chain for TK3-based raw output data including multi-step error filtering and optional gap-filling as described herein (Lüers 2011).

2 General structure of the *MSEF* software

In the following, we'll primarily refer to the TK3 software package. However, *MSEF* is conceived to run with input data from any other eddy-covariance software packages as well. Based on these eddy-covariance input data sets, *MSEF* executes a set of work steps in order to eliminate logically inconsistent values and statistical outliers. The procedure can be subdivided as follows (Zhao and Lüers 2012):

1. Incorrect values from the TK3 software are inherited
2. All measured parameters (e. g. horizontal and vertical wind speed) as well as the corresponding deduced parameters (e. g. covariances and fluxes) are checked with respect to user-defined reasonable consistency limits
3. Optionally, quality flags are applied in order to check the quality of the calculated 30-minute flux values (Foken and Wichura 1996; Foken *et al.* 2004) and eliminate measurements of certain quality classes
4. Optionally, a status-or-threshold value (*STV*), e. g. information from a present weather detector (*PWD*) (influence of rain, fog and snow) or as a diagnostic value like the combined *AGC* value, is applied in order to separate or mark certain periods
5. For each directly measured parameter and for all subsequently derived parameters, absolute deviation is calculated followed by a quantile check and a standard deviation filter to detect major outliers
6. Optionally, accrued small gaps of one or two inconsistent values can be filled using linear interpolation

Figure 2.1 summarizes the multi-step error filter as an important intermediate step between TK3-based data containing a certain amount of low-quality measurements on the one hand and high-quality data that can be used for further filling approaches of bigger gaps on the other hand.

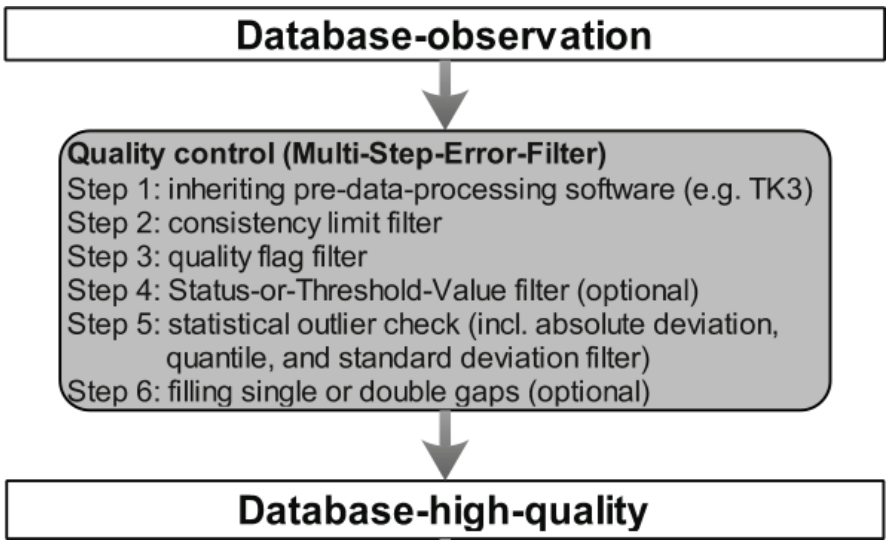

Figure 2.1: Transformation of a partly inconsistent TK3 data set to a high-quality measurement series by applying a multi-step error filter with optional gap-filling (Zhao and Lüers 2012).

3 Important components of the *MSEF* software

To get the program started and work correctly, the following section of this working report contains a technical overview followed by a detailed description focussing on the relevant components of the program. Users that aren't interested in too detailed knowledge about the software's specifics are referred to Chapter 4 that contains instructions to quickly launch the program.

3.1 Technical aspects, required folder structure and files

The *MSEF* software package is written in R (R Development Core Team 2011) and currently tested on a Windows 7 platform. As long as the R Runtime Environment is installed on the local computer, no further packages and thus no connection to the internet is necessary for a proper execution.

The program's root directory, referred to as *working directory* within the R script, contains the current version of the main R file (*MSEF_*.r*) as well as several subfolders, namely:

**config*	→	configuration file *config.csv* and its appending manual *config_info.doc* (cf. Chapter 3.2)
**functions*	→	file *functions.r* containing the *SpikeCheck* function (cf. Chapter 3.3)
**input*	→	input data sets (**result*.csv* and **qa_qc*.csv*)
**output*	→	location of the different output files after execution

Note that the *output* folder, if not existent, will be generated automatically when running the program. However, if one of the other folders or one of their corresponding files cannot be found by the R routine, an error message will be displayed and the program won't run correctly.

3.2 Configuration file

The configuration file (**config**config.csv*) is one of the program's main features because it enables the user to modify certain settings according to individual demands. That includes for each parameter:

- Reasonable minimum and maximum consistency limits
- Window sizes, limits and factors for the calculation of statistical parameters (absolute deviation, quantile check, standard deviation)
- Optional settings concerning gap-filling of single and double gaps
- Use of *AGC* value, footprint derived land use factor (*FLF*) and quality flags

Before running the program, the user just has to open the configuration file, adjust the default settings to individual needs if necessary, and save the alterations. During the next run, the program will automatically consider these user-defined settings when calling the *SpikeCheck* function.

For a detailed description of the single setting parameters, compare the appending manual (**config**config_info.doc*) or see Table 3.1. The default settings are shown in Tables 3.2a and 3.2b.

Table 3.1: Summary and description of the adjustable columns in the configuration file.

Column	Description
item	Column name in the eddy-covariance input data file
decimals	Number of decimal places in the output data files
pretreat.only	If TRUE, only reasonable consistency limits (*min*, *max*) will be checked
min	Lower reasonable consistency limit
max	Upper reasonable consistency limit
window	Window size for the calculation of absolute deviation
d.window	Window size for the quantile check
d.quantile	Quantile (e. g. 0.9 for 90% quantile)
d.quantile.factor	Quantile factor
sd.factor.lower	Multiplier for the calculation of the lower standard deviation
sd.factor.negative	If TRUE, the lower standard deviation is allowed to fall below 0
sd.factor.upper	Multiplier for the calculation of the upper standard deviation
sd.window	Window size for the calculation of standard deviation
fill.2gap	If TRUE, double gaps will be filled by linear interpolation
fill.1gap	If TRUE, single gaps will be filled by linear interpolation
gap.window	Window size for gap-filling
AGC.switch	If TRUE, *AGC* values will be checked
AGC.col	*AGC* column in the input data
AGC.min	Minimum *AGC* value
AGC.max	Maximum *AGC* value
landuse.switch	If TRUE, *FLF* values will be checked
landuse.col	FLF column in the input data
landuse.min	Minimum *FLF* value
landuse.max	Maximum *FLF* value
flag.switch	If TRUE, quality flags will be checked
flag1.col	1st flag column in the input data (required if *flag.switch = TRUE*)
flag2.col	2nd flag column in the input data (optional)
flag3.col	3rd flag column in the input data (optional)
flag.min	Minimum flag value
flag.max	Maximum flag value

Table 3.2a: Default settings of the adjustable columns in the configuration file (1st part).

item	decimals	pretreat.only	min	max	window	d.window	d.quantile	d.quantile.factor	sd.factor.lower	sd.factor.negative	sd.factor.upper	sd.window	fill.2gap	fill.1gap	gap.window
u[m/s]	2	FALSE	0	Inf	7	21	0.9	2.3	3	FALSE	3	29	TRUE	TRUE	5
v[m/s]	2	TRUE	0	0	7	21	0.9	2.3	3	TRUE	3	29	TRUE	TRUE	5
w[m/s]	2	FALSE	-0.15	0.25	7	21	0.92	1.8	3	TRUE	3	29	TRUE	TRUE	5
Ts[°C]	2	FALSE	-10	35	7	21	0.95	2.1	3	TRUE	3	29	TRUE	TRUE	5
a[g/m³]	2	FALSE	0	25	7	21	0.96	1.8	2.7	FALSE	2.7	29	TRUE	TRUE	5
CO2[mmol/m³]	2	FALSE	5	30	7	21	0.96	1.8	2.7	FALSE	2.7	29	FALSE	FALSE	5
Cov[vw]	7	TRUE	-0.4	0.4	7	21	0.92	1.8	3	TRUE	3	29	TRUE	TRUE	5
Cov[uw]	7	TRUE	-0.4	0.4	7	21	0.92	1.8	3	TRUE	3	29	TRUE	TRUE	5
Cov[wTs]	7	TRUE	-0.4	0.4	7	21	0.92	1.8	3	TRUE	3	29	TRUE	TRUE	5
Cov[wa]	7	TRUE	-0.4	0.4	7	21	0.92	1.8	3	TRUE	3	29	TRUE	TRUE	5
Cov[wCO2]	7	TRUE	-0.4	0.4	7	21	0.92	1.8	3	TRUE	3	29	TRUE	TRUE	5
dir[°]	2	TRUE	0	360	NA	NA	NA	NA	NA	TRUE	NA	NA	TRUE	TRUE	5
ustar[m/s]	2	FALSE	0	2	7	21	0.96	2	2.5	TRUE	3.5	29	TRUE	TRUE	5
HTs[W/m²]	2	FALSE	-200	300	7	21	0.95	1.8	2.5	TRUE	2.5	29	TRUE	TRUE	5
LvE[W/m²]	2	FALSE	-100	500	7	21	0.95	1.7	2.5	TRUE	2.5	29	TRUE	TRUE	5
z/L	4	TRUE	-20	20	NA	NA	NA	NA	NA	TRUE	NA	NA	TRUE	TRUE	5
FCstor[mmol/m²s]	7	FALSE	-0.1	0.1	7	21	0.95	1.3	2.5	TRUE	2.5	29	TRUE	TRUE	5
NEE[mmol/m²s]	7	FALSE	-0.1	0.1	7	21	0.95	1.3	2.5	TRUE	2.5	29	TRUE	TRUE	5

Table 3.2b: Default settings of the adjustable columns in the configuration file (2nd part).

item	AGC.switch	AGC.col	AGC.min	AGC.max	landuse.switch	landuse.col	landuse.min	landuse.max	flag.switch	flag1.col	flag2.col	flag3.col	flag.min	flag.max
u[m/s]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
v[m/s]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
w[m/s]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
Ts [°C]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
a[g/m³]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
CO2 [mmol/m³]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
Cov[ww]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
Cov[uw]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
Cov[wTs]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
Cov[wa]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
Cov[wCO2]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
dir[°]	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
ustar[m/s]	FALSE	NA	0	50	FALSE	NA	80	Inf	TRUE	statflag_ustar	itcflag_u	itcflag_w	1	6
H[sTs][W/m²]	FALSE	NA	0	50	FALSE	NA	80	Inf	TRUE	statflag_wTs	itcflag_w	itcflag_Ts	1	6
LvE[W/m²]	FALSE	NA	0	50	FALSE	NA	80	Inf	TRUE	statflag_wa	itcflag_w	NA	1	6
z/L	FALSE	NA	0	50	FALSE	NA	80	Inf	FALSE	NA	NA	NA	1	6
FCstor[mmol/m²s]	FALSE	NA	0	50	FALSE	NA	80	Inf	TRUE	statflag_wco2	itcflag_w	NA	1	6
NEE[mmol/m²s]	FALSE	NA	0	50	FALSE	NA	80	Inf	TRUE	statflag_wco2	itcflag_w	NA	1	6

3.3 Function file

The function file (**functions\function.r*) is the very part of the routine that's responsible for the multi-step error filtering. It's automatically invoked when running the *MSEF* software package and incorporates the default and/or user-defined settings from the configuration file. Figure 3.1 shows the typical procedure of the so-called *SpikeCheck* function for both a *pretreat.only* parameter (left branch, e. g. wind direction, all covariances) (cf. Table 3.1) and a parameter that should be checked qualitative and statistically including optional gap-filling (right branch, e. g. horizontal and vertical wind speed, absolute humidity).

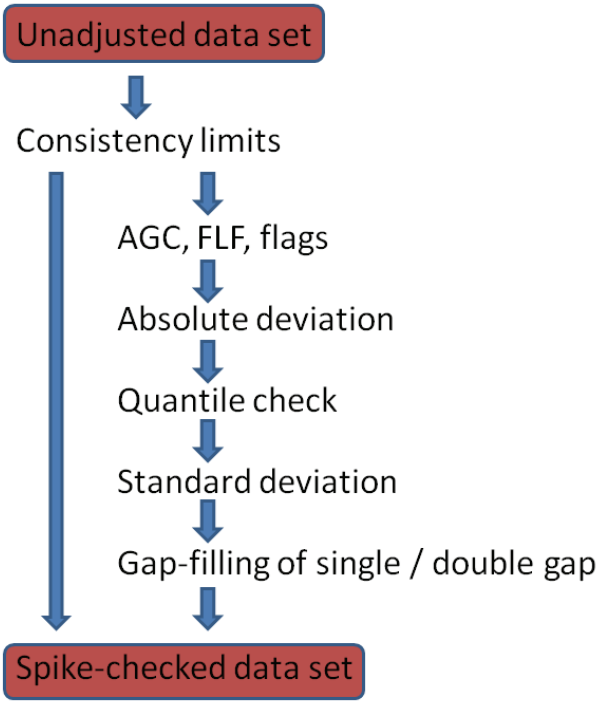

Figure 3.1: Typical procedure of the so-called *SpikeCheck* function for a certain parameter without (left branch) and with *STV* and statistical filtering (right branch).

3.3.1 Consistency limits

At first, reasonable minimum and maximum consistency limits are revised according to the user-defined columns *min* and *max* in the configuration file. Inconsistent values that exceed these reasonable limits are rejected and replaced by *NA* (cf. Figure 3.2).

```
54 ### 1 Reasonable consistency limits
55 spikecheck <- data.frame(ifelse(x < r.min | x > r.max | is.na(x), NA, x))
56 colnames(spikecheck) <- "threshold"
```

Figure 3.2: Rejection of values that exceed reasonable consistency limits.

Those parameters that needn't be statistically checked are already finished yet. They're referred to as *pretreat.only* parameters in the configuration file, meaning that no further qualitative and statistical checks should be applied after having tested the consistency limits (cf. Figure 3.3).

```
58 if (pretreat.only) {
59   spikecheck$threshold <- round(spikecheck$threshold, digits=n.dec)
60   return (spikecheck)
61 } else {
```

Figure 3.3: Return of *pretreat.only* parameters after having checked reasonable consistency limits.

3.3.2 *AGC*, *FLF*, flags

For any other parameter, a quality control in the form of a *STV* filter follows that optionally checks *AGC* and *FLF* values as well as quality flags. Note that none of these filters is activated by default except for the quality flags connected to *ustar*, *HTs*, *LvE*, *FCstor* and *NEE* (cf. Table 3.2b). The user has to modify *AGC.switch*, *landuse.switch* and *flag.switch* including related columns in order to enable additional quality checks. Again, if a certain value doesn't meet the defined quality settings, it is rejected and replaced by *NA* (cf. Figure 3.4).

```
63 ### 2 AGC value
64 if(agc.switch) {
65   spikecheck$AGC_filtered <- ifelse((agc > agc.max | agc < agc.min) & !is.na(agc),
66                                     NA, spikecheck$threshold)
67 } else {
68   spikecheck$AGC_filtered <- spikecheck$threshold
69 }
```

Figure 3.4: Rejection of values that exceed AGC thresholds.

As for the flags, the number of revised output columns that's stored in the detailed output file (cf. Chapter 3.4.8) depends on the setting parameters *flag.min* and *flag.max*. The quality flags are ranked from 1 to 9, with the prior one representing the highest and the latter representing the lowest measurement quality. By default, *flag.max = 6* for all parameters. Consequently, *MSEF* will perform a loop and firstly reject measured values with corresponding flag values of 9. Afterwards, measurements with flag values of 8 or 9 are thrown out and finally, values with related flag values of 7, 8 or 9 are replaced by *NA*. An example for the output structure of these flag operations is shown in Table 3.1. As demands on measurement quality increase, so does the number of rejected values due to insufficient quality.

Table 3.1: Exemplary quality check of subsequently measured values of *HTs* based on associated flags (settings: *flag.switch = TRUE*, *flag1.col = statflag_wTs*, *flag2.col = itcflag_w*, *flag3.col = itcflag_Ts*, *flag.min = 1*, *flag.max = 6*).

HTs[W/m²]	HTs[W/m²]_QCflag9	HTs[W/m²]_QCflag8	HTs[W/m²]_QCflag7
4.6423683	4.6423683	NA	NA
1.8053464	1.8053464	NA	NA
1.899873	1.899873	NA	NA
-0.7191497	NA	NA	NA
1.2077483	1.2077483	NA	NA
4.5456438	4.5456438	4.5456438	NA
13.9503574	13.9503574	13.9503574	13.9503574
-4.1969962	-4.1969962	NA	NA
-6.2815061	-6.2815061	-6.2815061	NA
7.75246	7.75246	7.75246	7.75246

3.3.3 Statistical parameters

After the quality control, the statistical part of the function begins by computing absolute deviation, quantile limits and standard deviation according to the default or user-defined settings. Values that don't fit within these statistical limits are rejected and replaced by *NA* (cf. Figure 3.5), resulting in a high-quality version of the input data set that's corrected for outliers and measurements beyond reasonable thresholds.

```
148   spikecheck$sd <- ifelse(((!is.na(spikecheck$sd_u)) & spikecheck$quantile > spikecheck$sd_u) |
149               ((!is.na(spikecheck$sd_l)) & spikecheck$quantile < spikecheck$sd_l),
150               NA, spikecheck$quantile)
```

Figure 3.5: Rejection of values that exceed lower and upper standard deviation limits.

3.3.4 Gap-filling of single and double gaps

As a final step, the function will automatically fill accrued single or double gaps by linear interpolation. However, this is an optional step, meaning that the user can switch on and off the gap-filling part by altering the particular columns in the configuration file.

At first, double gaps are filled by default as displayed in Figure 3.6a. In case of *gap.window = 5*, the former missing value is replaced by the mean of the two preceding values and the first subsequent value, whereas the latter is replaced by the mean of the last valid value before the gap and the two subsequent values.

Single gaps are filled afterwards as displayed in Figure 3.6b. Again for *gap.window = 5*, the missing value is replaced by calculating the mean of the two preceding and the two subsequent values.

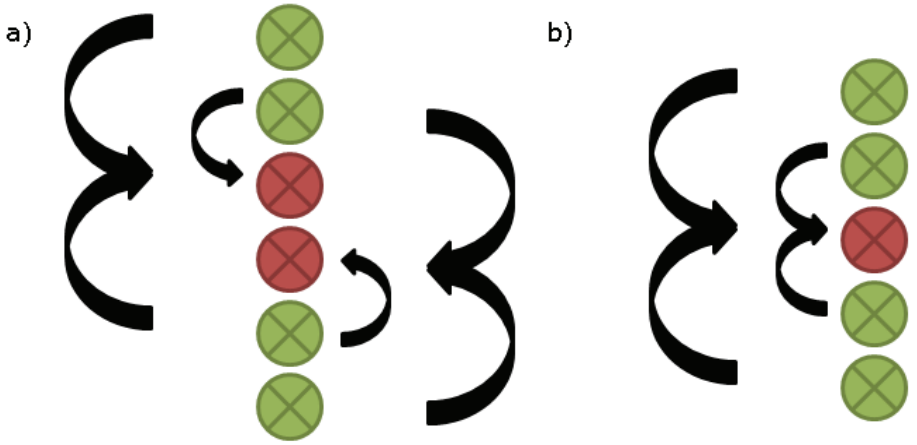

Figure 3.6: Gap-filling of a) double gaps and b) single gaps, both in case of *gap.window = 5*.

Figure 3.7 visualizes the function's whole process from untreated eddy-covariance data to reasonably and statistically rectified data, and illustrates the facultative application of gap-filling by linear interpolation as theoretically shown in Figure 3.6. The red-coloured peak values of *u* lie outside of either reasonable or statistic consistency limits and are consequently rejected. Afterwards, the emerging single gaps are filled by linear interpolation, which is also indicated by red-coloured curve sections.

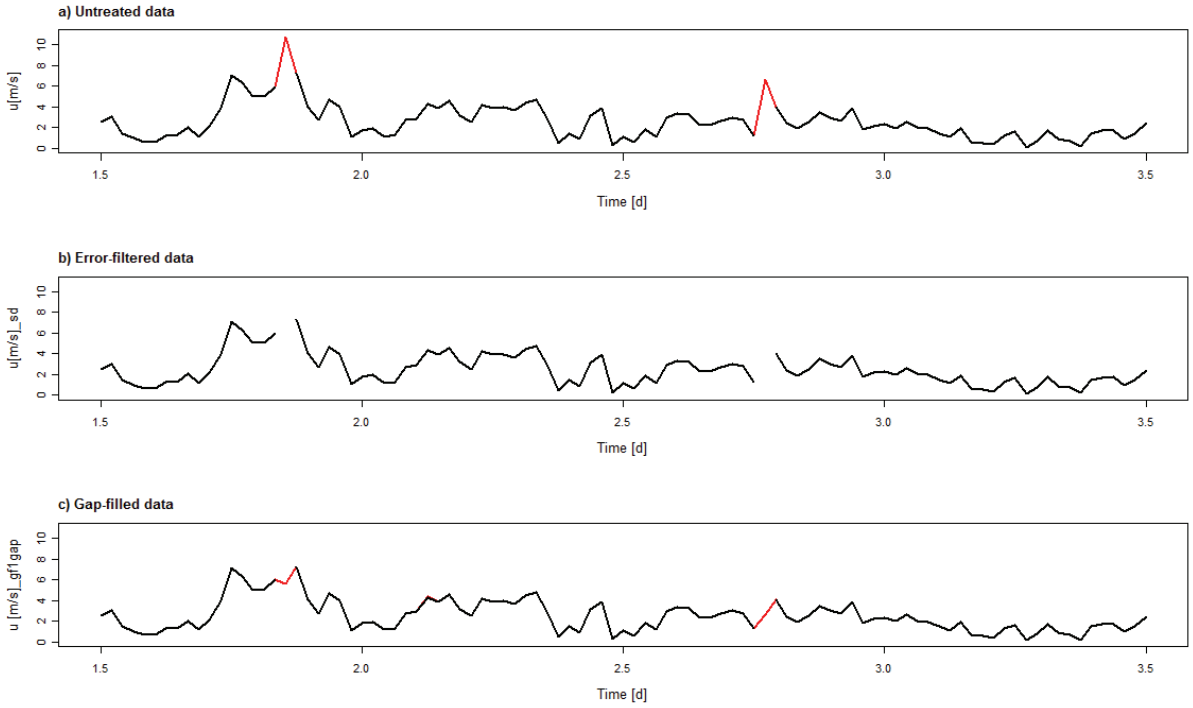

Figure 3.7: Visualization of a typical function run for the vertical wind speed *u* over time from a) untreated eddy-covariance data to b) error-filtered data. The thus arising single gaps are c) filled by linear interpolation according to Figure 3.6.

With regard to the cause of rejection in Figure 3.7a, exceeded minimum or maximum values can be ruled out as reasonable consistency limits of the vertical wind speed *u* range from 0 to ∞ by default (cf. Table 3.2a). Indeed, the red marked values crossed statistical limits, thus generating *NA* values. As displayed in Figure 3.8, the first red peak at ~ 1.8 d exceeded the upper limit of standard deviation (grey line) and has therefore been replaced by *NA*. The second red peak would have surely been kicked out for the same reason. However, it has been rejected earlier within the function because the absolute deviation (yellow line) surpassed the quantile limit (blue line) at that certain point in time.

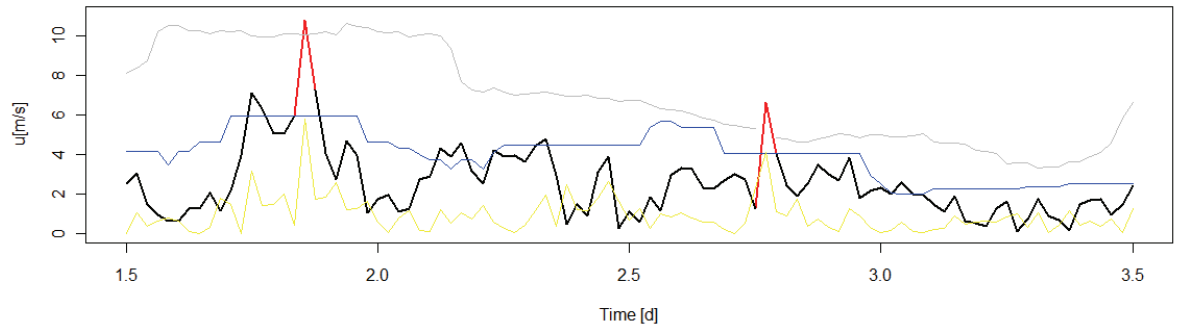

Figure 3.8: Vertical wind speed *u* over time with related upper limit of standard deviation (grey), absolute deviation (yellow) and quantile check (blue).

3.4 *MSEF* file

The *MSEF* file as the main R routine controls all actions performed by the software, thus serves as a kind of steering wheel. It's responsible for reading in the eddy-covariance data, performing the described multi-step error filtering and saving the thereby generated output files. Figure 5 illustrates the general structure of this script.

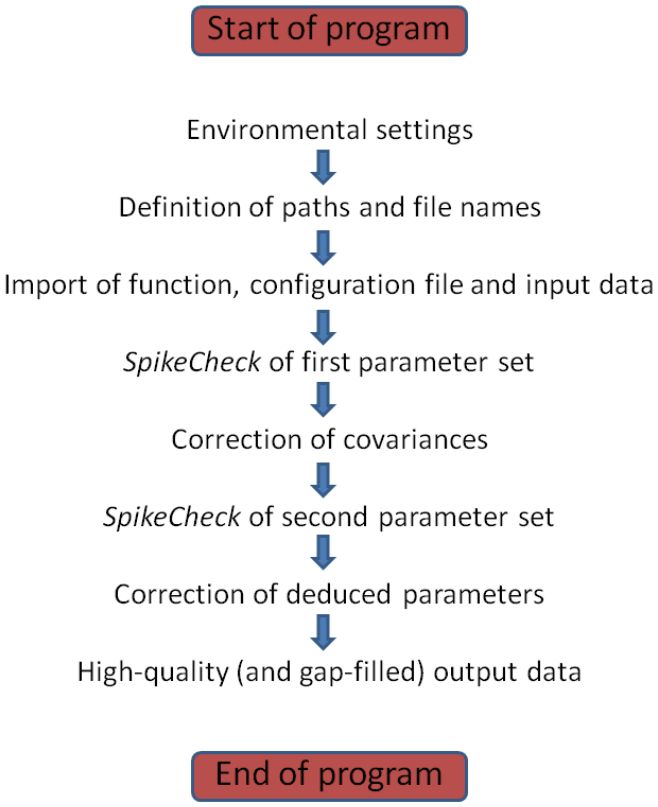

Figure 5: General structure of the *MSEF* routine.

3.4.1 Environmental settings

At the beginning, a couple of environmental settings are performed including workspace clearance, prohibition of scientific notation and the definition of the *working directory* (cf. Figure 6). The latter represents the absolute path to the program's root directory that contains the required subfolders (cf. Chapter 3.1) and has to be adjusted manually if the program is transferred to another computer.

```
### Environmental settings

# workspace clearance
rm(list = ls(all = TRUE))

# prohibition of scientific notation
options(scipen=10)

# path to working directory
wd <- "C:\\Users\\Climatology\\GAPFILL"
# set working directory
setwd(wd)
```

Figure 6: Environmental settings.

3.4.2 Definition of paths and file names

Subsequently, all relative paths (meaning subfolders) and filenames are defined so that the program 'knows' where to look for required data and store the output files, respectively. This includes the configuration file, the function file and the eddy-covariance input data (cf. Figure 7) as well as the filenames for output storage. The filenames of the input data and the corresponding QA/QC file are the second and last lines of code in the whole script that have to be adjusted manually in order to run the program with different data sets.

```
### Definition of paths and file names

# subfolder with function file
path.func <- "functions"
# function file
file.func <- "function.r"

# subfolder with configuration file
path.config <- "config"
# configuration file
file.config <- "config.csv"

# subfolder with input file
path.in <- "input"
# input files
file.in <- "KemaU_result_recomb_all.csv"
file.qa_qc <- "KemaU_QA_QC_recomb_all.csv"
```

Figure 7: Definition of paths and file names.

However, the QA/QC file is not mandatory as it contains information about quality flags that needn't implicitly be considered by the program. Note that in order to run the program correctly without using a QA/QC file, either the *flag.switch* column in the configuration file must be disabled (*FALSE*) or the following columns *flag1.col*, *flag2.col* and *flag3.col* must be set to *NA*.

After having manually modified this very filename, the software automatically checks whether the previously specified folders and files exist in the given working directory, and an error message is displayed on-screen if a particular folder or file is missing. Additionally, an output folder will be created by default in case it doesn't already exist. The different output files are named dynamically and inherit the first three characters of the input data set, an abbreviation of the content stored in the particular file, and both date and time of generation. For instance, running the program with the input files shown in Figure 7 would produce three different output data sets named *Kem_long_201205151322.csv*, *Kem_short_201205151322.csv* and *Kem_na_201205151322.csv* (cf. Chapter 3.4.8).

3.4.3 Import of function, configuration file and input data

After that, the program starts the actual import of said files by reading in the stored information (cf. Figure 8). Like that, the measurement series optionally including quality flags, the user-defined settings from the configuration file and the *SpikeCheck* function are transferred to the *R Workspace* and thus can be used for further work steps.

```
### Loading functions and reading in data

# load function file
source(paste(path.func, "\\", file.func, sep=""))

# read configuration file
settings <- read.csv(paste(path.config, "\\", file.config, sep=""),
                     row.names=1, stringsAsFactors = FALSE)

# read input file
dat.in <- read.table(paste(path.in, "//", file.in, sep=""), header=TRUE,
                     sep=",", fill=TRUE, stringsAsFactors=FALSE, check.names=FALSE)
```

Figure 8: Import of function, configuration file and input data.

If the user desires to include quality flags from the QA/QC file, the software will automatically import relevant columns from the very same, and merge these columns into the input data (cf. Figure 9).

```
# read QA_QC input file
if (file.exists(paste(path.in, "//", file.qa_qc, sep=""))) {
  dat.qa_qc <- read.table(paste(path.in, "//", file.qa_qc, sep=""), header=T,
                          sep=",", fill=TRUE, stringsAsFactors=FALSE, check.names=FALSE)
  # subset of dat.qa_qc with relevant flags
  dat.flag <- cbind(dat.qa_qc$"statflag_ustar", dat.qa_qc$"statflag_wTs",
                    dat.qa_qc$"statflag_wa", dat.qa_qc$"statflag_wco2",
                    dat.qa_qc$"itcflag_u", dat.qa_qc$"itcflag_w", dat.qa_qc$"itcflag_Ts")
  colnames(dat.flag) <- c("statflag_ustar", "statflag_wTs", "statflag_wa", "statflag_wco2",
                          "itcflag_u", "itcflag_w", "itcflag_Ts")
  # attach flags to dat.in
  dat.in <- cbind(dat.in, dat.flag)
} else {
  print("No QA/QC file specified!")
}
```

Figure 9: Import of QA/QC input file.

3.4.4 *SpikeCheck* of first parameter set

The multi-step error filter is at first applied to the actually measured parameters, i. e. to *u*, *v*, *w*, *Ts*, *a*, CO_2 and *dir*, as well as to the related covariances *Cov[vw]*, *Cov[uw]*, *Cov[wTs]*, *Cov[wa]* and $Cov[wCO_2]$ (cf. Figure 10).

```
### Spikecheck for u, v, w, Ts, a, CO2, dir and several covariances

col.sp1 <- c("u[m/s]", "v[m/s]", "w[m/s]", "Ts[°C]", "a[g/m³]", "CO2[mmol/m³]", "dir[°]",
             "Cov[vw]", "Cov[uw]", "Cov[wTs]", "Cov[wa]", "Cov[wCO2]")

for (i in col.sp1) {
```

Figure 10: Looping through the first set of parameters.

For this purpose, both the measured values of the particular parameter and the associated settings from the configuration file are handed over to the *SpikeCheck* function. In addition, facultative *AGC*, *FLF* and/or flag values are transferred, too (cf. Figure 11). The function, in turn, performs the actual multi-step error filtering based on the delivered parameters and returns a revised version of the eddy-covariance input data (cf. Chapter 2.3). In addition, it optionally fills small gaps of one or two missing values by linear interpolation.

```
## call function SpikeCheck
spikecheck <- SpikeCheck(x = dat.in[, i],
                         y = i,

                         agc.switch = settings[i, "AGC.switch"],
                         agc = agc,
                         agc.min = settings[i, "AGC.min"],
                         agc.max = settings[i, "AGC.max"],

#                          lu.switch = settings[i, "landuse.switch"],
#                          lu = lu,
#                          lu.min = settings[i, "landuse.min"],
#                          lu.max = settings[i, "landuse.max"],

                         flag.switch = settings[i, "flag.switch"],
                         flag = flag,
                         flag.min = settings[i, "flag.min"],
                         flag.max = settings[i, "flag.max"],

                         n.dec = settings[i, "decimals"],
                         pretreat.only = settings[i, "pretreat.only"],
                         r.min = settings[i, "min"],
                         r.max = settings[i, "max"],
                         w = settings[i, "window"],
                         n.row = n.row,
                         d.w = settings[i, "d.window"],
                         d.quantile = settings[i, "d.quantile"],
                         d.quantile.factor = settings[i, "d.quantile.factor"],
                         sd.w = settings[i, "sd.window"],
                         sd.factor.lower = settings[i, "sd.factor.lower"],
                         sd.factor.negative = settings[i, "sd.factor.negative"],
                         sd.factor.upper = settings[i, "sd.factor.upper"],

                         fill.2gap = settings[i, "fill.2gap"],
                         fill.1gap = settings[i, "fill.1gap"],
                         gap.w = settings[i, "gap.window"])
```

Figure 11: Transfer of input data and user-defined settings to the *SpikeCheck* function. Variable *i* contains the current parameter (cf. Figure 10).

The adjusted data is merged after every single run of the loop, producing a data frame *dat.out* that contains all the columns generated throughout the filtering procedure.

3.4.5 Correction of covariances

Before processing the deduced parameters, all the covariances need to be tested for potential rejected values in the measured parameters. By performing the *SpikeCheck*, *NA* values are assigned to invalid measurements that previously contained a numeric value. As a logical consequence, covariances that are dependent on that certain *NA* value need to be set to *NA* as well (cf. Figure 12).

```
### Check covariances in dat.out for NA values in related parameters

# Logical vectors indicating NA values for u, v, w, Ts, a, CO2
na.u <- is.na(dat.out$"u[m/s]_sd")
na.v <- is.na(dat.out$"v[m/s]_threshold")
na.w <- is.na(dat.out$"w[m/s]_sd")
na.Ts <- is.na(dat.out$"Ts[°C]_sd")
na.a <- is.na(dat.out$"a[g/m³]_sd")
na.CO2 <- is.na(dat.out$"CO2[mmol/m³]_sd")

# Setting related covariances to NA
dat.out$"Cov[vw]_threshold" [unique(c(which(na.v), which(na.w)))] <- NA
dat.out$"Cov[uw]_threshold" [unique(c(which(na.u), which(na.w)))] <- NA
dat.out$"Cov[wTs]_threshold" [unique(c(which(na.w), which(na.Ts)))] <- NA
dat.out$"Cov[wa]_threshold" [unique(c(which(na.w), which(na.a)))] <- NA
dat.out$"Cov[wCO2]_threshold" [unique(c(which(na.w), which(na.CO2)))] <- NA
```

Figure 12: Rejection of covariance values based on previously rejected values for *u*, *v*, *w*, *Ts*, *a* and CO_2.

3.4.6 *SpikeCheck* of second parameter set

In the following step, multi-step error filtering is applied to the calculated parameters *ustar*, *HTs*, *LvE*, *z/L*, *FCstor* and *NEE* including facultative gap-filling (cf. Chapter 3.4.4). As this happens analogous to the first set of parameters, we yield up illustrating this step.

3.4.7 Correction of deduced parameters

Again, these quality-controlled measurement series have to be checked for logical conflicts with previously rejected, associated values. This procedure is based on the same principle as shown in Figure 12.

For example, *Cov[vw]* is set to *NA* at a certain instant of time because it either exceeded the reasonable consistency limits or *v* respectively *w* were rejected before. Consequently, *ustar* is set to *NA* for that certain point in time as well, as these two parameters (*Cov[vw]* and *ustar*) are directly linked to each other. The thus rejected *ustar* value now directly affects *z/L*, turning the ratio's value at that time into NA, too.

The parameter *FCstor* is kind of a special case and, therefore, needs further explanations. Its calculation is mainly based on the directly measured CO_2 value, more precisely on the temporally previous and subsequent CO_2 values. As soon as one of these two CO_2 values is rejected, the *FCstor* value that lies in between turns into *NA* as well. *NEE* in turn depends on a valid *FCstor* value. However, if *FCstor* is set to *NA*, the corresponding *NEE* is not simply thrown away but replaced by the *Cov[wCO₂]* value at that certain time step (cf. Figure 13).

```
325 ### Check temporally adjacent CO2 values to validate FCstor
326
327 # FCstor column to be checked
328 z <- grep("^FCstor", colnames(dat.out)) [length(grep("^FCstor", colnames(dat.out)))]
329 # NEE column to be checked
330 y <- grep("^NEE", colnames(dat.out)) [length(grep("^NEE", colnames(dat.out)))]
331
332 # Logical vector for FCstor values with valid adjacent CO2 values
333 na.FCstor.na <- logical(length=0)
334 for (i in 2:(length(dat.out$"CO2[mmol/m³]_sd")-1))
335   na.FCstor.na[i] <- sum(is.na(c(dat.out$"CO2[mmol/m³]_sd"[i-1],
336                                  dat.out$"CO2[mmol/m³]_sd"[i+1]))) == 0
337
338 # Rejection of FCstor values with invalid adjacent CO2 values
339 dat.out[which(!na.FCstor.na),z] <- NA
340
341 # Correction of NEE values with invalid FCstor values by inserting Cov[wCO2] value
342 dat.out[which(!na.FCstor.na),y] <- dat.out$"Cov[wCO2]_threshold"[which(!na.FCstor.na)]
```

Figure 13: Validation of *FCstor* and *NEE* values by checking temporally adjacent CO_2 values

3.4.8 High-quality (and gap-filled) output data

Finally, the multi-step error filtered and optionally gap-filled eddy-covariance data set is saved as a *CSV* file in the desired output folder. At the moment, three different output types exist:

- A detailed output file *(*_long_*.csv*) that contains all the columns generated by the *SpikeCheck* function, the original input data and optionally the gap-filled measurement series
- A short output file *(*_short_*.csv*) that contains nothing but the spike-filtered column of the particular parameter, and optionally the gap-filled columns

- An output file that contains *NA* counts *(*_na_*.csv*) for each single column of the detailed output file in order to give a brief statistical overview of the increasing and decreasing amount of *NA* values for each operation (cf. Table 4)

Table 4: Cut-out of the output file containing *NA* counts for each parameter and *SpikeCheck* operation.

	NA (%)	NonNA (%)	Total (%)	CountNA	CountNonNA	CountSum
FCstor[mmol/m²s]	0	100	100	0	2163	2163
FCstor[mmol/m²s]_threshold	27.14	72.86	100	587	1576	2163
FCstor[mmol/m²s]_AGC_filtered	27.14	72.86	100	587	1576	2163
FCstor[mmol/m²s]_QCflag9_filtered	27.83	72.17	100	602	1561	2163
FCstor[mmol/m²s]_QCflag8_filtered	28.99	71.01	100	627	1536	2163
FCstor[mmol/m²s]_QCflag7_filtered	31.11	68.89	100	673	1490	2163
FCstor[mmol/m²s]_abs_d	32.5	67.5	100	703	1460	2163
FCstor[mmol/m²s]_quantile	33.56	66.44	100	726	1437	2163
FCstor[mmol/m²s]_sd_l	33.7	66.3	100	729	1434	2163
FCstor[mmol/m²s]_sd_u	33.7	66.3	100	729	1434	2163
FCstor[mmol/m²s]_sd	35.51	64.49	100	768	1395	2163
FCstor[mmol/m²s]_gf2gap	34.67	65.33	100	750	1413	2163
FCstor[mmol/m²s]_gf1gap	34.3	65.7	100	742	1421	2163

4 Quick-start instructions

This chapter is aimed at users that prefer a rather quick execution of the *MSEF* software without going too much into detail. However, we urgently recommend turning attention to Chapter 3.2 for further information about the configuration file.

The first step towards a successful execution is to open the configuration file (cf. Chapter 3.2) and adjust the settings to individual demands. Alternatively, the script can also be executed using predefined settings (cf. Tables 3.2a, 3.2b). However, data sets from different locations around the world usually require different settings, e. g. upper and lower reasonable consistency limits. Therefore, we strongly suggest altering the default values before executing the script.

Secondly, the user has to adjust the working directory (cf. Chapter 3.4.1) in case the program is transferred to another platform. For that purpose, the R script *MSEF_*.r* must be opened with a proper R software or an arbitrary text editor, respectively. Afterwards, the line of code defining the path to the current working directory (line 10 for *MSEF_v1.1.r*) must be altered, followed by saving the R file. As the script will automatically look for required subfolders in that very root directory, this step is inevitable for a proper execution.

Thirdly, the user has to insert filenames of both the result file and the QA/QC file in order to run the software with different data sets. Again, this step comprises opening the R file *MSEF_*.r*, altering the particular lines of code (lines 30 and 31 for *MSEF_v1.1.r*) and saving these modifications.

Finally, the program can be executed by opening the main R file *MSEF_*.r* with a proper R software and running the script. Previous changes including configuration file, working directory and input files will automatically be inherited.

5 Summary

The aim of this working report was to present a software approach for multi-step error filtering of eddy-covariance raw data like it's for example generated by the TK3 software package (Mauder and Foken 2011). This includes the consideration of reasonably defined consistency limits, a quality filter based on *STV*s and flags as well as a statistical spike check. To reduce the thus generated number of *NA* values, a gap-filling routine (see Work Report 59: Zhao et al. 2014) could optionally be applied in order to fill small gaps by linear interpolation.

As a next step during the continuous revision of the described *MSEF* software package, a method for singular spectrum analysis (*SSA*) of time series will be tested and implemented in order to fill not only small, but also larger measuring gaps with statistically consistent values.

References

FOKEN, T. & B. WICHURA (1996): Tools for quality assessment of surface-based flux measurements. - Agricultural and Forest Meteorology 78, 83-105.

FOKEN, T. (1999): Der Bayreuther Turbulenzknecht. University of Bayreuth, Dept. of Micrometeorology, Work Report No. **1**, Internet: ISSN 1614-8926

FOKEN, T., GÖCKEDE, M., MAUDER, M., MAHRT, L., AMIRO, B. D. & J. W. MUNGER (2004): Post-field data quality control. In: LEE, X., MASSMAN, W. & B. LAW (Eds.): Handbook of Micrometeorology. A Guide for Surface Flux Measurement and Analysis. Dordrecht: Kluwer Academic Publishers, 181-208.

LÜERS, J. (2011): Post-processing of atmospheric flux and meteorological basic data. Presentation at the University of Bayreuth, Department of Micrometeorology.

MAUDER, M. & T. FOKEN (2004): Documentation and Instruction Manual of the Eddy-Covariance Software Package TK2. University of Bayreuth, Dept. of Micrometeorology, Work Report No. **26**, Internet: ISSN 1614-8926.

MAUDER, M. & T. FOKEN (2011): Documentation and Instruction Manual of the Eddy-Covariance Software Package TK3. University of Bayreuth, Dept. of Micrometeorology, Work Report No. **46**, Internet: ISSN 1614-8926.

PAPALE, D., REICHSTEIN, M., AUBINET, M., CANFORA, E., BERNHOFER, C., KUTSCH, W., LONGDOZ, B., RAMBAL, S., VALENTINI, R., VESALA, T. AND D. YAKIR (2006): Towards a standardized processing of Net Ecosystem Exchange measured with eddy covariance technique: algorithms and uncertainty estimation. - Biogeosciences 3, 571–583.

R DEVELOPMENT CORE TEAM (2012): R. A language and environment for statistical computing. Vienna: R Foundation for Statistical Computing.

ZHAO, P. & J. LÜERS (2012): Improved determination of daytime net ecosystem exchange of carbon dioxide at croplands. - Biogeosciences Discussions 9, 2883-2919.

ZHAO, P., LÜERS, L., & T. FOKEN (2014): GaFiR: a gap-filling package for ecosystem-atmosphere carbon dioxide flux and evapotranspiration data. University of Bayreuth, Dept. of Micrometeorology, Work Report No. **59**, Internet: ISSN 1614-8926.

Previously published "Work Reports" of the University of Bayreuth, Department of Micrometeorology

No.	Author(s)	Title	Year
01	Foken	Der Bayreuther Turbulenzknecht	01/1999
02	Foken	Methode zur Bestimmung der trockenen Deposition von Bor	02/1999
03	Liu	Error analysis of the modified Bowen ratio method	02/1999
04	Foken et al.	Nachfrostgefährdung des ÖBG	03/1999
05	Hierteis	Dokumentation des Experimentes Dlouhá Louka	03/1999
06	Mangold	Dokumentation des Experimentes am Standort Weidenbrunnen, Juli/August 1998	07/1999
07	Heinz et al.	Strukturanalyse der atmosphärischen Turbulenz mittels Wavelet-Verfahren zur Bestimmung von Austauschprozessen über dem antarktischen Schelfeis	07/1999
08	Foken	Comparison of the sonic anemometer Young Model 81000 during VOITEX-99	10/1999
09	Foken et al.	Lufthygienisch-bioklimatische Kennzeichnung des oberen Egertales, Zwischenbericht 1999	11/1999
10	Sodemann	Stationsdatenbank zum BStMLU-Projekt Lufthygienisch-bioklimatische Kennzeichnung des oberen Egertales	03/2000
11	Neuner	Dokumentation zur Erstellung der meteorologischen Eingabedaten für das Modell BEKLIMA	10/2000
12	Foken et al.	Dokumentation des Experimentes VOITEX-99	10/2000
13	Bruckmeier et al.	Documentation of the experiment EBEX-2000, July 20 to August 24, 2000	01/2001
14	Foken et al.	Lufthygienisch-bioklimatische Kennzeichnung des oberen Egertales	02/2001
15	Göckede	Die Verwendung des Footprint-Modells nach Schmid (1997) zur stabilitätsabhängigen Bestimmung der Rauhigkeitslänge	03/2001
16	Neuner	Berechnung der Evaporation im ÖBG (Universität Bayreuth) mit dem SVAT-Modell BEKLIMA	05/2001
17	Sodemann	Dokumentation der Software zur Bearbeitung der FINTUREX-Daten	08/2002
18	Göckede et al.	Dokumentation des Experiments STINHO-1	08/2002
19	Göckede et al.	Dokumentation des Experiments STINHO-2	12/2002
20	Göckede et al	Characterisation of a complex measuring site for flux measurements	12/2002
21	Liebethal	Strahlungsmessgerätevergleich während des Experiments STINHO-1	01/2003
22	Mauder et al.	Dokumentation des Experiments EVA_GRIPS	03/2003
23	Mauder et al.	Dokumentation des Experimentes LITFASS-2003, Dokumentation des Experimentes GRASATEM-2003	12/2003
24	Thomas et al.	Documentation of the WALDATEM-2003 Experiment	05/2004
25	Göckede et al.	Qualitätsbegutachtung komplexer mikrometeorologischer Messstationen im Rahmen des VERTIKO-Projekts	11/2004
26	Mauder & Foken	Documentation and instruction manual of the eddy covariance software package TK2	12/2004
27	Herold et al.	The OP-2 open path infrared gas analyser for $CO_2$and H_2O	01/2005
28	Ruppert	ATEM software for atmospheric turbulent exchange measurements using eddy covariance and relaxed eddy accumulation systems and Bayreuth whole-air REA system setup	04/2005
29	Foken (Ed.)	Klimatologische und mikrometeorologische Forschungen im Rahmen des Bayreuther Institutes für Terrestrische Ökosystemforschung (BITÖK), 1989-2004	06/2005
30	Siebeke & Serafimovich	Ultraschallanemometer-Überprüfung im Windkanal der TU Dresden 2007	04/2007
31	Lüers & Bareiss	The Arctic Turbulence Experiment 2006 PART 1: Technical documentation of the ARCTEX 2006 campaign, May, 2nd to May, 20th 2006	07/2007
32	Lüers & Bareiss	The Arctic Turbulence Experiment 2006 PART 2: Visualization of near surface measurements during the ARCTEX 2006 campaign, May, 2nd to May, 20th 2006	07/2007
33	Bareiss & Lüers	The Arctic Turbulence Experiment 2006 PART 3: Aerological measurements during the ARCTEX 2006 campaign, May, 2nd to May, 20th 2006	07/2007
34	Metzger & Foken et al.	COPS experiment, Convective and orographically induced precipitation study, 01 June 2007 – 31 August 2007, Documentation	09/2007
35	Staudt & Foken	Documentation of reference data for the experimental areas of the Bayreuth Centre for Ecology and Environmental Research (BayCEER) at the Waldstein site	11/2007

No.	Author(s)	Title	Year
36	Serafimovich et al.	ExchanGE processes in mountainous Regions (EGER) - Documentation of the Intensive Observation Period (IOP1), September, 6th to October, 7th 2007	01/2008
37	Serafimovich et al.	ExchanGE processes in mountainous Regions (EGER) - Documentation of the Intensive Observation Period (IOP2), June, 1st to July, 15th 2008	10/2008
38	Siebicke	Footprint synthesis for the FLUXNET site Waldstein/Weidenbrunnen (DE-Bay) during the EGER experiment.	12/2008
39	Lüers & Foken	Jahresbericht 2008 zum Förderprojekt 01879 - Untersuchung der Veränderung der Konzentration von Luftbeimengungen und Treibhausgasen im hohen Fichtelgebirge 2007 – 2013	01/2009
40	Lüers & Foken (Eds.)	Proceedings of the International Conference of "Atmospheric Transport and Chemistry in Forest Ecosystems" Castle of Thurnau, Germany, Oct 5 to Oct 8, 2009	10/2009
41	Biermann et al.	Mesoscale circulations and energy and gas exchange over the Tibetan Plateau – Documentation of the micrometeorological experiment, Nam Tso, Tibet , 25 June 2009 – 08 August 2009	12/2009
42	Foken & Falke	Documentation and Instruction Manual for the Krypton Hygrometer Calibration Instrument	01/2010
43	Lüers & Foken	Jahresbericht 2009 zum Förderprojekt 01879 - Untersuchung der Veränderung der Konzentration von Luftbeimengungen und Treibhausgasen im hohen Fichtelgebirge 2007 – 2013	07/2010
44	Biermann & Leipold (Ed.)	Tibet Plateau Atmosphere-Ecology-Glaciology Cluster Joint *Kobresia* Ecosystem Experiment: Documentation of the first Intensive Observation Period (IOP 1) summer 2010 in Kema, Tibet	01/2011
45	Zhao et al.	Complex TERRain and ECOlogical Heterogeneity (TERRECO);WP 1-02: Spatial assessment of atmosphere-ecosystem exchanges via micrometeorological measurements, foot-print modeling and mesoscale simulations ; Documentation of the Observation Period May 12th to Nov. 8th, 2010, Haean, South Korea	03/2011
46	Mauder & Foken	Documentation and Instruction Manual of the Eddy-Covariance Software Package TK3	05/2011
47	Serafimovich et al.	ExchanGE processes in mountainous Regions (EGER)-Documentation of the Intensive Observation Period (IOP3) June, 13th to July, 26th 2011	11/2011
48	Hübner et al.	Documentation and Instruction Manual for the Horizontal Mobile Measuring System (HMMS)	12/2011
49	Lüers et al.	The Arctic Turbulence Experiment 2009 -additional laser Scintillometer measurement campaign 2009 at the Bayelva catchment on Svalbard: Technical documentation and visualization of the near surface measurements during the ARCTEX-2009 campaign, August, 10th to August, 20th 2009	02/2012
50	Foken	Klimawanderweg auf der Landesgartenschau in Bamberg 2012	04/2012
51	Ruppert et al.	Whole-air relaxed eddy accumulation for the measurement of isotope and trace-gas fluxes	05/2012
52	Foken	Jahresbericht 2010-11 zum Förderprojekt 01879 -Untersuchung der Veränderung der Konzentration von Luftbeimengungen und Treibhausgasen im hohen Fichtelgebirge 2007 – 2013	12/2012
53	Gerken et al.	Documentation of the Atmospheric Boundary Layer Experiment, Nam Tso, Tibet, 08th of July – 08th of August 2012	03/2013
54	Biermann (Ed.)	Tibet Plateau Atmosphere-Ecology-Glaciology Cluster Joint Kobresia Ecosystem Experiment: Documentation of the 2nd Intensive Observation Period (IOP 2) summer 2012 in KEMA, Tibet	05/2013
55	Babel et al.	Documentation of the EVENT-HMMS Experiment 2012 – Microclimatological effects of rain-out shelters within EVENT II	06/2013
56	Lüers et al.	160 Jahre Bayreuther Klimazeitreihe - Homogenisierung der Bayreuther Lufttemperatur- und Niederschlagsdaten	05/2014
57	Babel	An R routine for the simplified application of a footprint-based characterisation of a complex measuring site for flux measurements	06/2014
58	Lüers et al.	Application of a multi-step error filter for post-processing atmospheric flux and meteorological basic data.	06/2014

No.	Author(s)	Title	Year
59	Zhao et al.	GaFiR: a gap-filling package for ecosystem-atmosphere carbon dioxide flux and evapotranspiration data	06/2014
60	Foken et al.	Meteorologisches Instrumentenpraktikum an der Universität Bayreuth	08/2014