# Exercise 1
# Handling and preprocessing of measurement data

EWEM lab course, summer term 2016

Matthias Wächter, April 13, 2016

## Introduction

The aim of this exercise is to give an impression of how to handle raw measurement data such that quantitative analyses can be performed. In many cases this is the most difficult and time-consuming step in measurement data analysis.

The exercise is based on 1 Hz measurement data of the German FINO I platform from which also the 10 min mean values have been derived, which in turn will (probably) be used for later exercises. It is therefore possible (and encouraged) to compare results of this exercise to those of the later ones.

In order to keep the time and efforts limited, many important aspects of wind measurement are not covered in this exercise. Among them are tower shadow effects, vertical wind profiles, and quantitative outlier detection. There will be a lecture concerned with these topics in one of the first lessons.

## Report and evaluation

We expect one rather short report per group of students for the entire exercise, including an explanation of the method(s) you used to derive your results, figures showing your findings and a discussion afterwards. For each of the following numbered tasks at least one sentence should be written. For this exercise the documentation of your processing steps is of major importance. Therefore *submit your Matlab[1] code additional to your report, either as file or in printed form.*

Points are given for the different tasks of this exercise, which are noted in the following such as $\boxed{1}$.

## Software

As for the rest of this course, the basic idea was to solve the exercise in Matlab. Example solutions have been prepared and the teachers have (hopefully) sufficient knowledge of this

[1] All registered trademarks in this text are properties of their respective owners.

software to support you. Computers and licences for students are provided in the CIP room W2 2-249, have a look at `https://www.uni-oldenburg.de/physik/cip/`. Apparently students can now also use the license pool of the university. Hints are given in the introductory lecture during the first meeting.

An almost compatible alternative to Matlab is GNU Octave. It is free software (free both as in “freedom” and “free beer”). Few hints on Octave are given in this exercise. Make sure you are using version $\geq$ 4.0.0 in order to benefit from the new GUI. For most tasks the `statistics` package needs to be loaded.

If you prefer, it is also tolerated that you solve this exercise using any software of your choice. But please keep in mind that we can support you only in Matlab and Octave, with anything else you will be *completely on your own*.

## Tasks

Preparation:

1. Download the data from Stud-IP → WPMP → Files → General Folder → Task 1 → `1301.txt.zip`.
2. When uncompressed, the file contains an ASCII table with columns ”Time”, ”d90”, ”d33”, ”u100”, ”u90”, ”u80”, ”u70”, ”u60”, ”u50”, ”u40”, ”u33”. ”Time” is the time stamp, ”uXX” the cup anemometer wind speed at height XX m, and ”dXX” the wind vane direction at height XX m.

Tasks:

1. Importing the data to Matlab.

   Load the data into your Matlab or Octave workspace. Store the data using the Matlab binary format `*.mat` for efficient reading when working at the following tasks of this exercise. $\boxed{1}$

   Hint 1: Use the Matlab function `readtable()` for the import process. This seems to be the fastest of all options[2].

   Octave hints: In Octave, you are unfortunately left with `textscan()`. It does import the data successfully, but takes considerably longer. A possible command might look like:

   ```
   FinoRaw = textscan(file_handle, 'format_string', 'delimiter', '\t');
   ```

   The result will here be a so-called “cell array” because time stamp and measurement data are of differing data types. Thus the data have to be extracted into character and numeric arrays in a separate step.

[2] This was the solution of the 2015 group of J. Barnhoorn, M. Plut, and F. Börgel. Thank you!

2. Marking invalid data.

   Invalid data have been written as “-999” by the measurement system. Replace all occurrences by `NaN` to allow for efficient data handling by Matlab. It is nevertheless not guaranteed that any occurrence is present in the provided data. $\boxed{1}$

3. Generating continuous time axis.

   You will find that the time stamps are not continuously obeying the 1 s sampling time steps, but there are larger time lags from time to time. Fill the gaps such that the time steps of the month have all the identical time lag of 1 s. This is especially helpful when quantities have to be computed for specific time lags, such as in the last task of this exercise. $\boxed{1}$

   Hint: An efficient way to do this is to first convert all time stamps into the number of seconds since beginning of the measurement. Here the function `datenum()` is helpful. Then construct a suitable matrix, covering the complete duration of the measurement in steps of 1 s in a time column, with additional `NaN` columns for the measurement data. Now insert the measurement data at their correct position with respect to their time stamps[2]. This can be done most efficiently as an array operation.

4. Averaging to 10 min mean values.

   Compute 10 min mean values of all data and store for further use. Also include the standard deviations for all 10 min intervals. Here the Matlab binary format `*.mat` is appropriate. Consider the special needs of circular data (i.e., wind directions), following the separate document on this topic. $\boxed{1}$

   Plot the ten minute means of `u90` for the day 2013-01-30. Add to the same plot the values of $(\overline{u} + \sigma_u)$ and $(\overline{u} - \sigma_u)$ to indicate the range of fluctuations. The plot should look similar (but not identical) to figure 1. Discuss statistical properties of the data in your plot: Do you think the mean value and standard deviation are stationary in this 10 minute interval? Explain your opinion. Keep it in mind for the next point. $\boxed{1}$

5. Extreme values.

   From time to time, extremely high or low values are recorded by the measurement system, so-called “spikes”. An interesting question is whether these are correct measurements of real extreme wind situations or rather unphysical, erroneous measurements.

   Following a common procedure, in intervals of 10 min all values deviating more than $5\,\sigma$ from the mean would be considered erroneous and discarded. Find a ten minute interval of the 100 m anemometer which contains an event exceeding $5\,\sigma$ from the mean wind speed and plot this interval. The plot should include additional lines for mean and $\pm 5\,\sigma$ wind speeds, and look similar (but not identical) to figure 2. $\boxed{1}$

   Decide whether your example is a correct measurement or reflects just an error. Explain the reasons of your decision. Which assumptions are implied by the $5\,\sigma$ criterion? Are those assumptions fulfilled for wind measurements? Do you find more criteria to make up your decision? $\boxed{1}$

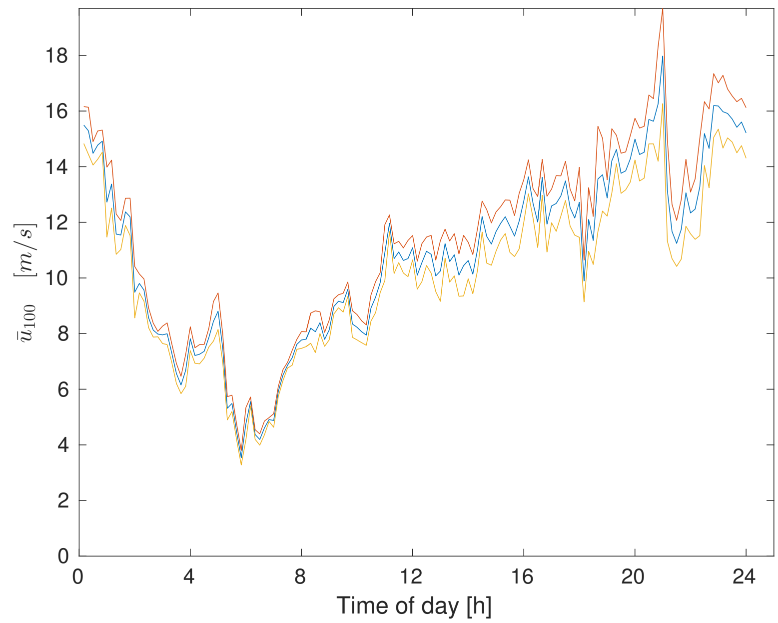


Figure 1: Example for ten minute mean values of wind speeds at the 90 m anemometer for one day, with additional lines for $(\overline{u} \pm \sigma_u)$. Note that this is a different day than the one asked for in the text.

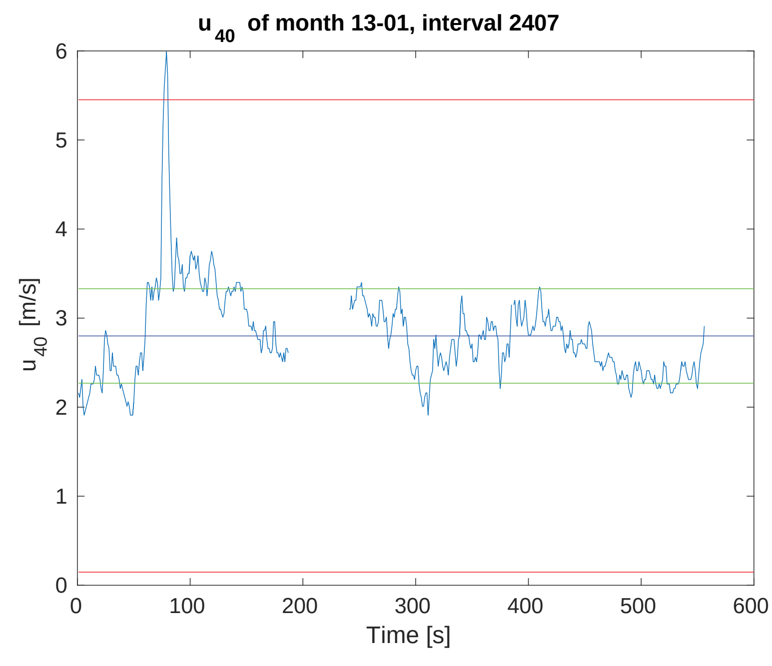


Figure 2: Example for an extreme wind speed event exceeding $5\,\sigma$ at the 40 m anemometer. Additionally, green lines mark $\pm\sigma$ and red lines $\pm 5\,\sigma$ values, respectively. Note that this is a different anemometer than the one asked for in the text.

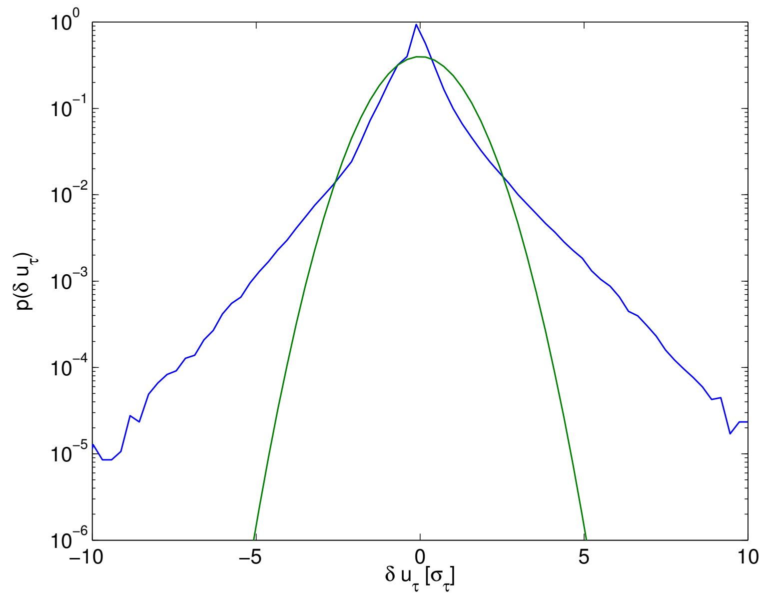


Figure 3: Example for an increment PDF of wind speeds at the 100 m anemometer, for the month 2012-03 and a time lag of $\tau = 2\,\mathrm{s}$. Note that these are different anemometer, month, and time lag than those asked for in the text.

6. Increment PDF

   Compute the probability density function (PDF) of the 90 m wind speed increments $\delta u_\tau = u(t+\tau) - u(t)$ for a time lag of $\tau = 1\,s$. Normalize the data to unit standard deviation *before* computing the PDF. Plot the PDF in semi-logarithmic scale and adjust the number of bins in the `hist()` function to obtain a rather smooth curve. Add a Gaussian PDF of unit standard deviation to the same plot. The result should look similar (but not identical) to figure 3. $\boxed{1}$

   What do you observe for small values and, especially, for extreme values of $\delta u$? How is the probability of extreme wind speed changes, compared to the Gaussian PDF? It should be noted that the Gaussian increment PDF is implicitly assumed in the respective IEC standards. $\boxed{1}$

## Presentation topics

(a) Explain your solution and results concerning tasks 1, 2, 3, 4, 5.

(b) Explain your solution and results concerning tasks 1, 2, 3, 5, 6.

## FINO 1 data legal issues

The FINO data were kindly provided by the “Federal Maritime and Hydrographic Agency” (BSH) and are free of charge for scientific use within the EU. Please delete all the data when you successfully passed the Laboratory Project. If you are interested in using FINO data for your own research, apply for an account via: `http://fino.bsh.de`