Albert-Ludwigs-University Freiburg

Master Thesis

A custom sensor network approach for detecting hydrological connectivity by soil moisture patterns

Author:
Mirko Mälicke

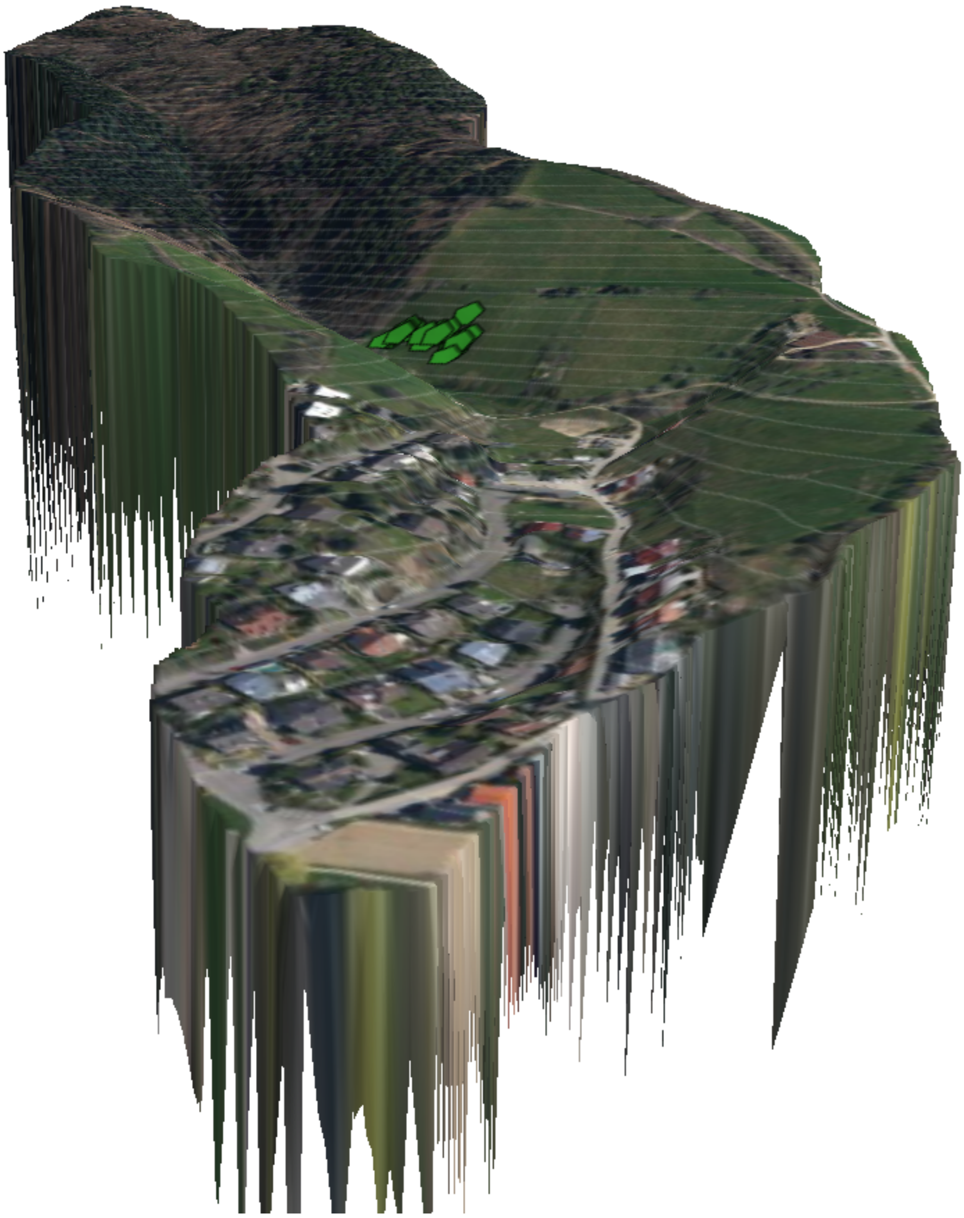
Supervisor:
Prof. Dr. Markus Weiler
Prof. Dr. Friederike Lang

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Chair of Hydrology
Institute for Earth and Environmental Science
Faculty for Environment and Natural Resources

March 9, 2016



3D model of the investigated Rütlitobel catchment with green marked sampling points indicating the study site.

Declaration of Authorship

I, Mirko Mälicke, born 08.09.1987 in Karlsruhe (matr. 2929008) declare that this thesis titled, 'A custom sensor network approach for detecting hydrological connectivity by soil moisture patterns' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Mirko Mälicke

Date:

"In the face of overwhelming odds, I'm left with only one option, I'm gonna have to science the shit out of this."

Matt Damon aka. Mark Watney in 'The Martian'

ALBERT-LUDWIGS-UNIVERSITY FREIBURG

Abstract

Faculty for Environment and Natural Resources
Institute for Earth and Environmental Science

Master of Science

A custom sensor network approach for detecting hydrological connectivity by soil moisture patterns

by Mirko Mälicke

The formation of runoff and flood prediction are still in the spotlight of hydrological research. While other studies back on cost intensive solutions like expert-driven sophisticated models or mainframe computers, this thesis introduces a cost-efficient 100 % open source, do-it-yourself data logger device. Among other objectives, this device had to be of sufficient precision and should perform as good as the Decagon® EM50™ logger in detecting patterns in soil moisture at a study site located in the Black Forest. It could be shown that there is a high potential of matching those slumbering in the device, but a short timeframe and power supply issues prevented the device from developing its full potential. While being more precise in time step management and performing well under controlled conditions, the custom device could not compete the commercial solution in the field in terms of data quality and explicitly not in terms of reliability.

Acknowledgements

First of all I want to thank my thesis supervisors for taking on this thesis. Further the constructive criticism during the work and helpful hints while being a little bit stuck in the wide field of semivariograms.
Secondly, it would not have been possible to build the used devices in such a short time, if it would not have been for the Arduino community and especially microcontroller.net. Therefore I want to thank nobody in special, but everybody who feels involved.
Another very important material input into this thesis was made by the Karl-Steinbuch-Scholarship awarded by the Film- und Mediengesellschaft Baden-Württemberg to my person, which was not intended for this thesis, but I could reuse a fully featured electronics workshop funded by the scholarship for another project.
Last but not least, the Hydrology-Scholarship awarded to my person by the Förderverein Hydrologie e.V. contributed the critical amount of money, which actually made the small scale series used in this thesis possible. Without this money the described power supply errors would not have been almost solved and the custom system would have been left without a single measurement.

Contents

Declaration of Authorship ii

Abstract iv

Acknowledgements v

List of Figures ix

List of Tables xi

Abbreviations xii

Symbols xiii

1 Introduction 1
- 1.1 Overview 1
- 1.2 Problem 2
- 1.3 Objectives 2
- 1.4 State of Science 5
  - 1.4.1 Hydrological Connectivity 5
- 1.5 State of Technology 7

2 Methods 10
- 2.1 Site Description 10
- 2.2 Data Verification 14
  - 2.2.1 Reference 14
  - 2.2.2 Custom Network 17
- 2.3 Hydrological Connectivity 21
  - 2.3.1 Discharge Data 22
  - 2.3.2 BFI & runoff coefficient 22
  - 2.3.3 spatial patterns 24
- 2.4 Test Scenarios 29
  - 2.4.1 Battery Lifetime 30
  - 2.4.2 Battery Characteristic Curve 30
  - 2.4.3 Accuracy & Precision 31

3 Sensor Network **33**
3.1 Overview 33
3.2 Data Logger 34
3.2.1 Hardware 34
3.2.2 Software 36
3.3 Sensors 37
3.4 Server - heart of the network 42
3.5 Web and Data Services 43
3.5.1 Web Service 44
3.5.2 Storage Service 45
3.5.3 Data Service 45

4 Results **47**
4.1 Overall Performance 47
4.2 Technical Lab Tests 49
4.2.1 Battery Life 49
4.2.2 Battery Characteristic Curve 50
4.2.3 Accuracy & Precision 51
4.3 Test Site Data 56
4.3.1 Time Step Data 56
4.4 Data Products 61
4.4.1 Soil Moisture 61
4.4.2 Connectivity - Range 68
4.4.3 Connectivity - Catchment Response 88

5 Discussion **93**
5.1 Custom Sensor Network 93
5.1.1 Hardware 93
5.1.2 Server 94
5.1.3 Limitations 94
5.2 Technical Lab Tests 95
5.2.1 Battery Life 95
5.2.2 Battery Characteristic Curve 95
5.2.3 Accuracy & Precision 96
5.3 Data Products 97
5.3.1 Commercial Network 97
5.3.2 Custom Network 98
5.3.3 Connectivity - Range 99
5.3.4 Connectivity - Catchment response 104

6 Conclusion & Outlook **106**
6.1 Custom Sensor Network 106
6.1.1 DataLogger Micro 106
6.1.2 Soil Moisture Sensor 106
6.1.3 Server 107
6.2 Technical Lab Tests 107
6.2.1 Battery Characteristic Curve 107

6.2.2 Accuracy & Precision . . . 108
6.3 Data Products . . . 108
6.3.1 Soil Moisture . . . 108
6.3.2 Connectivity - Range . . . 109
6.3.3 Connectivity - Catchment Response . . . 110
6.3.4 Coming back to the Objectives . . . 110
6.4 Outlook . . . 113

A Glossary **115**

B Data Access **117**

C Database **118**
C.1 Trigger Functions . . . 118

D Analysis Scripts **120**

E Datalogger **131**
E.1 Layout . . . 132
E.2 Schematics . . . 133

F Firmware **134**
F.1 Core Firmware . . . 134
F.2 Main Firmware . . . 135

G Photos **140**

H Kriging **144**

I Supplementary DVD **146**

Bibliography **147**

List of Figures

2.1 The Study Site location in the Ruetlitobel catchment. 10
2.2 Photo of the study site after the two sensor networks had been installed. . 12
2.3 Temperature effect on the frequency of oscillators. 18
2.4 Hydrograph separation using local minimum method. 22
2.5 Sample variogram 25

3.1 Mounted and deployed DataLogger Micro. 34
3.2 Soil moisture M1 schematics snapshot. 37
3.3 Senor electrode prototypes. 38
3.4 ADC Calibration curves for different internal ADC aggregation levels. . . 39
3.5 Voltage - Resistance diagram for the sensor circuit. 41
3.6 Banana Pro used as an experimental data service server. 42

4.1 battery characteristic curve from 20 different measurements in different test scenarios. 50
4.2 Calculated water content values and fitted exponential function. 51
4.3 The result of sensor accuracy and sensor precision test. 52
4.4 Accuracy and Precision test scenario measurement residuals. 55
4.5 Time step deviation of EM50 time stamps. 57
4.6 Time step deviation of DataLogger Micro time stamps. 58
4.7 The time step deviation of compensated and uncompensated time steps in ppm per second. 59
4.8 em50overviewResult overview of all measurements taken by the commercial system. 61
4.9 Overall sensor quality overview for all 5TE used in this work. 62
4.10 Sensor quality overview for each commercial sensors used in this work. . . 63
4.11 Overall sensor quality overview for all custom sensors used in this work. . 64
4.12 Sensor quality overview for each custom sensor used in this work. 65
4.13 Result overview of all measurements taken by the custom system. 67
4.14 Range time series and model value plots for the commercial system based on the gaussian semivariogram model. 70
4.15 Semivariance and Gaussian semivariogram model for the commercial system, 546 hours after the campaign started. 71
4.16 Semivariance Image based on the commercial system readings from 550 h to 650 h after measuring campaign start. 72
4.17 Range time series and model value plots for the custom system based on the modified exponential semivariogram model. 75
4.18 Semivariance and modified exponential semivariogram model for the custom system. 76

4.19 Semivariance Image based on the custom system readings from 147 h to 197 h after measuring campaign start. 77
4.20 Semivariance and modified spherical semivariogram model for the commercial system after bin reduction. 80
4.21 Range time series and model value plots for the commercial system based on the modified spherical semivariogram model. 81
4.22 In-depth representation of three variograms fitted to the EM50 data. 82
4.23 In-depth representation of three variograms fitted to the EM50 data after equalizing the bin size. 83
4.24 Range time series and model value plots for the commerical system based on the spherical semivariogram model after evening bin sizes and limiting the lags. 85
4.25 Semivariogram and semivariance image for the commercial system based on the water content measurements taken 598 hours after the campaign started. 86
4.26 Scatter plots relating the different result parameters to the soil moisture. 88
4.27 Overview of measured and calculated parameters describing the catchment response. 89
4.28 Extract of the discharge time series with calculated BFI values. 90
4.29 Double mass curve of rainfall and cumulative. 91
4.30 Cross correlation between the range and runoff coefficient for all observations over time. 92

E.1 PCB Layout for the 1.0 Version of the data logger 132
E.2 Schematics for the 1.0 Version of the data logger 133

G.1 Photo of the climate and gauging station. 140
G.2 Photo of the sensor calibration test. 141
G.3 One of the DataLogger Micro devices during the mounting process. 142
G.4 Production View 143

H.1 Interpolated soil moisture map of November 18, 2015. 145
H.2 Interpolated soil moisture map of November 20, 2015. 145

List of Tables

1.1 Summary of found measurands in reviewed publications. 9

2.1 Distance matrix for the commercial sensor network. 13

3.1 DataLogger Micro - Technical specifications 35
3.2 Dropping resistance values with the calculated power consumption and maximum soil moisture electrode distance. 40
3.3 Choosen Specification of Banana Pro™ from LeMaker® Homepage. . . . 43
3.4 Data Service Python package releases on GitHub. 46

4.1 Weights of the soil sample used in the accuracy and precision test. 52
4.2 Porosity of 9 soil samples taken on the study site. 63
4.3 Model selection result for the commercial sensor network. 68
4.4 Evaluation result for the commercial sensor network. 69
4.5 Model selection result for the custom sensor network. 74
4.6 Evaluation result for the custom sensor network. 74
4.7 Model selection result for the commercial sensor network after bin reduction. 78
4.8 Evaluation result for the commercial sensor network after bin reduction. . 79
4.9 Model selection result for the commercial sensor network after bin reduction and median usage. 86

B.1 Data access ids and web URLs. 117

D.1 Exact Python versions used for analysis. 120

Abbreviations

Prefixes

m	**m**illi	10^{-3}
K	**K**ilo	10^{3}
M	**M**ega	10^{6}
G	**G**iga	10^{9}

Units

A	**A**mpere
B	**B**yte
Hz	**H**ert**z**
V	**V**olt

Computer & Server

ADC	**A**nalog-to-**d**igital **C**onverter[a]
CMS	**C**ontent **M**anagement **S**ystem[a]
CPU	**C**entral **P**rocessing **U**nit
GUI	**G**raphical **U**ser **I**nterface[a]
MOSFET	**M**etal **O**xide **S**emiconductor **F**ield-**E**ffect **T**ransistor
PCB	**P**rinted **C**ircuit **B**oard[a]
RAM	**R**andom-**A**ccess **M**emory
SSH	**S**ecure **Sh**ell
SSD	**S**olid-**S**tate **D**rive[a]
SATA	**S**erial **A**dvanced **T**echnology **A**ttachment

Methods

BFI	**B**aseflow **I**ndex
RMSE	**R**oot **M**ean **S**quare **E**rror

[a] This term is explained in Appendix A.

Symbols

C	runoff coefficient	*dimensionless*
D	depth	m
f	frequency	Hz
I	current	A
P	precipitation	mm
Q	discharge	$m^3 * s^{-1}$
R	resistance	Ω
T	temperature	$^\circ C$
V	voltage	V
γ	semivariance	*dimensionless*
Θ	soil moisture content	$m^3 * m^{-3}$
Φ	porosity	$m^3 * m^{-3}$
σ	standard deviation	*dimensionless*

Chapter 1

Introduction

1.1 Overview

The formation of runoff and flood prediction are in the spotlight of hydrological research. Various publications could be found investigating just new approaches for flood prediction, that were all published in 2016 and headed this issue all around the world (Durocher et al., 2016, Seenath et al., 2016, Perez et al., 2016, Zin et al., 2016). Most of these research projects focus on cost intensive infrastructure like mainframe computer or complicated models run by experts. Especially for developing countries, these resources are in many cases not available. Recent innovations in the field of embedded systems gave birth to a large community dealing with topics and devices known as the Arduino[1] community. This is a small microcontroller based development platform that's easy to reproduce and easy to program since the device is open source, not only in software but also in hardware. By now, there are literally hundreds of derivatives and thousands of web resources, that make the development of new derivatives very easy. Appendix E shows exactly one of these devices created by the author prior to this work. This is an completely unmounted and untested approach for an cost-efficient soil moisture data logger, that should enable especially developing countries to build up very cost-efficient, precise and easy-to-handle soil moisture networks. In this thesis it will be shown, that a comprehensive soil moisture network is sufficient to estimate catchment's hydrological connectivity. This connectivity can be interpreted as a measurand for overall runoff formation (Meyles et al., 2003, McNamara et al., 2005) and therefore produce important

[1]Arduino Homepage. URL: http://arduino.cc. Accessed: February 18, 2016.

information for flood prediction. Furthermore, when developed, it is not only cost-efficient and simple to operate, but will be 100 % solar driven and even the processing server is open source and available all around the world. Therefore, to the knowledge of the author, this would evolve to an unique system in being not only independent from electrical connection but also from patents and commercial licenses throughout the complete data mining and processing workflow.

1.2 Problem

Two key issues can be formulated for this thesis. First, it has to be proven that a custom system can match all requirements that evolve from a scientific usage of this data. All applied methods or models have their specific requirements and the custom system has to match these. Therefore it is not enough to develop the custom data logger as good as possible in the given time frame with the given resources, but also to fulfill external requirements, like reliability or precision, within only this time frame and the given resources. This is expected to be the most challenging aspect of the thesis.
Secondly, only a limited measuring campaign can be handled. Developing a scientifically resilient method of estimating hydrological connectivity from soil moisture patterns is not within the means of this thesis. This thesis can be seen as a pilot study pointing into the correct direction for further investigations.

1.3 Objectives

There are various objectives in this thesis. Each of these can be summarized to one main objective:

> *"The custom sensor network approach for soil moisture measurements presented in this thesis can be used over established commercial solutions like the Decagon® EM50^{TM} and 5TE/5TM while being (i) cost efficient, (ii)* 100 % *open source, (iii) highly adaptable, (iv) of sufficient precision,(v) repeatable, (vi) suitable for answering an exemplary hydrological issue (vii) automated and (viii) in principle solar driven."*

This includes both, the software and hardware, from the data logger and sensor in the field over the server and database unit for storing and saving data to the Python interface for accessing, analyzing and visualizing the data. The objective can be broken down into eight single, measurable and checkable objectives.

i - cost efficient. The approximate price for one EM50^{TM} with 5 5TE/5TM accounts for 1.000 \$. The equivalent system of custom loggers would include three logger with two moisture sensors connected. The total price should not exceed $3 * 25\$ + 6 * 5\$ = 105\$$.

ii - 100 % **open source.** This applies to soft- and hardware. The schematics and PCB layout for the data logger unit will be published under open source license. The server has to be published the same way. The data logger firmware will be published open source. The server and complete data visualization and processing tool has to be build up on open source solutions, this also includes the operation system and database application.

iii- highly adaptable. The custom system shall offer configuration freedom way beyond the Decagon® system. The user can specify any time step, aggregation level and whether the measurements shall be taken integrative or at a given key point for each time step. By using and extending an exhausting Python database application called *Openhydro* enormous amounts of data can be processed and visualized in a short period of time, producing *typical hydrological* data products like hydrographs, duration curves or climate diagrams.

iv - sufficient precision. One important aspect for any custom build solution is precision. The system has to match a scientifically required precision, which may vary with each issue. The ideal data precision result is to find no statistically significant disparity between the custom and commercial system during various test scenarios. The minimum required precision is to measure the soil moisture with a precision and accuracy of 2.5 % each translated to water saturation level.

v - repeatable. Tested in the laboratory and during field work, the custom network has to be able to repeat its measurements. For the field test, two sensors applied at the same location shall not report an significant shift in values after calibration.

vi - exemplary hydrological issue. Beside a comparison of the custom and commercial system on a technical level as performed in (iv) and (v), both networks will evaluate the same hydrological issue by applying the same methods to their data. For the whole measuring campaign patterns in the spatial distribution of soil moisture will be detected. Both systems will relate these patterns to the runoff coefficient calculated for the whole campaign duration. These results will be compared and evaluated. The objective for the custom network will be to reach the results of the commercial system.

vii - automatization. A cost efficient system can increase the number of sensors significantly using the same budget. This is only administrable if the sensor network is established at a high level of automatization. For this thesis this does explicitly exclude wireless data transmission from each data logger to the central server unit (neither custom nor commercial system). The development would require a larger time frame and a bigger development budget. Therefore the automatization will be applied to the server side. From uploading raw sensor data to a server driven data service offering raw data, checked data and data products an automated workflow will be applied and evaluated.

viii - in principle solar driven. Especially when opening developing countries as a market for the custom sensor network, one has to take into account the level of electricity coverage. For many countries, solar driven system are crucial in order to establish a region-wide sensor network. A real solar power supply lies beyond the limits of this thesis, therefore the power consumption of each network part has to be decreased to a reasonable limit. The Python interface and processing package has to be as resource conserving as necessary, to be able to renounce powerful (and consumptive) CPUs. A theoretical solar supply satisfying the power consumption of the complete system, may not be more expansive than the system itself in order to match this objective.

1.4 State of Science

1.4.1 Hydrological Connectivity

As stated in 1.3 this thesis investigates a custom sensor network with the help of the use cases of *hydrological connectivity*. Unfortunately, this is not a well defined hydrological term with only one meaning all around the world (Ali and Roy, 2009). Therefore, a number of review articles elaborating *hydrological connectivity* could be found (Bracken and Croke, 2007, Tetzlaff et al., 2007, Ali and Roy, 2009, Lexartza-Artza and Wainwright, 2009, Bracken et al., 2013). Although most publications about connectivity were published within the last 10 to 15 years, the concept of relating different part of the hydrological cycle to each other is not new. The earliest described connection between rainfall and runoff through infiltration found was reported by Horton (1933).

A milestone in research on hydrological connectivity was the concept of *'preferred states'* (Grayson et al., 1997). Two catchments in Australia were found to be in a *wet* or *dry* state throughout most of the year. During dry state the water movement in the soil is driven by *local control*, like soil texture. In contrast, the wet state establishes catchment connectivity and water movement is driven by *non-local control*, like slope. These connected catchments produce fast hydrograph response (Kirkby, 1988, Lexartza-Artza and Wainwright, 2009). In consequence the processes of runoff generation are fundamental different in each state. This concept was refined by McNamara et al. (2005) by applying the concept to a small mountainous catchment in Idaho, USA. The key improvement was the insight that wet state does not connote a connected catchment. Beside two *transitional* states, McNamara et al. splited the wet state into an connected and non-connected wet state. This was necessary due to the high amount of snow (almost 50 % of precipitation) preventing the direct generation of runoff, although the catchment was in wet state.

A similar idea was developed by Ambroise (2004) making a distinction between an *contributing / non-contributing* and *active / inactive* area or period. The author takes into account, that an area generating runoff (an active area) might be non-contributing (not connected) to the stream. Therefore, for correct description and prediction of hydrograph response of an catchment, after identifying active areas and periods, the question has to be answered, whether they are contributing or not.

All these investigations found the catchment to be responding to precipitation in different ways due to the *state* the catchment was in. A state was related to a period within the year.

A factor driving the transition from one preferred state to another is the the soil moisture (Grayson et al., 1997, Meyles et al., 2003, McNamara et al., 2005). Farrick and Branfireun (2014) tried to identify a threshold value for runoff generation. In the case of semi-arid catchment in tropical Mexico, very similar to Grayson et al. (1997) Australian catchments, they were successful. In their specific case all runoff events were generated during wet state, which attuned at soil moisture contents of at least 289 mm. In consequence, the event rainfall amount did not strongly correlate with the event runoff amount.
In contrast to this findings, Nicolau et al. (1996) investigated a small mountainous catchment in southern Spain showing clearly defined preferred states. No threshold for switching states could be identified by the authors in this case. In a medium sized catchment in northern Italy, Penna et al. (2011) could observe a significant threshold of 45 % soil moisture as prerequisite for runoff generation, although this catchment did not show distinct preferred states.
As a consequence, the study area for this thesis will be investigated for threshold effects of soil moisture for runoff generation, as done in many more studies (Fitzjohn et al., 1998, Bracken and Croke, 2007, Ali and Roy, 2009, Lanni et al., 2012). It will be tried to quantify the threshold like in the described studies above. As the study area is clearly humid , this could shed some more light on soil moisture threshold characteristics. It is expected that not enough rainfall-runoff events will be recorded during this thesis, therefore the threshold will not be identified by an actual threshold value, but by a trend, when relating all moisture measurements to the rainfall.

Although also related or based upon Grayson et al. work, a number of publications tried to find patterns in *spatial connectivity* within a catchment (Meyles et al., 2003, Bracken and Croke, 2007). Following the concept of geostatistics, the soil moisture at a given point is stronger related to the soil moisture of nearer points than points far away (Webster and Oliver, 1990). As water movement in the soil is limited by its hydraulic conductivity, which is heavily dependent on the soil moisture (Van Genuchten, 1980), Meyles et al. (2003) used the range parameter of a semi-variogram as a indicator for hydrological connectivity. Using semi-variograms was not a new idea, as they are widely used for verification or description of soil moisture modeling results (Western et al., 1998, Fitzjohn et al., 1998, Bronstert and Bardossy, 1999, Rosenbaum et al., 2012). Unlike these authors, Meyles et al. did not use the semi-variogram for simple result verification, but as a result itself.

Effects of vegetation and especially vegetation changes or losses on hydrological processes have been an issue in hydrology ever since. The hydrological connectivity within a watershed might be affected as vegetation losses lead to sediment losses and impact runoff generation (El-Hassanin, 1983, Beasley et al., 1986, Hornbeck et al., 1986, Mann et al., 1988, Castillo et al., 1997, Ludwig et al., 2005, Valentin et al., 2008). Throughout most of the publications investigating on vegetation-connectivity interactions, the vegetation was assumed to change the infiltration properties by altering the infiltration capacity or bulk density (Boix-Fayos et al., 1998, Calvo-Cases et al., 2003). A lot of investigations on this effect took place in Spain, which has large landscapes of vegetation and bare-soil tessellations. Especially storm events can infiltrate under vegetation, while most runoff is generated on bare-soil with low or medium initial soil moisture (Dunkerley, 1999, Valentin et al., 1999, Boer and Puigdefábregas, 2005, Puigdefábregas, 2005). Puigdefábregas (2005) concluded vegetation to be a limiting factor for connectivity on the hillslope scale, not taking subsurface flow into account. It is expected, that the vegetation will have an significant influence on the results of this thesis. Due to time limitations, these influences will not be evaluated and connectivity-vegetation interactions do not lie within the objectives of this thesis. The study site will be declared in a way, that makes the vegetation a constant boundary condition.

1.5 State of Technology

In contrast to the *State of Science* section (see 1.4, p.5) this section will give an overview on the used sensing network. As this thesis will compare a custom, self-made, open source network to an established, commercial infrastructure, the system needs additional review.

The author evaluated an open source weather station in Kigali, Rwanda in 2013 (Mälicke, 2013) during his Bachelor thesis. This system used the Arduino Platform[2] for measuring

[2]Official Arduino Website. URL: http:arduino.cc. Accessed: March 9, 2016.

and processing the signals. At that time, the author hardly found any hydrology-specific publications involving Arduino for hydrological measurements. During the last almost three years the number of publications involving Arduino or comparable low-cost embedded systems increased dramatically, therefore only publications using custom sensors for monitoring hydrology-relevant measurands will be reviewed.

The IEEE Computer Society [3] is organizing several annual conference on topics like computing, wireless communication or sensor networks for a couple of years now. Therefore a number of publications introducing wireless sensor networks for measuring water quality parameters were found (Le Dinh et al., 2007, Rao et al., 2009, Zennaro et al., 2009, Wang et al., 2010, Nasirudin et al., 2011). While all the named publications did not focus on using low-cost and open source systems, but focused on applying new technology to make data harvesting more efficient. There was also a number of posts on different IEEE conferences explicitly focusing on open source embedded systems and system development, which were all published within the last two years (Rao et al., 2013, Lee et al., 2014, Islam et al., 2014, Usha Rani and Kamalesh, 2014). Buytaert et al. (2015) reported on developing hydrological sensor networks on the 2015 European Geosciences Union General Assembly (EGU) in Vienna. The content of this presentation was unfortunately not available to the author, just the abstract of speech. This work focused on summarizing available open source technologies for reading existing sensors and process their signals. All named proceedings from different IEEE conferences on sensor networks dealing with hydrological sensor networks were scanned for the used technology and measured parameters. An overview is shown in table 1.1 (p.9). Except Rao et al. (2009) none of the named developed custom networks. While the open source ZigBee platform is used most often, some authors only used commercial solutions. Soil moisture as a parameter was found only once. Usha Rani and Kamalesh (2014) build a open source network upon ZigBee and the so called *"GROVE - Moisture"* sensor. This is an extremely low-cost electrical moisture sensor, which is neither calibrated nor tested at any stage of production. The authors also used this sensor without evaluating it, as it was only used to produce a binary signal of the soil being dry or wet. Therefore it can be stated, that a lot of recent work on custom (open source) sensor networks is

[3]The IEEE Computer Society is a professionals society within the Institute of Electrical and Electronic Engineers (IEEE) with the scope "to advance the theory, practice, and application of computer and information processing science and technology" (IEEE Computer Society Constitution & Bylaws, art. 1, Sec. 2, 1971).

being published, but the community is lacking a cheap and open soil moisture sensor and specialized data loggers for field work.

TABLE 1.1: Summary of found measurands in reviewed publications.

Measurand	**Technology**	**Publications**
gauge	ZigBee[a]	Islam et al. (2014)
temperature, irradiation, wind speed	Arduino Mega[b]	Lee et al. (2014)
temperature, pH, conductivity, dissolved oxygen	Arduino Mega	Rao et al. (2013)
soil moisture	ZigBee	Usha Rani and Kamalesh (2014)
temperature, pH, conductivity, dissolved oxygen	90-FTL[c], SunSPOT[d]	Zennaro et al. (2009)
water level, salinity	Fleck3[e], PS100[f]	Le Dinh et al. (2007)
no specific, just the network developed	HCS08[g]	Rao et al. (2009)
pH, major ions	AT91R40008[h]	Wang et al. (2010)
pH, turbidity, dissolved oxygen	ZigBee	Nasirudin et al. (2011)

[a] open-source Arduino based wireless sensor node. http://www.zigbee.org/. Accessed: March 9, 2016.

[b] Arduino Version based on the Atmel®ATmega2560™microcontroller. http://arduino.cc. Accessed: March 9, 2016.

[c] Commercial Sensor by TPS. Specifications at: http://www.tps.com.au/products/combination/90-fl.htm. Accessed: March 9, 2016.

[d] Discontinued development platform by Oracle®. Open source embedded system similar to Arduino, not supported anymore. Informative website: http://www.sunspotdev.org/. Accessed: March 9, 2016.

[e] Commercial wireless microcontroller Node described in detail by Sikka et al. (2007). Although not open source offers a wide range of programmable options.

[f] Commercial waterproof differential pressure sensor by Goldtech. Specifications at: http://www.designflexswitches.com/switches/goldtech-ps100.php. Accessed: March 9, 2016.

[g] Reciever and transmitter both custom developments using the HCS08 microcontroller. The HCS08 is similar to Arduino's ATmega328P™also a 8bit AVR microcontroller produced by freecale. URL: http://www.freescale.com. Accessed: March 9, 2016.

[h] 32bit microcontroller by Atmel®. Very fast but expansive processor. Specifications at: http://www.atmel.com/devices/r40008.aspx?tab=overview. Accessed: March 9, 2016.

Chapter 2

Methods

2.1 Site Description

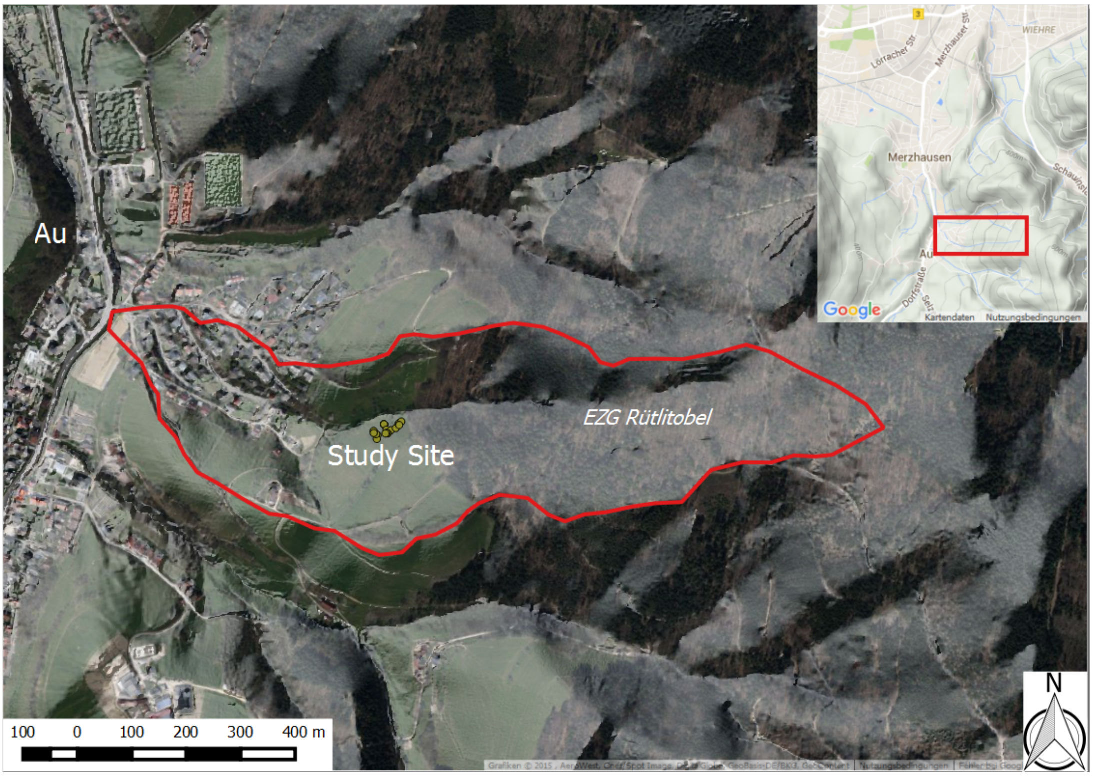


FIGURE 2.1: The Study Site (green points) location in the Ruetlitobel catchment (red outline). The data is shown upon a shading raster calculated from a 1m DTM with google® satellite image. The Ruetlitobel catchment is part of the catchment "alte Dreisam" South of Freiburg, Germany.

The study site is located in the catchment of the Rütlitobel in the Black Forest and part of the catchment "Alte Dreisam" located South of Freiburg in Southern Germany (47.957 °N, 7.838 °E) and has a size of 0.37 m^2. The Rütlitobel catchment outlines are shown in figure 2.1 in red, with the measuring plot locations on the study site represented by green dots within. The site was described in detail by Bachmair et al. (2012), the following informations were taken from precisely this publication. With elevations ranging from 340 m to 585 m above sea level it covers the low and medium elevations typically found in the black forest. The hillslope where the study site itself is located on has a North-Northeast orientation and very steep. The site is mainly covered by grassland, coniferous and mixed forest.
Bachmair et al. (2012) specifies the the rainfall with 970 mm and the mean annual temperature to 11 °C, based on data from a weather station near Freiburg (at the Weinbau Institut Freiburg) in the period from 2007 - 2011. The author used daily data from precisely the same station from 2006 to 2015. Here, the mean annual temperature was 11.6 °C and the mean annual rainfall 944 mm. In order to have comparable climatic data for the measuring campaign, this data was subset to the period from October to February, both inclusive. Here, the mean temperature was 5.9 °C and the mean rainfall sum for this period 346.5 mm.

The catchment is covered by $>$ 85 % by cambisols, the remaining area is covered by luvisols. This is characteristic for the Black Forest. Beside the town of Au in the lower, eastern catchment parts, the main landuse is forestry and pasture.

Test Design

For the study site in the Rütlitobel catchment the following test site design was chosen. A gauging and a climate station were already available on the study site. Limited to the cable length of the 5TE sensors, it was not possible to build up a completely random distributed test design for the commercial system. Therefore the commercial system is clustered into three plots on the study site. These will be referenced as Au1, Au2 and Au3[1] and are also shown in figure 2.2. Each plot contains five commercial and three custom sensors. The sensors have been deployed randomly into different directions from

[1]These are also the unique names in the *Openhydro* database application, which will be introduced later.

Figure 2.2: Photo of the study site after the two sensor networks had been installed. Beside the climate (M1) and gauging station (D1) the three measuring plots Au1, Au2 and Au3 are identified. Each plot contains five commercial and three custom sensors.

the EM50^{TM} logger, at different distances. It had been tried to vary the inter-plot sensor distances in a way that some sensors from the same plot have bigger distances than other sensors from different plots. Due to the cable lengths, the sensor distances are though clustered at about 25 m and 50 m. The full distance matrix for all commercial sensors is shown in table 2.1. Next to the middle all values are about 10 m, these are the the inter-plot distances. The outside edge table positions show the one cluster of Au1:Au3 distances, triangulary in between are the Au3:Au2 and Au1:Au2 distances forming the other cluster.

Table 2.1: Distance matrix for the commercial sensor network. All values are given in lag, this means the distance cutted to full meter.

sensor	Au1_A	Au1_B	Au1_C	Au1_D	Au1_E	Au2_A	Au2_B	Au2_C	Au2_D	Au2_E	Au3_A	Au3_B	Au3_C	Au3_D	Au3_E
Au1_A	0	7	4	16	11	24	26	32	24	22	49	41	48	48	54
Au1_B	7	0	5	9	3	21	22	28	20	17	47	38	46	46	52
Au1_C	4	5	0	12	8	24	26	32	24	22	50	42	49	49	55
Au1_D	16	9	12	0	7	25	23	30	21	19	50	40	48	49	55
Au1_E	11	3	8	7	0	19	19	26	18	15	46	36	44	44	50
Au2_A	24	21	24	25	19	0	7	8	6	7	26	17	25	25	31
Au2_B	26	22	26	23	19	7	0	8	2	4	28	18	26	27	33
Au2_C	32	28	32	30	26	8	8	0	9	11	20	10	18	19	25
Au2_D	24	20	24	21	18	6	2	9	0	3	29	19	27	28	34
Au2_E	22	17	22	19	15	7	4	11	3	0	32	21	30	30	36
Au3_A	49	47	50	50	46	26	28	20	29	32	0	11	4	1	5
Au3_B	41	38	42	40	36	17	18	10	19	21	11	0	8	10	16
Au3_C	48	46	49	48	44	25	26	18	27	30	4	8	0	2	7
Au3_D	48	46	49	49	44	25	27	19	28	30	1	10	2	0	6
Au3_E	54	52	55	55	50	31	33	25	34	36	5	16	7	6	0

2.2 Data Verification

As the comparison of different soil moisture sources is a main objective of this thesis, the methods used for comparing and evaluating the data quality are presented in this section. Each dataset will be qualified against reference measurements using established data collection infrastructure. This will be the EM50 datalogger and EC5 soil moisture sensor by Decagon.

2.2.1 Reference

plausibility

The quality control for the collected data used as reference measurements is divided into several stages. The first stage is a physical plausibility check. Soil moisture measurements have to satisfy the condition 2.1.

$$0 \quad < \quad \Theta \quad < \quad 1 \tag{2.1}$$

Where Θ is the soil water content measured in $m^3 * m^{-3}$. This is a very general approach, as the soil water content cannot exceed the porosity, therefore, as the porosity Φ is a known condition 2.1 should be changed to condition 2.2.

$$\Theta_r \quad < \quad \Theta \quad < \quad \Phi \tag{2.2}$$

Here, Θ_r is the residual soil moisture. In a laboratory test, none of the used sensors exceeded the value of $0.11 m^3 * m^{-3}$ in a completely dried soil sample. Therefore Θ_r defaults to 0.11 for the entire study site. The porosity can be calculated from soil samples taken from the field of investigation, like it will be done in this specific case, or somehow be calculated on a more global scale. This is especially useful, when field measurement are not possible. Dorigo et al. (2013) proposed calculating porosity from soil parameters, namely soil texture and organic carbon content. These parameters are available on a global scale in the *Harmonized World Soil Database*[2]. The porosity can then be estimated by a pedotransfer function, Dorigo et al. (2013) used one proposed

[2]Harmonized World Soil Database. Project Homepage including data access and visualization. URL: http://webarchive.iiasa.ac.at/Research/LUC/External-World-soil-database/HTML/. Accessed: December 1, 2015.

by Saxton and Rawls (2006).

This quality check was implemented on database level using the database trigger[3] and PL/PgSQL[4] function shown in listing C.1. This way it is possible mark the invalid data and exclude it from data queries while still preserving the original raw data. This is one of the multiple advantages of using a database over file-based solutions, as there are just different views defined onto the same data and not different data file versions.

consistency

Consistency checks of environmental measurements face on identifying periods of too rapid in- or decrease in measured values on the one hand, or too long periods without any changes on the other hand. Data showing clear diurnals like air temperature or air humidity can easily be checked against threshold values (Merchant et al., 2008). For soil moisture data, this is in many cases not possible, as long periods of dryness or rapid saturation processes are not uncommon. One approach for consistency checks on soil moisture measurements is relating them to a closely liked parameter like precipitation (Dorigo et al., 2013). Following Dorigo et al. (2011) this method is especially effective for quality checks on data from very different sources, like in the *International Soil Moisture Network*, a database and web service for collecting soil moisture measurements from all over the world (this network is also introduced and discussed in the named publication). Dorigo et al. (2013) refined the method used in Dorigo et al. (2011) (and the network), which will be adapted and presented here.

The basic idea is to flag data points as questionable or incorrect as the data points fails different quality checks. This data can then be discarded, corrected or replaced depending on the duration of resulting gaps and data availability for different correction methods. Dorigo et al. (2011) proposed a flagging of decreasing or static soil moisture values during and up to 24 hours after precipitation events. As this method lead to a significant overflagging, Dorigo et al. (2013) adjusted the method to *"flag[ging] an observation as questionable if there is a rise in soil moisture but no significant rainfall amount in the preceding 24 h"* (Dorigo et al., 2013, p.10).

[3]PostgreSQL TRIGGERS, official documentation. URL http://www.postgresql.org/docs/9.1/static/sql-createtrigger.html. Accessed: December 3, 2015.

[4]PL/pgSQL language, official documentation. URL:http://www.postgresql.org/docs/9.3/static/plpgsql-overview.html. Accessed: December 3, 2015.

This was expressed in the two conditions 2.3 and 2.4 (both adapted from Dorigo et al., 2013) each data point has to fulfill:

$$\Theta_t > \Theta_{t-1} \tag{2.3}$$

$$\Theta_t - \Theta_{t-n} > 2\sigma_{x[t-n,t]} \tag{2.4}$$

where t is the time step, $\sigma_{x[t-n,t]}$ is the standard deviation of Θ over the the preceding n time steps. The number n has to be chosen to meet the requirement $n * t = 24\ hours$. As precipitation was observed within the preceding 24 hours for a data point, Dorigo et al. (2013) suggests another check (condition 2.5) for a minimum precipitation value that has to be met:

$$\sum_{t-n}^{t} P > DA\Phi \tag{2.5}$$

where P is the precipitation in meter, D is the soil moisture sensor depth in meter, A is the accuracy of this sensor and Φ is the porosity. Dorigo et al. (2013) suggests the usage of 0.05 $m^3 * m^{-3}$ for A and 0.5 for Φ, in case not all values are known. The authors restrict this method to observations which show a direct response to precipitation events in the soil moisture measurements.

All determined porosity values were very similar and averaged to $\Phi = 0.66$. The manufacturer of the commercial sensors specifies the accuracy to $0.03 m^3 * m^{-3}$ [5]. The threshold within 24 hours can be calculated to be:

$$\sum_{0}^{24} P > 0.1 * 0.03 * 0.66 \;\equiv\; \sum_{0}^{24} P > 0.00198$$

Therefore the rainfall threshold T_r is $0.2 \frac{mm}{24h}$. As the first quality check described above, this more complex and more specific check was again integrated by a PostgreSQL trigger shown in listing C.2.

[5] Decogon devices 5TE technical specifications. URL: http://www.decagon.com/en/soils/volumetric-water-content-sensors/5tm-vwc-temp/. Accessed: November 22, 2015.

2.2.2 Custom Network

The custom sensor network is basically a small and cost-efficient data logging unit, able to connect a wide variety of sensors. The complete network is described in detail in chapter 3 (see p.33 ff.). The data collected will be treated for quality check in exactly the same way like the reference data (see section 2.2.1, p.14 ff.). Additionally some tests and use case scenarios are applied to the custom hardware comparing different hardware versions in terms of environmental usage. This section will give an overview on all methods applied to the sensor network. These are use case or test scenarios in most cases trying to evaluate the sensors performance.

sensor accuracy

As stated in section 2.2.1 (see p.14) the sensor accuracy is a necessary parameter for applying quality treatment on the measured values (see formula 2.5, p.16). For commercial sensors the sensor accuracy is a standard parameter which can be found in the relating datasheet. The accuracy of the self-build sensors used in this thesis will be determined by a test scenario.
This test scenario is described in section 2.4.3 (see p.31). This test is performed in the lab within defined and controlled conditions and was therefore moved to the test scenario section.

RTC versus 32.678 Khz oscillator

Any data logging unit needs an internal signal for determining time. This task is accomplished by an oscillator, swinging at a defined frequency f_0. Most AVR microcontrollers include an internal oscillator with a frequency of 8 MHz, defining the maximum amount of register changes which can be performed within one second. For determining time an oscillator with $f_0 = 32.678KHz$ is used. Some microcontrollers, like the ATmega328P™ used here, have an internal one (Atmel Cooperation, 2014*a*). Additionally the ATmega328P™ accepts external RTC sources. In this thesis two external sources will be used. Just a generic $32.678KHz$ oscillator on the one hand, and a RTC able to produce a $32.678KHz$ signal like the DS3231, by Maxim Integrated Inc., on the other hand. The benefit of using the external source over the internal is the ability to

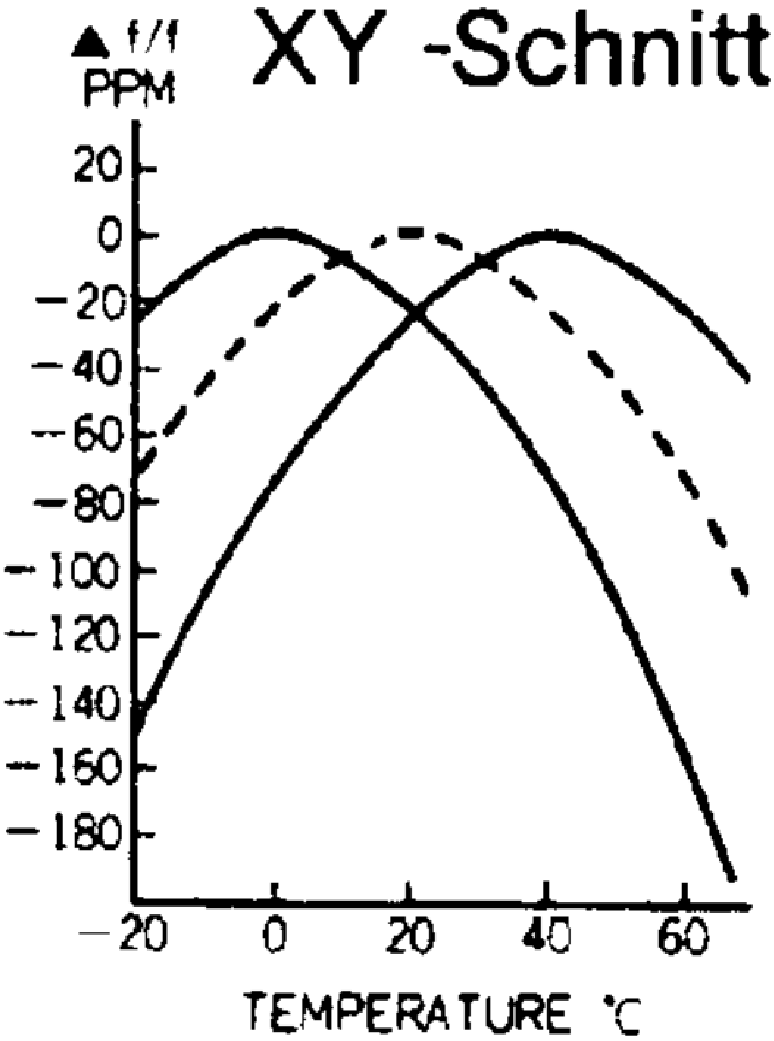


FIGURE 2.3: Temperature effect on the frequency of oscillators. Figure from Rein Elektronik (2015) datasheet. The used 32.678 KHz oscillator is represented by the dashed line.

shut the microcontroller almost completely down, as it doesn't need to produce a RTC signal.

Unfortunately, the nominal frequency f_0 is temperature depended (Frerking, 1987, chapter 4). This is of special interest in environmental applications where the temperature might change over a range of 40 Kelvin and possibly more. Most oscillators are designed to produce a stable f_0 at 25 °C but show falling frequencies with increasing or decreasing temperatures. This behavior is shown in figure 2.3, taken from Rein Elektronik (2015) datasheet, where the used oscillator is represented by the dashed line. Both, the internal oscillator and the RTC oscillator in the DS3231 are of the same type and follow the same dependency. Figure 2.3 shows a negative parabolic function with a vertex of 0 ppm deviation at 25 °C and about 60 ppm deviation at the temperature range limits of −10 °C and 60 °C (Rein Elektronik, 2015). This can be expressed as equation 2.6, which would result in 61.25 ppm at the temperature range limits.

$$f_T = f_0(1 - 0.05ppm(T - T_0)^2) \tag{2.6}$$

Where T is the actual temperature, T_0 is the nominal temperature for stable frequencies of 25 °C and f_T is the oscillator frequency at given temperature.
For temperature compensation, the frequencies can be reconstructed as the air temperature next to the datalogger is measured. In a second step the real time steps can be calculated as shown in equation 2.7 and 2.8:

$$\Delta f = f_0 - f_T \tag{2.7}$$

$$t_c = t_i - \frac{\sum_0^i \Delta f}{f_0} \tag{2.8}$$

Where Δf is the difference of the nominal frequency and the temperature compensated frequency calculated using equation 2.6 in Hz, t_i is the time step to be corrected, t_c is the compensated, real time step, and i is the amount of seconds t_i represents.
The compensated time steps, the time steps given by the already temperature compensated DS3231 (Maxim Integrated, 2015), and the time steps produced by the EM50 unit should have been compared. Unfortunately, it was not possible to operate the DS3231, due to different, in parts unsolved issues. The reasons are described in more detail in section 4.1 (see p.47). In consequence, the compensated time steps can only be compared to the original time steps based on the deviation from a manual reference time line. The temperature measurement from the weather station run by the Weinbau Institute in Freiburg will be used for the compensation0. This station not located on, or next to the study site, but it is the closest third party weather station, where the data is free available.

The master time series that should have been used to give the exact time of measurement should have been produced by the datalogger units with the DS3231 mounted. According to the datasheet, this ultra precise, temperature compensated RTC has a deviation of 2 minutes per year (Maxim Integrated, 2015, p.9). Then the residual time step offset to the oscillator driven data logger and the EM50 can be calculated. The difference of measurements per time unit will be used to calculate a mean deviation per measurement being equal in all measurements taken during the observed time unit. The time unit shall be chosen big enough to include at least 100 measurements. If there are too little measurements, the deviation will not sum up to a complete additional measurement. E.g. using a time step t of 5 Minutes, the time unit i can be estimated like:

$$100 * t = i = 100 * 5min = 500min$$

This are approximately 8 hours. The data loggers won't even be read on a daily basis, therefore there will be enough measurements within each time unit. The mean deviation can be calculated by exactly noting the time of reading and restarting any data logger unit. Exactly the same hardware (Laptop) will be used for reading the loggers any time and the system clock will be synchronized with the atomic clock in Braunschweig, Germany[6]. Both time series will be handled as if they were predicted time steps of the master time series and therefore the deviation of each will be evaluated using the RMSE for predicted values like defined in equation 2.9:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{j}(n(t_{c,i}) - n(t_{m,i}))^2}{j}} \tag{2.9}$$

Where i is the number of observed time unit from 0 to the total amount of time units n present in the time series, $n(t_{c,i})$ is the number of compensated time steps in i and $n(t_{m,i})$ is the corresponding number of time steps in the master time series.

As the deviation of the 32.768 KHz oscillator might be as high as 65 ppm, an expected deviation in time steps can be estimated, using the worst case scenario of persistent 65 ppm deviation. The deviation per second is then

$$32768Hz * \frac{65}{1000000} = 2.13Hz$$

In order to get a full second of deviation, 32678Hz have to sum up.

$$32768\frac{Hz}{s} * 2.13^{-1}Hz^{-1} = 15384s = 256.4min$$

[6]Timeserver service of the Physikalisch-Technische Bundesanstalt (PTB) in Braunschweig, Germany. URL: http://www.ptb.de/cms/en/ptb/fachabteilungen/abtq/fb-q4/ag-q42/time-synchronization-of-computers-using-the-network-time-protocol-ntp.html. Accessed: November 23, 2015.

This means about every 4 hours, the oscillator driven datalogger can deviate by one second. Using the chosen time step t in seconds, the expected deviation Δt per time step can be calculated using equation 2.10 and the needed time t_e to get an extra measurement using 2.11.

$$\Delta t = \frac{t}{15384} s \tag{2.10}$$

$$t_e = \frac{t}{\Delta t} \tag{2.11}$$

Taking the time step of 5 minutes from above into consideration, the Δt of only 0.02 s might seem negligible, but after only $t_e = 250min$ the data logger will produce a extra time step.

raw data

The ATMega328P® includes a 10 bit resolution ADC (Atmel Cooperation, 2014*a*) which is also described in more detail in section 3.3 (see p.37). Consequently the values which are saved to the flash memory are ADC values which can easily be translated to voltages, as the ADC gives the part of measured voltage compared to the reference voltage on a scale of 0 to 1023. Equation 2.12 shows the transformation of ADC values to the applied voltages.

$$V_{in} = \frac{A * V_{REF}}{1023} \tag{2.12}$$

Where V_{IN} is the actual raw data value, A is the ADC value from flash memory, V_{REF} is the reference voltage and 1023 the 10 bit resolution of the ADC. The V_{REF} used in this thesis was of a constant 1.1 V, 3.3 V or V_{BAT}, the actual supply voltage. In order to keep the data in this thesis consistent, all data produced by the custom data logger will be kept in the database application *Openhydro* as voltages and denoted as raw data.

2.3 Hydrological Connectivity

The hydrological connectivity will be estimated on event basis in order to track changes in the catchment connectivity over time. This way, it may be possible to relate connectivity changes to different conditions prior or during the rainfall-runoff event.

2.3.1 Discharge Data

The discharge data, that was intended to be used was somehow corrupted as it did neither match rainfall events nor did it show discharge dynamics that could be meaningful in any way. In consequence, stages data was used to calculate discharge data using a weir formula as presented reported by Aigner (2008) and shown in equation 2.13:

$$Q = \frac{8}{15}\mu\sqrt{2g}\ tan(\alpha)\ (\frac{h}{100} - w)^{2.5} \tag{2.13}$$

Where Q is the discharge in $\frac{l}{s}$, μ is the discharge coefficient of 0.67, which is depended on the opening angle α of 22.5° and the weir width b of 0.25 m. The weir height w is 0.04 m and h is the measured stage in **cm**. The script applying the described method can be found in appendix D on page 125.

2.3.2 BFI & runoff coefficient

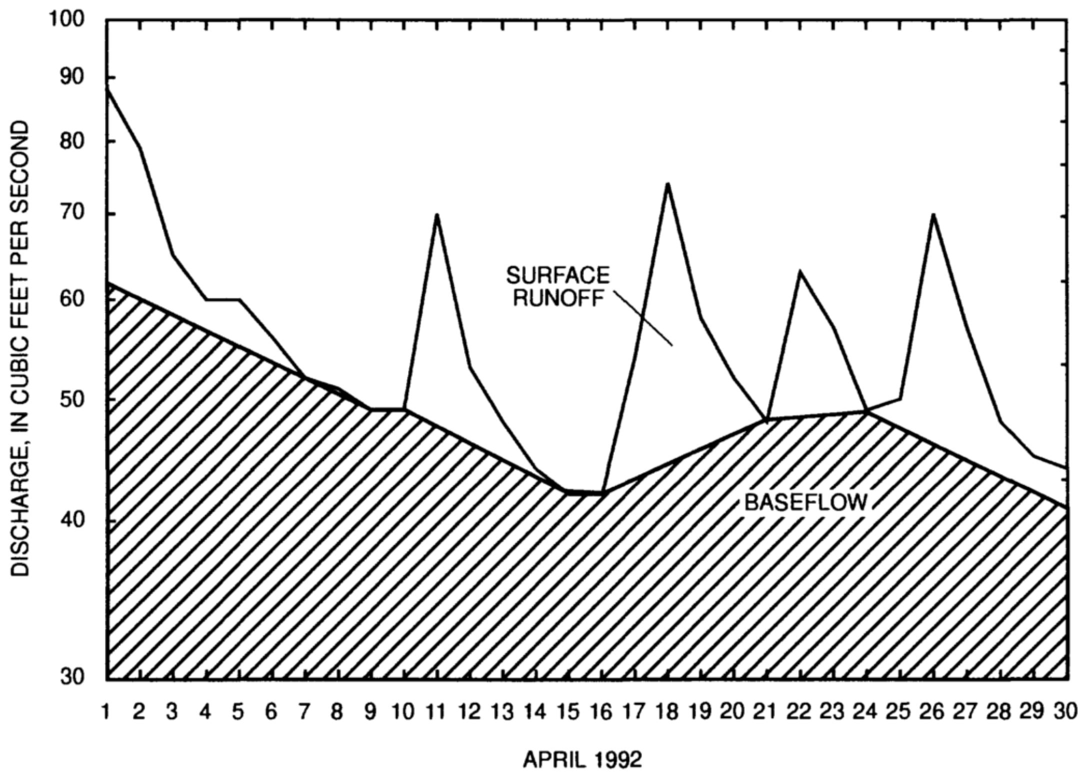


FIGURE 2.4: Hydrograph separation using the local minimum method near French Creek, Phoenixville, USA. Figure copied from Sloto and Crouse (1996).

The BFI is a dimensionless ratio of discharge from the aquifer Q_B, or more general referred to as *slow* discharge components or baseflow, and *fast* responding event discharge or direct runoff Q_E (Eckhardt, 2008). Here, the BFI will be defined as shown in equation 2.14.

$$BFI = \frac{Q_E}{Q_B} \tag{2.14}$$

As most of the analysis tools used in this thesis are implemented in the python programming language, the *rolling_minimum* function from the *pandas* package can be used to calculate the base flow. This function implements the method described by Sloto and Crouse (1996) as the 'local minimum method' for hydrograph separation and BFI calculation and is illustrated in figure 2.4.

The discharge at each time step is checked to be the lowest value within a moving window w described in equation 2.15:

$$w = \left[-\frac{2N-1}{2}, +\frac{2N-1}{2} \right] \tag{2.15}$$

Where N is the window size in time steps. For sizing the window, Sloto and Crouse (1996) suggested equation 2.16, Blume et al. (2007) adapted this suggestion to equation 2.17.

$$N = A^{0.2} \tag{2.16}$$

$$N = 0.827 * A^{0.2} \tag{2.17}$$

Where A is the catchment size in km^2. N in equation 2.16 and 2.17 will be given in days, though, this N has to be multiplied by 24, as the time series use a time step of one hour in this thesis. As stated in the site description (see 2.1; p.10), the catchment has a size of $0.37\,\mathrm{m}^2$. Multiplying equation 2.16 by 24 gives a window size of $24 * 0.37^{0.2} = 19.7$; for equation 2.17 the window size will be $19.7 * 0.827 = 16.3$. As Blume et al. (2007) developed this adaption especially for small catchments, a window size of 16 time steps will be used.

The BFI should not be confused with the *runoff coefficient* which is in general a ratio between a considered discharge volume and a precipitation volume (Zillgens et al., 2005, Blume et al., 2007). Here, the event discharge Q_E is considered and related to the event precipitation P_E, therefore, following Blume et al. (2007) and incorporating equation 2.14 the runoff coefficient C_R is calculated using equation 2.18:

$$C_R = \frac{Q_E}{P_E} = \frac{BFI}{Q_B * P_E} \tag{2.18}$$

A high C_R is an indicator for a well connected catchment, because a high proportion of precipitation becomes discharge within a short period of time . In opposite, low C_R indicate a unconnected catchment.

2.3.3 spatial patterns

Spatial pattern within the soil moisture in a given area at a given point of time could be identified visually by applying a number of spatial interpolations to the measuring points. A suitable method is ordinary kriging. When preparing the necessary data for an ordinary kriging approach, the semivariance S^2 has to be calculated for each point. Instead of applying an interpolation to the data, the semivariance can also directly be used to process information about spatial pattern in a more automated way, which won't need a subjective visual interpretation of the results. This approach will be presented here.

The semivariance is the half, mean, sqaured Euclidean distance between two measurements, and was first described by Yates (1948) and in detail by Curran (1988). S^2 for two soil moisture measurements Θ_i and Θ_j can be calculated by equation 2.19, following the remarks of Curran (1988):

$$S^2 = \frac{1}{2}\left[\Theta_i - \Theta_j\right]^2 \tag{2.19}$$

The distance between the locations both measurements were taken will be given in *lags* h, with $h \in \mathbb{N}$ in m. This step is necessary to normalize the distances between the measurements and form averages for each lag. Otherwise two measurements will most unlikely have exactly the same distance. As S^2 was calculated for all possible pairs of measurements, N is the amount of pairs found for a given lag h. Then, following Curran (1988), the mean, unbiased semivariance $\bar{S}^2$ for lag h is calculated by equation 2.20:

$$\overline{S}^{\,2} = \frac{1}{2N}\sum_{i=1}^{N}[\Theta(x_i) - \Theta(x_i + h)]^2 \tag{2.20}$$

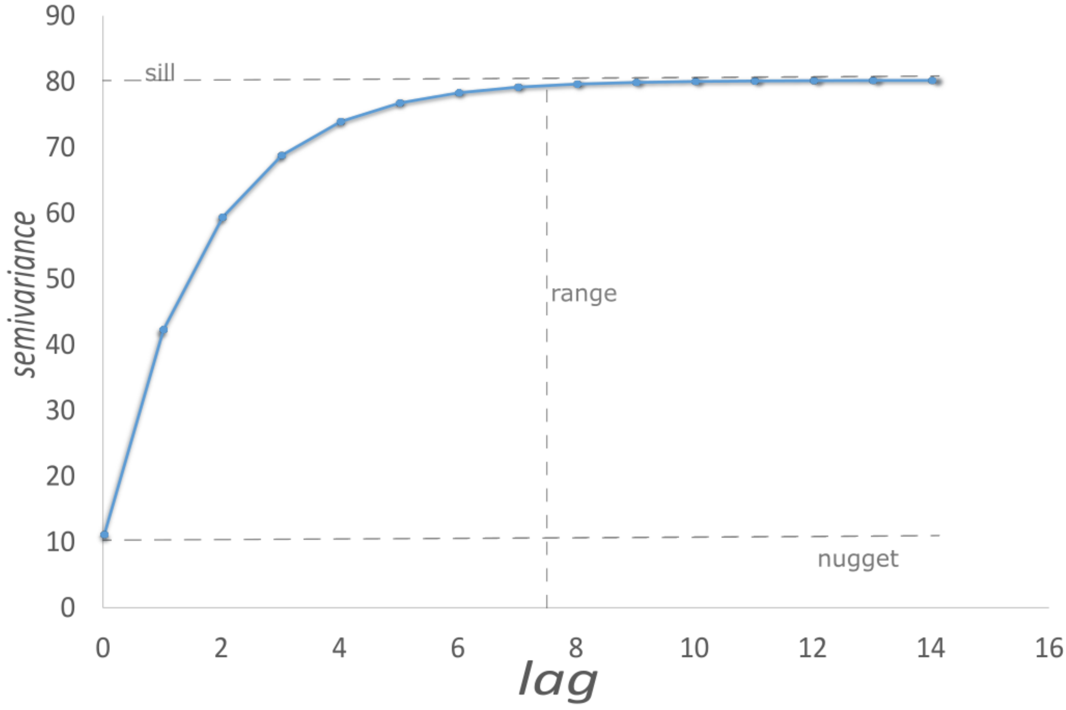


FIGURE 2.5: Sample variogram showing the typical limited growth of semivariance with increasing lag until the limit (*sill*) is reached. The lag value where the semivariance is hardly changing called *range*. The variance, which cannot be described spatially is called *nugget variance*.

All $\bar{S}^2$ for all observed h can be plotted over h, this is called a *variogram* or *semivariogram*, which is basically the same. High values of $\bar{S}^2$ indicate small spatial correlation between the measurements at the given distance (Curran, 1988). A variogram is described by three parameters, the *nugget, sill* and *range*, which are shown in figure 2.5. The sill is the limit in semivariance, which will not be exceeded, the nugget is the quantity of variance, which cannot be explained spatially and the range is the lag at which the variance does not change significantly. The range can also be used as an indicator of soil moisture pattern (Meyles et al., 2003), where similar range parameters observed over time indicate similar spatial distribution of soil moisture measurements within a catchment.

This thesis will try to detect relations between the range of soil moisture to the BFI and runoff coefficient within the catchment. This would suggest a well-connected catchment.

The change of this relation over time shall be visualized and tested for trends within the data.

For preparing the the semivariogram, a distance map is needed. This matrix will give the distance for each measuring point to any other in meter. This could be calculated by the haversine formula (Sinnott, 1984), but as all points are very close to each other on a global scale a different approach will be used. Each points coordinates will be transformed from unprojected WGS84 datum to Gauss-Krüger Zone 3 UTM coordinates. These are given in meter. Consequently, the distance can then be calculated using the formula of Pythagoras like shown in equation 2.21

$$d = \sqrt{(E_2 - E_1)^2 + (N_2 - N_1)^2} \tag{2.21}$$

Where d is the distance in meter, E is the Easting and N the Northing of coordinate 1 and 2 in meter. This distance matrix has to be further simplified into a *lag-matrix* giving the correct lag $h \in \mathbb{N}$, by cutting off all decimal points.
The range time series for the given soil moisture time series can be calculated on the same temporal resolution, which would be associated with great computational effort, or any higher aggregation level. The raw data has 2 minute resolution and a hourly range time series will be aggregated in order to increase the sample sizes. For each hour within the measuring campaign, the mean soil moisture value from each point will be used to calculate $\bar{S}^2$ using equation 2.20 for each lag. $\bar{S}^2$ will be visualized in a single image where one pixel on the x-axis represents one lag h and one pixel on the y-axis represents one time step, here one hour. The pixel value is determined by the RGB value of a color on a 'Red-Yellow-Blue' color bar ranging from $min(\Theta)$ to $max(\Theta)$.

In order to obtain a range value for each hour within the campaign, a model will be fitted to each variogram. This model can easily be described and the range easily determined from a mathematical point of view. The first model to be fitted is the spherical variogram model, described by a 2^{nd} order Bessel-function, like shown in figure 2.5. This model can be described by equation 2.22, which is a simplified version of Jian et al. (1996)'s Spherical function (referred to as model 1):

$$\overline{S}_s{}^2(h) = \begin{cases} C_0 * \left(\frac{3}{2}\left(\frac{h}{a}\right) - \frac{1}{2}\left(\frac{h}{a}\right)^3\right) & \text{if } h <= a \\ C_0 & \text{if } h > a \end{cases} \tag{2.22}$$

Where $\bar{S}_s^2$ is the modeled, mean, unbiased semivariance for given lag h, C_0 is the sill and a is a shape factor.

The second model is a modified version of 2.22 like shown in 2.23:

$$\overline{S}_{ms}{}^2(h) = \begin{cases} C_0 * \left(\frac{3}{2}\left(\frac{h}{a}\right) - \frac{1}{2}\left(\frac{h}{a}\right)^3\right) + b & \text{if } h <= a \\ C_0 + b & \text{if } h > a \end{cases} \tag{2.23}$$

Where b is a y-intercept and can be used to fit the nugget.

The third model is natural exponential model described by equation 2.24:

$$\overline{S}_e{}^2(h) = C_0 * \left(1 - e^{\left(-\frac{h}{a}\right)}\right) \tag{2.24}$$

Where C_0 is again the sill, a the shape and h the lag.

This model is a simplified version of 2.25, also taken from Jian et al. (1996) (here referred to as model 2).

$$\overline{S}_{es}{}^2(h) = C_0 * \left(1 - e^{\left(-\frac{3h}{a}\right)}\right) \tag{2.25}$$

The fourth model is a gaussian model fitted to the data, taken from Jian et al. (1996) (here referred to as model 4) shown in equation 2.26 and a simplified version of 2.26 shown in 2.27:

$$\overline{S}_g{}^2(h) = C_0 * \left(1 - e^{\left(-3*\frac{h^2}{a^2}\right)}\right) \tag{2.26}$$

$$\overline{S}_{gs}{}^2(h) = C_0 * \left(1 - e^{\left(-\frac{h^2}{a^2}\right)}\right) \tag{2.27}$$

Each of these model functions will be fitted to the semivariance using the least square method with a maximum of 1000 fitting iterations for each model and time step.

$$L = \sum_{i=0}^{n} \left(f(\overline{S}_i{}^2, \alpha) - \overline{S}_{i,x}{}^2\right)^2 \tag{2.28}$$

Where L is the squared error, $\bar{S}_i^2$ is the semivariance at time step i, $\bar{S}_{i,x}^2$ is one the modeled semivariance with $x \in [s, ms, e, es, g, gs]$ and α is the set of the corresponding model parameter, so a and C_0 for the gaussian, exponential and spherical and a, C_0 and b for the modified spherical model. The α, where L has the smallest value will be applied to the model.

The range r for each model function and time step is the lag h of first maximum $\bar{S}_{i,x}^2$

occurrence. Beside the range time series, a sill, RMSE and residual time series will be calculated. The sill s is the $\bar{S}^2_{i,x}(r)$ for each model and time step. The RMSE can be calculated using equation 2.29:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}\left(\overline{S}_i{}^2 - \overline{S}_{i,x}{}^2\right)^2}{n}} \tag{2.29}$$

Where n is the number of time steps in the time series. Finally the residuals of the applied model functions are defined like equation 2.30:

$$rr = \overline{S}_h{}^2 - \overline{S}_{h,x}{}^2 \tag{2.30}$$

Where h are all lags present for the actual time step. The absolute, mean value of rr defined as $\overline{||rr||}$ is calculated for each model function and time step.

The calculated RMSE and mean residual values will be compared to the precipitation and the original soil moisture time series in order to determine if there is a fitting quality dependence on environmental parameters. The models can be compared to each other using the RMSE. This error values makes the goodness of fit for each model comparable. For choosing a model, following Jian et al. (1996) the best goodness parameter is the Akaike information criterion AIC, as the "AIC takes into account not only the goodness of fit, but the parsimony of the model as well" (Jian et al., 1996, p.5). The AIC is calculated as shown in equation 2.31:

$$AIC = n * ln\left(\frac{R_m}{n}\right) + 2p \tag{2.31}$$

Where n is the number of points in the specific variogram, p is the number of parameters and R_m is the sum of the square of differences, which can easily be derived during $RMSE$ calculation.

For applying a model function, α, the set of parameters, need a initial guess, which are as follows:

$$\begin{aligned} C_0 &= max\left(\overline{S}^2(h)\right) \\ a &= \overline{h} \\ b &= 1 \end{aligned}$$

For further investigations the modeled semivariance will be visualized in an image like described above for the calculated semivariance. Additionally a semivariogram of measured and modeled values will be plotted for each model and time step.
For choosing one model over the others the one with the smallest AIC will be used. In order to clarify if there are significant differences in the goodness of fit between the chosen model and the others, the AIC cannot be used as the AIC values are not meaningful. In consequence, a Kruskal-Wallis-Test (Kruskal and Wallis, 1952) will be performed on the RMSE as it has meaningful values directly linked to the residuals, which do describe the difference between observation and model values.
The correlation between the range time series derived from this chosen model, by taking the range value for each time step, is then compared to the measured catchment response described by the runoff coefficient as described above. This comparison is executed by a cross-correlation like used in signal processing and defined by equation 2.32:

$$(f \star g) = \int_{-\infty}^{\infty} f^*(t)\ g(t)\ dt \tag{2.32}$$

The cross correlation $(f \star g)$ is defined as the integral of all differences of two time depended signals $f^*(t)$ and $g(t)$. Divided by the amount of observations, the cross-correlation is normalized to a 0 to one scale and the mean value ρ of all correlation values describes the similarity on the same scale, with a ρ of 1 describing two identical signals.

2.4 Test Scenarios

Actual measurements were taken in this thesis in different scenarios. This can include a short-timed test in the laboratory or a long-lasting measuring campaign. Some of the scenarios were evaluated and based on the conclusion the next scenario was planed. Therefore it might be necessary and sensible to read the results in chapter 4 (see p.47)

and the corresponding conclusions in chapter 6 (see p.106) of a specific test scenario before continuing to the methods of the next scenario. In some cases the methods were changed from one scenario to the other and therefore the following sections do only apply to the results and conclusions related to them. This will be stated in the single result sections.

2.4.1 Battery Lifetime

In order to use the data logger in the field, the battery lifetime has to be tested. This is the time, the battery is still charged enough to produce a stabilized voltage supply on the board. As the voltage falls below 2.8 V the flash memory might behave unsuspected. To evaluate this behavior is also part of this test, in order to be able to identify this error in the field data. The device board was produced exactly like shown in appendix E (see p.132) and programmed with the core firmware described in appendix F section F.1 (see p.134). The main firmware application is shown section F.2 (see p.135). This application will log the hexadecimal ASCII sign 0x43 representative for the letter ***'C'*** to the memory every 15 minutes. By setting different preprocessor variables the firmware will dump the memory content to the serial port, which can easily be saved into an ASCII file. The main objective of this test is to evaluate the battery lifetime in hours and conduct a recommended time span for battery changes for this thesis, from this result. The results can be found in section 4.2.1 on page 49.

2.4.2 Battery Characteristic Curve

A battery characteristic curve is a graph of battery voltage decreasing over time as the battery is used. The manufacturer will supply these graphs created at the laboratory at 25 °C and with constant current consumption. For this thesis, several battery characteristic curves are produced, at room temperature and in the field. The power consumption will be increased by decreasing the time step and keep the logger awake for a few hundred milliseconds. Other curves will be generated without setting the microcontroller to sleep. This will empty the battery within a few hours. The curves are generated several times with different durations and will then be aggregated to a single mean characteristic curve by normalizing the time. These characteristic curves will be needed to evaluate the ADC values and check the raw ADC value data for battery voltage bias. It is expected, that no correlation between the measurements and the characteristic curves is found. In order to make the measurements comparable and as the main objective of this test scenario is not to evaluate the battery lifetime, the time component will be normalized throughout all characteristic curves. Therefore the time will be given in percent of lifetime and as a consequence will be of the same length. As especially the measurements in the field might contain artifacts and the measurements get very noisy at low battery voltages, a polynomial function will be fitted to each characteristic curve. The expected function can be defined as shown in equation 4.2:

$$V(t) = a * t^3 + b * t^2 + c * t + d \tag{2.33}$$

Where $V(t)$ is the battery voltage at time step t and a, b, d, c are fitting parameters. These are again fitted by the least square method (see 2.28; p.27) with a maximum of 100 fitting iterations.
In case the used *DataLogger Micro* units used for soil moisture measurements differ in battery lifetime dramatically, the mean battery characteristic curve can be used to calculate residual values and relate their maxima occurrences to climatic parameters.

2.4.3 Accuracy & Precision

The sensor accuracy and precision test is performed in the laboratory. A flowerpod filled with a soil sample from the test site is saturated, weighted and kept at constant 40 °C inside a drying cabinet. Two soil moisture sensors are installed inside the pod at the same depth, a *DataLogger Micro* and a Decagon® EM50™ with a single 5TE sensor connected. The pod will be weighted twice a day and the water loss will be calculated. After a period of three days measuring time the soil sample will be completely dried and weighted a last time for calculating the effective porosity Φ of this soil sample. Using the weight at saturation and complete dryness, the soil water content can be calculated for each weight taken.
The standard deviation σ of each logger measurements will be interpreted as sensor precision, as it gives the mean deviation from a true value unregarded of shifts in mean value. Residuals of the *DataLogger Micro* and Decagon® values compared to the calculated soil moisture will be evaluated over time. Their mean value are an indication for shifts and therefore interpreted as sensor accuracy.

As there will be just a handful of calculated water content values, a time series will be modeled using an exponential approach like shown in equation 2.34. These functions reflect the expected soil moisture graph.

$$\Theta = a * e^{-c*100x} + b \tag{2.34}$$

Where Θ is the water content value to be modeled; a, b and c are unitless fitting parameters and x is the normalized time step giving the point of measurement on a 0 to 1 scale. It was necessary to multiply this scale by 100, in order to avoid very small values in the exponent. Without this factor the modeling results got significantly worse. This function is again fitted by the least square methods within 10000 iterations. These large amount of iterations was chosen to compensate for the lack of initial guesses for the fitting parameters.
In order to ensure statistical significance for this test scenario, various test will be applied to the resulting data. An Shapiro-Welk test (Shapiro and Wilk, 1965) will clarify if the residual values of each logger's measurements are normally distributed. In case the null hypothesis of normal distribution cannot be rejected a paired Student's t-test (Gosset aka. Student, 1908) will be used to test the EM50™ and *DataLogger Micro* for a true difference in mean values. The non-parametric alternative is the Wilcoxon signed-rank test for paired samples (Wilcoxon, 1946), which should not be confused with the Wilcoxon rank sum test, sometimes also reffered to as the Mann-Whitney-U test (Mann and Whitney, 1947). As the measurements in both samples are paired, neither the MWU test (non-parametric) nor the Welch t-test (parametric) can be used.

Due to a lack of time and money, this test cannot be repeated various times, in order to evaluate an reliable precision value. Therefore this test scenario will focus on determine whether the *DataLogger Micro* significantly differs from the 5TE's precision, which has been evaluated by the manufacturer. The precision values, expressed as the standard deviation of the two samples will be tested for equality based on the same test for normal distribution. The non-parametric case will be covered by the Levene test (Levene, 1960), the parametric alternative is the Bartlett test (Snedecor and Cochran, 1989). In fact, the Bartlett test can also be applied on non-parametric samples, but in these cases the Levene test is more robust.

Chapter 3

Sensor Network

3.1 Overview

The term *Sensor Network* includes different hard- and software components in this thesis. The hardware includes three component types: sensors, data loggers and web servers. For soil moisture measurements two types of sensors were used. One of them was commercial, the other one self-build. Each sensor type required a different type of data logger, therefore two types were deployed. Collected data was stored in a database which is running on a remote machine. In order to ensure data safety two servers were used, a commercial and a open source solution.

Along with the collection of necessary data for this thesis the second objective was to evaluate the open source based parts of the sensor network. This network is understood to be the prototype for an 100 % open source, solar driven and reproduceable measuring network, which components are available all around the world. By limiting the costs it could empower scientists in developing countries in particular, to increase their data availability.

The following sections will introduce all network components in detail. The network was not entirely developed as an object of this work. A draft of some components was developed prior to this work and not entirely by the author of this work. Therefore this section will also clarify, which parts of the network are developments of the author and whether they are part of this thesis or not. This includes hard- and software components.

3.2 Data Logger

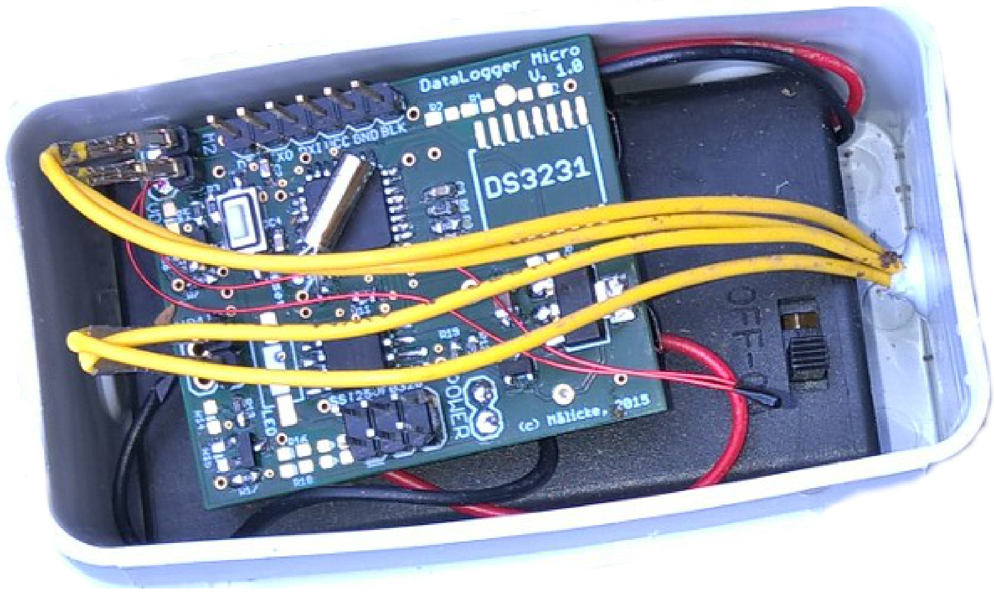


FIGURE 3.1: A mounted and deployed *DataLogger Micro* including a bigger battery pack and two sensors connected (yellow cables). This photo was taken during the measuring campaign between October 28, and December 15, 2015.
© Mirko Mälicke, 2016.

A data logger is defined to be a microcontroller based electrical unit able to read different sensors and save these readings over time, or pass them to another central saving unit like a web server. The data logger used in this thesis is a unchecked, unevaluated schematic and layout for a PCB [1], which was mounted and evaluated during this thesis. The unit is called *"DataLogger Micro"* and the used schematics and layout can be found in appendix E (see p.132). Figure 3.1 shows one of the loggers deployed on the study site during the field campaign. The one shown has two soil moisture sensors and one temperature sensor connected and is powered by an extra, external battery pack (shown underneath the PCB).

3.2.1 Hardware

From a technical point of view this datalogger is described in detail in the datasheet (Mälicke, 2016). The most significant technical specifications are shown in table 3.1.

[1]**PCB** - Printed Circuit Board. Explained in the glossary in appendix A (p.115).

Table 3.1: The most important technical specifications of the *DataLogger Micro* as used in this thesis. The shown informations are all from Mälicke (2016).

Specification	Value	Description
Microcontroller	ATmega328p®	datasheet Atmel Cooperation (2014*a*).
clock speed	*internal* 1 MHz	
external oscillator	32.768 KHz	used as optional real time clock.
real time clock	DS3231SN®	high precision real time clock used as reference; datasheet Maxim Integrated (2015).
memory chip	SST25VF032B	32 MBit (4 MB) serial memory array; datasheet Silicon Storage Technology (2006).
battery	LIR2032	any compatible 20 mm diameter and 3.2 mm height button cell at 3.6 V. **the one-way CR2032 has only** 3 V**!**.
sensors	5	2x soil moisture, 2x resistance comparator, internal supply voltage.
ADC resolution	10 bit	V_{REF} at 2.54 V, therefore resolution of 2.4 mV. Detailed description in Atmel Cooperation (2014*a*).
battery lifetime	6 months	estimated using nominal consumption, with 15 minute time step.
logging capacity	209,000 datasets	with all 5 measurements per time step (and 4 B per measurement).

The main design objectives for this data logging unit were size and price on the one hand and power consumption on the other hand. One would consider a microcontoller board as *"Arduino compatible"* as it can be programmed using the Arduino IDE[2] and Arduino programming language. The Arduino community is one of the biggest open source hardware communities and a device, which is compatible, has a big potential supporting community. Unfortunately the *DataLogger Micro* is not compatible, as the Arduino IDE requires the ATmega328p to be run from a external oscillator. As shown in table 3.1, the *DataLogger Micro* is driven by an internal 1 MHz oscillator, although it could be as fast as 20 MHz from an external source. The main reason was to decrease the power consumption as it falls to 1.5 µA instead of 5 mA at 20 MHz (Atmel Cooperation, 2014*a*).

The *DataLogger Micro* comes with two assembly variants for the RTC (see appendix

[2]Arduino official homepage with descriptions and download of the IDE and references for the Arduino programming language. URL: https://www.arduino.cc/en/Main/Software. Accessed: November 22, 2015.

E, p.132). This means either the DS3231 or a 32.768 KHz oscillator can be used. At high number of peaces the oscillators cost only a few cents[3], while the DS3231 is several magnitudes more expansive[4]. The DS3231 is extremely accurate and can be used as a reference for time step quality checks (see section 2.2.2, p.17), or if measurements at the second and sub-second temporal scale are needed. This thesis will evaluate the oscillators comparing to the RTC and the EM50 as a cost efficient commercial solution, when the temperature compensation done by the DS3231 automatically is applied to the oscillators. Therefore, the *DataLogger Micro* will be used in both variants, at the same location.

3.2.2 Software

The *DataLogger Micro* is a AVR-Microcontroller driven data logger and can therefore be programmed in the AVR-C or BASECOM-AVR language. Here, the AVR-C language was chosen. As different use cases are necessary in this thesis, different firmware versions have to be implemented. A usual approach would be the development of a powerful configurable firmware, which can be adapted by externally adjusting different parameters in the microcontroller's memory. There was not enough time to develop a professional firmware during this thesis and this untested complex firmware can be a additional source of error. It might also be hard to debug, therefore a different approach was chosen in this thesis. Nevertheless, a firmware as described will be developed after this thesis. The firmware used here was split up into two parts, called the *core firmware* and *main firmware*. The core firmware is a collection of different modules integrating core functionality like reading ADC-values, dumping the flash or power management, which are used in exactly the same version for all use cases. The main script running during a test is then called main firmware, containing only one file. This is the main application using the core firmware files. This main application defines the time step or the amount of sensors used. This way, although the different test scenarios need different applications, the functionality of the logger is the same throughout the whole thesis. Therefore differences in saving or rounding values by the microcontroller or errors

[3] see for example: 0.13 e pp. for 100 pieces at Conrad.de.
URL http://www.conrad.de/ce/de/product/168467/Quarz-fuer-allgemeine-Anwendungen-Frequenz-327680-kHz-Bauform-TC-38-x-H-3-mm-x-6-mm?ref=searchDetail. Accessed: November 22, 2015.

[4] DS3231 description by the manufacturer MAXIM integrated for 7.25 $ pp. for 100 pieces.
URL: http://www.maximintegrated.com/en/products/digital/real-time-clocks/DS3231.html/tb_tab3. Accessed: Novemver 22, 2015.

in the firmware can be excluded between the test cases. All used firmware scripts are described or shown in appendix F (see p.134).

3.3 Sensors

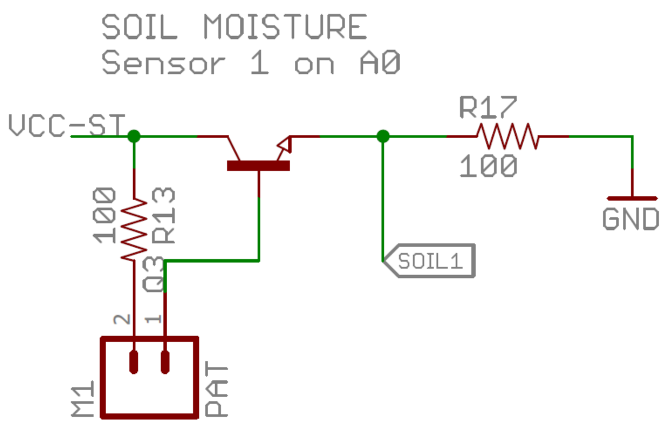


FIGURE 3.2: Snapshot of the DataLogger Micro schematics (see figure E.1 for complete schematics) showing the circuit for the *M1* labeled sensor unit. This is the internal sensor part, strengthening the soil signal in order to make it readable by the unit.

The soil moisture sensors developed for the original version of the data logger are not a single unit, but split up into a circuit on the data logger board and external components connected to the board. The external components did not exist prior to this thesis and were therefore not tested before. The data logger includes two connectors for soil moisture measurements, which are labeled *M1* and *M2* on the board layout shown in figure E.1 (see p.132). Both connectors include two pins, which shall be connected with a electrode each (as shown in figure 3.3). These electrodes are then deployed into the ground. The circuit is shown in figure 3.2 and is mainly made up of a transistor and two resistors. As the *VCC-ST* stabilized voltage supply of 3.3 V is turned on, it is applied between the two electrodes only connected by the soil. The wetter the soil is, the smaller is the soil resistance and the less this voltage will be decreased. This decreased voltage will not be read directly, but control the transistor. If the transistor is completely open due to very low soil resistance (saturated soil) the voltage of 3.3 V is decreased at R13 and then divided by R17 in order to reach exactly 1.1 V at saturation. This is the internal reference voltage for the ADC (Atmel Cooperation, 2014*a*, p.281, table 26-3, REFS0 and REFS1 bits set to 1.) and therefore the maximum voltage, which can be

FIGURE 3.3: Sensor electrodes prototypes as used in this thesis.
© Mirko Mälicke, 2015.

measured. A photo of the test determining R17 is shown in appendix G (see figure G.2; p.141). This illustrates the effort necessary for setting just a single resistor value.
In the course of sensor calibration the ADC has to be calibrated to suppress electrical noise within the measuring unit. This will be achieved by taking multiple measurements within a few milliseconds and averaging them. This has to be done independent from any data aggregation on second or minute time scales. The calibration has to find a meaningful amount of measurements, which cancels noise but at the same time still ensures immediate feedback to changing conditions. As an experimental design, six soil moisture sensors were connected with their corresponding electrodes (the "left" and "right" ones) wired together. Then, an external voltage of 0.43 V was applied between the "left" and "right" electrodes. Then the voltage was increased step wise to 0.54 V and 0.64 V. This was repeated two times and took place within just a few seconds. The time step was chosen to 200 ms, which should result in a total duration of about 3 s. This way, each sensor will measure exactly the same value from a electrical point of view. Any difference in the measurements result from different aggregation levels in the ADC. The chosen voltages should be represented by the ADC values of 400, 500 and 600.

The ADC calibration curves for this logger and sensor unit are shown in figure 3.4. The lines represent the values taken at different aggregation levels. Therefore each time step is presented in any line as a single point, but the amount of single measurements used to

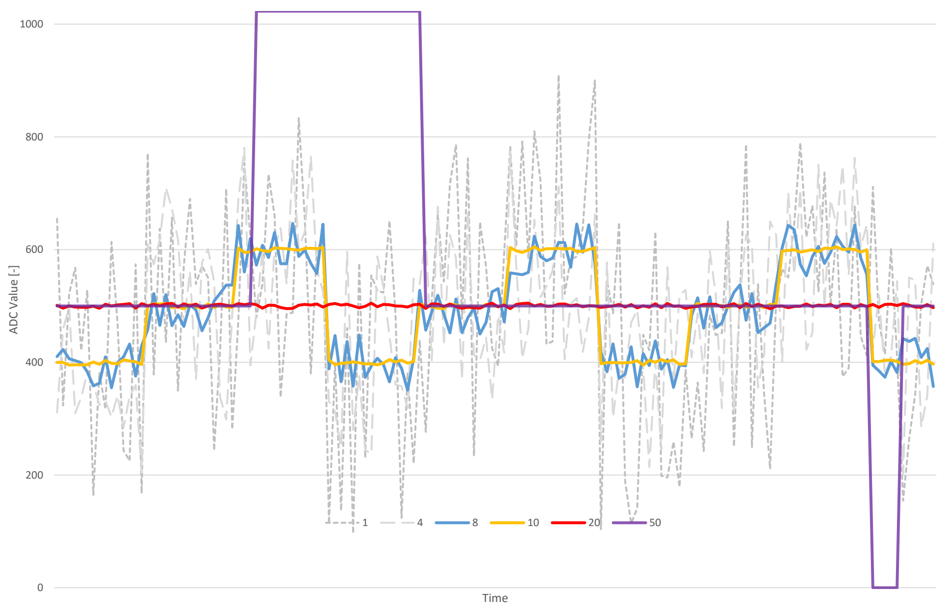


FIGURE 3.4: Six DataLoggerMicro Units were used with one soil moisture sensor conected and wired to each other. With physically measuing exactly the same values, the logger did only differ on thier ADC aggregation level. All measurings were taken within a few seconds. The lines show the internal ADC value over time. The excact aggregation level is given as legend entry.

compute the value differs. The two gray lines represent measurements without aggregation (points, labeled "1") and low aggregation level of 4 (dashed). The "1" line is closer to a complete random measurement of values between 200 and 800 than the real sample values. For the dashed line one would still need a portion imagination and optimism to interpret it as the expected step function. This function can be observed for the "8" (blue) and "10" (yellow) aggregation levels. Both show the expected values, but the "8" line is still very noisy. By increasing the aggregation level to 10, the measurements fit the expected values perfectly.

For higher aggregation levels, the signal get completely lost. For an aggregation level of 20, the reason might be overlapping measuring periods. As the time step was very short, a single measurement and its aggregation took longer than the time step period. Therefore only the main mean value of the whole test was measured. This overlapping effect got even worse for the 50 aggregation level. Here, it is believed, that a single aggregated measurement overlapped with a number of other measurements, which lead to unpredictable behavior. As the ADC can only take one measurement at a time, wrong register values might have been read. At level 50, the maximum value of 1023 and the minimum value of 0 was also measured. An aggregation level of 10 might seem to be the perfect choice under the given circumstances. It has to be considered that higher aggregation level result in longer measuring times and therefore in higher power consumptions. The voltage regulator has a nominal power supply of 1 A. The sensors are run at 3.3 V and following figure 3.2 (see p.37) the dropping resistor of 100 Ω will result in a power consumption of:

$$I \quad = \quad \frac{U}{R} \quad = \quad \frac{3.3V}{100\Omega} \quad = \quad 0.033A \quad [= 33mA]$$

As a consequence, the measuring time shall be kept as short as possible. In conclusion it can be stated, that the aggregation should be kept as low as possible, but as high as necessary to match sensor precision requirements. Additionally, if expected battery lifetimes are undercutting extremely, the dropping resistor value has to be increased by one or two potencies, as long as the measured values are preserved.

During the first field test, while all other sources of increased power consumption were identified and cleared off, another test was conducted for estimating the dropping resistor R13. Two data logger units with two soil moisture sensors connected to each were applied into a soil sample. The sensors were read and the distance between the two electrodes was increased until no more sensor readings were possible. After this test the soil sample was dried and weighted. The gravimetric water content was 0.55 $g^1cm^3 * g^{-1}cm^{-3}$. Each sensor used a different dropping resistance value of 100 Ω, 1 kΩ, 10 kΩ and 100 kΩ. The results of this test are given in table 3.2.

TABLE 3.2: Different dropping resistance values (R) for R13 with the calculated power consumption (I) and maximum soil moisture electrode distance (D), at which sensor readings are still possible; given in mm.

R	I	D
100 Ω	33 mA	> 50 mm
1 kΩ	3.3 mA	20 mm
10 kΩ	0.33 mA	3 mm
100 kΩ	33 µA	N.A.

For 100 kΩ no sensor readings were possible, although the electrodes touched each other. A reason might be the current leakage over resistors, which might be higher than the calculated I. In this case the current was transformed to heat. Although it might be desirable to decrease the current as low as 0.33 mA, the resulting D of 3 mm maximum would cause unmanageable problems during sensor installation, as they shall not touch. Therefore the appropriate dropping resistance value was set to 1 kΩ.

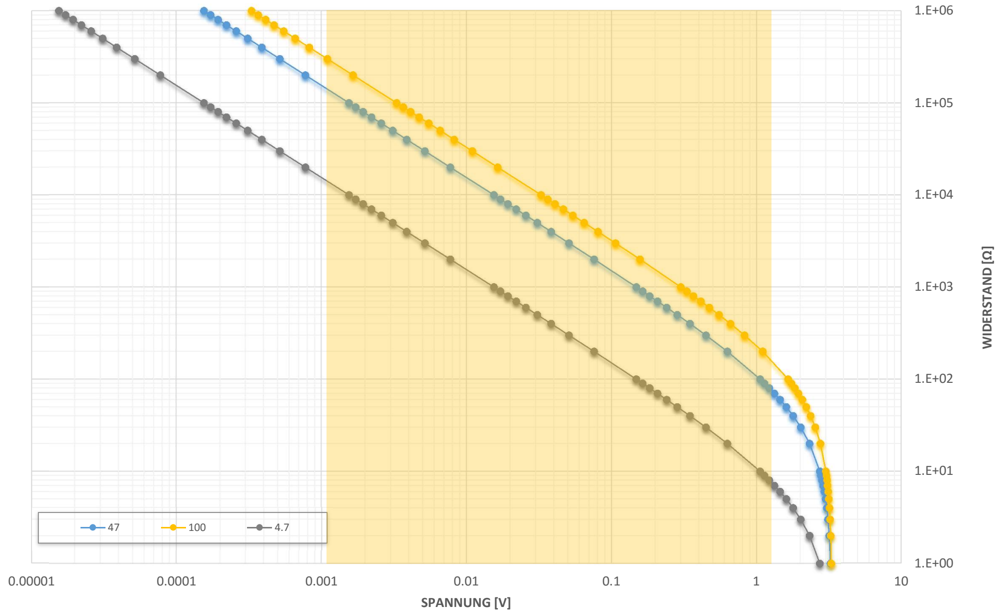


FIGURE 3.5: Voltage - Resistance diagram for the sensor circuit on the *DataLogger Micro* board. The three curves show the voltage to be expected for the corresponding resistance values in the soil. The yellow window indicates the measuring range of the ADC. The different colors indicate different comparator resistors used, such as 4.7 Ω (red), 47 Ω (blue) and 100 Ω (yellow).

Beside the dropping resistor labeled R13 in figure 3.2 (see p.37), there is also the second resistor called R17. This resistor will split the measured voltage by its relation to the value of 100 Ω. The need of a transistor was described earlier. Therefore, for the given circuit the signal has to be split in a way, that the maximum value of SI3.3V is still mapped to 1.1 V, while the dry soil measurements (with high resistance and small voltage) are still measurable. This was tested with three different resistance values mounted onto R17, 4.7 Ω, 47 Ω and 100 Ω. The relation between voltage and resistance for a resistance range of 1Ω to 10^6Ω is shown in figure 3.5. The yellow box indicates the voltage range measurable by the used ADC. Some field measurements in another soil type and other location were taken during testing and building the data loggers. The soil moisture was manually estimated to be close to field capacity. Using a typical multimeter the soil resistance was measured directly and these measurements averaged to some 10.000 Ω. Therefore the range of 10^4 to 10^5 should be covered by the yellow box. This is not the case for the red line representing a splitting resistance of 4.7 Ω. Finally the 47 Ω (blue) was chosen over 100 Ω (yellow), because it still covers the desired range, but reaches lower resistance values within the yellow box. This will result in slightly better resolution in wet soil, which are most likely expected during the measuring campaign.

3.4 Server - heart of the network

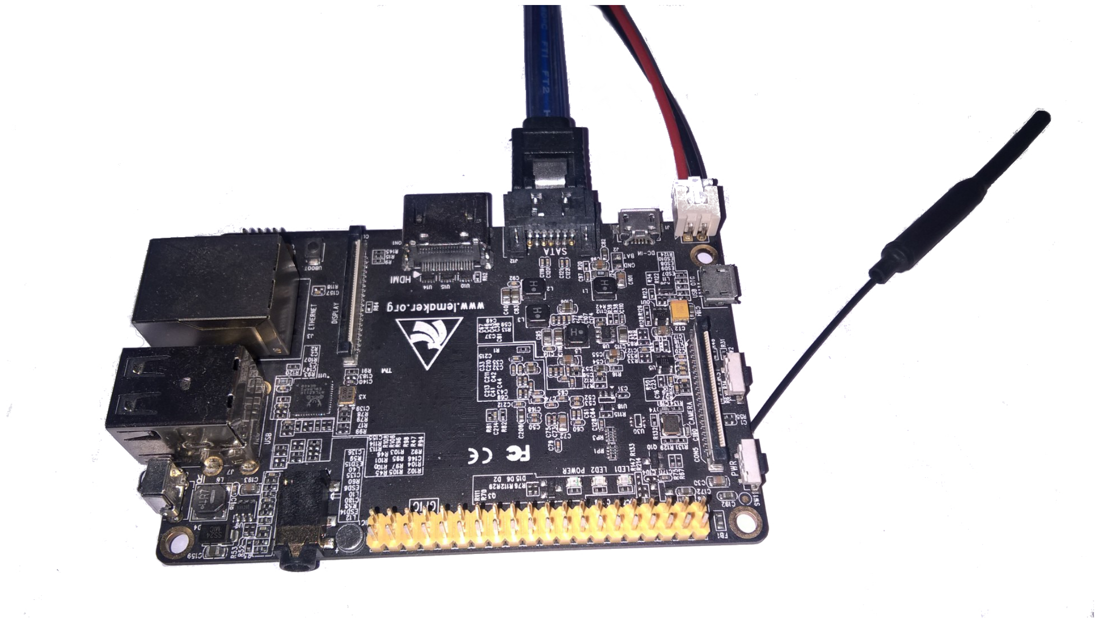


FIGURE 3.6: Banana Pro™ used in this thesis as an experimental data service server. The shown unit has an inter WiFi antenna for communication and an external SSD connected for storing data.

In 2012 the Raspberry Pi Foundation from Great Britain launched a small and cheap single-board computer, called Raspberry Pi. Until February 2015 over 5 Million devices were sold[5]. During these years of success for the company and the device, a lot of derivates evolved from the original board. The complete Raspberry Pi project was released under an open Source License, meaning anybody could reproduce and advance it. Most operation systems for the Raspberry Pi are based on Unix systems like Ubuntu or Debian, which are also published under open Source Software. These circumstances

[5]Twitter posting by Raspberry Pi Foundation.
URL: https://twitter.com/Raspberry_Pi/status/567708532334530560. Accessed: November 20, 2015.

brought many derivates to the market during the last three years. For this thesis the Banana Pro™ by the LeMaker®[6] Organisation is used and shown in figure 3.6.

TABLE 3.3: Choosen Specification of Banana Pro™ from LeMaker® Homepage.

Specification	Value
CPU	ARM® Cortex™-A7 Dual-Core 1GHz (ARM v7 instruction set)
RAM	1GB DDR3 (shared with GPU)
Wifi	WiFi 802.11 b/g/n
Storage	MicroSD card, SATA 2.0
USB	2 x USB 2.0 host, 1 x USB 2.0
Size	92 mm x 60 mm
Weight	45 g

Compared to the Raspberry Pi Model B+, which was a comparable computer as of this writing, the Banana Pro™ comes with doubled RAM (1 GB) and a faster 1 Ghz Dual-Core CPU (see Table 3.3). Another major improvement is the SATA 2.0 connector and the associated ability to use SSD. While the described sensor network was developed, the Raspberry Pi Foundation came up with the Raspberry Pi 2[7] , except the SATA 2.0, exceeding the performance characteristics of Banana Pro™ again.

As shown in table 3.3, beside a Dual-Core processor, the Banana Pro™ offers the latest DDR3 memory technology. In combination with a 2.5" SSD the system offers fast accessible disk space and computation capabilities at low cost. The Wifi module is used to connect the system to the local network and use it in remote mode. This means the graphical user interface is not used, but the operation system is accessed by SSH or via the Python package *openhydro*, which was written by the author for exactly this purpose. This is introduced in the next section.

3.5 Web and Data Services

Section 3.4 gave an overview on hardware setup. This is the necessary infrastructure to run several software tools and applications, referred to as *services* in the following. Three different kinds of services are needed to run a productive and meaningful network:

[6]Banana Pro on LeMaker official website. URL: http://www.lemaker.org. Accessed: March 9, 2016.

[7]Raspberry Pi 2 announcement on official Raspberry website. URL: http://www.raspberrypi.org/raspberry-pi-2-on-sale/. Accessed: March 9, 2016.

- web service, for offering access to data products
- storage service, for structuring, saving and securing the data
- processing service, for manipulating and correcting data in order to produce data products

Storage and processing services can be summarized as data service as they will be provided within the same framework. As stated in the objectives (see 1.3) of this thesis, the complete software suite shall be licensed as open source. In fact, this is true for 100 % of the following services and all related packages.

3.5.1 Web Service

In order to offer software products like time series, graphs or maps, a web server is needed as basic infrastructure. The Apache 2[8] web server used in version 2.2 fitted best into the given infrastructure. It is open source and very well supported due to a very big community. In combination with the Apache the programming language PHP[9] (version 5.4) and the database system MySQL[10] (version 5.5) are installed as well. Both meet the requirements as the web services will be embedded into a Content Management System (CMS), which are usually written in PHP. The web services presented in this thesis are implemented in a productive Contao CMS solution[11]. This implementation and basic web service development was preliminary work, which is necessary but not part of this thesis. Both data services and the web service are based on Python scripts and classes. Especially the data services were extended by different necessary functions in this thesis, which will be presented in more detail in section 3.5.3 (p.45). For the Apache web server the WSGI[12] module was an prerequisite in order to enable the python interpreter.

[8] Official Apache Server Project Website. URL: http://httpd.apache.org/. Accessed: March 9, 2016.
[9] Official PHP Website. URL http://php.net/. Accessed: November 12, 2015.
[10] Official MySQL Website. URL: http://www.mysql.com/. Accessed November 12, 2015.
[11] Official Contao Website. URL: https://contao.org/de/. Accessed: November 12, 2015.
[12] Apache module mod_wsgi on GitHub including documentation. URL: https://github.com/GrahamDumpleton/mod_wsgi. Accessed: November 12, 2015.

3.5.2 Storage Service

Apart from the MySQL server, which is part of the Web services (see 3.5.1), a PostgreSQL Server[13] , version 9.1, is installed as data server. PostgreSQL is used over MySQL for several reasons. Running a data server as a server instance on its own will keep website maintenance and website content data apart from the measured data. This is very important in order to keep the data clean and avoid mixing measurements with data presentation. This might simplify a future port or clone of the data server into another environment. Second, PostgreSQL can be extended by the PostGIS[14] package. This offers spatial functions and makes storing and querying geospatial data possible. Lastly, PostGIS can be integrated into existing GIS applications like Quantum GIS[15] . This feature offers a easy, fast and direct access to the database.

3.5.3 Data Service

Beside specialized data products, like a specific time series with given aggregation level, start- and endpoint, a sophisticated interface to the database will be programmed. For this purpose, the Openhydro database system will be used and extended to fit the data service requirements for this thesis. Openhydro is open source and was mainly developed by the author of this thesis, therefore adaptions can most easily be integrated and will keep the workload for developing a database as small as possible, while still have a powerful database application available. The programming language Python will be used as an interface to the database, thus the interface will be represented by two Python packages:

- **openhydro** - Input/Output streams to the database, will manage the database communication.
- **hydras** - Python package pandas[16] wrapper[17] , for data manipulation, calculations and conversions to other formats like csv, xls and many more.

[13]Official PostreSQL website with PostgreSQL 9.1 documentation under –documentation –9.1 in the main menu. URL: http://www.postgresql.org/. Accessed: November 12, 2015.

[14]Official PostGIS project website. URL: http://postgis.net/. Accessed: November 12, 2015.

[15]Official Quantum GIS website. URL: http://www.qgis.org/en/site/. Accessed: November 12, 2015.

[16]Pandas package website. URL: http://pandas.pydata.org/. Accessed: March 9, 2016.

[17]Wrapper in this context means: The hydras package is operating pandas in order to simplify the handling for the user. Therefore hydras is not really extending pandas, but making it more accessible for hydrology-related tasks.

TABLE 3.4: Data Service Python package releases on GitHub. These are the links to the exact versions used in this thesis.

Package	**Link**
openhydro	https://github.com/lordblaupause/openhydro
hydras	https://github.com/lordblaupause/hydras

As of this writing, all three packages are and will be under active development. They are published under CC BY-SA 4.0 license[18] and are available on GitHub (www.github.com). All packages are developed and maintained by the author of this thesis, therefore a version of each package is published called *Master Thesis* containing the exact version used in this thesis. The links to these specific package versions can be found in table 3.4. The development of any of these packages started prior to this thesis. Beside a versioning on GitHub, all changes on the packages, which are part of this thesis can be found in the appendix. In general, any commit published on Github between the date of this thesis beginning (the 15th September, 2015) and the commit called "Master Thesis" is in fact part of this thesis.

[18]Creative Commons Attribution-ShareAlike 4.0 International License. Open Source license.
Summary: http://creativecommons.org/licenses/by-sa/4.0/.
Legal Text: http://creativecommons.org/licenses/by-sa/4.0/legalcode. Both accessed: November 12, 2015.

Chapter 4

Results

This chapter is divided into four main parts - custom sensor network overall performance, technical lab test, technical test site data and connectivity results. The first part is a description on the handling, simplicity, general issues, drawbacks or successes experienced while using the custom sensor network. The second part depicts the results from laboratory tests driven under known and controlled conditions. The third parts depicts the results from a technical point of view for the sensors used at the test site. The last part depicts the results referred to as data products, which were calculated from the measured data and deal with soil moisture pattern and hydrological connectivity within the site.

4.1 Overall Performance

In the first measuring campaign design of this thesis the fieldwork was scheduled to the beginning of October 2015. However, the first measurements in the field took place on October 28, 2015. The main reasons for this delay are described in section 4.2 underneath. Especially the way too high power consumption, in combination with a design error on the PCB in the original files from Mälicke (2016) caused a redesign of major parts of the PCB along with the delay for printing and mounting them. This is also the main reason, why only 9 units were yielded during the campaign. Another consequence which turned out to be a serious one, was the lack of time to develop a proper

configuration and management GUI for the *DataLogger Micro* devices. Without a software tool like the ECH2O Utility[1] for the EM50^{TM} devices the hardware handling can get complicated. Really complicated. The author wrote three different and independent firmware scripts for the *DataLogger Micro*, one for managing the data collection ('MoisutreToSST.c'; see F.2; p.136), one for producing a memory dump ('SSTToSerial.c'; see F.2; p.138) and sent it via a serial protocol to a connected computer and a third to perform a memory erase command ('Erase.c'; see F.2; p.137). Developing three different firmwares was much faster than one sophisticated as all three performed fundamentally different actions and some of the used libraries turned out to be not compatible (In detail: serial communication and power saving influenced each other).
One outcome was the need of in-field reprogramming of the custom devices. Some devices had temperature sensors connected and other didn't. Five devices connected one soil moisture sensor and four used two sensors, therefore slightly different firmware versions were necessary. The adaption of AVR-C code in the field has to be done very carefully as there is no option for debugging or feedback on success by the device[2]. One could not even tell if the device is working or not. In terms of usability this is the most serious downside of the custom system. In fact the author run the memory-erasing script before running the memory-dumping script in four cases, which obviously ended up in producing a data gap.

Following section 3.3 (see p.37) in chapter 3, the aggregation level of the internal ADC is a main reason for the resulting noise found in the data record. This was illustrated in figure 3.4 (see p.39). The blue line represents the aggregation level of 8, which was chosen here. The noise in figure 3.4 found in the blue line in ADC units is about 60 which is about 6 % of the whole measurable range. Translated to water content measurements 6 % of the measurable range is about $0.04m^3 * m^{-3}$, which can be expected as noise in the measurements. This is possible as the voltage divider resistors operating on the ADC were chosen to map the maximum value to $0.68m^3 * m^{-3}$. This lays slightly above the porosity on the study site.

[1] ECH2O Utility product page. URL: http://www.decagon.com/en/data-loggers-main/software/ech2o-utility/. Accessed: January 7, 2016.

[2] Especially with slopes of 40 %, temperatures below 0 °C and programming times of more than 3 hours on the study site this turned out be be a challenging task.

4.2 Technical Lab Tests

4.2.1 Battery Life

After a period of 10 days the device was first dumped and contained 390 savings of the correct saved sign (the ASCII number for **'C'**) and one saving of a wrong sign at the very end. This results in a battery life of $390 \ / \ 4 \ = \ 97.5$ hours. As the used battery has a capacity of 45 mAh, the average power consumption was $45mAh \ / \ 97.5h \ = \ 461.5\mu A$.

According to an application note by Atmel Cooperation (2014*b*), the manufacturer of the ATmega328P, the power consumption during power save sleeping mode shall be less than 10 µA. This application note example is driven at the exact same clock speed and voltage (1 MHz, 3.3 V) as used by the *DataLogger Micro*. Therefore the difference of 451.5 µA (97.8 %) is current leakage. Following Ohm's law (equation 4.1) the current I will decrease with increasing resistance R at constant voltage V.

$$I = \frac{V}{R} \tag{4.1}$$

Consulting the data logger board schematics on figure E.2 (see p.133) there are two direct connections of voltage supply (VBAT) and ground (GND) only separated by a resistor. The internal voltage divider including resistor R8 and R9 as well as the MOSFET circuit for turning down the voltage regulator (called LD117AS33TR) where the resistor R12 separates the MOSFET gate from ground. R8 and R9 sum to a total resistance of $39k\Omega \ + \ 62k\Omega \ = \ 101k\Omega$ and R12 is of $10k\Omega$. In order to decrease power consumption R12 was raised to $1M\Omega$ and both, R8 and R9, were raised by one potency to $390k\Omega$ and $620k\Omega$, respectively. A further raising of these values might be questionable, as smaller currents are more error-prone in measurement and the MOSFET discharge time (for turning it off) increases. If the MOSFET needs more time for discharging, it will consume more energy.

A parallel resistance increase by one to two potencies should result in a power consumption reduction of one to two potencies and therefore in theory a power consumption of $10\mu A \ + \ 45\mu A$ to $10\mu A \ + \ 4.5\mu A$ is expected. This would result in a battery life of one to four months, which cannot really be tested within the given time frame.

In order to avoid data losses, the batteries will be changed once a month and be tested

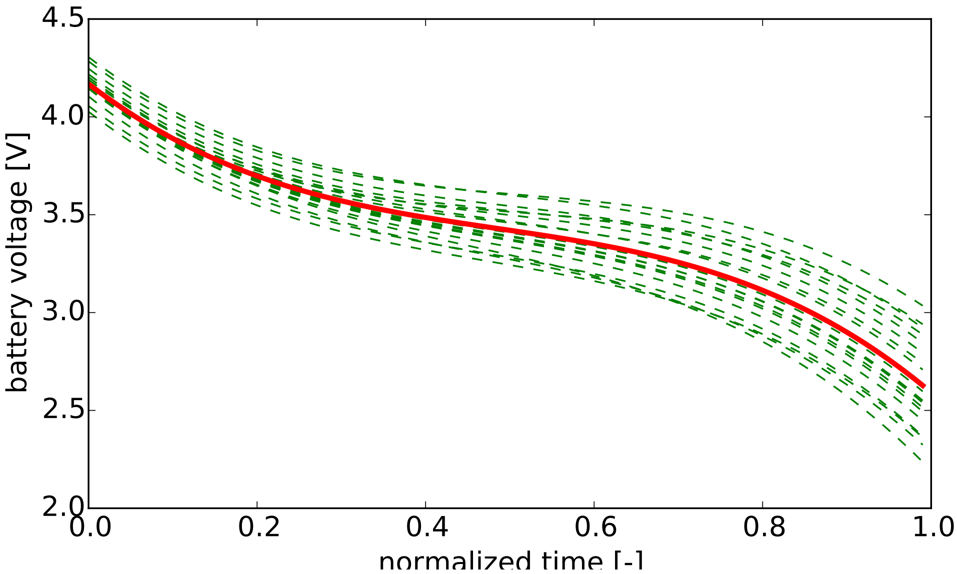


FIGURE 4.1: battery characteristic curve from 20 different measurements in different test scenarios, with stationary and instationary conditions. Each line represents a 2^{nd} grade polynomial function fitted to the measurements plotted over a normalized time line in percent of lifetime. The red line represents the mean battery characteristic curve.

for their capacitance. Additionally five of the devices will receive an external battery pack with about 1200 mA capacity in order to exclude system failures based on wrong assumptions concerning the battery lifetime.

4.2.2 Battery Characteristic Curve

As described in section 2.4.2 (see p.30) a battery characteristic curve was measured in the field as well as in the lab. This data was further described by various models in order to generate a overall, mathematical describable expected battery characteristic curve. This is necessary for debugging field observations, as the battery might not empty or behave as expected. Further this model is needed to evaluate the power consumption of the system. The result is shown in figure 4.1. The green, dashed lines represent one battery characteristic curve model fitted to one curve taken under changing conditions. Figure 4.1 illustrates the range of different emptying behavior based on different conditions and consumption rates. The red line represents the mean value of all fitted values and is therefore not associated to a single measuring campaign. This final, empirical fitted battery characteristic curve optimized in this thesis and shown in red can be described by equation 4.2. This equation can be used to reconstruct the capacity and the consumption rate for each device in retrospect.

$$V_{BAT} = -3.766 * t^3 + 5.488 * t^2 - 3.3 * t + 4.168 \tag{4.2}$$

4.2.3 Accuracy & Precision

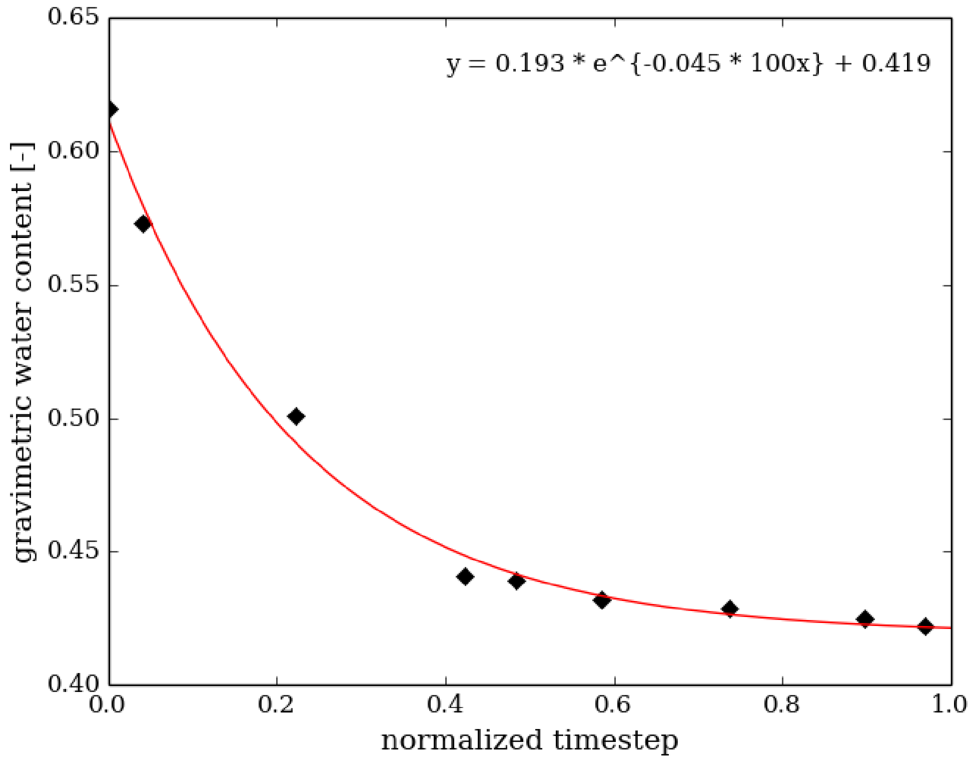


FIGURE 4.2: Calculated water content values (black points) and fitted exponential function (red line). The time is given as normalized timesteps, as the amount of total test time elapsed. Test was run from December 7, 2015 to December 9, 2015.

The sensor accuracy and precision assignation was executed as described in section 2.4.3 (see p.31). The accompanying code can be found in appendix D in listing D.1 (see section D, p. 121).

This test was performed from December 7, 2015 to December 9, 2015. The soil sample was weighted as often as possible, the results as well as the water content in g and $m^3 * m^{-3}$ are shown in table 4.1.

TABLE 4.1: Weights of the soil sample used in the accuracy and precision test scenario described in section 2.4.3 (see p.31). The water content in grams is the differenct to the weight after drying (4455 g), the gravimetric water content the fraction of the absolute water content to the sample weight at saturation (13 103 g).

date	**sample weight** [g]	**water content** [g]	$[m^3 * m^{-3}]$
07.12.2015 10:00	12526	8071	0.616
07.12.2015 12:00	11963	7508	0.573
07.12.2015 21:00	11019	6564	0.501
08.12.2015 07:00	10233	5778	0.441
08.12.2015 10:00	10207	5752	0.439
08.12.2015 15:00	10115	5660	0.432
08.12.2015 22:30	10076	5621	0.429
09.12.2015 06:30	10023	5568	0.425
09.12.2015 10:00	9984	5529	0.422

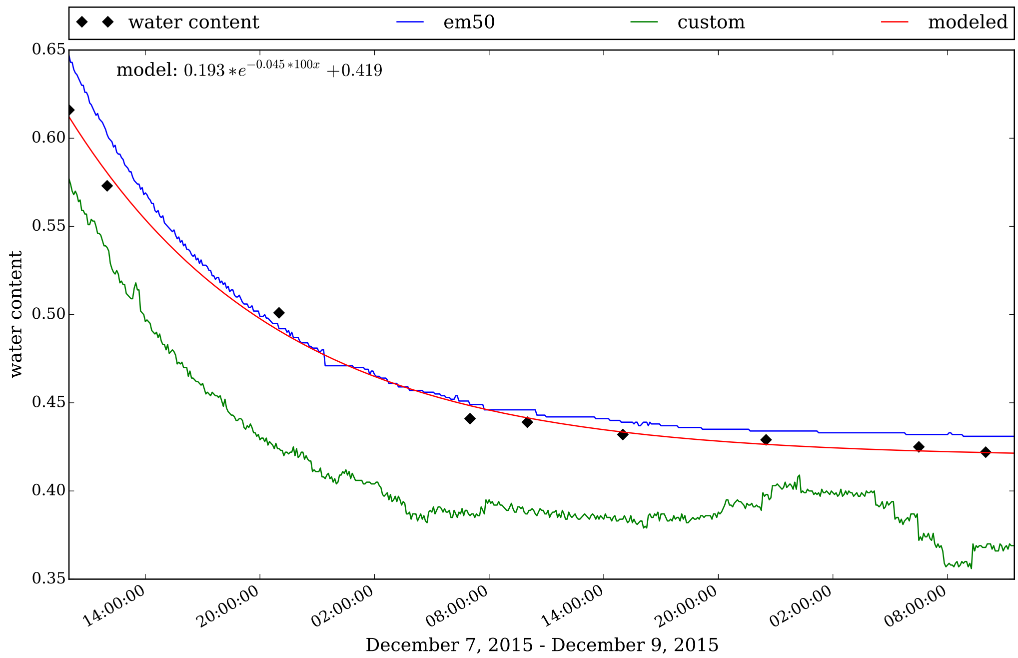


FIGURE 4.3: The result of sensor accuracy and sensor precision test. A exponential model (red line) was fitted to reference water content measurements (black dots). The 5TE sensor by Decagon ® (green line) and *DataLogger Micro* (blue line) measured the water content within the same flowerpot at the same depth.

The equation 2.34 (see p.32) was fitted to the gravimetric water content on a normalized time axis, again by the least square method described earlier. The fitting worked quite well as shown in figure 4.2. Here, the measurements are shown as black dots, whereas the red line represents the fitted model. Using this as the real reference water content inside the flowerpot makes it possible to compare EM50$^{\text{TM}}$ and *DataLogger Mirco* and evaluate their sensor accuracy and precision. These results are shown in figure 4.3. Beside the reference measurements from figure 4.2, which were re-projected to the real time axis, the results from the two sensor systems are shown, as well. The 5TE measurements generally fit very well to the real measurements. Not only the absolute values, but also the drying dynamics inside the flowerpot were measured very well, although these measurements were not calibrated by the author for this test. However, the results from *DataLogger Mirco* differ. After applying the initial water content conditions to the measured ADC values, the real water content values were underestimated. The higher drying rates until December 8, 2015 in the morning can also be observed, but the more stationary conditions in the second test half do not match. The custom system seems to be much more unstable in its measurements. Also, only by observing figure 4.3, the 5-minute measurements from the custom system seem to be significantly more noisy than the commercial system.

The sensor precision as described in section 2.4.3 (see see p.31), was specified using listing D.1 (see p.121) as shown in its output dump below:

```
Precision (Standard Deviations):
EM50:  0.052            Custom:  0.046
SNR with mu of model as expected value:
EM50:  8.88             Custom:  9.96

Shapiro Test
p: values < 0.05

Levene test for equal variances
p: 0.0460
H0 rejected, true difference in variances.
```

The commercial system measured with a standard deviation of 0.052 while the custom system showed only 0.046. Following Bushberg et al. (2006) the standard deviation can be used to calculate the signal-to-noise ratio (SNR), a feature used in medical imaging or physics to relate the noise of a measurement to the signal. Values higher than 1 point to a higher signal share, than noise overlay. Both systems show medium SNR values being smaller than 10 (this would equal 10 % noise). To evaluate a real difference between both systems, a Levene (Levene, 1960) test for equal variances was performed. A Levene Test was necessary as normal distribution could not be assumed due to null hypothesis rejection by the Shapiro-Welk (Shapiro and Wilk, 1965) test. As the null hypothesis of equal variances was rejected by the Levene test, there is a true difference in variances. Therefore, there is also a true difference in standard deviation (and SNR) as all three are proportional to each other.

However, in figure 4.3, the custom measurements (green) showed more noise. The standard deviation could be overlayed by the first test half, where the conditions where instationary, but the *DataLogger Micro* measurements were distinctly smaller. Therefore the tests were performed again only for the second test half, by adding the line below to listing D.1:

```
df = df['201512080800':]
```

Now, the precision specifications change like:

```
Precision (Standard Deviations):
EM50:   0.005             Custom:  0.011
SNR with mu of model as expected value:
EM50:   94.85             Custom:  33.95

Shapiro Test
p: values < 0.05

Levene test for equal variances
p < 0.001
H0 rejected, true difference in variances.
```

These results fit the visual inspection of figure 4.3 way better. The sensor precision for the commercial system is as small as 0.005 and the custom sensor precision is specified with 0.011. Derived from the SNR values, the commercial measurements include almost no noise (about 1 %), while the custom is more noisy (with about 3 %).

Aside from the sensor precision, the sensor accuracy was specified. This is the shift in mean measured values to the real mean. Therefore the residuals were calculated between the two measurements and the modeled real values. These residual values are shown in figure 4.4, where the Decagon® residuals are shown in the upper chart in green and the *DataLogger Micro* residuals in blue in the lower chart. Ignoring the first two hours, the commercial system never exceeds an accuracy worse than 0.02, which does match the accuracy of 0.02 to 0.03 specified in the data sheet[3]. The result of sensor accuracy specification is shown below as dumped by listing D.1:

```
Accuracy: (Mean Residuals):
EM50: 0.01                Custom: 0.05
```

[3]Decagon® 5TE official product page. URL: http://www.decagon.com/en/soils/volumetric-water-content-sensors/5te-vwc-temp-ec/. Accessed: December 14, 2015.

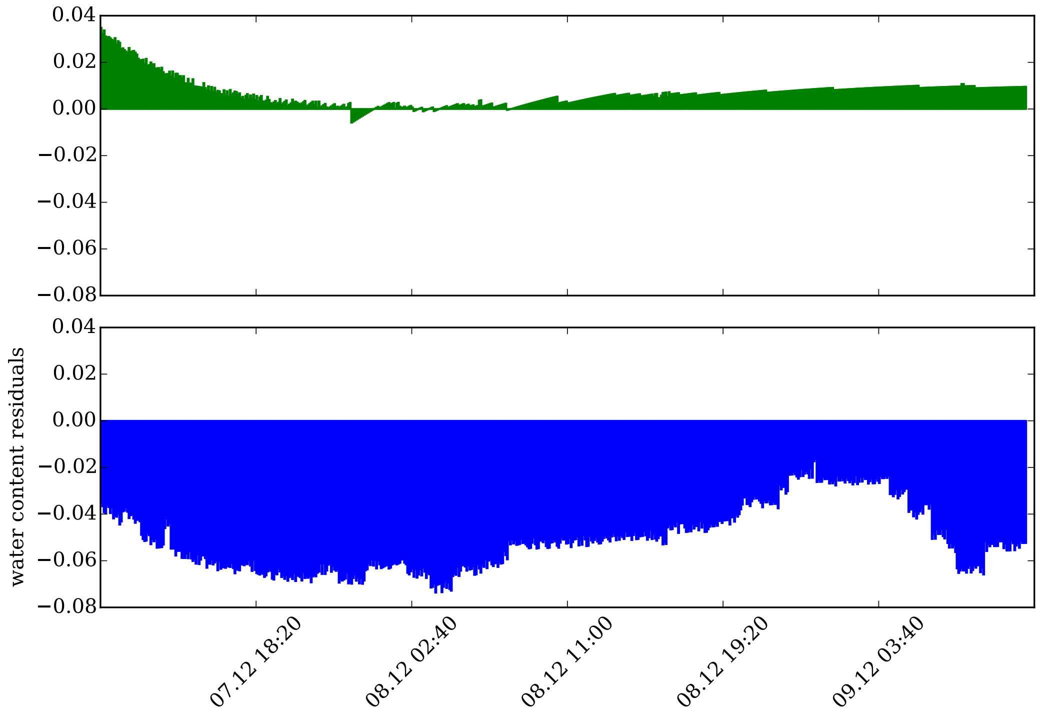


FIGURE 4.4: Accuracy and Precision test scenario measurement residuals. The Decagon® 5TE residuals to the modeled real water content in the upper chart and the *DataLogger Micro* residuals in the lower chart. Both charts share their x and y axis for illustration reasons.

```
Shapiro Test
p: values < 0.05

Wilcoxon signed-rank test for paired samples
p < 0.001
H0 rejected, real difference in ranks observed
```

The accuracy for the commercial system was specified to 0.01, while the custom system was as accurate as $0.05m^3 * m^{-3}$. Similar to the precision specification, the Shapiro-Welk test rejected the null hypothesis of normal distribution of the residuals. Therefore the non-parametric Wilcoxon signed rank sum test (Wilcoxon et al., 1963) was performed. The null hypothesis of both samples belonging to the same population was rejected. Thus, the two accuracies do differ significantly.

4.3 Test Site Data

4.3.1 Time Step Data

One objective of this work was (iv) - sufficient precision for scientific purposes. One scenario is the time step precision, as some scientific applications might be time depended. In order to evaluate the time step precision of the custom *DataLogger Micro*, the time step precision of the commercial EM50^{TM} system will be evaluated as described in 2.2.2 (see p.17).

EM50^{TM}

The first evaluation of the used time steps by the EM50^{TM} was not possible in detail for all units as one (labeled as Au3) produced impossible time stamps. The time steps from installation on October 28, 2015 until first data reading on October 31, 2015 where all dated incorrectly and looked like shown below:

```
...
15/10/28 14:22  0.237 10.7 0.02 ...
13/10/44 00:00  0.238 10.9 0.02 ...
13/10/44 00:02  0.237 10.8 0.02 ...
13/10/44 00:04  0.238 10.8 0.02 ...
13/10/44 00:06  0.238 10.9 0.02 ...
...
```

Either the format and time stamp itself changed or the time steps are completely corrupted. After installation (which was at 12:56 for this unit), the given time was correct, but after one hour of measurements the date format was changed by the EM50^{TM} from YY/MM/DD to DD/MM/YY and reseted to October 13, 1944 or 2044. Nevertheless, the measurements seemed to be correct, therefore the time steps were recreated by aligning the date range from installation to the number of time steps. As 2060 measurements were observed in this time period, the date range ended at October 31, 2015 9:34, but the real data was dumped at 9:40, which is a 6 minute offset. These measurements differed by 360 s within the real 247 560 s that passed. This equals:

$$\frac{360}{247560} * 10^6 = 1454ppm$$

This deviation is more than 20 higher than the expected worse case scenario of 65 ppm. Based on this information, it was suggested, that the EM50^{TM} does not compensate the time steps. In order to corroborate these suspicions, all time series were evaluated.

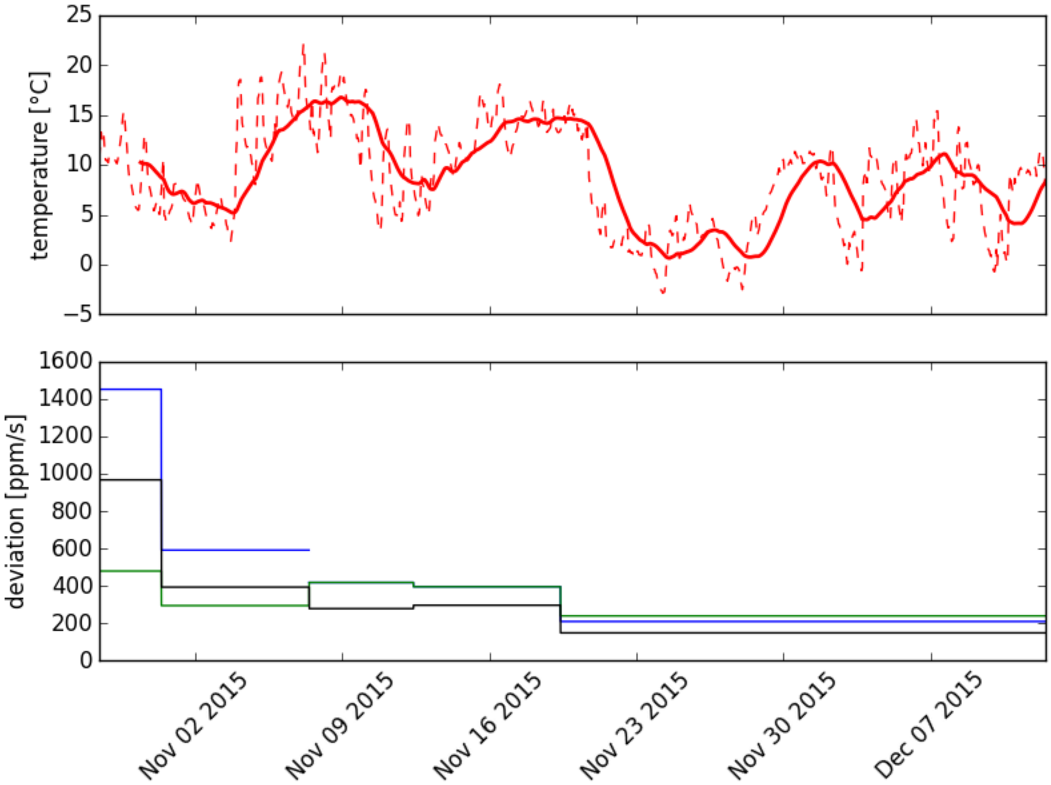


FIGURE 4.5: *Up:* Mean hourly temperature (dashed line) on the study site and 48 hrs-moving window mean (solid line). *Down:* Calculated RTC deviation per second in ppm for Au1 (blue), Au2 (green) and Au3 (black).

The deviation in time stamps was calculated and is given in ppm per second and plotted in figure 4.5 on the lower plot. Most observed deviations lay above 200 ppm and are at least more than three times as high as the worst case scenario of 65 ppm suggested. At the beginning of the measuring campaign, the highest values reaching deviations of the above described 1454 ppm. Although if not taking this logger into consideration, as the produced time stamps were corrupted as described above, the other EM50^{TM} units are still reaching values as high as 1000 ppm at beginning. All in all, the observed

deviations show a falling trend within the campaign and do not exceed deviations of 400 ppm after a few days of measuring. The upper chart in figure 4.5 shows the mean hourly air temperature measured at the Weinbau Institut Freiburg (WBI) as dashed line and their 48-hour moving window mean value as solid line.
In the observed period from October 28, 2015 to December 12, 2015 decreasing as well as increasing long term trends could be observed. This data is given for the same date range in order to relate the time stamp deviation to temperature trends. Neither the hourly data nor the rolling mean as a trend indicator show a relation to the observed deviations.

DataLogger Micro

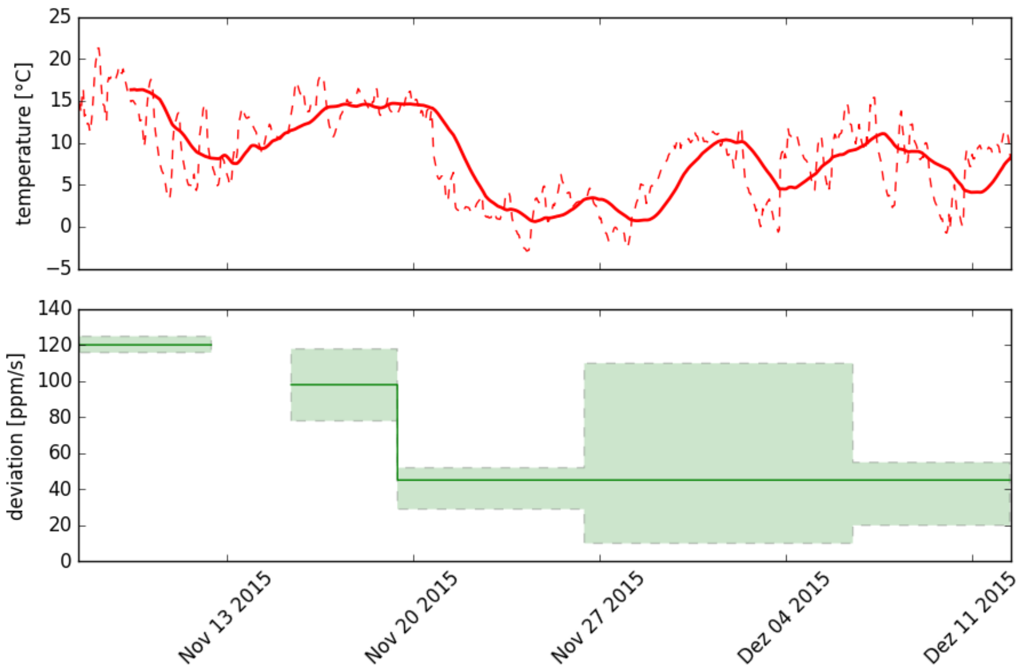


FIGURE 4.6: *Up:* Mean hourly temperature (dashed line) on the study site and 48 hrs-moving window mean (solid line). *Down:* Calculated RTC deviation per second in ppm for all *DataLogger Micro* units. The green line gives the mean deviation and is bounded by a box of maximum and minimum deviations found for the given timestamp.

For the time step evaluation and correction the proceeding was slightly different, due to the numerous data losses within the *DataLogger Micro* group. For each time stamp the mean deviation of all custom stations actually holding data were used for the entire group. In order to give an expression of the variations of deviations a box is drawn

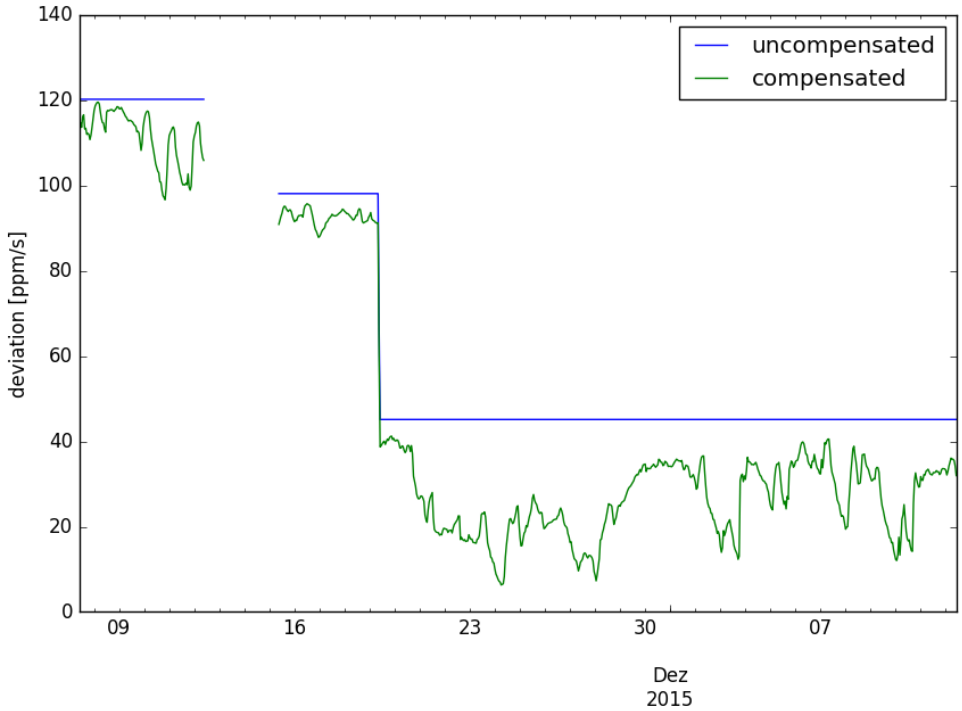


FIGURE 4.7: The time step deviation of compensated and uncompensated time steps of the custom sensor network in ppm per second.

around each time step bounded by the minimum and maximum deviation present. This is shown in figure 4.6, where the upper subfigure is exactly the same as in figure 4.5 and therefore won't need further description. Inspecting the lower subfigure reveals that the level of time step deviations is generally on a much lower level for the custom system than it was for the commercial one. Although the expected 65 ppm are exceeded at the very beginning of the measuring campaign, the mean deviation does not exceed this value in most parts. For the mean deviations there is again a general decreasing trend over time. The number of values used for calculating the mean deviation is varying due to sensor failures. Therefore the green boxes indicate the span of deviations observed. Although, when most sensors were working between November 27, 2015 and December 5, 2015, there were time stamp deviations higher than 65 ppm, none of these observations exceed 110 ppm.

Although the time step deviation could only be monitored very roughly and manually, the temperature data was available in higher resolution. Therefore the compensation was implemented and applied to the custom sensor network data. The result is shown in figure 4.7, where the blue line illustrates the observed per period mean deviations in ppm per second as shown in figure 4.6 (see p.58) and the red line shows the same deviations after they were compensated. The fact that the red line lies underneath the blue one, describing a decrease in deviation lies in the nature of the calculation. The power character of equation 2.6 (see p.19) will always lead to a positive result in equation 2.7 (see p.19). This is just an example for showing the simplicity of the compensation, which can be applied to any source of uncompensated time steps, as long as the used oscillator is known and documented by the manufacturer. Then only an temperature time series is needed for compensation. An statistical analysis or test of the or its significance cannot be executed as the reference system using the DS3231 failed. This has to be kept in mind at this point, there is no reference data for the compensation and in consequence this issue need definitely more investigation.

4.4 Data Products

4.4.1 Soil Moisture

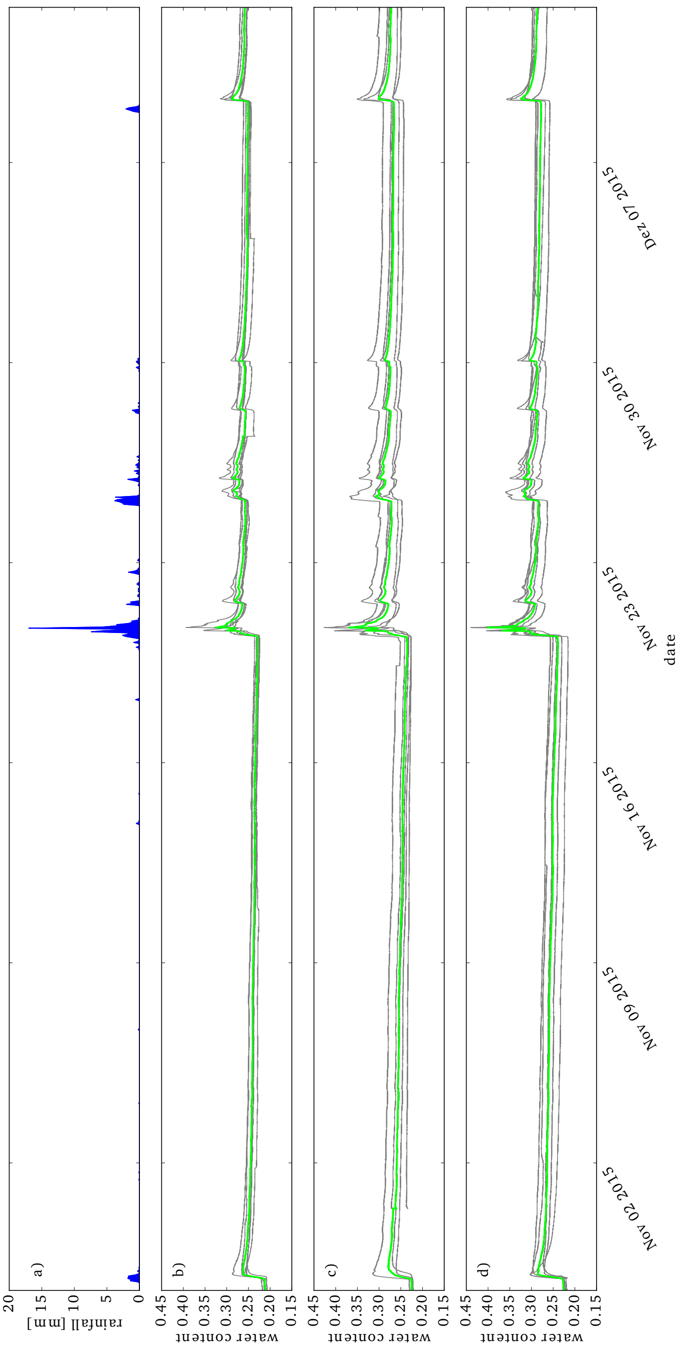


FIGURE 4.8: Result overview of all measurements taken by the commercial system. *a):* Hourly rainfall measurements taken by the WBI station in Freiburg (id=30). *b-d*: Decagon® EM50™ logger with 5TE sensors measuring at 2 minute intervals. *b):* Au1, id=504; *c):* Au2, id=503; *d):* Au3, id=502.

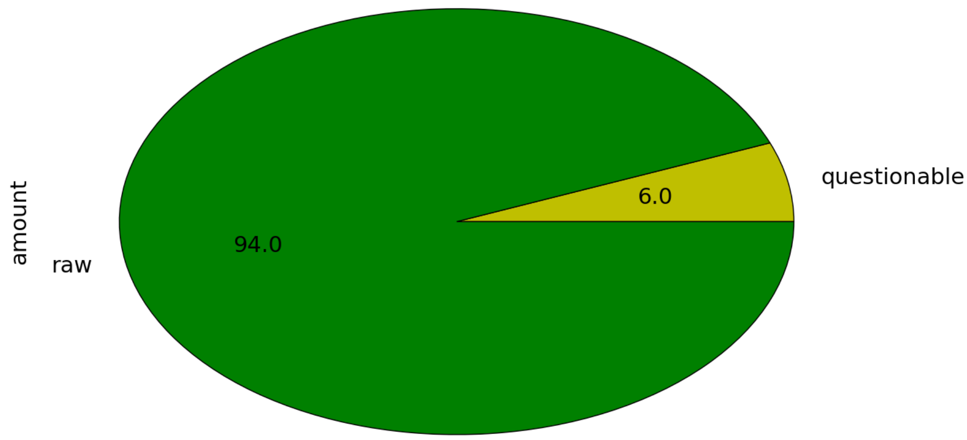


FIGURE 4.9: Overall sensor quality overview for all 5TE used in this work. The quality was categorized into incorrect (none observed), questionable (6 %) and raw values (94 %).

The first, most obvious, data product to be produced with each of the sensor network systems is a soil moisture time series record. An overview of all measurements taken by the commercial Decagon® driven system are shown in figure 4.8. For comparison reasons a rainfall record is shown in subplot a). The three EM50™ with five sensors connected to each, are shown in the subplots b) to d). Each gray line in figure 4.8 represents one sensor, while the light green line is their mean value. As the three loggers are spatially separated from each other at three different plots, the record splitting into three subplots was not only for clarity reasons.

Except for one rainfall event which took place only a few hours after the measuring campaign beginning, there was no rain for almost one month. As a consequence, all sensor signals showed descending values until a series of one big and many small rainfall events over a period of two weeks increased the mean signal level of all sensors for the second measuring campaign half. In general, all sensors show a direct rainfall response without noticeable delays. By visual time series inspection no periods of incorrect signals could be observed for any sensor, with one exception. In figure 4.8 d), station Au3, in connection to the last series of small rainfall events around November 30, one sensor shows step wise increasing values, while all four other sensors are measuring decreasing

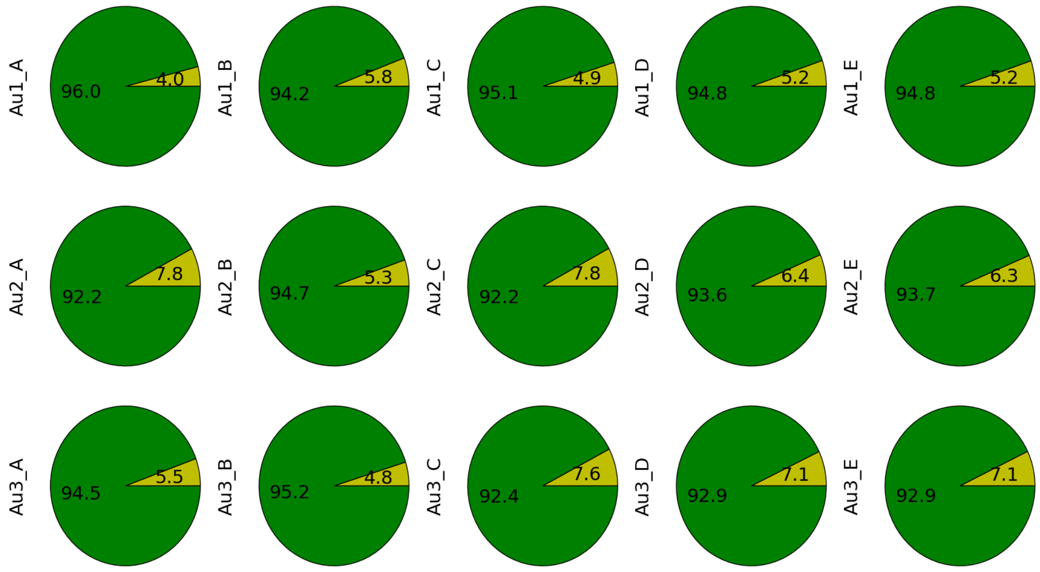


FIGURE 4.10: Sensor quality overview for each commercial sensors used in this work.

values. All measurements were calibrated by taking soil samples from the study site at the measuring campaign beginning. The calculated porosities are shown in table 4.2. As all nine measurements are very close together and differences are within calculation uncertities, a homogeneous study site wide porosity of 0.66 is assumed. The real gravimetric water content was calculated and the measurements were fitted to these values. Although all values are very close together at the beginning, as can be seen in figure 4.8, the water content on the study site is differing over time. These variations are increasing from plot Au1 over Au2 to Au3, which are laying on a vertical transect with Au3 being very close to the forest near the study site and Au1 being the furthest away. A similar trend can be observed in the maximum water content value after the big rainfall event at the mid of November. With all y-axes in figure 4.8 b)-d) having the same bounds, the peak heights can easily be compared. The closer a plot is to the forest, the higher the peak is.

TABLE 4.2: Porosity Φ of 9 soil samples taken on the study site. This was calculated by the difference in weight of the saturated and completely dried soil sample

	Au1			**Au2**			**Au3**		
Plot	A	B	C	A	B	D	B	C	E
Φ	0.66	0.67	0.64	0.66	0.66	0.65	0.64	0.66	0.64

Different automatic quality checks were integrated into the database as described in

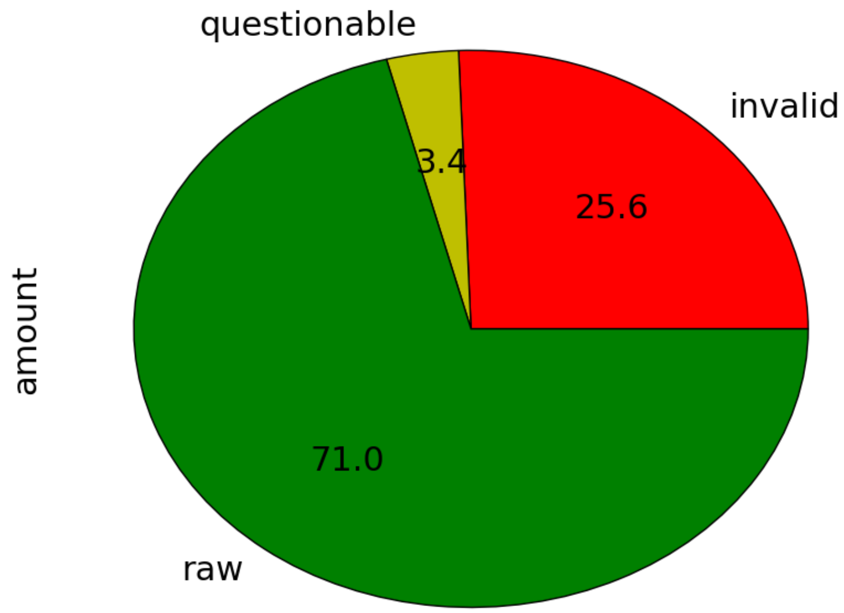


FIGURE 4.11: Overall sensor quality overview for all custom sensors used in this work. The quality was categorized into incorrect (25.6 %), questionable (3.4 %) and raw values (71 %).

section 2.2.1 (see p.14). These triggers marked incorrect and questionable data values, but were not corrected in figure 4.8. Generally, incorrect values will be dropped and interpolated, while questionable values will be checked manually and either kept or treated like incorrect values. Following this, the overall record can be marked as quality checked and interpolated, respectively. All commercial 5TE sensors were grouped by their quality flag and counted. This result for the complete commercial sensor network is shown in figure 4.9. Throughout the complete measuring campaign not one single measurement was marked as incorrect. 6 % were marked as questionable and 94 % of all values passed both quality checks. In figure 4.10, the same information is given for each of the 15 commercial sensors. As illustrated this ratio applies to all sensors, the amount of questionable values ranges between 4.0 % and 7.8 %.

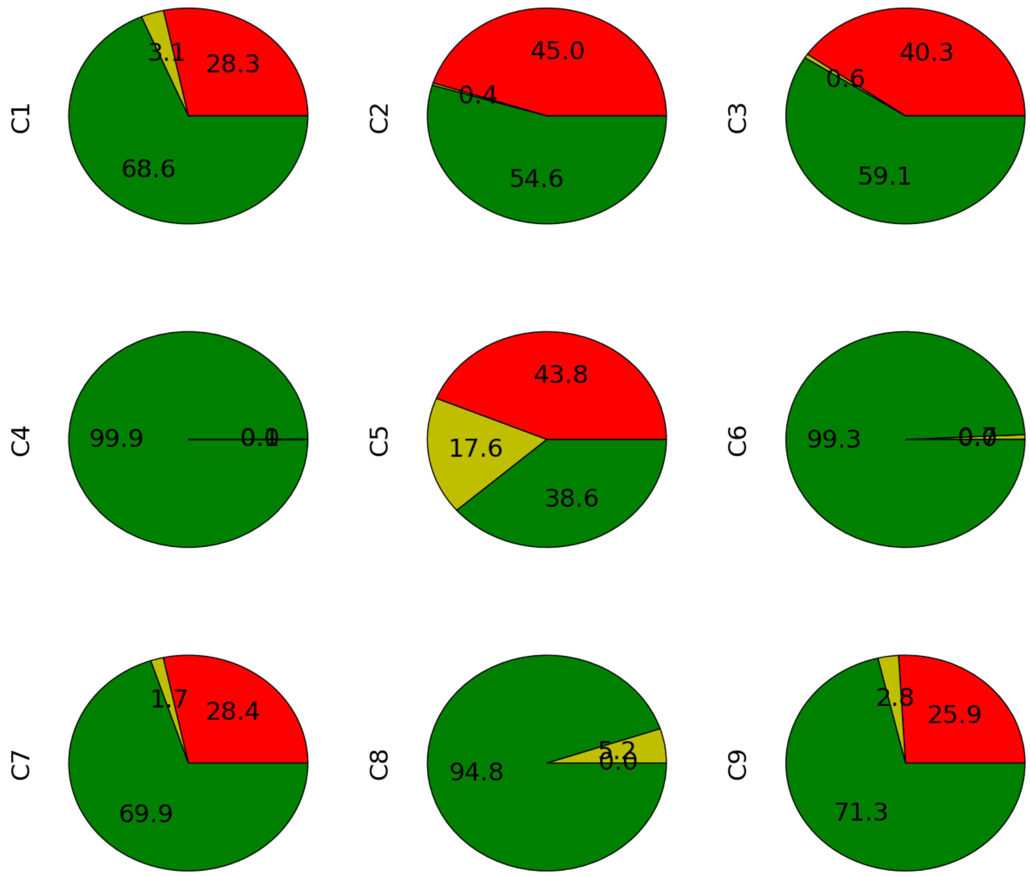


FIGURE 4.12: Sensor quality overview for each custom sensor used in this work.

Beside the described commercial system, a custom sensor network was used. As described throughout chapter 3 (from p.33), there were various problems in the power supply circuit. Solving these problems lead to a lack of time for a proper GUI development for those loggers. Consequently the devices had to be reprogrammed for only small parameter changes or data dumping. Therefore there was a limitation for the amount of devices and various data losses, which resulted from a wrong device handling.
Figure 4.13 shows all measurements taken by the custom system in comparison to the rainfall data within the same period, similar to figure 4.8 (see p.61). Each custom sensor data record ranging from C1 to C9 is shown in the sub-figures below. These plots do also show gaps in the record, which can be observed for almost every sensor. The custom sensor network does not produce nearly as homogeneous data records as the commercial

system by Decagon®. Above all the sensor C5 has to be mentioned, which does neither fit the other custom sensors nor the commercial system measurements, even with a noticeable amount of good will and imagination. It is not even close. Therefore it is quite obvious that this sensor measured under the influence of any superordinate error source, which did only affect C5. Additionally, the sensor C1 showed either very much noise or the same problem in the mid of November just before the biggest rainfall event set in. Apart from these obvious incorrect measurements, the general picture drawn by the custom sensor network is surprisingly homogeneous. The soil moisture dynamics of wetting ground after rainfall events and long periods of recessions afterwards were measured plausible by the other sensors. The peaks in soil moisture after rainfall events are of a similar magnitude and although the dryness level reached during recession differs more within the custom sensors than the commercial ones, it is still of a comparable level. In figure 4.13 this is illustrated by setting the y-axis bounds to the same extend for all sub figures except the first (rainfall) and 6$^{\text{th}}$ (C5).

When it comes to data quality, obvious differences can be accounted between the two systems. The sensor quality is tracked and saved for custom sensor network in exactly the same way as it was done for the commercial system. The occurrence ratios for the quality flags in the custom measurements grouped over all custom sensors in are shown in figure 4.11 (see p.64). Again, the same information was grouped for each senor individually and is shown in figure 4.12 (see p. 65) on a sensor basis. In both cases, the result is different from figure 4.9 and 4.10. According to figure 4.11, $^1/_4$ of all produced values are incorrect, 3.4 % are questionable and therefore only 70 % are raw data values to be checked visually. In comparison to the commercial system, where each sensor on its own showed similar ratios as the overall group, the custom system sensors behave completely different. Figure 4.12 shows the same graph for each custom sensor. The sensors C2 and C3 reach similar shares of incorrect values like C5, which was obviously complete wrong. The sensors C1, C7 and C9 produces similar quality ratios like the overall group with $^1/_4$ being incorrect and about 3 % questionable. On the other hand, C4, C6 and C8 did not produce any incorrect values and for C4 and C6, there was no occurrence of questionable data either.

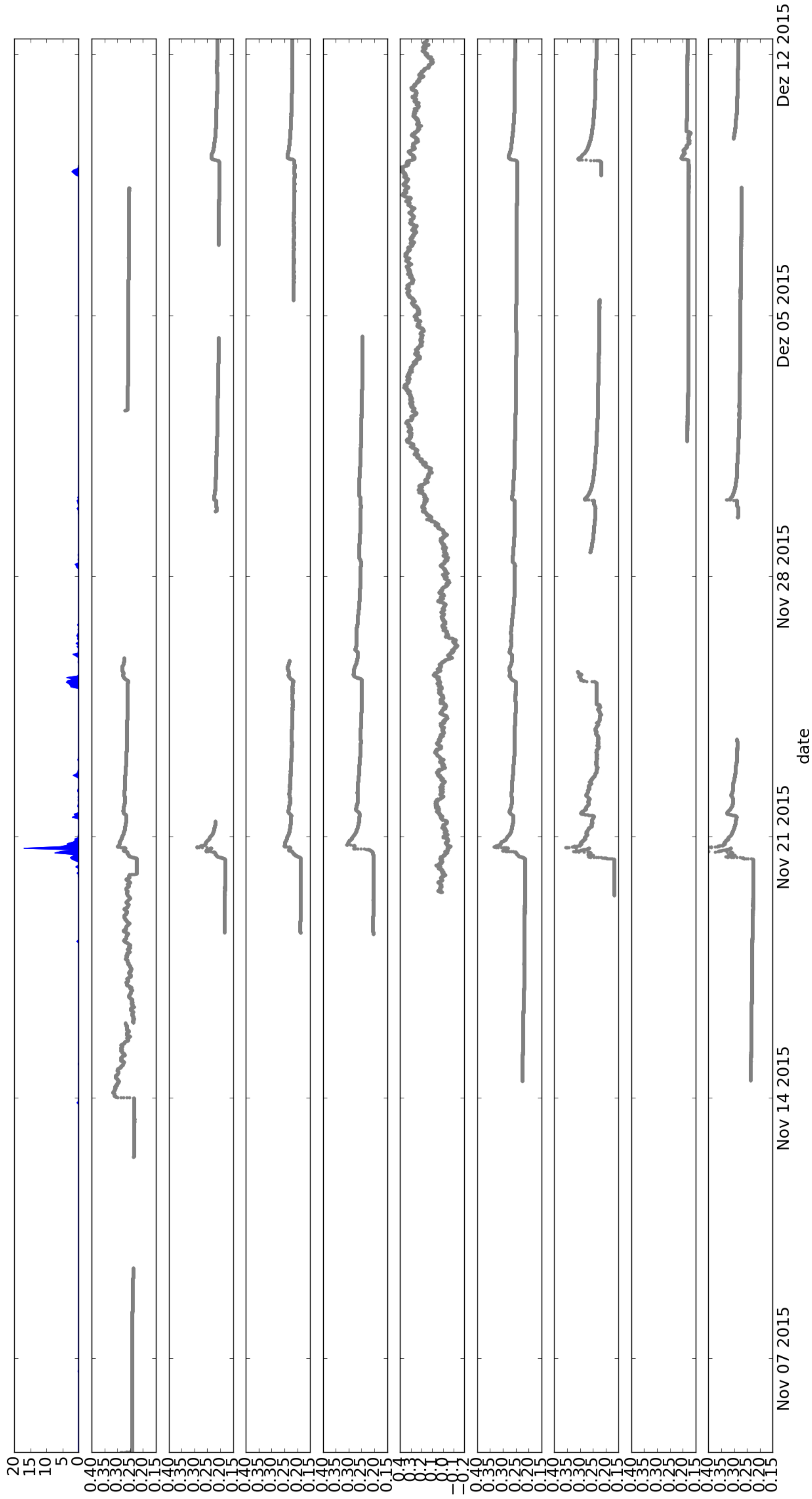


FIGURE 4.13: Result overview of all measurements taken by the custom system. *Top:* Hourly rainfall measurements taken by the WBI station in Freiburg (id=30). *Plot 2-10:* Measurements by the custom systems labeled as C1 (2^{nd} upper plot) to C9 (bottom plot).

4.4.2 Connectivity - Range

As described in section 2.3.3 (see p.24) the measured soil moisture by the sensor network was used to create a time series describing the connectivity on the study site for the campaign period. The connectivity is described by the range parameter of a fitted semivariogram. This fitting was performed for both the commercial and the custom sensor network on a hourly basis using six different approximation function. This produced a noticeable number of variogram graphs (23442 to be precise), which are attached on the supplementary disc (see appendix I) for obvious reasons[4].

Commercial System

The six different range time series were evaluated by the Akaike information criterion, the delta from the global minimum value and the ranking of each model based on this delta. All evaluation results are shown in table 4.3:

Table 4.3: Model selection result for the commercial sensor network. The overall AIC, Δ AIC and rank is described for all six approximation functions for the semivariogram models.

approximation function	**AIC**	**Δ AIC**	**Rank**
exponential	-276.064103	0.741563	4
gaussian	-276.805667	0.000000	1
modified exponential	-276.063169	0.742498	5
modified gaussian	-276.692574	0.113093	2
modified spherical	-274.738966	2.066701	6
spherical	-276.380581	0.425086	3

Based on the AIC, the Gaussian model is the most appropriate one, as it produces the smallest AIC value. Although the AIC value itself has no meaning, the similarity of the values seemed remarkable to the author and therefore more investigations on model assessment parameters was taken. This result is shown in table 4.4. Both, the RMSE as well as the sum of the mean, absolute residual value for each time step are almost the same. A Kruskal-Wallis-Test performed on the residual values (not the sums) could not reject the null hypothesis of all residual values be descended from the same population (H-statistic = 7.08, p-value = 0.214). The Gaussian model will be chosen anyway and

[4]you can find the files in the /results/range_*system*/ folders following the file name pattern of semivariograms_graphs_*model_name*.pdf

the results based on this model will be described. But these results might apply to the other models as well.

TABLE 4.4: Evaluation result for the commercial sensor network. The overall RMSE and sum of all mean, absolute residual are described for all six approximation functions for the semivariogram models.

approximation function	**residuals**	**RMSE**
exponential	37.323413	0.044417
gaussian	37.096175	0.044060
modified exponential	37.331514	0.044418
modified gaussian	37.006530	0.044159
modified spherical	37.057092	0.044088
spherical	37.213573	0.044346

The other five models might be fitting in a similar magnitude, therefore their results have to be highlighted as well. The discussion of this result is crucial for the overall value of all results based on or somehow related to the range parameter.

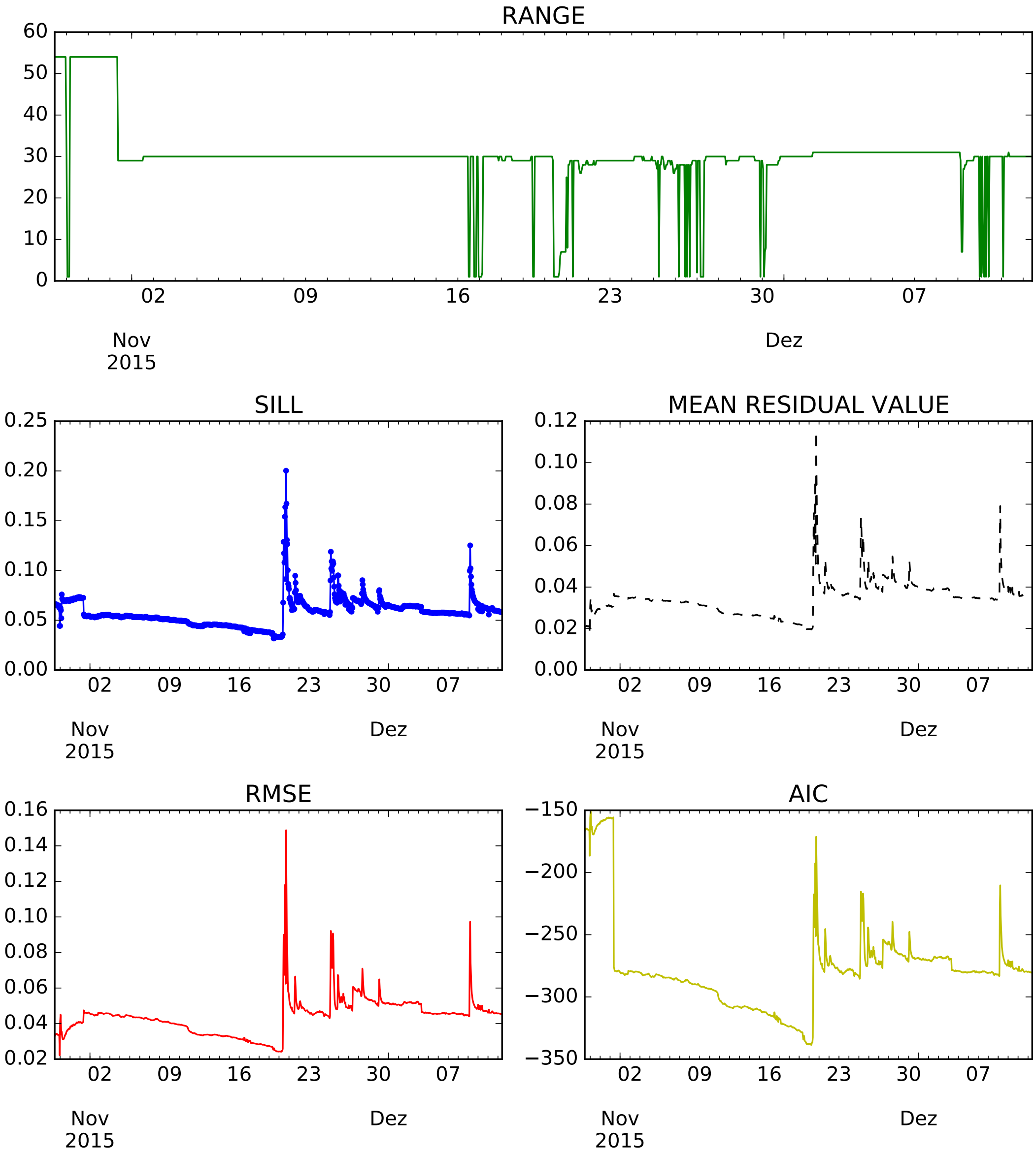


Figure 4.14: Range time series and model value plots for the commercial system based on the gaussian semivariogram model. The top subfigure (green) shows the range in meter over the measuring campaign, while the lower four subfigures show the semivariogram sill (blue), the mean, absolute residual values (black, dashed), the RMSE (red) and AIC (yellow) within the same period.

Figure 4.14 shows the result based on the Gaussian model described by equation 2.26 (see p.27). During the dry period lasting until mid of November the range stays at a constant 30 m, as expected. This means, that the variations in soil water content at a specific point on the study site do statistically correlate to other measurements within a 30 m range. In the second half of the measuring campaign, when rainfalls set in, the range is decreasing multiple times to a level near 0 for just a very short period. In between these spikes the level of a 30 m range is reached and established again.

The other four time series showing the semivariogram sill, the per hour absolute residual mean, RMSE and AIC, are of a very similar course. When reminding the soil moisture or rainfall measurements as shown in figure 4.8 (see p.61) or figure 4.13 (see p.67), the major rainfall (or soil moisture) peaks can be found in the model goodness graphs as their specific value peaking as well. In the case of sill, this is not very surprising as increasing values of soil moisture increase the sill of the semivariogram by its nature. For the RMSE and AIC there is a direct link to the residuals as increasing absolute residual values indicate an increased variability within the observed values. The RMSE is by definition a dimension of this variability. The differences in AIC and RMSE can be explained by the changing amount of sensors active at a given point of time. While the AIC is influenced by the number of readings, the RMSE is not.

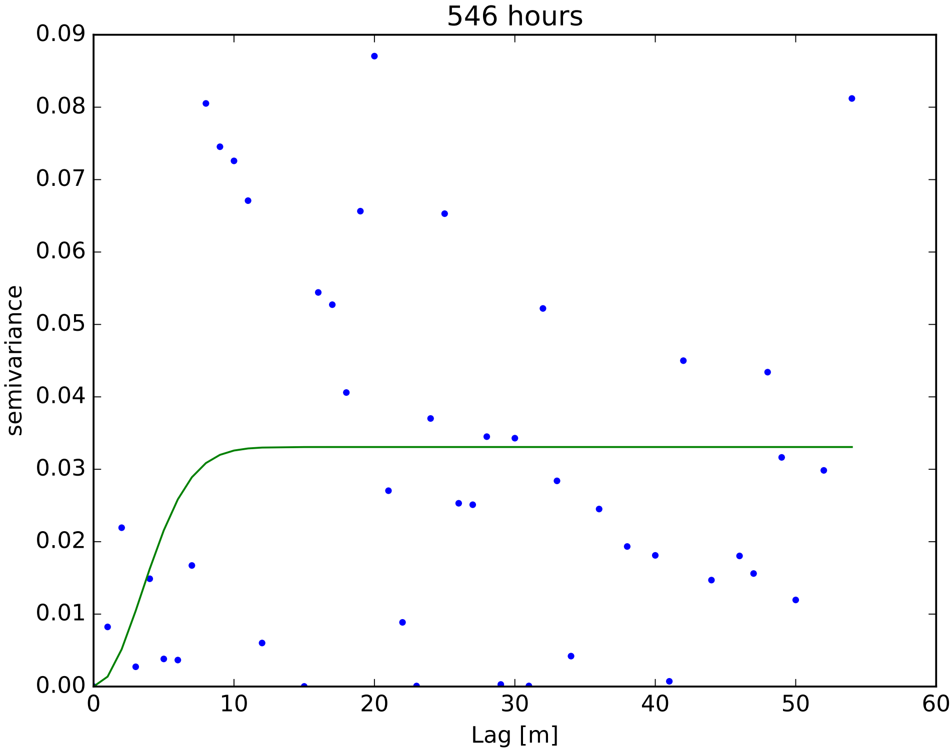


FIGURE 4.15: Semivariance (blue points) and Gaussian semivariogram model (green line) for the commercial system based on the water content measurements taken 546 hours after the campaign started. Measurements of one hour are included. This was approximately at November 19, 2015.

In order to shed some more light on the question of semivariogram model fit goodness,

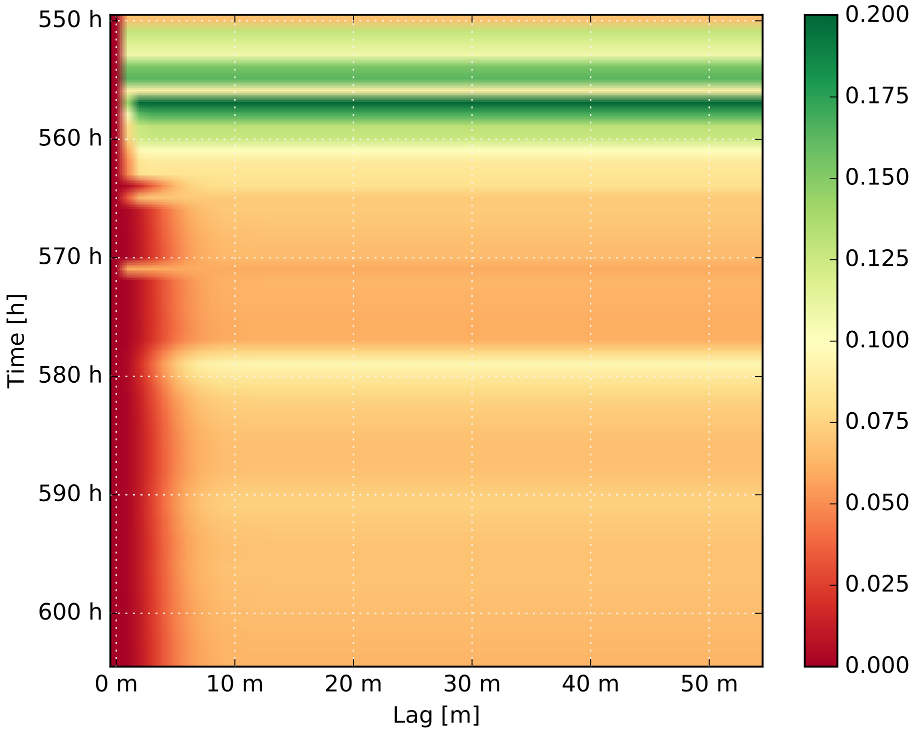


FIGURE 4.16: Semivariance Image based on the commercial system readings from 550 h to 650 h after measuring campaign start. The colors on the colorbar indicate the semivariance for each hour (y-axis) at each lag (x-axis). This image does only include semivariograms fitted by the Gaussian function as shown in equation 2.26 (see p. 27).

the produced AIC time series was ordered to find the best fitting models. The lowest value was observed 546 hours after the measuring campaign started. The corresponding semivariogram can be found in the *semivariograms_graphs_gaussian.pdf* on page 546 and is shown on figure 4.15. It must be made clear that the shown model is the *best* fitting one. Further, each of the semivariogram files was converted to a video file, which can be watched using any common video player. These videos illustrate the temporal changes in the observed semivariances and the models very well[5].

As a last result related to the catchment connectivity, the semivariance itself, as well as the semivariance given by the different models were combined into one image each. Showing the lag on the x-axis and the time passed since the measuring campaign started on the y-axis, each pixel in the image represents a color-coded semivariance value. Due

[5]The video files can be found in the /results/video/ folder and follow the file name pattern of semivariograms_*system_model*.mp4.

to the amount of data, these full images are only included in the supplementary DVD distributed wit this thesis. As mentioned before, the first half of the measuring campaign was dominated by dryness. This resulted in a mainly red picture representing very small semivariance values. The color does not change over the x-axis, as the soil moisture is on a comparable low level throughout the whole study site. Nevertheless, in November things changed. Several rainfall events evolved and the study site got wetter as described and shown before. Figure 4.16 shows an extract for the semivariance image based on the Gaussian model applied to the commercial system readings. The shown extract lasts from November 19, 2015 8:00 to 100 hours later (November 23, 2015 12:00). The big rainfall event setting in during the morning of November 19, 2015 lead to increasing soil moisture readings as shown in figure 4.8 (see p.61) and 4.13 (see p.67). This increased the sill of all semivariograms within the next few hours as indicated by the green colors. Within only 10 hours the sill diminished to the original level again and is increased slightly for only 3-4 hours with each following smaller rainfall event. The range can be read from the transition of small semivariance values to the constant sill in semivariance, indicated by the changing red color to a constant color. This transition is around 5 m on the x-axis for the whole extract, except for the rainfall events, where the range is decreasing to only 2 m. This matches the observations in the range sub-figure of figure 4.14.

Custom System

The same analysis presented above in section 4.4.2, was applied to the custom sensor network readings in exactly the same way. Table 4.5 presents the AIC values for each applied approximation function, the difference (Δ AIC) for each model from the minimum observed AIC value within this group and a ranking based on this difference.

The best fitting model in the commercial system, which was the Gaussian one, only ranks as fourth for the custom system. Here, it is the modified exponential model as defined by equation 2.24 (see p.27), which is highest ranking. As stated before, the similarity in the AIC is not meaningful in value, but remarkable, therefore the RMSE and mean, absolute residuals were taken into consideration as well, as shown in table 4.6 below.

Table 4.5: Model selection result for the custom sensor network. The overall AIC, Δ AIC and rank is described for all six approximation functions for the semivariogram models.

approximation function	**AIC**	**Δ AIC**	**Rank**
exponential	-24.907893	0.479729	5
gaussian	-24.949184	0.438438	4
modified exponential	-25.387622	0.000000	1
modified gaussian	-24.977468	0.410154	3
modified spherical	-24.711290	0.676332	6
spherical	-25.103812	0.283811	2

Table 4.6: Evaluation result for the custom sensor network. The overall RMSE and sum of all mean, absolute residual are described for all six approximation functions for the semivariogram models.

approximation function	**residuals**	**RMSE**
exponential	0.545827	0.616645
gaussian	0.518345	0.610462
modified exponential	0.547822	0.614985
modified gaussian	0.522365	0.610074
modified spherical	0.541848	0.617556
spherical	0.548199	0.616219

Again, no significant differences in the residuals or RMSE can be observed for the custom sensor network. Therefore it is again very important to discuss the following results and inspect the best fitting semivariogram in more detail. Nevertheless the best fitting, modified exponential model was chosen to calculate a range time series, indicating the connectivity on the study site.

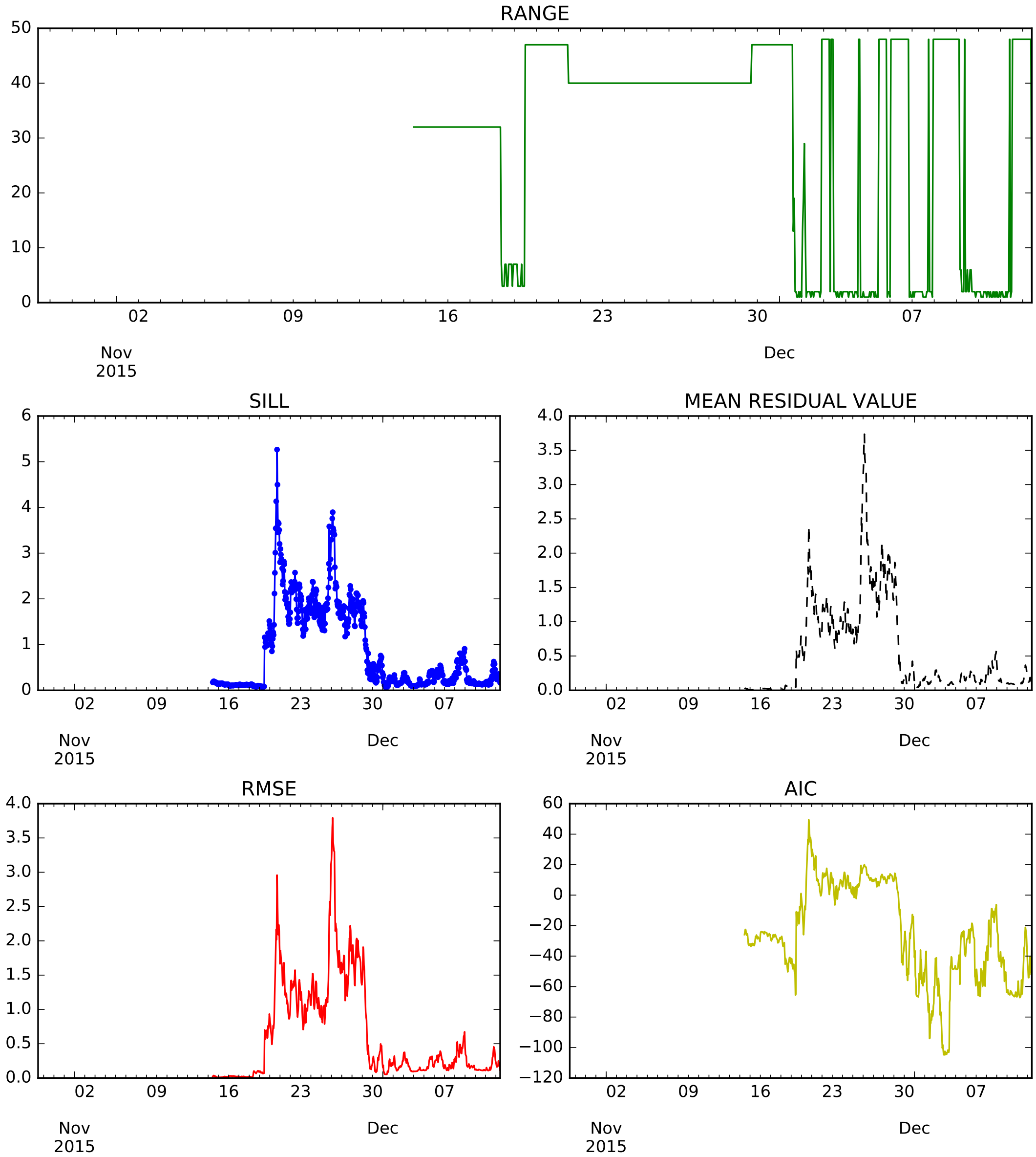


FIGURE 4.17: Range time series and model value plots for the custom system based on the modified exponential semivariogram model. The top subfigure (green) shows the range in meter over the measuring campaign, while the lower four subfigures show the semivariogram sill (blue), the mean, absolute residual values (black, dashed), the RMSE (red) and AIC (yellow) within the same period.

The range time series is shown in figure 4.17 in the top sub-figure (green). Similar to the Gaussian model and the related results as shown in figure 4.14 (see p.70) the range is decreasing during rainfall events and shows a static level during dryness. Unequal to the Gaussian model the static range level in the second half of November 2015 is higher with about 40 m. During the rainfall event on November 20, 2015 (57.7 mm), which was the most intense one, the range decreased to a level as small as only 2 m. During December,

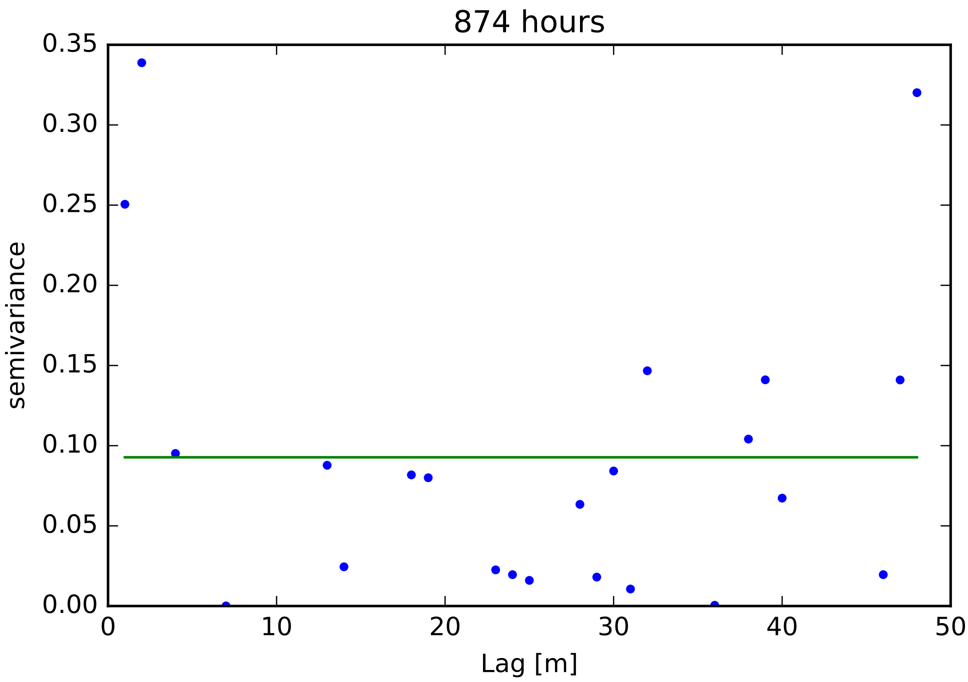


FIGURE 4.18: Semivariance (blue points) and modified exponential semivariogram model (green line) for the custom system based on the water content measurements taken 874 hours after the campaign started. Measurements of one hour are included. This was approximately at December 03, 2015.

the range was bouncing between only 1 m and 48 m, although only one rainfall event was observed on December 8, 2015 (8.8 mm). The other events on November 21, 22, 23 (5.1 mm, 4.8 mm, 0.4 mm) and November 25 (24.3 mm), did not cause any reaction in the range time series.

The mean, absolute daily residual values (black, dashed) and RMSE (red) do overlay and show the same graph as well as the same values. Opposite to the commercial system, where only a few sensor failures at the very beginning of the measuring campaign could have been observed, the custom system was struggling with numerous sensor breakdowns. As a consequence, the AIC gives a very different picture than the RMSE. This implies, that the changing amount of sensors during the measuring campaign leads to a different goodness of fit for the chosen model, which overlays the model quality as described by the RMSE over time.

The semivariograms for each observed hour were again sorted by their AIC value and the most negative one is shown in figure 4.18. As reported for the commercial system,

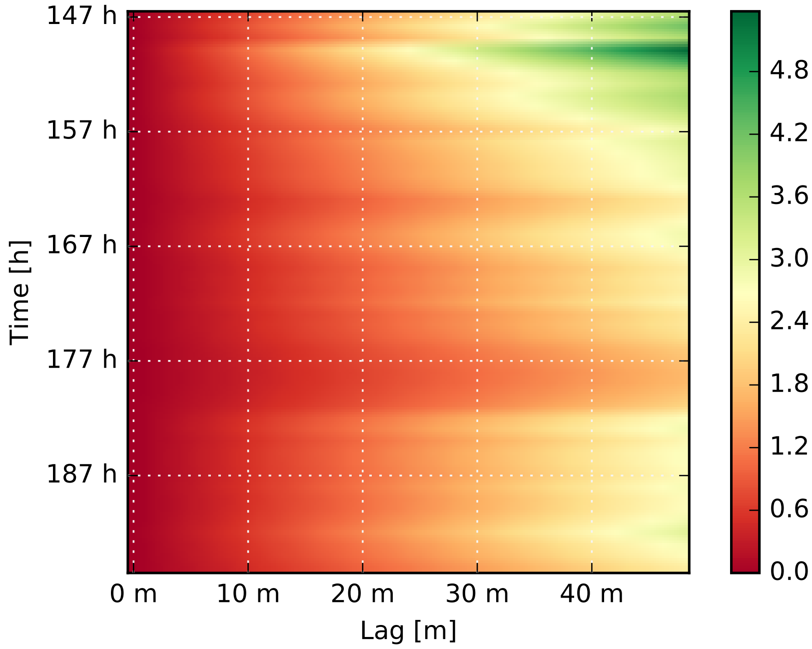


FIGURE 4.19: Semivariance Image based on the custom system readings from 147 h to 197 h after measuring campaign start. The colors on the colorbar indicate the semivariance for each hour (y-axis) at each lag (x-axis). This image does only include semivariograms fitted by the modified exponenital function as shown in equation 2.24 (see p. 27).

no spatial pattern could be observed. The blue points, each representing a semivariance at given lag, spread evenly. Again, the semivariogram was fitted in 10,000 iterating cycles using the least square method, but could not be parameterized better than shown in figure 4.18. The three parameters were chosen in a way, which lead to a strait line and not the expected exponential curve. Viewing the video of all modified exponential semivariograms fitted to the custom system reveals the fact, that this kind of parameterization was not uncommon and is also indicating the lack of a spatial pattern in soil moisture.

Although no spatial pattern could have been observed in neither the commercial nor the custom system, the semivariance image as shown in figure 4.16 (see p.72) was extracted within the same temporal limits for the custom system for reasons of completeness. This extract is shown in figure 4.19. The reason for the y-axis to range from 147 h to 197 h after campaign start differs from the y-axis shown in figure 4.16 (which was 550 h to 650 h, respectivly), is the different starting points for the two sensor networks. The custom sensors were started 403 h after the commercial one. As a consequence the, the durations for both datasets differ a the range for each figure as shown in 4.19 differ, as the same amount of sub-figures was produced for both systems. The reason for the values describing the color ramp ranging from 0 to as much as 4.8 is a factor of 100, which was multiplied to all semivariances before processing them to any kind of plot presented in this thesis. This was done to decrease the number of decimal places. This made labeling the figures easier, especially for the custom system. As mentioned before, the extract in figure 4.19 also covers the time while the most intense rainfall event set in and roughly the following two days. The range, which is the transition from red to yellow and green colors, is moving to lower lag values during the rainfall event and withdraws to a higher level within the next few hours. Overall the transition from low to high semivariances and therefore the visual identification of the range is not as sharp as it was drawn by the commercial system in figure 4.16 (see p.72).

Adapting Bin Size

TABLE 4.7: Model selection result for the commercial sensor network after bin reduction. The overall AIC, Δ AIC and rank is described for all six approximation functions for the semivariogram models.

approximation function	**AIC**	**Δ AIC**	**Rank**
exponential	-71.357037	3.730846	5
gaussian	-71.486216	3.601667	2
modified exponential	-71.357038	3.730845	4
modified gaussian	-71.486216	3.601667	3
modified spherical	-75.087883	0.000000	1
spherical	-71.356816	3.731067	6

In figure 4.15 (see p.71) for the commercial system and figure 4.18 (see p.76) for the custom system, respectively, each blue point represents the semivariance for each bin. This bin includes all points at the distance of the given lag for this bin. It was decided to make each bin overlay exactly one lag and therefore specify its size to 1 m. With the exception of one bin out of almost 60, none included more than 10 point pairs. As a consequence it was decided to increase the bin width in order to include more point pairs and calculate more robust semivariance values. The bin size was set to 5 m. Now, all bins but one include more than 20 point pairs.

Table 4.7 shows the results of semivariogram model selection based on the AIC for the commercial system after bin reduction. The AIC performs best for the modified spherical model. This was the common spherical model with a added parameter to vary the y-axis section. In terms of semivariograms this is the nugget. As before, the residuals and RMSE was inspected in order to estimate the difference in goodness of fit for the other models. These numbers are shown in table 4.8.

TABLE 4.8: Evaluation result for the commercial sensor network after bin reduction. The overall RMSE and sum of all mean, absolute residual are described for all six approximation functions for the semivariogram models.

approximation function	**residuals**	**RMSE**
exponential	0.016514	0.020377
gaussian	0.016295	0.020135
modified exponential	0.016514	0.020377
modified gaussian	0.016295	0.020135
modified spherical	0.012586	0.015663
spherical	0.016511	0.020377

Beside the fact that the residuals as well as the RMSE is smaller after the bin reduction, there seems to be a real difference in residuals between the best fitting model and the other ones. A Kruskal-Wallis-Test rejected the null-hypothesis of all residual value samples being descended from the same population (H-statistic: 563.62, p-value < 0.001). In consequence, the modified spherical model does fit better than the others. The goodness of fit is way better after the bin reduction. Now, the semivariogram modified spherical model does fit to the semivariances, as shown in figure 4.20. There is a relationship between the semivariance and distance of the points. Other than expected this relationship does not describe increasing semivariances with increasing distance, but the opposite. In the given example semivariogram the fitted model is decreasing for all lags < 30 m and of constant level for the other lags.

The range was calculated and is presented as usual in figure 4.21 (see p.81). The range is of 0 m throughout the whole measuring campaign, with the exception of November 20, 2015, where the 57.7 mm rainfall event took place. The range parameter is at the highest possible level of 50 m for exactly this day. Higher ranges than 50 m are not meaningful as the maximum lag calculated was the sorted into the 50 m bin. Prior to the bin reduction the maximum lag was 56 m and therefore higher ranges could be observed. The overall

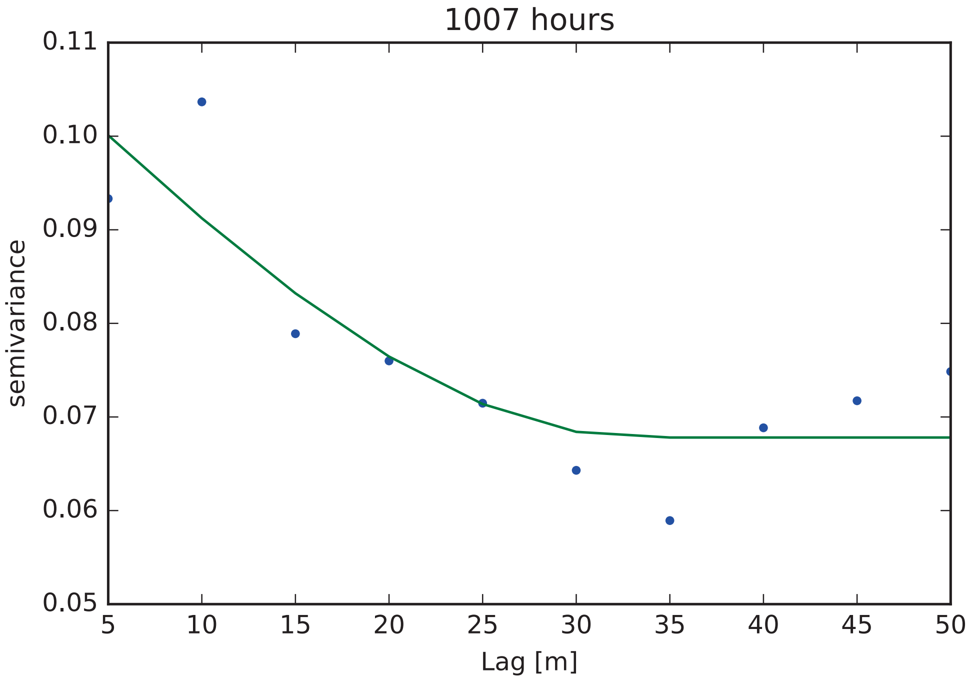


Figure 4.20: Semivariance (blue points) of reduced bin amount and modified spherical semivariogram model (green line) for the commercial system based on the water content measurements taken 1007 hours after the campaign started. Measurements of one hour are included. This was at December 09, 2015 09:00 to 10:00.

0 m range results from the range determination in the analysis script, where the x-axis value of the first occurrence of the maximum sill is defined to be the range. In case the first occurance of zero slope for the semivariogram model is defined to be the range, the range time series in figure 4.21 will be lifted to 30 m, wherever 0 m are observed. This has to be clarified, unless neither of both is more correct. This is a question of interpretation, as there are soil moisture similarities on the study site while the soil is overall very dry.

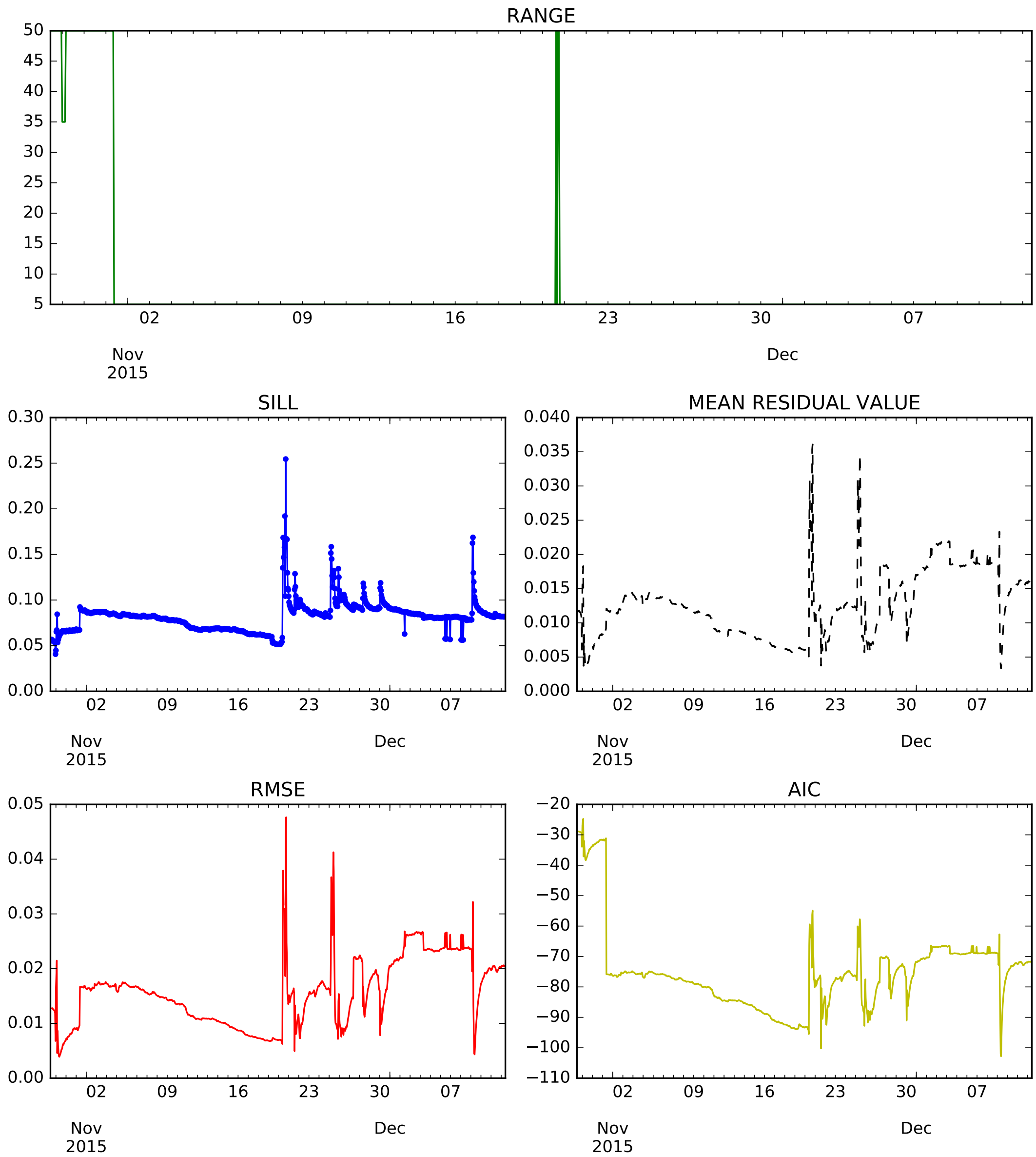


FIGURE 4.21: Range time series and model value plots for the commercial system based on the modified spherical semivariogram model. The top subfigure (green) shows the range in meter over the measuring campaign, while the lower four subfigures show the semivariogram sill (blue), the mean, absolute residual values (black, dashed), the RMSE (red) and AIC (yellow) within the same period.

The fact, that these types of variograms could not only hardly be found in the literature (see section 5.3.3, p.101 ff. for a few examples), but are really hard to interpret, lead to the insight to further investigate on the variograms. Due to the numerous sensor failures observed for the custom system neither the bin reduction nor any of the following could be applied to the data produced by this system. Therefore the main objective of comparing a commercial and custom system in a example analysis of hydrological relevance

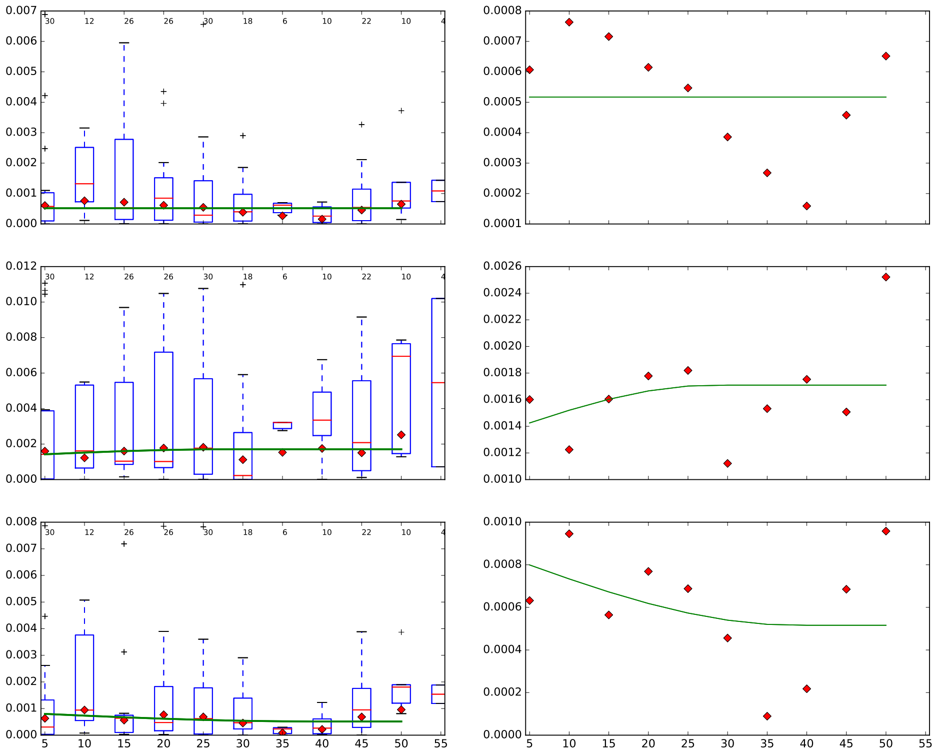


Figure 4.22: In-depth representation of three variograms fitted to the EM50^{TM} data. The second column shows the fitted semivariogram modified spherical models, as they can be found on the supplementary DVD. In the first column, the same semivariograms are shown, but for each bin a boxplot describing all semivariances of all point pairs present for this bin are shown. The first row is based on data 302 hours after campaign beginning, while the second row represents 558 hours and the third row 889 hours.

cannot be answered by the following results. In terms of hydrological connectivity it was done anyway.

Figure 4.22 shows the semivariograms for three different hours after campaign start. Each of them is based on one hour of measurement and was fitted by the modified spherical model after reducing the bins as described above. In the second column the semivariorams of hour 302 (1st row), 558 (2nd row) and 889 (3rd row) are shown as they can be found in the *semivariograms_graphs_mod_spherical.pdf* [6] file on the supplementary DVD. In the first column exactly the same variograms are given. Additionally to each semivariogram a boxplot based on all observed semivariances for the given bin is shown. The range between the 25 % and 75 % quartile is way higher than the mean semivariances. This makes the variograms look almost flat. It is observed, that quite a number of bins show outliers outside the whisker. Additionally, most of the shown bins

[6] This file is located in the /results/range_em50_bins folder.

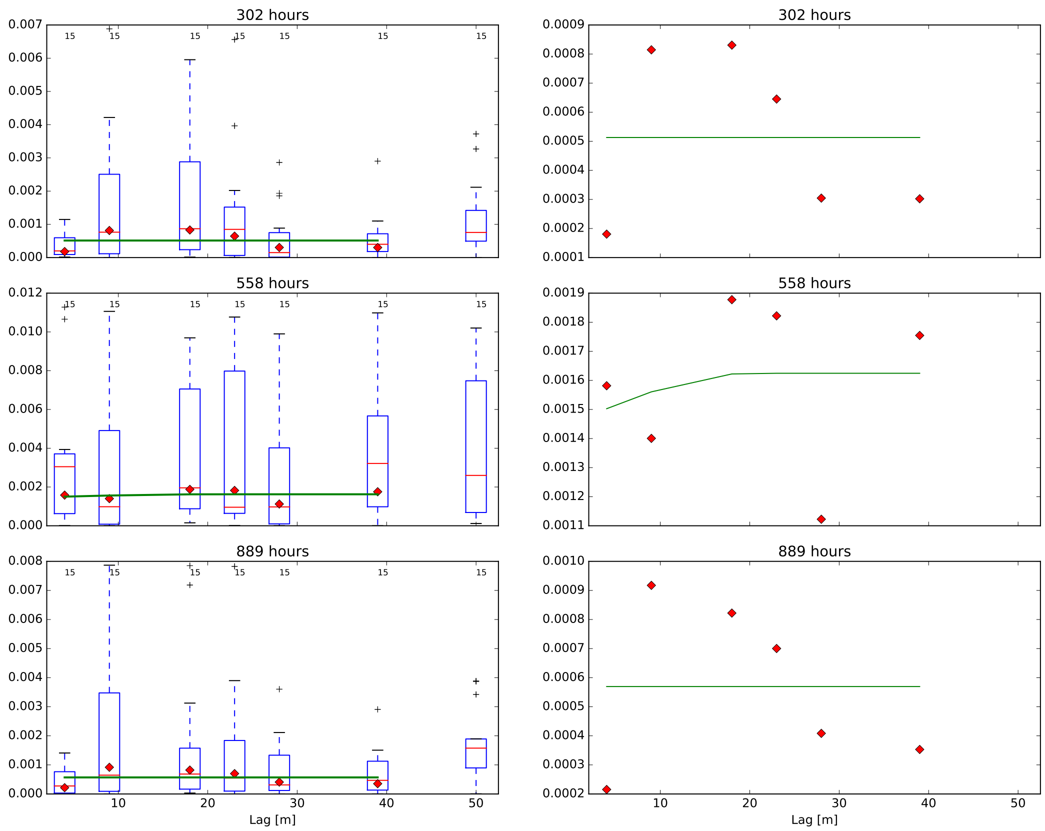


FIGURE 4.23: In-depth representation of three variograms fitted to the EM50^{TM} data. The second column shows the fitted semivariogram modified spherical models after reshaping the bins in order to equal their size. In the first column, the same semivariograms are shown, but for each bin a boxplot describing all semivariances of all point pairs present for this bin are shown. The first row is based on data 302 hours after campaign beginning, while the second row represents 558 hours and the third row 889 hours.

throughout all three rows show a shift between the mean semivariance (the red diamond markers) and the median (red line inside the box of each boxplot). Lastly, at the very top of each plot in the first column the sample size for each bin is given. This is the same for all three rows, as in all three cases the same amount of sensors was producing data. These numbers are very unevenly distributed and range from only 4 observations in the 55 m bin to 30 in the 5 m and 25 m bin. The sample size averages roughly to 17. Therefore the next attempt was to reshape the variograms to match a fixed sample size of 15 observations by varying the lag (The discussion leading to this step can be found in section 5.3.3, p.102 ff.). Fitted by the same model based on the same data, but using other bins, the figure 4.23 was produced. The semivariograms in the second column of figure 4.23 do look different from the ones in figure 4.22, jsut caused by the bin reshaping. The value range of the boxplots does not differ very much, but the red diamond markers giving the bin mean value of all semivariances do almost overlay the red lines inside the boxplots indicating the median value.

Using Median and Limiting Lag Range

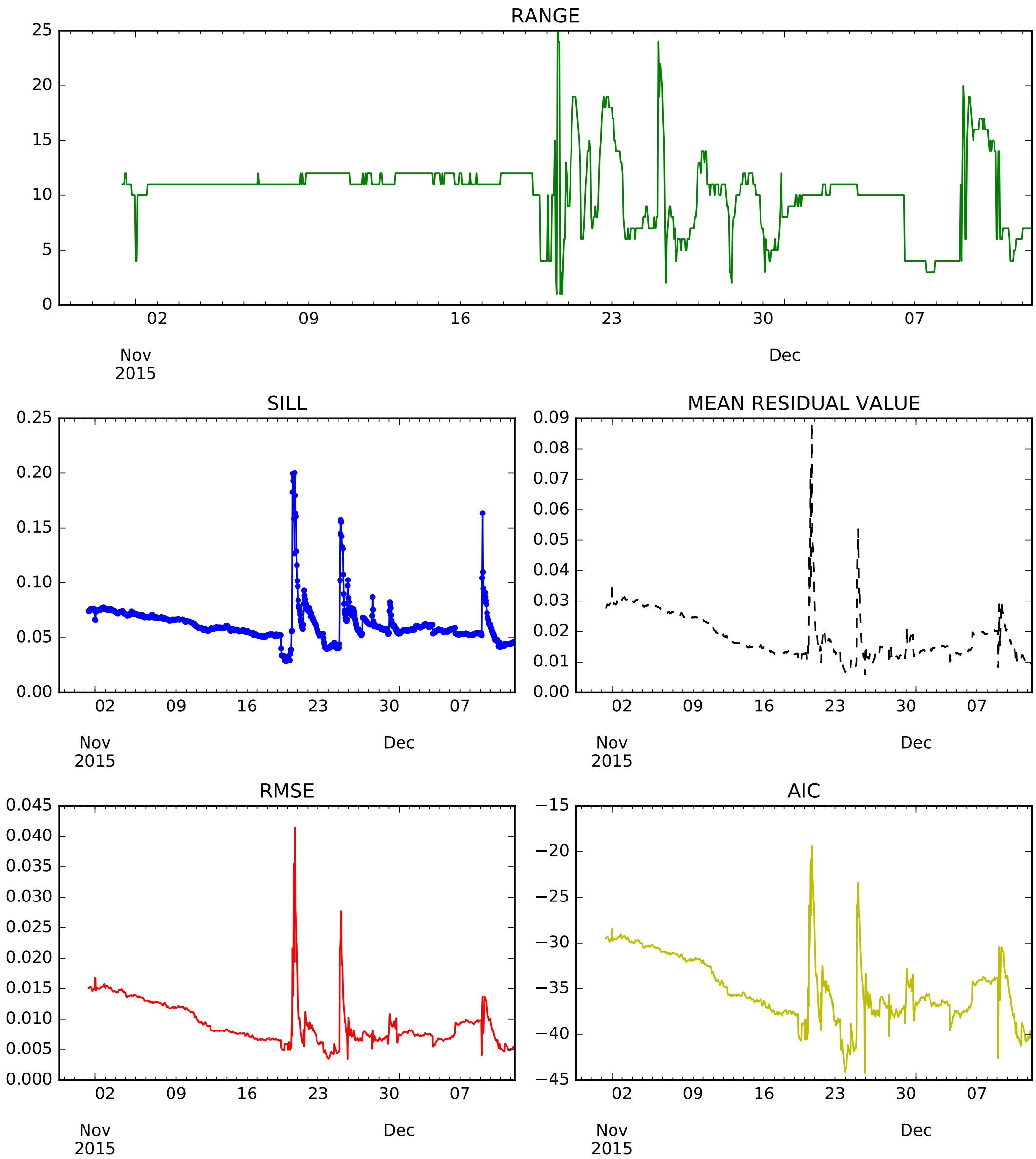


FIGURE 4.24: Range time series and model value plots for the commerical system based on the spherical semivariogram model after evening bin sizes and limiting the lags. The top subfigure (green) shows the range in meter over the measuring campaign, while the lower four subfigures show the semivariogram sill (blue), the mean, absolute residual values (black, dashed), the RMSE (red) and AIC (yellow) within the same period.

As this thesis was coming to an end, the whole analysis process was started a 6^{th} time. This time incorporating the median semivariance and the equalized bin sizes, as well as limiting the maximum variogram range to 60 % of the maximum measuring distance. This was one suggestion which came up while concluding and discussing the previous

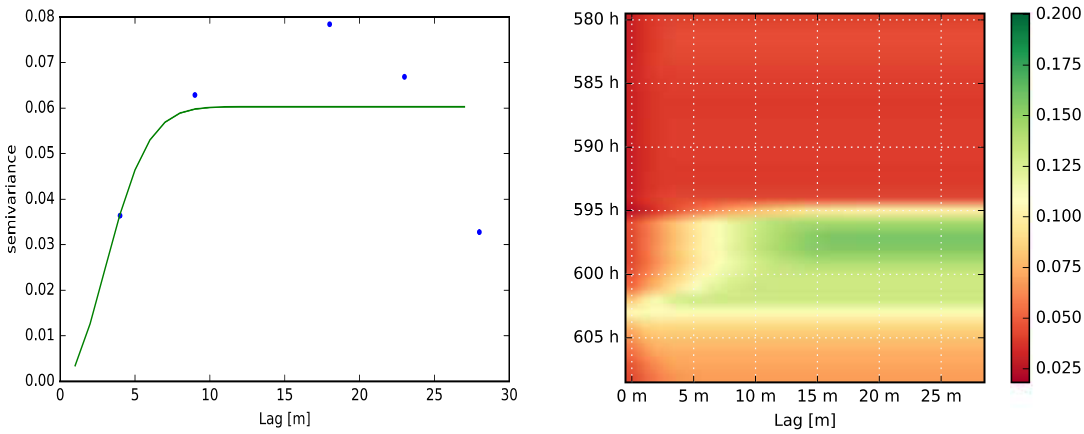


FIGURE 4.25: The shown graphs were made based on the data from the commercial system after limiting the maximum lag to 60 % of maximum measuing distance and evening the semivariogram bin sizes.
Left: Semivariance (blue points) spherical semivariogram model (green line) for the commercial system based on the water content measurements taken 598 hours after the campaign started. Measurements of one hour are included.
Right: Semivariance Image showing the extract from 580 h to 615 h after measuring campaign start. The colors on the colorbar indicate the semivariance for each hour (y-axis) at each lag (x-axis). This image does only include semivariograms fitted by the spherical function as shown in equation 2.22 (see p. 26).

results. As there is not enough time to describe all results as done a couple of times before, but these results are used in the next section, a very quick overview will be given here. Nevertheless, all results as presented before were produced and can be found on the supplementary DVD in the *results/range_em50_even_halfbins* folder. For the final data

TABLE 4.9: Model selection result for the commercial sensor network after bin reduction, median usage and lag limitation. The overall AIC, Δ AIC and rank is described for all six approximation functions for the semivariogram models.

approximation function	**AIC**	**Δ AIC**	**Rank**
exponential	-34.675563	0.337045	5
gaussian	-34.944301	0.068308	2
modified exponential	-34.703749	0.308859	4
modified gaussian	-34.922075	0.090533	3
modified spherical	-32.479281	2.533328	6
spherical	-35.012608	0.000000	1

treatment, including the evened bin sizes and lag limitation to 60 % of the maximum measuring distance, the spherical variogram model fits best as shown in table 4.9. The differences in RMSE and therefore in the goodness of fit are significant (H-value: 37.74;

p-value: < 0.001). Figure 4.25 shows on left side one of these semivariograms. It is obvious that the green line now fits the blue points way better. This was at 598 hours after campaign start. This point of time is also covered by the semivariance image on the right subfigure. Now, after the rainfall event at hour 595, the shift in range is visualized as the shifting transition zone. The colors again indicate the rise in sill. In the given example the range was increased by the rainfall event from 0 to about 13 m and decreased during the following 8 hours back to the original level. Additionally the transition in range directly after an rainfall event is much smoother, which might make more sense as the drying of a soil is not a binary and sharp process.

As a last result that will be presented in this thesis, the resulting parameters from each variogram were related to the soil moisture, that was recorded on the study site at that point of time. Namely, these parameters are the range, sill, AIC and RMSE as presented for example on figure 4.24 (see p.85), to name just one. These are shown in relation to the soil moisture in figure 4.26 based on the spherical variogram model like described by equation 2.22 (see p.26). In each of the four subplots the soil moisture for each recorded hour is shown on the x-axis, while the specific parameter is shown on the y-axis. In the top left subfigure the points are additionally surrounded by their convex hull. This shall illustrate the location of the point cloud and therefore visualize a pattern, if any. This becomes an interesting feature, when comparing the shown result to other models or other measurements from other study sites. Both comparisons lie beyond the scope of this thesis. Most range observations were made at about $0.27 m^3 * m^{-3}$ water content covering the full range of range values between 5 m and 20 m.

The other three subplots show a very similar shape. Each of them can be split into two parts, the drier soil moisture observations below $0.25 m^3 * m^{-3}$ and the ones above this threshold. In the drier part, each parameter shows an almost linear positive regression between the parameter and the water content. In the wetter part, there is also a positive linear relationship observable, but it is way more spreading. Between the two parts, there is a huge step, which almost aligns the smallest water content observations of each part by their parameter. This break is also present in the range scatter plot, but it is of a different shape. In the drier part, there is almost no variation in the range value. They are all grouping around 10 m. In the wetter part the variance is increasing dramatically, this increase has also a step-wise character at a water content of $0.25 m^3 * m^{-3}$.

In order to validate and compare these results, the BFI and runoff coefficient were used

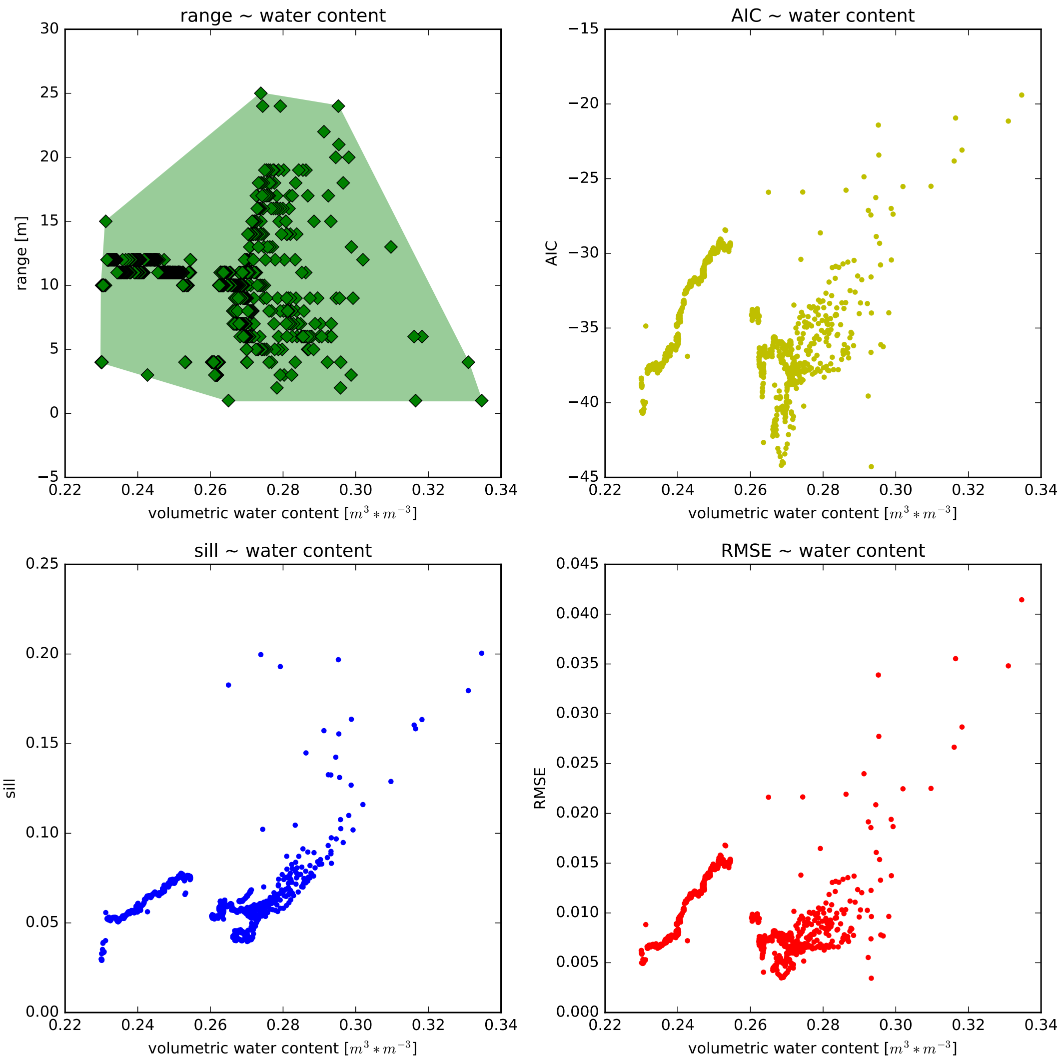


FIGURE 4.26: Scatter plots relating the different result parameters to the soil moisture. All shown parameters are calculated based on the spherical variogram model as shown in e equation 2.22 (see p.26). *Top left:* The range parameter for each measured hour is related to the mean soil moisture for the same hour. The points are surrounded by their convex hull. *Top right:* The AIC for each hour is related to the mean water content. *Bottom left:* The sill for each hour is related to the mean water content. *Bottom right:* The RMSE for each hour is related to the mean water content.

to measure and estimate the actual catchment response to rainfall, which is closely related to the connectivity. This is described in the next section.

4.4.3 Connectivity - Catchment Response

The previous sections described the process of calculating a range time series, which shall be used as a indicator for the catchment connectivity. In order to validate these

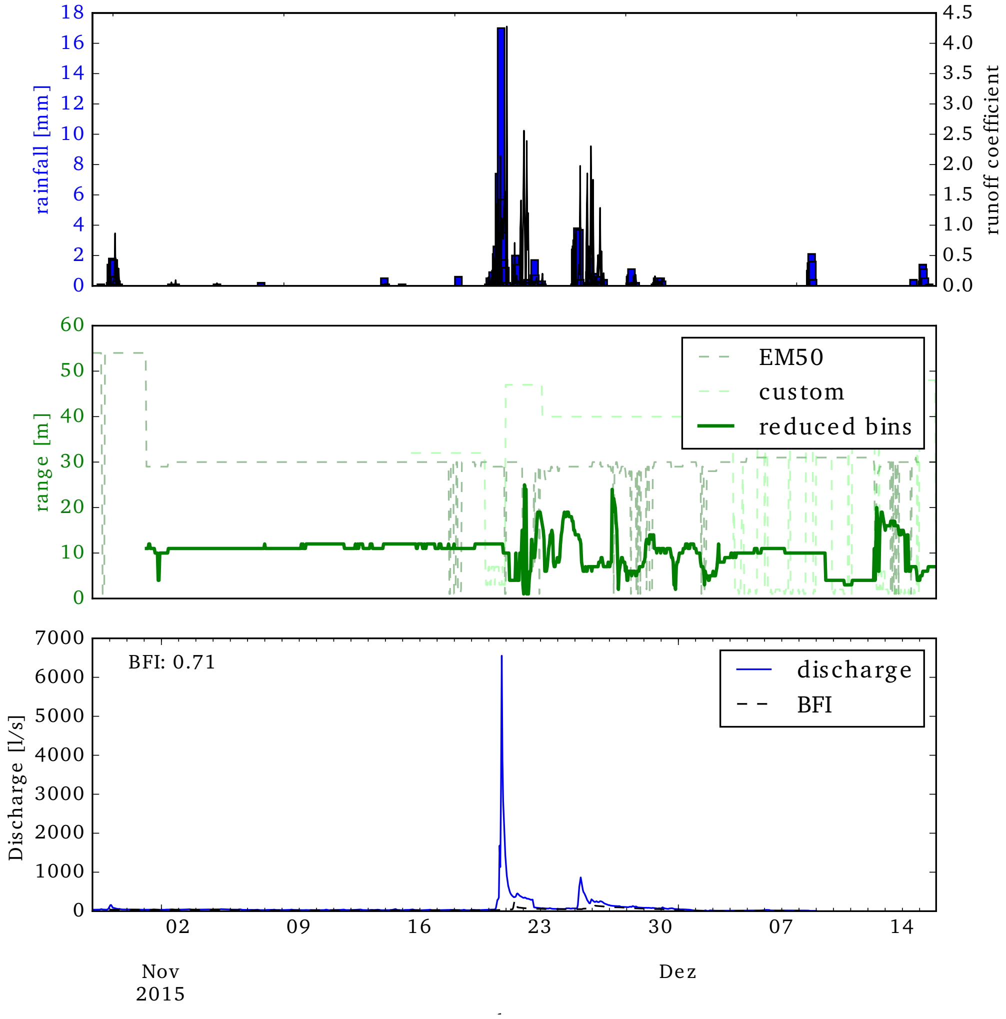


FIGURE 4.27: Overview of measured and calculated parameters describing the catchment response.
Top: Hourly rainfall data from the WBI station (blue bars) and calculated runoff coefficients (transparent-black line).
Middle: Range time series calculated by the spherical model based on the evened bins (in size) of the EM50^{TM} measurements, using the median semivariance for each bin and only including lags of 60 % maximum measuring distance (thick green line). The two original range time series, calculated without any adaptions are included as light green dashed lines.
Bottom: Discharge time series (blue) and BFI (dashed black) for the gauging station on the study site. All subfigures share their x-axis.

results, the actual catchment response to rainfall events was determined by measured data. Two parameters were chosen as indicators for the actual catchment response, the BFI and runoff coefficient. Both, as well as the range, discharge and rainfall, are shown in figure 4.27. The upper subfigure shows the rainfall and the runoff coefficient, both in hourly resolution. They fit very well as the runoff coefficient overlays the rainfall in

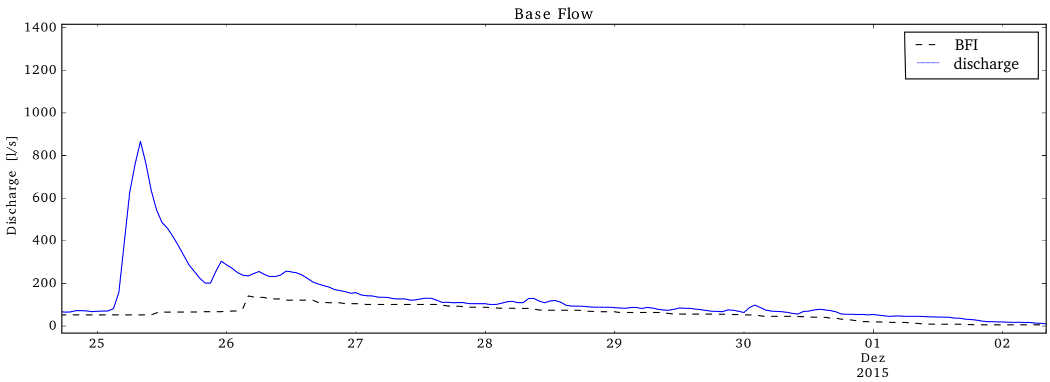


FIGURE 4.28: Extract of the discharge time series with calculated BFI values.

peaks and height in a way, that the rainfall plot can hardly be seen.

On November 20, 2015 the rainfall summed to 18 mm for one hour, which was the observed maximum. The runoff coefficient for the subsequent hour reached a value of almost 4.5, which was also the maximum. It has to be stated, that the runoff coefficient reaches these height values as the rainfall already decreased, when a rise in discharge was observed. This delay in the translation from rainfall to discharge within the catchment causes the high coefficients. The graph should therefore be interpreted by the area covered by the runoff coefficient, which is a integrated information, or one should use more aggregated data in case the coefficient value is of interest. This was not done here, as the dynamics in the catchment response are of more importance than the absolute runoff coefficient values.

The middle subfigure reveals a pattern of the range final time series and the runoff coefficient. This pattern seems to match visually for the two big rainfall events. The runoff coefficient as well as the range is under heavy changing conditions. If peaks do match and there is a real overlay will be described later in this section. The bottom subfigure of figure 4.27 shows the discharge time series, which was calculated from a stages time series based upon estimated parameters. Due to a lack of time, not all parameters of the used weir formula could be measured properly, therefore single measurements might be overestimated. This is especially true for the measurement of 7 m3/s during the heavy rainfall event of November 20, 2015. The dashed black line gives the base flow index, which averages to 0.71 for the whole measuring campaign. Due to the y-axis scaling this line can hardly be seen, therefore figure 4.28 gives an extract of this figure for the second big rainfall-runoff event. This dashed line separates the discharge into the catchment

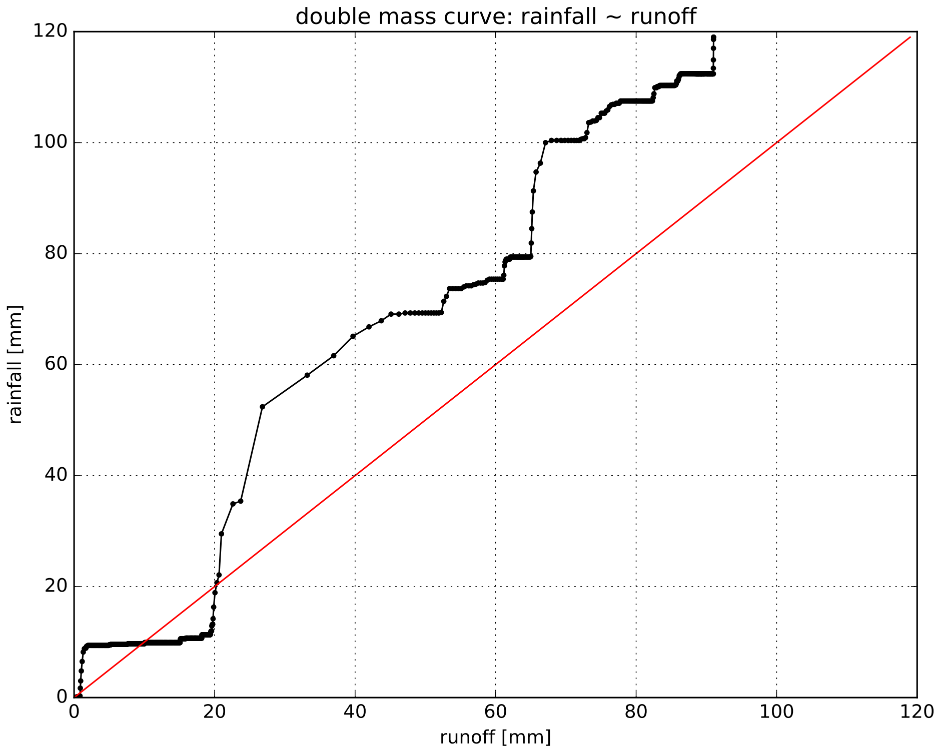


FIGURE 4.29: Double mass curve of cumulative rainfall and cumulative runoff for the whole measuring campaign (black line). All values are given in mm. The red line indicates a 1:1 relation, e.g. for 10 mm of rain 10 mm discharged.

baseflow ('below' the line), which would mainly be aquifer outflow or slow interflow, and eventflow ('above' the line) which is caused directly by a specific rainfall event and will discharge very quickly. Figure 4.28 shows a rise in discharge on November 25, 2015 for a few hours, but the baseflow was not affected by this rainfall event. A rise in baseflow can be observed more or less 24 hours later.

For evaluating the goodness of discharge calculations on the one hand and inspecting rainfall - runoff dynamics on the other hand, the double mass curve in figure 4.29 can be used. The red line indicates the 1:1 line, where the cumulative sum of rainfall matches the cumulative sum of runoff. This is possible as both measurands are given in mm. The double mass curve is represented by the black-dotted line, where each dot represents a single one hour time step from the discharge and runoff time series, that was present in both time series. Almost the half of all observed points are within the 20:20 square and represent the long period of dryness at the beginning of the campaign. The two big rainfall events are represented by two very steep jumps in the double mass curve. Here, as well as the double mass curve in general, the rainfall outweighs the runoff. Especially the first big rainfall event is only represented by a couple of points. Almost 50 % of all rainfall (57.7 mm) precipitated during this increase. But all in all the two measurands sum in a comparable amount, which indicates, that the weir formula parameter estimation was not too bad.

Figure 4.30 shows the cross-correlation between the range and runoff coefficient which

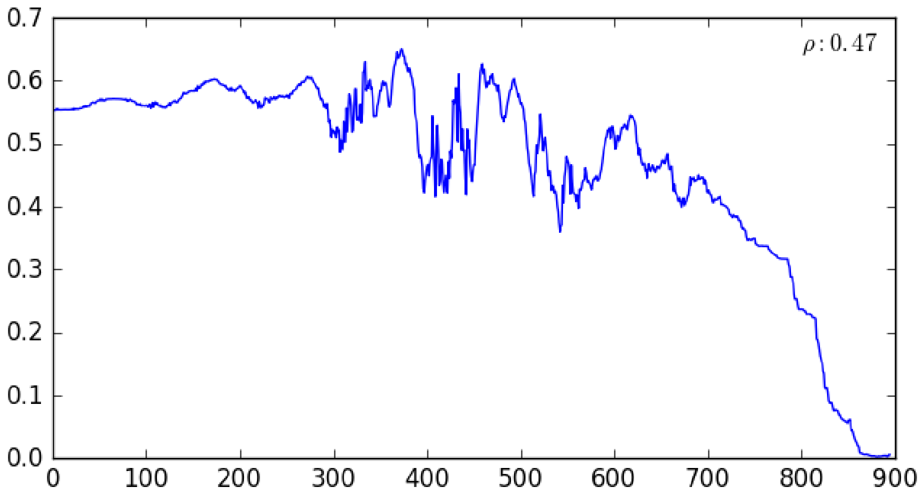


FIGURE 4.30: Cross correlation between the range and runoff coefficient for all observations over time.

averages to 0.47. While the y-axis shows the correlation value, the x-axis gives the index of the time step used for the calculation. As all time steps which did not have both, a range and runoff coefficient value were dropped for this calculation, the number given on the x-axis will not overlay with the hour after campaign start perfectly[7]. The cross-correlation differs over time. For the first approximately 300 hours it keeps a more or less constant level of 0.55, between 300 and 600 hours the correlation is highly variable ranging from 0.4 to 0.65 and for the last part it is constantly dropping and almost 0 for the last few hours of the campaign.

[7]In the raw measurement time series, there were 1078 time steps (in hours), while the treated time series used here obviously misses about 180 measurements.

Chapter 5

Discussion

5.1 Custom Sensor Network

5.1.1 Hardware

Prior to the presented work, there was only a schematic and layout draft for a custom data logging device, based upon the popular Arduino platform, but developed far beyond the limits of Arduino IDE[1]. Although the drafts were just mounted and used it has to be kept in mind that numerous errors were identified and workarounds as well as corrections were implemented and tested. As a consequence, it took the author about two and a half months to make the first unit fit for use. Ignoring the time pressure that was caused by the developing time extension of 1.5 months, the lack of time lead to an unsatisfying firmware. A software developer would call it a 'quick&dirty' solution.

The sensor was designed to compare the output to a 2.56 V reference voltage, but the ATmega328P™ internal ADC reference voltage was 1.1 V. In consequence, not the whole range of soil resistances could be measured. The sensor was designed to be more precise in wet soils, due to the season. As a result of the absence of rain throughout the first measuring campaign half, the soils were not wet enough. Therefore the sensors did not reach the level of precision they showed during the technical lab tests.

[1] The popular software for programming Arduinos; Arduino IDE homepage. URL: https://www.arduino.cc/en/Main/Software. Accessed: March 7, 2016.

5.1.2 Server

In contrast, it was exactly this custom framework in form of a Banana Pro^{TM} combined with sophisticated Python modules which made it possible to finish all planned analysis within the time frame of this work. More important is that all software packages performed as expected. Common software packages and infrastructures like the database environment were configured to use all memory and CPU resources at all times. With the exception of querying large amounts of data and very power consumptive calculations like the hourly variogram calculation, the Banana Pro^{TM} performed as fast as the dedicated server (from a human point of view.). The two Python packages *openhydro* and *hydras* speed up common data task like saving and querying or aggregation and interpolation. For all this tasks the author did not have to spend time and could focus on developing the analysis scripts, which could then, thanks to the server infrastructure, be run over and over again. Using other environments and no database application would have forced the user to adapt the scripts to every new input file and keep the machines running. The server was up anyway and copied the nightly results to the working machine every morning. The server could be concluded to make results available even before the coffee was finished.

5.1.3 Limitations

When it comes to limitations the firmware has clearly to be mentioned.as of this writing, this software is from a practical point of view only usable by the author. Only a person with detailed knowledge of the AVR-C language on the one hand and fundamental understanding of the board resources as well as basic knowledge of soil resistance relations to soil properties like soil moisture would be able to use the system at the current state. In addition, the case of sensor C5 showed that even if this multidisciplinary knowledge is available the system can still show unexpected and mysterious behavior. All this has to be kept in mind when relying on the data produced by the custom sensor network.
The workload of developing a custom sensor and a custom data logger at the same time and evaluating each unit and their combination was too high for a master thesis. Definitely more time would have been necessary for developing one device properly. The main power consumption and sensor precision issues were both caused by this combination. Therefore the main error was to use the supply voltage of the logger as the

reference voltage for the sensors. Lastly it has to be taken into consideration that measurement concept faced on measuring the in situ resistance of a soil sample between the electrodes applied into the ground. The resistance is depended on the soil moisture as it is on the salinity or the electrical conductivity, which is further depended on the pH-value in the soil. All these soil characteristics vary on a temporal and and spatial axis and will therefore produce a time and location depended error on the translation from soil resistance to soil moisture. As a consequence all these characteristics should be measured as well.

5.2 Technical Lab Tests

5.2.1 Battery Life

The battery life was designed to last about 4 - 6 months, which did not work out. Various attempts were made to solve issues and extend the battery life as first test revealed a life of just a few days. After these tests the battery life was extended and some devices used external battery packs. These interventions improved the battery life, but it is still off the designed 4 - 6 months. A reason for this could not be identified as point lab measurements of the actual consumption indicated, that the life span should be reached. There are two possible theses the author can draft. First, the temperature fluctuations lead to a decreased battery capacitance far worse than expected. Second, the sensors as applied in the ground still consumed more energy than in the lab for some mysterious reasons. Applying the sensors into a soil column and measuring the actual power consumption could shed some light onto this issue. Last but not least, a combination of both could have caused the deviations in battery life.

5.2.2 Battery Characteristic Curve

Section 4.2.2 (see p.50) summarized the results on creating battery characteristic curves for the custom system very briefly. This test was necessary to proof that the changing level of the battery voltage is a constant condition between two *DataLogger Micro* units. In case two of them behave fundamentally different, these curves can be used to recreate battery capacities and also discharge rates in retrospect. Figure 4.1 (see p.50) showed numerous characteristic curves at different conditions, all overlaying in their shape. The y-axis shift was expected, as the temperatures were different for each test run and influenced discharge rates of batteries. The mean battery characteristic curve can be used to estimate either battery life from battery voltage measurements at different points of time or recreate the battery voltage if only the lifetime is known. Further tests were not possible, as the routine for reading the battery voltage did not work properly and all measurements were corrupt. The reason was an error in the code, which did not provide an exception, but prevented the saving process from locating the correct memory address. In fact the very same address was used for any measurement. Manual measurements were done and the characteristic curve was used to roughly estimate the lifetime. This did workout very well, as only in one caste the battery run empty. In all other cases the batteries were changed in time (with some of them being dangerously empty).

5.2.3 Accuracy & Precision

When referring to the accuracy and precision test for the custom sensor devices, one has to take into consideration all the limitations described in 5.1. Especially considering the limits of translating soil resistance to soil moisture. Here, in this very special case the translation could be performed by a linear function, meaning all ADC values could be converted by multiplying them by a single factor. This indicates, that all influencing factors like the pH-value or conductivity are static conditions, not only during the lab test but also during the field campaign. It has to be considered, that during the lab test this might be true but during the field campaign these parameters could likely have changed over time. The fact that a linear function is still fitting may point to a false-positive fitting for this function. This means that the changes in two or more of these parameters over time equalize each other, this is also known as equifinality.

During the accuracy and precision test the soil column was dried. During this process some soil parameters might have changed. This could have caused the unexpected rise in soil moisture at the end of the test, which is in fact a rise in soil resistance. However, this is just an assumption as none of these parameters were measured.

5.3 Data Products

5.3.1 Commercial Network

The sensor accuracy was satisfying, with 94 % of all readings passing the applied quality checks. Not a single measurement was marked as incorrect and 6 % are questionable (see figure 4.9, p.62). As described, most of these questionable readings can be reduced to a sensor noise within the sensor precision and are therefore most likely not incorrect. This noise can be eliminated by aggregating the dataset or interpolating the questionable values. A not explainable increase in soil moisture higher than sensor precision without rainfall was recorded in only one case.
The overall quality remains only one issue about sensor quality in general. Beside the overall amount of incorrect or questionable readings, their distribution between the different sensors is very important. Only a small amount of sensors behaving noticeably different can be very hard to handle as long as the reason for this inconsistency cannot be identified. Figure 4.10 (see p.63) clearly showed a very even distribution of questionable readings being produced by all sensors in a similar amount.

Anyway, there is much more one could consider as a data quality parameter than just a application of a parameter range check and the relation of rainfall to soil moisture. If more quality parameters would have been included, the results might change. On the other hand, the application itself worked out well, as there are observations of readings that did not pass a quality check. Additionally, it has to be mentioned, that the measurements took place in a homogeneous system with same soil type and porosities (see table 4.2, p.63) at all sampling plots. A commercial system for almost 3000 $ (here) handling to sample correctly under this conditions is self-evident.
Building these capacities as a part of the database itself had several benefits. The processing times of this quality check did not really matter, as it was outsourced to the server, which is up all the time anyway. Secondly, using the database enabled the system to offer just different data views, which might or might not take the questionable or wrong readings into consideration. In consequence the analysis scripts did not have to be changed to include questionable measurements or vice versa. This decreased the processing times on the local machine and made more analyses possible in the same range of time. Using quality checks as implemented here, enabled the author to re-run

all analysis over six times. First using the data untreated and secondly treated. As only the database table to be queried for data has to be changed, this is very simple. There was just not enough time to describe results like this.

5.3.2 Custom Network

The custom sensor network performed worse, although it was measuring in the same, homogeneous environment. The overall data quality cannot be described as satisfying as only 71 % of all readings did pass both applied quality checks (see figure 4.11, p.64). The number of incorrect readings cannot be characterized other than horrifying, with every fourth measurement exceeding the bounds of physically possible values. Although one fourth of all readings being physically incorrect was denoted 'horrifying' in just the previous sentence, one has to consider, that the complete system was completely unchecked and never mounted prior to their use during this work. Before September 15, 2015, it was just an idea expressed as PCB layout and schematics. Building up, testing, evaluating, troubleshooting and producing a weatherproof small series of something within just 2.5 months is challenging. Therefore, the author recognizes the system as an success as the produced data was not complete nonsense. Nevertheless, this does not affect the scientific conclusion as described in the next chapter.
Figure 4.13 (see p.67) showed a complete failure of the unit C5. This unit was recording values, but obviously not the soil moisture (or soil resistance to be more precise). Most likely, the electrodes were not in sufficient contact to the soil matrix. A water accumulation could cause these readings. An error during the mounting process of C5 as a source of unpredictable behavior is also conceivable. This unit changes the overall performance of the system significantly as it has over 40 % wrong values and an almost 20 % questionable values. In fact, taking the soil moisture graph into consideration, one has to conclude, that C5 matched the quality check for 37 % of the readings by accident. Anyway, C5 was not the only bad sensor. Figure 4.12 (see p.65) unveils two other units (C2, C3) with $> 40\,\%$ incorrect and three units (C1, C7, C9) with at least 25 % of incorrect values. On the other hand three units (C4, C6, C8) performed very well. There was no obvious reason for different performances based on location or soil properties. The sensor orientation and land coverage were also almost the same for all nine units. Different software versions or the absence of the temperature sensor for four units didn't show a pattern correlating with sensor performance, neither. One could

give sympathetic considerations to unit C4, C6 and C8 to be a potentially satisfying logging system, and further investigate on sensor failure causes. On the other hand the absence of questionable or invalid measurements for two units out of nine could also be a coincidence. Developing an amended concept of measurement might be more satisfying and rewarding.

Based on the unsatisfying feeling not being able to understand his own sensor network, the author could not help to take further investigation on the mystery of the C5 readings. A possible explanation is an influence of the soil air humidity on the measurements. This was supposed to be highly unlikely as even saturated air should still isolate the two electrodes from each other even if their distance is below 1 cm. Another lab test proofed this assumption. Two electrodes were applied into a oven, which was saturated by evaporating an open water source. The electrodes had an distance of 0.5 mm and did not measure values above 0, which is the equivalent of a resistance between the electrodes exceeding the measurable bound. An accidental connection of the electrodes would result in constant measurement of the value 1023, which is the maximum possible value, representing a resistance value near 0 Ω. Another explanation is the accumulation of water between, or next to the electrodes. This water could have influenced the soil between the electrodes and overlayed the signal measured by the other sensors. A water cone in a small hole inside the soil matrix could have taken months of drying and constantly wet the sampled soil. Anyway, this does not describe the random drying recorded by the sensor. The last explanation considering a correctly working C5 could be the the sampling inside the rhizosphere of the grass. This sensor was only a few centimeters below the surface and if unhappily applied, the sensor could have measured the changes in resistance next to roots of the very same plant due to water uptake by this plant. Nevertheless, both theories cannot be proven and therefore the possibility of a C5 malfunction cannot be rejected. Unfortunately no approach of malfunction could be identified, that could explain these kind of misreadings.

5.3.3 Connectivity - Range

When looking at the first result presented in section 4.4.2 (see p.68ff.) the study site has to be taken into consideration. A common factor influencing all results is the extend. In fact the study site is very small with most sensor distances ranging around 10 m. The

maximum sensor distance is less than 60 m. It has to be considered, that this could be an distance at which spatial pattern in soil moisture are not yet observable and do only form at different scales. The same could apply to the measuring depth. As a consequence of all the errors and issues related to the system it was decided to apply the electrodes at about 10 cm depth. The main reason for not putting them deeper into the ground was, that it would also be harder to get them out again for troubleshooting. Anyway, the soil moisture patterns could just be more obvious or measurable at another depth. The limited amount of sensors forbid the application of sensors at different depths within the same locations. This is definitely a huge downside of the whole measuring approach. And lastly the measuring approach was clustered as the EM50^{TM} logger could connect five sensors with limited cable lengths. Therefore there were more or less three plots including 5 sampling points for the commercial system. The design for the custom system sticked to this approach in order to keep the results comparable.
It could not be proved, that the study site is somehow representative for the whole catchment. The study site was very steep and covered by grass, but the dominant landuse in the catchment was forest. The catchment response which was used to evaluate the range time series was the runoff coefficient and BFI for the whole catchment and not the study site only. In consequence, without further investigations of the whole catchment, the result are not really comparable. At least not in value.

In order to improve the results some adaptions were applied to the results which included as a first step the reduction of bins for the hourly variograms. The bin range was increased from 1 m to 5 m. On the supplementary DVD there is also a result folder for 10 m bins, which does not differ from the 5 m results. The reduction of bins lead to a increased sample size. Most statistical methods need a minimum sample size, which applies to the semivariance calculation as well. It was not easy to figure out a specific number from literature as this number is also depended on the usual variance present in the measurements. As described in section 4.4.2 (see p.78 ff.) the bin reduction did lead to a significant different in the RMSE values for the six different variogram models. At the same time, the AIC chose one model over the others and the best fitted model version for the values taken 1007 hours after campaign start (figure 4.20, p.80) looks better. The modeled values (green line) do describe the observed semi variances (blue points), Although there are still uncertainties in the variogram. Anyway, most noticeable concerning the variograms is not the improved goodness of fit, but the fact

they are *'upside down'*, somehow. They look like the covariance graph one would expect associated to the variogram dataset. In fact, it is hard to interpret a variogram like this. Following Jian et al. (1996) the range is the lag at which the sill becomes negligible. If one would take this definition very strictly, the range in figure 4.20 would occur at about 30 m as the semivariance values for all lags higher than 30 m are not influenced by the sill anymore and it is therefore negligible. The observed variogram, as well as almost all others for the reduced bin analysis[2], could not be attributed to any conceptual semivariogram present in the literature.
After reviewing more specific geostatistical related literature, 'non-classic' or 'periodical' variogram models could be found. Curran (1988) refers to variograms without a bounding sill as either *"'periodic' semivariogram [. . .] recorded across a repetitive pattern and the 'aspatial' semivariogram [. . .] recorded either along such a repetitive pattern, randomly on a homogeneous surface, or when using a support that is larger than the range"* (Curran, 1988, p.3, Describing the semivariogram). Unfortunately, Curran does not give further information on how to interpret these semivariograms properly, as they focused on causes for this effect related to the used remote sensing technology.
The result for the range time series after bin reduction is shown in figure 4.21 (see p.81). The range, and therefore also the connectivity, was very low during almost the whole measuring campaign. As described before, the level of 0 m is that low by definition and could easily be settled to another level. The important aspect about the range time series in figure 4.21 is the significant rise in connectivity at the very beginning and at November 20, 2015. In both cases high rainfall events were observed at the catchment becoming connected in succession seems to be a meaningful observation.

A hint on interpreting this 'aspatial' shape could be found in Pyrcz and Deutsch (2003), refering to the *hole effect*. This describes periodical semivariograms and Pyrcz and Deutsch try to relate the shape and type of periodical patterns in the semivariogram to statistical effects in the original dataset. These authors seem to belong to a group of scientists, who investigated these 'aspatial' and 'periodical' semivariograms from a mathematical point of view in order to produce better variogram functions. They published a number of articles in 2001, any of them dealing with what they called the *hole effect* in semivariograms (Ma and Jones, 2001, Jones and Ma, 2001, Gringarten and Deutsch,

[2]They can again all be found either in the specific PDF document or as a movie in the /results/range_em50_bins or /results/videos folder, respectively.

2001). The hole effect is a decreasing trend or a span of lags which are significantly below the level of semivariance, which would be expected in a common variogram. In other words, these are variograms somehow in between the periodical and aspatial ones and most likely, most variograms produced in this work are of the same kind. Each paper focuses only on the hole effect in lithology parameters from drill-hole observations, which might be the case, because all scientists were working for the Exxon Mobile oil company. A hint of how this could be transferred to the results presented here is given by Jones and Ma (2001) who concludes that a plateau at very short lags indicate a binary variable with *"high variability in size of the most abundant lithology, and low variability in the other."* (Jones and Ma, 2001, p. 13). This could be an indication of two driving processes related to, or expressed as the soil moisture, which are only relevant on a low $10^1 m$ scale. The cause for decreasing variability on that specific scale could be a overlay of the less variable process over the more abundant at small distances. All in all the author could neither interpret these variograms, nor was he able to verify or proof, that the found Exxon literature is relevant as it was not entirely understood due to its very mathematical character. This has to be further investigated as the author can only make suggestions at this point.

Another approach to reshape the variograms was to inspect the variations within the bins as shown by the boxplots in figure 4.22 (see p82). The number of outliers as well as the in some parts enormous shift between the mean and median semivariance value within one bin indicate a high influence of extreme values. One approach would therefore be the adaption of the semivariogram function in order to use the median of all semivariances within a bin. Another possible approach of limiting the influence of extreme values would be to eliminate all obersvations being lower than the 25 % or higher than the 75 % quartile limit. Unfortunately in this case the sample sizes are not big enough to eliminate observations, especially not for the custom system. Secondly, also as shown in figure 4.22, the variations in semivariance within one bin (represented by one boxplot box limit) are by a magnitude higher than the variations between the bins. Therefore just eliminating extreme values would not make a difference. In consequence the bin sizes were equalized as shown in figure 4.23 (see p.83.

The main characteristic of all detailed inspected variograms of showing a way higher variability within one bin than between two bins can again be attributed to the study site size. The soil moisture is a highly variable parameter on a spatial as well as temporal scale. Within the measured two months it was possible to measure 'most' of this variability in time. The study site was too homogeneous to cover the same extend of variability spatially. In consequence changing the descriptive variable from the bin mean value to its median lead to way better variograms and is somehow legit as both are statistical terms describing the bin. Nevertheless, Curran (1988) did not report about the usage of the median over the mean. It was changed as a better fitting of a median-based variogram was observed in the bin plot (figure 4.23, p.83), therefore the act of changing cannot be justified by the better fit. Lastly, it is quite common to limit the maximum lag used for the variograms to the half maximum measuring distance. Here, 60 % were used, as there is one bin lying slightly outside the 50 % limit and the author wanted to include this bin as well, in order to have one more point in the variogram for fitting reasons.

All in all figure 4.25 (see p.86) shows a semivariogram with a satisfying goodness of fit and based on that a semivariance image, which reveals the range and its dynamics as a result as expected. One can easily read the range at each point of time as well as its dynamics. The colors do not only illustrate the transition zone, but also give the sill at any time.

The question of *'preffered states'* as described by Grayson et al. (1997) or McNamara et al. (2005) is hard to answer with just two rainfall-runoff events recorded. Nevertheless, figure 4.26 (see p.88) illustrated an interesting relationship between the different result parameters (range, sill, RMSE, AIC) and the volumetric water content. The study site behaves different above and under $0.25 m^3 * m^{-3}$ water content. Below this threshold there is a unique range and the sill increases linearly with the soil moisture, which kind of makes sense. In addition, the variogram fitting becomes more error prone with increasing soil moisture. This could be explained by a increasing variance in soil moisture, which was neither calculated nor presented in this work but is ultimately illustrated by the increasing RMSE as it is another expression of the fitting residuals.

The remarkable point is the break in the graphs. The increasing variances in the parameter values in all four subplots of figure 4.26 can again be explained by the overall higher soil moisture values and the related higher variances. But the break itself and the huge shift in parameter value could be an expression of a transition of states in the catchment. In case the threshold is exceeded, the variograms fit better, the ranges diversify and the sill decreases. On the one hand, this could be a hint of changing driving processes during runoff generation. On the other hand it has to be noted that all these observations are only supported by two rainfall-runoff events and can be overprinted by the long dry periods, especially at the campaign beginning. In addition, the observations in both groups were not counted. Keeping the soil moisture measurements of figure 4.8 (see p.61) in mind, a significant imbalance can be expected with the drier part of the subfigures including dramatically more observations. In fact this is obvious from figure 4.26.

Finally, it has to be reminded, that the method of creating variograms was changed until the results were satisfying. Although all steps were justified from a scientific point of view, the act of changing the methods until the results are nice produces a bias in the results. As long as all limitations and downsides described above are kept in mind, a satisfying tool was presented to estimate and illustrate the catchment connectivity based on soil moisture pattern. The next section will give an overview on measured catchment response and discuss the question whether the estimated and measured response do correlate or not.

5.3.4 Connectivity - Catchment response

While the catchment was dry, the range and runoff coefficient correlated above average. With the beginning rain, the cross correlation became less steady but still on an above average level. Only towards the end of the campaign the cross correlation went close to zero for not traceable reasons. When taken under further investigation, the range could be or deliver an additional parameter for flood modeling. As the custom sensor network will work properly one day, this system can be an cost efficient solution especially for developing countries.
The double mass curve in figure 4.29 (see p.91) revealed, that there is no dramatic difference in the sums of rainfall and discharge during the measuring campaign. This is an indication for the measurements were flawless. Overall, the curve is located above the red 1:1 line. This means that more rain than discharge was observed. The difference, if not lead back to inaccuracies, is a rise in storage. A refill of aquifers during the winter is what one would expect for a black forest catchment. In consequence, in this very specific case, the range is a suitable parameter for estimating the connectivity in the catchment. Different aspects about this correlation have to be discussed, anyway. First, there might

be a correlation between these two variables, but the observation to be above average during the period of rainfall events is highly influenced by the not explained decreasing correlation values at the end of the measuring campaign. If it was not for this decrease, the observed correlation might just be average. Apart from this, the level of correlation values is not too high. Not taking the decrease at the end into consideration, the mean correlation level might be as high as 0.55. Concerning all the inaccuracies during the calculations, especially the not perfectly fitted variograms, a resulting cross correlation of 0.55 might seem sufficient. One has also to consider, that a very neat definition of hydrological connectivity was chosen in this work. Whenever estimating the connectivity within a catchment, or deriving conclusions from the observations presented in this work, one has to consider this definition. The observations and conclusions presented here are only true for this specific time frame, the climate zone, the weather conditions during the campaign, the catchment size, the soil type and especially the preconditions within the catchment (antecedent soil moisture). In order to take more general conclusions, at least the mentioned influences have to be varied, as most of them were constant.

In section 1.4 (see p.5 ff.) it was explained that a lot of other studies were published investigating hydrological connectivity. Most of them were performed in very different climate zones leading to very different results and most of the authors concluded the climate zone to be a very important factor influencing whether catchment develops preferred states or not. The question for preferred states could not really be answered within this work as there were only two noticeable rainfall events and both caused a discharge reaction of similar extent with similar ranges present on the study site. Similar conclusions could be found for the catchment size, which ranged from very small catchments (McNamara et al., 2005) to medium sized catchments (Grayson et al., 1997). The actual weather conditions during the campaign were uncommon. There was under average rainfall, not only during the campaign, but especially during the preceding summer and autumn. This lead to very dry soils in the catchment, which is very unusual for the Black Forest in winter. Last but not least it is questionable to measure only within a very small study site that may not represent the whole catchment in terms of land use and soils. Both are homogeneous parameters on the study site, but vary in the catchment. The measured catchment response is a catchment-wide response signal, that was correlated to a study site pattern. Therefore these investigations should be repeated over the whole catchment and the cross correlation shall be checked against the one shown in figure 4.30.

Chapter 6

Conclusion & Outlook

6.1 Custom Sensor Network

The custom sensor network approach presented in this thesis was new as the majority of the used network components were build up from scratch. Although the entire system is reproducible from 100 % open sources, it cannot be concluded as a 100 % success.

6.1.1 DataLogger Micro

A lot of errors occurred, which were related to the data logging unit itself. These were discussed in section 5.1 (see p.93) and were all based on an issue in the power supply. All described aspects lead to a logger inventory of only nine units. It was not possible to handle more because of the high effort needed to read data and reprogram the device and a lack of time for mounting more. Therefore the author started into the field campaign with a too little amount of devices and suffered from frequent sensor failure, which were to some extent again related to the shi**y firmware. In consequence, the *DataLogger Micro* is concluded to **still offer enough space for improvements**.

6.1.2 Soil Moisture Sensor

The main conclusion for the soil moisture sensors is that the power consumption is still too high. This is mainly caused by a design error, which uses the main stabilized voltage supply (VCC-ST) to measure the soil resistance. The objective **viii - in principle**

solar driven is mainly affected by the misconfigurations in the sensor and logger unit (as discussed in section 5.1; p.93). The system is still consuming too much energy, but even very small solar panels should be able to keep the system running. Small panels delivering about 100 mA cost less than 2€[1]. These panels do produce enough energy during the day that even a small rechargeable battery is suitable[2]. Therefore, regarding the logger and sensor unit the objective **viii - in principle solar driven** passes as fulfilled.

6.1.3 Server

When concluding the custom sensor network one has to cast an eye onto the server as well. It is not only a part of the network, but the most important processing and data delivery services were programmed to be working on the Banana Pro™ as well. The device itself will not be concluded, as it was not developed by the author. There are many resources on the web dealing with Banana Pro™, or Raspberry Pies including device limitation overviews. The server was one of the key features enabling the author to process the amount of data as presented in the results. The main reasons were outlined in section 5.1.2 (see p.94).

6.2 Technical Lab Tests

6.2.1 Battery Characteristic Curve

The described battery characteristic curve lab test can be concluded to have worked out very well, but the issues in the power supply lie beyond a simple consumption monitoring, what is basically what a characteristic curve can be used for. Nevertheless, the curves were used to estimate the remaining lifetime of the data loggers in the field and a battery emptying occurred only in a single case, thanks to the characteristic curves.

[1] Seeed Studio Bazaar homepage. URL: http://www.seeedstudio.com/depot/05W-Solar-Panel-55x70-p-632.html. Accessed: February 9, 2016.

[2] Seeed Studio Bazzar homepage, smallest LiPo found. http://www.seeedstudio.com/depot/Crazyflie-20-Spare-240mAh-LiPo-battery-p-2116.html?cPath=84_147. Accessed: February 9, 2016.

6.2.2 Accuracy & Precision

There are two main conclusions from the accuracy and precision test results described in section 4.2.3 (see p.51). First, the commercial system performed way better than the custom and way better than its specifications suggested. Second, taking data quality management into consideration, the custom sensor network matches the precision and accuracy objectives and therefore the objective **iv - sufficient precision** (see 1.3; p.2).

It was described that the sensor was designed to work well and precise especially for wet soils. Resistances were chosen to cover the expected soil resistances at wetter conditions (see 3.3, p37) as the soil was expected to be wet during the winter months. The test showed that the sensor in fact was performing better at wet conditions as it was for dry ones. The offset for the custom sensors to the modeled soil moisture inside the soil column based on the directly measured water contents are increasing while the soil was drying. At more static conditions the residuals decreased again (see figure 4.4, p.55), but obviously because there were misreadings in the custom system (see figure 4.3, p.52).

6.3 Data Products

6.3.1 Soil Moisture

The soil moisture dataset recorded by the Decagon® EM50™ system using 5TE soil moisture sensors can be concluded to having worked as expected. The main objectives for a commercial soil moisture system are to be reliable, reproduceable and accurate. With the exception of two sensor failures at the measuring campaign beginning, of which one was caused by the author carelessly forgetting to connect one wire to the EM50™, no failures occurred. The soil moisture measurements were comprehensible and, taking the rainfall into consideration, as expected. **Therefore the EM50™/5TE system is a very robust solution for recording reliable soil moisturetime series.**

The custom *DataLogger Micro* system performed differently. The author could not get on the track of sensor C5 and even if it would not have been for this sensor, the data quality is unsatisfying, the amount of noise in the measurements is varying and the system experiences too many failures. All this together decreases the repeatability and thus the robustness of the reading is not satisfying. **From this point of view, the custom system failed.**

6.3.2 Connectivity - Range

The second and somehow most important data product was the range time series, which should be used as a proxy for catchment connectivity. In fact it is enough to inspect the two figures 4.15 (see p.71) and 4.18 (see p.76) showing the best fitting semivariogram each for the commercial and custom system, respectively. Any search for spatial pattern and any other product based on these pattern are directly depended on the goodness of fit for every single semivariogram. There is no spatial relation between the soil moisture measurements and their distances. Both variograms fit a random function into a point cloud and do not fit at all. **Therefore it has to be concluded that there is no spatial pattern in soil moisture.** As a consequence, at that point, the sensor network performance cannot be concluded, because there is nothing to conclude. Changing the process of creating the variograms was discussed in detail in section 5.3.3 (see p.99), but did not bring any better results. Antithetical, the variogram shape became worse and although some literature could be found **the author saw no chance of interpreting these results in terms of soil moisture patterns**.

Lastly, the bin sizes were equalized and the lag range limited as discussed in detail from p.102 ff. Not only did the semivariograms change their shape, but more important, the mean and median semivariance values for almost every bin do overlay now. From this, it can be concluded that whenever[3] the measurand is expected to show high variances, it is crucial to keep the **semivariogram bins at equal size** over keeping them at equal lag. This makes statistical numbers like the mean or median more robust and representative. Secondly, in case the measurand tends to show extreme values in the semivariance, the **median shall be used over the mean semivariance** value due to its improved robustness concerning extremes.

[3] This *whenever* is limited to the boundary conditions and limitations as discussed throughout chapter 5 and especially section 5.3.3 (see p.99).

6.3.3 Connectivity - Catchment Response

The runoff coefficient as shown in figure 4.27 (see p.89) can be concluded to describe the catchment response well. Most of the high coefficient are reached during the last third of November where the biggest rainfall events, as well as the maximum discharge values are reached. The described range time series shows similar dynamics as the runoff coefficient. Once one has ensured, there is reliable variogram describing the observed data, it can be concluded that **the state of connection in the observed catchment can be estimated by soil moisture patterns.** This was made obvious in figure 4.30 (see p.92).

6.3.4 Coming back to the Objectives

The main objectives for this thesis were presented in section 1.3 (see p.2). The previous conclusion sections already related these objectives, which shall be summarized here, as the main conclusion for this thesis.

(i) cost efficient

The raw material costs for one data logger unit, as 100 are produced, is about 12€in case all materials are bought in the EU. The mounting time for one unit sums to approx. 30 - 45 minutes. A student assistant would cost aprrox. another 10€for this period, therefore undercutting an overall price of 25$ is possible.

(ii) 100 % open source

This objective was achieved, as the hardware is published by the author and open and the software is described in appendix F and section 3.5.3, table 3.4 (see p.46), including ways of distribution.

(iii) highly adaptable

The combination of the presented Python module *hydras* in combination with the degree of freedom in configuring the *DataLogger Mirco* satisfy this objective. In fact, the amount of freedom in configuration is way to high, which caused a lot of problems. A detailed description of the *hydras* can be found in the documentation book, which can be found on the supplementary DVD. Additionally the heavy semivariance analysis, which was adapted for each subsection in the results, could only be handled within the time limits of this thesis as *hydras* made most of the work.

(iv) sufficient precision

This objective was only matched in parts. The soil moisture measurements were fine (with limitations, e.g. C5), while the quality analysis was worse than expected. During the lab tests, the custom system performed better than expected. The minimum objective on 2.5 % accuracy and precision in saturation, was not completely fulfilled, with a precision of 1.6 % and accuracy of 7.5 %.

(v) repeatable

This objective cannot be answered as the two soil moisture sensors were placed too close together in all cases. This lead to an influence in measurements of each other. In consequence the two sensors measured the exact value in all cases. An sufficient distance for the sensors while still measuring the same 'sample' could not be found by trial and error.

(vi) suitable for answering a exemplary hydrological issue

This objective was heavily concluded above and has to be answered with 'maybe'. The amount of sensor failures made an answer to this objective at present impossible.

(vii) automated

This objective was matched satisfyingly. The server is at a development level, where the raw data can be uploaded with only a few lines of code and the data is organized in the database and quality checked autonomously. With a few more lines of code various data views including statistics, quality information and even spatial interpolation can be requested using the *openhydro* package or the web application. Visiting the website http://mathesis.openhydro.de is highly recommended. Now.

(viii) in principle solar driven

As stated in section 6.1.2 (see p.106), although the power consumption is still way above the expected level, a solar operation is in principle possible.

6.4 Outlook

This section will give an overview on future developments for the custom sensor network approach, in order to transfer it from an approach to a working system one day. For the custom sensor network hardware the power consumption is still the main theme, while for the software side, the development of an sophisticated GUI stand in the foreground. The power supply circuit can be adapted by splitting the whole board design into data logger and sensor unit. Then the power supply is also split up and does not supply the microcontroller and the sensor at the same time. The main advantage would be, that the sensor voltage supply can be dimensioned to a much smaller maximum flux, as just the modulation of the resulting voltage caused by the soil is of interest. There are no loads on the sensor power circuit and therefore just a few mA of power are enough. Another advantage of splitting the two units is, that a very common communication protocol can be used, like the SDI protocol[4]. This serial communication protocol is not only very common in environmental science and almost all commercial logger can use it (like the EM50TM or Campbell Scientific products), but it would empower the system to interchange the 5TE and custom sensors between the two presented systems, as the 5TE is in fact using the SDI protocol. This would have decreased the work load of this thesis and would have made some of the technical lab tests more comparable and comprehensive. Furthermore, a new field of operation would be opened for the custom device, as it is much cheaper and could connect one or two 5TE, a more evened or randomized campaign design would have been possible. Then the custom system would not compete, but expand the commercial system. Using synergies over competition might have lead to more satisfying results in this thesis.

As this thesis used variograms for the main part of the analysis, it was natural to further use these variograms for an actual spatial interpolation of all measured values. On the supplementary DVD in the scripts folder, an *interpolate.py* can be found, which will use ordinary kriging for interpolation. The results can be found in the *results/kriging* folder, also on the DVD. The script can easily adapted to produce hourly or 15-minute interpolation over daily. These files could be used to generate a WMS[5] of the results

[4] Wikipedia article about SDI. URL:https://de.wikipedia.org/wiki/Serial_Digital_Interface. Accessed: February 29, 2016.

[5] WMS: Web Map Service; used to distribute spatial data. Wikipedia article: URL: https://de.wikipedia.org/wiki/Web_Map_Service. Accessed March 2, 2016.

e.g. by the geoserver software. Precisely this was done by the author. This WMS is included into an online visualization tool, which can be found at http://common-environment.org/mathesis. This tool will be expanded in the future in order to show hourly over daily data and offer WMS for interpolated soil moisture based on different variogram models in order to illustrate the differences and dependence on the correct model selection.

Appendix A

Glossary

ADC **-** An analog-to-digital converter [. . .] is a device that converts a continuous physical quantity (usually voltage) to a digital number that represents the quantity's amplitude.
The conversion involves quantization of the input, so it necessarily introduces a small amount of error. Instead of doing a single conversion, an ADC often performs the conversions (”samples” the input) periodically. The result is a sequence of digital values that have been converted from a continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal.
(source: Wikipedia, URL: https://en.wikipedia.org/wiki/Analog-to-digital_converter. Accessed: Septemper 22, 2015.)

GUI **-** In computer science, a graphical user interface or GUI [. . .] is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.
(source Wikipedia, URL: https://en.wikipedia.org/wiki/Graphical_user_interface. Accessed: December 24, 2015.)

CMS **-** A content management system is a computer application that allows publishing, editing and modifying content, organizing, deleting as well as maintenance from a central interface. Such systems of content management provide procedures to manage workflow in a collaborative environment. These procedures can be manual steps or an automated cascade. CMSs have been available since the late 1990s.
CMSs are often used to run websites containing blogs, news, and shopping. Many corporate and marketing websites use CMSs. CMSs typically aim to avoid the need for hand coding, but may support it for specific elements or entire pages.
(source: Wikipedia, URL: http://en.wikipedia.org/wiki/Content_management_system. Accessed: September 22, 2015.)

***PCB* -** A printed circuit board mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs can be single sided (one copper layer), double sided (two copper layers) or multi-layer (outer and inner layers).[...] Printed circuit boards are used in all but the simplest electronic products. Alternatives to PCBs include wire wrap and point-to-point construction. PCBs require the additional design effort to lay out the circuit, but manufacturing and assembly can be automated. Manufacturing circuits with PCBs is cheaper and faster than with other wiring methods as components are mounted and wired with one single part. Furthermore, operator wiring errors are eliminated.
(source Wikipedia, URL: https://en.wikipedia.org/wiki/Printed_circuit_board. Accessed: September 22, 2015.)

***SSD* -** A Solid State Drive [...] is a data storage device that uses integrated circuit assemblies as memory to store data persistently. [...] Compared with electromechanical disks, SSDs are typically more resistant to physical shock, run silently, have lower access time, and less latency. However, while the price of SSDs has continued to decline over time,consumer-grade SSDs are still roughly six to seven times more expensive per unit of storage than consumer-grade HDDs.
(source: Wikipedia, URL: http://en.wikipedia.org/wiki/Solid-state_drive.
Accessed: September 30, 2015.)

Appendix B

Data Access

Two main ways of accessing most data used in this thesis are presented here. On the one hand access using the web-frontend, which can be found at http://openhydro.de, and using the Python interface module *openhydro* on the other hand. The table B.1 below gives an overview of each measuring station and sensor name during the field campaign, its id in the database and the URL for access. You will be guided to the raw database information. Accessing the data using the map on http://openhydro.de will guide you to a styled version inside a web application.

Table B.1: For any measuring station and sensor used during this thesis, the field campain name (name), the database id (id) and the URL for accessing the Information using the web-frontend (web) is given.

name	**id**	**web**
M1	501	http://openhydro.de/app/details.py/?station=501
Au1	504	http://openhydro.de/app/details.py/?station=504
Au2	502	http://openhydro.de/app/details.py/?station=502
Au3	503	http://openhydro.de/app/details.py/?station=503
D1	505	http://openhydro.de/app/details.py/?station=505
C1	601	http://openhydro.de/app/details.py/?station=601
C2	602	http://openhydro.de/app/details.py/?station=602
C3	603	http://openhydro.de/app/details.py/?station=603
C4	604	http://openhydro.de/app/details.py/?station=604
C5	605	http://openhydro.de/app/details.py/?station=605
C6	606	http://openhydro.de/app/details.py/?station=606
C7	607	http://openhydro.de/app/details.py/?station=607
C8	608	http://openhydro.de/app/details.py/?station=608
C9	609	http://openhydro.de/app/details.py/?station=609

Appendix C

Database

This appendix sums some special information about the used PostgreSQL database, that has to be focused, or is somehow important for this thesis.

C.1 Trigger Functions

The quality checks applied to the soil moisture data while uploading to the database are implemented by the database trigger functions listed below. These functions are only tested on the given PostgreSQL system using version 9.3 on a Debian 7 operating system.

```
-- DROP TRIGGER plausibility_check ON data.dtbl_default_soil_moisture;

CREATE TRIGGER plausibility_check
  BEFORE INSERT
  ON data.dtbl_default_soil_moisture
 FOR EACH ROW
  EXECUTE PROCEDURE data.moisture_plausibility();

-- DROP FUNCTION data.moisture_plausibility();

CREATE OR REPLACE FUNCTION data.moisture_plausibility()
  RETURNS trigger AS
$BODY$
BEGIN
-- set flag_id to 1 (invalid) if the minimum or maximum value is exceeded by val; defaults to 0 and 1
IF NEW.val <= (SELECT CASE WHEN (min_val IS NOT NULL) THEN min_val ELSE 0 END FROM structure.tbl_sensor WHERE id=NEW.sensor_id)
OR NEW.val >= (SELECT CASE WHEN (max_val IS NOT NULL) THEN max_val ELSE 1 END FROM structure.tbl_sensor WHERE id=NEW.sensor_id) THEN
        NEW.flag_id:=1;
END IF;
RETURN NEW;
END;
$BODY$
  LANGUAGE plpgsql VOLATILE
```

LISTING C.1: moisture_plausibility.sql.
The trigger function will replace the flag by the *invalid flag* for any record where the soil moisture value is exceeding the data range defined by the parent sensor as *min_val* and *max_val*. Defaults to 0 and 1, respectively. Will be triggered before each INSERT operation.

```sql
-- DROP TRIGGER persistancy_check ON data.dtbl_default_soil_moisture;

CREATE TRIGGER persistancy_check
  BEFORE INSERT
  ON data.dtbl_default_soil_moisture
  FOR EACH ROW
  EXECUTE PROCEDURE data.moisture_persistancy();

-- DROP FUNCTION data.moisture_persistancy();

CREATE OR REPLACE FUNCTION data.moisture_persistancy()
  RETURNS trigger AS
$BODY$
begin
if (select val from data.dtbl_default_soil_moisture where sensor_id=NEW.sensor_id and tstamp < NEW.tstamp order by tstamp desc limit 1) < NEW.val THEN
if (select sum(val) from data.dtbl_default_precipitation where sensor_id=30 and tstamp <= NEW.tstamp and tstamp > NEW.tstamp - interval '24 hours' group by sensor_id) < 0.2 THEN
        NEW.flag_id:=2;
end if;
end if;
return new;
end;
$BODY$
  LANGUAGE plpgsql VOLATILE
```

LISTING C.2: moisturePersistancy.sql.
The trigger function will replace the flag by the *questionable flag*. This is a custom trigger only valid for this thesis as it will always check for soil moisture raises against the rainfall sensor of id 30. Will be triggered before each INSERT operation.

Appendix D

Analysis Scripts

TABLE D.1: Exact python versions used in this thesis. This environment can be imitated by using the Anaconda Python environment (https://www.continuum.io/downloads) and install the excat version given below.

Module	Version
Python	2.7.10
IPython	1.1.0
numpy	1.8.0
pandas	0.15.1
matplotlib	1.3.1
scipy	0.13.3

This Chapter includes all used scrips based analysis of this thesis. This includes scripts for data management, data visualization, statistics and calculations. Where not otherwise state, the scripts are written in the Python programming language of version 2.7.10. As some of the core modules are under steady development, the version of the most important modules are given in table D.1. This is especially important for the pandas package as this is not fully backward compatible. If using a Python environment like Anaconda (https://www.continuum.io/downloads), what is highly recommended, a specific version instead of the actual version can be installed like:

```
$> conda install pandas==0.15.1
```

In the Anaconda command prompt, this will install the pandas package of version 0.15.1 in the active Python environment.

In the following, all used scripts are shown, either by their content or by a short description in combination with their location on the supplementary DVD. Unfortunately, due to a laptop breakdown during the field campaign, the original custom data logger dumps and their upload script got lost, as they were only saved on the damaged logger at that stage. Therefore only the uploaded data is available.

accuracy.py

file name	accuracy.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/accuracy.py"
description	Analysis script for accuracy and precision test scenario described in section 2.4.3 (see p.31). Calculates sensor accuracy and precision, opens visualization plots and dumps statistical test output.

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 09 10:31:13 2015

@author: Mirko Maelicke <mirko.maelicke@open
"""
import os
import numpy as np
import pandas as pd
from datetime import datetime as dt
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit


os.chdir(os.path.join(os.path.dirname(__file__), '../data/Test_raw/'))

# load the data
em50 = pd.read_csv("accuracy_em50.csv", sep="\t", parse_dates=[0], date_parser=lambda x:dt.strptime(x, '%Y/%m/%d %H:%M'), index_col=0)
custom = pd.read_csv("accuracy_custom.csv", sep="\t", parse_dates=[0], date_parser=lambda x:dt.strptime(x, '%Y/%m/%d %H:%M'), index_col=0)
# merge together
df = pd.merge(em50, custom, left_index=True, right_index=True)
df.columns = ['em50', 'custom']

# get the real data
w = pd.read_csv("accuracy_weights.csv", sep=";", decimal=',', skiprows=2, parse_dates=[0], date_parser=lambda x:dt.strptime(x, '%d.%m.%Y %H:%M'), index_col=0, usecols=['date', 'water content'])

#---------
# analysis

# get the indices of weight measurements on a relative scale
idx = [np.where((em50.index == w.index[i]))[0][0] / float(len(em50)) for i in range(len(w))]
val = [_[0] for _ in w.values]

def f(x, a, b, c):
        if type(x) == np.float64:
                return a * np.exp(-b*x*100) + c
        else:
                a = np.ones(x.size) * a
```

```
                b = np.ones(x.size) * b
                c = np.ones(x.size) * c
                return map(f, x, a, b, c)

# fit the model; max 10000 iterations
cof, cov = curve_fit(f, np.array(idx), np.array(val), maxfev=10000)

x = np.linspace(0, 1, len(df))
y = f(x, *cof)

# set the real values from model
df['modeled'] = y

#----------------------------------------------
# plot
#----------------------------------------------
# some options
plt.rc('text', usetex=False)
plt.rc('font', family='serif')

# first plot
ax = w.plot(style='Dk')
# second nito first
df.plot(ax=ax)

# only styling:
plt.xlabel('December 7, 2015 - December 9, 2015 ', fontsize=14)
plt.ylabel('water content', fontsize=14)
plt.legend(loc=3, bbox_to_anchor=(0.,1.02,1, 0.102), ncol=4, mode="expand", borderaxespad=0.)
plt.tight_layout()
# crazy texting stuff, ignore it
#plt.rc('text', usetex=True)
plt.text((plt.xlim()[1] - plt.xlim()[0]) * 0.05 + plt.xlim()[0], (plt.ylim()[1] - plt.ylim()[0]) * 0.95 + plt.ylim()[0],
  r"model: $%.3f * e^{-%.3f * 100x} +%.3f$" % tuple([float(_) for _ in cof]), fontsize=14)
plt.show()
# write some statistics into the frame

#----------------------------------------------
# residuals, accuracy etc.
#----------------------------------------------
residuals = pd.DataFrame({'em50':df.em50 - df.modeled, 'custom':df.custom - df.modeled})

# create the indexer
x = range(len(residuals))
f, axes = plt.subplots(2, 1, sharex=True, sharey=True)
axes[0].bar(x, residuals.em50.values, edgecolor='green', facecolor='green')
axes[1].bar(x, residuals.custom.values, edgecolor='blue', facecolor='blue')

# some styling
t = plt.xticks()
pos = [int(_) for _ in t[0]][1:-1]
names = [dt.strftime(_, '%d.%m %H:%M') for _ in residuals.index[pos]]
plt.xticks(pos, names, rotation=45)
plt.ylabel('water content residuals')
plt.tight_layout()
plt.show()
```

```python

#-----------------------------------------------
# statistics
#-----------------------------------------------
# residuals
out_acc = "Accuracy: (Mean Residuals):\n"
out_acc += "EM50: %.3f\t\tCustom: %.3f\n\n" % (residuals.em50.mean(), residuals.custom.mean())

# first get the shapiro test
from scipy.stats import shapiro
sp = (shapiro(residuals.em50.values)[1], shapiro(residuals.custom.values)[1])
out_acc += "Shapiro Test\n"
if  sp[0]>= 0.05 and sp[1] >= 0.05:
        # H0 cannot be rejected ==> normal distribution
        from scipy.stats import ttest_rel as test_function
        out_acc += "p: EM50 %.3f    Custom %.3f\n\n\n" % sp
        out_acc += "Students paired t-test\n\n"
else:
        # H0 is rejected ==> no normal distribution
        from scipy.stats import wilcoxon as test_function
        out_acc+= "p: values < 0.05\n\n\n"
        out_acc+= "Wilcoxon signed-rank test for paired samples\n\n"
# IMPORTANT:
# importing both as the same is possible, as both test reject H0 on p < 0.05
# this is not true for all parametric / non-parametric combinations
t, p = test_function(residuals.em50.values, residuals.custom.values)

# keep or reject H0
if p < 0.05:
        out_acc += "p < 0.001\n" if p < 0.001 else "p: %.4f\n" % p
        out_acc += "H0 rejected, real difference in ranks observed.\n"
else:
        out_acc += "p: %.4f\n" % p
        out_acc += "H0 cannot be rejected, residuals are random distributed.\n"
# print result and save to text dump
with open("accuracy_result.txt", "w") as fs:
        fs.write(out_acc)
        print out_acc


# accuracy
out_pre = "Precision (Standard Deviations):\n"
out_pre += "EM50:  %.3f\t\tCustom:  %.3f\n" % (df.em50.std(), df.custom.std() )

# first get the shapiro test
sp_pre = (shapiro(df.em50.values)[1], shapiro(df.custom.values)[1])
out_pre += "Shapiro Test\n"
if  sp_pre[0]>= 0.05 and sp_pre[1] >= 0.05:
        # H0 cannot be rejected ==> normal distribution
        from scipy.stats import bartlett as test_function
        out_pre += "p: EM50 %.3f    Custom %.3f\n\n\n" % sp_pre
        out_pre += "Bartlett test for equal variances\n\n"
else:
        # H0 is rejected ==> no normal distribution
        from scipy.stats import levene as test_function
        out_pre+= "p: values < 0.05\n\n\n"
        out_pre+= "Levene test for equal variances\n\n"
# IMPORTANT:
# importing both as the same is possible, as both test reject H0 on p < 0.05
```

```python
# this is not true for all parametric / non-parametric combinations
t, p = test_function(df.em50.values, df.custom.values)

# keep or reject H0
if p < 0.05:
        out_pre += "p < 0.001\n" if p < 0.001 else "p: %.4f\n" % p
        out_pre += "H0 rejected, true difference in variances.\n"
else:
        out_pre += "p: %.4f\n" % p
        out_pre += "H0 cannot be rejected, equal variances.\n"
# print result and save to text dump
with open("accuracy_precision.txt", "w") as fs:
        fs.write(out_pre)
        print "\n\n"
        print out_pre
```

Listing D.1: accurarcy.py

bcc.py

file name	bcc.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/bcc.py"
description	Analysis script for battery characteristic curve fitting and visualization like described in section 2.4.2 (see p.30). Shall be run in pylab mode.

```python
#-----------------------------------------------------
#
# bcc  -  battery characteristic curve
#
# by Mirco Maelicke as part of the master thesis
# A custom sensor network approach for detecting
# hydrological connectivity by soil moisture patterns
#-----------------------------------------------------

import pandas as pd
import numpy as np
from pandas import DataFrame, Series
import os, glob
from scipy.optimize import curve_fit

# jump to script location
os.chdir(os.path.dirname(__file__))

# load all test files
filelist = glob.glob("../data/Test_raw/BCC_*.txt")
chunks = []

# extract the adc values from each file and remove
```

```python
# file end character 255 and 25
for fname in filelist:
	with open(fname, "r") as fs:
		s = fs.read()
	s = [int(x) for x in s.split() if int(x) not in [25, 255]]
	chunks.append(s)

# organize into df
df = DataFrame(chunks).T

# still missing the coversion function

# now create the model
# to be fitted onto the data
def f(x,a,b,c,d):
	return a*x**3+b*x**2+c*x+d

all_models = []
# save and plot every bcc model
for i, bcc in df.iteritems():
	cof, cov = curve_fit(f, np.linspace(0, 1, len(bcc)),  bcc.values)

	# create model with 100 values
	mod_x = np.arange(0,1, 0.01)
	mod_y = f(mod_x, *cof)
	all_models .append(mod_y)
	#save
	pd.DataFrame({'t':mod_x, 'V':mod_y}).to_csv("../data/Test_raw/BCC models/BCC_{0}.csv".format(i), index=False)
	# plot
	plt.plot(mod_x, mod_y, '--g')

# plot_mean, label and save the figure
my = []
for i in range(100):
	my.append(mean([x[i] for x in all_models]))
plt.plot(np.arange(0,1,0.01), my, '-r', linewidth=2)
plt.xlabel('normalized time [-]')
plt.ylabel('battery voltage [V]')
plt.gcf().savefig('../graphs/bcc.pdf', dpi=400)
```

LISTING D.2: bcc.py

cconvert.py

file name	cconvert.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/cconvert.py"
description	Converts the CTD recorded gauging measurements by applying a weir formula as described in section 2.3.1 (see p.22) and uploads the result to the *Openhydro* database.

```python
# -*- coding: utf-8 -*-
"""
Created on Thu Feb  4 08:13:11 2016

@author: mmaelicke
"""

import openhydro as oh
import pandas as pd
import os, math

path = os.path.join(os.path.dirname(__file__), '../data/klima/')

# load the stages
df = pd.read_csv(os.path.join(path, "CTD_Au.txt"), skiprows=3, sep="\s+", header=
    None, parse_dates={'date':[0,1]}, date_parser=lambda x, y: pd.datetime.
    strptime('{0} {1}'.format(x,y),'%d.%m.%y %H:%M'), engine='python', index_col=
    'date')[2]

# parameters for conversion
# parameters from Au ar set as optional parameters
def weir(stage, mu=0.67, alpha=22.5, g=9.81, w=0.04, h=0.3, use_cm=True):
    t1 = 1000.*((8./15.)*mu*math.tan(math.radians(alpha)))
    t2 = math.sqrt(2.*g)
    t3 = math.pow((stage / 100. - w), (5./2.))

    return t1*t2*t3

# calculate discharge
q = pd.DataFrame({'discharge':df.apply(weir)})

# load to Sensor of id 32 into db
sensor = oh.Sensor(32)
if sensor['unit_symbol'] == 'm3/s':
    imp = q.apply(lambda x: x / 1000.).copy()
else:
    imp = q.copy()

# import
sensor.importData(imp, quality_flag=10)
```

Listing D.3: cconvert.py

em50import.py

file name	em50import.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/em50import.py"
description	After prepare.py was applied to the original EM50^{TM} CSV dumps, this script will import the p reprocessed files into the database automatically.

interpolate.py

file name	interpolate.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/interpolate.py"
description	Batch process for Kriging interpolation of the commercial system measurements. Based on the setting, one raster ASCII file of given resolution for each given time step is given. This script was used in appendix H (see p.144).

overview.py

file name	overview.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/overview.py"
description	Shall be run in pylab mode. This produces an overview plot for both, the custom and commercial sensor network. These results are shown in figure 4.8 (see p.61) and figure 4.13 (see p.67).

prepare.py

file name	prepare.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/prepare.py"
description	Processes the original EM50$^{\text{TM}}$ CSV dumps by simplifying the file structure and changing the NA value. This enables the em50import.py script to automatically upload a varying amount of temporal overlying EM50$^{\text{TM}}$ dumps into the db without producing duplicates. Pretty cool.

```
import os, glob

os.chdir(os.path.join(os.path.dirname(__file__), "../data/Au_raw/" ))

for station in ["AU1", "AU2", "AU3"]:
        filelist = glob.glob("{0}*.txt".format(station))

        for fname in filelist:

                with open(fname, "r") as f:
                        txt = f.read().replace("* * * ", "NA")

                lst = glob.glob("processed/{0}*.dat".format(station.lower()))
                with open("processed/{0}_{1}.dat".format(station.lower() ,len(lst) + 1), "w") as f:
                        f.write(txt.replace("\t", ","))
```

LISTING D.4: prepare.py

range_analysis.py

file name	range_analysis.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/range_analysis.py"
description	The results produced by the semivariance.py script and saved to CSV are checked for statistical significance in the results. This script does also dump most of the tables related to model selection shown in chapter 4 in LaTeX language.

response.py

file name	response.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/response.py"
description	Loads discharge and rainfall data from the database. All results presented in section 4.4.3 (see p88) are created by this script, including the double mass curve, BFI and runoff coefficient calculation and visualization.

semivariance.py

file name	semivariance.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/semivariance.py"
description	*Caution! Depending on your hardware, this script can take up to a few hours. Stop all other processes.* Calculates the semivariograms for all defined models for a given network and time step. Simply change the *grouper* option to change the temporal resolution. The results are saved into the "./results/" folder, including the semivariance images, result CSV, result overview plots and one PDF file for each model including all variograms. Basically all figures in the whole section 4.4.2 can be found in the result folders.

timestamp.py

file name	timestamp.py
author	Mirko Mälicke
license	Creative Commons Attribution-ShareAlike 4.0 International
location	"./scripts/timestamp.py"
description	After calculating the time step deviations per second for all tested periods, this script will visualize the results as shown in figure 4.5 (see p.57) and figure 4.6 (see p.58).

Appendix E

Datalogger

E.1 Layout

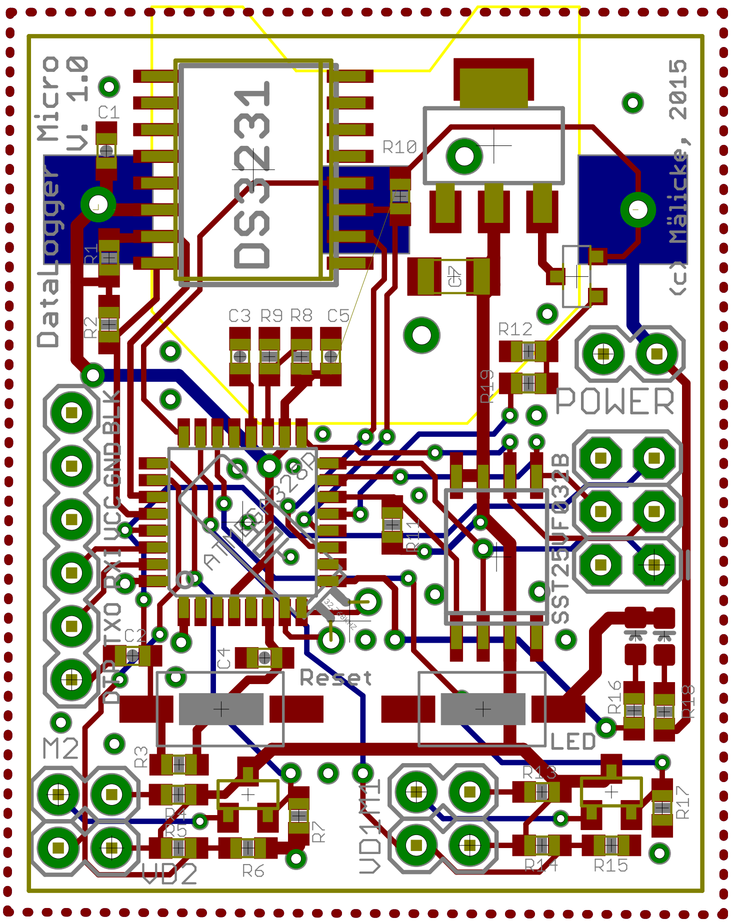


FIGURE E.1: PCB Layout for the 1.0 Version of the data logger used in this thesis. © Mirko Mälicke, 2015.

E.2 Schematics

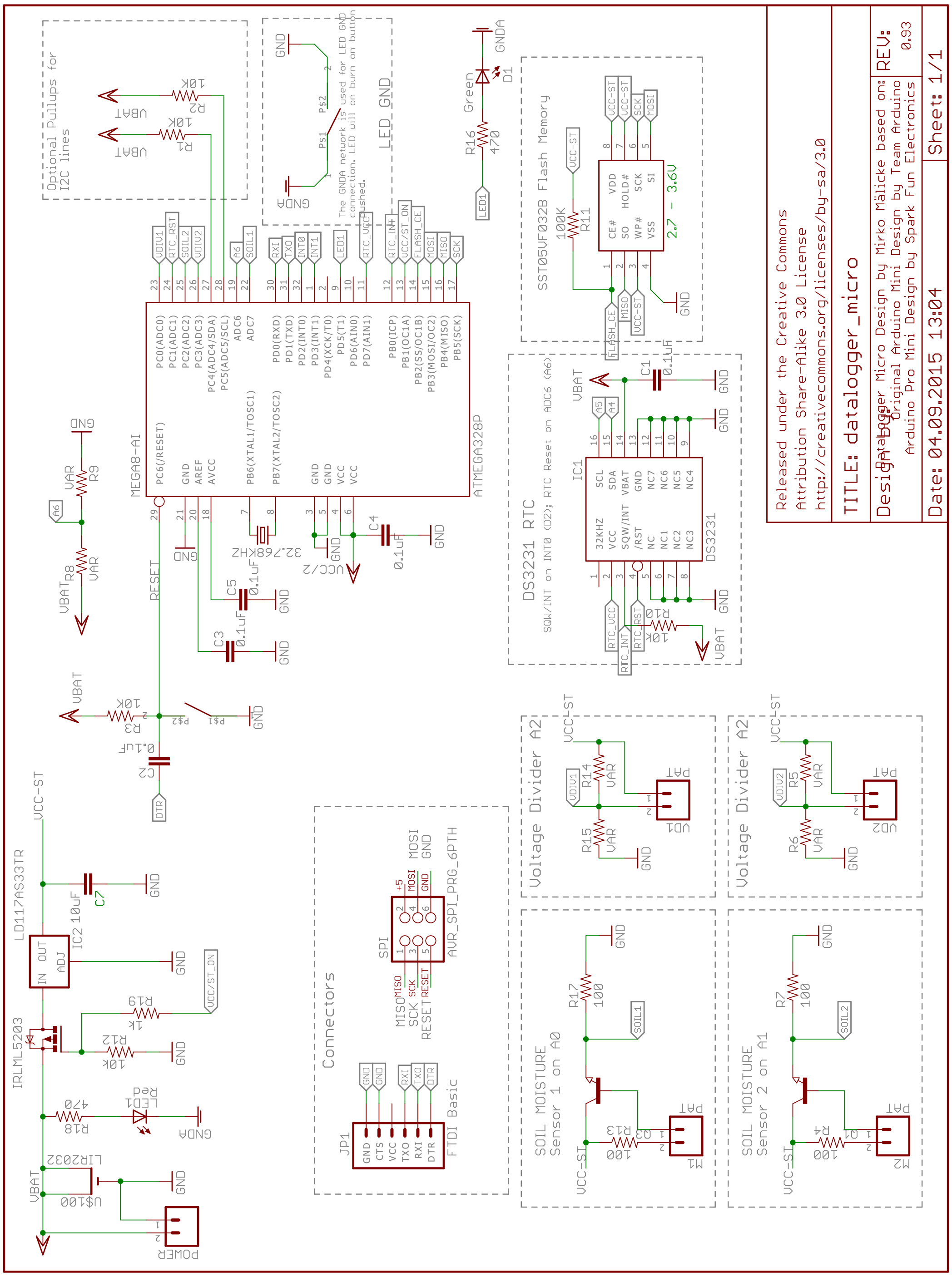


FIGURE E.2: Schematics for the 1.0 Version of the data logger used in this thesis. © Mirko Mälicke, 2015.

Appendix F

Firmware

This appendix includes all used firmware on the custom data loggers. This firmware was completely developed during and is therefore part of this thesis. If not otherwise noted, this software in written in the AVR-C language and property of the author.
A complete firmware version consists of a header-file (.h) and a C-file (.c) as well as all included files in the header file. The header file as well as all included files within this header are the same for all software versions presented here. These files can be found in section F.1 and will be referred to as the core-firmware. But different parts of this thesis use different firmware versions. These files are can be found in section F.2 and will be referred to as the main-firmware. A firmware version can be understood as a single script written for one purpose (like a battery characteristic curve test), that still uses the same resources as all other scripts. This way, a core function like saving a byte to the flash memory has to be implemented only once and is therefore part of the core firmware. These functions will behave exactly the same in all main versions.

F.1 Core Firmware

The core firmware, including the *DataLoggerMicro.h* header file, which is strictly speaking a part of the main firmware, can be found on the supplementary DVD, as they are all in all too long for listing them here. These files can be found in the "./firmware" folder. The knowledge of burning AVR-C firmware onto a ARM microcontroller is implied.

F.2 Main Firmware

BatteryLife.c

```c
/*
 * BatteryLife.c
 *
 * Can be used to evaluate Battery Life.
 * This script will use the 32.768Hz osc. as internal RTC to wakeup
 * every second and increase a timer counter until it equals TIMESTEP.
 * Then a marker is written to memory and the timer counter is reset.
 *
 * Created: 30.09.2015 09:03:11
 *  Author: Mirko Maelicke
 */

#include "DataLoggerMicro.h"

uint32_t tcounter = 0;

// redefine the Timestep
#ifdef TIMESTEP
#undef TIMESTEP
#endif
#define TIMESTEP 900   // in seconds

int main(void)
{
        DDR_LED |= (1 << DD_LED);               // LED1 output
        DDR_VCC_ST |= (1 << DD_VCC_ST); // VCC_ST output
        rtc32khz_init();                                // init RTC
        spi_init();                                             // init SPI

// UNCOMMENT THIS BLOCK FOR ERASE OR DUMP FLASH CONTENT
#if 0
        uart_init();                                    // init UART
        uart_puts("Hello, World!\nFlash Content:");
        sst_init();                                             // init flash
        vcc_off();
        delay_ms(1000);

        vcc_on();
        delay_ms(10);
        sst_init();
//      sst_erase();                    // uncomment for erasing
//      sst_putc(sst_last_addr(), 'A');
        sst_dump();                                             // dump content
        uart_puts("\n");
#endif
        vcc_off();
        sei();                                                  // enable interrupts

        while(1)
        {

                set_sleep_mode(SLEEP_MODE_PWR_SAVE);
                sleep_mode();
```

```
                // wait 30 us after wakeup
                _delay_us(30);

                if (tcounter == TIMESTEP)
                {
                        cli();                                  // disable interrupts
                        tcounter = 0;                   // reset timer counter

                        // PWR on and write C (== 0x043) onto flash
                        vcc_on();
                        delay_ms(5);
                        sst_init();
                        sst_putc(sst_last_addr(), 'C');
                        _delay_us(20);
                        sst_wait_busy();
                        vcc_off();
                        sei();                          // enable interrupts again
                }
        }
        return 0;
}
```

Listing F.1: BatteryLife.c

MoistureToSST.c

```
/*
 * DataLoggerMicro.c
 *
 * Created: 03.09.2015 09:03:11
 *  Author: Mirko Maelicke
 *
 * The main logging script.
 * logs the initialized sensors to SST.
 * The timestep is given in DataLoggerMicro.h
 */

#include "DataLoggerMicro.h"

/***********************/
/* global variables    */
/***********************/
// tcoutner declaration
#if TIMESTEP > 65535
volatile uint32_t tcounter = 0;
#endif
#if TIMESTEP <= 65535 && TIMESTEP > 255
volatile uint16_t tcounter = 0;
#endif
#if TIMESTEP <= 255
volatile uint8_t tcounter = 0;
#endif

/*
 * handle Timer2 overflow interrupt
```

```
 * the global variable tcounter has to be declared and defined
 */
ISR(TIMER2_OVF_vect){
	// increment the tcounter by one
	tcounter++;
}

int main(void)
{
	DDR_LED |= (1 << DD_LED);			// LED1		output
	DDR_VCC_ST |= (1 << DD_VCC_ST);		// VCC_ST	output
	rtc32khz_init();
	spi_init();
	measure_init();
	sei();		// enable interrupts

	while(1)
	{

		set_sleep_mode(SLEEP_MODE_PWR_SAVE);
		sleep_mode();
		// wait 30 us after wakeup
		_delay_us(30);

		if (tcounter == TIMESTEP)
		{
			cli();		// disable interrupts
			tcounter = 0;


//			vcc_on();
			delay_ms(20);
			adc_start();
			sst_init();
			delay_ms(10);

			log_port(VDIVin, SMOOTH_VDIV);
			log_port(SOIL1, SMOOTH_SOIL);
//			log_port(SOIL2, SMOOTH_SOIL);

//			vcc_off();
			sei();
		}
	}
	return 0;
}
```

Listing F.2: MoistureToSST.c

Erase.c

```
/*
 * Erase.c
 *
 * Created: 03.09.2015 09:04:22
 *  Author: Mirko Maelicke
 *
```

```
 * Will erase the SST
 */

#include "DataLoggerMicro.h"

/***********************/
/* global variables    */
/***********************/
// tcoutner declaration
#if TIMESTEP > 65535
volatile uint32_t tcounter = 0;
#endif
#if TIMESTEP <= 65535 && TIMESTEP > 255
volatile uint16_t tcounter = 0;
#endif
#if TIMESTEP <= 255
volatile uint8_t tcounter = 0;
#endif

int main(void)
{
        DDR_LED |= (1 << DD_LED);                       // LED1         output
        DDR_VCC_ST |= (1 << DD_VCC_ST);         // VCC_ST       output
        spi_init();

        uart_init();
        uart_puts("Erasing SST...\n");
        sst_init();
        sst_erase();
        uart_puts("done.\n");


        vcc_off();
        sei();

        while(1)
        {
                // just go to sleep
                set_sleep_mode(SLEEP_MODE_PWR_DOWN);
                sleep_mode();
        }
        return 0;
}
```

Listing F.3: Erase.c

SSTToSerial.c

```
/*
 * SSTToSerial.c
 *
 * Created: 03.09.2015 11:45:01
 *  Author: Mirko Maelicke
 *
 * Dumps the SST content to the serial port.
 */

```

```c
#include "DataLoggerMicro.h"

/************************/
/* global variables     */
/************************/
// tcoutner declaration
#if TIMESTEP > 65535
volatile uint32_t tcounter = 0;
#endif
#if TIMESTEP <= 65535 && TIMESTEP > 255
volatile uint16_t tcounter = 0;
#endif
#if TIMESTEP <= 255
volatile uint8_t tcounter = 0;
#endif

/*
 * handle Timer2 overflow interrupt
 * the global variable tcounter has to be declared and defined
 */
ISR(TIMER2_OVF_vect){
	// increment the tcounter by one
	tcounter++;
}

int main(void)
{
	DDR_LED |= (1 << DD_LED);				// LED1		output
	DDR_VCC_ST |= (1 << DD_VCC_ST);			// VCC_ST	output
	spi_init();

	uart_init();
	uart_puts("Hello, World!\nFlash Content:\n");
	sst_init();
	delay_ms(1000);

	// dump in 3 column layout,
	// change to amount of sensors + 1
	sst_dump_log(3);
	uart_puts("\n");
	vcc_off();
	sei();

	while(1)
	{
		// just go to sleep
		set_sleep_mode(SLEEP_MODE_PWR_SAVE);
		sleep_mode();
	}
	return 0;
}
```

Listing F.4: SSTToSerial.c

Appendix G

Photos

This appendix collates different photos taken during, or somehow related to this thesis. Some have only supportive or informative character and might not be referenced somewhere in this thesis.

Figure G.1: This is the gauging station (right) and the climate station (left) that were installed at the study site prior to this thesis. These two stations are referenced as D1 (gauging) and M1 (climate) in this thesis. This id does also identify the stations in the *Openhydro* database.

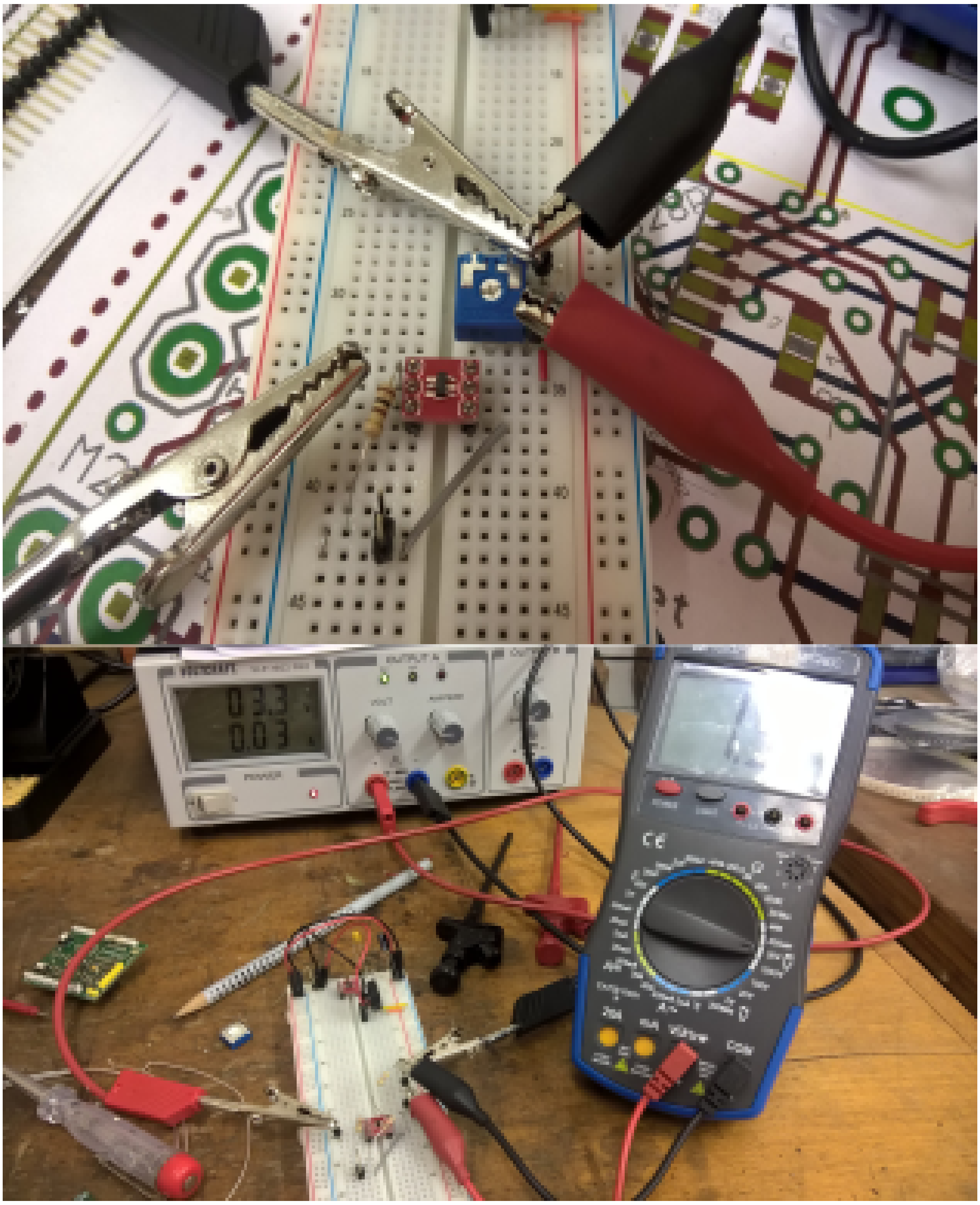


FIGURE G.2: Photo of the sensor calibration test. The sensor circuit was rebuild on the white breadboard shown in the middle and rebuild on the top photo. The power supply produced 3.3 V as shown in the display in the back. The resistors were fitted in their value until the mulitmeter (on the right side) showed exactly 1.1 V. On the top photo the resistor used for the sensor was replaced with a potentiometer in order to try different magnitudes of resistances until a suitable was found. The exact value was then determined in a second approached as shown in the bottom photo.

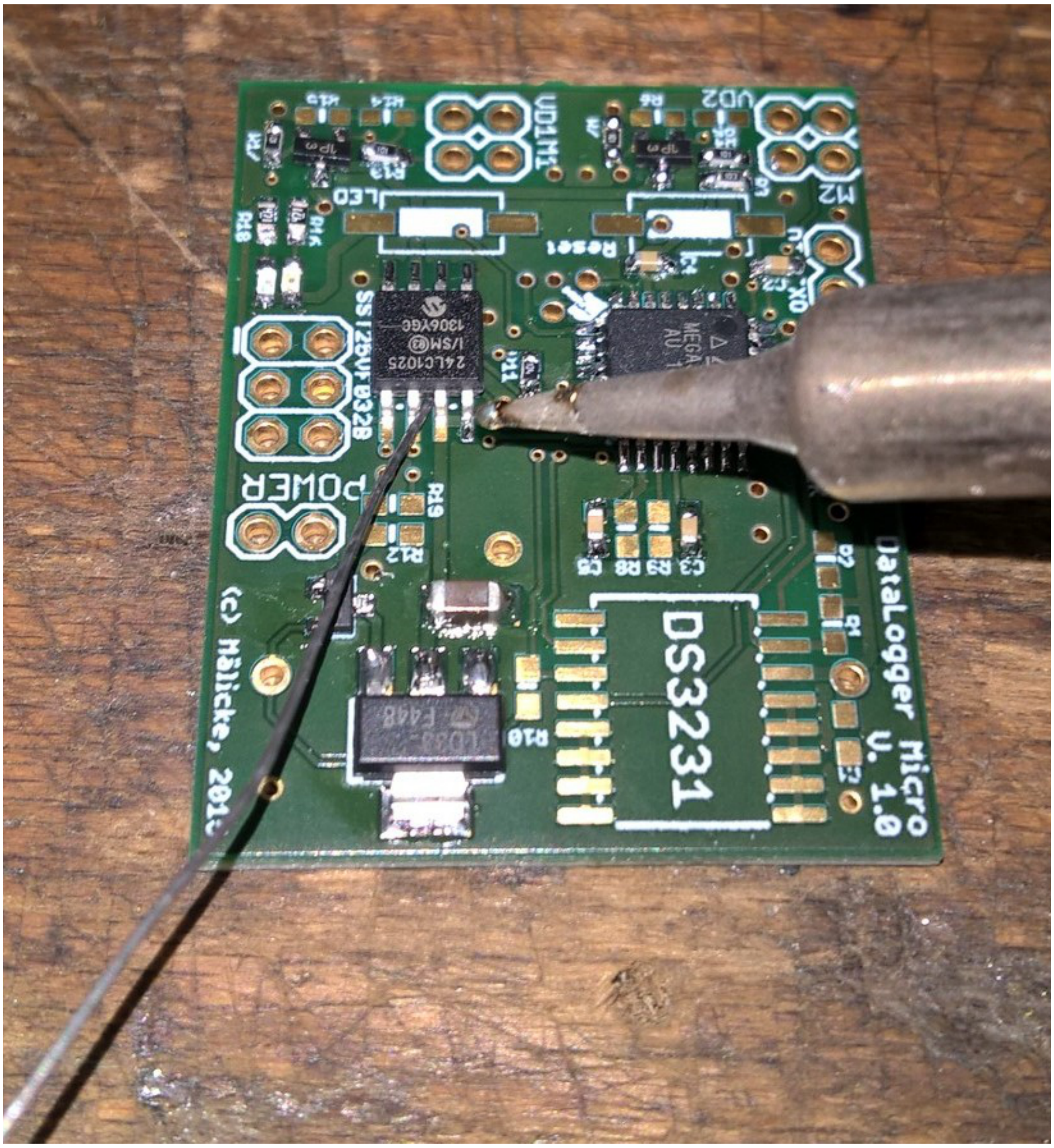


Figure G.3: One of the *DataLogger Micro* devices during the mounting process. On the photo, the flash chip is being mounted. For comparison, the soldewring iron is 0.5 mm in diameter and the contact foots of the flash chip are of 0.6 mm distance.

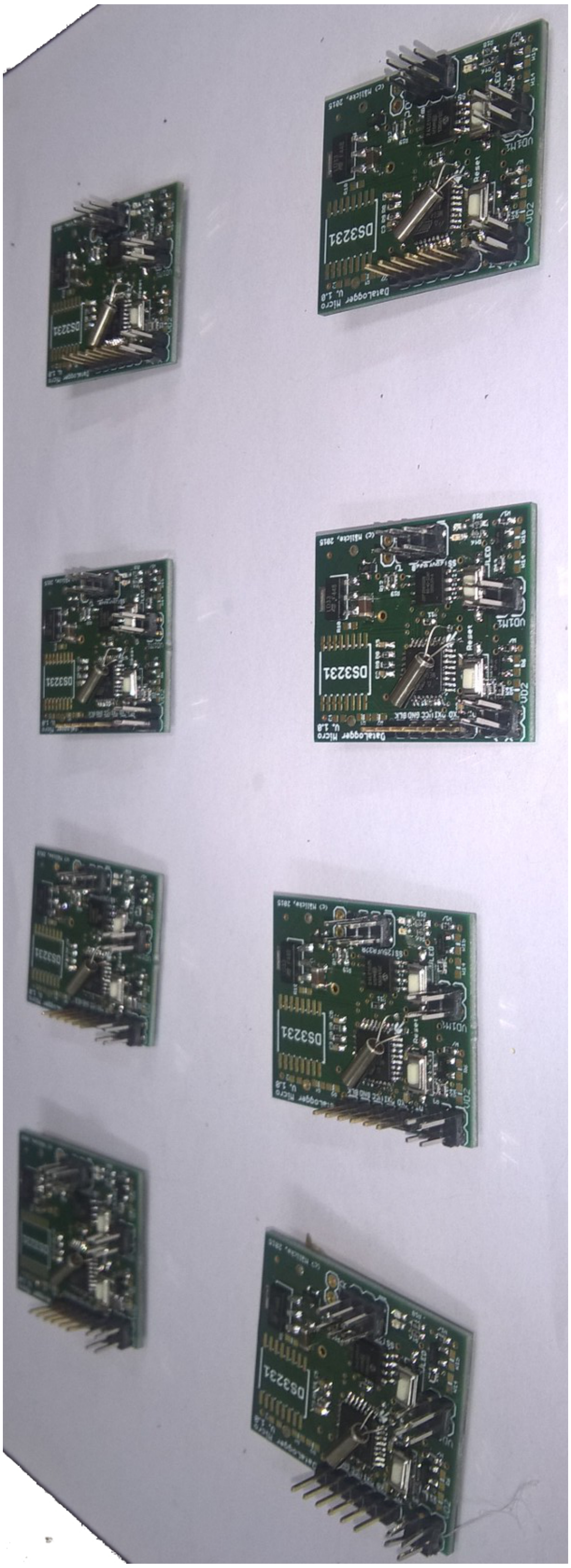


FIGURE G.4: Production View

Appendix H

Kriging

This chapter illustrates the benefits of using a sophisticated database application like *Openhydro* in combination with its Python modules. As a example application, a python script was written, which will use the recorded data to interpolate the soil moisture to the full extend of the study site by Ordinary Kriging. Hereby the script can handle different variogram models and temporal resolutions by simple option setting. The script can be found in the scripts folder of the supplementary DVD and is called *interpolate.py*. The option setting is shown below in order to illustrate the simplicity.

```
# params
cellsize = 1                  # cellsize in meter; data will be projected
                              # to DHDN Z3
variogram_model = 'linear'    # 'linear', 'gaussian', 'spherical',
                              # 'exponential' or 'power' supported
resolution = 'D'              # 'H' for hour; 'D' for day, 'M' for month
                              # 'min' for minute
                              # can be multiplied like '15min' for 15
                              # minute timesteps
```

The script was used to produce interpolated soil moisture maps of the study site. The interpolation for November 18, 2015 and November 20, 2015 are shown in figure H.1 and figure H.2. Both maps are oriented to north and are of 75 m x 40 m extend. These are of course not fully featured map including all necessary information like the coordinate system, orientation, name, shown value, color bar and so on, but for an on the wing calculation it's not that bad.

FIGURE H.1: Interpolated soil moisture map of November 18, 2015. The image is oriented to the north and the color represents saturation values ranging from 22.2 % to 23.14 % saturation.

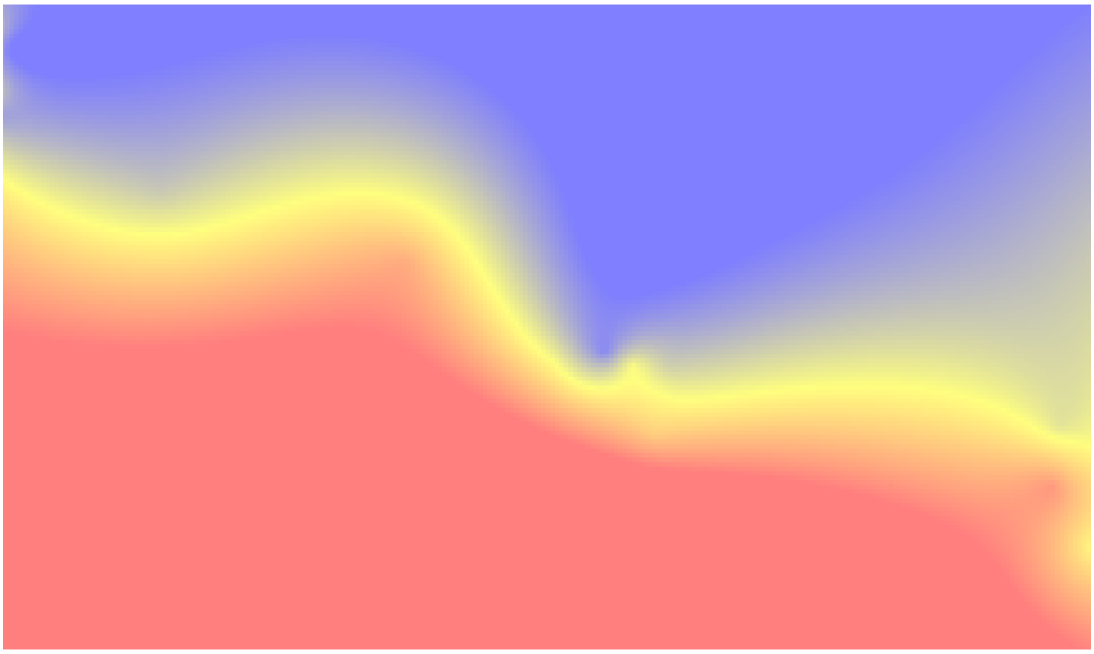


FIGURE H.2: Interpolated soil moisture map of November 20, 2015. The image is oriented to the north and the color represents saturation values ranging from 24.5 % to 31.7 % saturation.

Appendix I

Supplementary DVD

The author wrote a small commando line tool, that can interactively explain the folder structure. It can be started from the main folder from the command line by calling:

```
$> python explain.py
```

Bibliography

Aigner, D. (2008), 'Aktuelle forschungen im wasserbau 1993-2008', *Dresdner Wasserbauliche Mitteilung* 36, 162–176.

Ali, G. and Roy, A. G. (2009), 'Revisiting hydrologic sampling strategies for an accurate assessment of hydrologic connectivity in humid temperate systems', *Geography Compass* 3, 350–374.

Ambroise, B. (2004), 'Variable 'active' versus 'contributing' areas or periods: a necessary distinction', *Hydrological Processes* 18(6), 1149–1155.

Atmel Cooperation (2014*a*), '8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash', datasheet. Atmel Cooperation, San Jose, USA.

Atmel Cooperation (2014*b*), 'Atmel AVR134: Real Time Clock (RTC) Using the Asynchronous Timer'. Atmel Cooperation, San Jose, USA.

Bachmair, S., Weiler, M. and Troch, P. a. (2012), 'Intercomparing hillslope hydrological dynamics: Spatio-temporal variability and vegetation cover effects', *Water Resources Research* 48(5), 1–18.

Beasley, R. S., Granillo, A. B. and Zillmer, V. (1986), 'Sediment losses from forest management: Mechanical vs. chemical site preparation after clearcutting1', *Journal of Environment Quality* 15(4), 413.

Blume, T., Zehe, E. and Bronstert, A. (2007), 'Rainfall—runoff response, event-based runoff coefficients and hydrograph separation', *Hydrological Sciences Journal* 52(5), 843–862.

Boer, M. and Puigdefábregas, J. (2005), 'Effects of spatially structured vegetation patterns on hillslope erosion in a semiarid mediterranean environment: A simulation study', *Earth Surface Processes and Landforms* 30, 149–167.

Boix-Fayos, C., Calvo-Cases, A., Imeson, a. C., Soriano-Soto, M. D. and Tiemessen, I. R. (1998), 'Spatial and short-term temporal variations in runoff, soil aggregation and other soil properties along a mediterranean climatological gradient', *Catena* 33, 123–138.

Bracken, L. J. and Croke, J. (2007), 'The concept of hydrological connectivity and its contribution to understanding runoff-dominated geomorphic systems', *Hydrological Processes* 21, 1749–1763.

Bracken, L. J., Wainwright, J., Ali, G. a., Tetzlaff, D., Smith, M. W., Reaney, S. M. and Roy, A. G. (2013), 'Concepts of hydrological connectivity: Research approaches, pathways and future agendas', *Earth-Science Reviews* 119, 17–34.

Bronstert, A. and Bardossy, A. (1999), 'The role of spatial variability of soil moisture for modelling surface runoff generation at the small catchment scale', *Hydrology and Earth System Sciences* 3, 505–516.

Bushberg, J. T., Anthony, S. J., Leidholdt Jr., E. M. and Boone, J. M. (2006), *The Essential Physics of Mediacal Imaging*, 2nd edn, Lippincott Williams & Wilkins.

Buytaert, W., Vega, A., Villacis, M. and Moulds, S. (2015), 'Developing hydrological monitoring networks with arduino', *EGU General Assembly 2015* .

Calvo-Cases, a., Boix-Fayos, C. and Imeson, a. C. (2003), 'Runoff generation, sediment movement and soil water behaviour on calcareous (limestone) slopes of some mediterranean environments in southeast spain', *Geomorphology* 50, 269–291.

Castillo, V. M., Martinez-Mena, M. and Albaladejo, J. (1997), 'Runoff and soil loss response to vegetation removal in a semiarid environment', *Soil Science Society of America Journal* 61(4), 1116.

Curran, P. J. (1988), 'The semivariogram in remote sensing: An introduction', *Remote Sensing of Environment* 24(3), 493–507.

Dorigo, W. a., Wagner, W., Hohensinn, R., Hahn, S., Paulik, C., Xaver, A., Gruber, A., Drusch, M., Mecklenburg, S., Van Oevelen, P., Robock, A. and Jackson, T. (2011), 'The international soil moisture network: A data hosting facility for global in situ soil moisture measurements', *Hydrology and Earth System Sciences* 15(5), 1675–1698.

Dorigo, W., Xaver, A., Vreugdenhil, M., Gruber, A., Hegyiová, A., Sanchis-Dufau, A., Zamojski, D., Cordes, C., Wagner, W. and Drusch, M. (2013), 'Global automated quality control of in situ soil moisture data from the international soil moisture network', *Vadose Zone Journal* 12(3), 21.

Dunkerley, D. L. (1999), 'Banded chenopod shrublands of arid australia: Modelling responses to interannual rainfall variability cellular automata', *Ecological Modelling* 121, 127–138.

Durocher, M., Chebana, F. and Ouarda, T. B. (2016), 'On the prediction of extreme flood quantiles at ungauged locations with spatial copula', *Journal of Hydrology* 533, 523–532.

Eckhardt, K. (2008), 'A comparison of baseflow indices, which were calculated with seven different baseflow separation methods', *Journal of Hydrology* 352(1-2), 168–173.

El-Hassanin, a. (1983), Physical, chemical, and mineralogical characteristics of soil vs. erodibility., PhD thesis.

Farrick, K. K. and Branfireun, B. A. (2014), 'Soil water storage, rainfall and runoff relationships in a tropical dry forest catchment', *Water Resources Research* 50(12), 9236–9250.

Fitzjohn, C., Ternan, J. L. and Williams, A. G. (1998), 'Soil moisture variability in a semi-arid gully catchment: Implications for runoff and erosion control', *Catena* 32, 55–70.

Frerking, M. (1987), *Crystal Oscillator Desgin and Temperature Compensation*, 1st edn, Litton Educational Publishing, Inc.

Gosset aka. Student, W. S. (1908), 'The probable error of a mean', *Biometrika* 6(1), 1–25.

Grayson, R. B., Western, A. W., Chiew, F. H. S. and Blöschl, G. (1997), 'Preferred states in spatial soil moisture patterns: Local and nonlocal controls', *Water Resources Research* 33(12), 2897–2908.

Gringarten, E. and Deutsch, C. V. (2001), 'Teacher's aide variogram interpretation and modeling', *Mathematical Geology* 33(4), 507–534.

Hornbeck, J. W., Martin, C. W., Pierce, R. S., Bormann, F. H., Likens, G. E. and Eaton, J. S. (1986), 'Clearcutting northern hardwoods: effects on hydrologic and nutrient ion budgets.'.

Horton, R. E. (1933), 'The rôle of infiltration in the hydrologic cycle', *Transactions, American Geophysical Union* 14(1), 446.

Islam, A., Islam, T., Syrus, M. A. and Ahmed, N. (2014), Implementation of flash flood monitoring system based on wireless sensor network in bangladesh, *in* '3rd International Conference on Informatics, Electronics & Vision', IEEE Computer Society.

Jian, X., Olea, R. A. and Yu, Y.-S. (1996), 'Semivariogram modeling by weighted least squares', *Computers & Geosciences* 22(4), 387–397.

Jones, T. and Ma, Y. (2001), 'Teacher's aide: Geologic characteristics of hole-effect variograms calculated from lithology-indicator variables', *Mathematical Geology* 33(5), 615–629.

Kirkby, M. (1988), 'Hillslope runoff processes and models', *Journal of Hydrology* 100, 315–339.

Kruskal, W. H. and Wallis, W. A. (1952), 'Use of ranks in one-criterion variance analysis', *Journal of the American Statistical Association* 47(260), 583–621.

Lanni, C., Borga, M., Rigon, R. and Tarolli, P. (2012), 'Modelling shallow landslide susceptibility by means of a subsurface flow path connectivity index and estimates of soil depth spatial distribution', *Hydrology and Earth System Sciences* 16(11), 3959–3971.

Le Dinh, T., Hu, W., Sikka, P., Corke, P., Overs, L. and Brosnan, S. (2007), Design and deployment of a remote robust sensor network: Experiences from an outdoor water quality monitoring network, *in* 'Proceedings - Conference on Local Computer Networks, LCN', IEEE Computer Society, pp. 799–806.

Lee, S., Jo, J., Kim, Y. and Stephen, H. (2014), A framework for environmental monitoring with arduino-based sensors using restful web service, *in* '2014 IEEE International Conference on Services Computing', IEEE Computer Society, pp. 275–282.

Levene, H. (1960), Robust test for equality of variances, *in* I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow and H. B. Mann, eds, 'Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling', 1st edn, Stanford University Press, chapter 25, pp. 278–292.

Lexartza-Artza, I. and Wainwright, J. (2009), 'Hydrological connectivity: Linking concepts with practical implications', *Catena* 79(2), 146–152.

Ludwig, J. A., Wilcox, B. P., Breshears, D. D., Tongway, D. J. and Imeson, A. C. (2005), 'Vegetation patches and runoff–erosion as interacting ecohydrological processes in semiarid landscapes', *Ecology* 86(2), 288–297.

Ma, Y. Z. and Jones, T. a. (2001), 'Teacher's aide: Modeling hole-effect variograms of lithology-indicator variables', *Mathematical Geology* 33(5), 631–648.

Mälicke, M. (2013), Entwicklung und Evaluation einer open Source Messstation, Bachelor of science, Chair of Hydrology, Institute of Earth and Environmental Science, Albert-Ludwigs-University, Freiburg.

Mälicke, M. (2016), 'DataLogger Micro datasheet', Figshare, 2 edn, p. 4. DOI: 10.6084/m9.figshare.1297291, URL: http://figshare.com/articles/DataLogger_Micro_datasheet/1297291.

Mann, H. B. and Whitney, D. R. (1947), 'On a test of whether one of two random variables is stochastically larger than the other', *The Annals of Mathematical Statistics* 18(1), 50–60.

Mann, L. K., Johnson, D. W., West, D. C., Cole, D. W., Hornbeck, J. W., Martin, C. W., Riekerk, H., Smith, C. T., Swank, W. T., Tritton, L. M. and Van Lear, D. H. (1988), 'Effects of whole-tree and stem-only clearcutting on postharvest hydrologic losses, nutrient capital, and regrowth', *Forest Science* 34(2), 412–428.

Maxim Integrated (2015), 'Extremely Accurate I 2 C-Integrated RTC / TCXO / Crystal Extremely Accurate I 2 C-Integrated', datasheet. Maxim Integrated Products, Inc., San Jose, USA.

McNamara, J. P., Chandler, D., Seyfried, M. and Achet, S. (2005), 'Soil moisture states, lateral flow, and streamflow generation in a semi-arid, snowmelt-driven catchment', *Hydrological Processes* 19, 4023–4038.

Merchant, C., Le Borgne, P., Marsouin, A. and Roquet, H. (2008), 'Optimal estimation of sea surface temperature from split-window observations', *Remote Sensing of Environment* 112(5), 2469–2484.

Meyles, E., Williams, A., Ternan, J. L. and Dowd, J. (2003), 'Runoff generation in relation to soil moisture patterns in a small dartmoor catchment, southwest england', *Hydrological Processes* 17(January 2001), 251–264.

Nasirudin, M. a., Za'bah, U. N. and Sidek, O. (2011), Fresh water real-time monitoring system based on wireless sensor network and gsm, *in* 'Open Systems (ICOS), 2011 IEEE Conference on', IEEE Computer Society, pp. 354–357.

Nicolau, J. M., Solé-Benet, A., Puigdefábregas, J. and Gutiérrez, L. (1996), 'Effects of soil and vegetation on runoff along a catena in semi-arid spain', *Geomorphology* 14, 297–309.

Penna, D., Tromp-Van Meerveld, H. J., Gobbi, A., Borga, M. and Dalla Fontana, G. (2011), 'The influence of soil moisture on threshold runoff generation processes in an alpine headwater catchment', *Hydrology and Earth System Sciences* 15(2002), 689–702.

Perez, J. F., Swain, N. R., Dolder, H. G., Christensen, S. D., Snow, A. D., Nelson, E. J. and Jones, N. L. (2016), 'From global to local: Providing actionable flood forecast information in a cloud-based computing environment', *JAWRA Journal of the American Water Resources Association* pp. n/a–n/a.

Puigdefábregas, J. (2005), 'The role of vegetation patterns in structuring runoff and sediment fluxes in drylands', *Earth Surface Processes and Landforms* 30, 133–147.

Pyrcz, M. J. and Deutsch, C. V. (2003), 'The whole story on the hole effect', *Geostatistical Association of Australasia Newsletter 18* 18, 18.

Rao, a. S., Izadi, D., Tellis, R. F., Ekanayake, S. W. and Pathirana, P. N. (2009), Data monitoring sensor network for big net research testbed, *in* 'Proceedings of the 2009 Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing', IEEE Computer Society, pp. 169–173.

Rao, A. S., Marshall, S., Gubbi, J., Palaniswami, M., Sinnott, R. and Pettigrovet, V. (2013), Design of low-cost autonomous water quality monitoring system, *in* 'Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013', IEEE Computer Society, pp. 14–19.

Rein Elektronik (2015), 'SCC Quarze und Oszillatoren', datasheet. Rein Elektronik GmbH, Nettetal, Germany.

Rosenbaum, U., Bogena, H. R., Herbst, M., Huisman, J. A., Peterson, T. J., Weuthen, A., Western, A. W. and Vereecken, H. (2012), 'Seasonal and event dynamics of spatial soil moisture patterns at the small catchment scale', *Water Resources Research* 48.

Saxton, K. E. and Rawls, W. J. (2006), 'Soil water characteristic estimates by texture and organic matter for hydrologic solutions', *Soil Science Society of America Journal* 70(5), 1569.

Seenath, A., Wilson, M. and Miller, K. (2016), 'Hydrodynamic versus gis modelling for coastal flood vulnerability assessment: Which is better for guiding coastal management?', *Ocean & Coastal Management* 120, 99–109.

Shapiro, S. S. and Wilk, M. (1965), 'An analysis of variance test for normality (complete samples)', *Biometrika* 52(3/4), 591–611.

Sikka, P., Corke, P., Overs, L., Valencia, P. and Wark, T. (2007), Fleck - a platform for real-world outdoor sensor networks, *in* 'Proceedings of the 2007 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP', IEEE Computer Society, pp. 709–714.

Silicon Storage Technology (2006), '32 Mbit SPI Serial Flash', datasheet. Silicon Storage Technology Inc., Sunnyvale, USA.

Sinnott, R. (1984), 'Virtues of the haversine', *Sky and Telescope* 68(2).

Sloto, R. a. and Crouse, M. Y. (1996), 'Hysep: a computer program for streamflow hydrograph separation and analysis', *U.S. Geological Survey Water-Resources Investigations Report* 96(4040), 54.

Snedecor, G. W. and Cochran, W. G. (1989), *Statistical Methods*, 8th edn, Iowa State University Press.

Tetzlaff, D., Soulsby, C., Bacon, P. J., Youngson, A. F., Gibbins, C. and Malcolm, I. A. (2007), 'Connectivity between landscapes and riverscapes—a unifying theme in integrating hydrology and ecology in catchment science?', *Hydrological Processes* 21(10), 1385–1389.

Usha Rani, M. and Kamalesh, S. (2014), Web based service to monitor automatic irrigation system for the agriculture field using sensors, *in* '2014 International Conference on Advances in Electrical Engineering, ICAEE 2014', IEEE Computer Society.

Valentin, C., Agus, F., Alamban, R., Boosaner, A., Bricquet, J., Chaplot, V., de Guzman, T., de Rouw, A., Janeau, J., Orange, D., Phachomphonh, K., Podwojewski, P., Ribolzi, O., Silvera, N., Subagyono, K., Thiébaux, J. and Vadari, T. (2008), 'Runoff and sediment losses from 27 upland catchments in southeast asia: Impact of rapid land use changes and conservation practices', *Agriculture, Ecosystems & Environment* 128(4), 225–238.

Valentin, C., D'Herbès, J. M. and Poesen, J. (1999), 'Soil and water components of banded vegetation patterns', *Catena* 37, 1–24.

Van Genuchten, M. T. (1980), 'A closed-form equation for predicting the hydraulic conductivity of unsaturated soils', *Soil science society of America journal* 44(5), 892–898.

Wang, J. W. J., Ren, X.-l. R. X.-l., Shen, Y.-l. S. Y.-l. and Liu, S.-y. L. S.-y. (2010), A remote wireless sensor networks for water quality monitoring, *in* 'Innovative Computing & Communication, 2010 Intl Conf on and Information Technology Ocean Engineering (CICC-ITOE)', IEEE Computer Society, pp. 7–12.

Webster, R. and Oliver, M. A. (1990), *Statistical methods in soil and land resource survey.*, Oxford University Press (OUP).

Western, A. W., Blöschl, G. and Grayson, R. B. (1998), 'Geostatistical characterisation of soil moisture patterns in the tarrawarra catchment', *Journal of Hydrology* 205, 20–37.

Wilcoxon, F. (1946), 'Individual comparisons of grouped data by ranking methods.', *Journal of economic entomology* 39(6), 269.

Wilcoxon, F., Katti, S. K. and Wilcox, R. A. (1963), *Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test*, American Cyanamid Company.

Yates, F. (1948), 'Systematic sampling', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 241(834), 345–377.

Zennaro, M., Floros, A., Dogan, G., Sun, T. S. T., Cao, Z. C. Z., Huang, C. H. C., Bahader, M., Ntareme, H. and Bagula, A. (2009), On the design of a water quality wireless sensor network (wqwsn): An application to water quality monitoring in malawi, *in* '2009 International Conference on Parallel Processing Workshops', IEEE Computer Society.

Zillgens, B., Merz, B., Kirnbauer, R. and Tilch, N. (2005), 'Analysis of the runoff response of an alpine catchment at different scales', *Hydrology and Earth System Sciences Discussions* 2(5), 1923–1960.

Zin, T. T., Lin, J. C.-W., Pan, J.-S., Tin, P. and Yokota, M., eds (2016), *Genetic and Evolutionary Computing*, Vol. 387 of *Advances in Intelligent Systems and Computing*, Springer International Publishing.