

Freie Universität Berlin

Bachelor Thesis



Department of Mathematics and Computer Science

Examination regulations from 2010

Stably Stratified Atmospheric Boundary Layers: Event Detection and Classification for Turbulent Time Series

Supervisor:
Vercauteren and
Klein

Author:
Amandine Kaiser
4652563

February 11, 2016

Contents

1	Introduction	2
2	Turbulence in the Atmospheric Boundary Layer	3
2.1	Turbulence	3
2.2	The Origin of Turbulence in the Atmospheric Boundary Layer	4
2.3	A simple statistical model for stationary turbulence	6
3	Description of the Data	9
4	Description of the Method	10
4.1	Test by Philips-Perron	11
4.2	Test by Zivot and Andrews	13
4.3	Red noise test	13
4.4	White noise test	13
5	Application of the method	15
6	Conclusions	20
7	Acknowledgement	21
8	References	21
	Appendix	23
A	Appendix	23
B	Appendix	26
C	Appendix	43
D	Appendix	46
E	Appendix	55

1 Introduction

The atmospheric boundary layer is the lowest part of the atmosphere where life takes place. For understanding weather patterns affecting human lives a good understanding of the dynamics in this layer is required. One important process in the atmospheric boundary layer is turbulence. Decisions that affect human life must be made daily based on predictions of turbulent flows. Hence it is fundamental to understand the physical processes behind it. But so far there are still many processes which are not fully understood and cannot be statistically modelled. This is especially true for the stably stratified atmospheric boundary layer which typically occurs during the night or above cooler surfaces like glaciers. Apart from turbulence it includes small scale non turbulent motions of complex origins that are poorly understood by the scientific community. As stated in the paper by Vercauteren and Klein (2015), the presence of such motions could affect turbulent mixing to a large extent if the thermal stratification is very high. Usual approaches which aim to recognize events in the atmospheric boundary layer assume certain physical processes and then search for a trace of these in the atmospheric time series. This can be accomplished by searching for certain geometries or large amplitudes. However, many events in atmospheric series result from yet unidentified physical processes. Consequently a new approach is necessary. A statistical method was recently developed by Kang et al. (2015a) to detect events in noisy time series. This method does not assume any underlying physical processes to extract events from the time series. Nevertheless, physical mechanisms can be investigated a posteriori by analyzing the extracted events. In this thesis we will analyse parts of the Snow-Horizontal Array Turbulence Study (SnoHATS) dataset (Bou-Zeid et al. (2010)) which was collected over a glacier by using a slightly modified version of this event detection method. In their analysis, Kang et al. (2015) investigated events of a certain scale by defining a maximal duration of events, and filtering out small-scale variability. In this thesis, we will investigate different scales of motion by applying the method for multiple timescales. We will thereby test the sensitivity of the method to technical criteria related to timescales of variability. In this thesis we will focus on scales from 1 to 30 minutes. The rationale behind this choice of time window is that motions on scales between 1 and 30 minutes in stable conditions are typically dominated by wavelike motions, microfronts and other complex structures of unknown origin (Mahrt, (2011)). We will refer to such motions as submesomotions in the rest of the thesis.

In chapter 2 the concept of turbulence in the atmospheric boundary layer is

explained in more detail by first stating how we define turbulence and then illustrating the origin of turbulence in the atmospheric boundary layer. Afterwards, in chapter 3, the dataset is described and in chapter 4 the steps in the event detection method by Kang et al. (2015a) are explained. The multiscale approach and the results from the event detection procedure are described in chapter 5.

2 Turbulence in the Atmospheric Boundary Layer

2.1 Turbulence

Without turbulence in the atmospheric boundary layer, life would be extremely different from how we know it. The transport of heat and pollutants in the atmosphere and ocean would be much slower. Through its mixing properties, turbulence has an important impact on large-scale atmospheric flows, thereby affecting the weather and climate. Hence representing mixing properties of turbulence accurately is crucial to the performance of weather predictions or climate simulations. Conceptually, the airflow in the atmospheric boundary layer consists of three components: the mean wind, waves and turbulence. Turbulence occurs because of the shear (vertical difference in wind force and/or direction) in the mean wind which triggers instabilities that eventually result in turbulence. The temperature stratification can enhance or suppress turbulence through modifications of the buoyancy of the flow. An unstable density stratification resulting from the rise of warm air will enhance turbulence, whereas a stable density stratification will tend to damp it. Due to turbulence there are complications in modelling and measuring the atmospheric boundary layer.

There is no universally applicable definition of turbulence. In this thesis it will be described with the experiment by Reynolds. In the experiment flows of fluids were analysed. To make the characteristics of flows obvious, colour was added to the fluid. There were two different types of flows. When the fluid was moved with a slow velocity there was a smooth colour thread. This is called a laminar flow. However if the velocity was higher, the colour thread oscillated and at some point the whole fluid was coloured. The experiment illustrates the mixing properties of turbulent flows. Resulting from this experiment, six different characteristics for turbulence can be stated. It is three-dimensional, irregular and random (movements in the fluid are

chaotic), vortical, non linear; it transports and mixes, and dissipates energy. Moreover, Reynolds defined a non-dimensional number which includes characteristic length scale of the flow, velocity scale and viscosity. This number is defined as (Garrat (1994)):

$$Re = \frac{v_c l_c}{\nu}$$

where v_c is a characteristic velocity scale of the flow, l_c is a characteristic length of the flow and ν is the kinematic viscosity. Transition to turbulence starts when this number is above 10^2 . The atmosphere has typical values of 10^8 which makes the flow in general fully turbulent. Velocity and length scales decrease at night making turbulence less well developed and possibly non-stationary.

Contrary to laminar flows, there are several different types of vortices in turbulent flows. They differ in size and intensity. Their scale ranges from a fraction of a millimeter or centimeter to a few hundreds or thousands of meters.

2.2 The Origin of Turbulence in the Atmospheric Boundary Layer

The atmospheric boundary layer (ABL) is the part of the atmosphere which is directly affected by the planetary surface and is of high importance for us as we are spending most of our lives in it.

States and processes of the ABL mainly result from the physical characteristics of the surface. The surface influences the ABL by friction and by heat fluxes on the ground. The thickness of the ABL varies above the sea and mountains. On average it is 1000 m, but in mid-latitudes it can vary from 100 meters to 3 kilometers.

During day time, the surface is heated up by solar radiation. As a result, the surface radiates heat into the atmosphere. Different types of gases in the atmosphere absorb most of this radiation. There are two different mechanisms which result in mixing of the ABL. First of all warm air rises. Moreover the cooling of the top of clouds leads to a local drop of individual air packages. This is one of the processes which is described as turbulence. It can also be created by vertical wind shear. During a clear day, the boundary layer can be divided into several sublayers as shown in Figure 1. The first one is the roughness layer. In this layer air flows around roughness elements (eg grass, plants or buildings). The second layer is the surface layer which is also known as the constant flux layer.

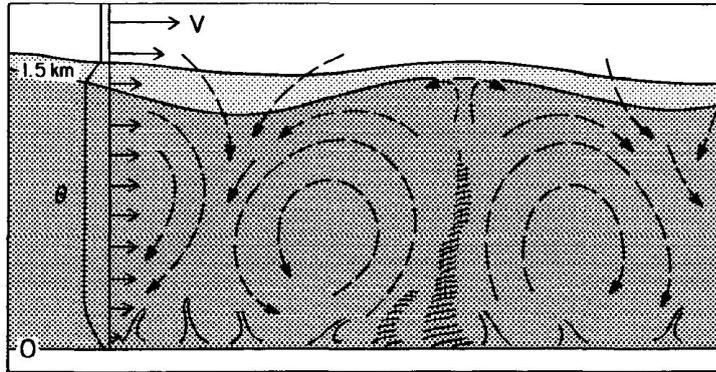


Figure 1: Graphic of the atmospheric boundary layer during the day (Wyngaard (1990))

The thickness of this layer is usually 100 m. Wind, temperature and humidity vary with altitude, and characteristics of turbulence are affected by the surface. The third layer is the well-mixed layer and the last one is the capping inversion. This layer confines air and pollution below it and within the boundary layer.

In this thesis there will be a focus on the stably stratified atmospheric boundary layer which is present during the night or above colder surfaces e.g. glaciers. Figure 2 is a sketch of the stable boundary layer during the night.

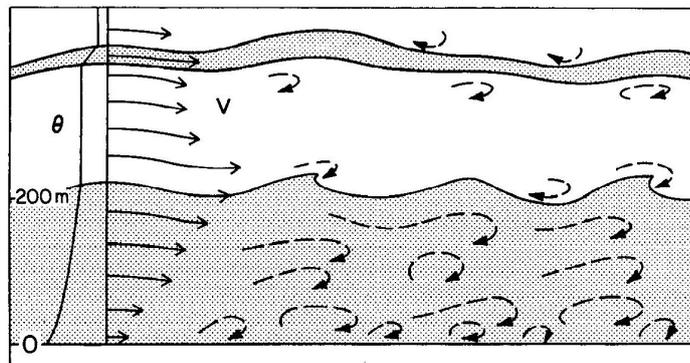


Figure 2: Graphic of the nocturnal stable stratified boundary layer with fluxes, waves, inversion layer and wind shear (Wyngaard (1990))

Because of the absence of incoming solar radiation and the presence of long-wave radiative cooling, the surface cools down during the night. The impulse for convection is therefore non-existent. Slightly above the surface, air cools down drastically. The presence of wind shear can however trigger fluxes

which transport the cold air higher. As a result, the daytime mixed-layer remains as a residual layer. Below the residual layer and over the surface layer a new layer is formed. This new layer is called the stable boundary layer. In this layer, turbulent structures occur as well. They can be formed through wind shear, small scale vortices and occasionally wave movements, but so far physical dynamics in the stably stratified atmospheric boundary layer are not fully understood.

2.3 A simple statistical model for stationary turbulence

This thesis will focus on detecting events that are not stationary turbulence, thus we introduce a simplified model for stationary turbulence that will be used later.

A stationary time series is defined as follows (Shumway et al. (2011)).

Definition 1: A strictly stationary time series is one for which the probabilistic behavior of every collection of values

$$\{x(t_1), x(t_2), \dots, x(t_k)\}$$

is identical to that of the shifted set

$$\{x(t_1 + h), x(t_2 + h), \dots, x(t_k + h)\}.$$

That is,

$$\begin{aligned} &P\{x(t_1) \leq c_1, x(t_2) \leq c_2, \dots, x(t_k) \leq c_k\} \\ &= P\{x(t_1 + h) \leq c_1, x(t_2 + h) \leq c_2, \dots, x(t_k + h) \leq c_k\}. \end{aligned}$$

for all $k = 1, 2, \dots$, all time points t_1, t_2, \dots, t_k , all numbers c_1, c_2, \dots, c_k and all shifts $h = 0, \pm 1, \pm 2, \dots$

If a time series is strictly stationary, then all of the multivariate distribution functions for subsets of variables must agree with their counterparts in the shifted set for all values of the shift parameter h .

According to Faranda et al. (2014) we can model Gaussian homogenous atmospheric turbulence as a first order autoregressive process (AR(1)). Autoregressive models are based on the idea that the current value of the series, $x(t)$, can be explained as a function of p past values, $x(t-1), x(t-2), \dots, x(t-p)$,

where p determines the number of steps into the past needed to forecast the current value. We formally define an autoregressive model of order p as follows (Shumway et al. (2011)).

Definition 2: An autoregressive model of order p , abbreviated AR(p), is of the form

$$x(t) = \phi_1 x(t-1) + \phi_2 x(t-2) + \dots + \phi_p x(t-p) + \epsilon(t),$$

where $x(t)$ is stationary, and $\phi_1, \phi_2, \dots, \phi_p$ are constants ($\phi_p \neq 0$).

Faranda et al. (2014) state that the evolution of the velocity and the position of a tracer particle (u, x) can be described by the following equations:

$$\begin{aligned} du &= a(x, u, t)dt + b(x, u, t)dW \\ dx &= udt \end{aligned} \quad (1)$$

with dW being the increments of a Wiener process. A Wiener Process refers to the mathematical model used to describe Brownian Motion (random motion of particles suspended in a fluid resulting from their collision with the quick atoms or molecules in the fluid). Though the two terms are sometimes used interchangeably. We define a Wiener process as follows (Shumway et al. (2011)).

Definition 3: A continuous time process $\{W(t); t \geq 0\}$ is called a Wiener process if it satisfies the following conditions:

- i*) $W(0) = 0$
- ii*) $\{W(t_2) - W(t_1), W(t_3) - W(t_2), \dots, W(t_n) - W(t_{n-1})\}$ are independent for any collection of points, $0 \leq t_1 < t_2 \dots < t_n$, and integer $n > 2$
- iii*) $W(t + \Delta t) - W(t) \sim N(0, \Delta t)$ for $\Delta t > 0$

The mixing properties of turbulence make neighbour particles loose correlation after a certain time, that can be described by a Lagrangian decorrelation scale T_L . Furthermore the rate on mean kinetic energy dissipation will affect the Eulerian diffusion and hence the variance of the Wiener process in equation (1).

Using these two parameters for gaussian homogeneous turbulence, it can be shown that the Langevin equation (1) takes the discrete form (Faranda et al. (2014)):

$$\Delta u = -\frac{u}{T_L}\Delta t + \sqrt{C_0\epsilon}\Delta W$$

with T_L the Lagrangian decorrelation scale, C_0 a constant and ϵ the mean kinetic dissipation rate. By rearranging this equation and denoting $\phi = (1 - \frac{\Delta t}{T_L})$ and $\sigma = \sqrt{C_0\epsilon\Delta t}$ we get

$$u(t) = \phi u(t-1) + \epsilon(t)$$

Hence, u_t is an AR(1) process and we will use an AR(1) model for stationary turbulence. Later in this thesis we will need red noise for the event detection method.

Definition 4: A red noise process is defined as:

$$y(t) = \phi y(t-1) + \epsilon(t)$$

for ϕ first order autocorrelation coefficient with $0 < \phi < 1$ and $\epsilon(t)$ white noise with standard deviation σ_ϵ .

Red noise is the same as a first-order autoregressive (AR(1)) stationary Gaussian process with a positive correlation at unit lag (Storch and Zwiers (1995)). Suppose we have an autoregressive process of order p

$$x(t) = \phi_1 x(t-1) + \phi_2 x(t-2) + \dots + \phi_p x(t-p) + \epsilon(t).$$

If $m = 1$ is a root of the characteristic equation

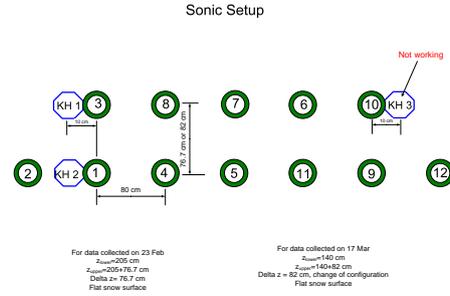
$$m^p - \phi_1 m^{p-1} - \dots - \phi_p = 0$$

then the $x(t)$ has a unit root. An AR(1) process has a unit root if $\phi = 1$. Moreover if an AR(1) process has a unit root it is non-stationary.

3 Description of the Data



(a) SnoHATS: side view of the 12 sonics array



(b) Sonic Set up

Figure 3: Set up of the sonics which collected the data for this analysis (Bou-Zeid et al. (2010))

The data was collected over the plaine Morte Glacier in the Swiss Alps (46.3863°N, 7.5178°E, 2750m elevation) from the 2. February to 19. April 2006. The measurement campaign included a structure supporting 12 sonic anemometers that measured 3D wind velocity components, temperature and humidity at 20 Hz. This analysis is based on the measurements of 4 sonic anemometers (5,6,7 and 8) out of 12 and we will focus on the wind velocity and temperature dynamics. Figure 3 shows the set up of the sonics.

The sonics were set up in two arrays with a distance of 76.7 cm on the 23 February and 82 cm afterwards. Their height varied from 1.40 m to 2.80 m depending on the accumulation of snow.

The analysis was restricted to wind directions of $\pm 60^\circ$ relative to the stream-wise sonic axis to ensure a fetch of 1500 m of flat snow. The analyses will be based on four timeseries extracted from four different data clusters that were isolated in a study by Vercauteren and Klein (2015). The turbulent flows in the clusters were shown to have different physical properties, but small-scale non-turbulent motions were not investigated in details. The turbulent event detection method described in section 4 will be applied to the four timeseries to investigate differences in the occurrences of submesomotions in the four clusters. The time series in cluster 1 is 9,6 hours long and starts on 18th of March 2006 at 5:40 pm. The time series in cluster 2 and 3 have the same length but the one in cluster 4 is 6 hours long. The second time series starts 3:50 pm (17.03.06), the third at 6:50pm (04.02.06) and the fourth at 5:40 pm (29.03.06).

4 Description of the Method

The event detection is based on a method developed by Kang et al. (2014). In this chapter the event detection process is theoretically described and in chapter 5 it is applied to the SnoHATS dataset.

The goal of this procedure is to separate events from noise in a time series. The main assumptions are that background noise is always present in the time series and that events are separated by noise. This assumption is verified in Chapter 5 for the four analyzed time series.

Moreover, it is assumed that an event can be defined as a non-AR(1) process which means that events are non-stationary, oscillatory and/or nonlinear motions. To justify the assumption that events can be defined as non-AR(1) processes Kang et al. (2015a) performed two tests. In the first test, they introduce a nonlinear component into the linear AR(1) model and examine the behaviour of the event extraction method. The changes of event number after phase randomization are investigated in the second test. The results from both tests verify the assumption that events as they are detected from this method are not AR(1) processes.

By using a sliding window with predefined length l , subsequences are obtained. The event detection method is applied to each subsequence. The q th subsequence can be expressed as:

Definition 5: If $x(t)$ is a time series the q^{th} subsequence is defined as:

$$x_q(t) = \{x(t_q), \dots, x(t_{q+l-1})\}$$

for $1 \leq q \leq m - l + 1$ and $m = \text{length}(x(t))$.

Note that these subsequences overlap. Hence the event detection is performed on overlapping sequences rather than on separated blocks. To separate events from noise, three steps are performed on these subsequences. Figure 4 shows the order in which the tests are applied to each subsequence.

First the Philip Perron (PP) test is applied to the subsequence and it checks if the subsequence is stationary. If it is stationary, according to the Philip Perron test, the Zivot and Andrews (ZA) test is performed. Otherwise the noise test is performed after the Philip Perron test. These tests are described in more detail in the following sections. Events are defined as those subsequences which are significantly different from pure noise. A p value is computed for each subsequence, according to one of the three above mentioned noise tests.

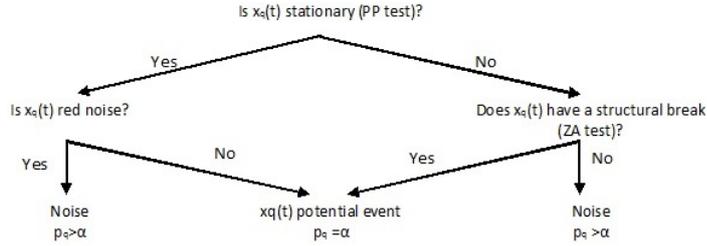


Figure 4: Flowchart of the event detection procedure for a subsequence $x_q(t)$ (Kang et al. (2015a)), In this analysis $\alpha = 0.5$ and p_q p-value.

The p value is the probability to obtain a result equal to or more extrem than the actual observation. If the p value, from the noise test or the ZA test, is smaller than a predefined significance level α , it shows that the subsequence is not only composed of pure noise and hence it is a potential event. A formal definition for an event will be given later in this chapter.

4.1 Test by Philips-Perron

The Philip Perron test is a nonstationarity hypothesis test. The red noise test is only applicable to stationary time series. Therefore, this test is necessary before applying the noise test. In a hypothesis test the null hypothesis H_0 , which usually refers to a general statement that there is no relationship between two measured phenomena, is tested against an alternative hypothesis H_1 . If the null hypothesis is rejected, it can be concluded that there is a relationship between the two phenomena. The null hypothesis H_0 for the Philip Perron test is: $x_q(t)$ is a unit root process and hence not stationary. If H_0 is rejected, the noise test can be performed. If $x_q(t)$ is nonstationary according to the test, another test has to be performed because the PP test does not reject H_0 for simple random walk processes ($\phi = 1$). For processes which are stationary but have a structural break, the null hypothesis is not rejected either. A structural break can appear when we see an unexpected shift in the time series. The following Lemma shows that simple random walk processes are not stationary and hence are not rejected by the Philip Perron test.

Lemma 1:

A simple random walk process

$$x(t) = \sum_{j=1}^t z(j) = x(t-1) + z(t),$$

with $z(t) \sim IID(\mu, \sigma^2)$ and $x(0) = 0$, is not stationary.

Proof: Let $x(t) = \sum_{j=1}^t z(j)$ be a simple random walk with independent random variables $z(j)$. Note that $\text{var}(x(t)) = \sum_{j=1}^t \text{var}(z(j)) = t\sigma^2$. Define $y = \sum_{j=t+1}^{t+h} z(j)$. $x(t)$ and y are independent random variables because they are sums of disjoint independent variables. Calculate the covariance of $x(t+h)$ and $x(t)$.

$$\begin{aligned} \text{cov}(x(t+h), x(t)) &= \text{cov}(x(t) + y, x(t)) \\ &= \text{cov}(x(t), x(t)) + \text{cov}(y, x(t)) \\ &= \text{var}(x(t)) + 0 \\ &= t\sigma^2 \end{aligned}$$

The covariance depends on t and not on h . Hence $x(t)$ is not covariance stationary. Which implies that a simple random walk process is not stationary. \square

A simple random walk process is not considered to be a potential event. We want to include stationary processes with a structural break because they are a potential event. A potential event is defined as follows.

Definition 6:

Let p_s, p_{s+1}, \dots, p_l be a sequence of p-values. If

- i $p_i \leq \alpha$ for $i = s, s+1, \dots, l$
- ii $(e - s) \geq \frac{l}{2}$

the event is defined as a segment from the time point $s + \frac{l}{4}$ to $(e + l - 1) - \frac{l}{4}$ where s is the starting point of the s^{th} potential event and $e + l - 1$ the ending point of the e^{th} .

$\frac{l}{4}$ is discarded to remove excess background noise at the start and end of the event (Kang et al. (2015a)). From this definition it follows $l_e \geq l$. In this analysis l is predefined and even though the event length l_e is flexible, structures with too large time scales may only be partially detected or not at all.

4.2 Test by Zivot and Andrews

The Zivot and Andrews procedure is another stationarity test. It tests whether the time series has a unit root against the hypothesis that it is stationary with a structural break. Recall from section 2.3 that if an AR(1) process has a unit root it is non-stationary. If the null hypothesis from the ZA test is rejected, $x_q(t)$ is defined as a potential event, or otherwise as noise. This test is performed to ensure that stationary processes with a structural break are included.

4.3 Red noise test

Before starting the noise test, the type of noise has to be chosen. Recall from section 2.3 that stationary turbulence and red noise can be well represented by an AR(1) process. Hence, we will use red noise.

It is crucial to detrend the subsequences before applying the red noise test, because the red noise test recognizes a process with a linear trend as a potential event. The subsequences are detrended if the goodness-of-fit, the discrepancy between observed values and the expected values, of the linear model fitted on the subsequence is larger than 0.85.

To perform the red noise test the AR(1) model $\tilde{x}_q(t) = \phi\tilde{x}_q(t-1)$ is fitted to $x_q(t)$ and the residuals $\epsilon(t) = x_q(t) - \tilde{x}_q(t)$ are calculated. Afterwards the white noise test is performed on the residuals. If the white noise test is positive, $x_q(t)$ is claimed to be red noise.

4.4 White noise test

To test for white noise, the Ljung-Box test is applied. It tests whether the data points are independently distributed, which is a characteristic of white noise in time series. The null hypothesis H_0 : data points are independently distributed, is tested against an alternative hypothesis H_1 : data points are not independently distributed. The following Theorem shows that, under

H_0 , the test statistic Q follows a chi squared distribution with h degrees of freedom.

Theorem 1: Under H_0 the test statistic

$$Q = n(n+2) \sum_{k=1}^h \frac{\hat{p}_k^2}{n-k}$$

for n sample size, \hat{p}_k sample autocorrelation at lag k and h number of lags being tested follows a $\chi^2(h)$ chi-squared distribution with h degrees of freedom.

Proof: Let Q be the test statistic and rearrange the formula for Q .

$$\begin{aligned} Q &= n(n+2) \sum_{k=1}^h \frac{\hat{p}_k^2}{n-k} \\ &= n \sum_{k=1}^h \frac{(n+2)\hat{p}_k^2}{n-k} \end{aligned}$$

Note that $\frac{n+2}{n-k} \rightarrow 1$ as $n \rightarrow \infty$ and define $\tilde{Q} = n \sum_{k=1}^h \hat{p}_k^2$. Q behaves asymptotically like \tilde{Q} because $\lim_{n \rightarrow \infty} \frac{Q}{\tilde{Q}} = 1$.

To show that \tilde{Q} is $\chi^2(h)$ under H_0 , consider the following result. Let $\hat{p} = (\hat{p}_1, \dots, \hat{p}_h)^T$. For a white noise process (this satisfies H_0)

$$y_t = \mu + \epsilon_t \text{ with } E[\epsilon_t]^4 < \infty$$

it holds that $\sqrt{n}\hat{p} \rightarrow_d N(0, I_h)$. Consequently the first h sample autocorrelations are multivariate normal with expected value 0 each and asymptotic covariance matrix equal to the identity matrix. They are asymptotically independent. This implies that each autocorrelation is asymptotically standard normal.

Moreover note that:

$$\tilde{Q} = \sqrt{n}\hat{p}^T \sqrt{n}\hat{p}$$

Due to the fact that the sum of h independent squared standard normal random variables is $\chi^2(h)$ it can be concluded that Q follows a $\chi^2(h)$ chi-squared distribution with h degrees of freedom. □

h is chosen to be approximately $\ln(n)$ as suggested in the paper from Kang et al. (2015a), which refers to Tsay (2005). The probability p of obtaining

a test statistic which is at least as extreme as the actually observed statistic under $\chi^2(h)$ is used. If the p value is less than a predefined significance level α , H_0 is rejected. This indicates that the data is not white noise. If the white noise test was performed on the residuals of $x_q(t)$, it indicates that the tested subsequence is not red noise and hence a potential event.

Kang et al. (2014) have also developed a feature based clustering method for the detected events. The detected events are clustered in groups with similar characteristics. First, each event is summarized by statistical measures of its characteristics and then clustered according to them. This clustering leads to significantly better results compared to clustering the raw data (Kang et al (2014)). This method is not described in detail as it is not relevant in this thesis. It is not relevant because the analysed dataset is relatively small and only a maximum of 12 events were detected.

5 Application of the method

In the event detection method from Kang et al. (201a) the length of the time window l has to be predefined. According to Kang et al. (2015a), better results are obtained by keeping l constant and block averaging the time series to a desired scale. This results from the fact that the test statistic Q from the white noise test depends on l and keeping l constant returns consistent results for all scales. The choice of the time window length is so far made subjectively and is based on experience and context. This could be solved by using a different method which determines the relevant time scales, before applying the event detection procedure. Another option which is used in this thesis is to use a multiscale approach. A multiscale analysis allows a verifiable choice of the scale, if it cannot be determined in a different way. In this chapter we will analyse the SnoHATS dataset with different scales and afterwards we will focus on the scale which is most appropriate. How we choose this most appropriate scale will be explained later. All functions which are needed for this analysis are stated and explained in Appendix E. Moreover the TED package (Kang,Y., et al. (2015)) is required.

The significance level α , which is needed in the noise tests, is chosen to be 0.5 and the time window length l 120 points for all scales. The SnoHATS dataset consists of 20 Hz temperature and wind component data. We will focus on the temperature data because the event detection method was designed for this type of data. However, we will also compare it with wind direction and wind speed events. As stated in the introduction we want to analyse

submesomotions which we defined as motions from 1 to 30 minutes. Hence, we use 1 s to 15 s averaged data. The 7 s, 11 s, 13 s and 14 s averaged data are not used because the lengths of the time series (691200 data points for cluster 1,2 and 3, and 432000 for cluster 4) are no multiple of these averages. By using the other averages we end up with event lengths, corresponding to the length of the data subsequence, from 2 mins to 30 mins. To average our data we use the arithmetic mean.

$$\bar{x} = \frac{1}{n} \sum_{k=1}^n x_k$$

with x_k for $k = 1 \dots n$ being the data points and n the length of the average, for example 3 s=120 points. Note that if events overlap in one averaged time series we merge them into one event. In Appendix A the multiscale approach is visualized for all clusters. Each line represents the time period recognised as an event. Figure 5 gives an example of the results from the event detection method.

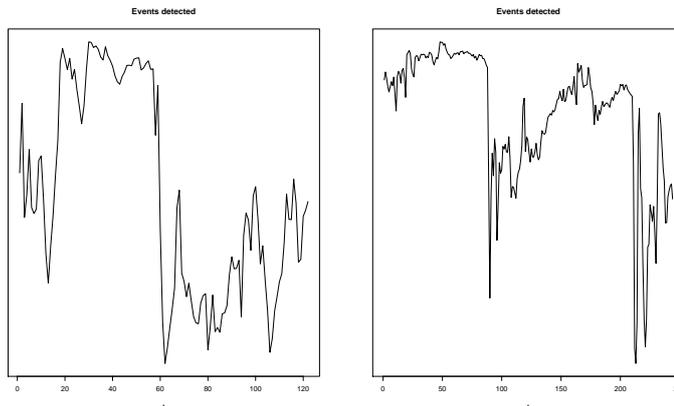


Figure 5: Examples of the results from the event detection method applied to the 6 s averaged temperature time series from cluster 2 sonic 6

For small averages there are in general many short events while for higher averages there are few long events. This is expected because we use the same window length for all scales which impacts the duration in time of the subsequence and hence the maximal event duration. It could be assumed that long events are partially detected by short time averages and fully by long time averages. But so far no obvious pattern, which describes when the same event is detected in different averaged time series, can be seen. For example in cluster 3 the third event from the 15 s averaged data is partially detected

in the 10 s averaged time series but not at all in the 12 s one. These results point to a large sensitivity of the method to the scale at which it is applied. In cluster 4 the detected events in the higher averaged data tend to be more to the beginning of the time series. Contrary, in cluster 2 they are more evenly distributed over the whole time period. It is noticeable that cluster 1 has shorter and less events than the other clusters. In general, the 6 s averaged data is a sensible choice because the detected events from the 6 s averaged data overlap the most with events from lower and higher averages. To verify this choice we will have a closer look at the detected events in the 3 s, 6 s and 12 s averaged data. Appendix B shows the detailed results from this analysis. The start and end points of all events, the plots of them and the percentage of events are stated in Appendix B. Moreover, there are several plots which visualize the overlaps between the detected events in the three different scales. Figure 6 shows the results for the 6 s based results for sonic 5 and all clusters.

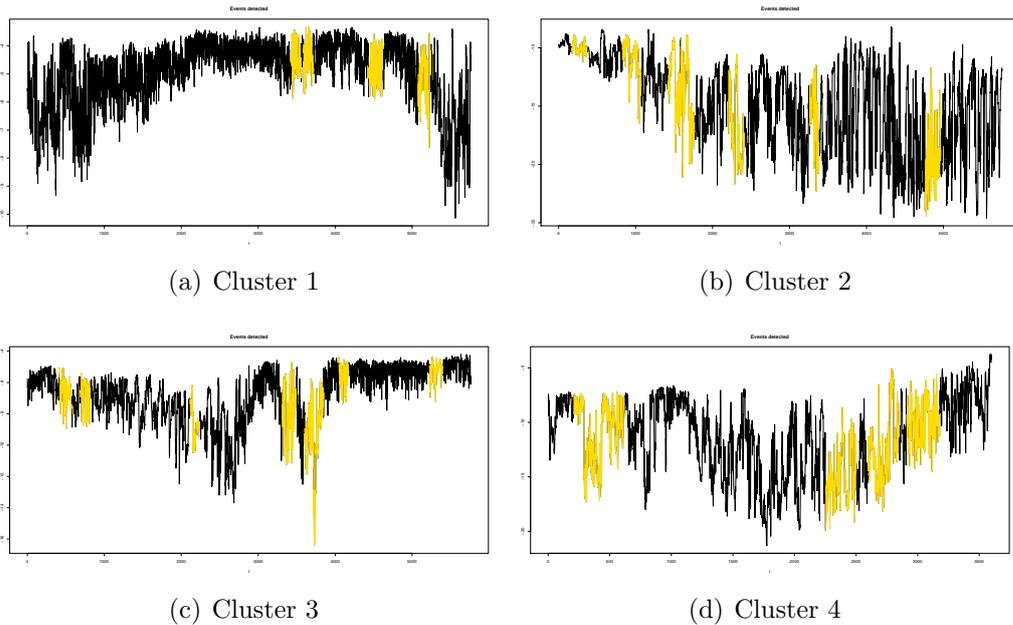


Figure 6: Detected events in 6 s averaged temperature data from sonic 5 and all clusters

By looking at the percentages, we can conclude that the 6 s averaged data has the highest percentage of detected events in all clusters except from cluster 2 in which the 12 s averaged data has a higher percentage. Cluster 2 and 4 have a higher percentage of detected events than cluster 1 and 3. On average roughly 20% of the time series were detected as events. Kang et al.

(2015a) obtained a similar percentage in their analysis.

In Appendix C the time periods recognised as events for different sonics are compared. Most of the events are detected by several sonics. Consequently, the method seems to give reliable results. Sonic 5, which is located closer to the surface, has the highest number of events for most of the clusters.

Even though Kang et al. (2015a) mainly used their method to analyse temperature data, we want to compare wind speed, wind direction and temperature events. Again, we use the 6 s averaged data and a time window length of 120 points. The wind speed is calculated with

$$\sqrt{u^2 + v^2}$$

and the wind direction with

$$\arctan\left(\frac{v}{u}\right)$$

u and v are the sonic coordinates. In Figure 7 there is a sketch of this coordinate system.



Figure 7: sketch of the sonic coordinate system

If $v = 0$ and $u > 0$ the wind is coming directly into the instrument. This corresponds to easterly wind. If $v > 0$ and $u = 0$ the wind is coming from the side of the instrument and if $u < 0$ the wind is coming from behind the sensor. In Appendix D the results of this analysis are stated. In cluster 1 the wind speed has mostly longer events than the ones in the temperature and wind direction time series. In cluster 2 the wind speed events cover significantly less of the full time series than the detected events in the temperature and wind direction time series. There are some overlaps between temperature, wind speed and wind direction events.

Furthermore we see some sharp changes over a short time period in the temperature events with up to 10 degrees difference but there are non in the wind events. In cluster 3 some overlaps between the events can be observed and the percentage of detected events is for all three components roughly the same. But in cluster 4 there is a very high percentage of events in the wind direction data. It is up to 45% of the total time series length. Moreover we observe some sharp changes in all three components. In general we can see that some of the wind direction events are very long. For example in cluster 4 sonic 6 one event is about 45 minutes long. To check if there is a relation between mean wind speed and wind direction we look at the wind roses for all clusters (Figure 8).

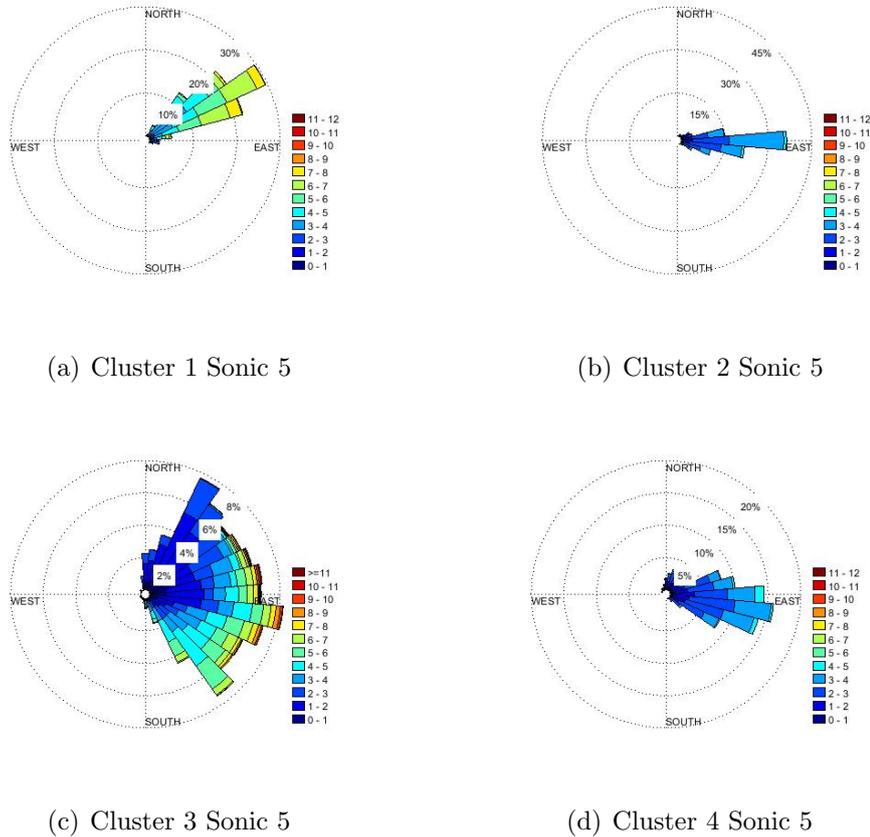


Figure 8: Wind roses for all clusters

In Figure 8 the colour shows the wind speed and the length shows the frequency of a wind direction. The wind roses look nearly the same for all sonics. Consequently, it is sufficient to focus on the ones generated from

sonic 5 measurements. Based on the wind rose plots we can conclude that in cluster 2 and 4 the mean wind is fairly slow with a maximum of 4 to 5 m/s. The difference between the two clusters is that in cluster 2 the frequency for the wind coming directly from the east is much higher than in cluster 4. We saw some sharp changes in the detected wind direction events in cluster 4. This behaviour can also be seen in the wind rose plot. Roughly 5% of the mean wind is changing its direction from north to south east. This wind is very slow. Hence, the wind direction is more stable in cluster 2, and variable in cluster 4 mainly for slow wind events.

Cluster 3 stands out because it has the highest wind speed, which is nearly twice as high as in cluster 2 and 4, and the wind direction is changing from south east to north east. In cluster 1 the wind comes from north east and the wind speed goes up to 8 m/s.

6 Conclusions

Events in the stably stratified atmospheric boundary layer were detected using a recently developed method (Kang et al. (2015a)). After a multiscale analysis the window size for this method was chosen to be 120 points on 6s averaged data. Before applying the event detection method the data was separated into four clusters characterized by different turbulence behaviour by Vercauteren and Klein (2015). The first cluster encompasses the shortest and the fewest events while cluster 4 tends to have the longest events and cluster 2 has the highest number of events. Moreover cluster 2 and 4 are characterised by weak wind which is mainly coming from the east. Contrary to this cluster 3 is decomposed of faster wind which is changing its direction from north east to south east.

As a conclusion it can be said that the method was determined to give reliable results. Though we came across some limitations. First of all, the results from the event detection method by Kang et al. (2015a) is sensitive to the scale at which it is applied. Testing why some events which were detected in the 6 s averaged time series were not detected in the 3 s one did not give any explanations for this sensitivity. Either the ZA tests defined them as unit root processes, or the noise test defined them as noise. A multiscale approach is an option to work around this scale dependency. In general, the multiscale approach is a good way to identify which average and time window gives the most reasonable output. The problem is that usually large datasets are analysed and hence a multiscale approach would result in an even higher computational effort. Moreover, the method assumes that

events are always separated by noise. This seems to be a valid assumption for the dataset which was analysed in this thesis but in general this assumption is not proven. Hence, events might not be detected by this method.

7 Acknowledgement

The author thanks the Environmental Fluid Mechanics Laboratory at Ecole Polytechnique Fédérale de Lausanne for providing the data and Nikki Vercauteren and Danijel Belušić for helpful discussions. I am grateful to two anonymous reviewers for their comments that helped to improve the thesis substantially.

8 References

- Bou-Zeid, E., C. Higgins, H. Huwald, C. Meneveau and M. B. Parlange (2010). Field study of the dynamics and modelling of subgrid-scale turbulence in a stable atmospheric surface layer over a glacier. *Journal of Fluid Mechanics*. 665: 480515, doi: 10.1017/S0022112010004015
- Faranda, D., F. M. E. Pons, B. Dubrulle, F. Daviaud, B. Saint-Michel, É. Herbert, and P. Cortet (2014). Modelling and analysis of turbulent datasets using ARMA processes, doi: 10.1063/1.4896637
- Garratt, J.R., (1994). The atmospheric boundary layer. *Cambridge University Press*, ISBN: 0-521-46745-4
- Kang, Y., D. Belušić, and K. Smith-Miles (2014). Detecting and Classifying Events in Noisy Time Series. *Q.J.R. Meteorol. Soc.*, doi: 10.1775/JAS-D-13-0182.1
- Kang, Y., D. Belušić, and K. Smith-Miles (2015a). Classes of structures in the stable atmospheric boundary layer. *Q.J.R. Meteorol. Soc.*, doi: 10.1002/qj.2501
- Kang, Y., D. Belušić, and K. Smith-Miles (2015b). [Package TED](https://cran.r-project.org/web/packages/TED/TED.pdf), url: <https://cran.r-project.org/web/packages/TED/TED.pdf>

Mahrt, L., (2011). The Near-Calm Stable Boundary Layer. *Boundary-Layer Meteorology* 140(3): 343360. doi: 10.1007/s10546-011-9616-2

Shumway, R. H., and D. S. Stoffer (2011). *Time Series Analysis and its Applications*. Springer Science+Business Media, doi: 10.1007/978-1-4419-7865-3

Storch, H., and F. Zwiers (1999). *Statistical Analysis in Climate Research*. Cambridge University Press, doi: 10.1017/CB09780511612336

Tsay, R. S., (2005). *Analysis of Financial Time Series*. 2nd ed. Wiley-Interscience, 576 pp.

Vercauteren, N., and R. Klein (2015). A Clustering Method to Characterize Intermittent Bursts of Turbulence and Interaction with Submesoscale Motions in the Stable Boundary Layer. *Journal of the atmospheric sciences* 72(4): 15041517, doi: 10.1175/JAS-D-14-0115.1

Weston, S. (2015). [Package foreach](https://cran.r-project.org/web/packages/foreach/vignettes/foreach.pdf),
url: <https://cran.r-project.org/web/packages/foreach/vignettes/foreach.pdf>

Wyngaard, J. C., (1990). Atmospheric Turbulence. *Annu. Rev. Fluid Mech.* 24: 205-33

A Appendix

Multiscale approach

The following graphics visualize the multiscale approach. The lines show the time periods recognised as events for different averaging times ranging from 1 second to 15 seconds. The sliding window length was chosen to be 120 points. Hence, the event length varies from 2 minutes to 30 minutes. The 7s, 11 and 13s average are not part of this analysis because the lengths of the time series are no multiple of these averages.

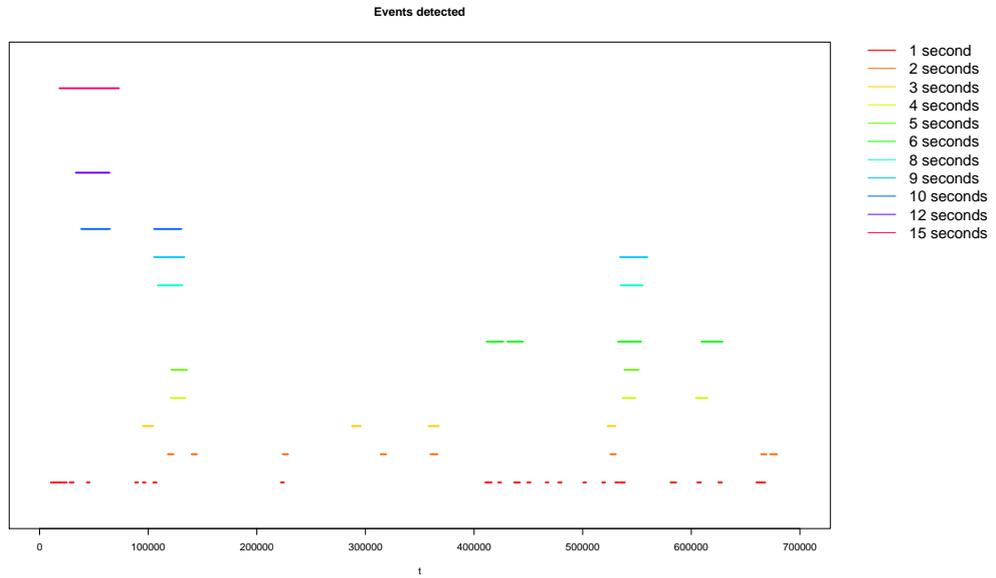


Figure 9: Graphic of the multiscale analysis for Cluster 1

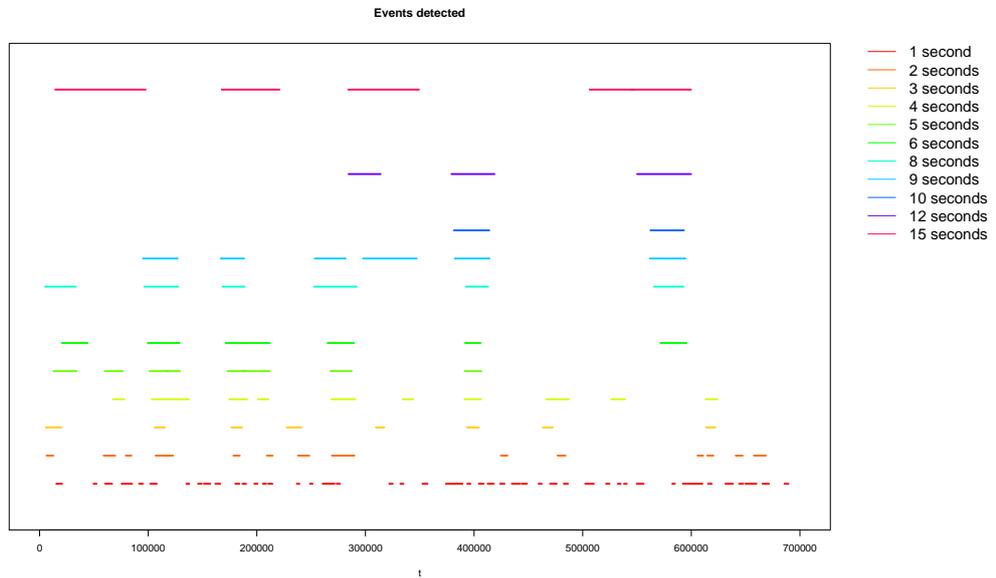


Figure 10: Graphic of the multiscale analysis for Cluster 2

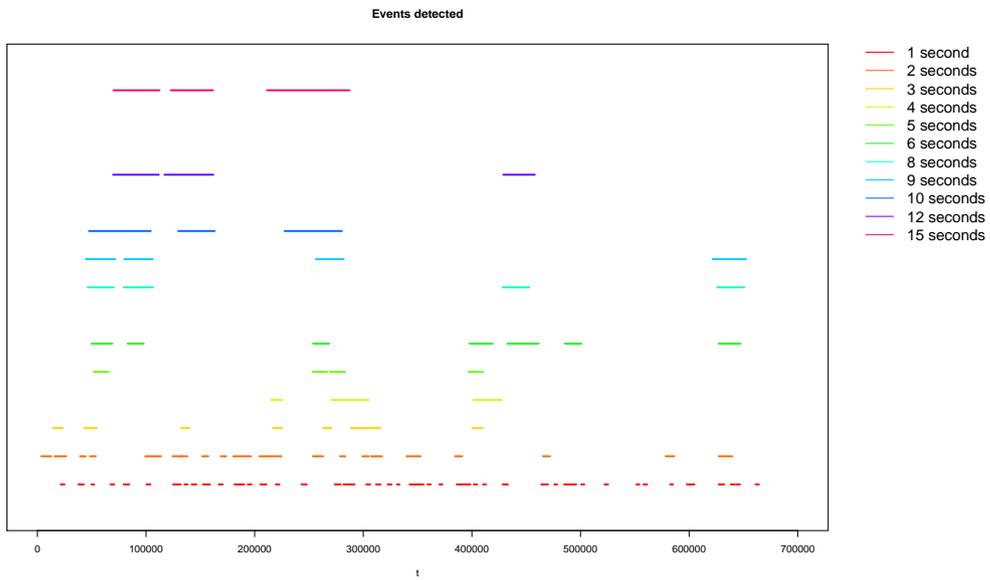


Figure 11: Graphic of the multiscale analysis for Cluster 3

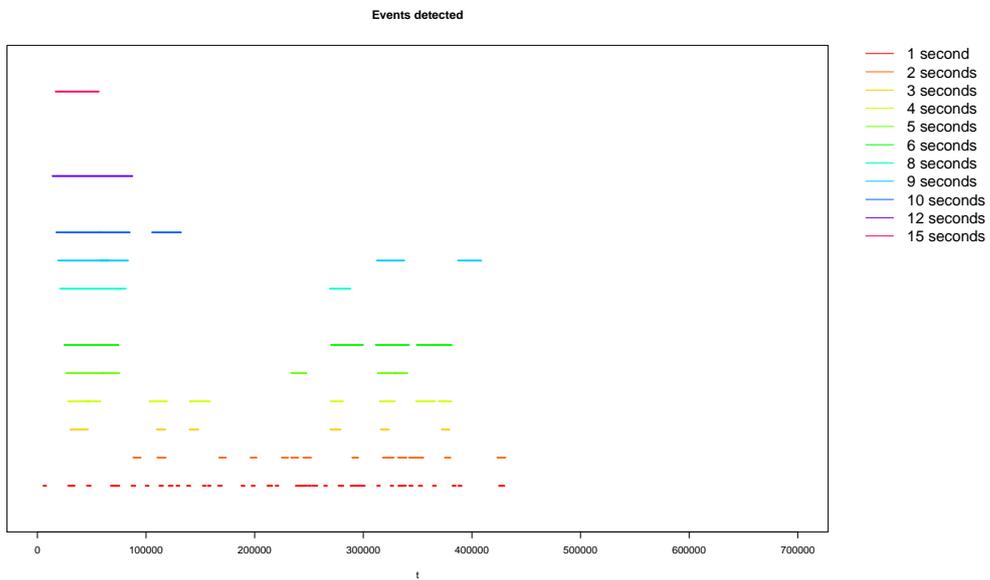


Figure 12: Graphic of the multiscale analysis for Cluster 4

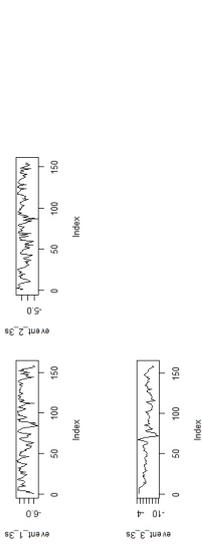
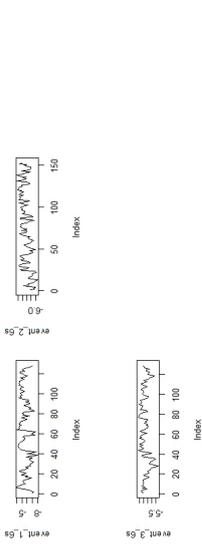
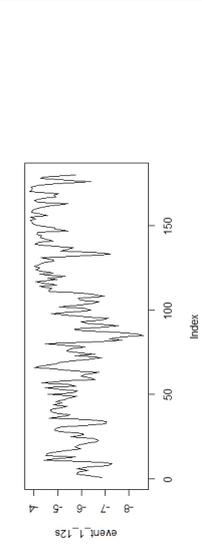
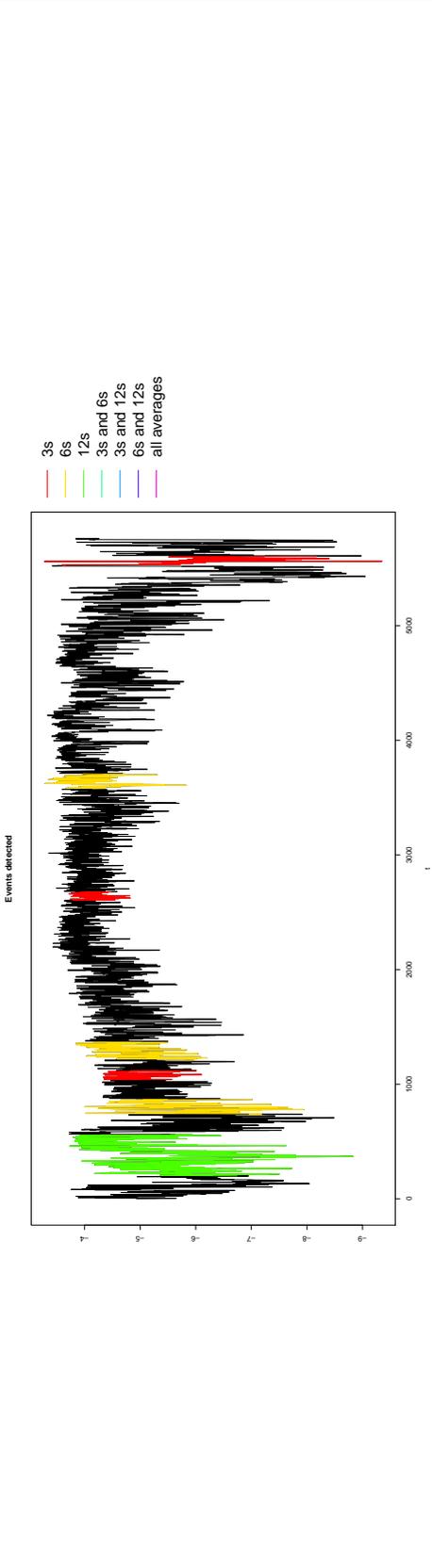
B Appendix

Event detection method applied to 3 s, 6 s and 12 s averaged temperature data

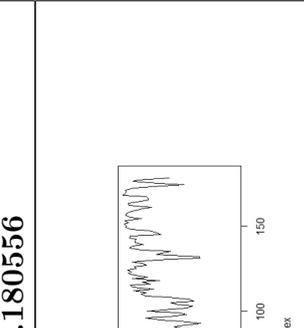
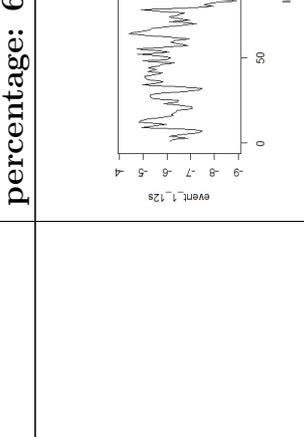
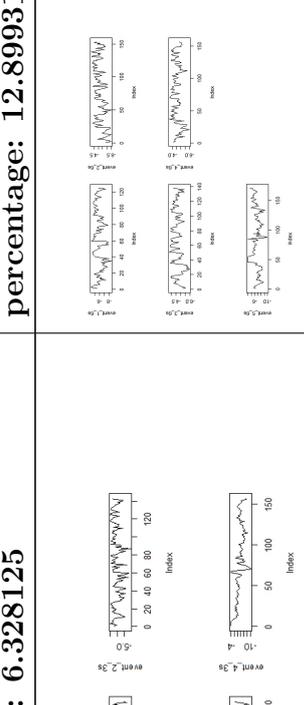
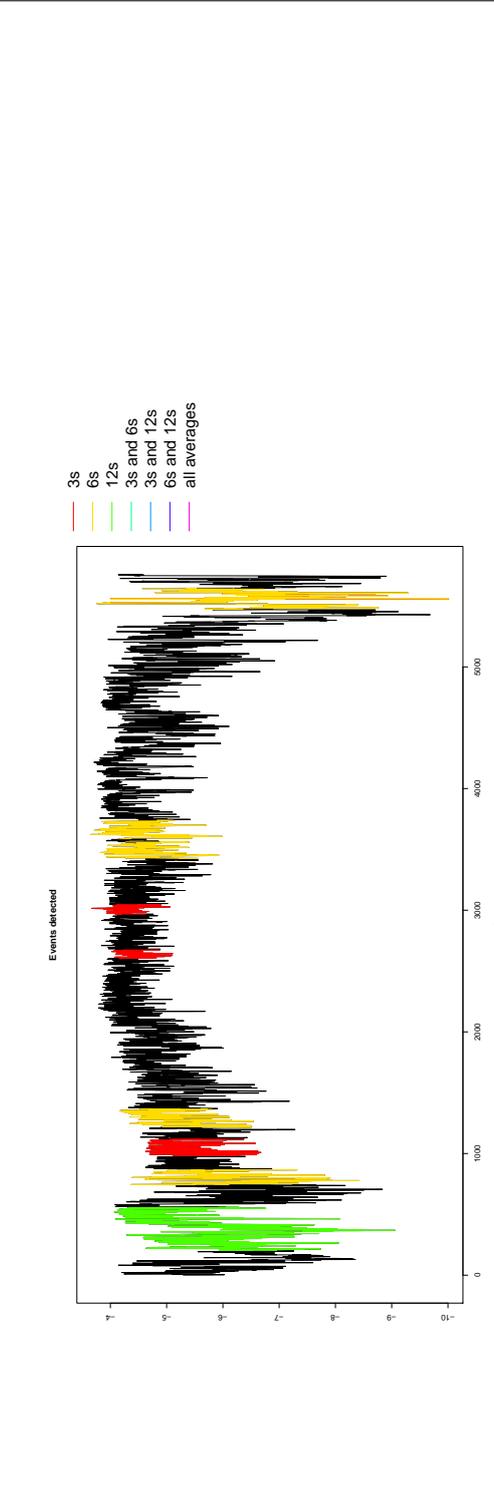
The following table shows the results from the event detection method applied to 3 s, 6 s and 12 s averaged temperature data with a window length of 120 points. For all 4 clusters the start and end points of all events, the percentage of events in the time series and the plots of all event are included in the table. Moreover the overlaps between the events detected in the 3 s, 6 s and 12 s averaged data are coloured in the plot of the 6s averaged time series.

Cluster 1 Sonic 5

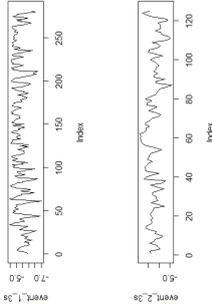
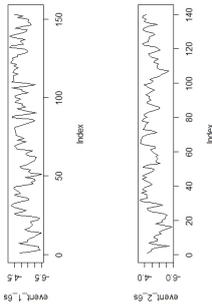
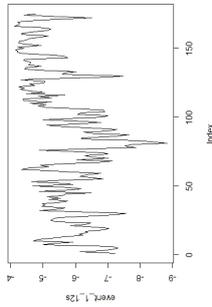
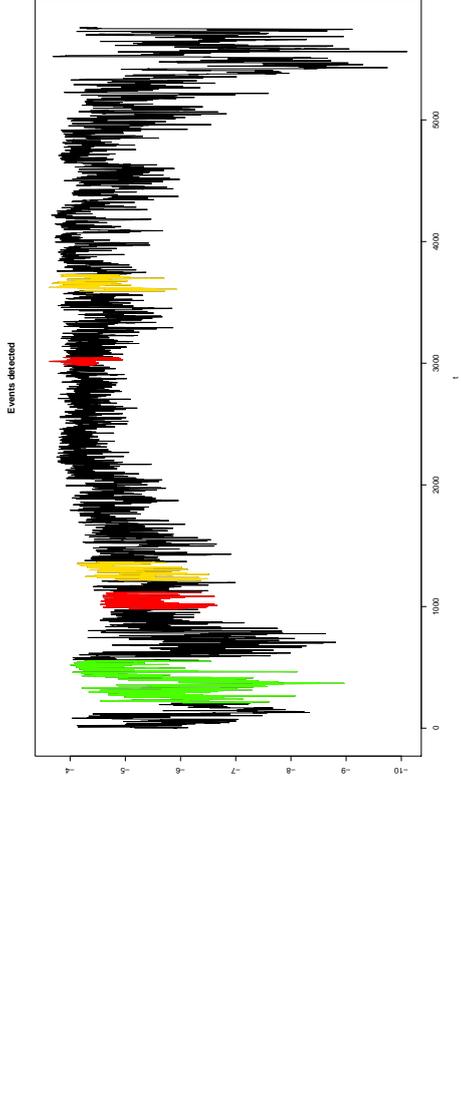
Cluster 1 Sonic 5	
<p>3 second averaged data</p> <p>[start-end]: [1583,1742]; [4794,4927]; [5970,6123]; [8715,8835]</p> <p>percentage: 4.939236</p>	<p>6 second averaged data</p> <p>[start-end]: [3430,3557]; [3589,3709]; [4438,4615]; [5078,5240]</p> <p>percentage: 10.24306</p>
<p>12 second averaged data</p> <p>[start-end]: [139,268]</p> <p>percentage: 4.513889</p>	<p>12 second averaged data</p> <p>[start-end]: [139,268]</p> <p>percentage: 4.513889</p>

Cluster 1 Sonic 6		
3 second averaged data [start-end]: [2079,2236]; [5204,5358]; [11051,11209] percentage: 4.097222	6 second averaged data [start-end]: [739,866]; [1216,1368]; [3579,3707] percentage: 7.118056	12 second averaged data [start-end]: [100,279] percentage: 6.25
		
		

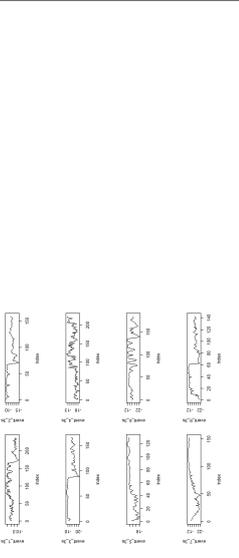
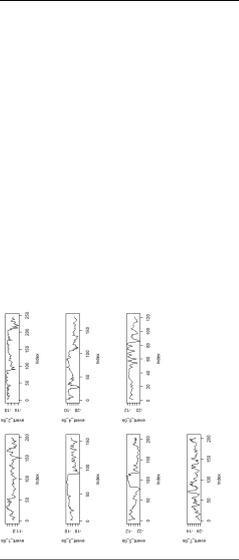
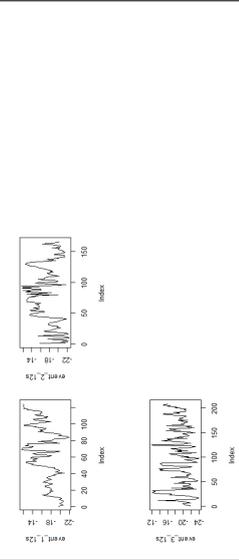
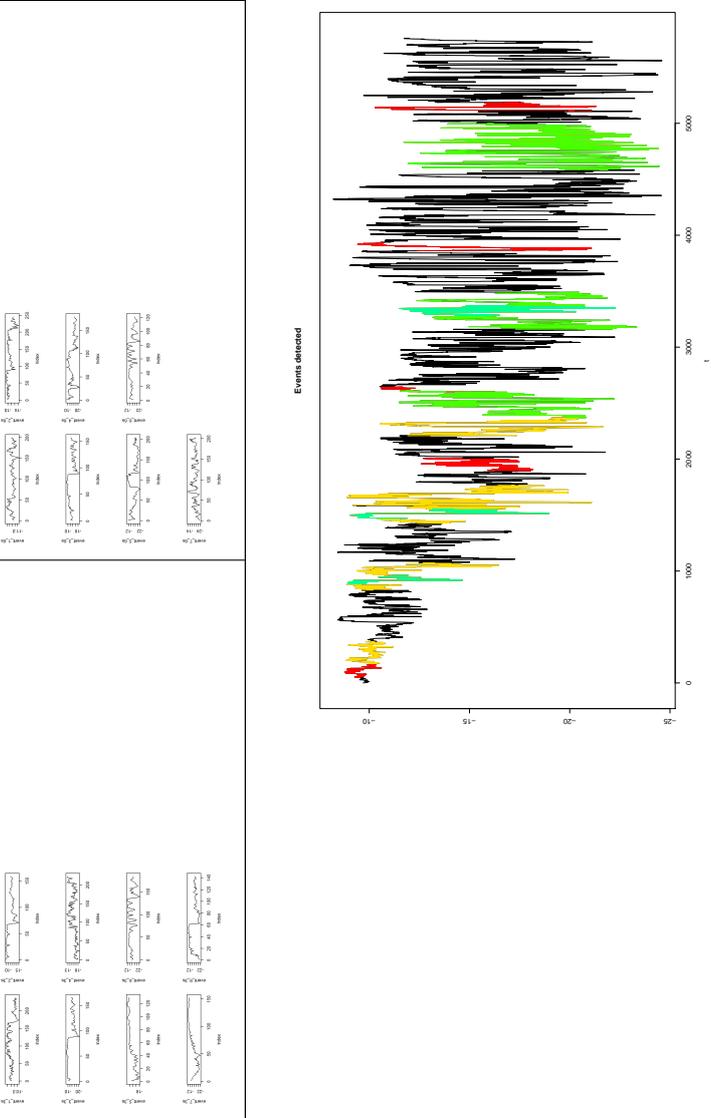
Cluster 1 Sonic 7

<p>3 second averaged data</p> <p>[start-end]: [1960,2088]; [5204,5346]; [5937,6092], [11052,11208]</p> <p>percentage: 6.328125</p>	<p>6 second averaged data</p> <p>[start-end]: [743,867]; [1216,1368]; [3424,3561]; [3586,3741]; [5475,5645]</p> <p>percentage: 12.89931</p>	<p>12 second averaged data</p> <p>[start-end]: [102,279]</p> <p>percentage: 6.180556</p>
		
		

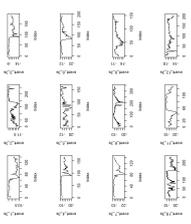
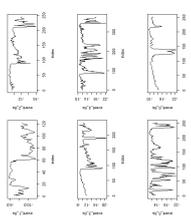
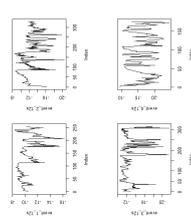
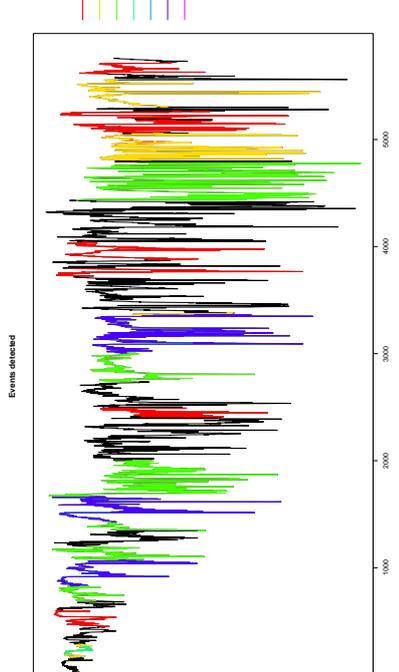
Cluster 1 Sonic 8

<p>3 second averaged data [start-end]: [1956,2236]; [5970,6094] percentage: 3.524306</p>	<p>6 second averaged data [start-end]: [1215,1367]; [3591,3730] percentage: 5.086806</p>	<p>12 second averaged data [start-end]: [104,278] percentage: 6.076389</p>
		
		

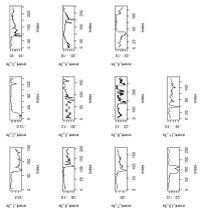
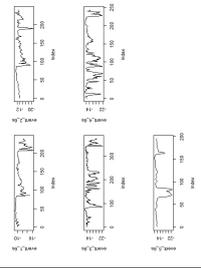
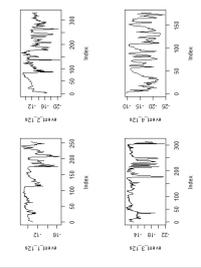
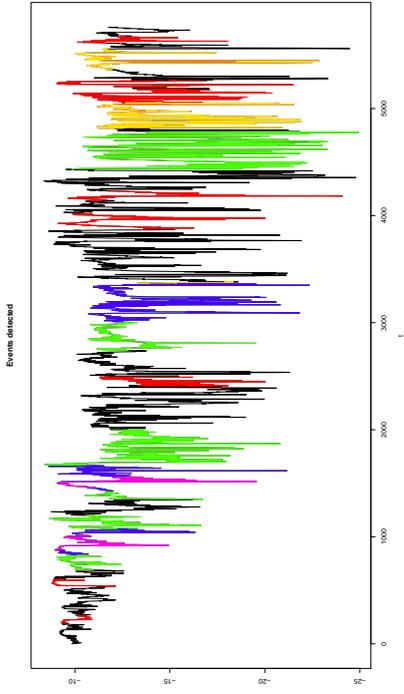
Cluster 2 Sonic 5

3 second averaged data	6 second averaged data	12 second averaged data
<p>[start-end]: [96,332]; [1765,1922]; [2942,3106]; [3793,4014]; [5159,5287]; [6559,6741]; [7722,7873]; [10226,10367]</p> <p>percentage: 12.04861</p> 	<p>[start-end]: [171,369]; [829,1073]; [1427,1583]; [1588,1766]; [2209,2412]; [3264,3384]; [4762,4963]</p> <p>percentage: 22.69097</p> 	<p>[start-end]: [1185,1308]; [1580,1745]; [2292,2499]</p> <p>percentage: 17.29167</p> 
		

Cluster 2 Sonic 6

3 second averaged data	6 second averaged data	12 second averaged data
[start-end]: [385,524]; [896,1205]; [1746,1935]; [2941,3106]; [4822,4986]; [6099,6292]; [7450,7591]; [7751,8091]; [9896,10045]; [10133,10364]; [10399,10527]; [11235,11438]	[start-end]: [166,287]; [829,1072]; [1427,1666]; [3005,3381]; [4806,5056]; [5307,5557]	[start-end]: [346,599]; [673,1000]; [1369,1638]; [2216,2391]
percentage: 20.51215	percentage: 25.78125	percentage: 37.25694
		
		

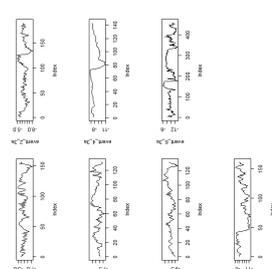
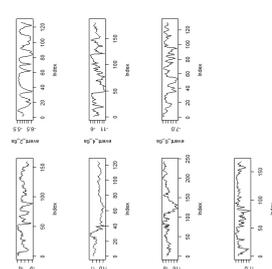
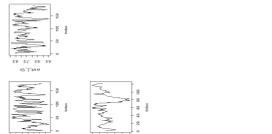
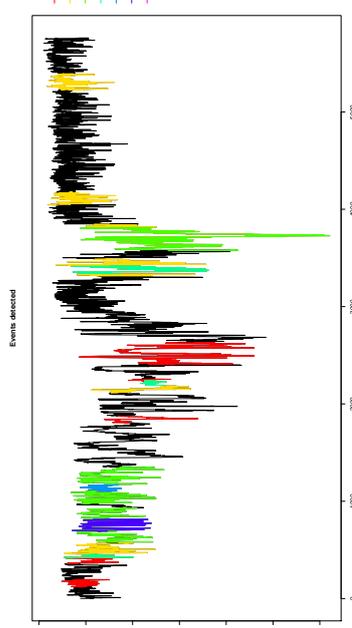
Cluster 2 Sonic 7

3 second averaged data	6 second averaged data	12 second averaged data
<p>[start-end]: [385,530]; [1064,1205]; [1750,2046]; [2919,3117]; [4822,4991]; [7750,8041]; [8285,8460]; [9907,10366]; [10398,10525]; [10774,10978]; [11220,11349]</p>	<p>[start-end]: [834,1072]; [1427,1666]; [3006,3380]; [4814,5056]; [5362,5556]</p>	<p>[start-end]: [346,599]; [672,1000]; [1370,1683]; [2216,2392]</p>
<p>percentage: 20.3559</p>	<p>percentage: 22.43056</p>	<p>percentage: 37.29167</p>
		
		

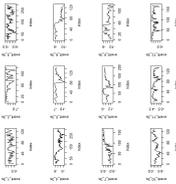
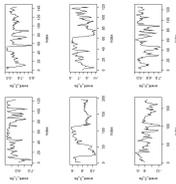
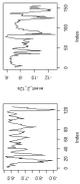
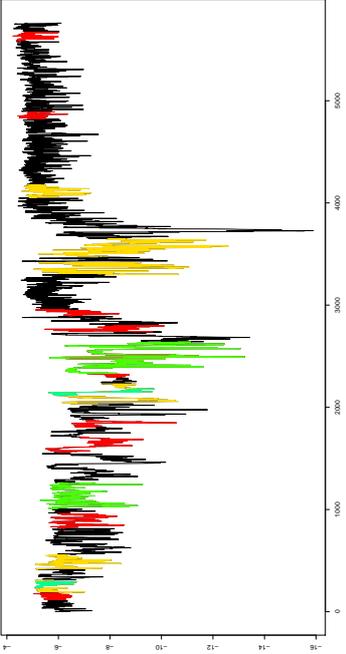
Cluster 2 Sonic 8

3 second averaged data	6 second averaged data	12 second averaged data
[start-end]: [332,512]; [1064,1204]; [1747,2100]; [2839,3106]; [4822,4997]; [7079,7251]; [7363,7579]; [8532,8654]; [9991,10140]; [10222,10366]; [10398,10527]; [10773,11141]; [11212,11362]	[start-end]: [835,1072]; [1427,1665]; [3008,3380]; [4099,4390]; [4814,5056]; [5361,5700]	[start-end]: [345,599]; [672,1003]; [1370,1683]; [2216,2392]
percentage: 22.37847	percentage: 29.94792	percentage: 37.43056

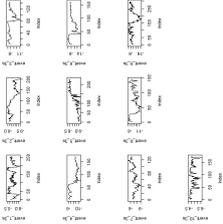
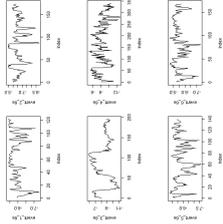
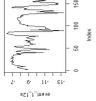
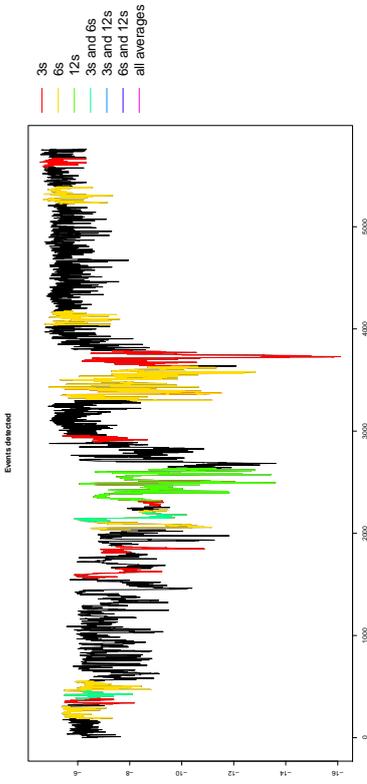
Cluster 3 Sonic 5

3 second averaged data	6 second averaged data	12 second averaged data
[start-end]: [238,393]; [718,907]; [2205,2335]; [3612,3754]; [4379,4510]; [4809,5264]; [6674,6834]	[start-end]: [416,574]; [691,816]; [2115,2238]; [3313,3492]; [3606,3492]; [4045,4174]; [5226,5393]	[start-end]: [290,466]; [487,675]; [1787,1908]
percentage: 11.88368	percentage: 19.60069	percentage: 16.94444
		
		

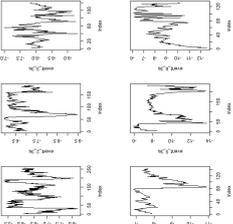
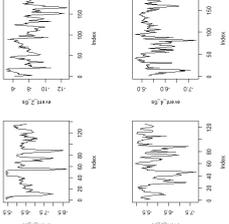
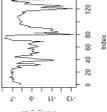
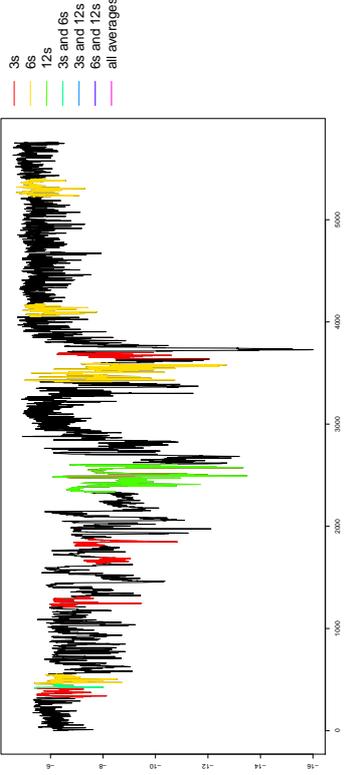
Cluster 3 Somic 6

3 second averaged data	6 second averaged data	12 second averaged data
<p>[start-end]: [235,362]; [471,592]; [1632,1924]; [3114,3415]; [3611,3753]; [4232,4359]; [4583,4745]; [4905,5104]; [5478,5579]; [5790,5917]; [9649,9782]; [11170,11340]</p> <p>percentage: 17.63889</p>	<p>[start-end]: [188,309]; [416,558]; [2035,2230]; [3297,3416]; [3477,3650]; [4052,4177]</p> <p>percentage: 15.29514</p>	<p>[start-end]: [504,631]; [1165,1320]</p> <p>percentage: 9.861111</p>
		
		

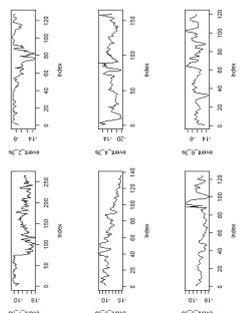
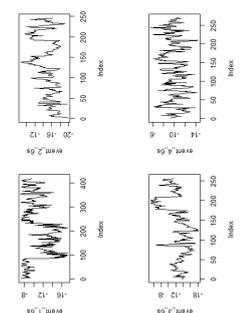
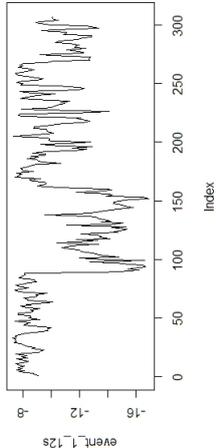
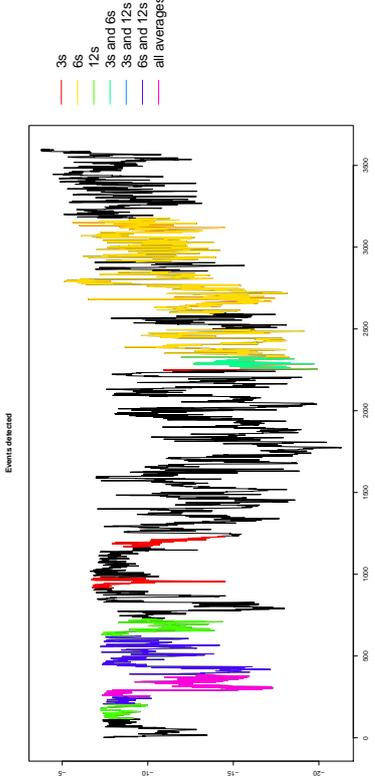
Cluster 3 Sonic 7

3 second averaged data	6 second averaged data	12 second averaged data
[start-end]: [654,907]; [3114,3313]; [3611,3753]; [4233,4402]; [4533,4759]; [4945,5105]; [5789,5917]; [6690,6835]; [7280,7574]; [11168,11339]	[start-end]: [185,305]; [384,557]; [2035,2229]; [3298,3635]; [4038,4176]; [5226,5393]	[start-end]: [1160,1319]
percentage: 16.46701	percentage: 19.70486	percentage: 5.555556
		
		

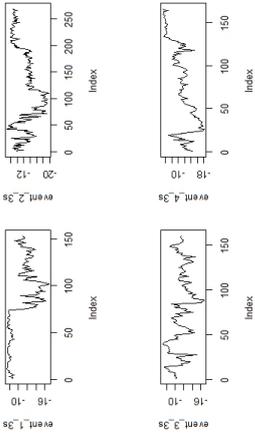
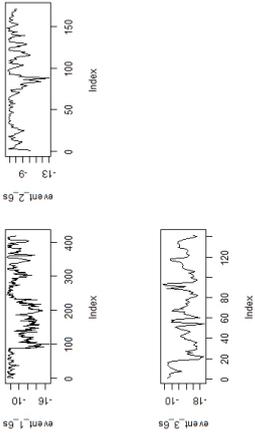
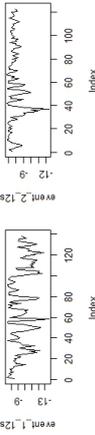
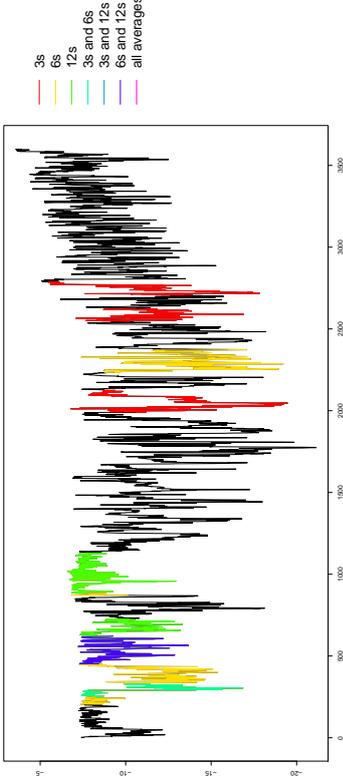
Cluster 3 Somic 8

3 second averaged data	6 second averaged data	12 second averaged data
<p>[start-end]: [653,906]; [2426,2613]; [3274,3393]; [3610,3752]; [4945,5177]; [7271,7401]</p> <p>percentage: 9.279514</p>	<p>[start-end]: [416,551]; [3414,3596]; [4052,4176]; [5225,5396]</p> <p>percentage: 10.69444</p>	<p>[start-end]: [1168,1303]</p> <p>percentage: 4.722222</p>
		
		

Cluster 4 Sonic 5

<p>3 second averaged data</p> <p>[start-end]: [507,772]; [1834,1960]; [2334,2469]; [4495,4653]; [5271,5394]; [6203,6322]</p> <p>percentage: 12.94444</p>	<p>6 second averaged data</p> <p>[start-end]: [207,623]; [2253,2498]; [2596,2851]; [2910,3178]</p> <p>percentage: 33</p>	<p>12 second averaged data</p> <p>[start-end]: [58,364]</p> <p>percentage: 17.05556</p>
		
		

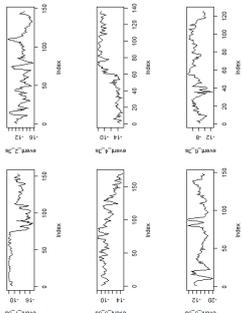
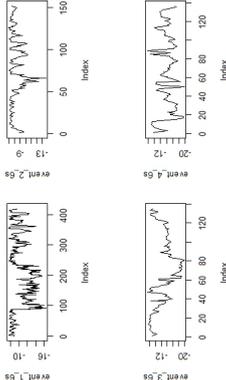
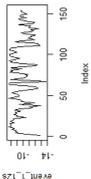
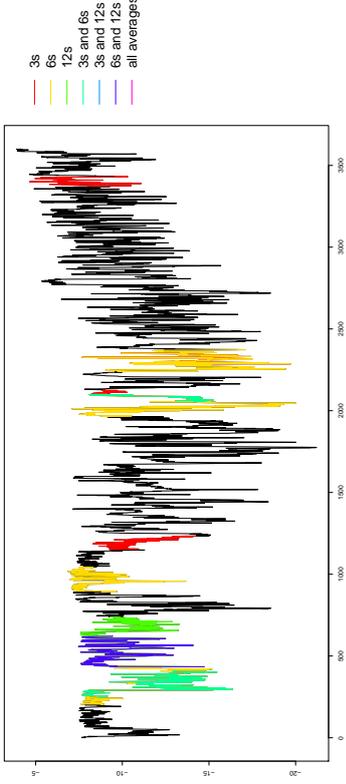
Cluster 4 Sonic 6

<p>3 second averaged data</p> <p>[start-end]: [507,659]; [3982,4250]; [5090,5249]; [5412,5577]</p> <p>percentage: 10.38889</p>	<p>6 second averaged data</p> <p>[start-end]: [204,622]; [871,1042]; [2234,2374]</p> <p>percentage: 20.33333</p>	<p>12 second averaged data</p> <p>[start-end]: [226,363]; [443,565]</p> <p>percentage: 14.5</p>
		
		

Cluster 4 Sonic 7

Cluster 4 Sonic 7		
<p>3 second averaged data</p> <p>[start-end]: [507,659]; [808,927]; [4114,4251]; [5090,5251]</p> <p>percentage: 7.958333</p>	<p>6 second averaged data</p> <p>[start-end]: [204,622]; [895,1042]; [2238,2374]</p> <p>percentage: 19.55556</p>	<p>12 second averaged data</p> <p>[start-end]: [217,364]; [443,565]</p> <p>percentage: 15.05556</p>

Cluster 4 Sonic 8

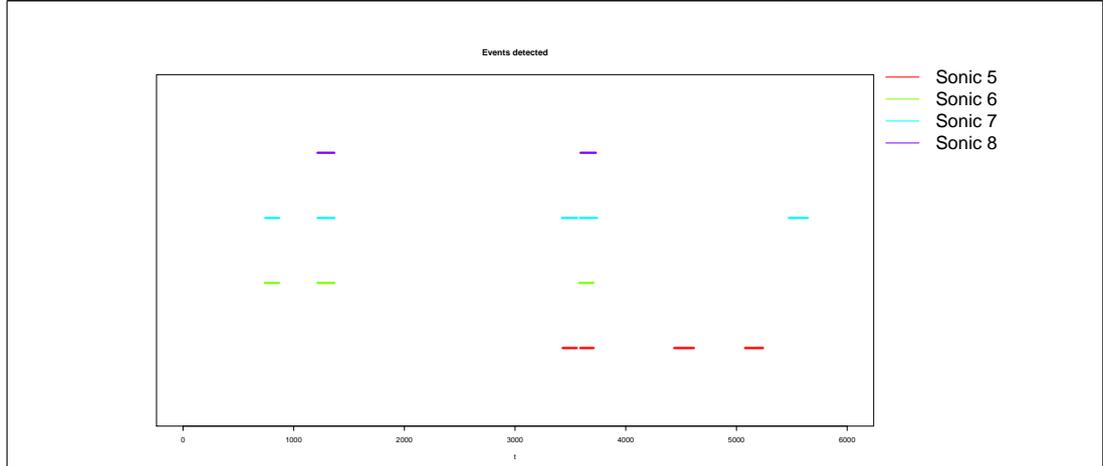
Cluster 4 Sonic 8		
3 second averaged data [start-end]: [507,659]; [670,815]; [2300,2468]; [4116,4251]; [4564,4711]; [6744;6868] percentage: 12.18056	6 second averaged data [start-end]: [204,622]; [893,1044]; [1966,2100]; [2238,2374] percentage: 23.41667	12 second averaged data [start-end]: [217,370] percentage: 8.555556
		
		

C Appendix

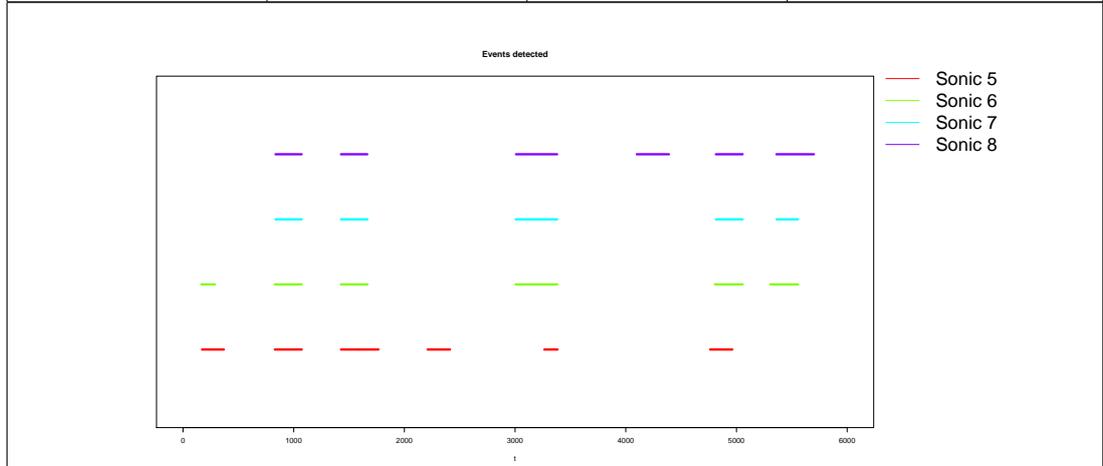
Sonic 5,6,7 and 8: detected events in 6 s averaged temperature data

The following table shows the results from the event detection method applied to 6 s averaged temperature data with a window length of 120 points. For all 4 clusters the start and end points of all events are included in the table. Moreover, the lines in the plots show the time periods recognised as events for different sonics.

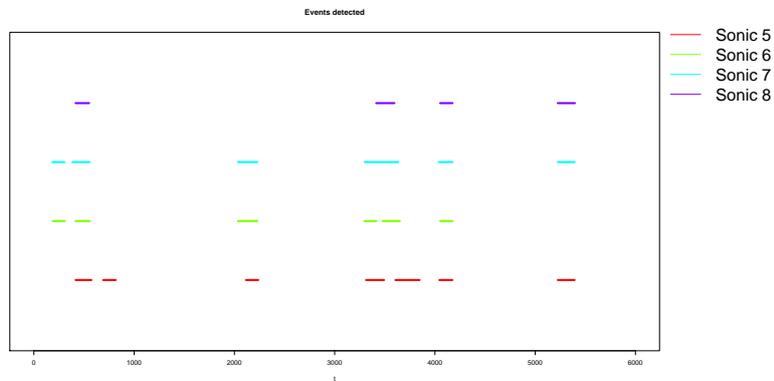
Cluster 1			
Sonic 5	Sonic 6	Sonic 7	Sonic 8
[start-end]: [3430,3557]; [3589,3709]; [4438,4615]; [5078,5240]	[start-end]: [739,866]; [1216,1368]; [3579,3707]	[start-end]: [743,867]; [1216,1368]; [3424,3561]; [3586,3741]; [5475,5645]	[start-end]: [1215,1367]; [3591,3730]



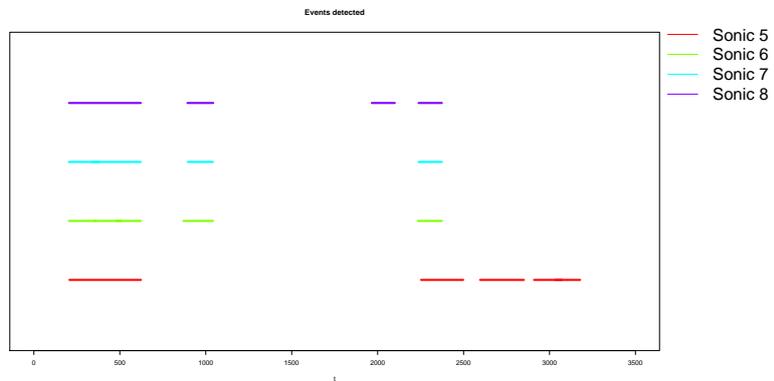
Cluster 2			
Sonic 5	Sonic 6	Sonic 7	Sonic 8
[start-end]: [171,369]; [829,1073]; [1427,1583]; [1588,1766]; [2209,2412]; [3264,3384]; [4762,4963]	[start-end]: [166,287]; [829,1072]; [1427,1666]; [3005,3381]; [4806,5056]; [5307,5557]	[start-end]: [834,1072]; [1427,1666]; [3006,3380]; [4814,5056]; [5362,5556]	[start-end]: [835,1072]; [1427,1665]; [3008,3380]; [4099,4390]; [4814,5056]; [5361,5700]



Cluster 3			
Sonic 5	Sonic 6	Sonic 7	Sonic 8
[start-end]: [416,574]; [691,816]; [2115,2238]; [3313,3492]; [3606,3492]; [4045,4174]; [5226,5393]	[start-end]: [188,309]; [416,558]; [2035,2230]; [3297,3416]; [3477,3650]; [4052,4177]	[start-end]: [185,305]; [384,557]; [2035,2229]; [3298,3635]; [4038,4176]; [5226,5393]	[start-end]: [416,551]; [3414,3596]; [4052,4176]; [5225,5396]



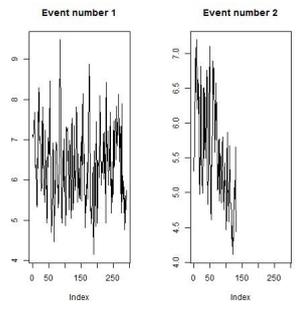
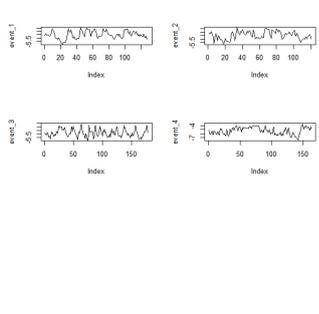
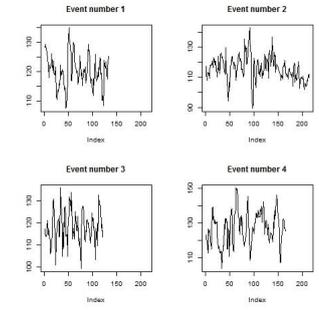
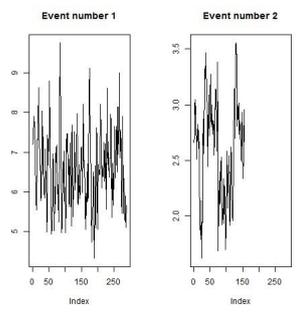
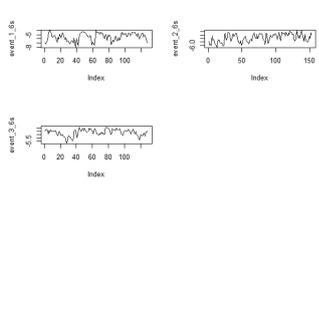
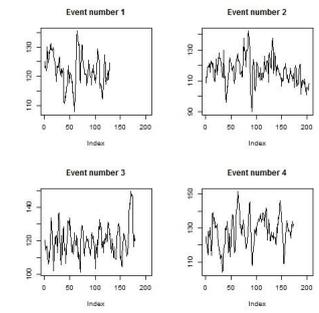
Cluster 4			
Sonic 5	Sonic 6	Sonic 7	Sonic 8
[start-end]: [207,623]; [2253,2498]; [2596,2851]; [2910,3178]	[start-end]: [204,622]; [871,1042]; [2234,2374]	[start-end]: [204,622]; [895,1042]; [2238,2374]	[start-end]: [204,622]; [893,1044]; [1966,2100]; [2238,2374]

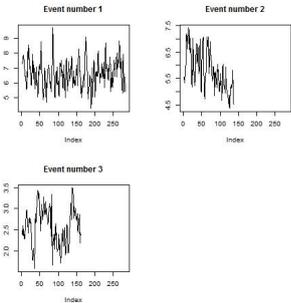
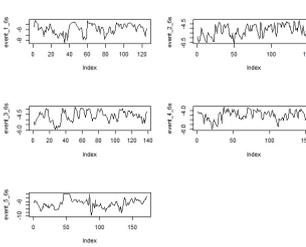
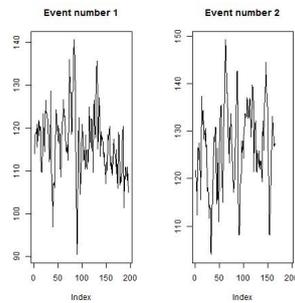
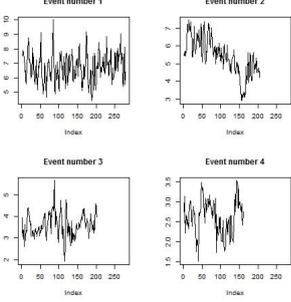
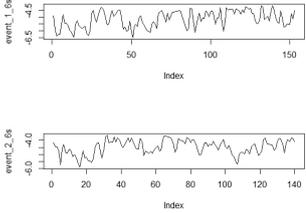
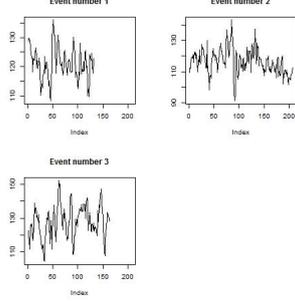


D Appendix

Event detection method applied to 6 s averaged temperature, wind speed and wind direction data

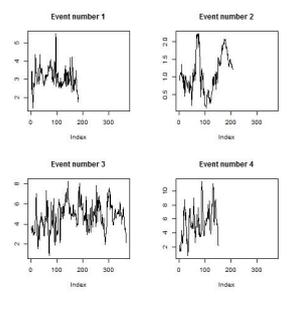
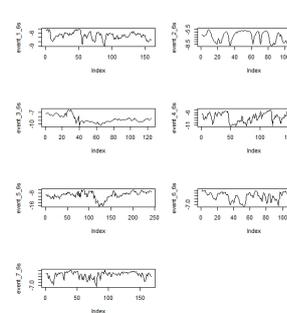
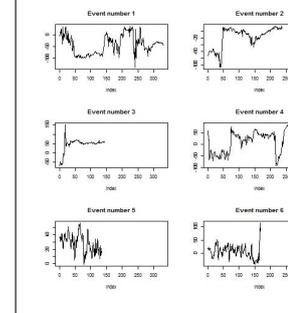
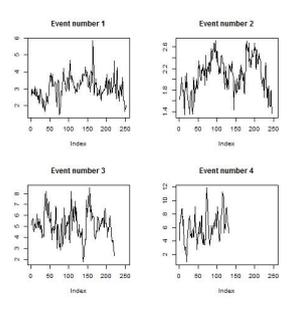
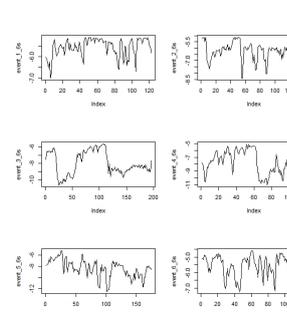
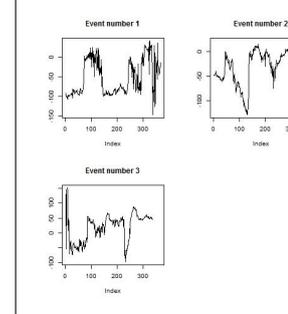
The following table shows the results from the event detection method applied to 6 s averaged temperature, wind speed and wind direction data with a window length of 120 points. For all 4 clusters the start and end points of all events, the percentage of events in the time series and the plots of all event are included in the table.

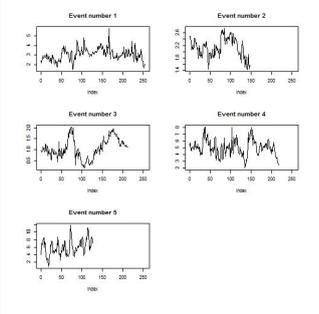
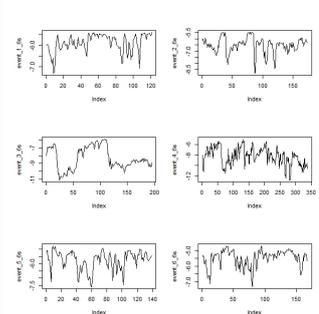
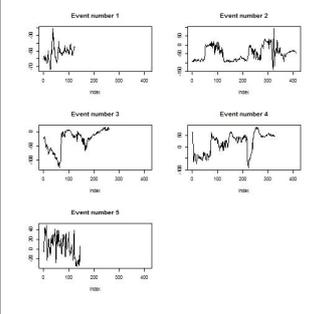
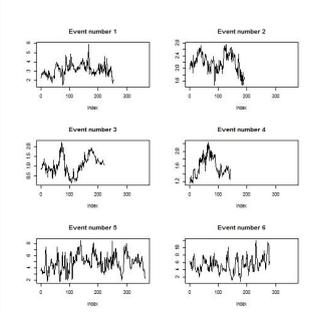
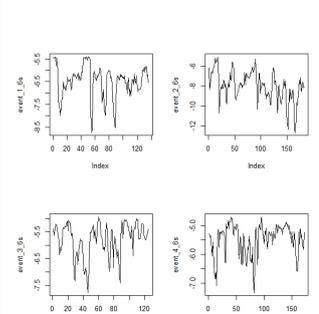
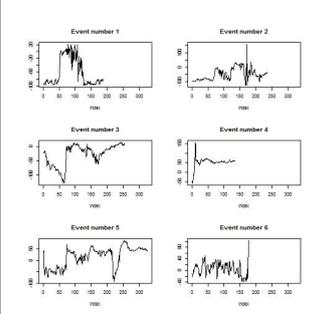
Cluster 1 Sonic 5		
wind speed	temperature	wind direction
[start-end]: [2186,2477]; [3134,3265]	[start-end]: [3430,3557]; [3589,3709]; [4438,4615]; [5078,5240]	[start-end]: [1484,1616]; [1667,1880]; [2537,2657]; [4859,5023]
percentage: 7.361111	percentage: 10.24306	percentage: 10.98958
		
Cluster 1 Sonic 6		
wind speed	temperature	wind direction
[start-end]: [2187,2474]; [5487,5642]	[start-end]: [739,866]; [1216,1368]; [3579,3707]	[start-end]: [1470,1597]; [1673,1876]; [2542,2719]; [4859,5031]
percentage: 7.708333	percentage: 7.118056	percentage: 11.85764
		

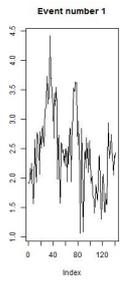
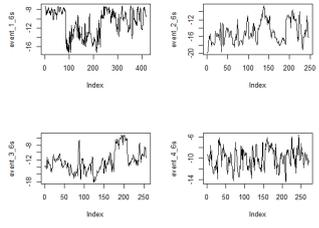
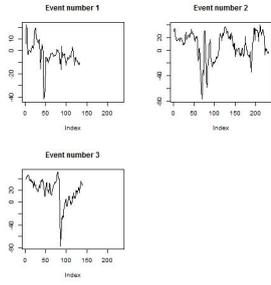
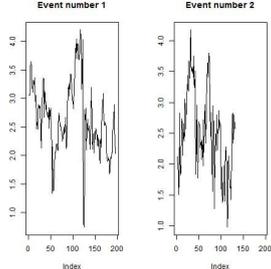
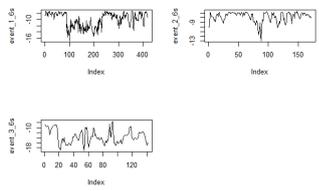
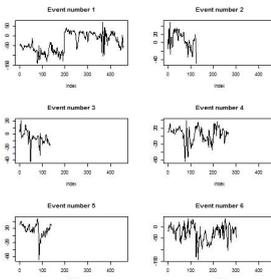
Cluster 1 Sonic 7		
wind speed	temperature	wind direction
[start-end]: [2187,2469]; [3130,3267]; [5478,5638]	[start-end]: [743,867]; [1216,1368]; [3424,3561]; [3586,3741]; [5475,5645]	[start-end]: [1675,1869]; [4860,5024]
percentage: 10.10417	percentage: 12.89931	percentage: 6.25
		
Cluster 1 Sonic 8		
wind speed	temperature	wind direction
[start-end]: [2188,2466]; [3128,3332]; [3932,4133]; [5473,5633]	[start-end]: [1215,1367]; [3591,3730]	[start-end]: [1484,1615]; [1673,1879]; [4860,5023]
percentage: 14.70486	percentage: 5.086806	percentage: 8.732639
		

Cluster 2 Sonic 5		
wind speed	temperature	wind direction
[start-end]: [1381,1514]; [1975,2132]; [3190,3405]; [5178,5371]	[start-end]: [171,369]; [829,1073]; [1427,1583]; [1588,1766]; [2209,2412]; [3264,3384]; [4762,4963]	[start-end]: [163,291]; [444,635]; [863,1228]; [1432,1615]; [2372,2544]; [3280,3407]; [4751,4915]; [5175,5296]
percentage: 12.1875	percentage: 22.69097	percentage: 25.32986
Cluster 2 Sonic 6		
wind speed	temperature	wind direction
[start-end]: [[3078,3206]; [3257,3386]; [4831,5079]	[start-end]: [166,287]; [829,1072]; [1427,1666]; [3005,3381]; [4806,5056]; [5307,5557]	[start-end]: [450,721]; [1012,1229]; [2078,2260]; [2346,2485]; [3280,3639]; [4197,4365]; [4402,4549]; [4751,4871]
percentage: 8.819444	percentage: 25.78125	percentage: 27.96875

Cluster 2 Sonic 7		
wind speed	temperature	wind direction
[start-end]: [3253,3389]; [3637,3795]; [4377,4507]; [4876,5080]	[start-end]: [834,1072]; [1427,1666]; [3006,3380]; [4814,5056]; [5362,5556]	[start-end]: [450,720]; [2346,2478]; [3265,3651]; [4402,4549]; [4751,4871]
percentage: 10.97222	percentage: 22.43056	percentage: 18.40278
Cluster 2 Sonic 8		
wind speed	temperature	wind direction
[start-end]: [1382,1546]; [3637,4036]; [4344,4509]; [4780,5080]	[start-end]: [835,1072]; [1427,1665]; [3008,3380]; [4099,4390]; [4814,5056]; [5361,5700]	[start-end]: [449,720]; [2346,2478]; [3279,3651]; [4403,4549]
percentage: 17.91667	percentage: 29.94792	percentage: 16.05903

Cluster 3 Sonic 5		
wind speed	temperature	wind direction
[start-end]: [142,324]; [1622,1828]; [4081,4446]; [4693,4843]	[start-end]: [416,574]; [691,816]; [2115,2238]; [3313,3492]; [3606,3847]; [4045,4174]; [5226,5393]	[start-end]: [1480,1810]; [2103,2338]; [2814,2956]; [3504,3833]; [4401,4534]; [5198,5365]
percentage: 15.74653	percentage: 19.60069	percentage: 23.29861
		
Cluster 3 Sonic 6		
wind speed	temperature	wind direction
[start-end]: [73,324]; [886,1130]; [4227,4446]; [4707,4838]	[start-end]: [188,309]; [416,558]; [2035,2230]; [3297,3416]; [3477,3650]; [4052,4177]	[start-end]: [1381,1746]; [2015,2333]; [3494,3828]
percentage: 14.73958	percentage: 15.29514	percentage: 17.70833
		

Cluster 3 Sonic 7		
wind speed	temperature	wind direction
[start-end]: [70,324]; [985,1131]; [1619,1831]; [4230,4446]; [4707,4834]	[start-end]: [185,305]; [384,557]; [2035,2229]; [3298,3635]; [4038,4176]; [5226,5393]	[start-end]: [668,792]; [1400,1810]; [2080,2340]; [3503,3828]; [5216,5361]
percentage: 16.66667	percentage: 19.70486	percentage: 22.03125
		
Cluster 3 Sonic 8		
wind speed	temperature	wind direction
[start-end]: [71,324]; [943,1132]; [1620,1840]; [2269,2411]; [4084,4446]; [4549,4827]	[start-end]: [416,551]; [3414,3596]; [4052,4176]; [5225,5396]	[start-end]: [1400,1587]; [1552,1786]; [2079,2332]; [2821,2954]; [3505,3829]; [5188,5364]
percentage: 25.17361	percentage: 10.69444	percentage: 22.79514
		

Cluster 4 Sonic 5		
wind speed	temperature	wind direction
[start-end]: [3239,3379] end: 3379	[start-end]: [207,623]; [2253,2498]; [2596,2851]; [2910,3178]	[start-end]: [1398,1530]; [1806,2037]; [2635,2774]
percentage: 3.916667	percentage: 33	percentage: 14.02778
		
Cluster 4 Sonic 6		
wind speed	temperature	wind direction
[start-end]: [2666,2861]; [3243,3374]	[start-end]: [204,622]; [871,1042]; [2234,2374]	[start-end]: [96,554]; [1396,1544]; [1799,2065]; [2634,2774]; [2891,3040]; [3322,3477]
percentage: 9.111111	percentage: 20.33333	percentage: 36.72222
		

Cluster 4 Sonic 7		
wind speed	temperature	wind direction
[start-end]: [1517,1654]; [2672,2862]; [3239,3374]	[start-end]: [204,622]; [895,1042]; [2238,2374]	[start-end]: [96,552]; [748,873]; [1396,1530]; [1799,2065]; [2635,2774]; [3269,3570]
percentage: 12.91667	percentage: 19.55556	percentage: 39.63889
Cluster 4 Sonic 8		
wind speed	temperature	wind direction
[start-end]: [2456,2628]; [2672,2862]; [3239,3380]	[start-end]: [204,622]; [893,1044]; [1966,2100]; [2238,2374]	[start-end]: [96,547]; [748,873]; [1396,1530]; [1800,2063]; [2635,2774]
percentage: 14.05556	percentage: 23.41667	percentage: 31.02778

E Appendix

Algorithms which were developed for this thesis

The following algorithms were used for this analysis. They are all written in the programming language R, require the foreach package (Weston (2015)) and use the results from the eventDetection function which is part of the TED package (Kang et al. (2015b)). At the end of this Appendix there is an example where the following functions are used.

Algorithm 1: Average

Description

Average data

Arguments

x: vector or time series

average: number

Value

a: vector consisting the averaged time series

```
average.events<-function(x,average){  
  l<-length(x)  
  group<-factor(rep(1:(l/average),each=average))  
  a<-tapply(x,group,mean)  
  result <- list(a=a)  
  return(a)  
}
```

Algorithm 2: Multiscale Analysis

Description

Analyse different scales

Arguments

x: vector or time series

smAverage: smallest average

alpha: significance level

nAverage: number of averages to be analysed

w: window length

Hz: Hz of the data

Value

plot of the different time scales

```
multiscale<-function(x,smAverage,alpha,nAverage,w,Hz){
  l<-length(x)
  second<-numeric(0)
  colour=rainbow(nAverage+1)
  options("scipen" = 20)
  plot(1, main = "Multiscale analysis", yaxt='n',type = "n", xlab
    = "t",ylab="",xlim=c(0, l+1000), ylim=c(0,nAverage+1))
  i=0
  for (i in 0:(nAverage-1)){
    average      <-smAverage+i*Hz
    sec          <-average/Hz
    group        <-factor(rep(1:(l/average),each=average))
    if(length(group)==1){
      x_averaged <-average.events(x,average)
      events     <-eventDetection(x_averaged, w, noiseType =
        'red',parallel = FALSE, alpha, data = 'real')
      a          <-events$start
      b          <-events$end
      #plot all events for this average
      n=1
      for (n in 1:length(a)){
        lines(c((a[n]*average):(b[n]*average)),
          rep(i+1,length.out=length((a[n]*average):(b[n]*average))),
          col = colour[i+1],lwd=3)
      }
    }
    else {i=i+1}
  }
}
```

Algorithm 3: Overlap between events

Description

Merge overlapping events

Arguments

events: events from event detection step

Value

events: events without overlap

```
events.overlap<-function(events){
  s<-events$start
  e<-events$end
  i=1
  foreach (i = 1:(length(s)-1))%do%{
    if(i==length(s)){stop()}
    if(e[i] >= s[i+1]){
      repeat{
        e[i]<-e[i+1]+1
        rem.s <- s[i+1]
        re.e <- e[i+1]
        s<-setdiff(s, rem.s) #remove start point
        e<-setdiff(e, rem.e) #remove end point
        e[i]<-e[i]-1
        print(paste0("Elements were deleted due to overlap"))
        if(e[i] < s[i+1]){break}
      }
    }
  }
  result <- list(events=events)
  return(events)
}
```

Algorithm 4: Percentage

Description

Calculate how many percentage of the full time series are detected as events

Arguments

events: events from event detection step

a: vector of averaged time series

Value

percentage: number

```
percentage<-function(events, a){
  sums=0
  i=1
  foreach (i = 1:length(events$start))%do%{
    sums=sums+length(events$start[i]:events$end[i])
  }
  percentage=(sums/length(a))*100
  result <- list(percentage=percentage)
  return(result)
}
```

Algorithm 5: Plot Events

Description

Plot events and save them in a predefined folder. Note this function is different to the `plotevents` function from TED package because the one from the TED package colours the detected events in the full time series. This function plots each event separately.

Arguments

`events`: events from event detection step

`mean`: averaged time series

`cluster`: cluster number

`sonic`: sonic number

Value

plot of the different events

```
plotting.events<-function(events, a,cluster,sonic){
  s<-events$start
  e<-events$end
  ma_events<-matrix(NA, length(s), (max(e-s)+1))
  n=1
  #fill matrix with events
  foreach (n =1:length(s))%do%{
    ma_events[n,1:(length(s[n]:e[n]))]<-a[s[n]:e[n]]
  }
  k=1
  mypath <-
    file.path("path",paste("DataCluster",cluster,"Sonic",sonic,".jpg",
    sep = ""))
  jpeg(file=myspath)
  #change plot window so that all plots fit in one
  par(mfrow=c(ceiling(nrow(ma_events))/2),2))
  foreach (k =1:nrow(ma_events))%do%{
    #plot each event
    plot(ma_events[k,], type = "l", ylab="", main=paste0("Event
    number ",k))
  }
  dev.off()
}
```

Example:

In this example the predefined functions are applied to make clear how they work. We analyse the temperature data from cluster 1 sonic 5. First we look at the scales from 2 mins to 30 mins and then we focus on the 6s (equals 120 points) averaged temperature data.

```
#attach data
attach(DataCluster1$V4)
x<-DataCluster1$V4
#time window length
w<-120
#significance level
alpha<-0.05
#find best scale
multiscale(x,20,alpha,14,w,20)
#average time series
av<-average.events(x,120)
#detect events
events<-eventDetection(av, w, noiseType = 'red',parallel = FALSE,
  alpha, data = 'real')
#merge overlapping events
events<-events.overlap(events)
#calculate percentage
percentage(events,av)
#plot events
plotting.events(events,av,1,5)
```
