TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DIPLOMA THESIS

Change Point Detection in Time Dependent Defect Spectroscopy Data

Performed at the

Institute for Microelectronics
at the Vienna University of Technology

Supervised by
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Tibor Grasser
and
Dipl.-Ing. Paul-Jürgen Wagner

by

Michael Waltl

Athalerstrasse 24
5122 Ach

Vienna, on November 10, 2011

Signature

Acknowledgment

First of all I want to thank the supervisor of my thesis Ao.Univ.Prof. Tibor Grasser. He gave me the opportunity to perform this work at the Institute of Microelectronics.

Many thanks are due to Dipl.-Ing. Paul-Jürgen Wagner for guiding me through this work. He has always been open for any ideas and questions as well as helping me with the documentation, especially correcting it. Through developing the algorithm he gave me very interesting and useful hints.

Last but not least I want to dedicate this piece of work to my parents Michaela and Ernst. Special thanks go to them, to my sister Magdalena and to my brother Bernhard for their emotional, intellectual and financial support, which helped me during the academic studies.

Abstract

One of the most important reliability effects observed in metal-oxide-semiconductor field-effect transistors (MOSFETs) is the threshold voltage shift when the device is stressed at high gate voltages at elevated temperatures, called the bias temperature instability (BTI). In the case of p-type MOSFETs the devices are stressed with negative gate voltages and the effect is therefore called negative bias temperature instability (NBTI). Positive bias temperature instability (PBTI) goes hand in hand with a positive gate voltage stress. PBTI is observed in n-type MOSFETs but is typically weaker than NBTI in p-type devices. In order to study BTI, a new method, the time dependent defect spectroscopy (TDDS), has been recently introduced. Very fast data acquisition equipment is necessary to obtain measurement data for this method. Evaluation and visualization of measurement results is done with the help of the detection of change points in TDDS measurement data. Three methods are presented and their results are discussed.

First a very fast and intuitive method based on the wavelet denoising techniques is presented. The application of the discrete wavelet transform (DWT) or the redundant discrete wavelet transform (RDWT) on the measurement data gives the wavelet transform coefficients. Smoothed transform coefficients are obtained by thresholding. Then the inverse transform gives a denoised version of the observations. A final filtering process extracts steps and emission times. Several wavelet basis functions, threshold denoising methods and their impact on detection results are discussed.

The second detection algorithm is based on histogram data analysis. Data is binned into histograms. After detecting peaks in the histograms, Gaussian distributions are fit to the peaks, and the denoised data is obtained by applying a maximum likelihood criterion. Histogram quality improvement by data interpolation and a histogram for not equally sampled data is presented.

The third and last method is a combination of cumulative sum statistic and bootstrap analysis combined with histogram supported evaluation. A threshold for change points is estimated and executed on the data set. Finally, a user defined minimal-acceptable step height filter is applied.

Contents

Chapter 1

Introduction

The bias temperature instability affects mainly metal-oxide-semiconductor field effect transistors (MOSFETs). The highest susceptibility show p-MOSFETs stressed with negative gate voltages at elevated temperatures up to 200°C. In the last few years this has become a serious reliability issue in silicon integrated circuits.

The degradation because of negative bias temperature instability (NBTI) has been known since 1966 [Miura and Matukura 1966]. Scaling of microelectronics devices leads to increasing oxide fields and increasing chip temperature. Also the replacement of buried p-channels by surface MOSFETs and the nitridation of oxides leads to increasing NBTI.

Before discussing NBTI, charge traps are discussed. Some basic concepts of the MOSFET and the impact of charge traps on threshold voltage is reviewed. Then the time dependent defect spectroscopy (TDDS) [Grasser et al. 2010] and the data acquisition is presented.

1.1 Charge Trap Centers

Two types of charge traps are speculated to be related to NBTI: The interface states at the Si/SiO_2 interface are the so called P_b centers. Also the creation of oxide charges is a reason for device degradation. Oxide charges are defects of the E' center family.

1.1.1 Interface states

The structural model of the Si/SiO_2 interface is shown in Figure 1.1. Silicon atoms have four valence electrons. In the silicon bulk material each silicon atom establishes four bonds to its neighboring silicon atoms. At the SiO_2 interface oxygen atoms surround a silicon atom. A defect free Si/SiO_2 interface is built by bonding silicon atoms with oxide atoms. At a real Si/SiO_2 interface not all silicon atoms bonds are sated because of the lattice mismatch of those two materials. After oxidation most interface states are saturated with oxygen atoms. The unsaturated bonds at the SiO_2 interface build the interface traps [Helms and Poindexter 1994].

Electrical active unsaturated bonds are called dangling bonds and are illustrated in Figure 1.1 for (111) and (100) orientated silicon interfaces. The interface states are known as the so called P_b centers in (111) and the P_b0 and P_b1 centers in (100) orientated Si/SiO_2 interfaces [Campell et al. 2007].

The P_b and P_b0 silicon dangling bonds are essentially identically. P_b and P_b0 centers are of amphoteric nature [Lenahan and Conley 1998] and show a broad density of states D_it below and above the middle of the silicon band gap.

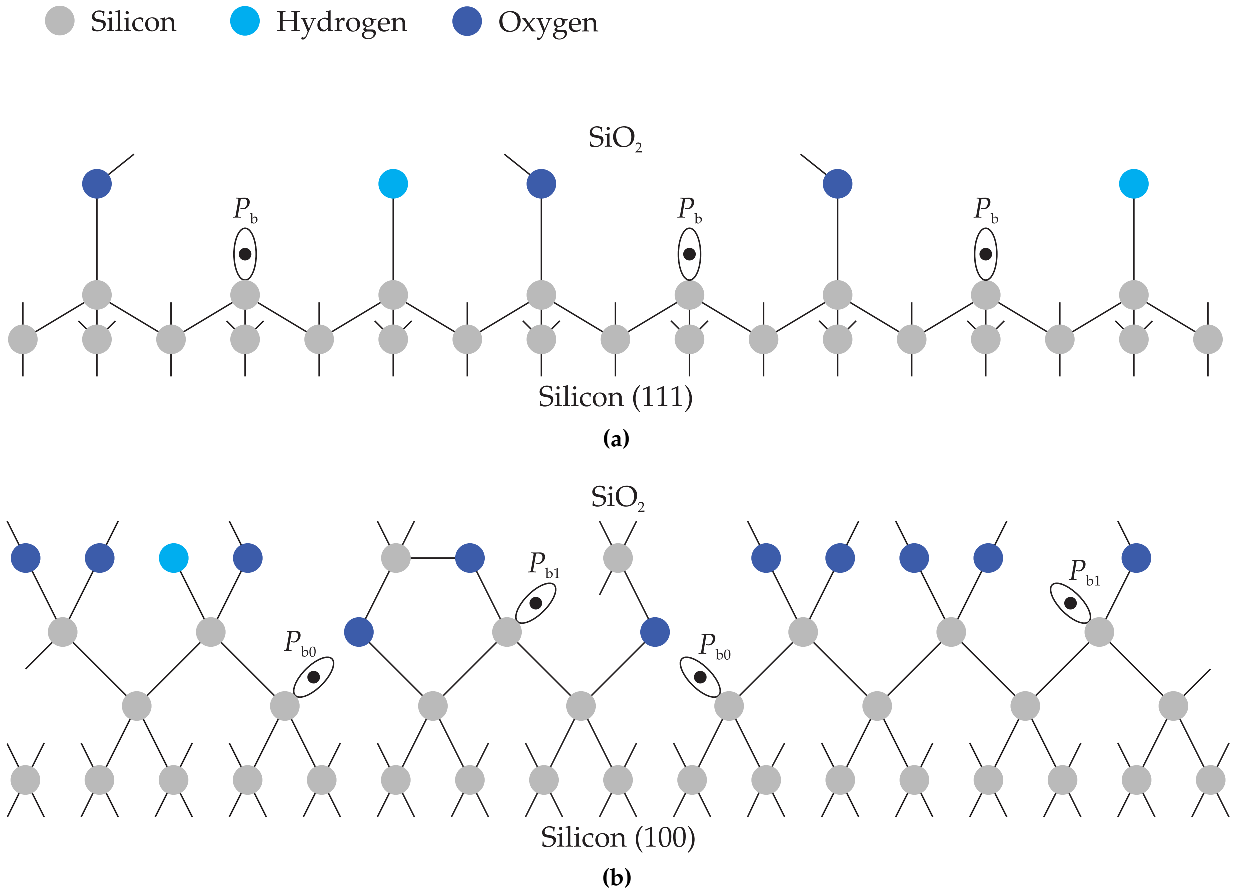


Figure 1.1: Atomic configuration for Si/SiO$_2$ interface in **(a)** (111) silicon with P_b trap centers and the interface in **(b)** (100) silicon with P_b0 and P_b1 trap centers [Helms and Poindexter 1994].

Near the valence band the P_b0 center is a donor like interface state. The two charge states can than be written as [Lenahan and Conley 1998]

$$P_\mathrm{b}^{+} + e^{-} = P_\mathrm{b} \qquad \text{and} \qquad P_\mathrm{b} + h^{+} = P_\mathrm{b}^{+}. \tag{1.1}$$

Interface states located in the upper half of the band gap are acceptor like, showing the charge state reactions

$$P_\mathrm{b} + e^{-} = P_\mathrm{b}^{-} \qquad \text{and} \qquad P_\mathrm{b}^{-} + h^{+} = P_\mathrm{b}. \tag{1.2}$$

The P_b1 centers show two peaks in their density of states D_it directly below the mid-gap separated only by a small gap. They likely have the same amphoteric nature as P_b and P_b0 centers.

1.1.2 Oxide Defects

Oxide trapping centers in nitrided SiO$_2$ are most likely E′ centers. They involve an unpaired electron of an silicon atom back-bonded to oxygen atoms in the oxide [Campell et al. 2007] . The most important one in the context of NBTI is the E'_γ center [Grasser et al. 2009a].The E'_γ center is created by hole trapping in the precursor structure, a Si-Si dimer inside the oxide. After breaking the Si-Si dimer bond, the E'_γ center transforms into a new equilibrium position. This results in a structural relaxation. E'_γ centers can be repeatedly charged and discharged. After an electron is captured the bond between the two silicon atoms does not immediately relax to its initial precursor state. They remain a while in a dipole state where an electron can be emitted very easily again. The idea that E' centers act like switching trap led to the formulation of the Harry-Diamond-Laboratories (HDL) model, cf. Figure 1.2. E' centers are considered as donor like defects. After remaining in an electric neutral state they are able to relax to their dimer configuration and anneal completely [Grasser et al. 2009a;b].

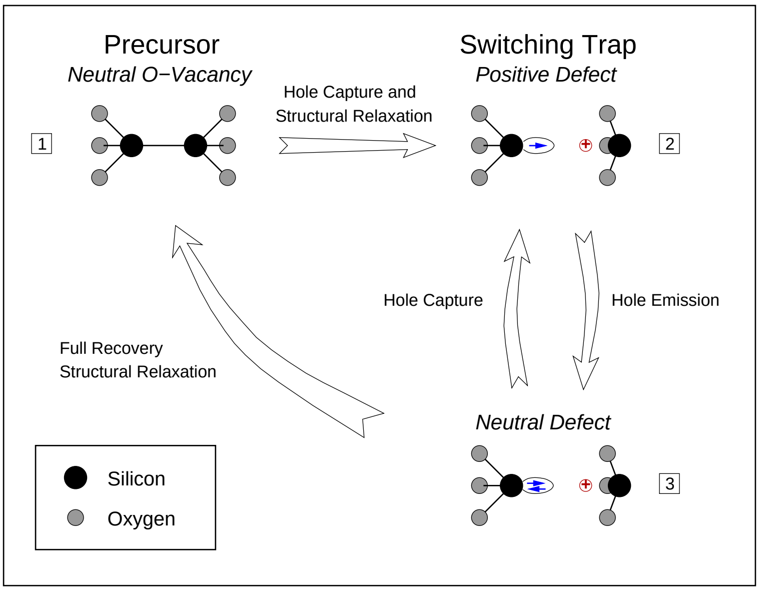


Figure 1.2: The HDL model: A neutral Si-Si dimer (state 1) can switch to a positive defect (state 2) by capturing a hole. A positive E'_γ center is created and can be neutralized by hole emission (state 3). From state 3 the neutral defect can either capture a hole again, leading to state 2, or the defect can completely anneal.

1.2 Negative Bias Temperature Instability

To understand the models of NBTI some basic MOSFET properties such as threshold voltage, interface trapped charge and oxide trapped charge are reviewed. As NBTI is most important the concepts are discussed based on a p-MOSFET. Finally NBTI models are presented. After the reaction-diffusion model, first published in the seventies, new ideas based on the Harry-Diamond-Labs (HDL) switching trap model are presented.

1.2.1 Basic MOSFET Properties

The MOS transistors threshold voltage is given by [Schroeder 2007]

$$V_\mathrm{T} = \phi_\mathrm{MS} - \frac{Q_\mathrm{ox}}{C_\mathrm{ox}} - \frac{Q_\mathrm{it}(\phi_\mathrm{S})}{C_\mathrm{ox}} - 2\phi_\mathrm{F} - \frac{Q_\mathrm{dep}}{C_\mathrm{ox}} \tag{1.3}$$

with the work function difference ϕ_MS describing the potential difference between the gate and substrate, the Fermi potential

$$\phi_\mathrm{F} = \frac{kT}{q} \ln \frac{N_\mathrm{D}}{n_\mathrm{i}}, \tag{1.4}$$

the fixed oxide charge per unit area Q_ox, the interface trapped charge per unit area Q_it, the depletion region charge per unit area $Q_\mathrm{diel} = (4qK_\mathrm{s}\epsilon_0\phi_\mathrm{F}N_\mathrm{D})^{1/2}$ and the oxide capacitance per unit area C_ox. The threshold voltage is the gate voltage leading to strong inversion, i.e. for the surface potential $\phi_\mathrm{S} = 2\phi_\mathrm{F}$.

Assuming that neither substrate doping density nor oxide thickness are changed during stress the threshold voltage ΔV_T shift only depends on changes of the oxide charge and the interface trapped charge [Schroeder 2007]

$$\Delta V_\mathrm{T} = -\frac{\Delta Q_\mathrm{ox} + \Delta Q_\mathrm{it}}{C_\mathrm{ox}}. \tag{1.5}$$

In contrast to doping atoms, interface traps act like acceptors above and like donors below the Fermi level. Therefore in p-type MOSFETs in inversion a positive interface trap charge lead to negative threshold voltage shifts. This is also the reason for the different effect in p-channel and n-channel devices. The interface trap charge is negative in n-type and positive in p-type devices. For constant oxide charge Q_ox for n-channel MOSFETs

$$\Delta V_\mathrm{T,n} = -\frac{\Delta Q_\mathrm{ox}}{C_\mathrm{ox}} + \frac{\Delta Q_\mathrm{it}}{C_\mathrm{ox}} \tag{1.6}$$

a positive but small threshold shift is observed, while for p-channel MOSFETs the contributions of Q_ox and Q_it have the same sign,

$$\Delta V_\mathrm{T,p} = -\frac{\Delta Q_\mathrm{ox}}{C_\mathrm{ox}} - \frac{\Delta Q_\mathrm{it}}{C_\mathrm{ox}} \tag{1.7}$$

and a larger negative threshold voltage shift is obtained.
In inversion the MOSFET operates like an ideal current source with drain current

$$I_\mathrm{D} = \frac{W}{2L}\mu C_\mathrm{ox}(V_\mathrm{GS} - V_\mathrm{T})^2 \tag{1.8}$$

and transconductance

$$g_\mathrm{m} = \frac{\partial I_D}{\partial V_\mathrm{T}} = \frac{W}{L}\mu C_\mathrm{ox}(V_\mathrm{GS} - V_\mathrm{T}). \tag{1.9}$$

Parameters leading to degradation of the drain current I_D and the transconductance g_m are threshold voltage shifts and mobility changes. The first ones result from charge trapping at the interface and in the oxide. Mobility changes are a consequence of scattering at interface traps.

1.2.2 Reaction-Diffusion Model

The first published model for NBTI is the reaction-diffusion theory introduced by Jeppson and Svensson [Jeppson and Svensson 1977] in 1977. It describes the NBTI as combination of two effects: At the Si/SiO$_2$ interface a large number of electrical inactive defects exists. Some of them can be electrical activated through an electrochemical reaction where the Si−H bonds get broken. A surface trap and a diffusing species X$_\mathrm{it}$ is build

$$\mathrm{Si{-}H} \rightleftharpoons \mathrm{Si}^+ + \mathrm{X}_\mathrm{it}. \tag{1.10}$$

The second part is the diffusion of the build hydrogen species X$_\mathrm{it}$ into the dielectric

$$\mathrm{X}_\mathrm{it} \xrightleftharpoons{\text{diffusion}} \mathrm{X}_\mathrm{diel}. \tag{1.11}$$

Also the reverse process is possible. After diffusion through the dielectric the hydrogen can re-bond with unsaturated silicon interface atoms, in turn re-passivating them.

Limitations of the reaction and diffusion models are mentioned in [Grasser et al. 2010]. The study of recovery of the degradation after stress show a recoverable and a permanent part. Also recovery starts immediately after stress release and can continue up to weeks. This is inconsistent to a pure diffusion process. The time dependent defect spectroscopy led to new insights into the mechanism behind NBTI.

1.2.3 Two-Stage Switching Trap Model

A two-stage model for NBTI was presented by [Grasser et al. 2009a]. In stage one during stress holes can be trapped and oxide trap centers are created. Poorly recoverable defects are involved in stage two. A recoverable process based on the HDL model for switching oxide traps [Lelis and Oldman 1994] and a permanent part with involved hydrogen passivated interface dangling bonds is illustrated in Figure 1.3. Starting with a neutral precursor in state 1 a hole capture and relaxation process leads to a trap center, most likely an E' center. Now two scenarios are possible. Either a reaction with a hydrogen atom from the passivated interface silicon P_b/H complex is possible, resulting in model state 4. Annealing of the E′ center is only possible from state 2. A hole emission process puts the system in state 3 where after structural relaxation a complete recovery from the stress is obtained.

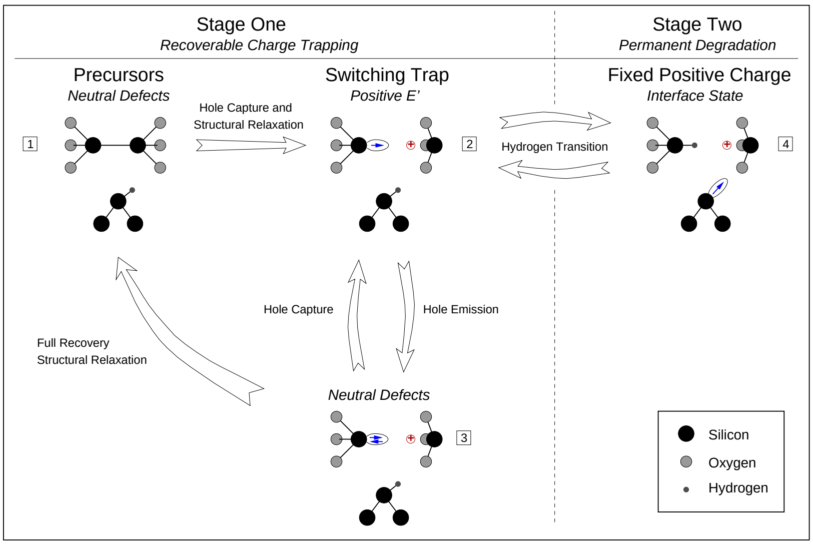


Figure 1.3: The two-stage model based on the HDL hole trap model for the recovery charge trapping and the link to interface states as permanent degradation [Grasser et al. 2009a].

The transitions between the four states are described with rate equations by

$$\begin{aligned}\frac{\partial f_1}{\partial t} &= -f_1 k_{12} + f_3 k_{31},\\ \frac{\partial f_2}{\partial t} &= +f_1 k_{12} - f_2 k_{23} + f_3 k_{32} - f_2 k_{24} + f_4 k_{42},\\ \frac{\partial f_3}{\partial t} &= +f_2 k_{23} - f_3 k_{32} - f_3 k_{31},\\ \frac{\partial f_4}{\partial t} &= +f_2 k_{24} - f_4 k_{42},\end{aligned} \tag{1.12}$$

with f_i the probability of being in state i and the transition rates from state i to state j given by k_{ij}. The four probabilities must fulfill

$$f_1 + f_2 + f_3 + f_4 = 0 \tag{1.13}$$

For a more detailed information on the two-stage switching trap model, especially the parameters and formulas related to the transition rates k_{ij}, see [Grasser et al. 2009a].

1.3 Time Dependent Defect Spectroscopy

To obtain measurement results for NBTI studies appropriate measurement setups are necessary. The main requirement for all methods is a very short delay between the stress and measuring operation. At this time three fast measurement methods are used [Reisinger et al. 2007b]:

- On the fly measurements do not interrupt stress during relaxation measurement. The electrical parameters are determined at or very close to the stress level.
- Interrupting stress and measuring the threshold voltage V_T is done at the *fast* V_T method. With achieved measurement delays of about 0.5μs the threshold voltage is recorded versus the recovery time.
- Another method is measuring of the drain current I_D at a gate voltage very near the threshold voltage. The measurement delay at the *fast* I_D method is determined by rise and fall times of the voltage sources, and this time nearly 1ms.

At the time dependent defect spectroscopy a measurement setup for *fast* V_T measurements is used. After presenting the measurement setup results and evaluation of the time dependent defect spectroscopy are discussed.

1.3.1 Measurement Setup

The measurement setup for the time dependent defect spectroscopy, a fast, method for measurement of relaxation after stress is presented in [Reisinger et al. 2007a]. The goal is to measure the threshold voltage as fast as possible at the desired operating condition. In order to obtain results undistorted by recovery a very short delay between the stress and measure phase is necessary. The experimental setup is divided into the stress and the acquisition part illustrated in Figure 1.4. First, with the switches in position stress, a voltage is applied to the gate of the MOSFET under test. For p-MOSFETS the drain contact is held at ground potential. For NBTI measurements

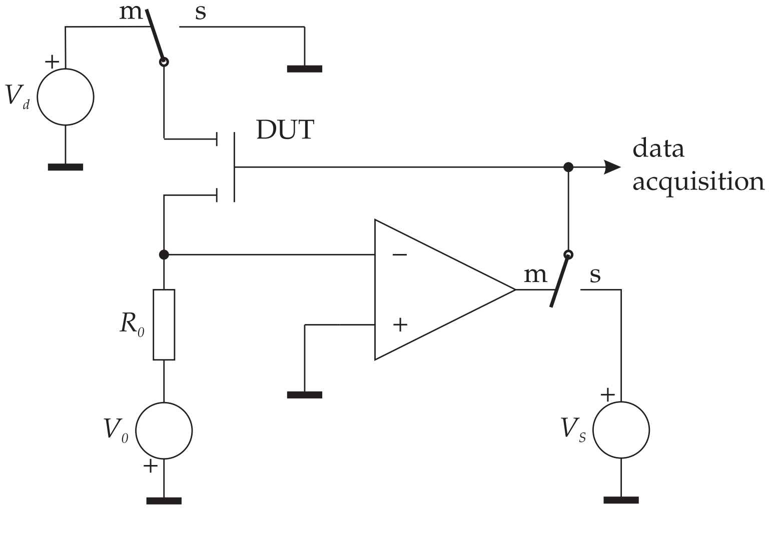


Figure 1.4: Experimental setup for the fast V_T method: In position 'm' the drain current equals V_0/R_0. If this value is set to a constant value, for example $V_0/R_0 = 70\mathrm{nA} \cdot W/L$, the gate voltage is about the threshold voltage for this single device. In switch position 's' the stress voltage V_S is applied to the gate [Reisinger et al. 2007a].

the drain-source voltage is defined to be nearly zero $V_{\mathrm{DS}} \approx 0$. After the stress time elapsed the switch position is put into the measure state, thereby closing a feedback loop built around an operational amplifier. This circuit establishes a V_{GS} such that a preset I_D is flowing. A current difference between the drain current I_{D} and a set-point current is zeroed by the operational amplifier. An appropriate gate-source voltage V_{GS} is therefore applied by the operational amplifier. The gate-source voltage V_{GS} controlled by the operational amplifier is then recorded, and any changes to it are attributed to a change in the threshold voltage V_{T}. The great advantage of this method is that V_{T} is ready to be monitored after the feedback loop setup time, which is about 0.5μs. So very fast events can be observed.

1.3.2 Evaluation of Measurement Data

Typically NBTI studies are made on large-area devices. An ensemble of defects are observed simultaneously. The time dependent defect spectroscopy [Grasser et al. 2010] uses small-area devices, so individual defects constituting the microscopic behavior are observed. In order to gain sufficient statistics, several stress and relaxation measurements of the device are taken, for example 100 measurements at one device.

The idea of the time dependent defect spectroscopy is very similar to the deep level transient spectroscopy [Lang 1974]. It is widely used to detect so called deep traps in the band gap.

For the time dependent defect spectroscopy several stress and recovery processes with the same device have to be performed. Relaxation of the stress devices in a step-like manner leads to a characteristic step amplitude associated which individual defects. The statistical analysis evaluates step heights d and emission times τ_{e}, time elapsed since stress released, of each step in measurement data. An example for a voltage recovery trace is shown in Figure 1.5. These pairs (τ_{e}, d) are processed into 2D-histograms called spectral maps. Typical spectral maps are shown in Figure 1.6 for different stress times t_{S}. Each emission of a particular defect form a cluster in the spectral map. The positions of the clusters in the spectral maps do not change with stress time, while step heights show slight variations. For increasing temperatures the same clusters

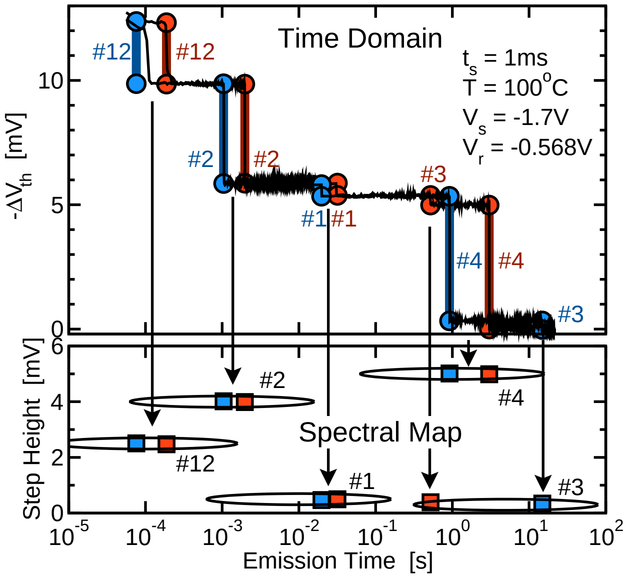
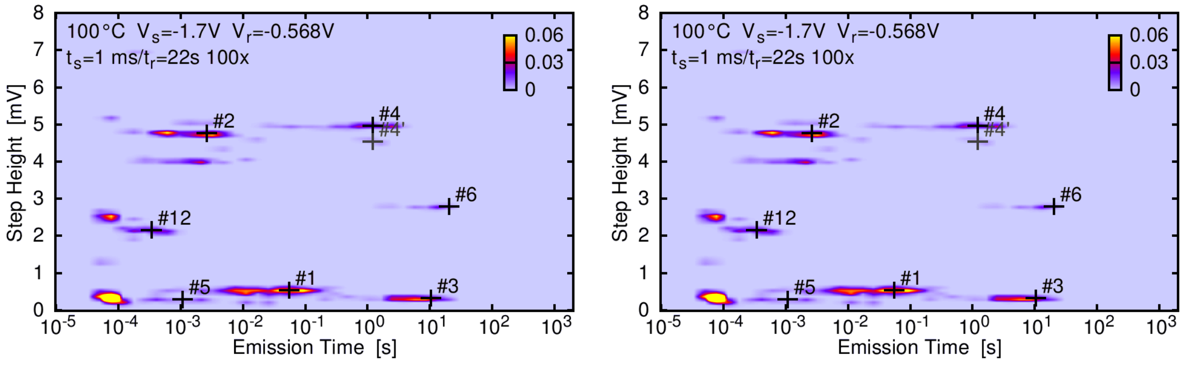


Figure 1.5: Two typical threshold voltage recovery signals of a previously stressed p-MOSFET. The thin blacks lines in the top figure are the measurement data. The thick blue and red lines mark the extracted step height and emission times [Grasser et al. 2010].



(a)

(b)

Figure 1.6: Spectral maps as a result of step height and emission time extraction from time dependent defect spectroscopy measurements at **(a)** stress time $t_S = 1ms$ and **(b)** stress time $t_S = 10s$. With increasing stress time the number of defects increase [Grasser et al. 2010].

are obtained but at shorter emission times. So the underlying process is thermally activated [Grasser et al. 2010].

With the time dependent defect spectroscopy measurement data the recoverable degradation of the devices are studied, cf. Figure 1.3. The analysis of the permanent degradation is done by comparing the threshold voltage before the device is stressed and after the relaxation duration. The main part of this thesis deals with the detection of change points in time dependent defect spectroscopy measurement data. Detected step heights and times are extracted to create spectral maps. Three methods for step detection are presented in the next chapters.

CHAPTER 2

WAVELETS AND CHANGE POINT ANALYSIS

Signal transformation in a mathematical manner is an approximation of a signal by specialized functions. In the case of the Fourier transform sinusoidal functions are used. The wavelet transform is an approximation with basis functions called wavelets. Since the introduction of Daubechies Wavelets [Daubechies 2006] the importance of the wavelet transform rapidly increased. Its biggest application fields are signal processing and picture compression.

The use of wavelets for change point detection is presented now. After that an overview of signal approximation is given using the continuous wavelet transform and the discrete wavelet transform. A computer based method for calculating the transform coefficients is the multiresolution analysis which leads to the fast wavelet transform. A very redundant type of transformation, of high interest with signal denoising, is the stationary discrete wavelet transform also in the literature often called redundant discrete wavelet transform.

Finally the change point detection algorithm, with the possibilities to configure the change point detection sensitivity, and the results are presented.

2.1 Continuous Wavelet Transform

The frequency composition of a time signal $r(t)$ is obtained by the Fourier transform [Moon and Stirling 2000]

$$\mathcal{F}\{r(t)\} = R(\mathrm{j}\omega) = \int_{-\infty}^{\infty} r(t)e^{-\mathrm{j}\omega t}\mathrm{d}t. \tag{2.1}$$

From its spectrum $R(\mathrm{j}\omega)$ it is very difficult to infer time localization of the signal. One method is to use a windowed version of the Fourier transform called the short time Fourier transform [Smith 2008]

$$\mathcal{F}\{r(t), t_0\} = \int_{-\infty}^{\infty} r(t)g(t-t_0)e^{-\mathrm{j}\omega t}\mathrm{d}t. \tag{2.2}$$

Instead of calculating the complete transform the time signal is split into sections by a window function $g(t)$, e.g. a Gaussian curve for the Gabor transformation [Feichtinger and Strohmer 1998].

A time frequency description, called the analysis of $r(t)$, is obtained by the continuous wavelet transform [Blatter 1998]

$$\mathcal{W}_{\psi_{a,b}}\{r(t)\} = \langle r(t), \psi_{a,b}(t)\rangle = |a|^{-1/2}\int_{-\infty}^{\infty} r(t)\psi^*\left(\frac{t-b}{a}\right)\mathrm{d}t, \tag{2.3}$$

with the wavelet function $\psi_{a,b}(t)$ and the inner product of the $L^2(\mathbb{R})$ space defined by

$$\langle f(t), g(t) \rangle = \int_{-\infty}^{\infty} f(t) g^*(t) \mathrm{d}t. \tag{2.4}$$

The synthesis, or inverse continuous wavelet transform, is a linear superposition of the wavelet functions

$$r(t) = \int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{a^2} \mathcal{W}_{\psi_{a,b}}\{r(t)\} \psi_{a,b}(t) \mathrm{d}b \mathrm{d}a. \tag{2.5}$$

So the wavelet transform is a function of the parameters $(a, b) \in \mathbb{R}_+ \times \mathbb{R}$ with $\mathcal{W}_{\psi_{a,b}}\{r(t)\} : \mathbb{R}_+ \times \mathbb{R} \longrightarrow \mathbb{C}$. The function

$$\psi_{a,b}(t) = |a|^{-1/2} \psi \left(\frac{t-b}{a} \right), \tag{2.6}$$

a scaled and shifted version of a mother wavelet $\psi(t)$, is called wavelet. Its time localization can be moved by the choice of b. So $\psi_{a,b}(t)$ is centered around $t = b$ analogues to the window function in the context with the short time Fourier transform. The main difference is that the wavelet transform is better suited for short time signals, for example transients, than the Fourier transform.

Also very important is that in (2.3) the wavelet is not restricted to a specific one. The theory of wavelet transform deals with the properties of the wavelets. Based on these properties specific wavelets can be created.

All wavelets $\psi(t) \in L^2(\mathbb{R})$ fulfill the admissibility condition [Blatter 1998]

$$\int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} \mathrm{d}\omega < \infty \tag{2.7}$$

for their spectrum $\Psi(\omega)$. This condition is necessary to reconstruct the signal without loss of information. Property (2.7) implies that the Fourier transform of wavelets vanish at zero frequency, cf. the wavelet function illustrated in Figure 2.1,

$$|\Psi(\omega)|^2_{\omega=0} = 0, \tag{2.8}$$

which equals zero mean in the time domain

$$\int_{-\infty}^{\infty} \psi(t) \mathrm{d}t = 0. \tag{2.9}$$

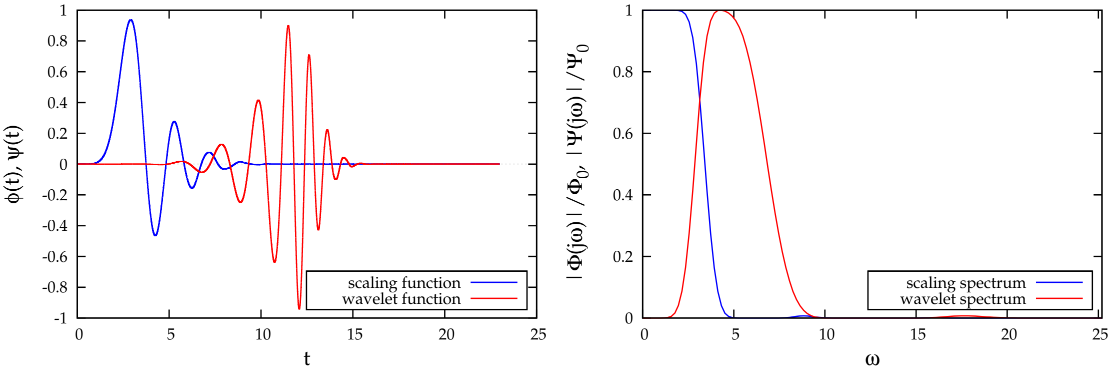
To have zero mean a function has to be less than zero and greater than zero at different time instances like the portrait of a wave. The continuous wavelet transform discussed so far has still properties that make it unpractical to use:

1. The calculation of the coefficients with the integral formula and the use of a scaled and shifted function deliver high redundancy in the calculated coefficients. This is not requested for many applications.

2. The number of possible wavelets is uncountable because of $(a, b) \in \mathbb{R}_+ \times \mathbb{R}$. A finite number of wavelet functions is preferred.

3. For most wavelets there is no analytical solution of the wavelet transform integral. It has to be solved numerically which is slow.

To handle these requirements the discrete wavelet transform is introduced next.



(a) **(b)**

Figure 2.1: Daubechies wavelet $m = 12$ **(a)** scaling and wavelet function(left) and **(b)** their spectrum normed to $\max(|\Phi(\mathrm{j}\omega)|) = 1)$ and $\max(|\Psi(\mathrm{j}\omega)|) = 1)$.

2.2 Discretization of the Wavelet Transform

The discrete wavelet transform restricts the parameters (a, b) to discrete values only. Setting the scaling parameter [Blatter 1998] to

$$a_j := \alpha^j \quad \text{with} \quad \alpha > 1 \tag{2.10}$$

and the shift parameter to

$$b_{j,k} := k a^j \beta \quad \text{with} \quad \beta > 0 \tag{2.11}$$

leads to a countable set

$$\{(a_j, b_{j,k}) | \quad j, k \in \mathbb{Z}\}. \tag{2.12}$$

From the full set of wavelet functions $\psi_{a,b}(t)$ only the ones belonging to the discrete points $(a_j, b_{j,k})$ are then taken into account. Therefore the discrete wavelet functions

$$\psi_{j,k}(t) = \alpha^{-j/2}\psi(\alpha^{-j}t - k\beta) \tag{2.13}$$

are obtained. Popular choices [Blatter 1998] are $\alpha = 2$ and $\beta = 1$. The so obtained set of wavelets, which is used from now on, is given by

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k). \tag{2.14}$$

Using the L_2 norm

$$\|f(t)\|_2 = \sqrt{\int |f(t)|^2 \mathrm{d}t} \tag{2.15}$$

the wavelet energy lies between two bounds [Daubechies 2006]

$$A\|r(t)\|_2^2 \leq \sum_{j,k} |\langle r(t), \psi_{j,k}(t)\rangle|^2 \leq B\|r(t)\|_2^2 \tag{2.16}$$

with the signal energy $\|r(t)\|_2^2$ and the energy of the transform coefficients. For stable reconstruction $A > 0$, $B < \infty$ and A,B independent of $r(t)$ is necessary. In the case of $A = B$ discrete

wavelets behave like an orthonormal basis. If $A \neq B$ the reconstruction is still possible but there are two different wavelet functions for decomposition and reconstruction.
The optimal choice for wavelet functions is to make them orthonormal

$$\langle \psi_{j,k}(t), \psi_{m,n}(t) \rangle = \delta_{j-m}\delta_{k-n}, \tag{2.17}$$

where δ_k denotes the Kronecker delta.
The coefficients of the discrete transform are given by

$$c^{(j)}[k] = \langle r(t), \psi_{j,k}(t) \rangle \tag{2.18}$$

with the inverse discrete wavelet transform as linear superposition of the corresponding wavelet functions

$$r(t) = \sum_{j,k} c^{(j)}[k]\psi_{j,k}(t). \tag{2.19}$$

On the way to generate a computer based algorithm for calculating the wavelet transform coefficients the multiresolution analysis is the basis for the fast wavelet transform and handled next.

2.3 Multiresolution Analysis

A multiresolution analysis [Vetterli and Kovačević 1995, Moon and Stirling 2000, Blatter 1998] consists of a sequence of subspaces in the space of all square integral able functions $L^2(\mathbb{R})$

$$\{0\} \subset \ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots \subset L^2(\mathbb{R}) \tag{2.20}$$

and satisfies several properties. First, the union of all these nested subspaces is the $L^2(\mathbb{R})$ itself

$$\bigcup_{j\in\mathbb{Z}} V_j = L^2(\mathbb{R}), \tag{2.21}$$

and their intersection is the zero space

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\}. \tag{2.22}$$

The multiresolutional part is a consequence of

$$r(t) \in V_j \Leftrightarrow r(2^j t) \in V_0. \tag{2.23}$$

Also necessary is the shift invariance

$$r(t) \in V_j \Leftrightarrow r(t-n) \in V_j \ \text{ for all } \ n \in \mathbb{Z}. \tag{2.24}$$

The final condition is that a function $\phi(t) \in V_0$ exists in such a way that all integer shifts build an orthogonal basis for V_0

$$V_0 = \text{span}\{\phi(t-n); \quad n \in \mathbb{Z}\}. \tag{2.25}$$

With the definition of a scaled and shifted version of a function $\phi(t)$

$$\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t-k) \tag{2.26}$$

for $\phi_{j,k}(t) \in L^2(\mathbb{R})$ a basis for a subspace V_j is given by

$$V_j = \text{span}\{\phi_{j,k}(t), k \in \mathbb{Z}\} = \text{span}\{2^{-j/2}\phi(2^{-j}t - k), k \in \mathbb{Z}\}. \tag{2.27}$$

The function $\phi(t)$ is called scaling function or father wavelet. In order to build an orthonormal basis the scaling functions must fulfill the shift orthogonality property

$$\langle \phi(t), \phi(t-k) \rangle = \delta_k. \tag{2.28}$$

Another wavelet basis can be constructed by extending the space at resolution 2^j in a way that the space at resolution 2^{j-1} is build. This wavelet builds the basis for another subspace W_j, the projection of the complement of V_j in V_{j-1}. That means

$$V_{j-1} = V_j \cup W_j \tag{2.29}$$

where W_j is the orthogonal complement of V_j

$$V_j = W_j^{\perp}. \tag{2.30}$$

The space W_j is spanned by functions $\psi_{j,k}(t)$ defined by

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k). \tag{2.31}$$

The function $\psi(t)$ is called mother wavelet or short wavelet. A summarized relationship between these subspaces is shown in Figure 2.2.

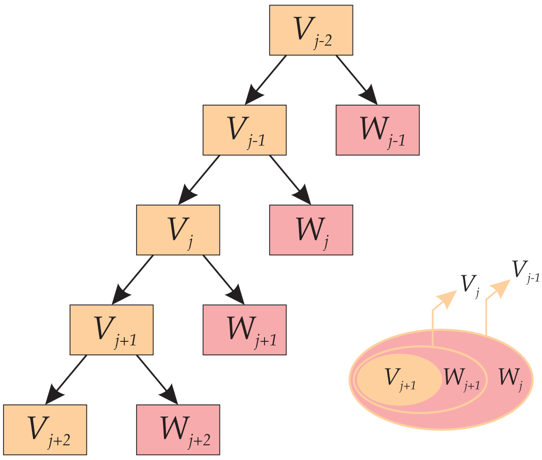


Figure 2.2: Relationship between nested subspaces V_j and their orthogonal complement W_j

Because of finer approximation steps resulting from $V_0 \subset V_{-1}$ the scaling function $\phi(t) \in V_0$ can be expressed by linear combination of $\phi_{-1,k}(t) \in V_{-1}$ together with (2.26) as

$$\phi(t) = \sum_k l[k]\phi_{-1,k}(t) = \sqrt{2}\sum_k l[k]\phi(2t-k). \tag{2.32}$$

This is the most important equation in context with the scaling function. Since W_0 is also a subset of V_{-1} there is a linear combination of $\phi_{-1,k}(t) \in V_{-1}$ for the wavelet function $\psi(t) \in W_0$ too. With (2.31) the linear combination is

$$\psi(t) = \sum_k h[k]\phi_{j-1,k}(t). \tag{2.33}$$

Because of the orthogonality of $\phi(t)$ and $\psi(t)$ the coefficient vectors $\mathbf{l}$ and $\mathbf{h}$ must also be orthogonal

$$\langle \mathbf{l}, \mathbf{h} \rangle = \mathbf{h}^H \mathbf{l} = 0, \tag{2.34}$$

from which

$$h[n] = (-1)^n l^*[N-n] \tag{2.35}$$

with the vector length N follows [Moon and Stirling 2000]. The second scale equation for wavelet functions is then obtained by

$$\psi(t) = \sum_k (-1)^k l^*[N-k] \phi(2t-k). \tag{2.36}$$

Based on the multiresolution analysis the fast wavelet transform is constructed now.

2.4 Fast Wavelet Transform

As shown in the multiresolution analysis $V_j = V_{j+1} \cup W_{j+1}$. A time signal $r(t)$ is expressed using the basis functions of each subspace by

$$\begin{aligned} r(t) &= \sum_k a_0[k] \phi_{j,k}(t) \\ &= \sum_k a_1[k] \phi_{j+1,k}(t) + \sum_k d_1[k] \psi_{j+1,k}(t) \end{aligned} \tag{2.37}$$

with $\mathbf{a}_0$, $\mathbf{a}_1$ and $\mathbf{d}_1$ the transform coefficients. Starting from $\mathbf{a}_0$ the coefficients $\mathbf{a}_1$ and $\mathbf{d}_1$ can be produced. This step is called analysis. The synthesis starts with $\mathbf{a}_1$ and $\mathbf{d}_1$ and produces $\mathbf{a}_0$.
The coefficients can be calculated by use of the inner product

$$\begin{aligned} a_1[n] &= \langle r(t), \phi_{j+1,n}(t) \rangle \\ &= \left\langle \sum_k a_0[k] \phi_{j,k}(t), \phi_{j+1,n}(t) \right\rangle \\ &= \sum_k a_0[k] \langle \phi_{j,k}(t), \phi_{j+1,n}(t) \rangle. \end{aligned} \tag{2.38}$$

The inner product of scaling functions [Moon and Stirling 2000] is

$$\langle \phi_{j,k}(t), \phi_{j+1,n}(t) \rangle = h_0[k-2n]. \tag{2.39}$$

For the coefficients

$$a_1[n] = \sum_k h_0[k-2n] a_0[k] \tag{2.40}$$

is obtained. The detailed coefficients calculation is very similar

$$\begin{aligned} d_1[n] &= \langle r(t), \psi_{j+1,n}(t) \rangle \\ &= \left\langle \sum_k a_0[k] \phi_{j,k}(t), \psi_{j+1,n}(t) \right\rangle \\ &= \sum_k a_0[k] \langle \phi_{j,k}(t), \psi_{j+1,n}(t) \rangle \end{aligned} \tag{2.41}$$

with the inner product

$$\langle \phi_{j,k}(t), \psi_{j+1,n}(t) \rangle = h_1[k-2n]. \tag{2.42}$$

So the detailed coefficients be

$$d_1[n] = \sum_k h_1[k-2n] a_0[k]. \tag{2.43}$$

Equations (2.40) and (2.43) are very similar to the convolution sum

$$(f * g)[n] = \sum_k f[n-k] g[k]. \tag{2.44}$$

Downsampling of discrete signals is done by taking each m-th value of $\mathbf{r}$. So downsampling by $m = 2$ can be interpreted as system with input $r[n]$ and output $y = r[2n]$ as shown in Figure 2.3.

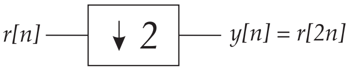


Figure 2.3: Schematic of downsampling by 2

To emphasize the relationship between the coefficients formula (2.40) and the convolution sum the formula (2.40) can be reformulated to

$$\begin{aligned} a_1[n] &= \sum_k h_0[k-2n] a_0[k] \\ &= \sum_k h_0[-(2n-k)] a_0[k] \\ &= (\hat{h}_0 * a_0)[2n] \end{aligned} \tag{2.45}$$

with $\hat{h}_0[n] = h_0[-n]$. Downsampling the convolution result from (2.45) results in transform coefficients $a_1[n]$. The same reformulation for (2.43) is

$$d_1[n] = (\hat{h}_1 * a_0)[2n] \tag{2.46}$$

with $\hat{h}_1[n] = h_1[-n]$. The calculation of the detailed transform coefficients $a_1[n]$ can be described by convolution of initial coefficients $a_1[n]$ with a filter $h_1[-n]$ and a following downsampling operation. A single transform filter bank containing the convolution and downsampling in illustrated in Figure 2.4.

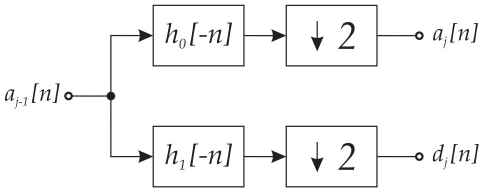


Figure 2.4: Single fast wavelet transform filter bank

The inverse fast wavelet transform follows from

$$\begin{aligned}
a_0[n] &= \langle r(t), \phi_{j,n}(t)\rangle \\
&= \left\langle \sum_k a_1[k]\phi_{j+1,k}(t) + \sum_k d_1[k]\psi_{j+1,k}(t), \phi_{j,n}(t)\right\rangle \\
&= \sum_k a_1[k]\langle \phi_{j+1,k}(t), \phi_{j,n}(t)\rangle + \sum_k d_1[k]\langle \psi_{j+1,k}(t), \phi_{j,n}(t)\rangle \\
&= \sum_k a_1[k]h_0[n-2k] + \sum_k d_1[k]h_1[n-2k].
\end{aligned} \tag{2.47}$$

Upsampling of a discrete time signal consists of inserting zeros between the values

$$y[n] = \begin{cases} r[n/2] & \text{for even } n \\ 0 & \text{else} \end{cases} . \tag{2.48}$$

The combination of (2.47), the upsampled versions of the transform coefficients $\tilde{a}_1[n] = a_1[2n]$ and $\tilde{d}_1[n] = d_1[2n]$ and the substitution of $l = 2k$ is

$$\begin{aligned}
a_0[n] &= \sum_k h_0[n-2k]a_1[k] + \sum_k h_1[n-2k]d_1[k] \\
&= \sum_l h_0[n-l]\tilde{a}_1[l] + \sum_l h_1[n-l]\tilde{d}_1[l] \qquad . \\
&= (h_0 * \tilde{a}_1)[n] + (h_1 * \tilde{d}_1)[n]
\end{aligned} \tag{2.49}$$

In (2.49) the time-forwarded filters $h_0[n]$ and $h_1[n]$ opposed to the analysis case are used. A single inverse filter bank, called the synthesis filter bank, is given in Figure 2.5.

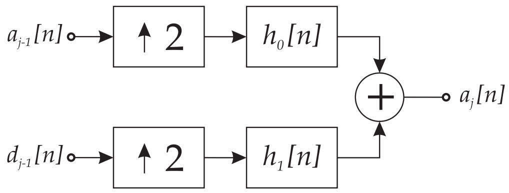


Figure 2.5: Single inverse fast wavelet transform filter bank

Further approximations steps in (2.37) lead, because of the subspaces nesting property, to

$$\begin{aligned}
r(t) &= \sum_k a_0[k]\phi_{j,k}(t) \\
&= \sum_k a_1[k]\phi_{j+1,k}(t) + \sum_k d_1[k]\psi_{j+1,k}(t) \\
&= \sum_k a_2[k]\phi_{j+2,k}(t) + \sum_k d_2[k]\psi_{j+2,k}(t) + \sum_k d_1[k]\psi_{j+1,k}(t)
\end{aligned} \tag{2.50}$$

which can be continued. Finally it leads to the fast wavelet transform analysis filter banks for the desired order, cf. Figure 2.6 for an analysis filter bank of level 3. For signal reconstruction the synthesis filter bank for level 3 is shown in Figure 2.7.

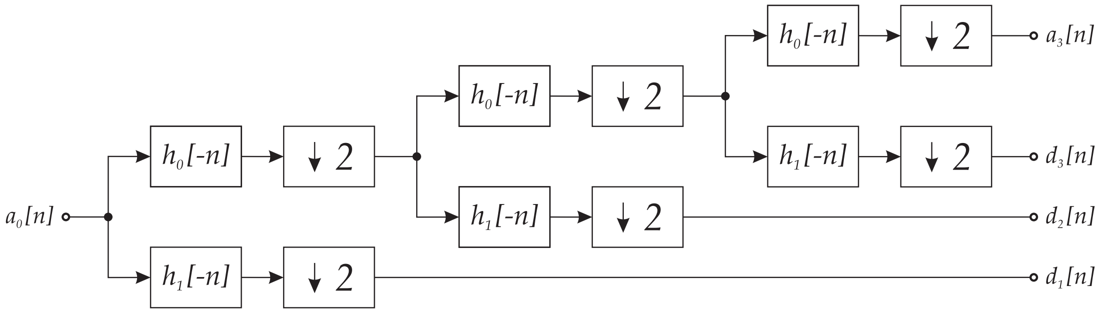


Figure 2.6: Fast wavelet transform analysis filter bank of level 3

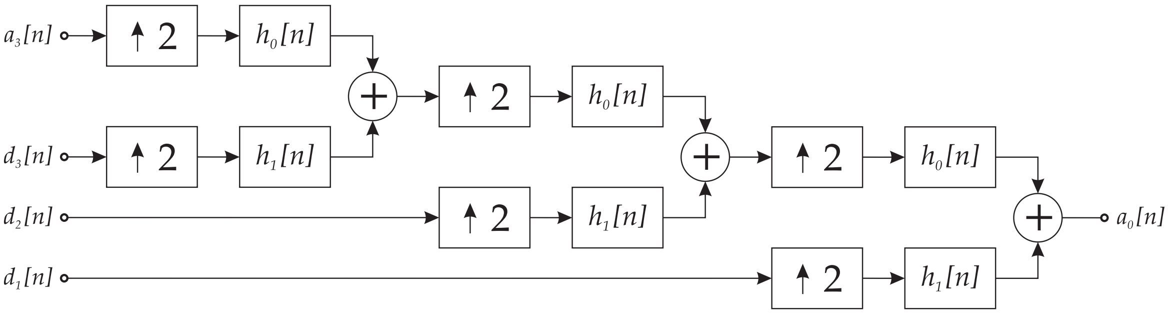


Figure 2.7: Fast wavelet transform synthesis filter bank of level 3

2.5 Stationary Wavelet Transform

The stationary discrete wavelet transform was initially developed by [Holschneider et al. 1989] as discrete approximation of the continuous wavelet transform. Other formulations [Shensa 1992] show that the discrete wavelet transform without downsampling has an overcomplete representation with shift invariance. In contrast, the shift variance of the discrete wavelet transform arises from downsampling. As a result, the transform coefficient vector of each scale is exactly the same length as the input signal.

The schematic of the stationary discrete wavelet transform analysis and synthesis of level 3 is illustrated in Figure 2.8.

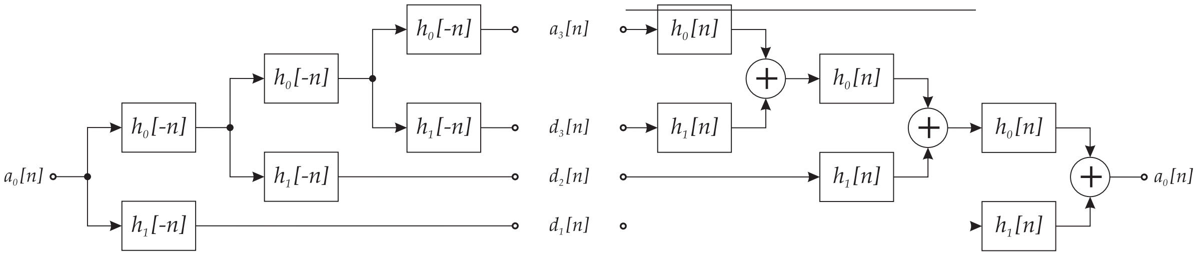


Figure 2.8: Stationary discrete wavelet transform analysis and synthesis filter bank level 3

The stationary discrete wavelet transform is not of interest for many applications because it results in highly redundant data, except for signal denoising. The most important wavelets are presented in the next section followed by the wavelet signal denoising technique.

2.6 Important Wavelets

Several wavelets have been designed so far. The most important are the Haar [Haar 1911], mexican hat, Morlet [Goupillaud et al. 1984], Meyer and the Daubechies [Daubechies 2006] wavelet. Of interest to us are the Haar and Daubechies wavelets.
The scaling function of the Haar wavelet is defined by

$$\phi(t) = \begin{cases} 1 & 0 \le t < 1 \\ 0 & \text{elsewhere} \end{cases} \tag{2.51}$$

and its mother wavelet function by

$$\psi(t) = \begin{cases} 1 & 0 \le t < 1/2 \\ -1 & 1/2 \le t < 1 \\ 0 & \text{elsewhere} \end{cases} \tag{2.52}$$

as illustrated in Figure 2.9. The filter coefficients for discrete wavelet transform calculation with filter bank structure from Figure 2.4 are

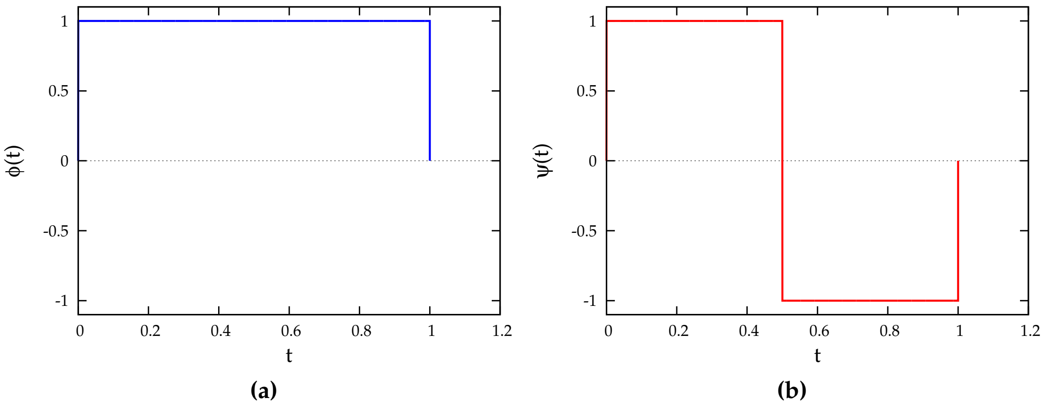


Figure 2.9: The Haar wavelets **(a)** scaling function $\phi(t)$ and **(b)** wavelet function $\psi(t)$

$$\begin{aligned} \mathbf{h}_0 &= \frac{\sqrt{2}}{2}[1,1] \\ \mathbf{h}_1 &= \frac{\sqrt{2}}{2}[1,-1] \end{aligned} \tag{2.53}$$

The Daubechies wavelets are chosen to have the highest number m of vanishing moments

$$\int_{-\infty}^{\infty} t^m \psi(t) \mathrm{d}t = 0. \tag{2.54}$$

For data evaluation the Daubechies wavelet with $m = 10$ is used, shown in Figure 2.10. The filter coefficients for the implementation of the fast wavelet transform and inverse fast wavelet transform filter banks, cf. Figure 2.4 and Figure 2.5, can be found in MATLAB.

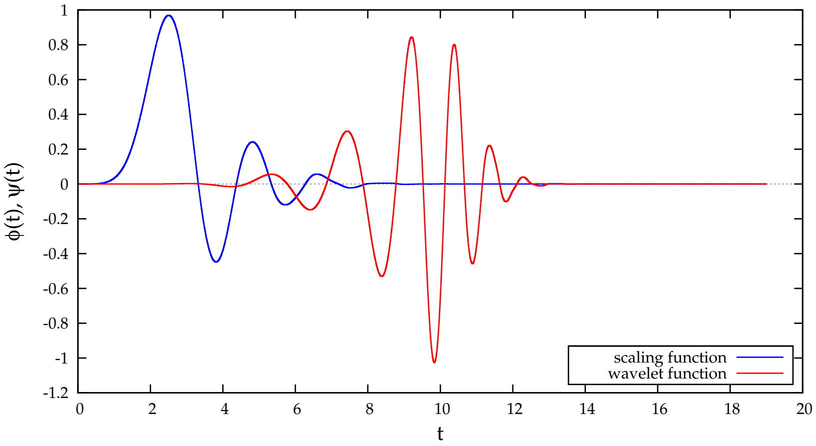


Figure 2.10: Scaling and wavelet function of Daubechies wavelet with $m = 10$

2.7 Signal Denoising

Several techniques are provided for signal denoising using the wavelet transform. A process overview is given in Figure 2.11. After applying the wavelet transform of J-th level the denoising threshold is calculated. Therefore four methods of threshold estimation are of interest.

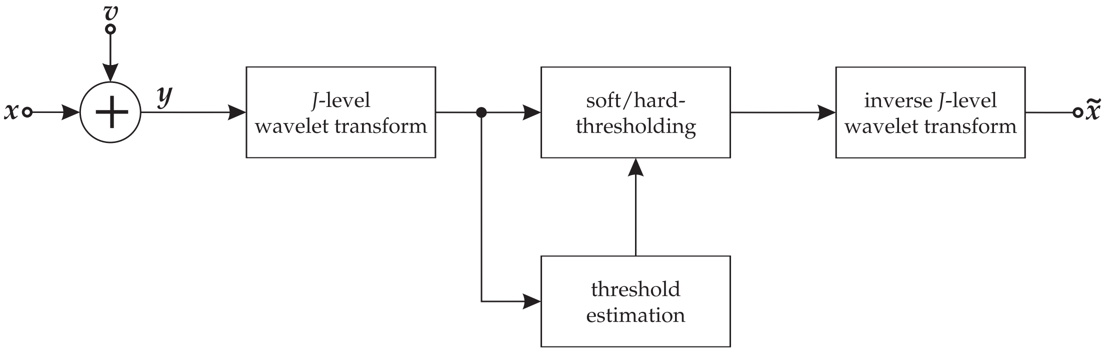

v
x
y
J-level wavelet transform
soft/hard-thresholding
inverse J-level wavelet transform
$\tilde{x}$
threshold estimation

Figure 2.11: Process flow of signal denoising using the wavelet transform and thresholding techniques

1. The Bayes shrink threshold estimation [Chang et al. 2000] based on the observation model

$$\mathbf{y} = \mathbf{x} + \mathbf{v} \tag{2.55}$$

where $\mathbf{y}$ is the vector of wavelet transform coefficients of the observations, $\mathbf{x}$ the wavelet transform vector of the original signal and $\mathbf{v}$ the wavelet transform vector of the noise. Since $\mathbf{x}$ and $\mathbf{v}$ are independent from each other their variances fulfill

$$\sigma_y^2 = \sigma_x^2 + \sigma^2. \tag{2.56}$$

The observations wavelet transform variance can be easily found by

$$\sigma_y^2 = \frac{1}{N} \sum_k (y_k - \mu_y)^2 \tag{2.57}$$

with μ_y the mean value of $\mathbf{y}$. The noise standard deviation σ is estimated by the robust median estimator [Elyasi and Zarmehi 2009]

$$\sigma = \frac{\text{median}|\mathbf{y}_H|}{0.6745} \tag{2.58}$$

with $\mathbf{y}_H$ the highest scale detailed transform coefficients. The Bayes shrink threshold is calculated by

$$T_{\text{BS}} = \frac{\sigma^2}{\sigma_x} \tag{2.59}$$

where

$$\sigma_x = \sqrt{\max(\sigma_y^2 - \sigma^2, 0)}. \tag{2.60}$$

In case of $\sigma_y^2 < \sigma^2$, the threshold T_{BS} is going towards infinity. In practice $T_B = \max|\mathbf{y}|$ is then taken and so all transform coefficients are set to zero.

2. The modified Bayes shrink [Elyasi and Zarmehi 2009] removes noise better than the BS method. Instead of one common threshold value the MBS threshold is different for each scale. It is calculated by

$$T_{\text{MBS}} = \frac{\beta\sigma_v^2}{\sigma_x} \tag{2.61}$$

with

$$\beta = \sqrt{\frac{\log N}{2j}} \tag{2.62}$$

where N denotes the complete number of wavelet transform coefficients and j is the decomposition level of the current scale.

3. The universal threshold [Rangarajan et al. 2002] is defined by

$$T_{\text{U}} = \sigma\sqrt{2\ln N} \tag{2.63}$$

with N the number of wavelet coefficients of each and σ the standard deviation of the detailed coefficients from first scale.

4. Histogram threshold estimation deals with creating a histogram from wavelet transform coefficients $\mathcal{H}_C(x)$ separately for each scale. The threshold T_{H} is then obtained by

$$\int_{-\infty}^{T_{\text{H}}} \mathcal{H}_C(x)\mathrm{d}x = \epsilon \tag{2.64}$$

with $\epsilon \in [0, 1]$ the sensitivity parameter. A good choice is $\epsilon = 0.95$. For this estimation the histogram has to be normalized

$$\int_{-\infty}^{\infty} \mathcal{H}(x)\mathrm{d}x = 1. \tag{2.65}$$

One of the estimated thresholds is now used to process the transform coefficients. The first method is the hard threshold the hard threshold, shown in Figure 2.12a applied on the function

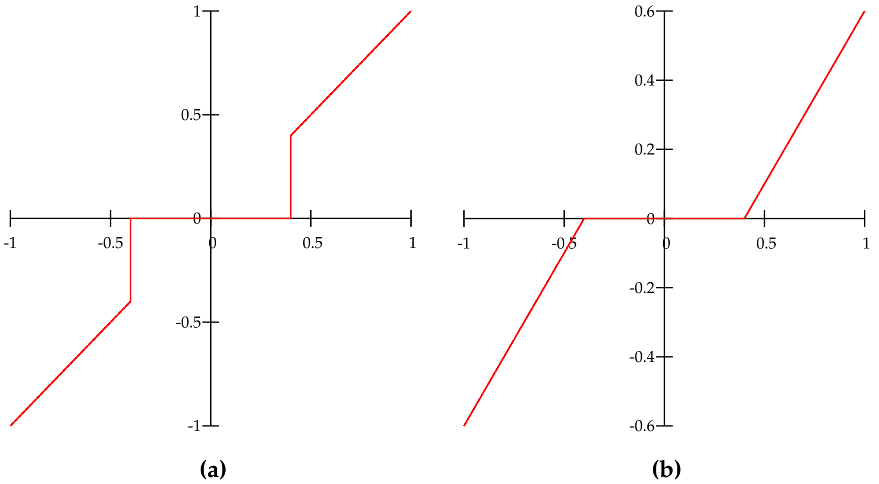


Figure 2.12: Result of **(a)** hard thresholding and **(b)** soft thresholding of the signal $y = x$ for threshold $T = 0.4$

$y = x$ with threshold $T = 0.4$. The second method is the soft threshold, illustrated in Figure 2.12b. Either the hard- or soft thresholding is applied on the wavelet transform coefficients. The so called hard thresholding for the coefficient c and threshold T is defined by

$$\eta_T^{\text{hard}}(c) = \begin{cases} c & \text{for } |c| > T \\ 0 & \text{otherwise} \end{cases} . \tag{2.66}$$

The soft thresholding is given by

$$\eta_T^{\text{soft}}(c) = \begin{cases} \text{sign}(c)(|c| - T) & \text{for } |c| > T \\ 0 & \text{otherwise} \end{cases} . \tag{2.67}$$

After discussing the theoretical approach the complete detection algorithm and the results are presented in the following.

2.8 Detection Algorithm

The detection algorithm is split into two parts. First a smoothed version of the sampled signal is obtained by signal denoising with wavelet threshold technique, cf. Figure 2.11, followed by application of the designed filter as illustrated in Figure 2.13. The available parameters for the

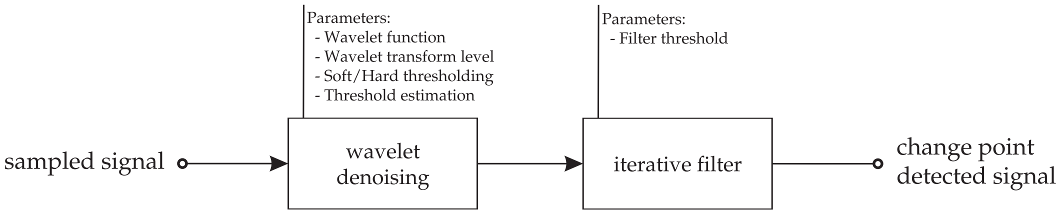


Figure 2.13: Schematic of complete wavelet change point detection algorithm

smoothing operation depicted in Figure 2.13 are:

- *Wavelet type*: Select the mother wavelet function for the smoothing process, e.g. the Haar or the Daubechies wavelet.
- *Wavelet transform level*: Set the number of the highest wavelet transform level. The preferred choice is $J = 3$.
- *Threshold type*: Both hard and soft thresholding are implemented.
- *Threshold estimation*: One of four threshold estimation processes, from Section 2.7, is selected.

The finally applied filter process is divided into following sequential steps:

1. Calculate the first derivative of the smoothed signal.
2. Detect the time instances where the signals first derivative changes the sign.
3. The highest changes of the original signal, between the time instances where the first derivative changes the sign, are identified as change points.
4. The data points between two change points are arithmetical averaged.
5. Iteratively remove estimated change points with step size smaller a desired filter threshold $\mathrm{T}_R(\kappa)$ and apply a median filter with window width $w = 5$ to remove single peaks. The filter threshold is set by

$$\mathrm{T}_R(\kappa) = \kappa \frac{\text{median}|r_i|}{0.6745} \tag{2.68}$$

with a filtering parameter κ. Setting $\kappa = 0$ disables the filter.

2.9 Evaluation of Change Point Detection

First results of the wavelet denoising technique is shown in Figure 2.14 [1]. The wavelet coef-

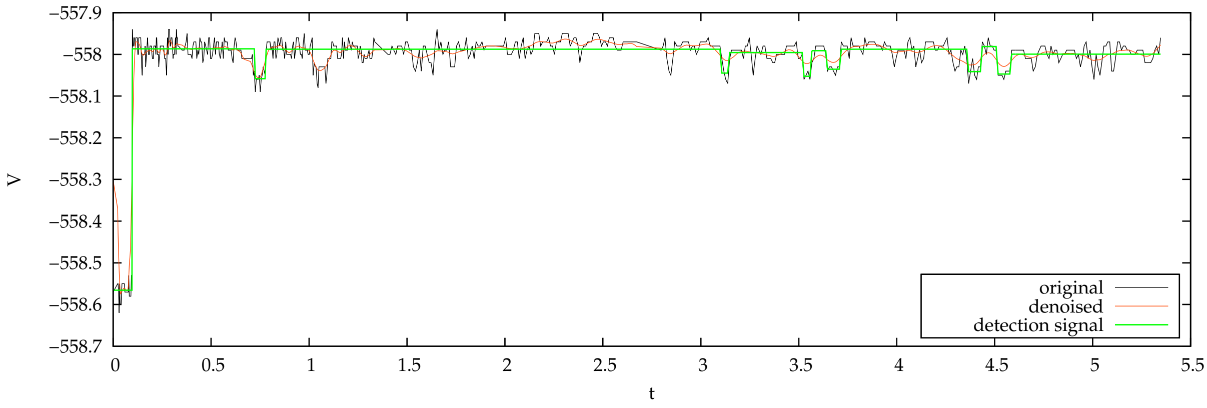


Figure 2.14: Wavelet change point detection with stationary discrete wavelet transform, Haar wavelet, transform level $J = 3$, modified Bayes shrink threshold estimation, soft thresholding and with $\kappa = 0.8$

[1] The definition of the signal to noise ratio and the mean square error used to emphasize the denoising quality is given in Section A.4.

ficients for a level $J = 3$ stationary discrete wavelet transform are illustrated in Figure 2.15. Detailed coefficients are $d_1[i], d_2[i]$ and $d_3[i]$. They are relative small values and describe the fast behavior of the signal. Slow events are found in coefficients $a_3[i]$. The threshold is estimated by modified Bayes shrink and applied by soft thresholding to the detailed wavelet transform coefficients.

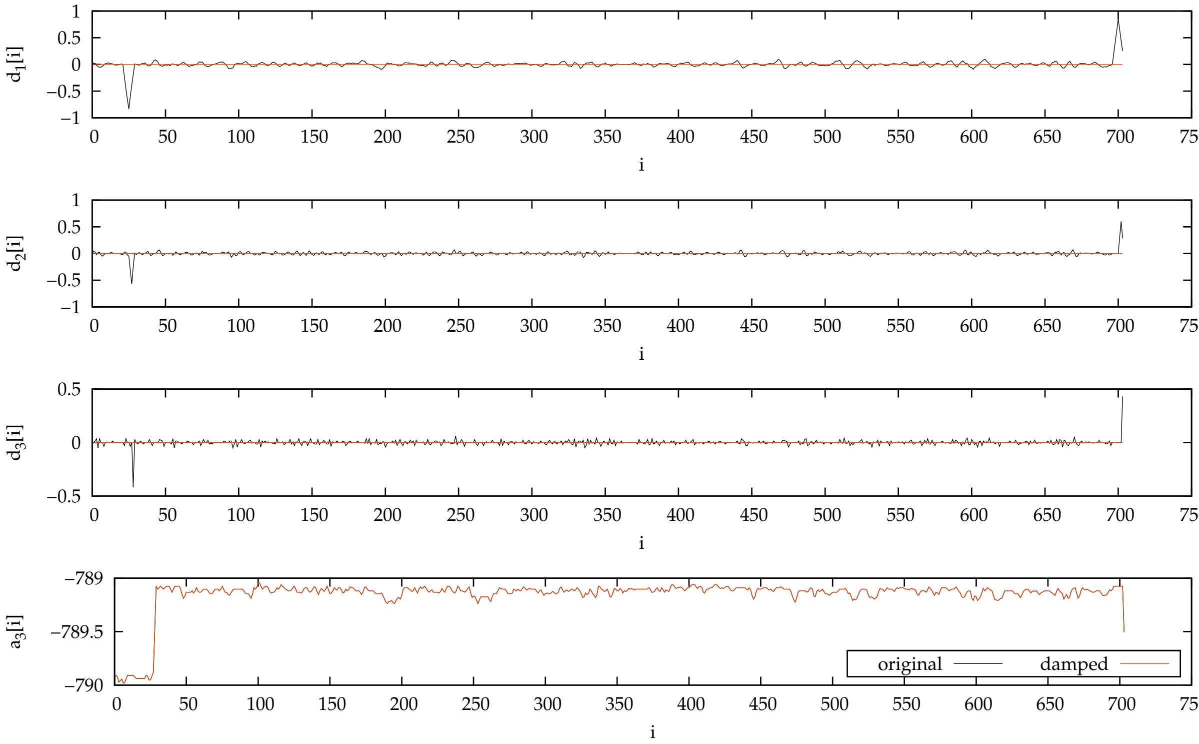


Figure 2.15: Stationary wavelet transform coefficients for the signal at Figure 2.14 and soft thresholding with modified Bayes shrink threshold estimation

A comparison of denoising with the stationary discrete wavelet transform with different threshold estimates and thresholding techniques is summarized in Table 2.1 for the signal of Figure 2.14.

Similar results are shown with the Bayes shrink and modified Bayes shrink threshold estimation in combination with soft and hard thresholding. Both have higher denoising efficiency than universal threshold and histogram threshold estimation. It must be noted that for histogram threshold estimation there is one degree of freedom. For the evaluation in Table 2.1 the sensitivity parameter $\epsilon = 0.95$ is selected. Higher values of ϵ improve the denoising quality. So this method can be adjusted to a given signal. Moreover there is hardly a difference between hard and soft thresholding the transform coefficients.

The study of histogram threshold estimation is done in Table 2.2 with the stationary discrete wavelet transform and soft thresholding. For higher values of ϵ smoother versions of the original signals are obtained.

Remaining is the comparison of denoising with the stationary discrete wavelet transform and the discrete wavelet transform. An evaluation of the discrete wavelet transform and thresholding is given in Table 2.3. A comparison with Table 2.1 shows very similar results. The advantage of the discrete wavelet transform is that the number of the transform coefficients is halved after each

		soft threshold				hard threshold			
		BS	MBS	UT	HT	BS	MBS	UT	HT
Haar	SNR	85.48	85.48	89.05	91.04	86.36	85.48	90.03	94.70
	MSE	-30.55	-30.55	-34.11	-36.10	-31.43	-30.55	-35.10	-39.77
DB5	SNR	85.66	85.64	87.40	90.31	87.33	85.64	89.40	92.96
	MSE	-30.73	-30.70	-32.47	-35.38	-32.40	-30.70	-34.47	-38.02
DB10	SNR	85.57	85.57	86.89	89.93	86.14	85.57	89.15	92.55
	MSE	-30.64	-30.64	-31.96	-35.00	-31.20	-30.64	-34.22	-37.62
DB15	SNR	85.54	85.54	86.84	89.57	86.07	85.54	88.64	92.01
	MSE	-30.60	-30.60	-31.70	-34.64	-31.14	-30.60	-33.71	-37.08
DB20	SNR	85.61	85.58	86.49	89.24	86.53	85.92	88.13	91.69
	MSE	-30.68	-30.64	-31.55	-34.31	-31.60	-30.99	-33.20	-36.75

Table 2.1: SNR (dB) and MSE (dB) overview of signal denoising for test signal from Figure 2.14 with stationary discrete wavelet transform; histogram threshold (HT) estimation with $\gamma = 0.95$

		Haar	DB5	DB10	DB15	DB20
$\epsilon = 0.92$	SNR	92.07	91.02	90.65	90.32	90.14
	MSE	-37.13	-36.08	-35.71	-35.39	-35.21
$\epsilon = 0.95$	SNR	91.04	90.31	89.93	89.57	89.24
	MSE	-36.10	-35.38	-35.00	-34.64	-34.30
$\epsilon = 0.98$	SNR	90.29	88.96	88.11	87.87	87.52
	MSE	-35.35	-33.96	-33.18	-32.94	-32.59

Table 2.2: Effects on denoising with histogram threshold estimation at different sensitivity parameters ϵ for stationary discrete wavelet transform and soft thresholding

single transform step because of downsampling of the transform coefficients, cf. Figure 2.4. So there are less coefficients than at the stationary discrete wavelet transform for applying the soft and hard threshold. The presented charge trap measurement signals have less than 10^3 samples. In this case there is no difference in transformation time between the discrete and stationary discrete wavelet transform.

All presented examples are transformed via the stationary discrete wavelet transform. Its advantage, often mentioned in the literature, is its shift invariance. This means that time shifted version of the input signal $r(t + T_0)$ lead to exact the same coefficients as the original $r(t)$ when the stationary discrete wavelet transform is used. With the discrete wavelet transform different transform coefficients are obtained in the two cases because downsampling by two at each scale.

An example of a detection process is shown in Figure 2.14. A comparison between original and detection signal reveals an SNR $= 88.32$dB. The detection process is executed with the stationary discrete wavelet transform with Haar wavelet and transform coefficients are filtered by modified Bayes shrink threshold estimation and soft thresholding. Finally a filtering process with $\kappa = 0.8$ leads to the steps shown in the figure. A very high step at the begin is detected. Several smaller changes occur later, which are very difficult to detect. Here a good choice of κ is highly

		soft threshold				hard threshold			
		BS	MBS	VS	HS	BS	MBS	VS	HS
Haar	SNR	89.00	88.86	89.38	90.17	89.48	89.31	90.22	91.85
	MSE	-34.07	-33.93	-34.44	-35.24	-34.55	-34.37	-35.28	-36.92
DB5	SNR	86.25	86.08	90.45	89.88	88.75	86.21	92.15	91.72
	MSE	-31.32	-31.14	-35.52	-34.95	-33.81	-31.27	-37.22	-36.79
DB10	SNR	86.97	86.35	90.19	89.60	88.75	87.62	92.16	91.43
	MSE	-32.04	-31.41	-35.25	-34.67	-33.82	-32.69	-37.23	-36.50
DB15	SNR	87.05	86.99	89.93	90.18	88.78	87.59	91.87	92.04
	MSE	-33.12	-32.05	-35.00	-35.25	-33.85	-32.66	-36.93	-37.11
DB20	SNR	86.69	86.65	90.15	88.36	87.75	86.76	92.05	90.32
	MSE	-31.76	-31.72	-35.22	-33.43	-32.82	-31.83	-37.12	-35.38

Table 2.3: SNR (dB) and MSE (dB) overview of signal denoising for test signal from Figure 2.14 with discrete wavelet transform; histogram threshold (HT) estimation with $\gamma = 0.95$

important.

Another example of evaluation of a good quality measurement signal shows Figure 2.16. The evaluation of the original signal and the reconstructed step signal gives an SNR = 78.26dB. Now the SNR is a little lower than with the signal of Figure 2.14. Here only two big changes occur and the rest of the signal is treated as noise. As a consequence it is clear that for lower calculated SNR the count of detected steps must decrease.

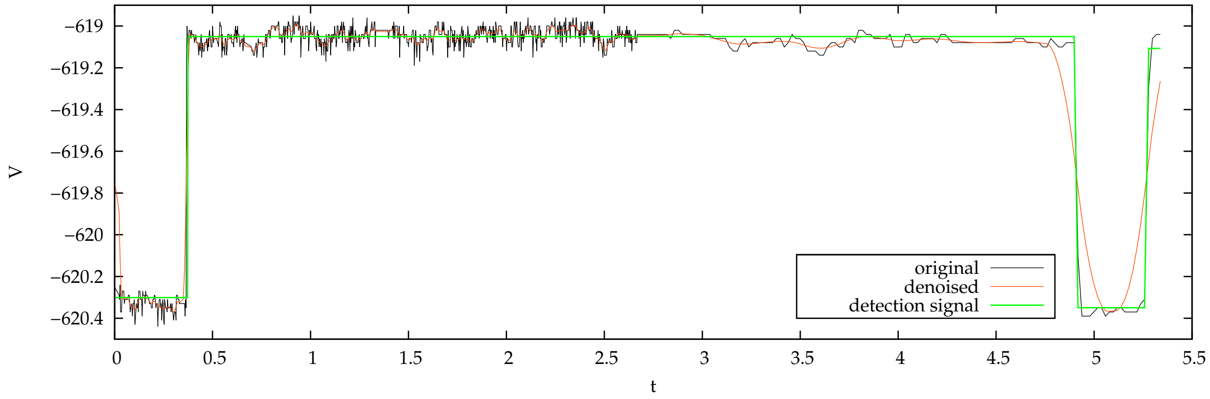


Figure 2.16: Origin measurement and detection signal with stationary discrete wavelet transform, Haar wavelet, transform level $J = 3$, modified Bayes threshold estimation, soft thresholding and final filtering with $\kappa = 0.8$

A very noisy signal is presented in Figure 2.17. With the choice $\kappa = 0.8$ the execution of the algorithm on gives an SNR = 73.78dB. In this case the denoising is very effective. Nearly the complete noise is removed and the peaks remain. Then the peaks can be detected very easily. On such signals the wavelet-denoising-based detection method shows very good results and works quite fast.

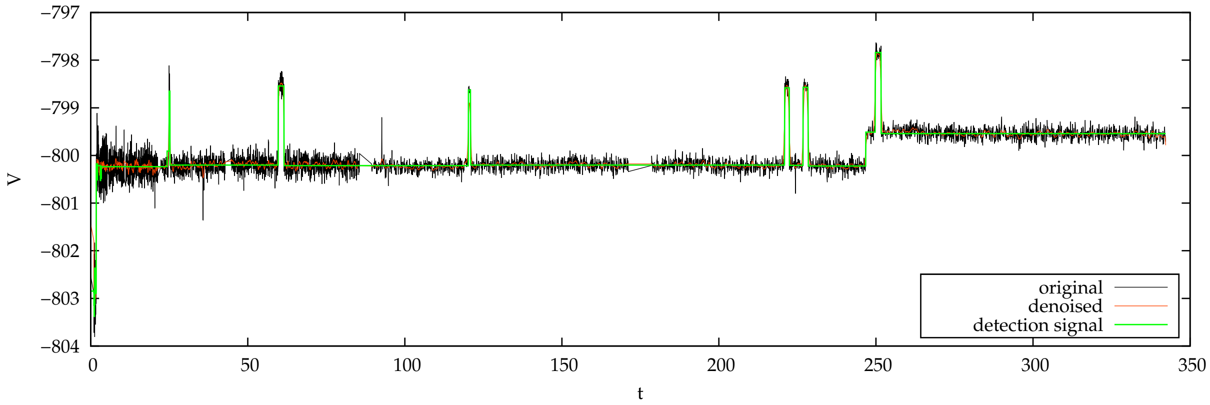


Figure 2.17: Noisy signal change point detection with stationary discrete wavelet transform, Haar wavelet, transform level $J = 3$, modified Bayes shrink estimation, soft thresholding and final filtering with $\kappa = 0.8$

CHAPTER 3

HISTOGRAM-BASED CHANGE POINT DETECTION

One of the simplest but still very effective method to analyze statistical parameters are histograms. Since its introduction in [Pearson 1895] they have been used in many applications. It is the classical tool for nonparametric density estimation and also of interest for change point analysis.

First some basic properties of histograms are described. Then a smoothed form obtained by inserting interpolation points between observations is presented. As Gaussian distributed data is assumed estimation of the parameters expectation and variance is discussed. Before the evaluation, a method for non-uniformly sampled data histograms is given.

3.1 Introduction to Histograms

A histogram essentially is a function $\mathcal{H}(x)$ which counts values belonging to disjoint data intervals called bins. The histogram is defined on an equally spaced mesh with bin width

$$h = x_{i+1} - x_i \tag{3.1}$$

and $i \in \mathbb{N}$. In order to interpret $\mathcal{H}(x)$ as the probability for $x \in [x_i, x_{i+1}]$ the histogram is normalized to

$$\sum_i \mathcal{H}(x_i) = 1. \tag{3.2}$$

For a multi-level signal with additive Gaussian noise the histogram is illustrated in Figure 3.1. Very important is the choice of number of bins. With a given set of observations

$$\mathbf{r} = [r_1, r_2, \dots, r_N] \tag{3.3}$$

the number of bins calculates to

$$K = \text{ceil}\left\{\frac{\max \mathbf{r} - \min \mathbf{r}}{h}\right\}. \tag{3.4}$$

There is no best choice, but several theoretical approaches to it. Since the density is usually unknown, assumed Gaussian density leads to the choice

$$h_n = \frac{3.49\sigma}{N^{1/3}} \tag{3.5}$$

with σ the observations standard deviation and N the observations vector length [Scott 1979].

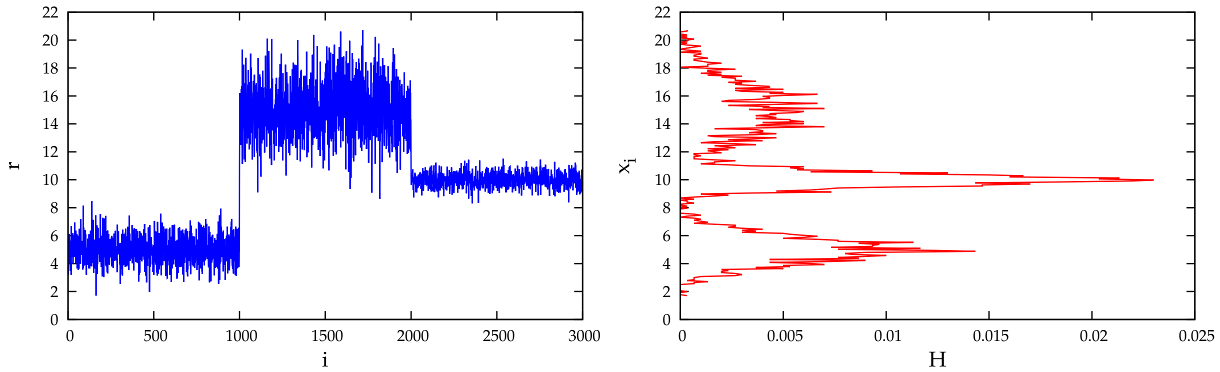


Figure 3.1: Example observations r_i and their histogram for the Gaussian distributions $N(5,1)$, $N(10,0.5)$ and $N(15,3)$ with the number of bins set to $K = 265$

3.2 Increase Histogram Quality with Interpolation

Histograms of short observation vectors are often of poor quality. So it is very difficult to estimate the density function of the observed data set. To increase quality it is useful to insert data points by interpolation. After the discussion of the regression problem the increase quality performance is demonstrated.

3.2.1 Formulation of the Approximation Problem

Assume a set of observations in vector notation

$$\mathbf{r} = [r_1, r_2, \dots, r_N]\,. \tag{3.6}$$

For the least square approximation in the Hilbert space [Moon and Stirling 2000] the l_2 norm is defined by

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{N} |x_i|^2 \right)^{1/2} \tag{3.7}$$

and the inner product by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^H \mathbf{x}. \tag{3.8}$$

The approximation problem is to find coefficients $c_1, c_2, \dots, c_m$ so that an estimation of $\mathbf{r}$

$$\hat{\mathbf{r}} = c_1 \mathbf{p}_1 + c_2 \mathbf{p}_2 + \ldots + c_m \mathbf{p}_m = [\mathbf{p}_1 \mathbf{p}_2 \ldots \mathbf{p}_m] \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \tag{3.9}$$

with $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m$ a set of linearly independent vectors has the minimum least square error vector

$$\|\mathbf{e}\|_2^2 = \|\mathbf{r} - \hat{\mathbf{r}}\|_2^2. \tag{3.10}$$

In matrix notation

$$A = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m], \qquad \mathbf{c} = [c_1, c_2, \dots, c_m]^T$$

the problem can be written as

$$\|\mathbf{e}\|_2^2 = \|\mathbf{r} - A\mathbf{c}\|_2^2. \tag{3.11}$$

The minimum least square solution is obtained if the error vector **e** is orthogonal to all vectors $\mathbf{p}_i$. Therefor all inner products have to be zero

$$\langle \mathbf{r} - \hat{\mathbf{r}}, \mathbf{p}_i \rangle = 0, \tag{3.12}$$

in matrix notation summarized by

$$\langle \mathbf{r} - A\mathbf{c}, A \rangle = 0. \tag{3.13}$$

Then the least square coefficients are

$$\mathbf{c} = \left(A^H A\right)^{-1} A^H \mathbf{r}. \tag{3.14}$$

To increase the importance of some data points for regression a weighted least square approximation [Moon and Stirling 2000] is used. It is implemented into the inner product, which leads to the weighted inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle_W = \mathbf{x}^H W \mathbf{y}. \tag{3.15}$$

with a diagonal weighting matrix W. So the least square solution is

$$\mathbf{c} = \left(A^H W A\right)^{-1} A^H W \mathbf{r}. \tag{3.16}$$

For given data points (x_i, r_i) a polynomial approximation of order m is given by

$$\hat{r}_i = c_m x_i^m + c_{m-1} x_i^{m-1} + \ldots + c_0. \tag{3.17}$$

The approximation problem in detail is

$$\begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_n \end{bmatrix} = \begin{bmatrix} c_m x_1^m + c_{m-1} x_1^{m-1} + \ldots + c_0 \\ c_m x_2^m + c_{m-1} x_2^{m-1} + \ldots + c_0 \\ \vdots \\ c_m x_n^m + c_{m-1} x_n^{m-1} + \ldots + c_0 \end{bmatrix} \tag{3.18}$$

Under introduction of

$$X = \begin{bmatrix} x_1^m & x_1^{m-1} & \ldots & 1 \\ x_2^m & x_2^{m-1} & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ x_n^m & x_n^{m-1} & \ldots & 1 \end{bmatrix}$$

the problem is reformulated to

$$\hat{\mathbf{r}} = X\mathbf{c} \tag{3.19}$$

with the solution

$$\mathbf{c} = \left(X^H X\right)^{-1} X^H \mathbf{r}. \tag{3.20}$$

3.2.2 Effects of Interpolation

To show the effect of increasing observation points through interpolation, three methods are mentioned. They are the

1. linear (order $m = 1$), the
2. quadratic (order $m = 2$) and the
3. polynomial (order $m > 3$) regression.

Since the problem of linear and quadratic regression is clear, the method of polynomial regression has to be discussed. A sliding data window with an odd window width w is used. The mean value of the data window is given by

$$\mu_w = \frac{1}{w} \sum_{i=i_s}^{i_e} r_i \tag{3.21}$$

with i_s and i_e, note that $w = |i_s - i_e|$, the first and last index of the window data points respectively. The polynomial regression distinguish between

$$m = \begin{cases} r_{i_s} = \mu_w & \text{and} & r_{i_e} = \mu_w & 1 \\ r_{i_s} < \mu_w & \text{and} & r_{i_e} < \mu_w & \text{even} \\ r_{i_s} > \mu_w & \text{and} & r_{i_e} > \mu_w & \text{even} \\ r_{i_s} < \mu_w & \text{and} & r_{i_e} > \mu_w & \text{odd} \\ r_{i_s} > \mu_w & \text{and} & r_{i_e} < \mu_w & \text{odd} \end{cases} . \tag{3.22}$$

The $m + 1$ polynom coefficients are determined and, with $i_c \in [i_s, i_e]$ the index of the center data point in the interval $[r_c, r_{c+1}]$, the interpolated data is calculated and binned to the histogram. An evaluation of histograms for different approximations is given in Figure 3.2. For the polynomial regression the odd order $m = 5$ and the even order $m = 4$ is used. The polynomial approximation window width is set to $w = 5$. For increasing order of the interpolation polynom smoother histograms are obtained. Best results shows the polynomial regression.

3.3 Fitting Gauss Distributions to Peaks

In order to fit Gaussian distributions to the histogram peaks a combination of two techniques is necessary. First the histogram baseline is estimated with a sensitive nonlinear iterative peak (SNIP) clipping algorithm [Morháč and Matoušek 2008, Morháč 2009]. With an original histogram function $\mathcal{H}(x_i)$ with N bins the baseline is evaluated by a second order filter with decreasing clipping window $p = m, m-1, \ldots, 1$. Assume

$$\mathcal{N}_m(x_i) = \mathcal{H}(x_i) \quad \text{for} \quad i \in [0, N-1] \tag{3.23}$$

is the input histogram at estimation step $p = m$. Applying the second order filter at iteration step p

$$a_p(x_i) = \mathcal{N}_p(x_i) \qquad b_p(x_i) = \frac{\mathcal{N}_p(x_{i-p}) + \mathcal{N}_p(x_{i+p})}{2} \tag{3.24}$$

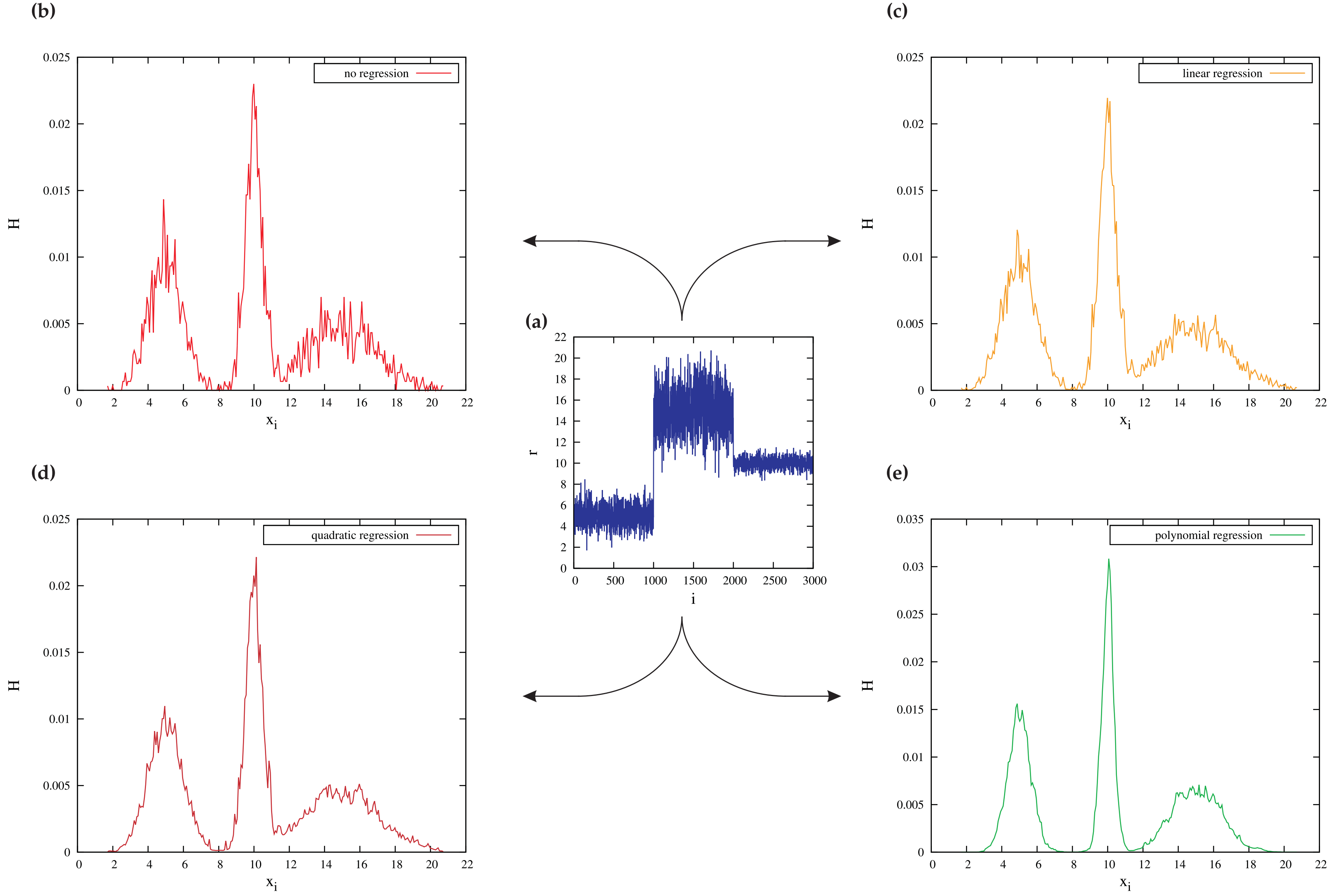


Figure 3.2: Comparison of different histograms of a **(a)** data set produced by **(b)** no, **(c)** linear, **(d)** quadratic and **(e)** polynomial interpolation of data points between observations with 250 interpolation points and approximation window width $w = 5$

leads to

$$\mathcal{N}_{p-1}(x_i) = \min(a_p(x_i), b_p(x_i)). \tag{3.25}$$

For one estimation step p the iteration of $i \in [p, (N-1) - p]$ is necessary. The maximal number of fitting steps is set by

$$2m + 1 = w \tag{3.26}$$

with w the number bins related to the widest peak of the histogram. The estimated baseline of the histogram $\mathcal{N}_0(x_i)$ from Figure 3.1 is shown in Figure 3.3.

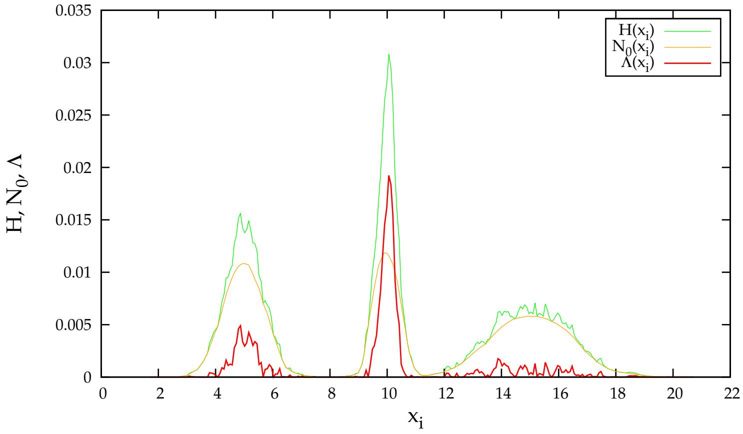


Figure 3.3: Baseline estimation and elimination of histogram from Figure 3.2e

Next, Gaussian distributions can be fitted to the peaks in the background-eliminated histogram $\Lambda(x_i) = \mathcal{H}(x_i) - \mathcal{N}_0(x_i)$. For this a convolution algorithm is used [Likar and Vidmar 2003, Morháč 2009]. The idea is to convolute the histogram two times with the first order derivative of an unnormalized Gaussian function

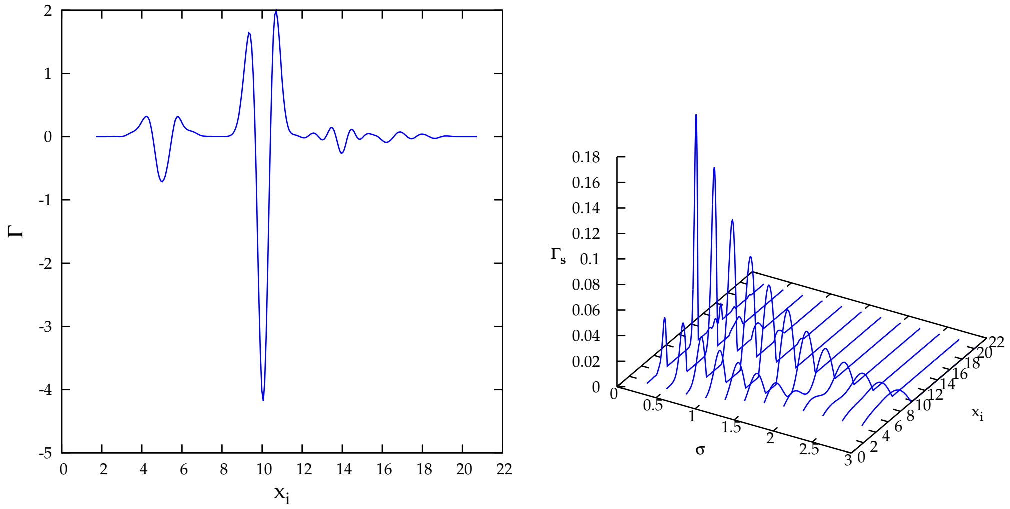
$$g(x, \mu, \sigma) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{3.27}$$

$$g'(x, \mu, \sigma) = \frac{\mathrm{d}g(x, \mu, \sigma)}{\mathrm{d}x} = -\frac{x-\mu}{\sigma^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{3.28}$$

The double convolution of the histogram shown in Figure 3.2e with the unnormalized Gaussian derivative $g'(x, \mu, \sigma)$ results in $\Gamma(x_i)$ presented in Figure 3.4a. The better σ in the convolution function $g'(x, \mu, \sigma)$ fits to the standard deviation of the peaks in the histogram, the more negative the peaks of $\Gamma(x_i)$ get. To evaluate several standard deviations a scanning process is applied. By reformulating $\Gamma(x_i)$ to

$$\Gamma_s(x_i) = |\min\left(\Gamma(x_i), 0\right)| \tag{3.29}$$

the result of a σ scan is illustrated in Figure 3.4b. Then the peak positions and their standard deviations are estimated. The results of this method are the better the histogram peaks are similar to Gaussian peaks.



(a) **(b)**

Figure 3.4: Histogram **(a)** convolution with first order Gaussian derivative with $sigma = 0.216$ and **(b)** σ scan to evaluate standard deviations of other peaks for the histogram illustrated in Figure 3.2e

3.4 Weighted Histogram for Non-Uniformly Sampled Data

In case of non-uniformly sampled data the noise power in the signal can not be assumed to be constant. Consider the Fourier transform

$$\mathcal{F}[r(t)] = R(\mathrm{j}\omega) = \int_{-\infty}^{\infty} r(t)e^{-\mathrm{j}\omega t}\mathrm{d}t \qquad (3.30)$$

for time continuous signals which transforms a time signal $r(t)$ into the frequency domain with its spectrum $R(\mathrm{j}\omega)$. Of interest is the scaling property of the Fourier transform. For a non zero real number a the corresponding Fourier transform pair is

$$r(at) \circ\!\!-\!\!\!\bullet \frac{1}{|a|}R\left(\mathrm{j}\frac{\omega}{a}\right). \qquad (3.31)$$

It is shown that the scaling factor $|a| > 1$ represents a time expansion and in the frequency domain a compression. In contrast a time compression $|a| < 1$ leads to frequency expansion. So fast operations result in a wide spectrum and vice versa.

Assuming white noise, with constant auto-correlation function, the Fourier transform scaling property shows that fast-sampled signals contain more noise power that slower sampled ones. In case of increasing sample time the signals noise power decreases. To handle this fact weighted histograms are introduced.

Assume a set of observations r_i sampled at non-equidistant time instances t_i with $i \in [0, N-1]$. The idea of a weighted histogram is instead of increasing a bins value by one, as it is done in classical histograms, the value of the sample time $T_i = t_i - t_{i-1}$ is added to the bin related to r_i. Normalizing the histogram (3.2) leads to the probabilities that a value is contained in the individual bin.

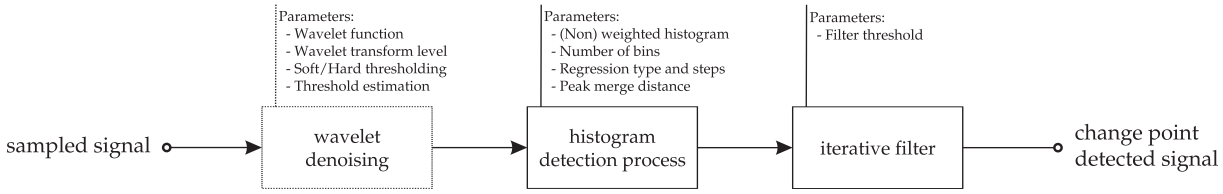


Figure 3.5: Structural overview of histogram-based detection algorithm

3.5 Histogram Detection Algorithm

The structure of the detection process and its parameters are given in Figure 3.5. After an optional denoising step based on the stationary discrete wavelet transform, cf. Chapter 2, a histogram is created from the input signal. The histogram generation parameters are:

- *Weighted/non-weighted*: Choose between weighted or non weighted histograms.
- *Number of bins*: Estimate the optimally number of bins or set the number manually.
- *Regression*: Set regression type and number of interpolated data-points between observations.
- *Baseline estimation*: The window-width for histogram baseline estimation can be set or the default value $w = 5$ is used.
- *Peak merge parameter*: Define the minimum distance between two Gaussian distributions estimated from the data set to be merged to one single distribution. Two Gaussian functions are merged to a single one in case of $|\mu_n - \mu_{n+1}| < \epsilon$.

Creating a histogram from the observations is followed by a Gaussian peak detection process via convolution scans discussed previously. Histograms generated from time dependent defect spectroscopy measurement data show a great number of peaks, most of them resulting from quantization of the analog-to-digital converter. The great number of peaks demand merging of Gaussian distributions with only a small difference in their means. The peak merging sensitivity ϵ is either set by the user or estimated with the robust median estimator

$$\epsilon = \frac{\text{median}|r_i|}{0.6745}. \tag{3.32}$$

So now all Gaussian distributions with $|\mu_n - \mu_{n+1}| < \epsilon$ are iteratively merged to a single Gaussian peak before applying the smoothing on the data set. Finally the iterative filter of Section 2.8 is applied.

3.6 Evaluation of Histogram-Based Change Point Detection

The evaluation of time dependent defect spectroscopy measurement data with the presented algorithm, cf. Figure 3.5, is now discussed[1]. First the results obtained without the wavelet denoising technique from Chapter 2 are presented. Finally the effect of wavelet denoising on measurement data is presented.

[1] The definition of the signal to noise ratio and the mean square error used to emphasize the denoising quality is given in Section A.4.

3.6.1 Simple Detection Algorithm

The way from observations to detected signal is discussed with the help of a typical set of time dependent defect spectroscopy measurement data. An example for a successful step detection is illustrated in Figure 3.6.

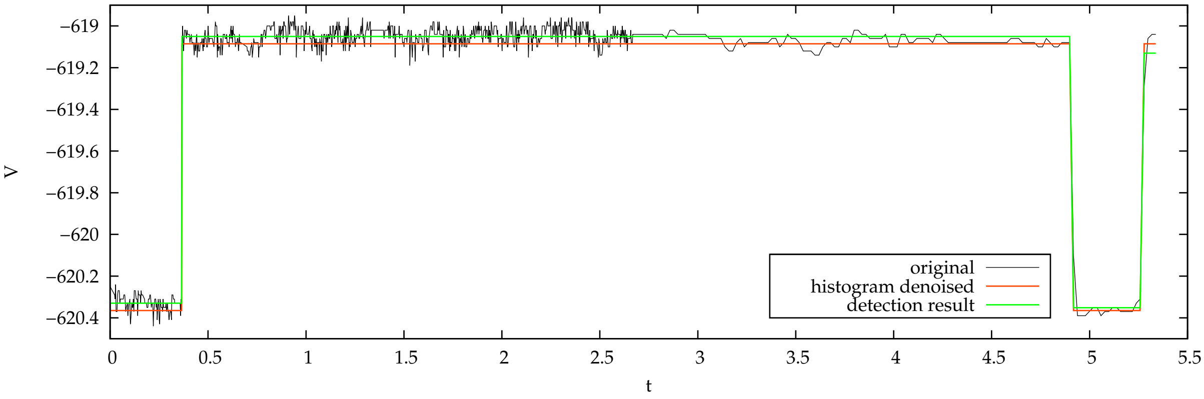


Figure 3.6: Histogram change point detection with a weighted histogram, polynomial regression type with 251 interpolation steps, baseline estimation window width $w = 5$, number of bins set to 100, $\kappa = 0$ and without denoising using the wavelet denoising technique

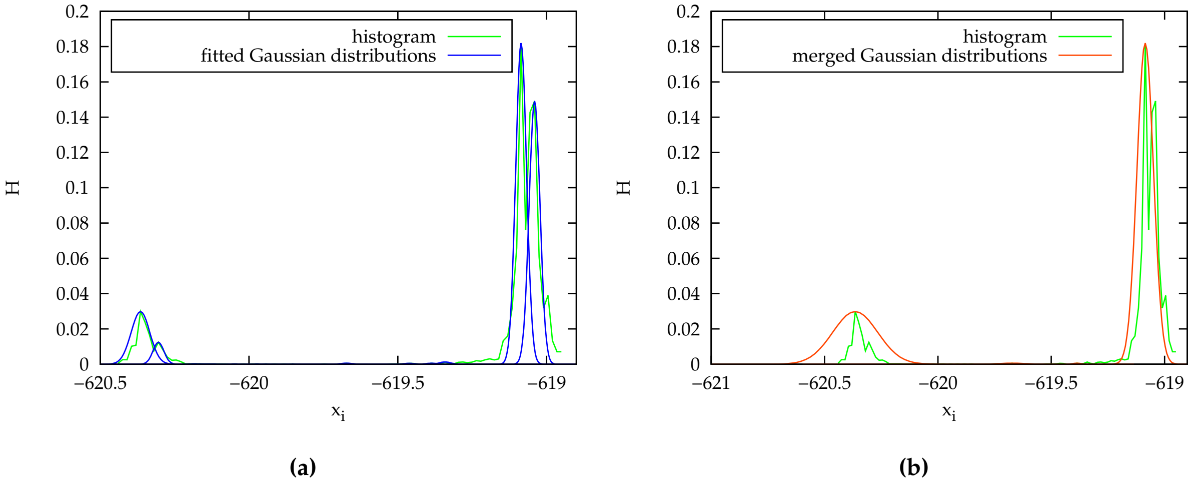


Figure 3.7: Histogram for measurement data from Figure 3.6 and **(a)** Gaussian distributions fitted by a convolution scan and **(b)** merged Gaussian distributions with ϵ estimated by the robust median estimator

First the data values are binned into a histogram shown in Figure 3.7a. Then with convolution the highest peak is estimated. Several convolution processes have to be done to detect the variance σ_H^2 which fits the peak best. The smallest value of the convolution process obtained by different variances gives the optimal choice of σ_H. Notice that the convolution is only executed in the range of the highest peak in the histogram in order to make the algorithm fast. After a first Gaussian distribution estimate a convolution scan of the hole histogram in the standard deviation range $0.1\sigma_H \leq \sigma \leq 2\sigma_H$ is applied. The best results for the estimated Gaussian distributions are then obtained from the convolution scan. The estimated Gaussian distributions from the convolution scan are illustrated in Figure 3.7a. Because of the quantization of the analog-to-digital conversion many histogram peaks do not belong to step data. Therefore several Gaussian distributions with $|\mu_n - \mu_{n+1}| < \epsilon$ are merged to a single one, cf. Figure 3.7b. Now the observations are mapped to the Gaussian distribution mean values by their probability. Finally all steps are fitted to the original data values by building means over their flat level. The result from this detection is given in Figure 3.6 which leads to SNR = 83.54dB. Another example is presented in Figure 3.8 for an SNR = 72.49dB. This configuration is preferred with similar noisy signals.

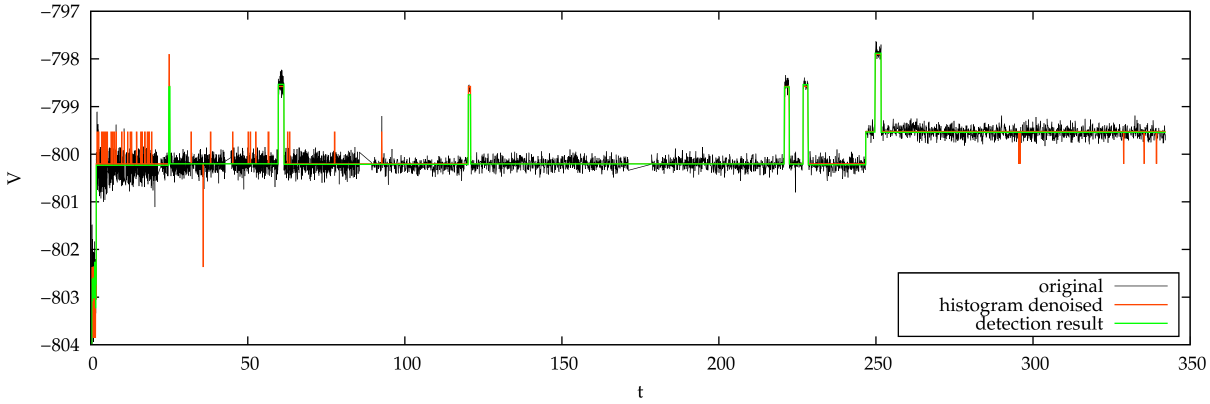


Figure 3.8: Detection of change points of a very noisy signal with a weighted histogram, polynomial regression type with 251 interpolation steps, $w = 5$, $K = 100$, $\kappa = 0.8$ without wavelet denoising

3.6.2 Histogram Algorithm on Wavelet Denoised Signals

The effect of signal denoising on the detection method is discussed now and presented in Figure 3.9. There is only a small difference between the two obtained signals. In case of denoising a better result is given in the heavily noisy part at the beginning of the signal. Summarized, the denoising process is preferred for heavily noisy signals.

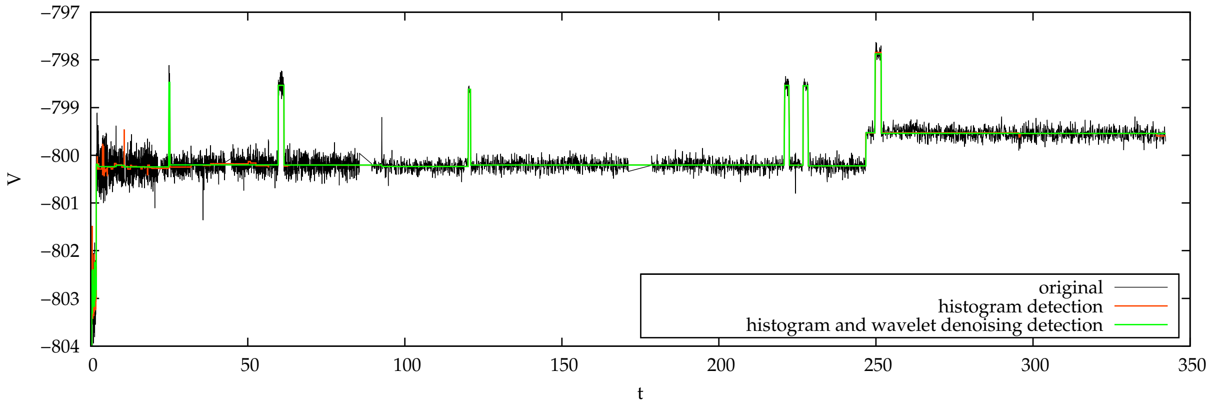


Figure 3.9: Comparison of change point analysis of very noisy signal with and without using the wavelet denoising technique before the histogram step detection algorithm is applied for the signal from Figure 2.17 and from Figure 3.8, the iterative filter is disabled, e.g. $\kappa = 0$

CHAPTER 4

BOOTSTRAP AND CUMULATIVE SUMS FOR CHANGE POINT ANALYSIS

The next method for detecting change points is inspired by [Tayler 2000] and a combination of cumulative sum statistics and bootstrapping. With an additional confidence parameter a detection sensitivity can be set.

First an overview of the cumulative sums is given. The definition and their properties in case of a change point is discussed. Then the bootstrap is presented and the first ideas leading to bootstraps are reviewed. The application of these two techniques to time dependent defect spectroscopy measurements for change point concludes this chapter.

4.1 Cumulative Sum Statistics

The cumulative sum statistic has been introduced by [Page 1954]. Since then it has been used in many applications to monitor process-mean shifts. When one specific amount of shift can be assumed the cumulative sum chart can be optimally designed to alert when a change in signal mean occurs.

The responsible model for mean shift detection with cumulative sum charts is given by

$$r_i = \begin{cases} x_0 + v_i & 0 \leq i < i_C \\ x_1 + v_i & i_C \leq i < N \end{cases} \tag{4.1}$$

with the observations r_i as sum of constant values x_0 before the mean shift at $i = i_C$ and x_1 after, and a random noise process v_i. The point i_C is called the change point. The goal of the cumulative sum charts is to detect the mean shift as fast as possible.

The upper cumulative sum statistic is defined as

$$\mathcal{C}_i^+ = \max\left(0, \mathcal{C}_{i-1}^+ + r_i - k\right) \tag{4.2}$$

and the lower cumulative sum statistic is as

$$\mathcal{C}_i^- = \max\left(0, \mathcal{C}_{i-1}^- - r_i + k\right), \tag{4.3}$$

where k represents the shift threshold value. A change point occurs when $\mathcal{C}_i^+$ or $\mathcal{C}_i^-$ reaches a predefined control limit $h > 0$. In the case of a known mean shift

$$\delta = |x_0 - x_1| \tag{4.4}$$

the optimal choice of the shift threshold is [Ryu et al. 2010]

$$k = \frac{\delta}{2}. \tag{4.5}$$

An example is given in Figure 4.1. The observations r_i are given by

$$r_i = \begin{cases} 0 + v_i & \text{for } 0 \leq i < 50 \\ 1 + v_i & \text{for } 50 \leq i < 100 \end{cases} \tag{4.6}$$

with v_i Gaussian noise $\mathcal{N}(0, \sigma^2)$, cf. Figure 4.1a. The test signal is prepared with an SNR $=$ 40dB. With (4.5) the optimal threshold value is $k = 0.5$. Some background on creating noisy signals can be found in Appendix A.1.

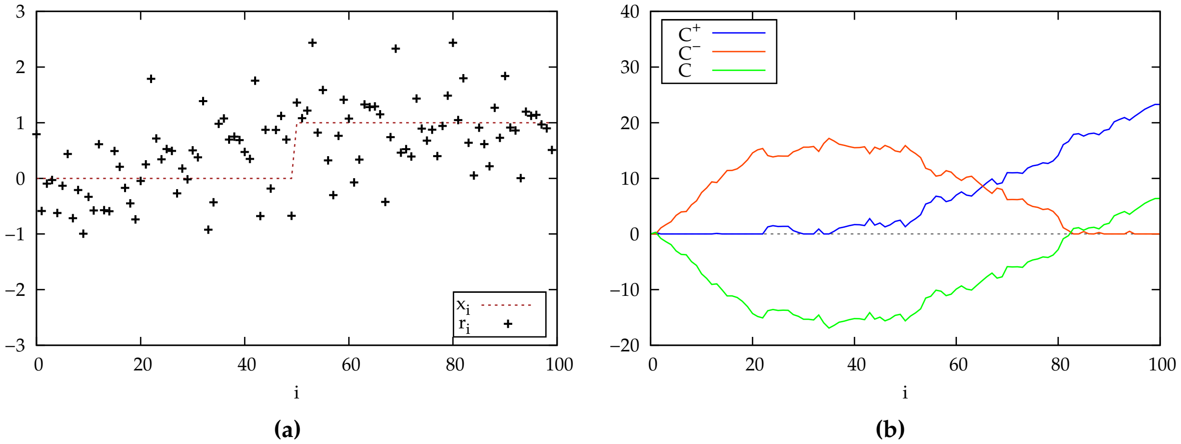


Figure 4.1: Example of **(a)** a noisy signal with SNR $=$ 40dB and **(b)** cumulative sum statistics for a positive mean shift

As presented in Figure 4.1b, observations below δ hold the positive cumulative sum nearly zero. After a change in the mean above the detection threshold, $\mathcal{C}^+$ increases rapidly. So one can set a control limit h to decide if a change in signal mean occurred or not. In contrast to $\mathcal{C}^+$ the lower cumulative sum $\mathcal{C}^-$ is design to detect negative mean shifts in the same way as $\mathcal{C}^+$ detects positive ones. An evaluation of a negative mean shift test signal from Figure 4.2a is illustrated in Figure 4.2b.

Both cumulative sum statistics $\mathcal{C}^+$ and $\mathcal{C}^-$ differ in their detection ability of the mean shift direction. $\mathcal{C}^+$ is used to detect positive mean shifts, while $\mathcal{C}^-$ only leads to meaningful results in case of negative mean shifts. A combination of both, necessary in the case of an unknown shift direction, leads to the complete cumulative sum statistic defined as

$$\mathcal{C}_i = \sum_{j=0}^{i} \left(r_j - k\right). \tag{4.7}$$

In Figure 4.1b the complete cumulative sum $\mathcal{C}$ describes a turning point with negative second derivative for a positive mean shift and a positive second derivative for negative mean shifts. This results in extreme points of $\mathcal{C}$ when applied to signals with several positive and negative mean shifts.

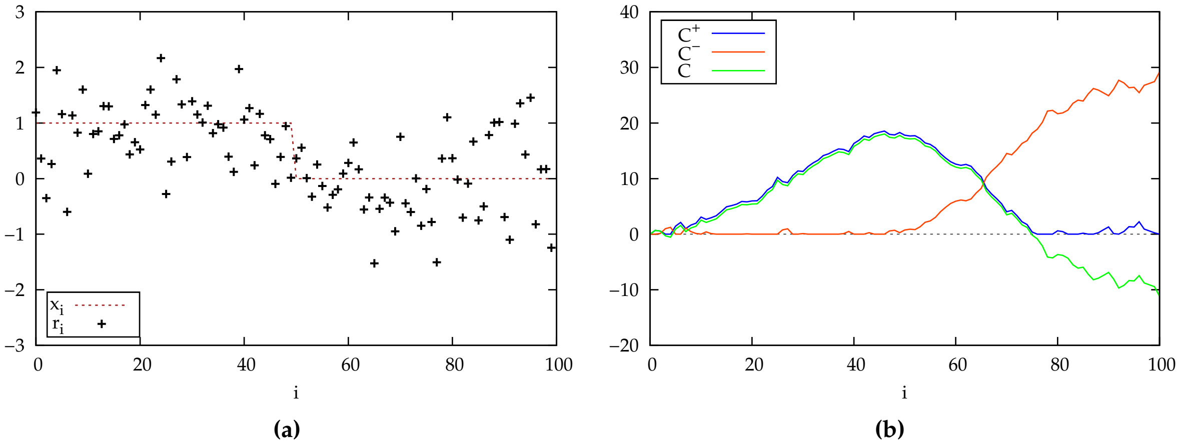


Figure 4.2: Evaluation of **(a)** a noisy signal, with SNR = 40dB and with a negative mean shift, and **(b)** their corresponding cumulative sums

In practice the mean shift is rarely known and the true shift is far away from δ, so the arbitrary choice of k like leads to poor performance. So the estimation of occurring shifts is of high importance to guarantee good performance of the cumulative sum statistic. Many techniques have been developed to handle the problem of an unknown δ. An example is the implementation of a weighting function $w(\delta)$, which leads to weighted cumulative sum charts [Yashchin 1989].

4.2 The Bootstrap — A Resampling Method

The resampling methods offer a way to solve problems in probability and statistics. In contrast to analytical formulation of problems the resampling methods analyze a statistical model with techniques that are intuitive. The statistical properties are estimated with repeated random sampling.

In statistics there are several methods of resampling. The most important are:

1. The *Jacknife* [Efron 1982] deals with estimation of statistical properties like means and variances by building subsets of observations.

2. The *Bootstrap* estimation of observation properties works by resampling with replacement.

3. The *permutation test* is a simple way to compute the sampling distribution for any test statistic, under the strong null hypothesis that a set of genetic variants has absolutely no effect on the outcome. More information on permutation test can be found in [Good 2000].

Assume a set of observations

$$\mathbf{r} = (r_1, r_2, \ldots, r_N) \sim F \tag{4.8}$$

of independent and identically distributed samples from an unknown probability distribution F.

The bootstrap was first introduced by [Efron 1979] with the goal to estimate the standard deviation σ of F. At that time it was presented as a computer based method to estimate statistical

properties of a data set. Its big advantage is that it is fully automatic and it does not matter how complicated the mathematical model for the probability distribution F is.
A bootstrap sample is defined as a random sample $\mathbf{r}^*$ of size N draw from $\mathbf{r}$, denoted as

$$\mathbf{r}^* = (r_1^*, r_2^*, \dots, r_N^*). \tag{4.9}$$

The bootstrap data set $\mathbf{r}^*$ consists of samples of $\mathbf{r}$ where some of them can occur twice or more times in $\mathbf{r}^*$. The procedure is called resampling of $\mathbf{r}$ with replacement.
With the bootstrap data set $\mathbf{r}^*$ the bootstrap standard deviation can now be calculated as

$$\sigma^* = \left[\frac{1}{N}\sum_{i=1}^{N}(r_i^* - \mu^*)^2\right]^{1/2} \tag{4.10}$$

with μ^* the expectation of the bootstrap sample data set $\mathbf{r}^*$ given by

$$\mu^* = \frac{1}{N}\sum_{i=1}^{N} r_i^*. \tag{4.11}$$

For each bootstrap sample $\mathbf{r}_b^*$ the standard deviation σ_b^* is calculated. The estimation of the standard deviation by bootstrap is then given by

$$\sigma_B = \left[\frac{1}{B}\sum_{b=1}^{B}(\sigma_b^* - \mu_B)^2\right]^{1/2} \tag{4.12}$$

with the expectation of standard deviations σ_b^*

$$\mu_B = \frac{1}{B}\sum_{b=1}^{B}\sigma_b^*. \tag{4.13}$$

The complete bootstrap estimation is implemented as Monte Carlo algorithm shown in Figure 4.3. Therefore the resampling process has to be done several times, for example $B = 1000$.

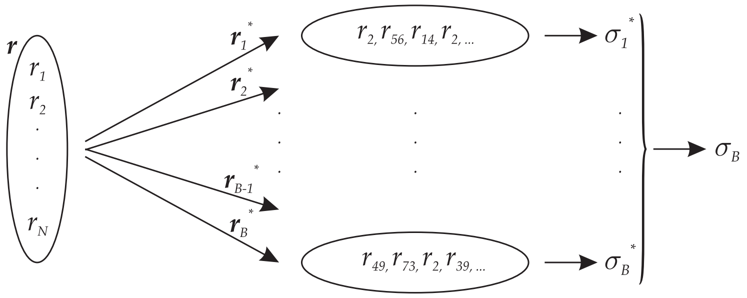


Figure 4.3: Schematics of bootstrap standard deviation estimate algorithm

By repeating the presented algorithm for infinite times, σ_B approaches σ, the standard deviation of F,

$$\lim_{B\to\infty}\sigma_B = \sigma. \tag{4.14}$$

The idea of bootstrap estimation can be extended to many other statistical properties, for example the sample mean.

But why use bootstraps if there are textbook formulas? Not all problems can be formulated in an analytical way easily. For a first approach, if the statistical model is unknown bootstrap is an easy way to access statistical properties.
A typical problem is presented in [Efron 1979]. A set of observations **r** from an unknown distribution function F with unknown statistical properties ψ is given. The task is to find the problem-identifying properties ψ^*. Note that ψ and ψ^* represent all possible properties. The bootstrap estimate is not only limited to means and deviations. For example it can also be used for linear regression developed in [Efron and Tibshirani 1993].
With the Monte Carlo algorithm shown in Figure 4.3, estimation of different properties becomes very easy. So the bootstrap offers a method to make statistical evaluations possible without knowing any analytical model of the problem.

4.3 Bootstrap and Cumulative Sum Detection Algorithm

To detect a change point in data, a combination of the presented methods is necessary. The schematic of the complete algorithm is shown in Figure 4.4.
Beginning with the observations **r**, bootstraps are created. Every bootstrap $\mathbf{r}_b^*$ is now conditioned in a cumulative sum chart with the shift threshold set to

$$k = \frac{1}{N}\sum_{i=1}^{N} r_{b,i}^*, \tag{4.15}$$

the mean of $\mathbf{r}_b^*$. After that, the difference of the cumulative sum statistics can be obtained by

$$\gamma_b^* = \max \mathcal{C}_b^* - \min \mathcal{C}_b^*. \tag{4.16}$$

The so obtained cumulative sum amplitudes γ_b^* are binned into a histogram $H(\gamma^*)$. The histogram is normalized to

$$\sum_{\gamma^*} H(\gamma^*) = 1. \tag{4.17}$$

With the introduction of a confidence value $\epsilon \in [0,1]$ the cumulative sum amplitudes reference level γ_ϵ is now determined such that

$$\sum_{\eta=\gamma^*_{\min}}^{\gamma_\epsilon} H(\eta) = \epsilon. \tag{4.18}$$

The question if the observations **r** contain a change point is answered by comparing the difference of the original values cumulative sum chart γ and the estimate γ_b. When

$$\gamma \leq \gamma_\epsilon \tag{4.19}$$

a change point occurs at the position where $|\mathcal{C}|$, the cumulative sum chart of the original values, has its maximum value. Otherwise the observations **r** do not show a significant change in mean for the desired confidence interval ϵ and the so estimated threshold γ_ϵ.
The discussed procedure is able to detect one change point. If there is one change point in **r** the vector of observations is split in two parts at the change point position to

$$\mathbf{r} = [\mathbf{r_1}, \mathbf{r_2}]. \tag{4.20}$$

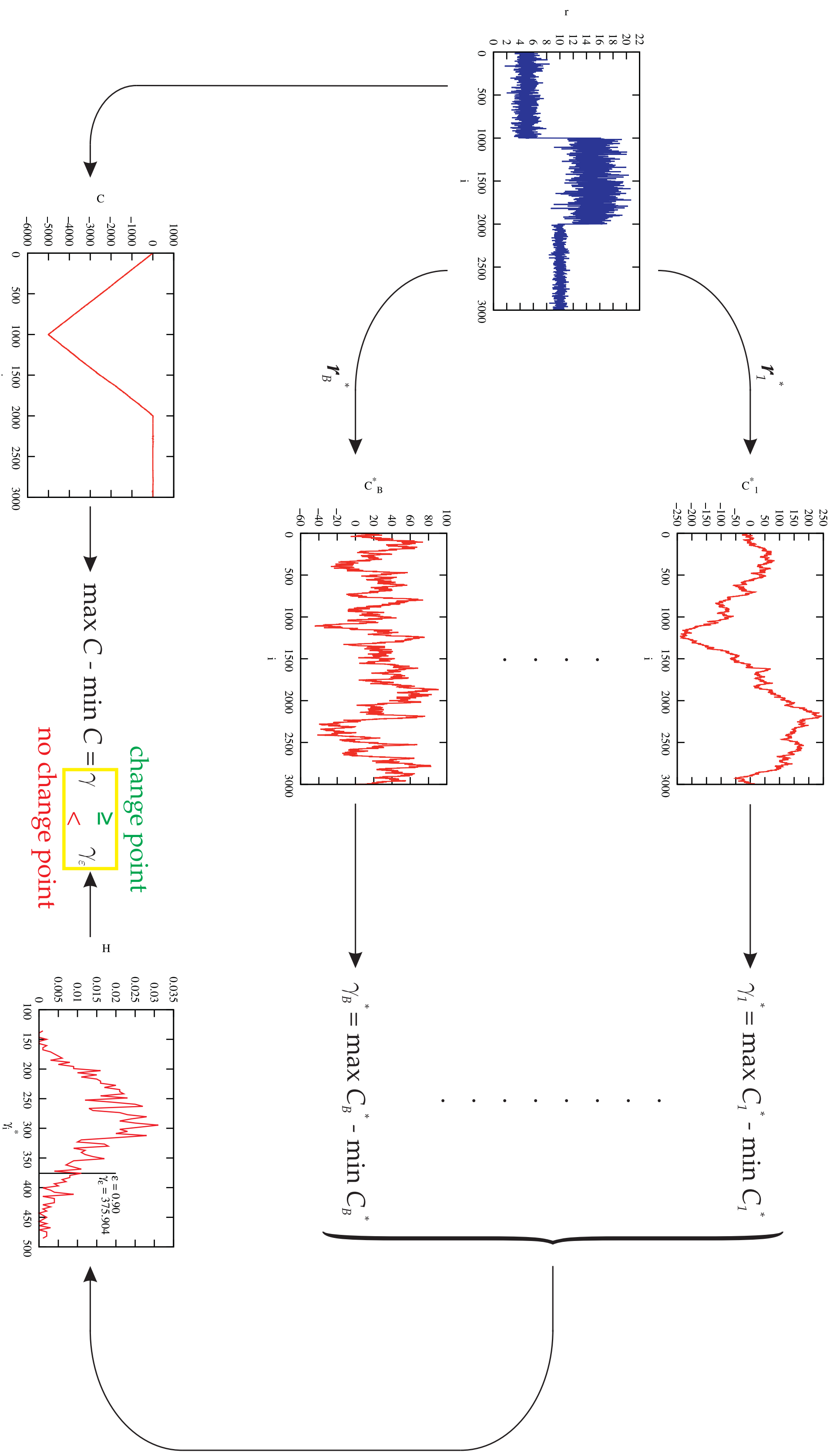

r
i
r_1^*
r_B^*
C^*_1
C^*_B
C
$\gamma_1^* = \max C_1^* - \min C_1^*$
$\gamma_B^* = \max C_B^* - \min C_B^*$
H
γ_i^*
$\varepsilon = 0.90$
$\gamma_\varepsilon = 375.904$
max C - min C = γ ≥ γ_ε
max C - min C = γ < γ_ε
change point
no change point

Figure 4.4: Detailed schematic of detection of a single change point with the bootstrap and cumulative sum method

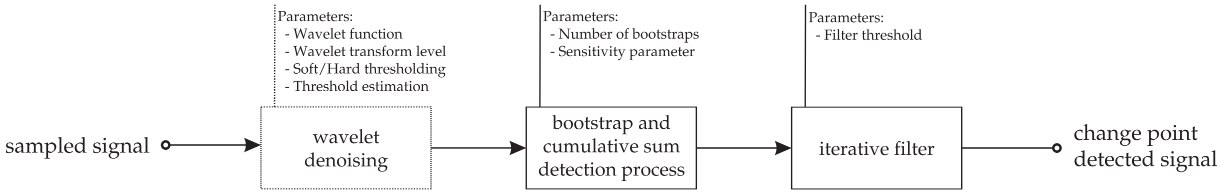


Figure 4.5: Structural overview of bootstrap and cumulative sum statistic change point detection algorithm

Now the new vectors $\mathbf{r_1}$ and $\mathbf{r_2}$ are analyzed if they contain a change point and so on until all change points are detected. The divide and conquer algorithm of change point detection of a hole data set $\mathbf{r}$ can be summarized to:

1. Check if there is a change point in $\mathbf{r}$.

2. If there is one split $\mathbf{r}$ into two vectors at the change point position, if not the process is ready and continue with 4..

3. Resume with step 1 for $\mathbf{r_1}$ and $\mathbf{r_2}$.

4. Finally the observation vector $\mathbf{r}$ is of the form

$$\mathbf{r} = [\mathbf{r_1}, \mathbf{r_2}, \dots, \mathbf{r_P}] \tag{4.21}$$

with P the number of detected change points.

To create a step function out of $\mathbf{r}$ the means of the vectors $\mathbf{r_1}, \mathbf{r_2}, \dots, \mathbf{r_P}$ are calculated. A new vector $\mathbf{s}$ is created by appending the corresponding means μ_{r_i} the vectors N_{r_i} times,

$$\mathbf{s} = [\underbrace{\mu_{\mathbf{r_1}}, \dots, \mu_{\mathbf{r_1}}}_{N_{r_1}}, \underbrace{\mu_{\mathbf{r_2}}, \dots, \mu_{\mathbf{r_2}}}_{N_{r_2}}, \mu_{\mathbf{r_3}}, \dots\dots, \mu_{\mathbf{r_{P-1}}}, \underbrace{\mu_{\mathbf{r_P}}, \dots \mu_{\mathbf{r_P}}}_{N_{rP}}]. \tag{4.22}$$

The detection process is given in Figure 4.5. After an optional denoising step, cf. Chapter 2, the bootstrap and cumulative sum statistic algorithm is applied. The parameters for the bootstrap and cumulative sum detection process are:

- *Number of bootstraps*: Set the number of bootstrap samples generated. A value $B \geq 10^3$ is preferred.

- *Sensitivity parameters*: The sensitivity parameter $\epsilon \in [0,1]$ sets the decision level between change point and no change point.

Finally an application specified filtering process designed in Section 2.8 is applied.
It must be noted that the detection result is not always exactly the same because of the random process of resampling. For the results to be repeatable the number of bootstraps should at least be $B \geq 10^3$. For small data sets or enough time for analysis values of $B = 10^4$ or $B = 10^5$ should be preferred.

4.4 Application of Bootstraps and Cumulative Sum Statistics to Charge Trap Measurements

Now some examples of application to charge trap measurements are presented[1]. First the algorithm is applied on observations without previous wavelet denoising. After that the effect of denoising in combination with the bootstrap and cumulative sum statistic detection algorithm is shown.

4.4.1 Evaluation of Basic Bootstrap and Cumulative Sum Algorithm

An example of change point analysis with the bootstrap and cumulative sum algorithm is illustrated in Figure 4.6. This result is obtained with a number of bootstrap samples $B = 1000$, the sensitivity parameter set to $\epsilon = 0.8$ and the iterative filter parameter $\kappa = 0.6$. The choice of $\epsilon = 0.8$ leads to a very sensitive detection process resulting in an SNR = 89.05dB. For higher values of ϵ less steps are detected. This circumstance leads to a smaller value for the signal to noise ratio.

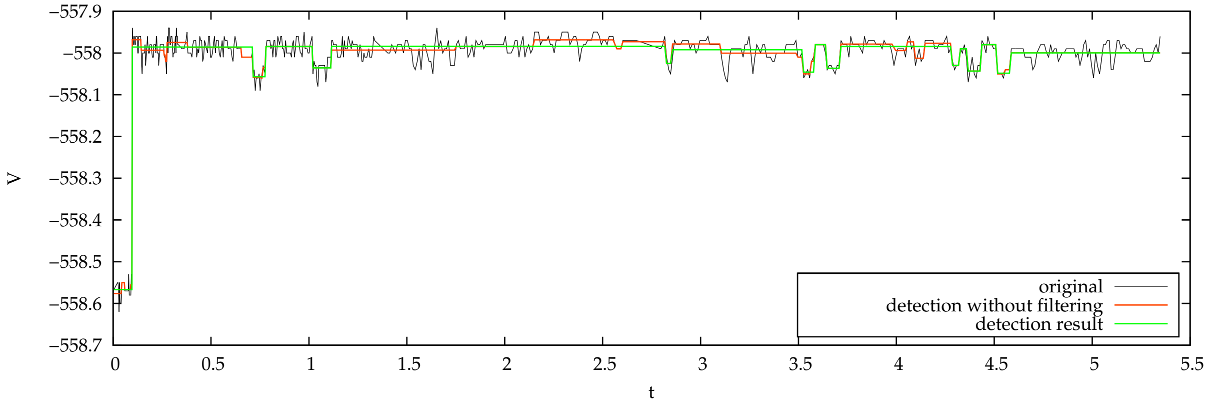


Figure 4.6: Bootstrap and cumulative sum change point detection without previously denoising of the signal, the number of bootstraps $B = 1000$, the sensitivity parameter $\epsilon = 0.8$ and the iterative filtering parameter $\kappa = 0.6$

The effect of the sensitivity parameter $\epsilon \in [0, 1]$ is studied with the help of the signal from Figure 4.6. Detection results obtained by varying ϵ are shown Figure 4.7. The detection sensitivity strongly depends on the choice of ϵ. For a smaller value the count of detected steps increases. There is no formula to estimate the optimal choice of ϵ. So a convenient value for ϵ which fits best with the required step detection sensitivity has to be selected.

The choice $\epsilon = 1$ also leads to detection results because of the probabilistic character of the bootstraps. Setting the sensitivity parameter to his maximum acceptable value delivers only a few change points of a signal. Small peaks, consisting of only a few data points, will not be detected.

A detection process of steps in a very noisy signal is shown in Figure 4.8 resulting in an SNR = 72.44dB. On heavily noisy signals the bootstrap and cumulative sum algorithm works quite

[1] The definition of the signal to noise ratio and the mean square error used to emphasize the denoising quality is given in Section A.4.

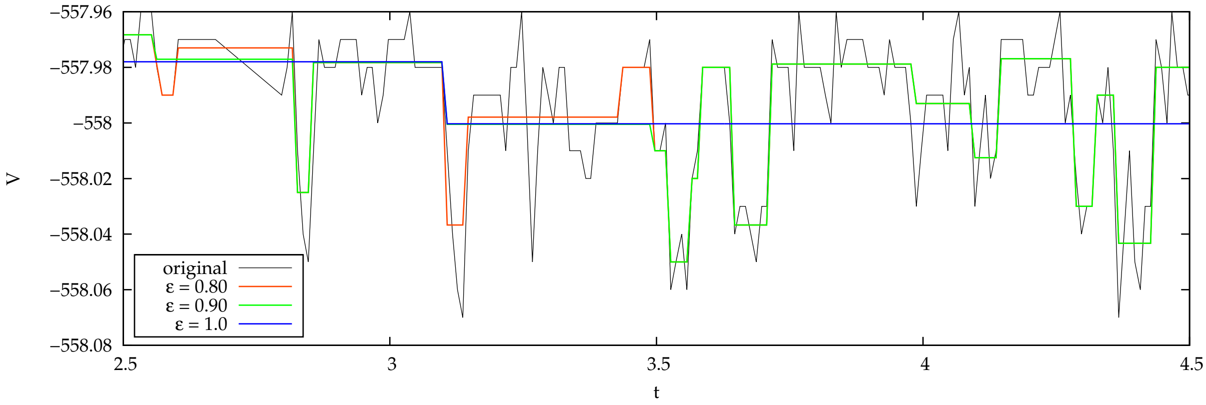


Figure 4.7: Influence of the sensitivity parameter on the detection result for signal presented in Figure 4.6

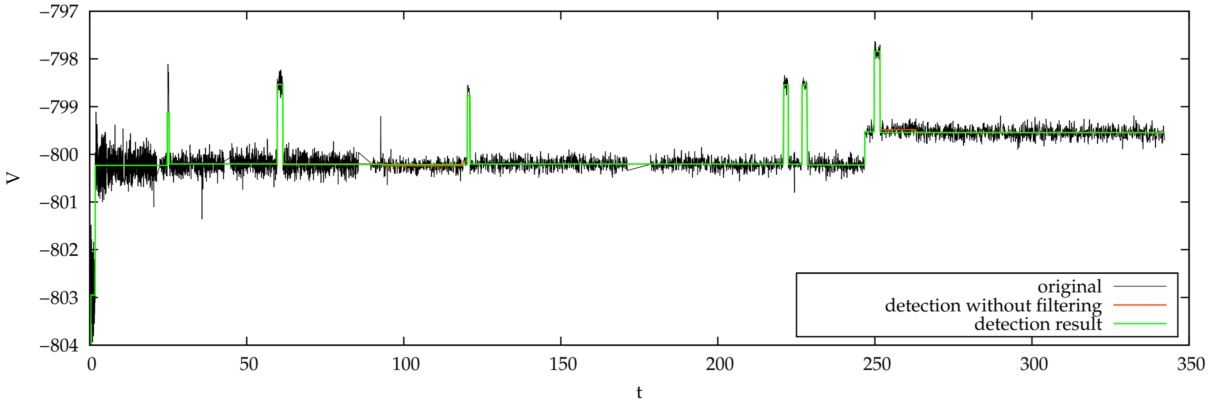


Figure 4.8: Change point detection of a very noisy signal by bootstrap and cumulative sum detection algorithm without previously denoising of the signal, the number of bootstraps $B = 1000$, the sensitivity parameter $\epsilon = 1$ and the iterative filtering parameter $\kappa = 1$

well. All relevant steps are detected with the parameters $\epsilon = \kappa = 1$. A previous signal denoising step is not necessary with this detection method.

The step detection of a nearly noiseless signal is shown in Figure 4.9. The resulting signal to noise ratio is SNR = 78.24dB. With the sensitivity parameter $\epsilon = 1.0$ and the filter parameter $\kappa = 1$ the two major steps in this signal are detected. The bootstrap parameter is set to $B = 50$. For significant steps like the two illustrated in Figure 4.9 a small choice of B leads to very fast detection results. A comparison of the unfiltered and filtered detection signal shows that a final filter process is necessary. Otherwise very small mean changes are detect too. In the case of nearly noiseless signals the bootstrap and cumulative sum algorithm is very efficient.

4.4.2 Detection of Wavelet Denoised Signals

Now the influence of signal denoising discussed in Chapter 2 on the bootstrap and cumulative sum statistic algorithm is presented. The result of the bootstrap and cumulative sum change point detection after denoising using the wavelet transform for the signal from Figure 4.6 is shown in Figure 4.10 resulting in an SNR = 85.48dB. A comparison between Figure 4.6 and Figure 4.10 shows a very similar detection result in both cases for the same bootstrap and cumu-

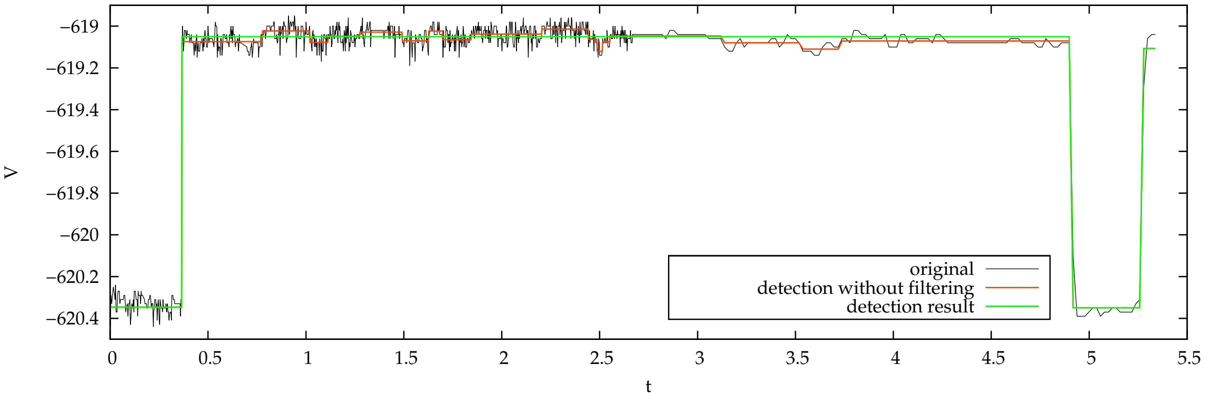


Figure 4.9: Detection of steps in a nearly noiseless signal by bootstrap and cumulative sum detection algorithm without previously denoising of the signal, the number of bootstraps $B = 50$, the sensitivity parameter $\epsilon = 0.9$ and the iterative filtering parameter $\kappa = 1$

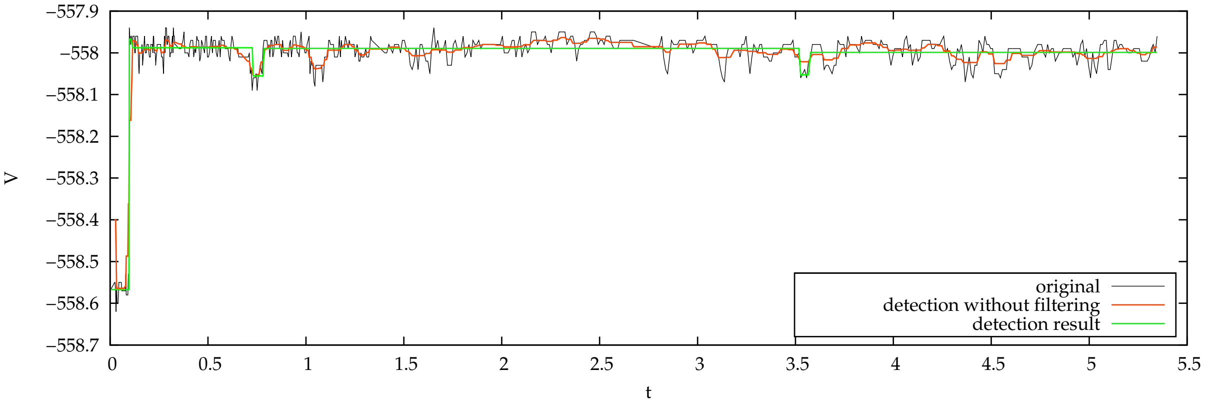


Figure 4.10: Bootstrap and cumulative sum statistic algorithm with wavelet denoising using the stationary discrete wavelet transform, wavelet transform level $J = 3$, Haar wavelet, soft thresholding with modified Bayes shrink threshold estimation, the number of bootstraps $B = 1000$, the sensitivity parameter $\epsilon = 0.8$ and the iterative filtering parameter $\kappa = 0.6$

lative sum detection configuration. In summary, the wavelet denoising process is not necessary for change point detection of time dependent defect spectroscopy signals with the bootstrap and cumulative sum detection algorithm.

CHAPTER 5

STATISTICAL EVALUATION OF TIME DEPENDENT DEFECT SPECTROSCOPY DATA

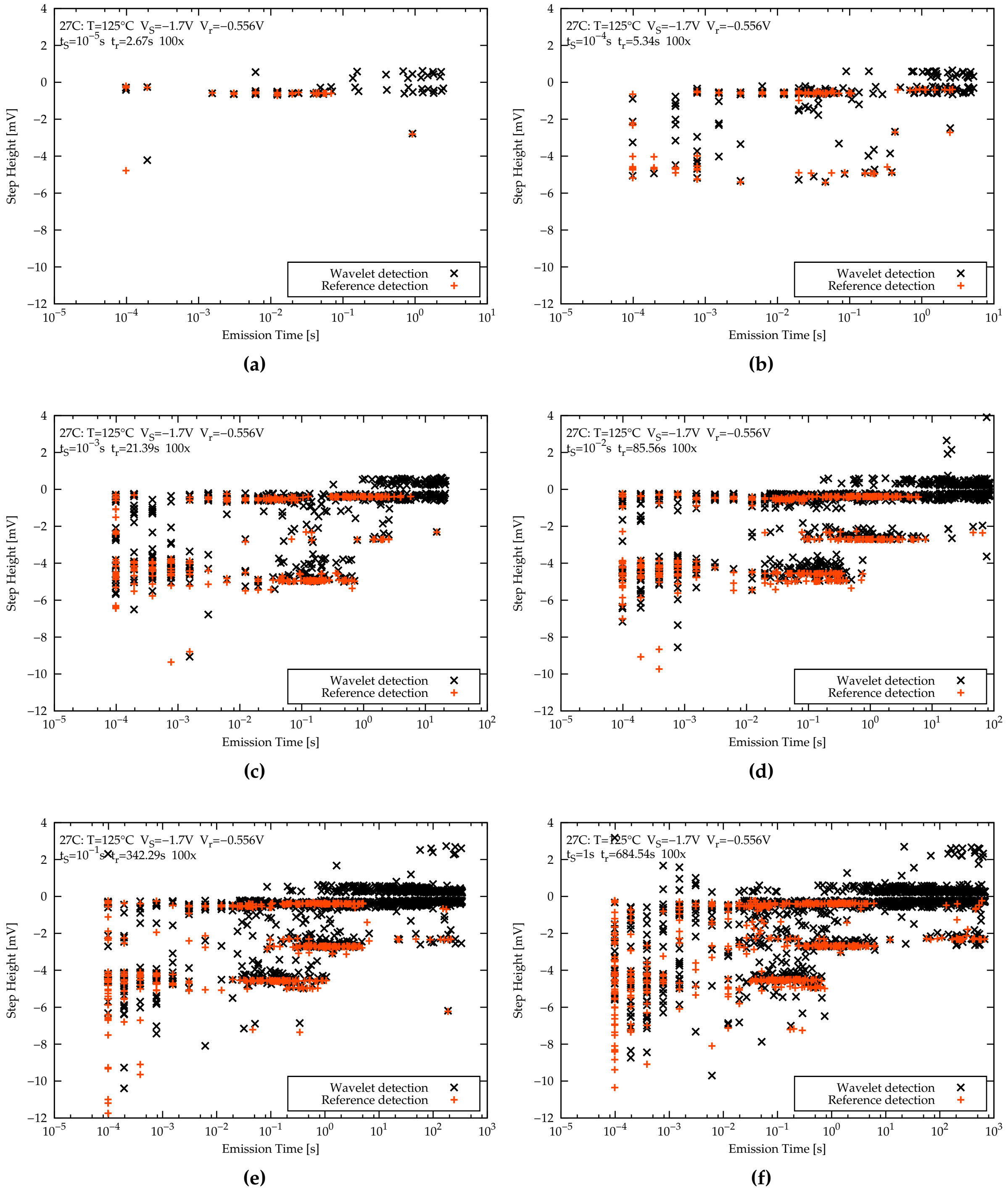
The results of data extraction, step heights and emission times, from time defect dependent spectroscopy data of a p-MOSFET are now presented. All detected steps are displayed as points in a step height d versus emission time τ_e chart called a *spectral map*. The detection results of each algorithm are compared to reference detection results. Finally the differences between the spectral maps obtained from each of the three methods are discussed.

5.1 Wavelet Denoising Change Point Detection

The wavelet denoising detection algorithm configuration for the calculated spectral maps is (command line options are denoted in parentheses):

- *Wavelet transform*: Stationary discrete wavelet transform (`-w s`)
- *Wavelet transform level*: $J = 3$ (`-l 3`)
- *Wavelet type*: Haar wavelet (`-y haar`)
- *Threshold type*: Soft thresholding (`-d soft`)
- *Threshold estimation*: Histogram-based threshold estimation with $\gamma = 0.999$ (`-s histogram -g 0.999`)
- *Transient time*: $t_\mathrm{trans} = 2 \cdot 10^{-5}\mathrm{s}$ (`-t 2e-5`)
- *Filter parameter*: $\kappa = 0$ (`-f 0`)
- *Minimum step height written to switch files*: $m = 0.2$ (`-m 0.2`)

The configuration of the detection process is set to highest denoising efficiency. Extracted step heights d and emission times τ_e for several stress times detected by the wavelet denoising detection algorithm and the reference steps are shown in Figure 5.1. The detected steps for stress time $t_\mathrm{S} = 10^{-5}\mathrm{s}$ are shown in Figure 5.1a. All the steps given by the reference data are detected by the wavelet denoising method. The deviation in the emission time in case of the change point at position $(\tau_\mathrm{e}, d) \approx (10^{-4}\mathrm{s}, 4\mathrm{mV})$ results from the detection process: The emission time is calculated



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.1: Spectral maps obtained by the wavelet denoising detection algorithm for stress times **(a)** $t_\mathrm{S} = 10^{-5}$s, **(b)** $t_\mathrm{S} = 10^{-4}$s, **(c)** $t_\mathrm{S} = 10^{-3}$s, **(d)** $t_\mathrm{S} = 10^{-2}$s, **(e)** $t_\mathrm{S} = 10^{-1}$s and **(f)** $t_\mathrm{S} = 1$s.

as averages before and after the change point. The detected step height d may differ from the reference data because the algorithms are designed to detect mean shifts. Between two change points the signal level is approximated by the average of the data. Additional small steps are detected at $\tau_e \approx 1$s.

For a stress time of $t_S = 10^{-4}$s Figure 5.1b presents the resulting spectral map. Many more steps compared to the case of $t_S = 10^{-5}$s are found now. The detection results matches the reference data very well. A lot of small steps are detected for emission time $\tau_e \approx 1$s. The step heights and emission times also compare well to the results from the stress case $t_S = 10^{-5}$s. The emission centers are all nearly the same, now with more detected data points. The detected results go hand in hand with the fact that a increasing stress time does not shift the emission time. More steps at different step heights are observed then.

The evaluation of relaxation measurements for stress time $t_S = 10^{-3}$s, $t_S = 10^{-2}$s, $t_S = 10^{-1}$s and $t_S = 1$s are given in Figure 5.1c, Figure 5.1d, Figure 5.1e and Figure 5.1f respectively. It is noticed that the clusters of the spectral map are similar to the reference data clusters.

The disadvantage of the wavelet denoising detection technique is the high number of detected steps of relaxation measurements at higher stress times. The spectral map for a stress time of $t_S =$ 1s, cf. Figure 5.1f, is very noisy. Detecting clusters in very noisy spectral map gets difficult. The presented results are obtained by the most efficient denoising configuration. The advantage of the wavelet denoising detection algorithm is that it works quite fast compared to the histogram-based and bootstrap and cumulative sum method.

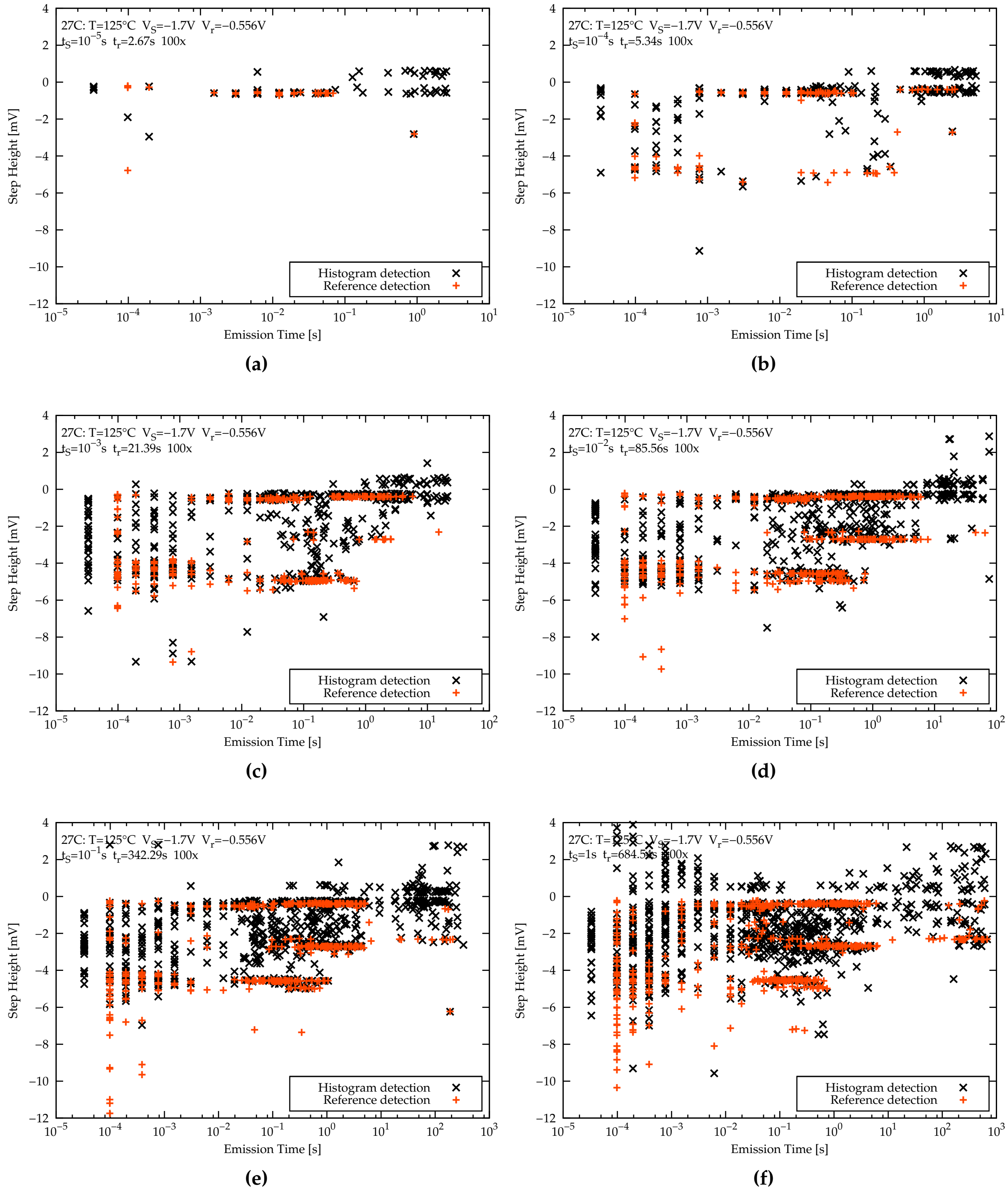
5.2 Histogram-Based Detected Steps

The spectral maps obtained by executing the histogram-based detection algorithm on time dependent defect spectroscopy data are given in Figure 5.2. The detection configuration is (command line options are denoted in parentheses):

- *Number of bins*: Estimate optimal number of bins automatically (`-b -1`)
- *Interpolation type*: Polynomial with 113 points (`-i poly -k 113`)
- *Peak merge distance*: $\epsilon = 0.3$ (`-e 0.3`)
- *Transient time*: $t_{\text{trans}} = 2 \cdot 10^{-5}$s (`-t 2e-5`)
- *Filter parameter*: $\kappa = 0$ (`-f 0`)
- *Minimum step height written to switch files*: $m = 0.2$ (`-m 0.2`)

In all four presented spectral maps steps at emission times $\tau_e < 10^{-4}$s are detected. They result from the transient part of the signal before the relaxation is recorded. It is necessary to choose t_{trans} carefully.

For stress time $t_S = 10^{-5}$s the result is shown in Figure 5.2a. Except for one all steps from the reference data are detected and several small more at higher emission times. With increasing stress time more steps are detected, cf. Figure 5.2b for $t_S = 10^{-4}$s. The reference steps and several more smaller steps are obtained by the histogram-based algorithm. At relaxation measurements after device stress of $t_S = 10^{-3}$s at Figure 5.2c and $t_S = 10^{-2}$s, see Figure 5.2d, many steps with varying step height in the range $3 \times 10^{-2}\text{s} \leq \tau_e \leq 10^1$s are obtained by the detection process.



(a) (b) (c) (d) (e) (f)

Figure 5.2: Results of step detection by the histogram-based algorithm of time dependent defect spectroscopy measurement data for stress times **(a)** $t_S = 10^{-5}$s, **(b)** $t_S = 10^{-4}$s, **(c)** $t_S = 10^{-3}$s, **(d)** $t_S = 10^{-2}$s, **(e)** $t_S = 10^{-1}$s and **(f)** $t_S = 1$s.

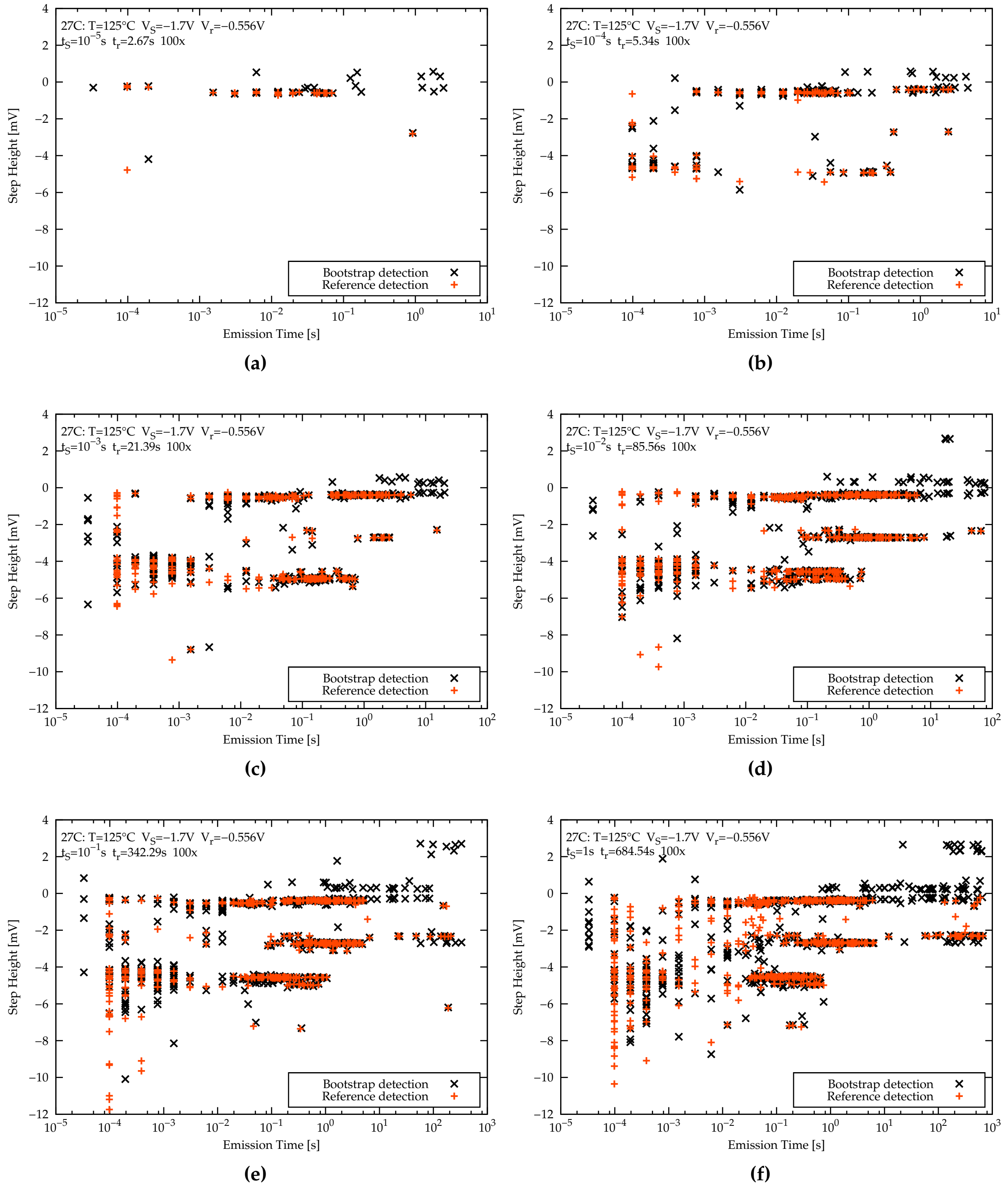
Also many steps are detected in relaxation measurement data for stress times $t_S = 10^{-1}$s and $t_S = 1$s, cf. Figure 5.2e and Figure 5.2f.
The histogram-based detected spectral maps are very noisy. Especially at higher emission times at step heights $d \approx 0$ making the detection result not practical for the time dependent defect spectroscopy data evaluation.

5.3 Bootstrap and Cumulative Sum Detection Evaluation

The bootstrap and cumulative sum detection algorithm configuration is (command line options are denoted in parentheses):

- *Number of bootstraps*: $B = 100$ (`-b 100`)
- *Sensitivity parameter*: $\epsilon = 0.81$ for $t < 10^{-2}$s and $\epsilon = 1$ for $t \geq 10^{-2}$s (`-e 0.81,1e-2,1.0`)
- *Wavelet transform*: Stationary discrete wavelet transform (`-w s`)
- *Wavelet transform level*: $J = 3$ (`-l 3`)
- *Wavelet type*: Haar wavelet (`-y haar`)
- *Threshold type*: Soft thresholding (`-d soft`)
- *Threshold estimation*: Histogram-based threshold estimation with $\gamma = 0.86$ (`-s histogram -g 0.86`)
- *Transient time*: $t_{\text{trans}} = 2 \cdot 10^{-5}$s (`-t 2e-5`)
- *Filter parameter*: $\kappa = 0$ (`-f 0`)
- *Minimum step height written to switch files*: $m = 0.2$ (`-m 0.2`)

The detection algorithm results for several stress times t_S are shown in Figure 5.3. The first evaluation in Figure 5.3a for $t_S = 10^{-5}$s almost equals the reference data. At emission times $\tau_e > 10^{-1}$s several small steps are additionally detected. At $\tau_e < 10^{-3}$s only a few measurement points are available. In order to detect steps for emission times $\tau_e < 10^{-3}$, a small value for the sensitivity parameter $\epsilon = 0.81$ is necessary. A small sensitivity parameter goes hand in hand with a high number of detected steps. The difference in detected step emission time at $\tau_e = 10^{-4}$s and step height $d \approx -4$mV results from the detection process. The emission time is calculated by the average time before and after a detected step. A visual inspection shows that this step occurs at the time instance detected with the bootstrap and cumulative sum algorithm.
The next relaxation data from Figure 5.3b is recorded after stressing the device for $t_S = 10^{-4}$s. It is noted that a few steps show different heights as the one from the reference evaluation. The difference is because the bootstrap and cumulative sum algorithm detects mean shifts. All data values between two steps are averaged. So differences of about 0.5mV are possible in the detected step heights compared to the reference data. For emission times $\tau_e > 5 \cdot 10^{-1}$s several small steps are detected resulting from the sensitivity parameter $\epsilon = 0.81$.
Because of the increasing number of detected steps at higher emission time a higher value for the sensitivity parameter ϵ is necessary. The bootstrap and cumulative sum option can change



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 5.3: Spectral maps obtained by the bootstrap and cumulative sum detection algorithm for device stress times **(a)** $t_\mathrm{S} = 10^{-5}$s, **(b)** $t_\mathrm{S} = 10^{-4}$s, **(c)** $t_\mathrm{S} = 10^{-3}$s, **(d)** $t_\mathrm{S} = 10^{-2}$s, **(e)** $t_\mathrm{S} = 10^{-1}$s and **(f)** $t_\mathrm{S} = 1$s.

ϵ at a given time instance during the detection process. A change from $\epsilon = 0.81$ to $\epsilon = 1$ at $t = 10^{-2}$s greatly improves the quality of the calculated spectral maps.

After stressing the device for $t_S = 10^{-3}$s and recording the relaxation for one-hundred times the bootstrap and cumulative sum algorithm obtains the spectral map from Figure 5.3c. As mentioned above the very sensitive choice of $\epsilon = 0.81$ leads to detected steps at emission times $\tau_e < 10^{-4}$. Most of the steps equal the reference ones. At emission times $\tau_e > 10^1$ several small steps are detected with the sensitivity parameter $\epsilon = 1$. The spectral maps for the stress times $t_S = 10^{-2}$s, $t_S = 10^{-1}$s and $t_S = 1$s are shown in Figure 5.3d, Figure 5.3e and Figure 5.3f respectively. Compared to Figure 5.3c almost the same clusters of points are obtained. The difference of the reference steps and the steps obtained by the bootstrap and cumulative sum algorithm is shown in Figure 5.4.

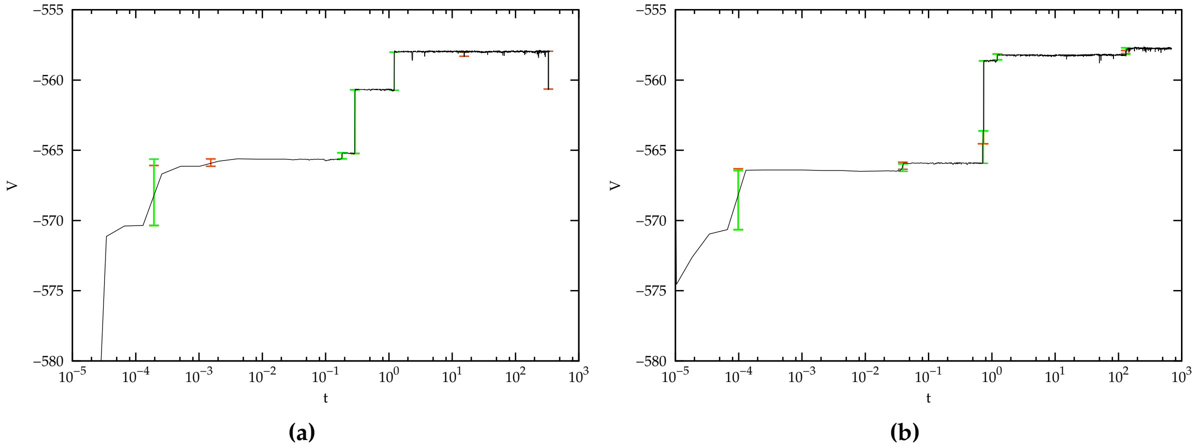


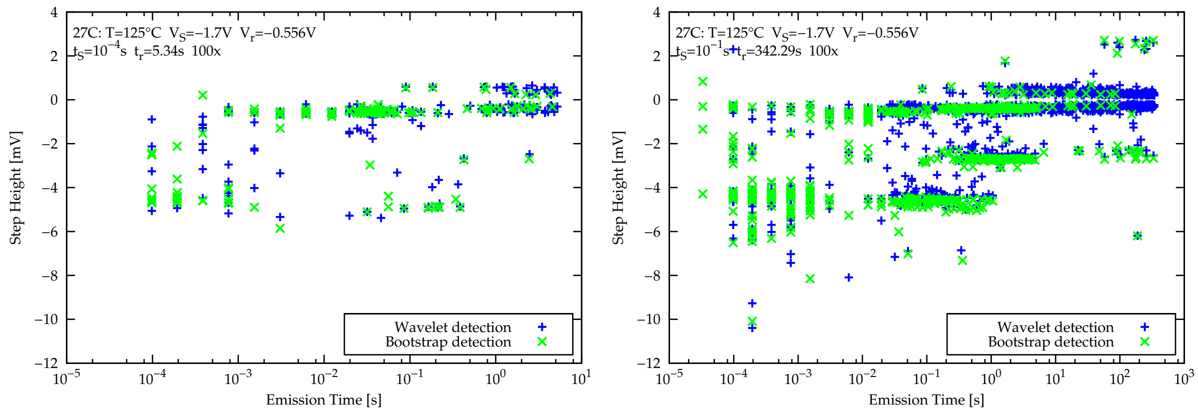
Figure 5.4: Difference between reference steps (green) and steps detected with the bootstrap and cumulative sum algorithm (orange) for a single signal (black) from measurements for stress times **(a)** $t_S = 10^{-1}$s, **(b)** $t_S = 1$s.

5.4 Comparison of Detection Results

First the differences of the wavelet denoising and bootstrap and cumulative sum detection algorithm are discussed. The spectral maps are given in Figure 5.5a for the stress time $t_S = 10^{-4}$s and for $t_S = 10^{-1}$s in Figure 5.5b.

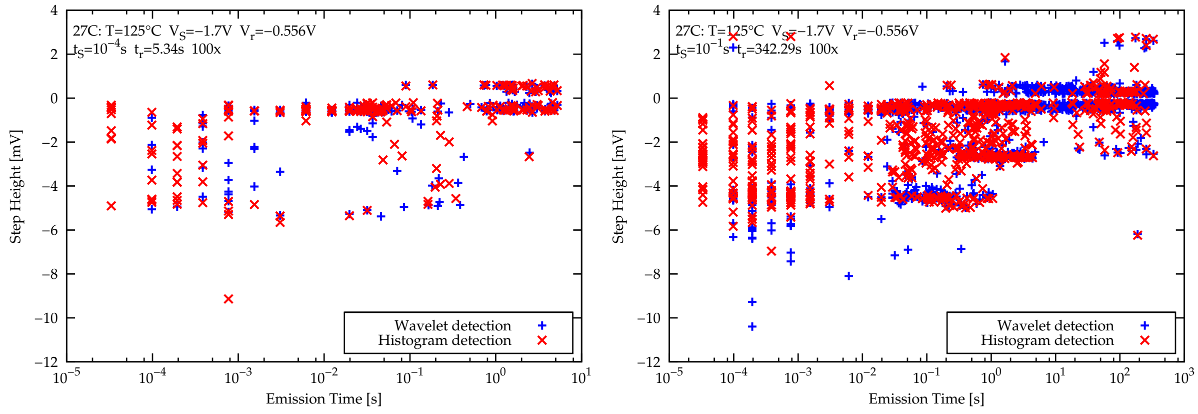
One difference is the detection of events for emission times $\tau_e \leq 10^{-4}$s. Because of the high detection sensitivity of the bootstrap and cumulative sum algorithm also fast events are detected. For higher emission times the detection results are similar. With the wavelet denoising technique the step amplitudes vary a little bit more than the ones detected with bootstrapping. The advantage of the wavelet denoising detection algorithm is that it works very fast compared to the bootstrap and cumulative sum method.

The differences of the detection results of the wavelet denoising technique and the histogram-based detection algorithm are shown in Figure 5.6. A comparison of both methods shows that with the histogram-based algorithm more steps are detected as with the wavelet detection algorithm. Another difference is that the histogram-based detection spectral map shows steps at emission times $\tau_e < 10^{-4}$s. They result from the transient part of the relaxation signal.



(a) **(b)**

Figure 5.5: Comparison of the spectral maps obtained by the wavelet denoising and bootstrap and cumulative sum detection algorithm from Figure 5.1 and from Figure 5.3 for the stress times **(a)** $t_S = 10^{-4}$s and **(b)** $t_S = 10^{-1}$s.

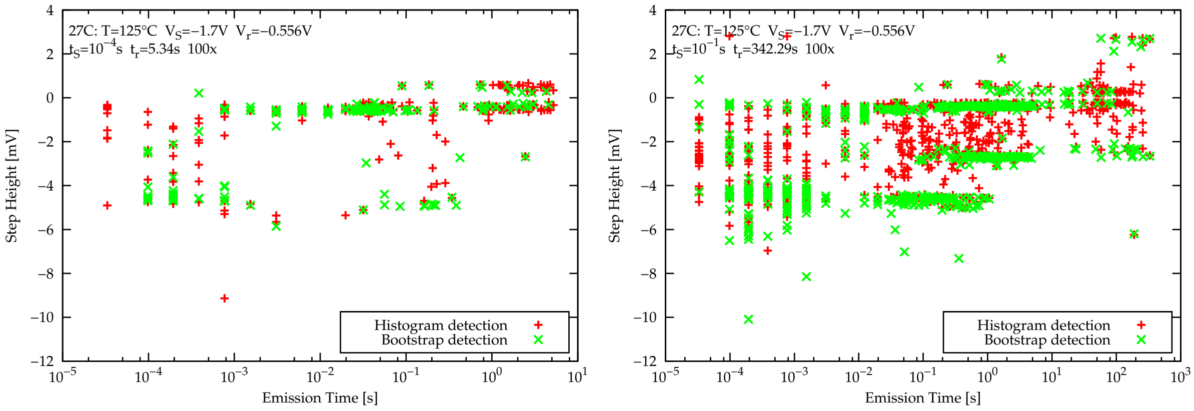


(a) **(b)**

Figure 5.6: Spectral maps of the wavelet denoising and histogram-based detection algorithm from Figure 5.1 and Figure 5.2 for the stress times **(a)** $t_S = 10^{-4}$s and **(b)** $t_S = 10^{-1}$s.

The wavelet denoising algorithm does not detect such single data point events. The choice of $t_{\text{trans}} = 2 \cdot 10^{-5}$s for the algorithm configuration does not cut off all transients in all signals. Higher choices of t_{trans} may remove relevant change points from the observations. Also the detected step heights vary much more at the histogram-based detection algorithm, cf. Figure 5.6b.

Finally the comparison of the histogram-based and the bootstrap and cumulative sum detection results are presented in Figure 5.7. Both spectral maps show steps detected at emission times $\tau_e < 10^{-4}$s resulting from previously mentioned signal transients. The choice of $t_{\text{trans}} = 2 \cdot 10^{-5}$s is necessary for both algorithms. The detection results for stress time $t_S = 10^{-4}$s from Figure 5.7a are very similar. The main difference is the detected step heights in the emission time range $10^{-1}\text{s} \leq \tau_e \leq 10^{1}\text{s}$, cf. Figure 5.7b.



(a) **(b)**

Figure 5.7: Spectral maps obtained by the bootstrap and cumulative sum and the histogram-based detection algorithm from Figure 5.3 and Figure 5.2 for the stress times **(a)** $t_S = 10^{-4}$s and **(b)** $t_S = 10^{-1}$s.

5.5 Summary

One difference between the three methods are their configuration parameters. The wavelet denoising technique has the most parameters, followed by the histogram-based method. Only two configuration parameters are available with the bootstrap and cumulative sum algorithm. Another difference is the dependence of the execution time on the measurement signals. An overview is given in Table 5.1.

	N	Files	Wavelet	Histogram	Bootstrap and cumulative sum
			Figure 5.1	Figure 5.2	Figure 5.3
(a)	454	100	15s	13m 39s	3m 16s
(b)	710	100	26s	19m 02s	4m 32s
(c)	1478	100	1m 02s	42m 31s	9m 38s
(d)	3014	100	2m 44s	1h 33m 01s	21m 46s
(e)	6086	100	8m 35s	4h 24m 08s	54m 23s
(f)	9134	100	11m 19s	6h 29m 51s	1h 16m 20s

Table 5.1: Overview of execution time for the wavelet denoising, the histogram-based and the bootstrap and cumulative sum algorithm measurement data signals of length N. For creating a spectral map the algorithms have to be applied on 100 data sets. All evaluation configurations are identical to the ones used in Section 5.1, Section 5.2 and Section 5.3 respectively.

One notices that the wavelet denoising method is the fastest one. The execution time difference between the stationary discrete wavelet transform and the discrete wavelet transform is negligible and the increase of the algorithm execution time with increasing number of data points of a signal is intuitively clear. A disadvantage is the high number of degrees of freedom given by four parameters.

The longest execution time is observed with the histogram-based method. The reason for high execution times of the histogram-based method is the interpolation of data points between ob-

servations. A good choice is 113 interpolation points. Without data interpolation the algorithm works quite fast but the results have not the quality as the ones with polynomial interpolation.
The high execution of time the bootstrap and cumulative sum detection algorithm results from the necessity to calculate bootstrap samples with the same length as the data set. This means for $B = 100$ and a signal length of $N = 1000$ samples the cumulative sum charts for 10^5 samples have to be considered. Because of the divide-and-conquer process a higher number of detected steps increases the execution time. For each additionally detected change point B bootstraps and cumulative sum charts are calculated.
Summarized, the best detection results are obtained by the bootstrap and cumulative sum detection algorithm. With an acceptable execution time all the steps given by the reference data set are detected and the clusters are easily found in the spectral map. The advantage of the bootstrap and cumulative sum detection algorithm is that the detection sensitivity can be changed during the step detection process of a measurement signal. With less detection sensitivity not so many steps at higher emission times are produced compared to the wavelet and histogram-based method. The wavelet denoising technique works quite fast, but for higher stress times too many steps are produced. Therefore it is not practical for this class of signals. Really disappointing is the histogram-based method. There are too many spurious steps detected and the spectral maps are too noisy. So this method in not recommended for data extraction of time dependent defect spectroscopy measurement data.

APPENDIX **A**

MATHEMATICAL AND PROGRAMMING TOOLS

Some basics of creating noisy test signals, determining endianess of a given computer architecture and floating point representation is discussed now.

A.1 Creating Noisy Signals

For many algorithms it is very useful to create noisy test signals with a specified signal to noise ratio (SNR). Usually the SNR is given in decibel (dB).
Assume the following model for a discrete signal

$$r_i = x_i + v_i \tag{A.1}$$

with x_i the noise free signal samples and the noise v_i. In this model x_i is known and now observations r_i with a specified SNR are created.
The SNR is defined by

$$\mathrm{SNR} = \frac{P_{\text{Signal}}}{P_{\text{Noise}}}. \tag{A.2}$$

From a given SNR in dB, the desired noise power is calculated from

$$P_{\text{Noise}} = P_{\text{Signal}} \cdot 10^{-\frac{\mathrm{SNR}}{10}}. \tag{A.3}$$

In vector notation

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \tag{A.4}$$

the signal power P_{Signal} is given by

$$P_{\text{Signal}} = \mathbf{x}^T\mathbf{x}. \tag{A.5}$$

We assume white Gaussian noise $\mathcal{N}(0, \sigma^2)$ with the distribution function

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right). \tag{A.6}$$

The SNR then calculates to

$$\mathrm{SNR} = \frac{P_{\text{Signal}}}{\sigma^2}. \tag{A.7}$$

With the use of a random number generator, noisy test signals for algorithm verification can be created.

A.2 Write Endianess Independent Code

In order to make files compatible to different CPU architectures endianess is very important. A definition is given in [Intel 2004]:

Definition A.1 (Endianess) *Endianess is the format to how multi-byte data is stored in computer memory. It describes the location of the most significant byte (MSB) and least significant byte (LSB) of an address in memory. Endianess is dictated by the CPU architecture implementation of the system. The operating system does not dictate the endian model implemented, but rather the endian model of the CPU architecture dictates how the operating system is implemented.*

According to the definition, two format types are distinguished, the so called *big-endian* and *little-endian* formats. The *big-endian* memory format stores the most significant byte at the lowest and the least significant byte at the highest memory address. As it is intuitively clear the *little-endian* stores the least significant byte at the lowest and the most significant byte at the highest memory address.

For binary data operation, especially writing or coding values in binary format to and from files, it is very important to know if the data is in *little-endian* or *big-endian* format. The binary curve file format described in Section C.1 uses the *little-endian* model.

In Python values can be easily stored byte-wise with the *pack* and read byte-wise with the *unpack* command. In both cases the programmer is able to specify the type of endianess. By default, and in all programs created in context with this thesis, the *little-endian* format is used.

A.3 Floating Point Arithmetic

A common floating point format is defined in the IEEE 754 standard [IEEE754 2008]. It contains several sub formats. First an overview of standard and specific floating point formats is given, followed by their conversions from decimal numbers to binary data streams and backwards. Also special floating point numbers are discussed.

A.3.1 Floating Point Values and Formats

The most common formats are the single and double precision formats. Both of them, as well as two self-designed ones, are illustrated in Figure A.1.

Data ranges, exponent ranges and number of bytes of the tiny float, small float, single and double precision floating point formats are presented in Table A.1. Each floating point value consists of a sign bit, several bits for the exponent and several bits for the mantissa. In case of normalized numbers the value of a binary stream is given by

$$(-1)^s \cdot 1.\text{fraction} \cdot 2^{e-e_{\text{bias}}}. \tag{A.8}$$

A special type of numbers are denormalized numbers. Their floating value representation is

$$(-1)^s \cdot 0.\text{fraction} \cdot 2^{e_{\text{min}}}. \tag{A.9}$$

For special numbers such as zero, infinity or the mentioned denormalized number formats there are special bit combinations for representing them. Table A.2 gives an overview of these bit combinations.

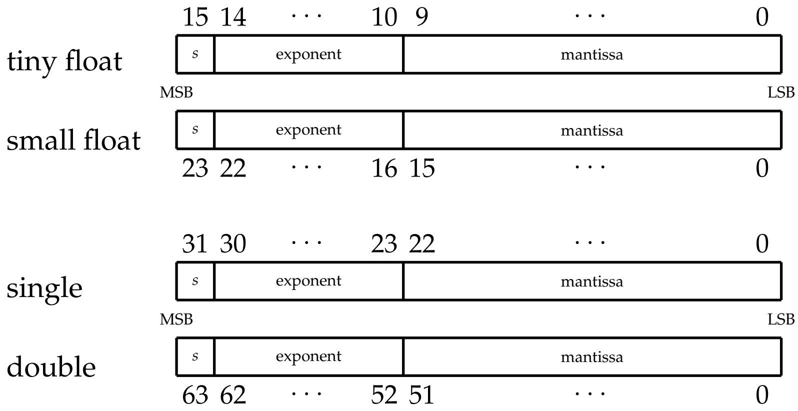


Figure A.1: Binary stream composition of self-defined tiny float and small float formats and common single precision and double precision floating point formats

	tiny float	small float	single precision	double precision
size [bit]	16	24	32	64
sign [bit]	1	1	1	1
exponent [bit]	5	7	8	11
mantissa [bit]	10	16	23	52
exponent bias	15	63	127	1023
exponent range	$-14 \leq e \leq 15$	$-62 \leq e \leq 63$	$-126 \leq e \leq 127$	$-1022 \leq e \leq 1023$
$\|\text{min}\|$	$6.109 \cdot 10^{-5}$	$2.168 \cdot 10^{-19}$	$1.175 \cdot 10^{-38}$	$2.225 \cdot 10^{-308}$
$\|\text{max}\|$	$6.550 \cdot 10^{4}$	$1.844 \cdot 10^{19}$	$3.402 \cdot 10^{38}$	$1.797 \cdot 10^{308}$
$\|\text{min}_{\text{denormalized}}\|$	$5.960 \cdot 10^{-8}$	$3.308 \cdot 10^{-24}$	$1.401 \cdot 10^{-45}$	$4.940 \cdot 10^{-324}$
$\|\text{max}_{\text{denormalized}}\|$	$6.109 \cdot 10^{-5}$	$2.168 \cdot 10^{-19}$	$1.175 \cdot 10^{-38}$	$2.225 \cdot 10^{-308}$

Table A.1: Properties of tiny, small. single precision and double precision floating point values

A.3.2 Conversions between Decimal Numbers and Binary Streams

First step is to evaluate the sign bit s of a decimal number d by

$$s = \begin{cases} 0 & \text{for } d \geq 0 \\ 1 & \text{for } d < 0 \end{cases} . \tag{A.10}$$

Further the exponent e is calculated by the relationship

$$e = \text{floor}\left(\log_2 |d|\right) \tag{A.11}$$

which results in a fraction value $\in [1,2]$. The value of e must lie between its minimal and maximal value for the desired floating point value type. If it is greater than its maximal value the exponent is set to its representation of $+/-$ infinity. In case of a smaller exponent than its minimal exponent the special bit combination for zero value is selected. For denormalized numbers, which are used when (A.11) delivers $e = e_{\text{min}} - 1$, the exponent is set to $e = e_{\text{min}}$. Here the fraction is assumed to contain a leading zero instead of a leading one, cf. (A.9).

values	sign (s)	exponent (e)	fraction (f)
not a number (nan)	0	max	> 0
+/- infinity (+/- inf)	0/1	max	0
+/- zero	0/1	0	0
denormalized	0/1	0	> 0

Table A.2: Special bit combinations for floating point values

The fraction f for normalized values is obtained by

$$f_{\text{norm}} = \left(\frac{|d|}{2^e} - 1\right) \cdot 2^{\text{bits}_{\text{fraction}}} \tag{A.12}$$

and for denormalized numbers by

$$f_{\text{denorm}} = \frac{|d|}{2^{e_{\text{min}}}} \cdot 2^{\text{bits}_{\text{fraction}}}. \tag{A.13}$$

The value of f is rounded to the nearest integer and before converting e to its binary value e_{bias} (given in Table A.1) is added. Now the binary stream containing the sign s, the exponent e and the fraction f can be created.

With a given binary stream the sign, the exponent and the fraction values represent the normalized floating point number

$$d_{\text{norm}} = (-1)^s \cdot \left(1 + \frac{f}{2^{\text{bits}_{\text{fraction}}}}\right) \cdot 2^{e - e_{\text{bias}}}, \tag{A.14}$$

and in the case of denormalized representation the number

$$d_{\text{denorm}} = (-1)^s \cdot \frac{f}{2^{\text{bits}_{\text{fraction}}}} \cdot 2^{e_{\text{min}}}. \tag{A.15}$$

A.4 Signal to Noise Ratio and Mean Square Error

The evaluation of the quality of denoising is done by calculating of the signal to noise ratio given by

$$\text{SNR} = 10\text{dB} \cdot \log_{10} \left\{ \frac{\sum_n f[n]^2}{\sum_n (f[n] - g[n])^2} \right\} \tag{A.16}$$

which emphasis's the removed noise where f and g denote the original and denoised signal respectively, both signals of length N [Moon and Stirling 2000]. Another parameter for noise removing quality evaluation is the mean square error defined as

$$\text{MSE} = 10\text{dB} \cdot \log_{10} \left\{ \frac{1}{N} \sum_n (f[n] - g[n])^2 \right\}. \tag{A.17}$$

Appendix B

Manuals for Auxiliary Programs

This chapter contains manuals for the programs developed in context of this thesis. These tools for data evaluation are:

1. *Binary curve file converter*: Saving long tables of measurement data values in binary form instead of ASCII-form can save a lot of disk space. A typical value contains a sign, three numbers before and after the comma and the comma itself. This leads to 8 bytes per value. The representation with the developed tiny floating point format gets along with 2 bytes for the same value. So about 75% of disk space can be saved.

2. *Measurement data to (binary) curve file converter*: Converts the data obtained by the TDDS setup of [Reisinger et al. 2007a] to binary curve files and prepares it for change point analysis.

3. *Wavelet change point detection*: Implementation of the wavelet-transform-based algorithm from Chapter 2.

4. *Histogram based detection algorithm*: Implementation of the detection process using histograms and techniques from Chapter 3.

5. *Bootstrap and cumulative sum detection algorithm*: Step detection in TDDS data by using the bootstrap statistics and the cumulative sums as discussed in Chapter 4.

B.1 Binary Curve File Converter

For saving disk space, curve files can be converted into a binary file format specified in Section C.1. The python script for this operation is *bcrv.py*. For execution following command line has to be used

python bcrv [options] ⟨input file⟩ ⟨output file⟩.

A list of possible options is given in Table B.1.
The ⟨*input file*⟩ is the path to the original file and the ⟨*output file*⟩ tells the tool where to create and how to name the resulting binary curve file. An example for conversion is

python bcrv -f 32 -t ./foo.crv ./foo.brv.

Options		Type	Description	Default
short	long			
-c	- - columnindex	flag	comma separated list of column indices to be exported into binary file	off
-f	- - format	int	floating point format of binary file, possible values are 16, 24, 32, 64, cf. Appendix A.3	64
-n	- - columnnames	string	comma separated list of column names to be exported into binary file	off
-s	- - scale	flag	scale values in order to fit the data values into the floating point formats data range	off
-t	- - time	flag	show conversion process time after operation is complete	off
	- - version		returns tool version, current version is 1.0	

Table B.1: List of binary curve file converter options

B.2 Measurement Data to (Binary) Curve File Converter

Data of TDDS measurements are delivered in a special file format, the **.tx4* format discussed in Chapter C.2. In order to convert them into the curve- or the binary curve file format the script *tx4tobrv.py* is used. The command is

python tx4tobrv [options] ⟨*input file*⟩ ⟨*output file formater*⟩

The options are presented in Table B.2.

Options		Type	Description	Default
short	long			
-c	- - curve	flag	output file format is a **.crv* file (if not specified a binary curve file **.brv* is created by default)	off
-f	- - format	int	floating point format of binary file, possible values are 16, 24, 32, 64, cf. Appendix A.3	64
-s	- - scale	flag	scale values in order to fit them into the floating values range	off
-t	- - time	flag	show conversion process time after operation is complete	off
	- - version		returns tool version, current version is 1.0	

Table B.2: List of measurement data to (binary) curve file converter options

The ⟨*input file*⟩ specifies the original data file. With the ⟨*output file formater*⟩ every data column of the input file is written into a separated file. The output specifier should be like

⟨*output file formater*⟩ = *./foo%d.brv*.

The output files contain, in this order, the relative time for the single sequence, the absolute time for the complete measurement sequence and the measurement values.
An example for execution is

python tx4tobrv -f 32 -t -s ./foo.tx4 /out/foo%d.brv.

B.3 Wavelet Change Point Detection

The algorithm discussed in Chapter 2 is implemented via a Python script and called by

python chpwavelet ⟨input file⟩/⟨input file formater⟩ [options].

The options are separated into common parameters in Table B.3 and wavelet denoising parameters in Table B.4.

Options		Type	Description	Default
short	long			
-p	- - plots	flag	create output files for documentation plots	off
-a	- - alpha	int	first index of ⟨*input file formater*⟩	0
-o	- - omega	int	last index of ⟨*input file formater*⟩	0
-t	- - transient	float	skip transient of specified duration of input signal in seconds	0
-r	- - result	flag	calculate MSE and SNR and write to file	off
-f	- - filter	float	minimum step size greater than multiple of estimated variance	0.8
-m	- - minstep	float	only steps with absolute value greater than the minimum step value are exported to the switch file	0.0
-c	- - calctime	flag	show detection process time after operation is complete	off
	- - version		returns tool version, current version is 1.0	

Table B.3: List of wavelet detection algorithm common options

The ⟨*input file*⟩ either specifies a single data file or the ⟨*input file formater*⟩ is used. Then the first and last index of filenames are set via the options *-a* and *-o*. The format has to be like

⟨*input file formater*⟩ = *./foo%d.brv*.

Input files can either be of the curve file format IUE [2004] or the binary curve file format discussed in Appendix C.1.
At the end of each detection process an output file containing the detected steps is created. This file is created in the original files folder with name *foo_switch.crv* and is described in Section C.3. If one of - *-plots* or - *-result* flag is set a new directory *./*⟨*input file*⟩*_documentation/* is created. In case of - *-result* a text file named *msesnr.txt* is created containing the calculated MSE and SNR

Options		Type	Description	Default
short	long			
-w	- - wavelet	string	choose wavelet transformation type, ’s’ for SDWT or ’d’ for DWT	s
-y	- - type	string	choose type of wavelet for denoising, possible values are: ’haar’, ’dbX’ for X $\in [2, 20]$	haar
-d	- - threshold	string	denoising threshold type, possible choices are: ’soft’, ’hard’	
-l	- - level	int	level of wavelet decomposition	3
-s	- - shrink	string	type of denoising threshold calculation, possible choices are: ’bayes’, ’mbayes’, ’visu’, ’histogram’	mbayes
-g	- - gamma	float	sensitivity parameter for histogram shrink	0.95

Table B.4: List of wavelet detection algorithm denoising options

after wavelet denoising and after the complete detection process. If the - *-calctime* flag is set a text file named *conversationtime.txt* is created containing the duration of the change point analysis process. A list of files created under use of the - *-plots* flag is given in Table B.5.

Filename	Description	Column contents
origin.crv	original signal data	1: time 2: data values
result.crv	detection data	1: time 2: data values
(S)DWT_wtc_originX.crv	wavelet transform coefficients, X represents the transform scale	1: coefficients 2: index
(S)DWT_wtc_dampedX.crv	damped wavelet transform coefficients, X represents the transform scale	1: coefficients 2: index
waveletdenoised.crv	wavelet-denoised data signal	1: time 2: data values

Table B.5: Files for data-plots created by wavelet detection algorithm

B.4 Histogram based Detection Algorithm

The algorithm discussed in Chapter 3 is implemented via a Python script and called by

python chphistogram ⟨input file⟩/⟨input file formater⟩ [options].

The options are presented splitted into common ones in Table B.3, histogram parameters in Table B.6 and wavelet denoising parameters from Table B.4.

Options		Type	Description	Default
short	long			
-u	- - nonweighted	flag	create an equally weighted histogram	off
-b	- - bins	int	number of histogram bins, for $b = -1$ the number of bins is set automatically	-1
-i	- - regr_type	string	choose interpolation function polynom, must be one of ’none’, ’linear’, ’quadratic’, ’poly’	poly
-k	- - regr_steps	int	number of interpolation points	251
-n	- - noisewnd	int	width of histogram baseline estimation window	5
-e	- - epsilon	float	peak merge distance, for $\epsilon = -1$ the distance is calculated with the robust median estimator	-1

Table B.6: List of histogram parameters

The ⟨*input file*⟩ either specifies a single data file or the ⟨*input file formater*⟩ is used, cf. Chapter B.3. If the - -*plots* flag is set the files handled in Table B.7 and in Table B.5 are created.

Filename	Description	Columns content
convolution_scan_2d.crv	2-dimensional profile of the 3-dimensional convolution scan	1: bin mean values 2: data values
convolution_scan_3d.crv	3-dimensional σ - scan, data is stored block-wise for each value of σ	1: σ 2: bin mean values 3: data values
convolution_second.crv	second convolution graph of highest histogram peak	1: bin mean values 2: data values
gauss_distributions.crv	graph of Gaussian distributions to be plotted with the histogram	1: bin mean values 2: data vales
gauss_distributions_concat.crv	graph of concatenated Gaussian functions	1: bin mean values 2: data values
histogram_origin.crv	histogram from observations	1: bin min values 2: bin max values 3: bin mean values 4: data values
histogram_baseline.crv	estimated histogram baseline	1: bin mean values 2: data values
histogram_denoised.crv	difference between origin histogram and estimated baseline	1: bin min values 2: bin max values 3: bin mean values 4: data values
histogram_fit.crv	observations fitted to Gaussian distributions estimated from the histogram	1: time 2: data values

Table B.7: Additionally created files for data-plots with the histogram detection algorithm

B.5 Bootstrap and Cumulative Sum Detection Algorithm

The algorithm discussed in Chapter 4 is implemented via a Python script and called by

python chpbootcu $\langle$*input file*$\rangle$/$\langle$*input file formater*$\rangle$ *[options]*.

The options are presented splitted into common ones in Table B.3, bootstrap depending ones in Table B.8 and wavelet denoising parameters from Table B.4.

Options		Type	Description	Default
short	long			
-b	- - bootstraps	int	number of bootstraps	1000
-e	- - epsilon	string	detection sensitivity parameter $\epsilon \in [0, 1]$	0.90

Table B.8: List of bootstrap parameters

The detection sensitivity parameter can be used to change the detection sensitivity to ϵ_2 after t_{split} when used like

-e ϵ_1, t_{split}, ϵ_2.

The $\langle$*input file*$\rangle$ either specifies a single data file or the $\langle$*input file formater*$\rangle$, cf. Section B.3.
If the - -*plots* flag is set following the files from Table B.9 are additionally created to the ones from Table B.5.

Filename	Description	Columns content
cusum_min_(no)chp.crv	minimum difference CUSUM chart in (no) change point case	1: index 2: data values
cusum_max_(no)chp.crv	maximum difference CUSUM chart in (no) change point case	1: index 2: data values
cusum_origin_(no)chp.crv	CUSUM chart of original signal in (no) change point case	1: index 2: data values
histogram_cusumdiff_(no)chp.crv	histogram of CUSUM chart differences in case of (no) change point	1: bin min values 2: bin max values 3: bin mean values 4: data values
conf_val_(no)chp.txt	contains the parameters ϵ, γ_ϵ and differences of origin CUSUM chart in case of (no) change point	

Table B.9: Additionally created files for data-plots at the bootstrap and CUSUM detection algorithm

APPENDIX C

FILE FORMATS

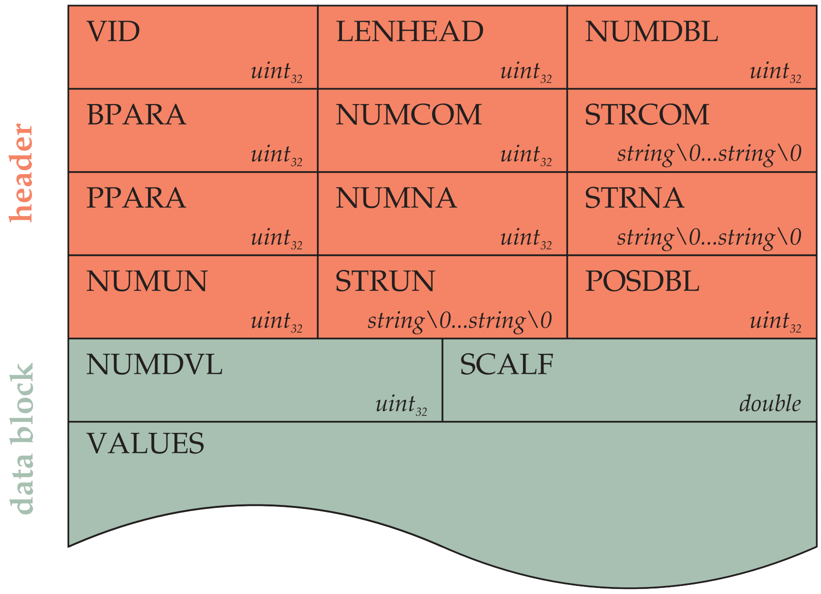
Important file formats related to the Python implementations for this work are presented now.

C.1 Binary Curve File

The curve file format was originally introduced as one of the output formats of the minimosNT simulator [IUE 2004]. In its binary version the binary curve file (*brv*) format characters and values are saved as binary instead of ASCII numbers. The big advantage is that values are saved with a defined number of bytes. So they need less memory space in contrast to the ASCII text format. The saving of memory will not be noticed for a small number of values but can improve performance for long rows points, e.g. with measurement data.

The basic layout of a binary curve file consists of a header with variable length, followed by data blocks. The header consists of the basic file information, while the data blocks contain the measurement values. The values of the type unsigned integer or floating point values are stored in *little-endian* format. For further information on endianess refer to Section A.2.

Figure C.1 shows the principal layout of a binary curve file. In the header section all cells are important and none can be omitted.



header	VID *uint*$_{32}$	LENHEAD *uint*$_{32}$	NUMDBL *uint*$_{32}$
	BPARA *uint*$_{32}$	NUMCOM *uint*$_{32}$	STRCOM *string\0...string\0*
	PPARA *uint*$_{32}$	NUMNA *uint*$_{32}$	STRNA *string\0...string\0*
	NUMUN *uint*$_{32}$	STRUN *string\0...string\0*	POSDBL *uint*$_{32}$
data block	NUMDVL *uint*$_{32}$	SCALF *double*	
	VALUES		

Figure C.1: Binary curve file format, header and data blocks, for VID = 1

The header presented in Figure C.1 consists of:

1. **VID**: The version id (VID) represents the current version of the binary curve file format. This option is introduced to enable further development, to ensure the compatibility of the file format and the conversion tools. An ASCII character which is not interpreted as printable character is recommended, preferably from the range of ASCII 0 to ASCII 31. With the VID it is possible to detect if the file is a binary file or not.

2. **LENHEAD**: The length of the file header in bytes. The first data block starts at the offset LENHEAD.

3. **NUMDBL**: The number of data blocks.

4. **BPARA**: Data values can be represented by different floating point formats. The bytes parameter (BPARA) can either be 16, 24, 32 or 64. Information on composition of floating point formats can be found in Section A.3.

5. **NUMCOM**: The number of comment strings following.

6. **STRCOM**: The comments are stored as null terminated strings.

7. **PPARA**: Number of parameters ('#p'-line in the curve format).

8. **NUMNA**: The number of quantities (equals the number of columns).

9. **STRNA**: The NUMNA strings of names stored as null terminated strings.

10. **NUMUN**: The number of units (NUMUN) must either be equal to the number of names (NUMNA) or zero (in case the ASCII curve file does not contain a '#u'-line).

11. **STRUN**: Null terminated strings of units.

12. **POSDBL**: Finally the header contains the absolute start positions of the data blocks in the binary curve file, counted from the beginning of the binary curve file.

After the header is discussed, the composition of the data blocks is now shown:

1. **NUMDVL**: First the number of data values is stored.

2. **SCALF**: To ensure that each value can optimally be represented by floating numbers with the size defined by BPARA scale factors are introduced for each column separately. They are saved column wise. The following values have all be multiplied with their scaling factors to obtain the original values.

3. **VALUES**: Now all the values are stored column wise. Each column has an equal number of values. Therefore the number of value tuples can be calculated by dividing the number of data values NUMDVL by the number of names NUMNA.

It is possible that there are several data blocks (separated by empty lines in the ASCII curve format). Their absolute start positions in the file are given by the headers parameter POSDBL.

C.2 Measurement Data File Format

Values obtained by experimental setup are saved in *tx4* files. There are two different file specifications, which can be treated by all scripts written in the scope of this thesis. In order to deal with this file type one should remove all ASCII 0 characters first. The single cells are separated by a horizontal tabulator, ASCII character 9. All the lines have the same number of cells.
The header contains the file identification, the file name and optionally a date and time cell. The first cell in this file must always be equal to 4. It is followed by the file name and the date and time of creating the measurements. Because all lines count the same number of cells all the following cells in the first line are empty.
After the header the two versions have to be distinguished. This is done by looking for the cell names *'Rep'* and *'S_Rep'*.
The whole format for version 1 is shown in Table C.1. Version 1 headers do not contain the date and time field. The first column without a name contains a value $t_{\log}$ representing the relative relaxation time. It is converted by

4	SPT9_ZA910186#20_T=170°_hyst_								...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
							Rep1	Rep2	...
							0	1.E-6	...
						-5.921	-707.25	-708.01	...
						-5.444	-732.61	-734.20	...
						-5.119	-737.43	-738.68	...
						-4.807	-737.16	-738.43	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table C.1: Typical **.tx4* file format for version 1

$$t = 10^{t_{\log}} \cdot 1s. \tag{C.1}$$

The first cell under the column name contains the stress time.
In version 2, which is shown in Table C.2, the structure is very similar to version 1. In contrast to version 1 the header now contains date and time information. The stress time is the row above the column names *'S_RepX'*. The line below the names are additional information and not necessary for evaluation. The second line below the column names contains the cumulated time in seconds. There are also some process parameters stored in the first columns, which are neglected for change point evaluation.

C.3 Change Point Detection File Format

The format of a typical switch file is shown in Listing C.1. The header, declared as comments in the curve file style, contains the conversion date and time as well as the used detection algorithm. In line 3 the command line used to create this file is given.
In the data section the first column identifies the file number where the step was detected. The corresponding filename is given in the header section. The second column represents the occurrence time of the change point followed by the step size in the third column.

4	SPT9_ZA910168#20...	24 Sep 2009	15:50:56						...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
							9.E-6	1.E-5	...
							X	S_Rep0	...
							0	1.E-5	...
							0.93201	343.04	...
						-5.398	-1001.87	-1007.94	...
	id=	0				-5.194	-1001.64	-1005.36	...
	RS=	5100				-4.983	-1001.87	-1005.28	...
	RP=	2150				-4.735	-1001.72	-1007.24	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table C.2: Typical **.tx4* file format for version 2

```
## switch file for change point detection created on 2011-08-17 at 20:44
## detection script: wavelet detection
## python main ./data/switchfilefordoku/#21_%d.crv -a 10 -o 15 -t 0.0 -y haar -d soft -l 3 -s histogram -g 0.95
## nr = 0: origin file name: SPT9_ZA910186#20_cp#45H_SPFETDLHC_dr23_T=130._hyststr_; stress time: 1.000100e-05s; Rep9
## nr = 1: origin file name: SPT9_ZA910186#20_cp#45H_SPFETDLHC_dr23_T=130._hyststr_; stress time: 1.000100e-05s; Rep10
## nr = 2: origin file name: SPT9_ZA910186#20_cp#45H_SPFETDLHC_dr23_T=130._hyststr_; stress time: 1.000100e-05s; Rep11
## nr = 3: origin file name: SPT9_ZA910186#20_cp#45H_SPFETDLHC_dr23_T=130._hyststr_; stress time: 1.000100e-05s; Rep12
## nr = 4: origin file name: SPT9_ZA910186#20_cp#45H_SPFETDLHC_dr23_T=130._hyststr_; stress time: 1.000100e-05s; Rep13
#p 1
#b 64
#n    nr   t   d
#u    1    1   1
   0.000000e+00   3.069526e-03   -6.622727e+00
   0.000000e+00   3.178605e-02   -2.700000e-01
   0.000000e+00   4.606901e-02   4.525000e-01
   0.000000e+00   5.606712e-02   -4.631250e-01
   0.000000e+00   3.067635e-01   1.386250e-01
   1.000000e+00   3.849505e-04   -9.707500e+00
   1.000000e+00   7.692567e-04   -4.780205e+00
```

Listing C.1: Switch file format

Bibliography

C. Blatter. *Wavelets: a primer*. A K Peters, Ltd, 1998.

J. P. Campell, P. M. Lenahan, C. J. Cochrane, A. T. Krishnan, and S. Krishnan. Atomic-Scale Defects Involved in the Negative-Bias Temperature Instability. *IEEE Transactions on Device and Materials Reliability*, 7(4):540–557, Dec. 2007.

S. G. Chang, B. Yu, and M. Vetterli. Adaptive Wavelet Thresholding for Image Denoising and Compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, Sept. 2000.

I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 9 edition, 2006.

B. Efron. Bootstrap Methods: Another Look at the Jacknife. *The Annals of Statistics*, 7(1):1–26, 1979.

B. Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, 1982.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall Inc., 1993.

I. Elyasi and S. Zarmehi. Elimination Noise by Adaptive Wavelet Threshold. *World Academy of Science, Engineering and Technology*, 56:462–466, 2009.

H. Feichtinger and T. Strohmer. *Gabor Analysis and Algorithms: Theory and Applications*. Birkenhäuser Boston, 1998.

P. Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer Verlag GmbH, 2000.

P. Goupillaud, A. Grossmann, and J. Morlet. Cycle-octave and related transforms in seismic signal analysis. *Geoexploration*, 23:85–102, Oct. 1984.

T. Grasser, B. Kaczer, W. Goes, T. Aichinger, P. Hehenberger, and M. Nelhiebel. A Two-Stage Model for Negative Bias Temperature Instability. In *IEEE International Reliability Physics Symposium*, pages 33–44, Apr. 2009a.

T. Grasser, B. Kaczer, W. Goes, T. Aichinger, P. Hehenberger, and M. Nelhiebel. Understanding negative bias temperature instability in the context of hole trapping. *Microelectronic Engineering*, 86:1876–1882, 2009b.

T. Grasser, H. Reisinger, P.-J. Wagner, F. Schanovsky, W. Goes, and B. Kaczer. The Time Dependent Defect Spectroscopy (TDDS) for the Characterization of the Bias Temperature Instability. In *IEEE International Reliability Physics Symposium (IRPS)*, pages 16–25, May 2010.

A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 71(1):38–53, 1911.

C. R. Helms and E. H. Poindexter. The silicon-silicon dioxide system: Its microstructure and imperfections. *Reports on Progress in Physics*, 57(8):791–852, 1994.

M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform. In J.-M. Combes, A. Grossmann, and P. Tchamitchian, editors, *Wavelets. Time-Frequency Methods and Phase Space*, pages 286–+, 1989.

IEEE754. IEEE Standard for Floating-Point Arithmetic. *IEEE Std 754-2008*, pages 1–58, 2008.

Intel. Endianess White Paper. Technical report, Intel Corporation, 15 Nov. 2004. URL `http://www.intel.com/design/intarch/papers/endian.pdf`.

IUE. *MINIMOS-NT 2.1 User's Guide*. Institut für Mikroelektronik, Technische Universität Wien, Austria, 2004. URL `http://www.iue.tuwien.ac.at/software/minimos-NT/`.

K. O. Jeppson and C. M. Svensson. Negative bias stress of MOS devices at high electric fields and degradation of MNOS devices. *Journal of Applied Physics*, 48(5), May 1977.

D. V. Lang. Deep-level transient spectroscopy: A new method to characterize traps in semiconductors. *Journal of Applied Physics*, 45(7):3023–3032, July 1974.

A. Lelis and T. Oldman. Time Dependence of Switching Oxide Traps. *IEEE Transactions on Nuclear Science*, 41(6):1835–1843, Dec. 1994.

P. M. Lenahan and J. F. Conley. What can electron paramagnetic resonance tell us about the Si/SiO$_2$ system? *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures*, 16:2134–2153, July 1998.

A. Likar and T. Vidmar. A peak-search method based on spectrum convolution. *Journal of Physics D: Applied Physics*, 36:1903–1909, July 2003.

Y. Miura and Y. Matukura. Investigation of Silicon-Silicon Dioxide Interface Using MOS Structure. *Japanese Journal of Applied Physics*, 5(2):180–180, 1966.

T. K. Moon and W. C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall Inc., 2000.

M. Morháč. An algorithm for determination of peak regions and baseline elimination in spectroscopic data. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 600(2):478–487, Mar. 2009.

M. Morháč and V. Matoušek. Peak Clipping Alorithms for Background Estimation in Spectroscopic Data. *Applied Spectroscopy*, 62(1):91–106, 2008.

E. Page. Continuous Inspection Schemes. *Biometrika*, 41(1/2):100–115, June 1954.

K. Pearson. Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material. *Philosophical Transactions of the Royal Society of London. (A.)*, 186:343–414, 1895.

R. Rangarajan, R. Venkataramanan, and S. Shah. Image Denoising using Wavelets — Wavelets & Time Frequency —, Dec. 2002.

H. Reisinger, O. Blank, W. Heinrigs, W. Gustin, and C. Schlünder. A Comparison of Very Fast to Very Slow Components in Degradation and Recovery Due to NBTI and Bulk Hole Trapping to Existing Physical Models. *IEEE Transactions on Device and Materials Reliability*, 7(1):119–129, Mar. 2007a.

H. Reisinger, U. Brunner, W. Heinrigs, W. Gustin, and C. Schlünder. A Comparison of Fast Methods for Measuring NBTI Degradation. *IEEE Transactions on Device and Materials Reliability*, 7 (4):531–539, Dec. 2007b.

J.-H. Ryu, H. Wan, and S. Kim. Optimal Design of a CUSUM Chart for a Mean Shift of Unknown Size. *Journal of Quality Technology*, 42(3), July 2010.

D. K. Schroeder. Negative bias temperature instability: What do we understand? *Microelectronic Reliability*, 47:841–852, 2007.

D. W. Scott. On Optimal and Data-Based Histograms. *Biometrika*, 66(3):605–610, 1979.

M. Shensa. The Discrete Wavelet Transform: Wedding the À Trous and Mallat Algorithms. *IEEE Transactions on Signal Processing*, 40(10):2464–2482, Oct 1992.

J. O. Smith. *Spectral Audio Signal Processing*. `http://ccrma.stanford.edu/~jos/sasp/`, Oct. 2008.

W. Tayler. Change-Point Analysis: A Powerful New Tool For Detecting Changes, 2000. `http://www.variation.com/cpa/tech/changepoint.html`.

M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice Hall PTR, Apr. 1995.

E. Yashchin. Weighted Cumulative Sum Technique. *American Statistical Association and American Society for Quality*, 31(3):321–338, Aug. 1989.