

# A Novel Wavelet Based Approach for Time Series Data Analysis

Zur Erlangung des akademischen Grades eines  
Doktors der Wirtschaftswissenschaften

**(Dr. rer. pol.)**

von der Fakultät für  
Wirtschaftswissenschaften  
des Karlsruhe Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Dipl.-Math. Thomas Meinl

Tag der mündlichen Prüfung: 10. 05. 2011

Referent: Prof. Dr. Svetlozar Rachev

Korreferent: Prof. Dr. Karl-Heinz Waldmann

2011 Karlsruhe



*To Jack.*



# Acknowledgements

This work would not have been possible without the loving support of all my family, my dog, and my ex-wife. Thank you all, I am most deeply obliged to you for being along my side over all these years.

I also thank my supervisor Prof. Dr. Svetlozar Rachev and Dr. Edward Sun for their guidance and their invaluable input which without this thesis could not have been finished in such a short time and with such excellent results. I also thank Christof Weinhardt who provided me with the opportunity to undertake some serious research at his institute, not to mention the fun I had with all the nice colleagues. Thank you, we had some times together we will remember for sure.

During the time this thesis was written the author undertook a stay abroad in Brazil as well as in Japan, which was only made feasible thanks to the highly appreciated support of the Karlsruhe House of Young Scientists. My thanks also goes to all the people there who welcomed me so heartily.

Karlsruhe, May 2011

Thomas Meinel



# Abstract

Time series analysis is still a very wide field of research from both a theoretical point of view as well as amongst practitioners. Among the very first tasks in the analysis procedure is the estimation of long-term trends, that is, the separation of this generally slowly evolving component from any short-term fluctuations. Usually, the trend curve, which in most cases is expected to be smooth, can be extracted by a variety of different methods. However, in many application scenarios the trend must also account for sudden changes. These sudden changes comprise of not only jumps, but also other phenomena like steep slopes and valleys. This challenge constitutes an on-going problem for traditional trend estimation methods. While established filtering techniques either fail to capture these sudden changes accurately or are sensitive to high-amplitude fluctuations, the application of parametric methods is challenging due to the generally unknown trend and the innumerable shapes that these sudden changes can assume.

This thesis proposes a trend extraction approach based on wavelet methods. The new algorithm, named local linear scaling approximation (LLSA), is developed by analyzing specific wavelet coefficient step response structures and by transferring these structures onto real signals. This procedure enables the analyst to extract a trend whose smoothness is comparable to the output of linear filtering techniques, while at the same time capturing the details of sudden changes with arbitrary shapes, an area in which usually most nonlinear filters excel. Therefore, LLSA can be seen as a novel approach to bridge the gap between linear and nonlinear filters. The algorithm was developed to be applicable on homogeneous time series without any further requirements on these, and to work with only two additional input parameters, which can also be set in a heuristic manner, yielding a directly implementable and usable method.

---

Moreover, the algorithm's properties are shown, namely its computational complexity, its local linearity, and its impulse and step response. The robustness of LLSA is first shown analytically, and then substantiated by several analyses performed on simulated signals as well as on empirical data. LLSA's performance is further evaluated in two separate application scenarios, that are, price volatility estimation and value at risk. The algorithm's superior performance in relation to two benchmark filtering techniques is shown for a considerable number of cases, and several aspects (i. e., possibilities and limitations) of LLSA's general application are discussed.

# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Requirements and Research Questions . . . . .	3
1.2. Contributions of this Thesis . . . . .	5
1.3. Structure . . . . .	6
<b>2. Methods of Trend Extraction</b>	<b>9</b>
2.1. Time Series Analysis and Trend Extraction . . . . .	9
2.2. Linear Filters . . . . .	22
2.2.1. General Formulation . . . . .	22
2.2.2. Transfer Functions: Time vs. Frequency Domain . . . . .	23
2.3. Nonlinear Filters . . . . .	26
2.3.1. General Perception . . . . .	26
2.3.2. Filter Examples . . . . .	30
2.4. Further Related Methods . . . . .	34
2.4.1. Algorithms for Jump Detection and Modeling . . . . .	35
2.4.2. Alternative Methods for Trend Estimation . . . . .	37
2.5. Summary . . . . .	42
<b>3. Wavelets and Their Transforms</b>	<b>45</b>
3.1. Wavelets . . . . .	45
3.2. Wavelet Transforms . . . . .	51

3.3. Wavelet Trend Extraction and Denoising Methods . . . . .	62
3.4. Summary . . . . .	66
<b>4. The Local Linear Scaling Approximation</b>	<b>67</b>
4.1. Methodology and Implementation . . . . .	67
4.1.1. Derivation . . . . .	67
4.1.2. Final Formulation and Remarks . . . . .	75
4.1.3. Implementation and Usage . . . . .	78
4.2. Properties . . . . .	81
4.2.1. Computational Complexity . . . . .	81
4.2.2. Local Linearity . . . . .	81
4.2.3. Impulse and Step Response . . . . .	83
4.3. Summary . . . . .	84
<b>5. Evaluation and Application</b>	<b>87</b>
5.1. Robustness and Performance Studies . . . . .	87
5.1.1. Analytical Consistency . . . . .	88
5.1.2. Simulated Signals . . . . .	90
5.1.3. Empirical Results . . . . .	97
5.2. Applications . . . . .	102
5.2.1. General Application and Examples . . . . .	102
5.2.2. Price Volatility Estimation . . . . .	110
5.2.3. Estimating Value at Risk of High-Frequency Data . . . . .	114
5.3. Summary . . . . .	116
<b>6. Conclusion and Outlook</b>	<b>119</b>
6.1. Summary . . . . .	119
6.1.1. Requirement Satisfaction and Research Questions . . . . .	119
6.1.2. Contributions . . . . .	124
6.2. Future Research Directions . . . . .	125
6.2.1. Algorithmic Extensions . . . . .	125
6.2.2. Further Applications . . . . .	129

<b>A. Empirical Robustness Study</b>	<b>133</b>
A.1. Statistical Ratios Tables . . . . .	133
A.2. Bootstrap Confidence Interval Tables . . . . .	142
<b>B. Empirical One-step-ahead Forecasting</b>	<b>151</b>
B.1. Conditional Mean One-step-ahead Forecasting . . . . .	151
B.2. Empirical Percentiles . . . . .	158
<b>Bibliography</b>	<b>169</b>



# List of Figures

1.1. Structure . . . . .	7
2.1. Hourly Wikipedia server requests and filtered trends . . . . .	30
3.1. Haar squared gain functions . . . . .	60
3.2. $D4$ squared gain functions . . . . .	60
4.1. Wavelet coefficient structure for step functions . . . . .	73
4.2. LLSA(Haar & $D4$ ) step response functions, $K = 1$ . . . . .	85
4.3. LLSA(Haar) impulse response with different $K$ . . . . .	85
4.4. LLSA( $D4$ ) impulse response with different $K$ . . . . .	86
5.1. Robustness test signal with different noise levels . . . . .	92
5.2. MAE/MSE for the Haar wavelet with 5 jumps and $\sigma = 1$ . . . . .	95
5.3. MAE/MSE variances for the Haar wavelet with 5 jumps and $\sigma = 1$ . . . . .	95
5.4. Wikipedia refinement using the Haar wavelet . . . . .	105
5.5. Wikipedia refinement with $D4$ wavelets . . . . .	106
5.6. Wikipedia refinement with $LA8$ wavelets . . . . .	106
5.7. Filtered trend of the SAP stock data . . . . .	107
5.8. LLSA(Haar) filtered trend of the SAP stock data . . . . .	108
5.9. LLSA( $D4$ ) filtered trend of the SAP stock data . . . . .	108
5.10. LLSA(Haar and $D4$ ) filtered trend of the SAP stock data, $K = 3$ . . . . .	109
5.11. SAP stock price percentile deviations, $K = 2$ . . . . .	114
6.1. Flow diagram of wavelet transforms . . . . .	126

6.2. Two-dimensional response structures . . . . .	130
6.3. Further applications . . . . .	131

# List of Tables

2.1. Today's algorithms' requirement fulfillment . . . . .	43
4.1. $n_\alpha, n_\beta$ and $L^{\text{wvlt}}$ for different wavelets . . . . .	72
4.2. LLSA default input parameters . . . . .	79
5.1. MSE mean and variance, 5 jumps . . . . .	94
5.2. MAE mean and variance, 5 jumps . . . . .	94
5.3. MSE mean and variance, 10 jumps, $\sigma = 1$ . . . . .	96
5.4. MAE mean and variance, 10 jumps, $\sigma = 1$ . . . . .	97
5.5. LLSA inferior performance amount in percentile exceedances . . . . .	113
5.6. LLSA inferior performance amount in VaR and ES estimation . . . . .	117
6.1. Today's algorithms' and LLSA's requirement fulfillment . . . . .	122
A.1. Kolmogorov-Smirnov distance statistics, $\kappa = 4$ . . . . .	134
A.2. Anderson-Darling distance statistics, $\kappa = 4$ . . . . .	135
A.3. Kuiper distance statistics, $\kappa = 4$ . . . . .	136
A.4. Cramér-von Mises distance statistics, $\kappa = 4$ . . . . .	137
A.5. Kolmogorov-Smirnov distance statistics, $\kappa = 10$ . . . . .	138
A.6. Anderson-Darling distance statistics, $\kappa = 10$ . . . . .	139
A.7. Kuiper distance statistics, $\kappa = 10$ . . . . .	140
A.8. Cramér-von Mises distance statistics, $\kappa = 10$ . . . . .	141
A.9. Mean difference bootstrap confidence intervals, $\kappa = 4$ ) . . . . .	143
A.10. Mean difference bootstrap confidence intervals, $\kappa = 5$ ) . . . . .	144
A.11. Mean difference bootstrap confidence intervals, $\kappa = 6$ ) . . . . .	145

A.12. Mean difference bootstrap confidence intervals, $\kappa = 7$ ) . . . . .	146
A.13. Mean difference bootstrap confidence intervals, $\kappa = 8$ ) . . . . .	147
A.14. Mean difference bootstrap confidence intervals, $\kappa = 9$ ) . . . . .	148
A.15. Mean difference bootstrap confidence intervals, $\kappa = 10$ ) . . . . .	149
B.1. Conditional mean one-step-ahead price forecasting, $K = 1$ . . . . .	152
B.2. Conditional mean one-step-ahead price forecasting, $K = 2$ . . . . .	153
B.3. Conditional mean one-step-ahead price forecasting, $K = 3$ . . . . .	154
B.4. Conditional mean confidence intervals, $K = 1$ . . . . .	155
B.5. Conditional mean confidence intervals, $K = 2$ . . . . .	156
B.6. Conditional mean confidence intervals, $K = 3$ . . . . .	157
B.7. Empirical percentiles (Price), $K = 1$ . . . . .	159
B.8. Empirical percentiles (Price), $K = 2$ . . . . .	160
B.9. Empirical percentiles (Price), $K = 3$ . . . . .	161
B.10. Mean percentile width confidence intervals . . . . .	162
B.11. Empirical percentiles (P/L), $K = 1$ . . . . .	163
B.12. Empirical percentiles (P/L), $K = 2$ . . . . .	164
B.13. Empirical percentiles (P/L), $K = 3$ . . . . .	165
B.14. VaR percentile exceedances (P/L) . . . . .	166
B.15. ES percentile exceedances (P/L) . . . . .	167

# Chapter 1.

## Introduction

Time series analysis is still a very wide field of research from both a theoretical point of view as well as amongst practitioners. Time series often do not contain one driving force only but consist of several components that furthermore superimpose each other. These components are in the majority of cases categorized into trends, seasonalities, and noise, which are additionally related to long-, medium- and short-term time periods. Trends are usually linked to long-term periods, seasonalities can be either long-, medium-, or short-term, and noise is often thought of as short-term varying fluctuations. However, other associations are also possible, depending on the respective time series. Note that though this thesis refers mostly to time series, nearly all arguments can be carried over directly to any one-dimensional discrete signal. Therefore, both terms are used analogously.

For many analytical purposes and models it is more convenient or even mandatory to examine the above mentioned components separately. However, as there is only one control variable (i. e., time) and the components are (additively) superimposed, this information is not readily available. It is therefore necessary to divide the time series into its several components, that is, to extract the trend and estimate the seasonalities and short-term variations. In this thesis the focus lies on the trend extraction, which is usually the first step undertaken.

The goal of trend extraction (also: trend estimation) is to determine a smooth trend that depicts the long-term evolution of the time series or its respective underlying system,

where *smooth* must be seen in relation to the whole time series and its interfering noise. However, it is commonly recognized that this notion of smooth trends is only sufficient for certain time series, and that in many practical applications trends must also account for sudden changes like jumps. These are usually caused by singular events like the financial crisis that started in 2007, and exhibit significantly larger amplitudes than the regular surrounding noise than they could otherwise be attributed to. Therefore, these rarely occurring sudden changes form an integral part of the otherwise slowly evolving trend. The key challenge is to derive a trend that on the one hand captures these sudden changes with a high resolution, but is relatively smooth otherwise. This is made even more challenging when the trend exhibits not only pure jumps but also steep slopes and valleys, which unlike jumps have no predetermined structure. As will be outlined, the problem lies in the fact that the methods that do provide a sufficiently smooth trend tend to blur out the details of sudden changes, and that methods that capture these changes well are relatively sensitive to high levels of noise. Therefore, this thesis is oriented to bridge this gap.

The above descriptions consider time series in their usual domain, that is, time. Another very useful point of view is to look at time series from the frequency domain. As a time series can be divided into long-, medium-, and short-term components, this division is made even more clear in terms of frequency ranges, that is, the components are categorized into low and high frequencies. For example, the trend falls into the former category, while the noise usually belongs to the latter. However, sudden changes in the trend again complicate a well-defined division. Jumps, for example, differ from the ordinary short-term fluctuations only in the height of their amplitudes while being located in the same high frequency ranges. Therefore, given the presence of sudden change phenomena, trend extraction methods that rely on the separation of a given signal into components located in different frequency ranges will also not perform well. They are only able to provide either a smooth trend, or capture the details of sudden changes, but not both at the same time. Hence, today's most established methods often rely on parametric approaches, and thus, make additional assumptions and/or explicitly require information about the trend or the noise structure, or are essentially a tradeoff between trend smoothness and the accurate display of sudden changes.

In this thesis the focus lies on economic and financial time series, specifically high-frequency stock price data, particularly as this kind of data is known to exhibit the kind of trends discussed above. Due to the advances in information technology, today these time series are measured in higher frequencies, that are, in hourly and minutely intervals. This new availability of huge amounts of frequently generated, measured and stored data brings up new challenges to their processing techniques. Ordinary approaches that model such time series on a daily basis need to be revised, structures of the noise distributions must be investigated again. Therefore, not many assumptions about these high-frequency time series are generally available, nor do all traditional requirements hold. Furthermore, the rapid increase in the sheer amount of data to be processed, and the potential involvement of time-critical applications require that the effort to extract the trend cannot be disproportionately high, that is, the method used must be computationally tractable. In addition to this, as often high (monetary) values may be at stake, it must be ensured that the trend estimation error is bounded, that is, the results provided by the method must be reliable up to a certain degree without any further explicit verification.

## 1.1. Requirements and Research Questions

Based upon the above discussion, new approaches are needed that account for these challenges. Therefore, in this thesis a algorithm shall be developed that suffices all aspects mentioned above. The first and at the same time key aspect is that the algorithm should be capable of providing a smooth trend estimation and a good resolution of details in the areas of sudden changes. Furthermore, in this thesis, no additional assumptions about the time series and/or its noise are made, that is, the new approach should be applicable on all kinds of discrete signals without requiring any further input parameters based on the signal's structure.

Specifically when working with financial high-frequency data, the aspect of different frequency levels becomes more important, that is, there may be information about the different frequencies contained in the data. Therefore, the algorithm should provide a functionality to use this information, that is, which frequency ranges should be blocked or

be contained in the output, though this is not regarded as a mandatory input parameter.

In order to ensure the algorithm's applicability in real scenarios, its computing time should be comparable to established methods, while, as mentioned above, it is essential to have upper bounds on the estimation errors. Thus, the following requirements that have to be met by the algorithm that is to be developed in this thesis, can be stated directly and explicitly as follows.

- R1) Deliver a smooth trend (i. e., if possible without any ripples) that at the same time preserves the details of jumps, slopes, and so on.
- R2) The output should be manageable a priori in terms of frequencies.
- R3) The algorithm should be applicable to a wide range of time series, without any specific requirements about their noise or other components.
- R4) The algorithm should be computationally tractable, that is, its computing time should be in the same complexity class as comparable filters.
- R5) The algorithm should be robust in the sense that it delivers deterministic, reliable results, independent of the time series at hand or specific parameter sets.

The development of an algorithm that conjointly fulfills the above stated requirements is the challenging task of this thesis. The research questions that are of relevance regarding this development are as follows.

- RQ 1) How can a trend estimation algorithm be designed that fulfills Requirements R1 to R5? Is it possible to fulfill all requirements at the same time or only partially to a certain degree? Specifically, how can the seemingly conflicting Requirements R1 and R2 be brought in line?
- RQ 2) What are the characteristics of such an algorithm, that is, concerning Requirements R3 to R5, what are its application requirements and its properties that can be stated explicitly?
- RQ 3) How does the new algorithm presented in this thesis perform with respect to

alternative benchmarks, also taking RQ 2 into account? What benefits in time series analysis can be expected in what kind of applications, and what are the limitations?

## 1.2. Contributions of this Thesis

In this thesis a new algorithm for the trend estimation of one-dimensional discrete signals will be developed. Given specific requirements on the algorithm and the trend that is to be extracted, today's available methods will be analyzed regarding their suitability and the need for a new approach will be shown. It will also be reasoned why wavelet methods are a promising starting point for the development of this new approach.

The developed algorithm will be analyzed regarding its fulfillment of the above named requirements. Furthermore, its unique properties that in conjunction differentiate it from alternative methods will be shown. By answering the above research questions the overall benefit of the algorithm's application will be demonstrated and the contribution to today's state-of-the-art trend estimation methods highlighted. In detail, these are:

- The development of a new trend estimation algorithm that fulfills (almost) all requirements R1 to R5 at the same time.
- An analytical proof of the algorithm's properties, namely its computational complexity, local linear filtering property, and its nonlinear impulse and step response.
- A threefold evaluation of the algorithm's behavior, that is, two of another independent robustness studies (one via simulations and the other using empirical data) as well as an analysis of the algorithm's applicability and benefits with respect to practical questions. Additionally, concrete advantages (i. e., more accurate results) of the algorithm's application in the areas of volatility estimation and value at risk are pointed out and discussed.

### 1.3. Structure

The structure of this thesis is as follows: In Chapter 2 an overview over today's most common methods and algorithms for trend extraction in time series is given. The methods being close to the algorithm (with respect to their requirements as well as their functionality) will be outlined in detail and thus can serve as potential benchmarks. Significant advantages and disadvantages that led to the idea of the algorithm proposed in this thesis are discussed. Chapter 3 introduces wavelet methods and points out their favorable properties that are important for the novel approach presented in this thesis. Chapter 4 is concerned with the algorithm itself. First, the methodology is presented that leads to the algorithm's mathematical formulation. Then, several properties are proven. In Chapter 5 the algorithm's robustness is analyzed in different settings, that is, via simulations as well as using empirical data. Furthermore, concrete benefits of its application are shown. The contribution of this work is summed up in Chapter 6, where additionally the most interesting directions for future research are outlined.

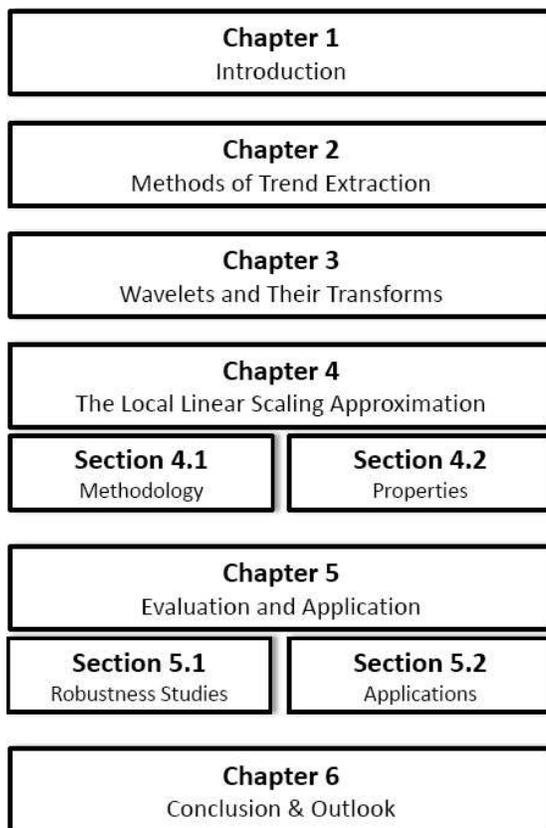


Figure 1.1.: Structure



## Chapter 2.

# Methods of Trend Extraction

This chapter outlines the classic and today's established methods for trend extraction in time series analysis. Section 2.1 first gives a short overview over the most important works in the area of time series analysis itself and highlights the crucial significance of an accurate trend estimation. It also sketches the different approaches and a respective classification of these. Based on this classification the focus of this work on nonparametric methods (i. e., linear and nonlinear filters) is justified, and their potential and limits are examined in Sections 2.2 and 2.3, respectively. In Section 2.4 a review over the literature of research that has been done particularly in the area of jump detection and modeling is given. Moreover, for the sake of completeness, today's used most established methods in practice are depicted, with their preferential domains and limits of application.

### 2.1. Time Series Analysis and Trend Extraction

As time series analysis has become a very wide field of research, there are already many textbooks available that can be considered standard references in this area. Among them, providing a thorough introduction on the general topic of time series analysis are [18,20–23,28,43,50]. The content of these books covers all basic aspects and advanced concepts from time series. Where not indicated otherwise, all the following topics in this section can be found in the references cited above.

### **Motivation and practical problems**

The motivation for time series analysis is mostly the same among all authors, that is the understanding of the underlying system, past events and probably the prediction of future development. However, depending on the particular time series at hand different problems and challenges arise. For example, while for certain systems the inherent noise structure is a priori known, for other time series it is a challenging task just to make the time series stationary, due to the quantity of additional layers (e. g., seasonal deterministic together with long- and short-term stochastic fluctuations in economic time series). The exact nature of these challenges derived from practical applications usually determines the focus of the authors' research contribution.

### **Univariate and multivariate time series and their characteristics**

As multivariate time series are a generalization of univariate time series, the methods applied there usually carry over to the unidimensional case. However, as the latter at the same time imposes more restrictive assumptions, it also leads to stronger results, for example, more efficient (specifically tailored) algorithms or theorems. In this work the focus lies on time series with not more than one control variable (i. e., time). Although there exist multivariate versions or extensions of the basic models, in many cases the transfer of univariate methods to the multidimensional case is not straightforward. As mentioned in the introduction, there is no reason to exclude other univariate signals whose control variable does not necessarily denote time, as technically time series are only a specific instance of these. Yet, considering the above cited literature, the majority of unidimensional signals is measured over time.

### **Fundamentals and general concepts**

The fundamentals discussed in the above cited literature and of most relevance to this thesis are the following.

*Stochastic processes and probability distributions.* Stochastic processes are often derived from a measured time series and used to model these or outline explanatory factors. Stochastic processes themselves comprise deterministic as well as purely stochastic components. Estimating and verifying these is part of the model derivation. Whether a time

series is described by either a stochastic processes or simply probability distributions depends also on the analysis' goal. For example, for derivative price calculation stochastic processes are preferred (see [14]), while other applications, like value at risk (VaR), just prefer the notion of probability distributions (see [38]).

*Time and frequency domains.* Sometimes it becomes advantageous to consider time series not only in their time domain, but rather in their frequency domain. Today's most common tool for this task is certainly the Fourier analysis and transformation (see Section 2.2.2), which represents the respective frequency components contained in the signal. However, as these components are analyzed over the whole signal at once, any time-related information is thus lost. In case this aspect is mandatory, wavelet methods (see Chapter 3) have become the first choice, as they were specifically developed for time-frequency analysis (i. e., they regard both aspects at the same time) of a signal that is decomposed on different scales.

*Spectral and correlation analyses.* Closely related to the two above concepts are spectral density estimation and correlation analysis. While the former estimates not only the signal's inherent probability distributions but also their evolution over time, correlation determines the interrelationship and dependencies either between different time series or the time series to itself (i. e., autocorrelation). If, for example, a time series exhibits significant long lags in its autocorrelation structure, the time series is said to have a long-term memory effect. This is particularly the case in many financial time series, see, for example, [71].

### **Time series transformation**

This area is mostly concerned with extracting the relevant features of a time series and make them presentable for human comprehension. Among these methods are:

*Aggregation and decomposition.* Aggregation of the information contained in time series can take several forms. For example, it can either denote the process of aggregating raw data to make a time series homogenous (see below and Section 5.1.3 for further details), or extract the relevant information by discarding the irrelevant data points and thus, achieve fewer aggregated information points for further processing. A related

complementary task is to decompose the time series into its several components, that is, its long-term trend, seasonalities, and other stochastic processes. Considering each of these components separately enables an easier interpretation of the time series, as its behavior is fully described by these elements.

*Filtration and approximation.* Filtration connotes the removal of undesired elements from the time series. While linear filters can be associated either with time or frequency domains, this does not generally hold for other filters, particularly nonlinear filters, which have no frequency representation. Approximation, on the other hand, denotes the attempt to best represent a certain component of the time series by neglecting the other contained elements. Though sometimes both methods reach the same ends, they always have different points of view: While with traditional filtration generally no explicit underlying structure is assumed, approximation often tends to capture the desired output by representing it via a specific structure (e.g., trigonometric or spline functions).

*Smoothing and denoising.* Both terms are closely related to the above terms of *filtration* and *approximation* and are even used by some authors synonymously. Smoothing denotes the removal of jagged details (e.g., sharks' fins) and thus yields a "smooth" version of the original time series. This can be achieved, for example, either by filtration or approximation. Denoising, however, specifically targets the removal of noise (e.g., short-term variations with relatively low amplitudes or energy) from the time series, which does not necessarily lead to a smooth signal, as the output may still contain, for example, short-term seasonalities (see also Section 3.3).

### **General time series models and methods**

There are different views on time series models and their related methods. The ones relevant for the understanding of this thesis are as follows.

*Stationary and nonstationary.* It is important to distinguish between stationary and nonstationary time series (models). For the former it holds that its statistical properties, like the mean and variance, do not change over time, that is they are independent of the only control variable  $t$  and, thus, remain constant. Many analytical tools and methods

require a stationary time series. Although lately there has been extensive research in the set up of nonstationary time series models (like the ARIMA model or the Kalman filter), today's most common approaches still model stationary time series only. Therefore, the process of making a time series stationary (i. e., transform it into one) is among the first and most important tasks in time series analysis. This usually includes the removal of seasonalities as well as the extraction of any long-term trends, that is, to separate and discard the information belonging to different frequency ranges. This denotes the primary task considered in this thesis.

*Linear and nonlinear.* Since the outstanding work of [18], linear time series models became very famous and were successfully employed by many practitioners. The most prominent approaches are autoregressive (AR), moving average (MA), and the combined ARMA and ARIMA models, where (only) the latter can be applied directly on nonstationary time series. However, in many applications these approaches have proven not to be flexible enough to fully capture the time series characteristics, and initiated the development of nonlinear models. The most notable ones are the autoregressive conditional heteroscedastic (ARCH) model (introduced by [39]) and its generalized GARCH version (see [17, 112]). The notion of linear and nonlinear also holds for other aspects, like filtration and approximation methods. For example, if the long-term trend of a time series is assumed to be linear, it can be approximated (and is reasonable to do so) with a linear function as well. However, these kind of assumptions hold only for very few time series in practice. Other nonlinear approximation approaches, like splines (see [40, 114]), have proven to provide better results than too simplistic models. Linear and nonlinear filters are discussed in detail in Sections 2.2 and 2.3.

*Parametric and nonparametric.* When modeling a time series or a particular part of it (i. e., trends, seasonalities, or other components like jumps), one usually has to choose between a parametric and a nonparametric approach. One can state, that while nonparametric models are much more general (applicable) than their parametric counterparts, they usually do not achieve the same overall quality (i. e., accuracy) if the part to be modeled fits the parametric assumptions. On the other hand, this means that if the parametric model is chosen badly, it can lead to a poor estimation of the model

parameters and, subsequently, the model itself. In extreme cases this may even yield highly misleading results. Nonparametric models usually do not expose themselves to this risk as they omit any such particular assumptions, though of course these models have defining parameters, but do not determine themselves on a particular model structure. For further details about this topic, please refer to [43] and the references therein.

### **Model derivation**

Model derivation in time series analysis can generally be subdivided into the following three steps.

*Identification and selection.* The first step usually is to set up an appropriate model that best represents the time series and its properties. This choice mainly depends on what is already known from the measured sample about the time series and its underlying system, and is to a good part based on reason. For example, seasonal models are assumed if it can be observed that the time series is (directly) correlated with human work or recreation cycles (see, for example, [62, 121]), or there exist electricity usage cycles in summer and winter seasons (see, for example, [72, 104]). Similarly, some methods like the Kalman filter (see Section 2.4.2 below) require explicit knowledge of the interrelations from one state of the system to another (i. e., the state transfer functions must be known). This can yield very efficient and accurate state space models, which, on the other hand, may be sensitive to precise estimates or measures of the initial conditions. Other approaches, like regression methods, may even in the ideal case not deliver such accurate results, but are in most cases more robust considering other input factors and control variables. Therefore, model selection itself (taking also into consideration above specifications) can easily be identified as to be one of the most crucial steps.

*Fitting (i. e., parameter estimation).* Independently of the particular time series model used, the parameters that determine the model need to be estimated. Common procedures in the literature are maximum likelihood estimation, least-squares approximation, and Bayesian methods.

*Verification and validation.* After having estimated the parameters of the selected

model, it is mandatory to verify whether the now completely derived model in fact represents the empirical time series. This can be done, for example, by considering several measures, like goodness-of-fit tests, or by analyzing the deviations of the values predicted by the model to empirical (historical) samples, i. e., backtesting.

### **Forecasting and prediction**

The analysis of signals or time series is usually not done as an end in itself but yields a deeper understanding of the underlying drivers of the system the time series is derived from, which in turn is used to predict or forecast future events. Note that in this thesis the terms *forecast* and *prediction* are used synonymously. As these methods are usually at the end of the chain of time series analysis methods (unless they are based on recursive procedures used for further a posteriori calibration), their accuracy depend on the accuracy of all prior undertaken steps.

### **Steps in Time Series Analysis**

Above depicted framework permits the statement that time series analysis is indeed a very wide field and raises challenges to researches as well as practitioners from the most distinct fields. As already remarked, in this work focus lies on the very particular aspect of trend extraction that is integrated in the steps taken in each complete time series analysis. These are usually as follows:

1. Time Series Measurement

The very first task is to collect and store the data itself in an appropriate format. While data collection and measurement depend on the system that the time series is derived from, the choice of where and how to store the raw data should be made dependent on the goal of the analysis itself, the tools involved, and which of the next steps must be undertaken.

2. Raw Data Preprocessing

In this step the raw data is cleansed from obvious outliers (which might be due to measurement errors) and adequately prepared for being able to undertake the

next steps. This includes, for example, making the data homogenous if the raw data is irregularly spaced or if data points are missing, or transform the time series to be handled into the desired domain. This might cover aspects such as looking at return series instead of price data, or log transforms to stabilize the time series' variance (see, for example, [31] or [73] for a specific treatment of this topic in high-frequency finance).

### 3. Transform to Stationary Time Series

For many analyses this is the most important step, as otherwise many succeeding results would be falsified. Making a time series stationary means to extract all (deterministic) features such that its statistical properties (like mean and variance as well as correlations) become independent over time, that is, they are constant. This includes the following two sub-steps (which sometimes may also be done simultaneously, see, for example, 2.4.2):

#### a) Trend Extraction (Detrending)

This step is concerned with removing the non-periodic trend(s) from the time series. Being the basic topic of this thesis and the main motivation for developing the algorithm, this will be discussed in detail below.

#### b) Seasonality Extraction

In this step all deterministic periodic functions are extracted from the series. This may include cycles of different period lengths. For example, in economic and financial time series one can expect seasonalities with periods of one year (or longer), as well as monthly, weekly and daily cycles, depending on the time series itself, its underlying system, and the measurement frequency. This notion also includes seasonal effects, for example, passenger counts in the airline industry, which vary deterministically according to each season.

### 4. Final Analysis

The final analysis is carried out according to the goals of the whole time series

analysis itself. This might be, for example, an analysis of variance (ANOVA), analyses of special or irregular patterns, the estimation, calibration and application of time series models and stochastic processes, as well as the succeeding forecasting. Note that for some of these procedures not necessarily all previous steps must be undertaken and can sometimes be omitted.

### The Notion of Trends with Sudden Changes

In this work the focus lies on sub-step 3a, that is, the extraction of trends of already preprocessed time series data. While there is no universal definition of *trend* which applies to all fields of application, it is generally accepted that a trend is a slowly evolving component that is the main driving force for long-term development beneath the system. [90] characterizes trend as being limited to certain low frequencies. This notion excludes any noisy influences and fluctuations from higher frequency levels. However, this notion of trends is not satisfactory for many time series encountered in practice. Though theoretical models (like the Black-Scholes model developed by [16], or the Black 76 model in [15]) do not incorporate aspects like seasonalities or even jumps, they are still widely used today in practice, assuming perfect division between the trend and stochastic fluctuations. Yet, this is insufficient for many time series measured today: Especially when considering trends over longer periods (i. e., years or decades for economic and financial data), there appear significant jumps or steep slopes which cannot be attributed to be a part of the persistent stochastic noise, but are caused by external factors like the financial crisis, which began in the year 2007. Another example comes from electricity markets, where it has been found that jumps occur on such a regular basis that it became reasonable to model these by specific stochastic processes, see [104]. Although these patterns contradict the slow evolving characteristic and the low frequency notion generally associated with trends, nevertheless nowadays they are considered to be an inherent part of these.

A basic time series model (amongst others proposed by [43]) is stated as follows. Given a time series  $X_t$ ,  $t \geq 0$ , this series can be decomposed into

$$X_t = \vartheta_t + s_t + Y_t, \tag{2.1}$$

where  $\vartheta_t$  represents a *slowly varying function known as the trend component*,  $s_t$  a single of combination of periodic functions (i. e., seasonal components) and  $Y_t$  the stochastic component, assumed to be stationary.

For the empirical parts of this work the main focus is on high-frequency financial time series data, where jumps occurring in these have the following perception. As pointed out by, for example, [113], jumps in financial time series, and particularly in high-frequency data, are attributed to external events, like the increase or drop in interest rates by some governmental financial institution. These events can be considered to happen only occasionally, and are very sparse in relation to the frequency the data is measured, that is, for the majority of the measured data there do not occur any jumps or similar patterns at all. In the field of high-frequency financial data analysis, jumps are thus assumed to be extreme events that happen with low probability, but form nevertheless part of the stochastic distribution and must be considered to be modeled there. Thus,  $Y$  will be modeled either by stochastic processes including jump components (see, for example, [104]) or by a distribution itself, depending on the model. Such a distribution for high-frequency financial data has then found to be heavy-tailed, that is, jumps happen with enough regularity that they cannot simply be discarded as non-recurring events, see, for example, [94]. However, as these extreme events can have an enormous impact on the stochastic variance analysis and its succeeding usage, and furthermore could lead to misleading results in the regions of the signal without any jumps, for certain analysis purposes it may be preferred to rather include such sudden changes into the trend component  $\vartheta$  than to attribute them to the stochastic component  $Y$ .

In, for example, [115], a definition of an  $\alpha$ -cusp in a continuous function  $f$  at  $x_0$  is given, that is, if there exists an  $\epsilon > 0$  so that

$$|f(x_0 + h) - f(x_0)| \geq C|h|^\alpha$$

holds for all  $h \in [x_0 - \epsilon, x_0 + \epsilon]$  and some constant  $C$ . For  $\alpha = 0$ ,  $f$  is said to have a jump at  $x_0$ . Though to the author's best knowledge there exists no precise definition for the discrete case, it is commonly agreed that the jump should significantly differ from the other fluctuations (i. e., the noise) in the signal. As said above, jumps are just one

particular pattern of extreme events one usually is interested in. Others are steep slopes, roofs and valleys, which in [64] are defined by having a jump in the first-order derivative of the regression curve.

Other extreme events frequently occurring in many practical applications are spikes and outliers. However, these are usually undesirable features that should not be included in the trend or affect it by any means. This is due to the following reasons. First, in many cases these outliers or spikes consist only of one or very few points often caused by measurement errors, and it is obvious that they were caused by some factor that plays no vital role in the ongoing time series analysis (unless the focus is on what caused these outliers). Second, while jumps imply a permanent change in the whole time series, outliers do not contribute to this. While the distinction between a few (adjacent) outliers and roofs/valleys may not be precise, from the context of the time series in most cases it is evident whether an occurrence should be considered as an outlier that is to be neglected or a significant feature to be included in the trend. Summarizing the above pointed out aspects, the notion of trends relevant for this thesis is given in the following definition.

**Definition 2.1.** *A trend is a mostly slow (with respect to the noise) evolving pattern in a time series, with its driving force not being attributed to any noise present in the signal. Trends may also exhibit edged frontiers (i. e., jumps and sudden regime changes) as well as steep slopes, roofs and valleys (see [64]), as long as these patterns can be contributed to the long-term dynamics of the time series and do not stem from any seasonalities or the noise component responsible for short-term variations.*

It is important to note that trend must always be interpreted with respect to the time series at hand (i. e., the period coverage of the data) and the goal of the data analysis, that is, on which scale the trend and the noise are relevant.

**Remark 2.2.** In this work any distinction between long-, intermediate-, and short-term trends and/or seasonalities is omitted, as these usually depend on the context of the time series. For example, financial markets time series can have, for example, secular and daily trends, as well as weekly cycles. Though besides the intermediate and short-

term seasonalities and cycles there may be according (e. g., weekly and daily) trends observable, the for this thesis relevant description is the one given in Definition 2.1.

Note that in this work only additive noise as in Equation (2.1) is considered. Though is it not explicitly excluded that the approach presented in this thesis also works with other kinds of noise (e. g., multiplicative noise, for which edge enhancing methods exist, see [103], the implicitly stated assumption is that the underlying trend, disregarding any jumps or steep slopes, can be estimated or approximated by some sort of basic averaging or ranking within an over the signal moving window.

### **Trend Estimation Model Aspects**

The time series considered in this work are homogeneous. However, particularly in Sections 5.1.3 and 5.2, where the consistency and application benefits of the algorithm on empirical high-frequency financial time series data will be analyzed, this data is initially irregularly spaced. Homogeneity in this context means that for a given series all time steps are equally spaced. This is not always the case for empirical time series, especially in the area of financial high frequency data. In this case it is necessary to preprocess the inhomogeneous (i. e., irregularly spaced) time series by interpolation methods in order to regularize the raw data (see Section 5.2). Though there exist models that can handle inhomogeneous time series directly (see [31], but they also remark that most today's models are suited for regularly spaced time series only), all time series are assumed to be homogeneous ones. This is because the approach of this thesis being based on weighted moving averages, which are not compliant to irregular spaced data.

An additional requirement is that the method used for trend extraction should be robust, that is, the results are reliable and the error can be estimated or is at least bounded in some way. In many cases (see, for example, [35]) the robustness of a method is shown by proving its asymptotic consistency, that is, its convergence towards a certain value for certain parameters tending towards infinity. It should be remarked that the robustness should be independent of the time series itself and/or any specific algorithm parameter sets, in order to be applicable in practice. Of course this does not disregard specific assumptions on the time series that must be met or parameter ranges for which

the algorithm is defined.

In this work the focus lies on nonparametric methods for trend extraction. This is due to the reason that in most time series analyzed in this work one cannot reasonably assume any model for the underlying trend. Yet, as noted in the framework above, in case such assumptions hold it can be expected that those models perform better than nonparametric models, since they are able to exploit information that nonparametric approaches cannot. Furthermore, the commitment to certain parametric time series or trend models can be seen as a restriction when considering the general case, and which may lead even to misleading results in case the trend does not match model, as certain patterns might not be captured or considered by the model itself. This can easily be seen at a most basic example, in case a linear trend is expected, which in most cases will only be a poor estimator for any nonlinear trend curve. In this case nonparametric approaches are less restrictive and can more generally be applied, while of course not delivering the same accuracy as parametric models which exactly match the underlying trend, with only their parameters to be calibrated. If, for example, the trend follows sinusoidal curve, a sinusoidal curve with its parameters being estimated by the least-squares method will almost surely provide a better accuracy than any other nonparametric approach. On the other hand, if the underlying trend is linear or contains even only marginal deviations from a perfect sinusoidal curve, a parametric sinusoidal fit can lead to confusing results and conclusions.

Therefore, as one cannot reasonably assume any model for time series in general, in this work only nonparametric approaches are considered. Within these approaches there are two main branches for trend extraction: Linear and nonlinear filters. This is due to the reason that linear filters are known and have proven to deliver a very smooth trend (given the filtering window size is large enough), while nonlinear filters excel at preserving characteristic patterns in a time series, i. e., especially jumps. Both methods in general require only very few (or none at all) information about the underlying data, besides their configuration of weights and calibration parameters (see Sections 2.2 and 2.3), and are thus applicable to a wide range of time series, independent of the field the data was measured in.

Though there exists a variety of other nonparametric methods, most of these already rely on specific assumptions or choices of parameters which in general cannot easily be derived for any time series data or different analysis goals. Nevertheless, for the sake of completeness, in Section 2.4.2 there are listed some alternative methods, also including parametric approaches, which have been applied in economic, financial or related time series data.

## 2.2. Linear Filters

Linear filters are probably the most common and well known filters used for trend extraction and additive noise removal.<sup>1</sup> First, a most general notion of this filter class is provided in Section 2.2.1, followed by two viewpoints of linear filters and how they can be characterized in Section 2.2.2. While this characterization on the one hand is one of the most distinguishable advantages of linear filters, on the other hand at the same time it leads to the exact problem faced in this work, that is, the representation of sharp edges in otherwise smooth trends.

### 2.2.1. General Formulation

The filtered output depends linearly from the time series input. Using the notation of [43], a linear filter of length  $2h + 1$  can be defined as

$$\hat{X}_t = \sum_{i=-h}^h w_i X_{t+i}, \quad (2.2)$$

with  $\hat{X}_t$  the filtered output and  $w_i$  the filter weights. These kind of filters are also known as (discrete) convolution filters, as the outcome is the convolution of the input signal with a discrete weight function. Thus, for every data point  $X_t$  the filtered output  $\hat{X}_t$  is the result of weighted summation of data points around  $t$ . Applied for the whole time series this results in weighted average window of size  $L = 2h + 1$  which is moved throughout the series. The size of this window is also called the *bandwidth* of the filter.

---

<sup>1</sup>For a detailed discussion about the differences between signal approximation and denoising please refer to Section 3.3

The probably best known linear filter is the mean filter, with  $w_i = 2h + 1$ , that is, all filter weights are uniformly distributed. A more general viewpoint is given by the notion of *kernel filters*. Given a kernel function w.l.o.g. with support  $[-1, 1]$ , this function assigns the weights according to

$$w_i = \frac{K(i/h)}{\sum_{j=-h}^h K(j/h)}.$$

Commonly used examples are the *Epanechnikov kernel*

$$K^E(u) = \frac{3}{4}(1 - u^2)_+,$$

the *Gaussian kernel*

$$K^G(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right),$$

and the symmetric Beta family

$$K_\gamma^B(u) = \frac{1}{B(1/2, \gamma + 1)} (1 - u^2)^\gamma I_{|u| \leq 1}.$$

For the values  $\gamma \in \{0, 1, 2, 3\}$  the kernel  $K_\gamma^B(u)$  corresponds to the uniform, Epanechnikov, biweight and triweight kernel functions, respectively.

As will be outlined in Sections 3.1 and 3.2, wavelets are also linear filters and their respective transform and the resulting signal decomposition can be interpreted as a cascade of linear filters with different bandwidths.

### 2.2.2. Transfer Functions: Time vs. Frequency Domain

The previous section depicts the linear filtering method in the time domain, that is, the time series  $X_t$  and its respective filtered output  $\widehat{X}_t$  evolve over time  $t$ . Another perception can be given by taking the frequency domain into account. For all linear filters there cannot only be given the definition as in Equation (2.2), but also another one with respect to the frequencies the filters let pass. This notion can be derived as follows.

While the sequence of filter weights  $w_i$ , also called *impulse response sequence*, determines the filtered output in the time domain (or equivalent: are the linear filter's time domain representation), via the discrete Fourier transform (DFT) one can derive the *transfer function*

$$\mathcal{W}(f) = \sum_{j=-\infty}^{\infty} w_j e^{-i2\pi f j}, \quad (2.3)$$

which denotes its counterpart in the frequency domain, also called *frequency response function*. Alternatively, if this formulation is given in the first place, one can also derive the weights via the inverse transform

$$w_j = \int_{-1/2}^{1/2} \mathcal{W}(f) e^{i2\pi f j} \, df.$$

Obviously, these two formulations are equivalent, as one can be derived from the other, and vice versa. By considering the transfer function's polar representation

$$\mathcal{W}(f) = |\mathcal{W}(f)| e^{i\theta(f)},$$

with  $|\mathcal{W}(f)|$  the gain function. The magnitude in gain  $|\mathcal{W}(f)|$  (or the in wavelet analysis more common *squared gain function*  $|\mathcal{W}(f)|^2$ , see [88] and Section 3.2) describes the linear filters behavior in the frequency domain, that is, what kind of frequencies and their respective proportions will be let passed or be blocked. Usually it satisfies to distinguish between high- and low-pass filters, that is, filters that let pass either the high frequencies and block the lower ones, or vice versa. In addition to this, other filter types exist, for example, by combining high- and low-pass filters (e.g., in a filter cascade) one can derive band-pass and -stop filters, so that the frequency domain output will be located only in a certain frequency range. In this work the specific interest is in low-pass filters, as they block the high frequency noise and the remaining output consists of the generally low frequency trend.

In case the weights  $w_j$  are real valued, one can show (see, for example, [88]) that  $\mathcal{W}(-f) = \mathcal{W}^*(f)$ , and, with  $|\mathcal{W}^*(f)| = |\mathcal{W}(f)|$ , it follows that  $|\mathcal{W}(-f)| = |\mathcal{W}(f)|$ . Therefore, the transfer functions are symmetric around zero. Due to its periodicity,

it suffices to consider  $\mathcal{W}(f)$  only on an interval of unit length. For convenience, this interval is often taken to be  $[-1/2, 1/2]$ , i. e.,  $|f| \leq 1/2$ . Therefore, with above depicted symmetry, it suffices to consider  $f \in [0, 1/2]$  in order to fully specify the transfer function.

While stating that certain frequencies are blocked and others are passed, this holds only approximately true, since the design of such exact frequency filters is not possible, since there is always a transition between the blocked and passing frequencies. The goal of many linear filters is either to minimize these transitions (i. e., the range of by this affected frequencies), which, on the other hand, inevitably causes ripples in the other frequencies, that is, they are not any longer blocked or let passed completely (see [12] and the references therein for further details about this topic).

As was pointed out above, linear filters can be designed either from a time or a frequency perspective. The time domain usually focuses on putting weights on the surrounding events (i. e., events that recently before or occurred shortly after) and thus, gives an (economic) interpretation similar to the, for example, ARMA and GARCH models. On the contrary, the frequency domain is based on the point of view that certain disturbances are (almost) exclusively located in a certain frequency range, and are, thus, isolated from the rest of the signal. Also, the slowly evolving trend can be seen to occupy only the lower frequency ranges. Thus, the (economic) meaning lies here in the frequency of events, see, for example, [31].

Although linear filters can be designed to block or let pass certain frequencies nearly optimally, at the same time this poses a severe problem when facing trends that exhibit jumps or slopes. As these events are also located in the same (or in case of slopes: the adjacent) frequency range as the high frequency noise, this has the effect that jumps and edged frontiers are blurred out, while steep slopes mostly are captured with poor precision only. Hence, from a frequency perspective, a smooth trend and edge preservation are two conflicting goals. This is as the linear filters are not capable to distinguish between the persistent noise and single events that, while located in the same frequency range, usually have a higher energy and thus, significantly larger amplitude. Thus, the same filtering rule is applied throughout the whole signal, without any adaption. Note, however, that linear filters still give some weight to undesirable events like outliers, due

to their moving average nature. Thus, while some significant features like jumps are not displayed in enough detail, other unwanted patterns, like spikes, still partially carry over to the filtered output. Therefore, it is easy to see that the Requirements R3 to R5 are usually met by most linear filters, but due to above reasons they must fail at fulfilling R1, which is considered to be the primary task in this thesis. To overcome all these drawbacks for that kind of trends or signals, besides the class of linear filters the class of nonlinear filters has been developed.

## 2.3. Nonlinear Filters

As was shown, linear filters tend to blur out edges and other details even though these may form an elementary part of the time series' trend. In order to avoid this, a wide range of nonlinear filters has been developed which on one hand preserve those details, while on the other try to smooth out as much of the noise as possible. Nonlinear filters are not only applied on time series, but they were in many cases developed specifically for the denoising of two-dimensional signals, specifically images, where the original image, probably corrupted by noise during data transmission, consists mainly of edges, which form the image.

### 2.3.1. General Perception

While linear filters generally provide a very smooth trend achieved through averaging, two characteristics pose a problem for this class of filters:

- Outliers and Spikes

Single, extreme outliers and spikes can cause the whole long term trend to deviate in the same direction, though they obviously do not constitute a part of it.

- Jumps, Slopes and Regime Changes

Whenever there occurs a sudden external event in the underlying main driving force, it causes the trend to jump, that is, contrary to spikes it changes permanently onto another plane. While slopes are not that extreme, they also show a similar

behavior as they decay or rise with an for the trend unusual degree.

The reasons for the deviation sensitivity to these events is given by one of the most favorable linear filters' characteristics themselves: It follows directly from them being characterizable in terms of frequency passbands (explained in Section 2.2.2) that all frequencies are treated the same (i. e., filtered according to the same rule) throughout the whole signal. This means that no distinction is made (and cannot even be made) between the noise and sudden change patterns, as they are located in approximately the same frequency range. Technically, as long as an outlier or a jump is contained in the weighted moving average filtering window, also a weight is assigned to these outlier data points or the points before and after the jump. Nonlinear filtering procedures try to avoid this by using a different approach, for example, by considering a single value only (instead of multiple weighted ones) that was selected from an ordered (i. e., ranked) permutation of the original values located in the filtering window.

Though nonlinear filters cannot be characterized in the same way as linear filters (i. e., by transfer functions), according to [86] it is possible divide the whole class of these filters into several subclasses that share the same or similar approaches. Among them, there are stack filters, weighted median filters, polynomial filters and order statistic filters. [9] provide two different taxonomies for further classification, though they remark that these divisions are not unique. In their work, they extensively show how those different filters behave (i. e., their characteristics) when applied onto different benchmark signals with respect to the mean absolute error (MAE) and mean squared error (MSE) error measures.

The behavior of nonlinear filters is generally characterized by their impulse and step response, that is, the filtered output when the input consists of a single impulse or step only. These impulses generally are given by the sequence  $[\dots, 0, 0, 0, a, 0, 0, 0, \dots]$  and  $[\dots, 0, 0, 0, a, a, a, \dots]$ , respectively, with  $a \neq 0$ . Though in most cases no analytical result can be given, these characteristics assist to understand how the nonlinear filter behaves with respect to those patterns, for which the linear filters generally fail to deliver adequate results.

Despite their undoubtedly good ability to preserve outstanding features in a time series while extracting the trend, nonlinear filters also suffer some drawbacks, that is,

1. Insufficient Smoothness

Though most nonlinear filters try to deliver a smooth trend and a good resolution of edged frontiers, by experimenting with several nonlinear filters (taken from [9]) one finds that beyond the jumps' surrounding regions they fail to deliver a smooth trend as accurately as even a simple linear filter (e.g., the mean filter) provides. Yet, by applying further smoothing procedures (e.g., by recursive filters or some kind of linear filtering on the nonlinear output, with a smaller bandwidth, as most of the noise is already smoothed out) comes at the price that the prior preserved details of jumps or slopes tend to be lost again.

This effect is even aggravated when the high frequency noise present in the signal has a high volatility, that is, the trend which evolves besides the jumps quite slowly, is dominated by a noise with extremely large amplitudes. An example is given in Example 2.4. Though certain filters may exist that try to counter this effect, these filters either

- provide only a tradeoff between an overall smooth signal and poor jump resolution, or a trend with preserved edges but still exhibiting ripples, or
- rely on further information about the time series itself, that is, the noise component and its structure, the jumps or the trend itself.

This is due to the problem, that most filtering rules are applied throughout the whole signal, that is, they do not adapt themselves sufficiently fast when the filtering window approaches a jump. An overview of these and other nonlinear filters' performance is given in [9], proving again the well known insight that there cannot exist one solution that performs optimal for all cases. Though the authors also report some approaches that try to incorporate that behavior, these filters provide only dissatisfactory results (see the examples below in Section 2.3.2). Yet, in Section 4.1.1 an analog procedure is the key concept of the algorithm's derivation,

that is, to apply a different filtering rule on whether or not any jump is contained in the filtering window.

## 2. Lack of Frequency Control

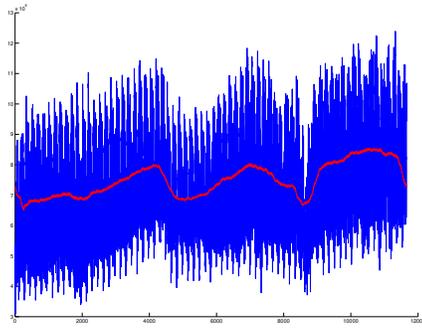
Another feature nonlinear filters lack is the ability to regulate the filtered output in terms of frequency passbands, as linear filters do. Since [90] defines the trend in terms of frequency bands and [98,99] points out that frequency analysis is an important aspect in financial time series, so is frequency control. Though not necessary for all applications, the ability to a priori control and regulate the filters output (in contrast to an only a posteriori frequency analysis of the filtered result) may be useful when one wants to ensure that certain frequencies are not contained in the output. That can be the case when certain information about the noise frequencies is at hand, and therefore, the analyst can decide before the actual filtering process (and thus, without any try and error procedures) what frequency parts should be filtered out. Although a nonlinear filter can also provide the same or a similar result, no theoretical results or statements are available before the filtering procedure has been carried out completely. This incapacity of the nonlinear filter follows directly from the fact that nonlinear filters do not rely on frequency passbands, as they must be able to handle time series components like noise and jumps, even though these are located in nearly the same frequency range.

**Remark 2.3.** As [9] classifies linear filters to be a subclass of the class of nonlinear filters, the above statement does not exactly hold true, that is, it would mean that a subclass of nonlinear filters can be characterized by their transfer function and frequency output. In this thesis however, the classes of linear and nonlinear filters are separated according to whether or not a filter can be described by a transfer function, which marks a strict division between these two classes.

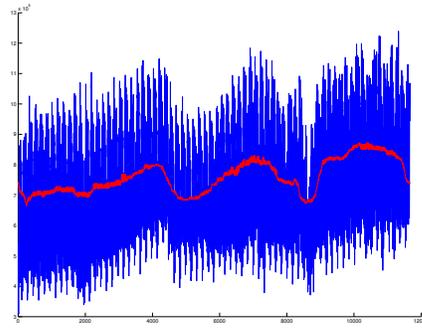
**Example 2.4.** The data in Figure 2.1 represents the hourly measured English Wikipedia server requests measured from January 2007.<sup>2</sup> It can be seen that the trend provided by the linear mean filter in Figure 2.1a is not even very smooth, due to the high amplitudes

---

<sup>2</sup>available at <http://dammit.lt/wikistats/>, last accessed on the 22th of November 2010.



(a) Original Data with Mean Filtering



(b) Original Data with Median Filtering

Figure 2.1.: Hourly Wikipedia server requests and filtered trends

of daily fluctuations. In order to extract a trend of acceptable smoothness the bandwidth is set to  $2^9$ . Choosing a larger bandwidth than this value would probably provide a smoother trend, but also fail to capture accurately the steep slope around summer holiday season as well as the valley around Christmas and New Year's Eve (roughly between data points 4000 to 5000, and 8000 to 9000, respectively). On the other hand, the nonlinear median filter (applied with the same bandwidth as the mean filter) contains even more ripples while seemingly not capturing the extreme areas named above significantly better.

### 2.3.2. Filter Examples

To illustrate the different procedures of nonlinear filters, in this section several examples of above named subclasses are outlined. As this list cannot be exhaustive by any means, of course, filters that already rely on specific assumptions of the systems beneath the time series themselves, are not taken into account.

### Trimmed Mean Filter

This filter works essentially as the mean filter, with the difference, that the extreme values of the *ordered* series  $X_{(i)}$ <sup>3</sup> are trimmed. Therefore, an  $(r, s)$ -fold trimmed mean filter is given by

$$\frac{1}{N - r - s} \sum_{i=r+1}^{N-s} X_{(i)}.$$

A special case is the choice of  $r = s$ . A further modification of the trimmed mean filter is not to discard the ordered values beyond  $X_{(r)}$  and  $X_{(s)}$ , but instead replace them by  $X_{(r+1)}$  and  $X_{(s+1)}$  themselves. This is the Winsorized mean filter

$$\frac{1}{N} \left( r \cdot X_{(r+1)} + \sum_{i=r+1}^{N-s} X_{(i)} + s \cdot X_{(N-s)} \right).$$

In these methods the  $(r, s)$  tuple is dependent on the data itself. Other filters consider to make these values independent from the data or dependent from the central sample itself, i. e., nearest neighbor techniques. All those filters have in common, that they discard all samples from the ordered series being too far away (respectively) according to some measure.

### L-Filters and Weighted Median

$L$ -filters (also called order statistics filters) make a compromise between the weighted moving averages of linear filters and the nonlinear ordering operation. The idea is that the filtered output is generated by weighted averages over the ordered samples, that is,

$$\sum_{i=1}^N w_i X_{(i)},$$

with  $w_i$  the weights analog as in Equation (2.2). A similar notion is given by weighted median filters, where the weights are assigned to the time ordered sample  $X_t$ , and where the weights denote a duplication operation, i. e.,  $w_i \circ X_t = X_{t,1}, \dots, X_{t,w_i}$ . The output

---

<sup>3</sup>Index  $t$  is omitted here as the order is no longer in concordance with time.

is then given by

$$\text{median}\{w_1 \circ X_1, \dots, w_N \circ X_N\}.$$

### Ranked and Weighted Order Statistic Filters

A  $r$ th ranked order statistic filter is simply given by taking  $X_{(r)}$  as the filter output. Examples are the median, the maximum ( $r = N$ ), and the minimum ( $r = 1$ ) operation. This can also be combined with weights as depicted above, that is,

$$r\text{th order statistic}\{w_1 \circ X_1, \dots, w_N \circ X_N\}.$$

### Hybrid Filters

Another approach is the design of nonlinear filters consisting of filter cascades, that is, the repeated application of different filters on the respective outputs. A general formulation is, for example, given by

$$r\text{th order statistic}\{F_1(X_1, \dots, X_n), \dots, F_M(X_1, \dots, X_n)\},$$

where  $F_1, \dots, F_M$  can denote any other filtering procedure. A concrete example is the median hybrid filter that combines a prior linear filtering procedure with a succeeding median ordering operation, that is,

$$\text{median}\left\{\left(\frac{1}{k}\right) \sum_{i=1}^k X_i, X_{k+1}, \left(\frac{1}{k}\right) \sum_{i=k+2}^N X_i\right\}.$$

### Selective Filters

An interesting approach is given by the principle of switching between different output rules depending on some selection rule. For example, based on the fact that the mean filter delivers a larger (smaller) output than the median filter when the filtering window approaches an upward (downward) jump, a selection rule could be given by

$$\text{mean}\{X_1, \dots, X_N\} \geq \text{median}\{X_1, \dots, X_N\}.$$

A certain drawback of this selection rule is that it is onesided, that is, it considers only the first half of the region around the jump. This is due to the fact that the mean for the second half, after the jump has occurred, is generally smaller (larger) than the median. Other rules can include thresholds and aim at deciding whether a jump has actually occurred or if there was an impulse in the signal that was not caused by the noise distribution, but happened due to some other explanatory effect.

### Local Polynomial Smoothing

Local polynomial smoothing is a generalization of the idea to approximate the trend via local constant or local linear functions. In these methods it is assumed that the trend  $\vartheta_t$  is represented by

$$X_{t+i} \approx f_t + \vartheta_{t+i} \quad \text{or} \quad X_{t+i} \approx f_t + f'_t(t-i) + \vartheta_t \quad \text{for} \quad |t-i| \leq h,$$

respectively. Using one of the weighting kernels from Section 2.2.1, the trend is then estimated by  $\widehat{\vartheta}_t = (2h+1)^{-1} \sum_{t=-h}^h X_{t+i}$  or by minimizing

$$\sum_{t=1}^T (X_t - a - b(t-i))^2 K_h(t-i),$$

respectively. A general formulation for polynomial fitting is given in [41]. The general idea is to fit a polynomial function at every data point to the data weighted via the kernel function. However, this method also lacks the same drawbacks as the other methods, that is, since the same rule is applied throughout the whole signal, the filtered output does not adapt itself fast enough to sudden changes.

### Summary

Above depicted examples of nonlinear filters should give the reader an overview over the most common methods applied in practice. For detailed information about each filter's characteristics, their advantages and drawbacks the reader is referred to [9], where there are also the references to the original works to be found. Yet, basically most of these filters rely on some ordered statistics, with their input or output modified prior

or afterwards, respectively. Since this basic principle applies to most filters not directly dependent on some specific characteristic or assuming a certain structure of the original series to be filtered, the different methods pointed out above can be combined in numerous ways. In many cases, however, even though only the basic nonparametric methods (which, however, are also the ones most established in practice) are portrayed here, one notes that almost all of them already incorporate an implicit or explicit choice of additional parameters besides the filter bandwidth, either by weights, rules, or thresholds. These choices introduce further biases into the filtering process. Though some of these parameters can be chosen to be optimal in some sense (i. e., minimize a certain distance measure, for example, the MSE for the  $L$ -filters) they lack the concrete meaning of linear filter weights (see Section 2.2.2) and, specifically, wavelets, which are derived according to additional characteristics (see [32] for details).

Above discussion yields the conclusion that jumps and other sudden changes are usually captured well by nonlinear filters. However, practical applications (see Example 2.4) show that these filters usually do not deliver a smooth trend once any noise with high amplitudes is present. Hence, in these cases Requirement R1 is not met. Also, to the author's best knowledge, Requirement R2 is not fulfilled by any nonlinear filter in today's available literature. As practical applicability is an important issue during the development of nonlinear filters in general, the remaining requirements can be seen as fulfilled.

## 2.4. Further Related Methods

In this section further related methods are listed that are concerned with the estimation of long-term trends exhibiting edged frontiers, i. e., jumps and/or steep slopes. In Section 2.4.1 methods are reviewed that were explicitly developed either only for the detection of jumps in a signal corrupted by noise, or approaches that also include capturing (i. e., modeling) those very jumps. The advantages and limits of applications of these methods are shown, highlighting in which aspects further research is still necessary. This chapter is then concluded by listing some of the most in practice well-known methods in Section 2.4.2. It is explained in what way these approaches differ too much (with

respect to their requirements as well as their outputs) as they could be applied to the general scenarios considered in this thesis.

### 2.4.1. Algorithms for Jump Detection and Modeling

The issue of detecting and modeling jumps in time series has been recognized as an essential task in time series analysis and therefore, has already been considered extensively in the literature. Though wavelet methods are introduced in the next chapter only, the works based on these methods are listed here as well without going into details. It is only noted that wavelets, based on their characteristics, make excellent tools for jump and spike detection, as it is this what they were developed for in the first place (see [81]). Generally, the most appreciated procedures in the recent literature can be seen as two different general approaches. One is via wavelets, while the other uses local (linear) estimators and derivatives.

One of the first approaches using wavelets for jump detection in time series, besides the classical wavelet literature, for example, [75], was given by Wang in [115, 116]. He uses wavelets together with certain data-dependent thresholds in order to determine where in the signal jumps have happened and whether they are significantly different from short-varying fluctuations, and additionally provides several benchmark signals. Assumptions about the noise structure were made according to [35], that is, the approach is applicable for white (i. e., uncorrelated) Gaussian noise only. This work was extended by [96] to include even more general cusp definitions. More recent contributions extend these works to stationary noise (see [122]), other distributions (see [97]), and also provide theoretical results about asymptotic consistency. A further application specifically on high-frequency data is given by [42].

The other line is stated by Qiu in [93], who estimates jumps using local polynomial estimators. This work is continued in [92] where jumps are not only detected but also represented in the extracted time series.<sup>4</sup> Gijbels et al [45] further refine these results by establishing a compromise between a smooth estimation for the continuous parts of a curve, and a good resolution of jumps. Again, this work is limited to pure jumps

---

<sup>4</sup>This is analog to the wavelet procedure named above.

only, and, since it uses local linear estimators as the main method, has no frequency interpretation available. [64] and [107] use derivatives to test the signal for jumps and to represent them.

A completely different approach is presented in [67]: A purely graphical tool for the recognition of probable jumps (and also: areas where almost certainly no jumps occurred). Yet, the authors confess that their work is only to be seen as a complementary approach, and refer to [26] for further thorough investigation.

The above review shows that there exists already a good body of work about how to detect (i. e., estimate their location), and even how to model jumps (i. e., estimate their height), though most works are bounded by strict requirements on the noise or the trend model. However, although most models will also automatically include the detection of steep slopes, they fail at modeling the slope itself. While a jump can easily be presented (either by indicator functions or any other methods used in the cited works above), matters are different with slopes: Since there can be given no general formulation or model of the exact shape of a slope, any parametric approach will fail or deliver only a poor approximation if the model does not fit the occurred slope. Examples of such different kinds of slopes are innumerable: Sine, exponential or logarithmic decay are only the most basic forms to approximate such events, which in practical examples rarely follow such idealized curves. Naturally, only nonparametric approaches will adapt themselves to the true shape of the underlying trend but generally suffer the same drawbacks as all linear and nonlinear methods pointed out above, that is, they always have to balance a tradeoff between bias and signal fidelity.

Although it is difficult to generalize over all models named above, one can grant Requirement R1 to be sufficiently met (though often only pure jumps are considered, and no other kinds of edged frontiers), at the cost of additional information about the time series, which conflicts with Requirement R3. As all approaches were motivated by practical issues and not purely theoretical grounds, Requirements R4 and R5 are also met. However, for none of these algorithms that include the modeling of jumps holds Requirement R2, that is, we have no a priori frequency control.

### 2.4.2. Alternative Methods for Trend Estimation

This section is concluded by outlining today's most popular and established filtering methods. These methods are only stated for the readers convenience and are not mandatory for comprehension of this thesis' contribution.

#### General Least-squares Approaches

The most general approach can be seen by setting up a parameterized model and calibrating this model afterwards, generally using some minimization procedure with respect to some error measure. This can be seen as a straightforward approach, requiring only the initialization of an appropriate model and choice of error measure.

An example from the energy market is given by [72, 104], who set up a trigonometric model in conjunction with indicator functions and minimize the squared error for each time step in order to estimate the deterministic trend as well as values for different seasons and days. Though they find that their model works well in practice, for general cases it might be difficult to set up an appropriate model, especially when there is no information about the trend, its seasonal cycles and other (deterministic or stochastic) influences, since, for example, the data set covers only a short period of time. In this case it is also hard to reason, why a specific model and its estimated trend are appropriate for the respective time series. This requires either a rigorous a priori analysis of the series itself or further information about the external factors (i. e., the system the time series is derived from) and their interaction. In addition to this, the estimation can never be better than the model and to which accuracy it approximates the true trend.

Two other probably critical issues arise when using indicator functions in combination with least-squares estimation. First, usage of indicator functions confines the model to jumps only, that is, slopes or similar phenomena cannot be captured by that approach, as the indicator functions automatically introduce jumps in the trend component. Thus, indicator functions excel at modeling jumps, but perform poorly with other types of sudden changes. Second, for this approach it is extremely important to determine the location of the jump as exact as possible, as otherwise the estimated trend in this area may be highly inaccurate.

It is therefore dubious, whether such parametric approaches are appropriate to handle economic and financial time series, though they will perform very well, if their requirements are met. Regarding the requirements for this thesis trend extraction task, it is easy to see that generally Requirement R2 is not met, while Requirement R1 and R5 depend heavily on the correct model estimation. Requirement R3 usually holds, while Requirement R4 depends on the actual implementation.

### Smoothing Splines

Smoothing splines, though also utilizing the least squares methods, on the contrary do not rely on a specific model assumption. Instead, they penalize the regression spline with respect to its roughness, i. e.,

$$\min_m \sum_{t=1}^N (X_t - m(t))^2 + \omega \int (m''(x))^2 dx.$$

It follows directly from this definition, that for  $\omega = 0$  this yields an interpolation, while for  $\omega \rightarrow \infty$  the resulting solution  $m$  will approximate a linear regression.

In practice, there emerges another difficulty: In many cases, the (optimal) choice of the smoothing (or penalizing) parameter  $\omega$  remains unclear. Although there exist several works that have established some data dependent rules for this, in many cases, when the assumptions about the noise do not hold or the time series incorporates additional deterministic (e. g., cycles) or stochastic components (e. g., outliers that are part of the system and not due to measurement or other errors), the choice of  $\omega$  is a challenging task that has been and is still undergoing extensive research, see [25, 58, 60, 69, 82]. A particular method that also plays a role in similar wavelet procedures (see Section 3.3) is the cross-validation method that is used to determine the optimal smoothing parameters, see [46]. Furthermore, though  $\omega$  is eventually responsible for the degree of smoothness (i. e., on which scale or level the trend shall be estimated), one can hardly neither impose nor derive any additional meaning on or from this parameter. For smoothing splines it can be said that likewise the linear and nonlinear filters Requirement R1 only holds insufficiently. Requirement R2 is certainly not met, while the remaining requirements

usually seem to be fulfilled.

### The Hodrick-Prescott Filter

The Hodrick-Prescott (HP) filter was first introduced by [70] and later became popular due to the advanced works of [55]. In order to extract the trend  $\tau = [\tau_0, \dots, \tau_{N+1}]$  from a given time series  $X$ , this trend is derived by solving

$$\min_{\tau} \sum_{t=1}^N (X_t - \tau_t)^2 + \omega ((\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1}))^2.$$

Besides from not using a spline basis for approximation this approach can be seen as a discretized formulation of the smoothing spline. The smoothing parameter  $\omega$  plays the same role, while the penalized smoothness measure is the discretized version of the second derivative. Thus, though several authors propose explicit rules (of thumb) for choosing  $\omega$  (see, for example, [34, 100, 102]), some researchers like [52], also recognize that in some way this choice still remains kind of arbitrary or problematic for many time series that cannot be associated with the same terms. Regarding the fulfillment of the this thesis' requirements the same statements as for smoothing splines hold.

### The Kalman Filter

Another sophisticated filter was developed by [66]. It is a state-space system specifically designed to handle unobserved components models, and can be used to either filter past events from noise as well as for forecasting. A good introduction to the Kalman filter can be found in [117], and a thorough discussion in [53].

The basic Kalman filter assumes that the state  $x \in \mathbb{R}^n$  of the underlying process in a time series can be described by a linear stochastic difference equation

$$x_t = Ax_{t-1} + Bu_{t-1} + z_{t-1},$$

with  $A$  the state transition model, that relates in conjunction with the (optional) control input model  $B$ , the respective (external) control input  $u_t$ , and the noise component  $z_t$  the previous state to the next. In above equation  $A$  and  $B$  are assumed to be constant,

but generally may also change over time. Of this model (i. e., the true state  $x_t$ ) only

$$y_k = Hx_k + v_k$$

can be observed. Both noise components are assumed to be independent of one another and to be distributed according to

$$z \sim N(0, Q) \quad \text{and} \quad v \sim N(0, R).$$

With  $A$ ,  $B$ ,  $Q$ , and  $R$  assumed to be known, the filter predicts the next state  $x_t$  based on  $x_{t-1}$  and also provides an estimate of the accuracy of the actual prediction.

Since its first development, the Kalman filter has become popular in many areas, see, for example, [47, 78]. However, a serious drawback of this procedure is that many real world systems do not fit the assumptions of the model, for example, the above requirement of a linear underlying system is often not met. Although there exist extensions for nonlinear systems (see, for example, [65]), there still exists the problem that one or more of the required parameters are unknown. While for many technical systems (e. g., car or missile tracking systems) based on physical laws the state transition model  $A$  is exactly known, this becomes a difficult issue in many other application areas, including finance. Additionally, as [79] notes, the performance of the Kalman filter can be very sensitive to initial conditions of the unobserved components and their variances, while at the same time it requires an elaborate procedure of model selection. He also notes that in macroeconomic time series the Kalman filter does not work with annual data. Therefore, while the Kalman filter unquestionably delivers excellent results in many areas (and has also been applied for financial time series as well, though not without critique), its usage is not convenient for the general cases of high-frequency data considered in this work, as it requires more assumptions and knowledge about the underlying model than available. Thus, regarding the stated requirements, it can be said that dependent on the state transition model the Kalman filter actually might meet Requirement R1 quite well, and also R4 holds. However, using the Kalman filter, the output cannot directly be controlled in terms of frequencies, hence Requirement R2 is not met. Also, as this method

requires the state transition matrix  $A$  to be stated explicitly, this limits its application as this information might not always be available. Thus, Requirement R3 is generally not met. Furthermore, since the filter is sensitive to the correct and estimation of  $A$  and the initial conditions  $x_0$ , Requirement R5 does not unconditionally hold.

### **Other Filters and Literature**

Above examples state only the most common and established filtering procedures used in academia and practice. Of course, such a list cannot be exhaustive by any means. Yet, for the sake of completeness, some further literature is sketched that also treats this topic.

[13] conduct a comparison of other filters for trend estimation, namely the GLAS, Henderson, Lowess, Kalman filters, and smoothing splines. In their work, they focus on the detection and modeling of turning points in the trend at the boundaries (i. e., events that only recently happened). Their results, based on an empirical analysis of the M4 flows series of the Bank of England, show that in this case weighted moving average filters perform worst, which can be explained by the fact that this kind of filters suffers from boundary distortions near the beginning and the end of the signal (see also Sections 3.2 and 5.2.1).

In [79] the HP filter is extended in order to handle stochastic trends and cycles and is evaluated at the GDP series of several European countries. Long-term trends with shorter (medium business) cycles are also considered in [54], where the authors compare the performances of different methods, also including the HP filter, which has become widely spread for application in economic time series.

[89] develops square-wave filters for detrending economic time series. Yet, while these filters are found to perform superior to the in the paper considered alternatives (e. g., the HP filter), they also belong to the class of linear filters, which makes them suffer the same drawbacks when jumps in the trend occur.

## 2.5. Summary

In this chapter different methods and approaches were reviewed that might be suitable for the task of extracting the trend of a time series where there is none or very little information about the series, the noise, and other deterministic or stochastic structures, besides that the trend is expected to be slowly evolving, with occasional jumps or slopes. From a frequency point of view it is known that the extraction of this kind of trends contains two conflicting goals, mainly because the noise and the significant jump features are located in the same frequency range.

This fact already disqualifies the whole class of linear filters: Though they provide a very smooth trend in general, their incapacity to adapt themselves to these sudden changes results in a poor resolution of these. Nonlinear filters, on the other hand, capture such occurrences pretty good, but in most cases do not deliver a smooth output as their linear counterpart, that is, the result still contains many remains from the noise, which are reflected in ripples throughout the whole trend, and are undesirable, as they also denote an inaccurate approximation in general. Furthermore, by using nonlinear filters, one cannot control the output a priori in terms of frequency passbands, which is desirable in many applications, where the trend is located in a certain frequency range only, with exception of the jumps.

Other methods here presented rely either on parametric approaches, where the model selection itself can be challenging and where the optimal result is bounded by the model itself, or additionally require the choice of specific input parameters, which might not be known. Any inaccuracy in these very first steps will result in worse approximations of the underlying trend, while for every time series the parameters have to be adapted or selected again, which makes these approaches hard to apply for different scenarios, especially when there are not many information about these time series, for example, in case they stem from newly established markets.

Although specifically for the task of jump estimation in trends there exists a bulk of work in the literature, all of these require information about the structure of the noise within, and fail to model all diverse kind of slopes, even if we admit that Requirement R1

Table 2.1.: Today's algorithms' requirement fulfillment

	R1	R2	R3	R4	R5
Linear filters	○	●	●	●	●
Nonlinear filters	◐	○	●	●	●
Dedicated jump models	●	○	◐	●	●
General least-squares models	◐	○	◐	◐	◐
Smoothing splines & HP filter	◐	○	●	●	●
Kalman filter	●	○	◐	●	◐

is met. The conclusions that can be drawn by the analysis of above presented methods is that Requirement R2 is only met with linear filters, and Requirement R1 mostly holds only if there are additional requirements on the time series itself or if a parametric model is available. The latter case can lead to instable solutions (and limits the fulfillment of Requirement R3) if the model selection has even slight deviations, and thus, denotes a violation of Requirement R5. Therefore, today's models, while they may excel in specific tasks, are not suitable to met all the stated requirements at the same time, where R1 and R2 seem to be the most critical to be fulfilled conjointly, as they represent opposing goals. This is summarized in Table 2.1.



## Chapter 3.

# Wavelets and Their Transforms

Wavelets and their respective transforms have become important and approved tools in signal analysis since the pioneering works of Daubechies and Mallat [32, 74]. In this section, the specific characteristics that make wavelets such useful tools utilized in the most distinct application areas are outlined. The notion of wavelets themselves is introduced in Section 3.1, followed by their respective transforms in Section 3.2. Practical considerations of their application to the purpose of trend extraction are discussed in Section 3.3. Since wavelet theory has become a large field, the material provided in these sections can only give a most basic overview and states the most necessary facts for general understanding of the wavelet concept, while enabling the reader to classify the novel approach presented in this thesis. Unless specifically mentioned otherwise, further reading with a more thorough treatment of almost all topics discussed in this chapter can be found in [32, 88].

### 3.1. Wavelets

Wavelets were originally developed to analyze spikes and abrupt changes in sonic wave signals from geophysics applications, that is, to discover and estimate oil fields (see [81]). Fourier analysis, the most traditional signal frequency analysis at that time, was not suitable for this task as these spikes were not isolated, but their signal and frequency contribution distributed over time, due to the non-localization of the trigono-

metric Fourier bases.

Essentially, wavelets are bases of  $L^2(\mathbb{R}^d)$  and are usually denoted by  $\psi$ . As this work is restricted to unidimensional time series,  $d = 1$  always holds. The name wavelet stems from the fact that additional conditions are imposed on these bases to give them certain desirable properties which are pointed out below. Among the most important requirements are the so-called admissibility condition ( $\Psi$  denotes the Fourier transform of the wavelet  $\psi$ )

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

the normalization

$$\int_{-\infty}^{\infty} \psi^2(u) du = 1, \tag{3.1}$$

and

$$\int_{-\infty}^{\infty} \psi(u) du = 0. \tag{3.2}$$

The latter condition signifies that  $\psi$  oscillates around zero, hence the name wavelet.

What makes wavelet bases so remarkable is that contrary to most traditional frequency analysis tools, like Fourier analysis, they provide a localized time-frequency analysis on different scales. Similar to Equation (2.3) the discrete Fourier transform of a given signal  $X_t$  of length  $N$  is given by

$$\mathcal{X}(f) = \sum_{t=1}^N X_t e^{-i2\pi ft}.$$

As the bases of the Fourier transform are the trigonometric sine and cosine functions  $e^{it} = \cos(t) + i \sin(t)$  which have no compact support, it is easy to see that changing  $X_t$  at even one data point only, will result in a completely different Fourier transform. In order to avoid this problem one would need either bases that outside a certain interval decay sufficiently (e. g., exponentially) fast towards zero (this would not solve the problem but at least reduce its effect) or bases with compact support. There exist wavelets types for either.

The very first wavelet was introduced by Haar in [49], though wavelet theory and the

term *wavelet* itself had not yet been established at that time. Today it is called the Haar wavelet and is given in its continuous form by

$$\psi^{\text{Haar}}(t) = \begin{cases} 1 & \text{for } 0 \leq t < 1/2, \\ -1 & \text{for } 1/2 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3.3)$$

Naturally this wavelet already has compact support but is not very smooth, as it is not even differentiable. While any signal can still be represented with such a basis, as the shape of the extracted trend depends on the shape of the wavelet itself, this wavelet will lead to a rather blocky structure of the estimated trend. Yet, among practitioners the Haar wavelet has become one of the most well known and applied wavelets in time series analysis, mainly due to its simplicity and its very good localization, that is, its very small support.

The for the Haar wavelet exemplarily depicted formulation in Equation (3.3) is also referred to as the *mother wavelet*. In order to analyze a signal on different scales  $\lambda > 0$  and over time  $t$  it is necessary to adapt the mother wavelet  $\psi$  as follows.  $\psi_{\lambda,t}$  denotes the by  $\lambda$  scaled (also: dilated) and by  $t$  shifted version, that is,

$$\psi_{\lambda,t}(\cdot) = \frac{1}{\sqrt{\lambda}} \psi\left(\frac{\cdot - t}{\lambda}\right). \quad (3.4)$$

That is, the wavelet basis is derived through dilation and translation of the mother wavelet.

In wavelet analysis, the mother wavelet  $\psi$  is always accompanied by the scaling function  $\varphi$ . To be more exact, the wavelet itself is derived from this function (since in practice  $\varphi$  is easier to determine). For example, the Haar scaling function is given by

$$\varphi^{\text{Haar}}(t) = \begin{cases} 1 & \text{for } 0 \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Together,  $\psi$  and  $\varphi$  form a *multiresolution analysis* (MRA), that is, a sequence of nested subspaces

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots, \quad (3.5)$$

for which

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R}) \quad \text{and} \quad \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$$

holds. The spaces  $V_j$  are called *approximation spaces*, and are nested via dyadic scaling, that is,  $f \in V_j$  holds if and only if  $f(2 \cdot) \in V_{j+1}$ .<sup>1</sup> Thus, with  $\lambda_j = 2^j$  and  $\varphi$  forming a basis of  $V_0$ ,  $\varphi_{\lambda_j, t}$  is a basis of  $V_j$ , i. e.,

$$V_j = \overline{\text{span}\{\varphi_{\lambda_j, k}(\cdot) : k \in \mathbb{Z}\}}. \quad (3.6)$$

**Remark 3.1.** Through Equations (3.5) and (3.6) one can derive the following two-scale relationship

$$\varphi(t) = \sqrt{2} \sum_{l=-\infty}^{\infty} g_l \varphi(2t - l). \quad (3.7)$$

Though especially for higher dimensions (e. g., image decomposition and analysis) there exists a body of work in the literature considering usage of non-dyadic grids (see, for example, [91]), this thesis is confined to dyadic lattices. This may be seem restrictive, but is commonly accepted in practice and considered to be sufficient to handle most time series adequately. In this case Equation (3.7) generalizes to

$$\varphi(t) = \sqrt{|\det M|} \sum_{l \in \mathbb{Z}^d} g_l \varphi(Mt - l).$$

Both  $\psi$  and  $\varphi$  are now linked in the following way.  $W_j \subset V_{j+1}$  denotes the orthogonal complement of  $V_j$  in  $V_{j+1}$ , also called the *detail space*, and  $\psi_{\lambda_j, t}$  a basis of  $W_j$ . It can be shown that

$$\bigoplus_{j \in \mathbb{Z}} W_j = L^2(\mathbb{R})$$

---

<sup>1</sup>Dependent on dimension  $d$  other meshes are also commonly used, yet for  $d = 1$  the dyadic scaling is the one most applied in practice.

and thus, wavelets indeed form an orthogonal basis, which is due to their construction using the scaling function. Thus, in an MRA the scaling function  $\varphi$  provides the representation of the original signal in approximation spaces with different resolutions which are in a dyadic relation to one another. The wavelets  $\psi$  provide their respective complement, that is, the complementary differences from one approximation space to the next. Thus, the very basic idea behind wavelet analysis is, while scaling functions are associated with coarse resolutions (one could also say: an averaging or aggregating viewpoint) of the original signal, wavelets are concerned with the differences between those resolutions. This very short overview depicts only the intuition behind the motivation of wavelet analysis, and is exhaustively discussed, for example, in [74].

The first wavelet ever constructed that had sufficient smoothness (it was even infinitely differentiable) was the Meyer wavelet, see [76,77], yet it had no compact support. Among all the other wavelets that followed, the ones most noticed and generally applied are the Daubechies wavelets, which are associated with specific smoothness spaces in respect to their order. The wavelet of the lowest order, usually denoted by  $D2$ , coincides with the Haar wavelet depicted above.

The above continuous formulation of wavelets and scaling functions is convenient for many theoretical aspects and analyses using the continuous wavelet transform (see Section 3.2). Yet, for this thesis' research purpose (that is, the motivation of extracting a trend of a time series described in discrete time spots) it is not conducive. For the discrete transform and application, through the two scale relationship one can derive for every wavelet and scaling functions the discrete filter coefficients, which equal the weights of a moving average filter as in Equation (2.2).

For example, the Haar wavelet and scaling filter coefficients, usually denoted by  $h_i$  and  $g_i$ , respectively, are given by

$$h_0 = -h_1 = 1/\sqrt{2} \quad \text{and} \quad g_0 = g_1 = 1/\sqrt{2}.$$

The scaling filter coefficients can be derived by analyzing the two-scale relationship for

the scaling function, which is based on the nested subspace structure in Equation (3.4):

$$\varphi(t) = \sqrt{2} = \sum_{k \in \mathbb{Z}} \overline{g_k} \varphi(2t - k). \quad (3.8)$$

For the wavelet function one has the similar relationship

$$\psi(t) = \sqrt{2} = \sum_{k \in \mathbb{Z}} \overline{h_k} \varphi(2t - k).$$

The wavelet filter coefficients are then given through

$$\overline{h_k} = (-1)^k \overline{g_{1-k-L}}, \quad (3.9)$$

with  $L$  the filter length, that is, the amount of non-zero scaling (and thus, wavelet) coefficients. Thus, in general for the discrete wavelet transforms one has the filter banks  $h_0, \dots, h_{L-1}$  and  $g_0, \dots, g_{L-1}$ . For discrete wavelet filters the conditions

$$\sum_{l=0}^{L-1} h_l = 0 \quad (3.10)$$

and

$$\sum_{l=0}^{L-1} h_l^2 = 1 \quad \text{and} \quad \sum_{l=0}^{L-1} h_l h_{l+2n} = 0 \quad (3.11)$$

must hold for all  $n \in \mathbb{N}^*$ . These conditions are the discrete counterpart to the oscillation and orthonormality conditions on the continuous wavelet functions, Equations (3.2) and (3.1). Due to the relation between the scaling and wavelet filter coefficients in Equation (3.9) analog conditions (must) hold also for the scaling filter bank, that is

$$\left| \sum_{l=0}^{L-1} g_l \right| = \sqrt{2}$$

and

$$\sum_{l=0}^{L-1} g_l^2 = 1 \quad \text{and} \quad \sum_{l=0}^{L-1} g_l g_{l+2n} = 0.$$

Further specific details about wavelets' characteristics and desiderata which play a role during the wavelet construction process can be found in [32]. In this work the focus lies on the wavelets most applied in practice, namely the Haar, the *Daubechies D4* and the *least asymmetric LA8* wavelet. These wavelets are subject to different construction criteria and at the same time are the ones of their type with the smallest support. Note that the conditions in Equations (3.10) and (3.11) are necessary, but not sufficient to construct reasonable high- and low-pass filters, which require further regularity conditions. In this work none of these details about the construction of wavelets are discussed. Instead, the focus is now on their utilization in the respective wavelet transforms.

## 3.2. Wavelet Transforms

As with wavelets, their respective transforms can be basically separated into the *continuous wavelet transform* (CWT) and several further subdivided *discrete wavelet transforms* (DWTs). Both kinds of transform have in common that in regard to the scaling function filter, the scaling coefficients are associated with averages on a certain scale (a general notation used in the literature is  $\lambda$  for the CWT, and  $j$  for the DWTs), while wavelet coefficients are associated with the differences between those averages on these scales. This makes wavelets suitable for detecting specific features in a signal, like spikes or jumps, that is, in case the wavelet coefficients are small, from the viewpoint of this scale the signal is relatively smooth, while high coefficients indicate a rather jagged patterns. Thus, wavelets are a tool to estimate, determine and characterize signals in respect to their degree of smoothness on a certain scale, and also, due to their localization, serve to detect rapid changes. This already hints at why wavelets can be considered to be suitable for the task at hand.

### The Continuous Wavelet Transform

Given a continuous signal  $X(t)$  and the continuous formulation of any mother wavelet  $\psi$ , the wavelet coefficients given through

$$W(\lambda, t) = \int_{-\infty}^{\infty} X(u)\psi_{\lambda,t}(u) du \quad (3.12)$$

span over the entire real time axis  $t$ . In this way one can see how wavelets implement the idea of differences between weighted moving averages, with the bandwidth given by  $\text{supp}(\psi_{\lambda,t})$ . For the practical application of the CWT the coefficients are evaluated on certain points of a predefined mesh consisting of the intersection points along the scale and the time axes. However, while the time axis and the mesh on it is naturally limited by the length of the measured signal and the amount of registered data points, there is no obvious restriction or choice for the grid in respect to the scale  $\lambda$ . This results in conveying the analysis of a one dimensional signal to the analysis of a two dimensional image, which besides that, is analytically more challenging to handle. Though the picture depicted by the wavelet coefficients (i. e., each point's brightness is correlated with the wavelet coefficient's value) easily reveals to the human eye certain structures or outstanding events and their evolution across different scales, further processing is not straightforward. This makes the direct application of the CWT not suitable for this thesis' purpose.

### The Discrete Wavelet Transform

Contrary to the CWT, the discrete wavelet transform, as its name suggests, relies on an initial discrete decomposition of a signal  $X_t$  defined at discrete time spots  $t$ . Furthermore, while for  $\lambda$  in Equation (3.12) there is no obvious upper bound, the maximum (reasonable) decomposition scale denoted by  $J$ , can be derived by the signal's length.

A DWT of level  $J$  performs a decomposition of a given signal  $X$  into one scaling coefficient vector  $V_J$  and  $J$  wavelet coefficient vectors  $W_j$ ,  $1 \leq j \leq J$ . The length of each vector is given by

$$W_j \in \mathbb{R}^{N/2^j} \quad \text{and} \quad V_J \in \mathbb{R}^{N/2^J}. \quad (3.13)$$

Applying on these vectors the inverse DWT yields an additive decomposition of the original signal, that is

$$X = S_J + \sum_{j=1}^J D_j, \quad (3.14)$$

in which  $S_J$  is the scaling approximation and  $D_j$  stands for the different detail vectors. The scaling approximation is the outcome of a weighted moving average filter applied on  $X$ , with the weights determined by the wavelet and the filtering bandwidth determined by  $J$ . Though mathematically this bandwidth also depends on the wavelet and its basic filter length  $L^{\text{wvlt}}$ , the filter is associated with a bandwidth of  $2^J$  for practical reasons and, therefore, independent of  $L^{\text{wvlt}}$ . Also, the relation

$$S_{\tilde{J}-1} = S_J + \sum_{j=\tilde{J}}^J D_j, \quad (3.15)$$

holds, that is, the detail vectors represent the details lost at every coarser approximation level, yielding a dyadic *multiscale decomposition*. Thus, the intuition behind this decomposition is that one successively applies each time on level  $j$  a low-pass (i. e., the scaling) filter on each preceding scaling approximation of the next lower level  $j - 1$ , rendering a filter cascade. The separated differences between those two approximation levels of different coarse resolutions are then the outcome of the respective high-pass (i. e., wavelet) filter on the same lower level approximation  $S_{j-1}$ . Though every scaling approximation  $S_j$ ,  $1 \leq j < J$  can be reconstructed, based on the vectors given in Equation (3.14) usually one is interested only in those vectors named there.

As was seen in Section 3.1, different wavelets have supports of different lengths, hence the scaling approximations  $S_j$  are associated with different effective filter lengths (i. e., moving window filter sizes). For every wavelet these window sizes on scale  $j$  are given by  $L_j^{\text{wvlt}} = (2^j - 1)(L^{\text{wvlt}} - 1) + 1$ . However, this is not of much relevance in practice, since many of the scaling filtering coefficients  $g_i$  near the borders  $g_0$  and  $g_{L-1}$  are very close to zero, they are not of much relevance to the filtered outcome. Thus, for practical purposes it suffices to ignore the length of the wavelet filter, and assume instead for every scale  $j$  the scaling approximation  $S_j$  (or the scaling coefficient vector  $V_j$ ) to be associated with

(weighted) moving averages over  $\lambda_j = 2^j$ . A similar argumentation holds for the wavelet filter coefficients  $h_i$ , for which  $D_j$  and  $W_j$  are associated with (weighted) differences over  $\tau_j = 2^{j-1}$ .

The calculation and derivation of  $W = [W_1, \dots, W_J]$  and  $V_J$  as well as  $D_j$  and  $S_J$  can be done via the matrix-vector multiplication

$$W = \mathcal{W}X \quad \text{and} \quad X = \mathcal{W}^\top W,$$

with  $\mathcal{W} = [\mathcal{W}_1, \dots, \mathcal{W}_J, \mathcal{V}_J]$  determined by the wavelet and scaling filter banks. However, as this is computationally inefficient, the by [74] introduced *pyramid algorithm* is used instead, which has a computational complexity of  $\mathcal{O}(N)$ .

Although the DWT has been used in many applications to detect cusps or jumps (see also Section 2.4.1) it proved to be unsuitable for the here developed approach. This is due to the fact that the by the DWT provided  $S_J$  and  $D_j$  series are not associated with zero phase filters. That means, when circularly shifting the original time series  $X$  by a certain amount, the decomposition output will not simply be the series shifted by that same amount, but rather a series of a completely different shape. A very simple and intuitive example of this is given in [88]. Thus, the multiresolution analysis' series are sensitive to the point where the measurement of the time series itself begun. This, in addition to the DWT's wavelet coefficient vectors being of different lengths as stated in Equation (3.13), renders the DWT not convenient for the new approach, as will be seen in Section 4.1.1. Though the possibility that the realization of a similar approach or extension that will be presented in the next chapter may also be feasible for the DWT is not excluded explicitly, for certain procedures it does not see fit, as will be outlined at the respective steps of the algorithmic solution proposed in this thesis.

### The Maximal Overlap Discrete Wavelet Transform

The MODWT<sup>2</sup>, upon which the algorithm is based, has already been considered extensively in the literature, see [88] and the references therein.

---

<sup>2</sup>In the literature this discrete wavelet transform is also known under the name undecimated, non-decimated, shift invariant, translation invariant, time invariant, and stationary DWT.

As for the DWT, the MODWT of level  $J$  performs a decomposition of a given signal  $X$  according to Equations (3.14) and (3.15). However, unlike for the DWT, all wavelet and scaling coefficient vectors yielded by the MODWT are in  $\mathbb{R}^N$ , with  $N$  the length of signal  $X$ . The reason for choosing the redundant MODWT over the ordinary one-to-one DWT is due to the fact that the former transform is associated with *zero phase* filters, that is, its filtered output detail and approximation series are *shift-invariant*. Roughly speaking, if the signal  $X$  is shifted by  $\Delta t$ , unlike in the DWT, the MODWT generated  $V_J$ ,  $W_j$ ,  $S_J$  and  $D_j$  series will shift by  $\Delta t$  as well. As the in the next chapter developed algorithm depends on local detail reconstruction, this feature is vital. Due to the inherent redundance, the price to pay is a higher computational complexity of  $\mathcal{O}(N \log_2 N)$  compared to  $\mathcal{O}(N)$  of the DWT. Yet, this complexity class is the same as one has for the fast Fourier transform (FFT), see [19].

Principally, the MODWT wavelet and scaling filter banks, denoted by  $\tilde{h}_l$  and  $\tilde{g}_l$  respectively, can be calculated analogously as they were derived for the DWT. However, an easier way is to derive them directly from the DWT filter banks through the following relationships, that is

$$\tilde{h}_l = h_l/\sqrt{2} \quad \text{and} \quad \tilde{g}_l = g_l/\sqrt{2}. \quad (3.16)$$

Note that the MODWT is no longer an orthogonal transform like the DWT. This may be a drawback for certain applications in practice, but it has no effect on the analyses of time series at hand. The most important effect this has is that with an orthogonal transform white (i. e., uncorrelated) noise will also yield uncorrelated wavelet coefficients carrying the information of the noise, that is, the white property carries over from the signal domain to the wavelet coefficient vectors. Nevertheless, denoising schemes that build upon, for example, white Gaussian noise, remain still applicable for the MODWT, though with slight adaptations.

### Common Remarks

The vectors provided by both the DWT and the MODWT depend on the wavelet used, though the dependence in case of the MODWT is much less than for the DWT. Different wavelets are thus used for different reasons: While the *Haar* wavelet has very

small support and forms jumps and edges very well, wavelets of higher orders, such as the *Daubechies* (*D4*) and the *least asymmetric* (*LA8*), are bases of higher smoothness spaces but have also bigger support. This ensures a smoother shape of  $S_j$ ,  $1 \leq j \leq J$ , at the expense that each wavelet and scaling coefficient carries more information due to the increased filter width, and is thus not as local as wavelets of lower orders. Although the algorithm proposed in this thesis has been designed to work with any wavelet, the representation of jumps or sudden regime changes in an otherwise smooth trend is the main intention here. Therefore, in this work the focus lies on the wavelets named above, being the ones most applied in practice.

As with other filters, wavelet transforms can be classified into the two classes of infinite and finite impulse response filters, commonly denoted by IIR and FIR, respectively. Given a signal (either continuous or discrete) that consists of a single impulse only and is zero otherwise, filters can be classified whether their impulse response, that is the filtered output of this impulse signal, is either finite (i. e., has compact support) or not. It is easy to see, that in terms of moving windows filters an IIR filter corresponds to a filter with an infinite window size, that is the filter function or filter coefficient series has no compact support. Thus, they are also said to have infinite memory, as regardless of where the moving filter window is centered, the impulse still has some effect on the output. Analogously, FIR filters are characterized by a finite window size, as is common with the (non) linear filters presented in Sections 2.2 and 2.3.

From the above discussion it is easy to see that discrete wavelet transforms (either DWT or MODWT) are always finite impulse response filters for wavelets with compact support. That is, for each of these transforms the impulse response also has compact support. Infinite impulse response filters are also possible for both the CWT and DWTs using wavelets with infinite support. Though these wavelets are still localized (while decaying sufficiently fast towards zero), they always maintain some part in each impulse response coefficient calculation, as the filtering windows size is also infinite. This, of course, holds only for theoretical analyses. In numerical implementations the filtering window size must be limited for reasons of computational feasibility, and thus, the wavelet cut off beyond a certain threshold.

Note that in this work the scales are denoted by  $1 \leq j \leq J$ , with  $J$  wavelet coefficient vectors  $W_j$  and detail series  $D_j$ , and one scaling coefficient vector  $V_J$  and the corresponding scaling approximation series  $S_J$ . In this case, for the highest scale  $J$  the series  $S_J$  is the coarsest approximation in the multiresolution analysis. That means that the lowest scale  $j = 1$  contains the highest frequencies.

In summary, discrete wavelets can be interpreted as moving weighted average filters and thus, linear filters. What makes their usage favorable is the multiscale analysis they yield, while remaining computationally efficient. Furthermore, the coefficient vectors allow to analyze and compare distinct features on different scales at the same time, analyzing probable relations. A very thorough mathematical treatment of the above topics is provided in [118].

### Wavelet Transfer and Squared Gain Functions

The discrete wavelet transform is essentially a linear filter and, therefore, it can also determine a priori how frequencies in the signal  $X$  will be attenuated, that is, to what degree they will pass to the final detail and approximation vectors  $D_j$  and  $S_J$ , respectively. However, unlike the ordinary linear filters introduced in Section 2.2, there is not only one low-pass filter that blocks the higher frequencies in a certain range, but rather, due to the multiscale decomposition, there are a series of high- and band-pass filters that are linked to the wavelet filters and their respective coefficient and detail vectors. Additionally, there is one low-pass filter that refers to the scaling filter and its approximation  $S_J$ . These functions are in relation to one another as follows.

Given the basic wavelet and scaling filter coefficients (i. e., the weights) as in Equation (2.2), one can derive their transfer functions (see Equation (2.3))  $H$  and  $G$  as usual via the discrete Fourier transform, that is

$$H(f) = \sum_{j=0}^{L-1} h_j e^{-i2\pi f j} \quad \text{and} \quad G(f) = \sum_{j=0}^{L-1} g_j e^{-i2\pi f j},$$

where  $h_j$  and  $g_j$  are the respective wavelet and scaling weights, i. e., filter coefficients. As each decomposition level corresponds to a filter cascade (i. e., a sequence or concatenation

of filters) the transfer functions for each level  $2 \leq j \leq J$  are given through

$$G_j(f) = \prod_{i=0}^{j-1} G(2^i f)$$

and

$$H_j(f) = H(2^{j-1} f)G_{j-1}(f) = H(2^{j-1} f) \prod_{i=0}^{j-2} G(2^i f). \quad (3.17)$$

From these functions the according squared gain functions can be derived, which determine how frequencies are attenuated for each scale. Equation (3.17) also provides a clear notion that each scaling approximation on level  $j$  is a concatenation of the previous  $1, \dots, j-1$  scaling filters, while the wavelet filter  $H_j$  results from applying the next lower level filter  $H_{j-1}$  onto the scaling approximation of the very same level. As remarked in Section 2.2.2 there exist no exact frequency stops, which holds for wavelet transforms being linear filters, as well. For practical issues the low-pass scaling filter related to any Daubechies high- and band-pass wavelet filters are associated with frequency pass-bands

$$0 \leq |f| \leq 1/2^{J+1} \quad \text{and} \quad 1/2^{J+1} \leq |f| \leq 1/2^j,$$

respectively. This association is independent of any specific wavelet chosen, and is thus only to be considered to be a practical convenience, though following standard engineering practice.

The reason why in wavelet analysis one often refers to the squared gain function and not only to the gain function is due to the wavelets construction process. One can derive a condition on this squared gain function that is equivalent to Equations (3.10) and (3.11), that is,

$$\mathcal{H}(f) + \mathcal{H}(f + \frac{1}{2}) = 2 \quad \text{or alternatively} \quad \mathcal{G}(f) + \mathcal{G}(f + \frac{1}{2}) = 2$$

for all  $f$ . Due to the direct dependence between the scaling and wavelet filters (see

Equation (3.9)), this condition can alternatively be formulated as

$$\mathcal{G}(f) + \mathcal{G}(f + \frac{1}{2}) = 2 \quad \text{or} \quad \mathcal{G}(f) + \mathcal{H}(f) = 2.$$

Above statements hold for all DWT transfer and squared gain functions. As in Equation (3.16) their MODWT counterparts are also related to the DWT ones through

$$\tilde{H}(\cdot) = H(\cdot)/\sqrt{2} \quad \text{and} \quad \tilde{G}(\cdot) = G(\cdot)/\sqrt{2}, \quad (3.18)$$

which leads to analog conditions for the MODWT.

**Example 3.2.** In Figures 3.1 and 3.2 the Haar and  $D4$  squared gain functions for both the DWT and the MODWT are depicted, with the scaling level set to  $J = 2$ .<sup>3</sup> For both transforms the squared gain functions differ only in magnitude, but not in shape. This is in accordance to the scaling of filter coefficients in Equations (3.16) and (3.18). Furthermore, it can be seen clearly how  $G_1$  is always a low-pass filter,  $H_1$  a high-pass filter, and  $H_2$  a band-pass filter. For higher decomposition levels, i. e.,  $J > 2$ , the wavelet filters for  $j > 1$  form additional band-pass filters. However, from the perspective of the level itself, i. e.,  $G_{j-1}$ ,  $G_j$  is always a low-pass, and  $H_j$  always a high-pass filter.

### Handling Boundary Distortions

As with all moving window filters, the question arises how to handle, i. e., to filter, the data points at the extreme ends of the signal, when the window passes over the signal's natural (i. e., measured) length. In this case, there are several alternatives the analyst can choose from.

1. One possibility would be to ignore these points completely, that is, if  $L_W$  denotes the moving window size, the signal is only filtered until the window reaches the boundaries of  $X$ . Thus, instead of a filtered output  $\hat{X}_1, \dots, \hat{X}_N$  for the whole signal, this yields only  $\hat{X}_{L_W/2}, \dots, \hat{X}_{N-L_W/2}$ . This shortened signal may not be

---

<sup>3</sup>These figures were plotted using the *WMTSA Wavelet Toolkit for MATLAB*, available at <http://www.atmos.washington.edu/~wmtsa/>, last accessed on the 22th of November 2010.

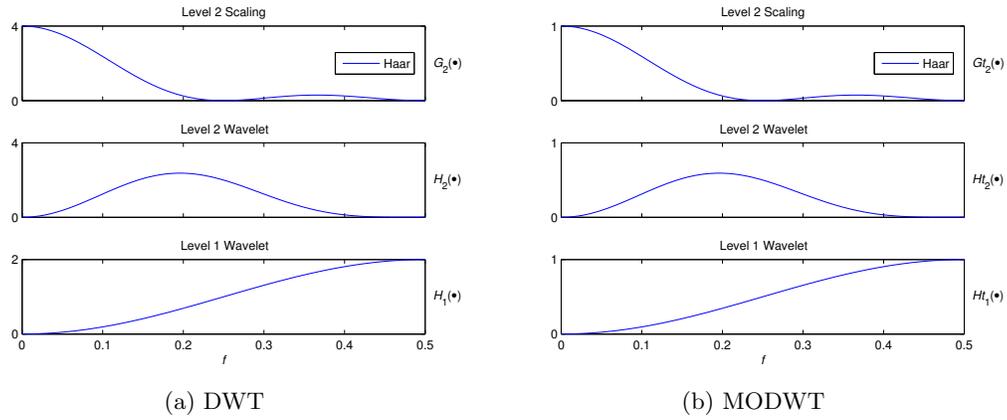


Figure 3.1.: Haar squared gain functions

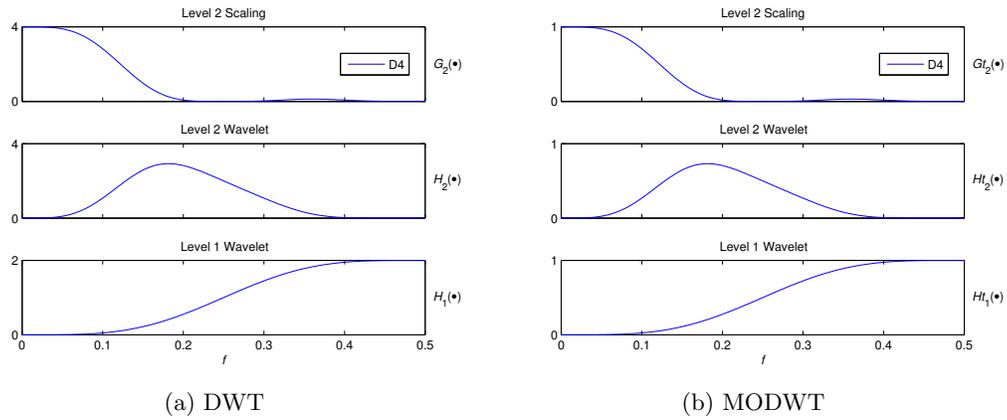


Figure 3.2.:  $D_4$  squared gain functions

satisfying if one is interested in the end points of the signal as well, even if they are biased in some way.

2. As the moving window approaches the end of  $X$ , one can also cut the window (i. e., shorten its length) and adapt the weights of the filter accordingly. Though no bias is introduced in this way, several properties of the filter (e. g., its frequency attenuation, see Section 2.2.2) change or probably do not hold any longer.
3. Another approach is to not ignore the extreme filtered points but to extend the signal artificially beyond its measured extremes by methods that try to minimize the bias introduced by this extension. The most common methods in practice are as follows.
  - a) The most simple method is to set the required data points  $X_{-L_W/2}, \dots, X_0$  and  $X_{N+1}, \dots, X_{N+L_W/2}$  to zero. This, of course, is not a reasonable choice in most applications and thus this trivial method should only be used in exceptions, for example, when the following alternatives are not reasonable due to some reason.
  - b) The missing data points are estimated according to an extension of the original signal via a linear, polynomial or spline regression curve through the latest data points of  $X$  at both ends, see [24] for further details on these procedures. Depending on the model, in many cases this approach delivers reasonable results, but can also introduce an additional bias, if the model was chosen badly.
  - c) The procedure favored in this thesis and which is applied throughout this work for LLSA as well as the benchmarks compared to it, is the one of circular extension, that is, the respective first (last) points of  $X$  substitute the required unknown points at the other end. Though this method must be used with caution and usually only on signals where it is reasonable (i. e., where circularity can be assumed in some way) it works well in practice.

Since any of the methods in 3b and 3c introduce an unknown bias, the choice should depend on the signal itself and/or the analysts preferences. However, circular extension is certainly less computationally extensive than the other non-trivial methods.

**Remark 3.3.** The reader is urged not to confound the circular boundary extensions with the circular shifts that will be introduced in Section 4.1.1. While above circular boundary extensions enable to filter a signal  $X$  at its borders where the window exceeds the end of the measured signal, the later introduced circular shifts serve only to align the different coefficient vectors in time.<sup>4</sup>

### 3.3. Wavelet Trend Extraction and Denoising Methods

The vector  $S_J$  is the trend estimation of a wavelet transform from the perspective of the wavelet used. As was mentioned, the output from the DWT is much more dependent on the choice of wavelet than the MODWT. From a practical point of view this already favors usage of the MODWT, as the result will generally be more robust, that is, in this case independent of the particular choice of wavelet. Furthermore, the estimated trends of the DWT reflect the structure of the wavelet (i. e., a blocky structure for the Haar wavelet, shark fins for the  $D4$ , and so on), which renders the result being biased (i. e., the time series itself is not the only main decisive factor) by the wavelet choice and makes it difficult to compare the output and its performance to other benchmarks (consisting of either other wavelets or distinct filtering methods). For example, the DWT employing the Haar wavelet estimates a trend consisting only of jumps itself.

Furthermore, as the focus of the here considered analysis is on the recognition and representation of singular, isolated events in a time series, these events should also be captured with a high accuracy. The DWT is also not robust in this case: As the shape of the estimated trend as well as the detail series are dependent on the relative starting point of the time series, measuring the same time series from slightly different starting points will usually lead to different coefficient vectors and output series. This is unfavorable, as it means that even for the same wavelet and the same time series one cannot rely on

---

<sup>4</sup>This displacement in time happens during the transformation into wavelet coefficient vectors, using the pyramid algorithm.

these results, that is, for different starting points also different jumps and slopes might be estimated.

Thus, the non-robustness of the DWT with respect to the choice of wavelets as well as the starting measurement point makes it unfavorable for the purpose of jump analysis. The MODWT, however, is insensitive to these two factors, that is, since the MODWT can always be associated with a zero phase filter, analyzing the by  $m$  shifted time series  $\mathcal{T}^m X$  will simply result in equivalently shifted vectors  $\mathcal{T}^m D_j$  and  $\mathcal{T}^m S_J$ . The scaling and wavelet coefficient vectors are also shifted accordingly (i. e., they are shifted versions of the original vectors), though the amount of shift depends on the wavelet as well as on the scale  $j$  (q. v., Section 4.1.1).

Regarding the wavelet, though there is still a difference in the filtered output  $S_J$  dependent on the choice of wavelet, it is insignificant. One only has to consider that, due to the different supports each wavelet possesses, there might be larger differences near the boundaries, as the moving window size depends on the scale as well as on the wavelet's support. However, this is the same issue with the other (non) linear filters presented in Sections 2.2 and 2.3.

Yet, the choice of wavelet still remains an important issue in the MODWT analysis due to their different transfer (and thus, squared gain) functions, as they provide different frequency passbands. Also, when considering the primary goal, though the differences for each wavelet in  $S_J$  are negligible, the representation of jumps and steep slopes in long-term trends is carried out with different performance and effects. This topic is further discussed in Sections 5.1.2 and 5.2.1.

When considering nonlinear wavelet transforms or the manipulation of wavelet coefficient vectors as LLSA does for detail reconstruction, readers proficient in the area of wavelets may think about the well-known works of [35–37] and the many applications building upon their framework. They propose several thresholding and shrinkage methods dependent on the signal's inherent noise (given that it can be measured or estimated), to obtain a signal without this noise.

The idea behind these approaches is that if a deterministic signal is corrupted by

(additive) noise as in Equation (2.1), the relevant information, that is, the information that contains information about the signal itself but not the noise, is carried mainly in the wavelet coefficients exceeding a certain threshold  $\delta$ . All coefficients below this threshold  $\delta$  can be contributed to noise. Several techniques have been proposed to handle (i. e., adjust) these low value coefficients and their transition to the ones exceeding the threshold. The most common methods are hard and soft thresholding. Hard thresholding simply sets all coefficients below  $\delta$  to zero, that is, for all  $1 \leq j \leq J$  holds

$$W_{j,t} = \begin{cases} W_{j,t} & \text{for } |W_{j,t}| > \delta \\ 0 & \text{otherwise.} \end{cases} \quad (3.19)$$

Further employed thresholding techniques are soft, mid and firm thresholding. They differ from hard thresholding in the way that coefficients not exceeding a certain threshold (not necessarily  $\delta$ ). Additionally to thresholding there exists the akin concept of shrinkage. Shrinkage, by definition, can differ from thresholding in the sense that nonzero coefficients are *always* scaled to nonzero values.

Above cited works rely on specific kinds of noises, mostly additive white (i. e., independent and identically distributed) Gaussian noise, for which these rules can be shown to be asymptotically optimal or optimal in other senses or measures. Additionally, these works and have been extended to other noise settings like non-Gaussian and correlated noise, also including level dependent thresholds, see [6, 63, 85].

Though these denoising and signal estimation schemes and thresholds were originally developed for the DWT only, with slight modifications (due to the strong connections between the filter coefficients in both transforms) they can be applied to the MODWT as well, and thus, are also a promising candidate. Therefore, in the literature there have been considered several kinds of thresholding, denoising and shrinkage rules for the original as well as the undecimated DWTs.

All above cited denoising techniques, though they have shown to perform (asymptotically) optimal under certain assumptions and requirements, suffer certain drawbacks: First of all, they assume that the noise structure (e. g., white or correlated Gaussian

noise) and its variance (or the respective threshold) are known or can be estimated through various techniques like cross-validation (see [83]) or via Stein's unbiased risk estimator (SURE) (see [106]). Progressing works generalize these works even more, that is, they weaken these requirements, and extend the notion of thresholding and shrinkage to, for example, non-necessarily Gaussian noise [10, 11]. Yet, all these approaches base their method on the assumption that some kind of stochastic noise is present. However, this does not include the notion of, for example, medium- and short-term deterministic business cycles or fluctuations that follow a regular pattern. These kind of phenomena would still be included if one uses denoising techniques not for signal, but for trend estimation. Furthermore, this procedure leaves it unclear where to classify jumps, which also can be seen as coming from a noise distribution with heavy tails, which has been elaborated, for example, in the financial sector, see [94, 95].

Though both approaches can be considered similar at first glance, they fundamentally differ from each other: As [99] already observed, there is a distinct difference between the complementary approaches of denoising (as it is done in above references) and noise smoothing, which is this thesis' method for trend estimation. While denoising assumes some kind of well-defined noise present in the signal and attempts to separate one from another, noise smoothing often assumes that the noise (being of some kind maybe not more explicitly specified, including stochastic as well as deterministic components) can be extracted by some form of averaging over the signal, which usually tends to smooth out sharp features of the underlying signal. Note that in case of, for example, additive Gaussian noise, one can expect both methodologies to approximately deliver equal results, as averaging over normally distributed noise should render the same effect as removing it, given an appropriate filtering window size.

Thus, the latter method of smoothing implicitly contains the basic assumption that the signal itself, which is corrupted by noise, is smooth. This of course does not hold for this work, where a signal containing sudden changes as the final result of the smoothing process is explicitly expected. Hence, in this particular case, the notion of trend estimation via smoothing weighted averages, as it is also seen from a wavelet perspective, is not appropriate and cannot applied any longer as this. The algorithm developed in the

next chapter tries to achieve both: A filtered signal that captures the signal's features like sharp jumps and steep slopes, and is smooth otherwise.

### **3.4. Summary**

In this chapter wavelets and their several transforms were introduced. In the discrete setting wavelets are usually seen as filter banks that decompose via the discrete wavelet transform a given signal into several scales. Their characteristics (e. g., linearity and locality) and their already established application fields in jump detection, signal approximation, and noise reduction indicate the wavelets' suitability to serve as a base for a new approach that satisfactorily handles the tasks stated in Section 2.1, and at the same time fulfills the requirements in Section 2.5.

## Chapter 4.

# The Local Linear Scaling Approximation

In this chapter the main contribution of this thesis is given. The algorithm is introduced in Section 4.1, comprising its derivation, a mathematical subsumption and remarks, as well as its usage. The algorithm's properties, that are, its computational complexity, the local linear filtering property, as well as its impulse and step response function, are shown in Section 4.2.

### 4.1. Methodology and Implementation

This section discusses the algorithm itself. In Section 4.1.1 the local linear scaling approximation (LLSA) is derived. Section 4.1.2 summarizes the algorithm and states additional remarks. Section 4.1.3 contains further information about implementation issues and the concrete usage of LLSA.

#### 4.1.1. Derivation

Nonlinear filters often combine nonlinear with subsequent linear operations to further smooth out details having passed the nonlinear edge preserving filtering procedure. The MODWT multiscale decomposition enables the development of an algorithm that follows a different approach. The main idea is to start with the very coarse approximation  $S_J$  and reconstruct the lost details near the jumps that have been smoothed out, by using the information contained in the detail series  $D_j$ . The main difference, contrary to

many methods presented in Sections 2.2 to 2.4, is that the algorithm does not apply the same filtering rule throughout the whole time series (this is only done for achieving the starting series  $S_J$ ). Instead, the final nonlinear filtering rule, consisting of the primary linear filtering process and the nonlinear reconstruction of details, adapts itself according to the data. This data dependent approach is thus, localized like the wavelets themselves.

The basic procedure of LLSA is as follows. Based on what was discussed in Chapter 2, it can easily be seen that the approach must start with a linear filter, as otherwise Requirement R2 could hardly be fulfilled anymore. Therefore, this linear filter (i. e., the MODWT) is then extended into a nonlinear procedure that allows for a more accurate display of sudden changes, thus, satisfying Requirement R1. The algorithm then detects these sudden changes in order of their significance employing a heuristic rule that also might be exchanged, depending on the information available on the time series (see Section 6.2). However, as will be seen, this detection is interweaved with the reconstruction of details. This step is the most crucial one of the algorithm, as it determines whether in the modified output signal all details of jumps and other phenomena are captured, and if breaks occur between the smooth and refined areas of the output signal.

In its very basic design, the LLSA algorithm requires that in addition to the scaling level  $J$  and the wavelet, the *expected* amount of jumps  $K$  to be stated, along with  $0 \leq \Lambda \leq J$ , which determines up to which scale details should be reconstructed. These minimal additional input parameters that can be set heuristically ensure that the proposed algorithm is still applicable on all discrete signals the MODWT can be applied onto, in this way attending Requirement R3. Then, LLSA automatically detects the regions to be refined and restores details, thus improving the shape of jumps and slopes that are blurred in  $S_J$ .

The main intuition behind LLSA is to analyze the abstract jump wavelet coefficient structure one has in  $W_j$  and to carry this structure over to the actual signal. As depending on the respective nature of the sudden change, the heights of this structure might change as well. The idea is that these structures can still be determined through their change of signs that are also independent of the respective scale  $j$ , but nonetheless depend on the wavelet used. Thus, considering the necessary change of signs on the wavelet

coefficient plane to capture a pure jump without any noise, this also (approximately) holds for a noisy time series. Using this observation complete sudden phenomena can be captured without adding unnecessary details to the output series.

Note that in this section the terms *reconstruction* and *refinement* are used analogously. From the viewpoint of a moving window filter the procedure of LLSA equals the narrowing of the bandwidth of the filter, that is, the window size. However, from the viewpoint of the time series itself and its multidecomposition, it is a reconstruction of details lost on scale  $J$ .

In the following, the algorithm is stated in detail, by deriving it step by step. As the computations for each scale  $j$  depend on the results derived on scale  $j + 1$ , first the wavelet coefficient vectors have to be aligned. Then, the algorithm's rule of detecting the first jump on scale  $J$  is explained, followed by the determination of the area in which lost details should be reconstructed. After that, by applying the same rules, the algorithm successively determines the remaining  $K - 1$  jumps and their areas. Having determined all jumps on scale  $J$ , a modified rule is applied on the lower scales  $1 \leq j < J$ . Finally, the lost jump details are reconstructed.

### Aligning the Wavelet Coefficient Vectors

When applying the MODWT onto  $X$ ,  $J$  wavelet coefficient vectors  $\overline{W}_j$  are yielded, which, as derived through the computationally efficient MODWT pyramid algorithm (first introduced for the DWT by [74]), are not exactly aligned with the events in  $X$ . Yet, as the reconstruction of jump details on lower scales depends on the local information derived on the higher scales, LLSA is sensitive to coefficient vectors not aligned correctly. The correctly aligned vectors  $W_j$  are derived by circularly shifting each  $\overline{W}_j$  by  $-L_j^{\text{wvlt}}/2$ , that is,

$$W_j = \mathcal{T}^{-L_j^{\text{wvlt}}/2} \overline{W}_j, \quad (4.1)$$

with  $\mathcal{T}^{-1}$  the circular left shift operator

$$\mathcal{T}^{-1}W = \mathcal{T}^{-1}[w_1, w_2, \dots, w_{N-1}, w_N] = [w_2, w_3, \dots, w_{N-1}, w_N, w_1]$$

and

$$\mathcal{T}^{-n}W = \mathcal{T}^{-n+1}\mathcal{T}^{-1}W, n \in \mathbb{N}$$

defined successively. The right shift operator  $\mathcal{T}^n$  is defined analogously.  $L_j^{\text{wvlt}}$  denotes the effective wavelet filter width

$$L_j^{\text{wvlt}} = (2^j - 1)(L^{\text{wvlt}} - 1) + 1$$

for each scale  $j$  and  $L^{\text{wvlt}}$  the basic filter length as stated in Table 4.1. Note that the alignment shift in Equation (4.1) holds only for the wavelets named in Section 3.2. For others, the shift-amount might change dependent on the order and type of the wavelet (see [88] for further details).

### First Jump Detection

In Section 2.4.1 a number of references that consider jump detection were already cited. In this thesis, a simple rule of determining jump locations is considered, which, however, can be substituted or combined with any other of these methods, as is depicted in Chapter 6. The methodology of LLSA is based on the observation that the wavelet coefficients on scale  $J$  denote the differences between the weighted moving averages in  $S_J$ . Thus, as one is usually interested in refining successively only those jumps containing the highest potential of improvement for  $S_J$ , the first index indicating the biggest jump is determined by

$$l^W := \operatorname{argmax}_t(|W_{J,t}|).$$

This provides the first sudden change with the highest irregularity from the point of view of the original smoothed signal  $S_J$ , ignoring the wavelet coefficient vectors  $W_j$  associated with higher frequencies on the lower scales  $1 \leq j < J$ . Note that this step is not directly dependent on the wavelet used (though the wavelet coefficient vector  $W_J$  is).

### First Reconstruction Boundary Determination

Having determined the first jump, the boundaries around it (in which the details shall be restored) have to be fixed. It is presupposed that these boundaries are set accordingly to ensure that

- The complete jump should be captured,
- No unnecessary details beyond the jump should be added, and
- The transition between the modified sections and the original output  $S_J$  should be smooth (i. e., with no artificial jumps being added).

The heuristic rule to be derived, coping these requirements by determining the boundaries to each side of the jump, depends on the wavelet. The intuition behind this rule is the observation that with respect to jumps the general structure of the wavelet coefficients remains the same, whether any noise is present or not. This is due to the fact that higher scales almost only contain low frequencies and are, thus, hardly affected by high-frequency noise. Therefore, if one analyzes the jump wavelet coefficient structure<sup>1</sup> for a particular wavelet and transfers it to the real signal, this will approximately capture the noisy jump there as well. The following definition states the relevant parameters for this procedure.

**Definition 4.1.** *Be  $X^s$  a signal containing a single step (and being constant otherwise), and be  $W_j^s$ ,  $1 \leq j \leq J$  the wavelet coefficient vectors of the MODWT applied on  $X^s$ . With*

$$l_j^s := \operatorname{argmax}_t(|W_{j,t}^s|)$$

and

$$l_{\min,j}^s := \min\{t \mid t \in \operatorname{supp}(W_j^s)\}, \quad (4.2)$$

$$l_{\max,j}^s := \max\{t \mid t \in \operatorname{supp}(W_j^s)\}, \quad (4.3)$$

---

<sup>1</sup>Another naming convention would be *wavelet coefficient step response*, see Section 4.2.3

Table 4.1.:  $n_\alpha$ ,  $n_\beta$  and  $L^{\text{wvlt}}$  for different wavelets

Wavelet	$n_\alpha$	$n_\beta$	$L^{\text{wvlt}}$
Haar	1	1	2
$D4$	2	4	4
$LA8$	5	4	8

one defines

$$n_{\alpha,j}^{\text{wvlt}} := \sum_{t=l_{\min,j}^s}^{l_j^s-1} \frac{|\text{sign } W_{j,t+1}^s - \text{sign } W_{j,t}^s|}{2}, \quad (4.4)$$

$$n_{\beta,j}^{\text{wvlt}} := \sum_{t=l_j^s+1}^{l_{\max,j}^s} \frac{|\text{sign } W_{j,t-1}^s - \text{sign } W_{j,t}^s|}{2}. \quad (4.5)$$

As the general shape of this jump wavelet coefficient structure in  $\text{supp}(W_j^s)$  remains the same for all  $1 \leq j \leq J$  and all wavelets considered in this work, one may write

$$\begin{aligned} n_\alpha^{\text{wvlt}} &= n_{\alpha,j}^{\text{wvlt}}, \\ n_\beta^{\text{wvlt}} &= n_{\beta,j}^{\text{wvlt}} \end{aligned}$$

for all  $1 \leq j \leq J$ . In Figure 4.1 the  $W_j^s$  for the Haar and  $D4$  wavelets are depicted. Also, keeping in mind that  $n_\alpha^{\text{wvlt}}$  and  $n_\beta^{\text{wvlt}}$  depend on the wavelet (because the  $W_j^s$  vectors themselves depend on the wavelet used in the transform), the identifier for the wavelet can be omitted and simply written as  $n_\alpha$  and  $n_\beta$ . For the Haar,  $D4$  and  $LA8$  wavelet these values are stated in Table 4.1.

**Remark 4.2.** Theoretically this approach only works for wavelets with compact support, as otherwise in Equations (4.2) and (4.3) would always yield

$$l_{\min,j}^s = -\infty, \quad l_{\max,j}^s = \infty$$

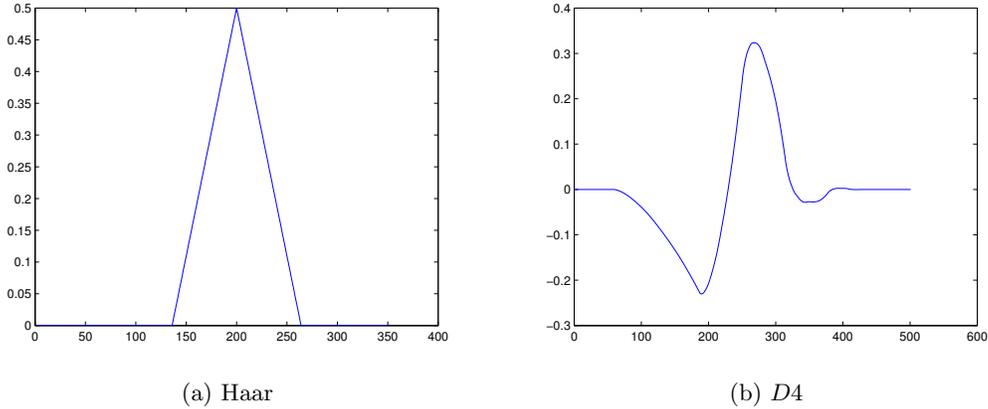


Figure 4.1.: Wavelet coefficient structure for step functions

and  $n_\alpha = n_\beta = \infty$ . Yet, in practical applications, as the wavelets themselves decay exponentially, their small coefficients can be cut off, with an assessable loss of precision.

In order to reconstruct the whole jump in  $X$  the left boundary  $\alpha$  is set to be the maximal index so that at least  $n_\alpha$  change of signs occur between  $\alpha$  and  $l^W$ . Also, the right boundary  $\beta$  is set to be  $n_\beta$  according minimal index. Thus, the boundaries satisfying above requirements are given by

$$\alpha = \max \left\{ l \in [1, l^W - 1] \left| \sum_{t=l}^{l^W-1} \frac{|\text{sign } W_{J,t+1} - \text{sign } W_{J,t}|}{2} \geq n_\alpha \right. \right\} \quad (4.6)$$

$$\beta = \min \left\{ l \in [l^W + 1, N] \left| \sum_{t=l^W+1}^l \frac{|\text{sign } W_{J,t-1} - \text{sign } W_{J,t}|}{2} \geq n_\beta \right. \right\}. \quad (4.7)$$

This area around the jump is denoted by  $\Omega^W := [\alpha, \beta]$ .

The here depicted procedure distinguishes LLSA from other alternative filtering procedures. As LLSA essentially enables a refinement that could also be achieved by narrowing any other filter's bandwidth, only the above heuristic employs the possibility

to automatically determine these refinement sections without requiring any additional thresholds to determine their limits.

**$k$ th Jump Detection and Boundary Determination on Scale  $J$**

Detecting further jumps follows the same procedure as above, with the difference that the algorithm has to exclude already detected jumps and their surrounding areas. For this, it suffices to define

$$\Omega_1^W := \Omega^W$$

and subsequently, for detecting the  $k$ th jump,  $1 < k \leq K$ , to determine

$$l_k^W := \operatorname{argmax}_t (|W_{J,t}| \setminus \bigcup_{i=1, \dots, k-1} \Omega_i^W). \quad (4.8)$$

As above, the corresponding  $\Omega_k^W := [\alpha_k, \beta_k]$  are set accordingly, whereas in Equations (4.6) and (4.7)  $l^W$  is substituted by  $l_k^W$ .

**$k$ th Jump Detection and Boundary Determination on Scales  $j < J$**

The above procedure yields  $\Omega_k^W$ ,  $1 \leq k \leq K$  for the wavelet coefficients on scale  $J$ . For the lower scales  $J - \Lambda \leq j < J$  the parameters  $l_{j,k}^W$  are determined by

$$l_{j,k}^W := \operatorname{argmax}_t (|W_{j,t}| \mid t \in \Omega_{j+1,k}^W).$$

If one would determine the initial refining points according to Equation (4.8), other jump locations than the ones in  $W_J$  could be detected, as the jumps would be detected according to the point of view of the higher frequencies on the lower scales  $j < J$ . The regions

$$\Omega_{j,k}^W := [\alpha_{j,k}, \beta_{j,k}]$$

covering the jumps are determined as above.

### Detail Reconstruction

Once all areas  $\Omega_{j,k}^W$  have been determined, for  $1 \leq j \leq J$  are set

$$\widetilde{W}_{j,t} = \begin{cases} W_{j,t} & \text{for } t \in \bigcup_{k=1, \dots, K}^{j=J-\Lambda, \dots, J} \Omega_{j,k}^W, \\ 0 & \text{otherwise.} \end{cases} \quad (4.9)$$

This way, only the wavelet coefficients containing information about the jump details are kept, and all other coefficients are discarded by setting them to zero. Applying the inverse MODWT onto the modified circularly backshifted wavelet coefficient vectors

$$\widetilde{\widetilde{W}}_j = \mathcal{T}^{L_j^{\text{wvlt}}/2} \widetilde{W}_j$$

yields the modified detail vectors  $\widetilde{D}_j$ , now containing only the details near the jumps up to scale  $J - \Lambda$ . The adapted approximation is given by

$$S_j^{\text{LLSA}} = S_J + \sum_{j=1}^J \widetilde{D}_j = S_J + \sum_{j=J-\Lambda}^J \widetilde{D}_j. \quad (4.10)$$

Therefore, the algorithm can now be formulated as follows:

#### 4.1.2. Final Formulation and Remarks

Given the input  $(J, K, \Lambda)$  together with the wavelet, the algorithm  $\text{LLSA}(J, K, \Lambda)$  determines for every  $1 \leq k \leq K$  and  $J - \Lambda \leq j \leq J$

$$\Omega_{j,k}^W := [\alpha_{j,k}, \beta_{j,k}] \quad (4.11)$$

with

$$\alpha_{j,k} := \max \left\{ l \in [1, l_{j,k}^W - 1] \left| \sum_{t=l}^{l_{j,k}^W - 1} \frac{|\text{sign } W_{j,t+1} - \text{sign } W_{j,t}|}{2} \geq n_\alpha \right. \right\}, \quad (4.12)$$

$$\beta_{j,k} := \min \left\{ l \in [l_{j,k}^W + 1, N] \left| \sum_{t=l_{j,k}^W + 1}^l \frac{|\text{sign } W_{j,t-1} - \text{sign } W_{j,t}|}{2} \geq n_\beta \right. \right\}, \quad (4.13)$$

and

$$l_{J,k}^W := \text{argmax}_t (|W_{J,t}| \setminus \bigcup_{i=1, \dots, k-1} \Omega_{J,i}^W), \quad (4.14)$$

$$l_{j,k}^W := \text{argmax}_t (|W_{j,t}| \mid t \in \Omega_{j+1,k}^W) \text{ for } J - \Lambda \leq j < J. \quad (4.15)$$

After that, the modified wavelet coefficient sectors are set as in Equation (4.9), onto which the inverse MODWT is then applied, with the final output signal given as in Equation (4.10).

This section is concluded with the following remarks (these were collected here in order not to distract from the algorithm's derivation, to which they are not vital).

### Boundary Distortions

It is important to note that in Definition 4.1 it is assumed that  $X^s$  is large enough, so that the coefficients in  $\text{supp}(W_j^s)$  are not affected by any boundary distortions, i. e.,

$$\forall t \in \text{supp}(\overline{W}_j^s) \text{ holds } t \notin [1, \dots, \min\{L_j^{\text{wvlt}} - 1, N\}]$$

with  $\overline{W}_j^s$  the backshifted (i. e., unaligned) wavelet coefficient vector.

### Simplification for Haar Wavelets

In case of the Haar wavelet, due to its simple jump coefficient structure (see Figure 4.1a), Equations (4.12) and (4.13) simplify to

$$\begin{aligned}\alpha_{j,k} &= \max \{l \in [1, l_{j,k}^I - 1] \mid \text{sign } W_j(l) \neq \text{sign } W_j(l_{j,k}^I)\}, \\ \beta_{j,k} &= \min \{l \in [l_{j,k}^I + 1, N] \mid \text{sign } W_j(l) \neq \text{sign } W_j(l_{j,k}^I)\}.\end{aligned}$$

### Change of Sign for Coefficients Being Zero

The above definitions of  $\alpha_{j,k}$  and  $\beta_{j,k}$  in Equations (4.12) and (4.13) also accounts for wavelet coefficients being zero. These coefficients, with their sign being zero as well, are negligible, as they do not contain any information, that is, they do not add any further details to  $\tilde{D}_j$ . A repeated change from a positive (negative) sign to coefficients being zero and back is treated equal to a change between opposite signs.

### Neglected Selection of Outliers

Note that, fortunately, outliers do not very much affect wavelet coefficients on higher scales and therefore will usually not be selected for refinement before any significant jumps by the rule in Equation (4.14).

### Generalization of $K$ and $\Lambda$

With the formulation of the algorithm as above  $K$  sections are detected and each of them is refined up to  $\Lambda$  scales. By interpreting this as a  $K$ -vector  $(\Lambda, \dots, \Lambda)$  this notion can be generalized by considering the  $K$ -vector  $(\Lambda_1, \dots, \Lambda_K)$  and refine, either in their order of detection or their occurrence in time, the respective section up to  $\Lambda_k$ ,  $1 \leq k \leq K$ . By substituting  $\Lambda$  through  $\Lambda_k$  in Equation (4.15) the formulation of the algorithm remains exactly the same. This is also how the implementation in MATLAB for evaluation of LLSA in this thesis was done, see Section 4.1.3.

### Extreme values for $\alpha_{j,k}$ and $\beta_{j,k}$

Naturally, the extreme values for  $\alpha_{j,k}$  and  $\beta_{j,k}$  are given by 1 and  $N$ , respectively. The disadvantage is that one cannot state any bounds on these prior to any further

analysis of the multiresolution analysis itself. On the other hand, if the extreme values are reached, this is still in complete concordance with the algorithm, as it means that from the perspective of the wavelet coefficients on that respective scale the jump is not yet captured completely. However, the empirical studies (see Sections 5.1.3 and 5.2) lead to the conclusion that these extreme cases usually do not happen in practice, unless the jump itself is near the limits of the time series itself. This possibility was excluded in this implementation, as the jump detection for wavelet coefficients affected by boundary conditions was excluded as well.

### Limited Transition Possibility to the DWT

Considering Equations (4.12) to (4.15) it is clear why only the MODWT is suitable for this approach while the DWT can be seen to be an insufficient starting point. As mentioned in Section 3.2 the length of the DWT generated wavelet coefficient vectors is not equal to  $N$  but with  $N/2^j$  limits the signal length of  $X$  to be a multiple of  $2^j$ . Furthermore, the quantity of wavelet coefficients depends on scale  $J$ , that is, for higher levels there is only a sparser representation that also causes the approximation and detail series to be particularly dependent on the shape of the wavelet. Specifically this sparse representation is it what makes the DWT not suitable for LLSA, as the algorithm requires the step wavelet coefficient structure to be consistent over all scales. For the DWT, there does not even exist a consistent shape due to the sparse (but non-redundant) representation, while additionally the wavelet coefficients differ in shape even if the whole signal  $X$  is only circularly shifted. Therefore, it can be concluded that the DWT is not suitable for this approach based on consistent wavelet coefficient structures.

### 4.1.3. Implementation and Usage

The algorithm was implemented in MATLAB using the *WMTSA Wavelet Toolkit for MATLAB*<sup>2</sup>. LLSA's MATLAB code is freely available<sup>3</sup> and requires, besides above named package, no further components. It is directly usable without any further installation process and is used as follows.

---

<sup>2</sup><http://www.atmos.washington.edu/~wmtsa/>, last accessed on the 22th of November 2010.

<sup>3</sup>To obtain it please contact the author of this work.

Table 4.2.: LLSA default input parameters

Parameters	Values	Default
<code>wavelet</code>	'haar', 'd4', 'la8'	'haar'
<code>J</code>	$0 \leq J \leq \log_2(1.5N)$	$\log_2(\frac{N}{L-1} + 1)$
<code>K</code>	$0 \leq K \leq N$	1
<code>Lambda</code>	$0 \leq \text{Lambda} \leq J$	J

### Basic Usage

The function call is given by

```
[SJt_llsa, SJt_modwt] = llsa(data, wavelet, J, zeros(K,1) + Lambda);.
```

The output consists of the LLSA's and the MODWT's signal approximation vectors  $S_J^{\text{LLSA}}$  and  $S_J$ , `SJt_llsa` and `SJt_modwt`, respectively. `data` denotes the univariate time series vector (either row or column). This is the only mandatory parameter. All other input parameters, if not specified, are set to the defaults shown in Table 4.2. `wavelet` specifies the wavelet filter to be used in the MODWT and its LLSA extension. Though the MODWT works with a variety of other wavelets not considered in this work, for now its usage is restricted to the ones listed in Table 4.1, since LLSA requires the according  $n_\alpha$  and  $n_\beta$  parameters. Scale `J` is restricted to be a positive integer with an upper bound derived through the length  $N$  of the input `data`. Although theoretically `J` can be chosen to be arbitrarily high, [88] explains the reasonable upper bound (e. g., at least one coefficient in  $V_J$  must remain unaffected by boundary distortions) set as the default value. Naturally, `K`, the number of sections to be refined, must be a positive integer as well and is bounded by  $N$ , too. For the refinement scale `Lambda` the case is similar, but it is bounded by the initial approximation scale `J`.

### Advanced Usage

Further optional input and output parameters are implemented and serve for a better understanding of LLSA's procedure, that is, in order to avoid a *black box* impression.

The following additional parameters are specified and can be accessed as follows.<sup>4</sup> The input parameters are now given by

```
(data, wavelet, J, Lambda_vec, ind_sec, not_ind_sec).
```

While `data`, `wavelet`, and `J` remain as above, a generalization of the `zeros(K,1) + Lambda` parameter is now given through `Lambda_vec`. This is the parameter stated in the respective remark in Section 4.1.2, that is, instead of refining all sections up to the same scale `Lambda`, the  $k$ th section is now refined according to the  $k$ th entry in vector `Lambda_vec`. `Lambda_vec(k)` may either refer to the  $k$ th discovered section as in Equation (4.14) or the afterwards chronologically sorted regions. In this implementation the latter method is used. Note that the vector `Lambda_vec` already contains implicitly the information about the  $K$  sections to refine. Furthermore, the double column vectors `ind_sec` and `not_ind_sec` are stated. The former indicates sections where the LLSA algorithm is forced to determine the initial points for refinement  $l_{J,k}^W$ , while the latter prohibits the automatic detection (outside of `ind_sec`) for any of these in the specified intervals. For the output parameters there are

```
[SJt_llsa, SJt_modwt, fin_part, DJt_modwt, DJt_llsa, WJt_modwt, WJt_llsa].
```

Besides the above explained LLSA and MODWT approximation vectors, `fin-part` provides the final partition of LLSA's detected refineable regions  $\Omega_{j,k}^W$  for all  $J - \Lambda_k \leq j < J$ , that is, a matrix of dimensions  $(K, 2, \max(\text{Lambda\_vec}))$ . Additionally, to monitor which and how many details were reconstructed at each level, the original detail series and wavelet coefficient vectors, `DJt_modwt` and `WJt_modwt`, respectively, are provided, as well as the according adapted output from LLSA, `DJt_llsa` (for  $\tilde{D}_j$ ) and `WJt_llsa` (for  $\tilde{W}_j$ ).

A simple example for the input parameters can be given as follows.

```
...= llsa(data, 'd4', 7, [4 2], [450 550], [101 200; 301 400]);
```

This conducts a level 7 MODWT on the `data` with a  $D4$  wavelet. LLSA will refine the first section (ordered in time, independent whether it was the first or second to be

---

<sup>4</sup>The naming convention was chosen freely by the author and may be adapted in MATLAB according to the user's preferences.

discovered) by 4 scales, the second one by 2. The first section that is discovered is forced to be in the interval  $[450\ 550]$ , i. e.,  $l_{J,1}^W \in [450\ 550]$ , and  $l_{J,2}^W$  is excluded to be neither in  $[101\ 200]$  nor  $[301\ 400]$ .

## 4.2. Properties

In this section several properties of the LLSA algorithm are shown, namely its computational complexity (Section 4.2.1), the local linear filtering property (Section 4.2.2), and its impulse and step response (Section 4.2.3).

### 4.2.1. Computational Complexity

The computational complexity of the LLSA algorithm is proven in the following corollary.

**Corollary 4.3.** *The computational complexity of LLSA is given by  $\mathcal{O}(N \log_2 N)$ .*

*Proof.* As stated in [88] the computational complexity of the MODWT on which LLSA is built upon, is given by  $\mathcal{O}(N \log_2 N)$ . Determination of  $\alpha_{j,k}$  and  $\beta_{j,k}$  defined as in Theorem 4.4, can be done by at most  $N$  comparisons for every  $j = 1, \dots, J - \Lambda$  and  $k = 1, \dots, K$ . Furthermore, manipulation of the wavelet coefficients requires also at most  $N$  additional operations for each of all  $J$  levels. Thus, the computational complexity of  $\mathcal{O}(N \log_2 N)$  is preserved.  $\square$

This proves that the LLSA extension of the MODWT does not increase its computational complexity, which shows that in this respect there are no restrictions in LLSA's applicability, and thus, the algorithm satisfies Requirement R4.

### 4.2.2. Local Linearity

Though LLSA belongs to the class of nonlinear filters, it still retains the properties of a linear filter on certain subintervals, as the reconstruction of prior lost details affects only the extracted trend in the immediate proximity of the jump. Thus, outside of these regions, the output from LLSA coincides with the output of the linear MODWT

it extends, and therefore can be controlled a priori in terms of frequency passbands.

**Theorem 4.4.** *Be  $X$  a signal of length  $N$  to be filtered and  $S_J^{\text{LLSA}}$  the output generated by  $\text{LLSA}(J, K, \Lambda)$ . Then, the subintervals, which can be interpreted as the output of a linear filter, are given by*

$$S_J^{\text{LLSA}} \setminus \bigcup_{\substack{j=J-\Lambda, \dots, J \\ k=1, \dots, K}} \Omega_{j,k}^S, \quad (4.16)$$

with

$$\begin{aligned} \Omega_{j,k}^S := & [\max\{1, \alpha_{j,k} - L_j^{\text{wvlt}} + 1\}, \alpha_{j,k} + L_j^{\text{wvlt}}] \\ & \cup [\beta_{j,k} - L_j^{\text{wvlt}}, \min\{\beta_{j,k} + L_j^{\text{wvlt}} - 1, N\}]. \end{aligned} \quad (4.17)$$

*Proof.* The subintervals of  $S_J^{\text{LLSA}}$ , which can be interpreted as the output of a linear filter, are those sections affected only by either the wavelet coefficients retained or by either the ones set to zero. Thus, for every refined section  $k = 1, \dots, K$  and every level  $j = J - \Lambda, \dots, J$  the indices of  $\tilde{D}_j$  have to be determined that are affected by both coefficient types at the same time. For each jump, the first coefficients of  $\tilde{W}_j$  set to zero are given by  $\alpha_{j,k}$  and  $\beta_{j,k}$  on the left and right-hand side, respectively. With  $L_j^{\text{wvlt}}$  denoting the respective wavelet filter lengths, the indices of  $\tilde{D}_j$  affected by both retained and discarded wavelet coefficients are given by  $[\alpha_{j,k} - L_j^{\text{wvlt}} + 1, \alpha_{j,k} + L_j^{\text{wvlt}}]$  and  $[\beta_{j,k} - L_j^{\text{wvlt}}, \beta_{j,k} + L_j^{\text{wvlt}} - 1]$ . When including the natural boundaries  $[1, N]$ , this leads to Equation (4.17). For the reconstructed signal  $S_J^{\text{LLSA}}$  in Equation (4.10), mutually excluding all  $\Omega_{j,k}^S$  yields Equation (4.16).  $\square$

In practical applications one can relax Equation (4.17) as follows: Following the same argumentation as in Section 3.2 and brought forward by [88], it can be stated that the strict view of having filters with a width  $L_j^{\text{wvlt}}$  can be neglected, since generally the coefficients characterizing each wavelet are very small at the ends. Thus, the filters on each scale  $j$  have rather a width of  $2^j$ . This yields

$$\widehat{\Omega}_{j,k}^S := [\max\{1, \alpha_{j,k} - 2^j + 1\}, \alpha_{j,k} + 2^j] \cup [\beta_{j,k} - 2^j, \min\{\beta_{j,k} + 2^j - 1, N\}].$$

The author's observations for several benchmark plots support this interpretation, that is, the differences between  $S_J$  and  $S_J^{\text{LLSA}}$  for the regions

$$\bigcup_{\substack{j=J-\Lambda, \dots, J \\ k=1, \dots, K}} \Omega_{j,k}^S \setminus \bigcup_{\substack{j=J-\Lambda, \dots, J \\ k=1, \dots, K}} \widehat{\Omega}_{j,k}^S$$

are negligible. However,  $S_J^{\text{LLSA}}(t) = S_J(t)$  holds only for all  $t \notin \Omega_{j,k}^S$ .

Hence, the sections specified through Equation (4.16) can a priori be controlled in terms of frequencies as in the MODWT. However, note that one cannot make any statement about exact proportions of frequencies contained in  $\Omega_{j,k}^S$ . Yet, there still exists a lower bound of frequencies *not* contained in there. As any detail levels beyond  $J - \Lambda$  are not included, this bound can easily be derived by analyzing the transfer functions of the linear MODWT for scales  $1, \dots, J - \Lambda - 1$ . This is useful for practical applications, for which one might only be interested to ensure that certain high-frequencies (like daily fluctuations) are not contained in the trend in order not to compromise succeeding volatility analyses that aims exactly at estimating the structure in these frequency ranges. Thus, the above depicted property satisfies Requirement R2 at least partially.

### 4.2.3. Impulse and Step Response

With LLSA being a nonlinear filter, as in [9], one is interested in its characterizing impulse and the step response.<sup>5</sup> Note that this impulse response function is the same as its counterpart for linear filters (i. e., a characterization in time), as explained in Section 2.2, with the difference that for nonlinear filters one cannot state any filter weight sequence  $w_i$ . Since LLSA is a discrete filter and as only wavelets with compact support are used, it follows directly that LLSA must be an FIR filter (see Section 3.2). First, the filter's output is analyzed for the case when the input consists just of a single impulse or step. In the absence of any other characteristics in the signal this area will be selected automatically by LLSA for refinement. As the algorithm depends on the parameter triple  $(J, K, \Lambda)$ , one is primarily interested in what happens when  $K$  and  $\Lambda$  are varied, while  $J$  is kept fixed (changing the latter would only result in a finer or

---

<sup>5</sup>While a step is equivalent to a jump, this naming convention has become more common.

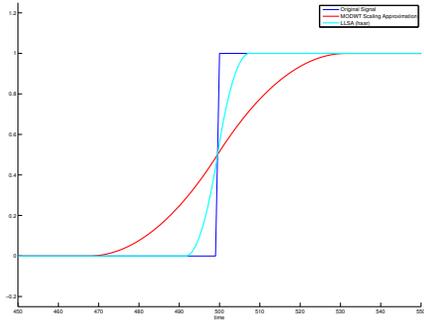
coarser resolution, but would not change the characteristic of the output).

For all wavelets, in order to restore a single step, it suffices to set  $K = 1$ , as it would be expected due to LLSA's reconstruction procedure depending on the step wavelet coefficient structure itself. The output then depends on the choice of  $\Lambda$ . For  $\Lambda = 0$  the output equals  $S_J$  since no reconstruction has taken place.  $\Lambda = J$  will provide the unfiltered signal  $X$  as, in this case, all details of the jump are reconstructed. For any other  $0 < \Lambda < J$  the output is in-between the two extreme cases. As there are no other details in the signal present, the step response of  $\text{LLSA}(J, 1, \Lambda)$  equals a MODWT scaling approximation  $S_{J-\Lambda}$ . The Haar and  $D4$  step response functions are depicted in Figures 4.2a and 4.2b, respectively.

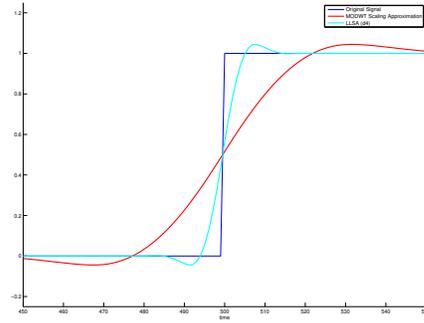
Contrary to the step response, the impulse response does not depend only on the wavelet but also on the choice of  $K$ . By analyzing the structure of the wavelet coefficients for a single impulse, one finds that, for example, when using the Haar wavelet one has to set  $K^{\text{Haar}} = 2$  in order to capture the whole impulse (see Figure 4.3b) as it is interpreted as two consecutive steps. Choosing  $K = 1$  results in the impulse only being partially refined, see Figure 4.3a. This is unlike, for example, the  $D4$  wavelet where it suffices to set  $K^{D4} = 1$  in order to fully reconstruct the impulse's details, and setting  $K = 2$  does not lead to any significant changes, see Figure 4.4. For other wavelets, one needs to analyze its respective impulse wavelet coefficient structure in order to determine  $K^{\text{wvlt}}$ . As above, the output depends on  $\Lambda$ . Similar to the step response, it can be concluded that the impulse response of  $\text{LLSA}(J, K^{\text{wvlt}}, \Lambda)$  equals a MODWT scaling approximation of level  $J - \Lambda$  for any wavelet and the proper choice of  $K^{\text{wvlt}}$ . This shows that the LLSA extension preserves the finite impulse response property of the MODWT (for wavelets with bounded support).

### 4.3. Summary

In this chapter the main contribution of this work was stated. The algorithm based on discrete wavelet transform was derived in Section 4.1.1. It was shown that this approach enables the local refinement, that is, the reconstruction of during the linear

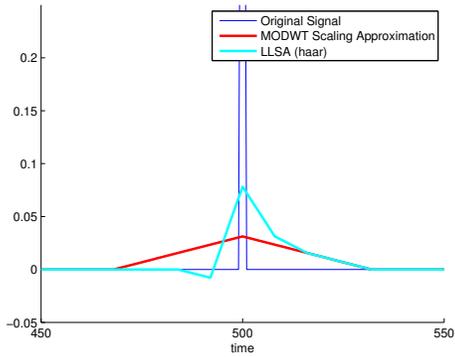


(a) Haar

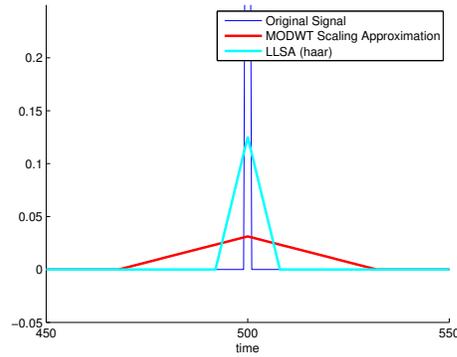


(b)  $D_4$

Figure 4.2.: LLSA(Haar &  $D_4$ ) step response functions,  $K = 1$

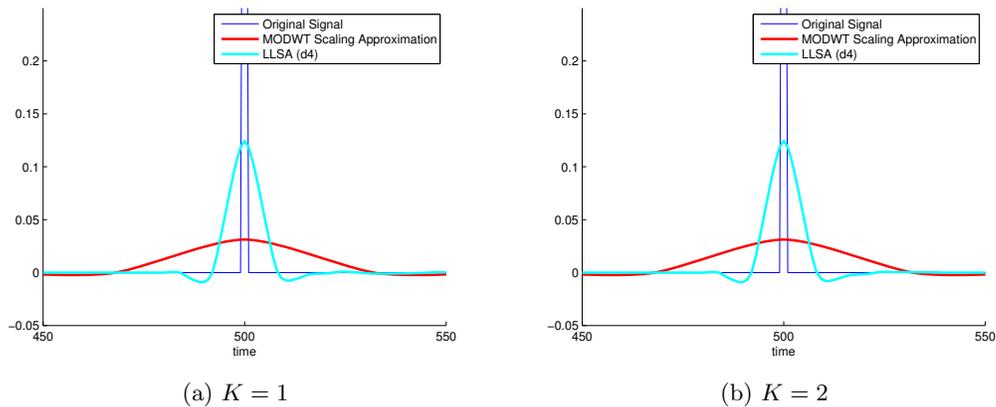


(a)  $K = 1$



(b)  $K = 2$

Figure 4.3.: LLSA(Haar) impulse response with different  $K$

Figure 4.4.: LLSA( $D4$ ) impulse response with different  $K$ 

filtering process blurred out details, while leaving the rest of the filtered signal mainly untouched. The final mathematical formulation and complementary remarks were given in Section 4.1.2, while in Section 4.1.3 LLSA's usage for its MATLAB implementation was reported, including several extensions to allow for more flexible possible uses.

The novelty of the proposed algorithm lies in its ability to transfer the wavelet coefficient step response onto the signal from which the trend is to be extracted. All technical details of the algorithm's stages were thoroughly discussed and enable a direct implementation. Requirements R1 to R3 had a direct impact on how the algorithm was designed, while Requirement R4 was shown to be fulfilled as well. LLSA extends the established MODWT, requiring only minimal additional input parameters that can also be set in a heuristic manner. The algorithm preserves all of the MODWT favorable characteristics (i. e., a smooth trend and a priori frequency control) in the areas where no sudden changes occur. However, in the near proximity of these phenomena, lower bounds for frequency control can still be provided, and the lost details are reconstructed according to the degree specified by the user. To the author's best knowledge no other algorithm exists that conjointly fulfills these requirements as LLSA does. Requirement R5 will be discussed in Section 5.1.

## Chapter 5.

# Evaluation and Application

In this chapter the algorithm that was proposed in Chapter 4 is evaluated with respect to its consistency and its application possibilities. Proving the algorithm's consistency means to show its robustness in the sense that output results provided by LLSA are reliable. Therefore, in Section 5.1 it will be shown that the trend estimation error of LLSA is bounded and that its application on simulated and empirical data can lead to more accurate results, that is, in a considerable number of scenarios and use cases the long-term trend derived by LLSA follows the real trend more closely than the alternative benchmarks. The implications on the eventual applications are discussed in Section 5.2. After pointing out two real data examples, which show the algorithm's behavior, two concrete applications are highlighted, namely, price volatility estimation and value at risk.

### 5.1. Robustness and Performance Studies

In this section the robustness and the general performance (in respect to other benchmark algorithms) of the LLSA algorithm is investigated. It is shown that independent of the choice of wavelets and the  $(J, K, \Lambda)$  LLSA parameters, the results of this new approach are consistent, that is, the errors from the optimal solution are bounded and converge asymptotically. The purpose of this section is to show that Requirement R5 holds.

After an analytical proof of LLSA's bounded estimation error and asymptotical convergence in Section 5.1.1, a robustness and performance study on synthetic (i. e., artificially constructed) signals is conducted in Section 5.1.2. There, LLSA's performance is measured in relation to two benchmark filtering techniques of which each one serves as a representative for the (non)linear filtering class. This is followed by an empirical robustness study using real data in Section 5.1.3. While for the simulation study the optimal solution is known and the error can easily be calculated using certain error measures, this does not hold for the empirical (stock price) data. In this case, in order to estimate how closely the unknown trend is followed by LLSA and the other benchmarks' output, four distance measures that relate the estimated trend to the empirical distribution are used.

### 5.1.1. Analytical Consistency

First an analytical proof is given that LLSA's trend estimation error is bounded and that the filtered output converges asymptotically towards the MODWT.

**Theorem 5.1.** *Be given any signal  $X$  of length  $N$  with the trend  $\vartheta(X)$ . Then, the error of LLSA's estimated  $S_J^{\text{LLSA}}$  is bounded, that is, for any choice of  $(J, K, \Lambda)$  there exists a constant  $A > 0$  so that*

$$\sum_{t=1}^N |\vartheta_t(X) - S_{J,t}^{\text{LLSA}}| < A \quad (5.1)$$

*holds. Furthermore, for  $K$  fixed and  $N \rightarrow \infty$  holds*

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N |S_{J,t}^{\text{MODWT}} - S_{J,t}^{\text{LLSA}}| = 0, \quad (5.2)$$

*i. e., LLSA converges asymptotically towards the estimated trend of the MODWT.*

*Proof.* Let  $J$  be fixed. Considering  $0 \leq \Lambda \leq J$  and  $0 \leq K \leq N^1$ , by setting either  $\Lambda = 0$  or  $K = 0$  this yields  $S_J^{\text{LLSA}} = S_J^{\text{MODWT}}$ . For any other choice of  $K$  and  $\Lambda$  more details will be added to the estimator  $S_J^{\text{LLSA}}$ , according to  $\Omega_{j,k}^W$ . After the reconstruction

---

<sup>1</sup>This upper bound is given naturally, as there cannot be more critical sections than signal data points.

(see Equation (4.10)) these additional details correspond to the estimator  $S_{J-\Lambda}^{\text{MODWT}}$  and are therefore bounded as well. Unfortunately, as during the refinement process several wavelet coefficients are set to zero, this match only holds approximately, i. e., no exact description of  $S_{J,t}^{\text{LLSA}}$  is available, since this procedure causes the information contained in the wavelet coefficients on different levels to be intermixed. However, knowing that the MODWT approximation of any level is bounded by the signal itself, one can set up an  $\epsilon$ -tube around the initially estimated trend by

$$\epsilon := \left| \max_t X_t - \min_t X_t \right| \quad (5.3)$$

As

$$\min_t X_t \leq S_{J,t}^{\text{MODWT}} \leq \max_t X_t$$

holds for all  $1 \leq j \leq J$ , and therefore

$$|\vartheta_t(X) - S_{J,t}^{\text{MODWT}}| \leq \epsilon,$$

this must also hold for LLSA, i. e.,

$$\min_t X_t \leq S_{J,t}^{\text{LLSA}} \leq \max_t X_t \quad \text{and} \quad |\vartheta_t(X) - S_{J,t}^{\text{LLSA}}| \leq \epsilon.$$

Thus, one can estimate an upper bound for the error by  $A = N \cdot \epsilon$ , which suffices Equation (5.1). To prove the asymptotic consistency it can be assumed that for a fixed  $K$  the  $\Omega_{j,k}^W$  are ordered in time, that is

$$\alpha_{j,k+1} \geq \beta_{j,k}$$

for all  $1 \leq j \leq J$  and  $1 \leq k \leq K - 1$ . Hence, after

$$t_K := \max_{t,j} \{t \in \Omega_{j,K}^S\}$$

$S_{J,t}^{\text{LLSA}} = S_{J,t}^{\text{MODWT}}$  holds for all  $t > t_K$ . As the prior deviations between both estimators

were bounded, e. g., by the same  $\epsilon$  as in Equation (5.3), for all  $N$

$$\sum_{t=1}^N |S_{J,t}^{\text{MODWT}} - S_{J,t}^{\text{LLSA}}| = \sum_{t=1}^{t_K} |S_{J,t}^{\text{MODWT}} - S_{J,t}^{\text{LLSA}}| \leq t_K \cdot \epsilon,$$

holds, which proves Equation (5.2). □

The bounds stated in Equation (5.3) are very generous, that is, they consider the most extreme case. Due to the moving average nature of LLSA and the MODWT the bound  $A$  will never be reached (disregarding trivial, e. g., constant, signals). However, without any further specific assumptions or information about the signal, its trend and noise components, no smaller bound can be stated. As bounds for the error and its asymptotic consistency given in Theorem 5.1 do not provide any further insights about LLSA's behavior in practice, further investigation is required. This is the subject of the following Sections 5.1.2 and 5.1.3, where further robustness analyses are conducted, using artificial signals and empirical data, respectively.

### 5.1.2. Simulated Signals

In order to test for the robustness of the LLSA algorithm, a signal of length  $2^{10}$  is set up. Though in this thesis no specific trend and signal patterns are analyzed separately, the setup of the simulation is carried out analogously to [5, 42]. Jump occurrences in this signal were uniformly distributed (thus, coinciding with a Poisson arrival rate being observed in many systems), with their height being a random number drawn from a normal distribution with mean 0 and variance 1. The signal is constant between the jumps, which should render a very smooth trend in the output. As usual in this kind of simulations, white Gaussian noise was added afterwards. Note, as this synthetic signal comprises pure jumps only, it should be optimal for the median filter, and a very hard case for moving average filters. The simulation was run with different setups, varying the amount of jumps  $\tilde{K}$ , the noise variance  $\sigma$ , and the wavelets. Each setup was run  $2^9$  times, with  $m$  different time series generated for the  $m$ th run, yielding 131,328 different time series generated in total. This amount has proven to be sufficient to get clear results. For each run the mean squared and the mean absolute error were aggregated over all  $m$

outcomes and compared to the errors of the linear MODWT and the nonlinear median filter that were applied onto the same series. The advantage of using such a simulation setting is that the optimal solution is known, hence the mean squared error (MSE) and the mean absolute error (MAE) for every single run can be measured.

**Remark 5.2.** Considering the rich amount of available linear and nonlinear filters it might certainly be questioned whether these filters are the best benchmarks. It can be stated almost surely that no filter is generally the best choice for all time series at the same time. That is, for example, a trigonometric filter will almost surely in most cases provide the best results and outperform any other filter, if the underlying signal is of a sine form. Similar holds for linear regression methods when the trend is strictly linear. Choosing more advanced filters will introduce the bias of choosing weights, thresholds, or other parameters. Thus, this study is restricted to two benchmark filters, which can be seen as the most basic version for the respective filter class. The median filter thus represents the class of non-linear filters, while the MODWT acts as the prototype for the class of linear filters. Though the very first prototype for this class could be seen as the mean filter, the MODWT was chosen, since it is extended by LLSA and also have proven to provide a very good performance in several application areas (see, for example, [30, 48, 51, 84, 87, 119]).<sup>2</sup>

For all three filters the same bandwidth  $2^J = 2^7$  was selected, as providing a smooth trend for all noise variances stated below. First, it is assumed that the amount of jumps is known to LLSA, i. e.,  $K = \tilde{K}$ . Further,  $\Lambda = 3$  was chosen independently of the noise variance. Note that these parameter choices are certainly not optimal for all different setups, but nevertheless allow the comparison of the robustness and performance of the algorithm to the alternative filters without giving any specific data-dependent input other than  $K$ .

Though the choice of the wavelet is not as important for the MODWT as for the DWT (see [88]), the robustness of LLSA is tested with the above introduced wavelets (i. e.,

---

<sup>2</sup>One additional reason the mean filter was not chosen is that fluctuations with high amplitudes still lead to disturbances in the mean filtered trend, where the MODWT performs better, see Examples 2.4 and 5.3

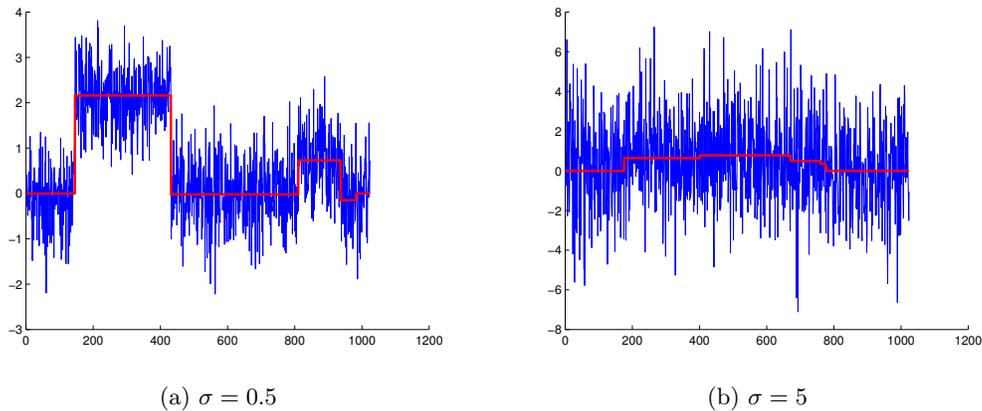


Figure 5.1.: Robustness test signal with different noise levels

*Haar*, *D4* and *LA8*) being the most utilized ones in practice. The wavelet dependent parameters necessary for LLSA are stated in Table 4.1. It was set  $\sigma \in [0.01, 0.5, 1, 2, 5]$  as the input set for the noise variance. While 0.01 signifies that the signal is almost noise free, the jumps and the noise share the same distribution for variance 1. Given a noise variance of 5 or higher, the jumps are hardly discernible any more (see Figure 5.1b), i. e., the signal is completely dominated by the noise. The amount of jumps  $\tilde{K}$  was fixed for all 512 simulation runs and selected from the set  $[5, 10]$ . These are reasonable parameter sets for the algorithm, as they imply that the constructed signals exhibit smooth trends with occasional jumps and slopes, that is, the simulation represents the problem class depicted in the Introduction and Chapter 2.

In Tables 5.1 and 5.2, the mean of the MSE, MAE, and their corresponding sample variances (in brackets below the error) over all 512 runs is reported. Although not scale-invariant, these two error measures are common to investigate and compare the performance of different (non)linear filters (see [9]) in case the optimal solution is known. Since the simulated time series are all on the same scale, these two measures are sufficient. For a further discussion about alternative error measures, their pitfalls and applicability, please refer to [27, 59].

It can be observed that for all simulations after at certain number of runs, the measured errors of the different algorithms are strictly higher (lower, respectively) than the others (exemplarily depicted in Figures 5.2 and 5.3). Furthermore, the errors of all algorithms increase with a higher noise variance. Independent of the choice of the wavelet, for  $\sigma \in \{0.01, 0.5, 1, 2\}$  LLSA always outperforms the MODWT (utilizing the same wavelet). This does not hold for  $\sigma = 5$ , for which the error of LLSA is larger. Analyzing several output results, this leads to the conclusion that in case of such high noise levels, LLSA tends to restore details in areas where actually no jump has occurred. In this way, LLSA reconstructs jumps that deviate from the ones of the original signal, thus leading to a higher error in contrast to the MODWT, which simply ignores those jumps caused by noise. Also, for a very low noise level  $\sigma = 0.01$ , the median filter performs better (in respect to the MAE, but not the MSE) than both  $\text{LLSA}_{(D4)}$  and  $\text{LLSA}_{(LA8)}$ . Almost the same holds for very high noise  $\sigma = 5$  for which their MSE/MAE are higher than the ones of the median filter. In this case,  $\text{LLSA}_{(LA8)}$  slightly outperforms  $\text{LLSA}_{(D4)}$ . This is surprising, as in all other cases the contrary, that is,  $\text{LLSA}_{(LA8)}$  is outperformed by  $\text{LLSA}_{(D4)}$ , can be observed. In all other cases, LLSA yielded lower errors, independent of the wavelet utilized.

One finds that the variances of the errors of LLSA are lower than the ones of the other algorithms, with the exception of the *LA8* wavelet for  $\sigma = 1$ . Also, between the different wavelets in LLSA, the Haar wavelets yielded the lowest variance of the error, followed by the *D4* wavelet. Again, an exception can be seen for  $\sigma = 5$ . Also note that the convergence of the error variances is much faster for the LLSA algorithm than for the alternatives. Figures 5.2 and 5.3 depict exemplarily some plots of both the errors and the variances.<sup>3</sup> The results for a higher number of jumps  $\tilde{K} = 10$  are similar and lead to the same conclusions.

As pointed out above, a high noise level dominating the signal and the inherent jumps can yield a higher error for LLSA. A similar case may happen if the number of expected jumps  $K$  exceeds (or simply deviates from) the number of jumps  $\tilde{K}$  that ac-

---

<sup>3</sup>Plots from other configuration settings are completely analog, which is why they are omitted here. They are available upon request.

Table 5.1.: MSE mean and variance, 5 jumps

	$\sigma = 0.01$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$
LLSA <sub>(Haar)</sub>	0.0142 ( $2.39 \cdot 10^{-6}$ )	0.0239 ( $4.17 \cdot 10^{-6}$ )	0.0333 ( $4.48 \cdot 10^{-6}$ )	0.0521 ( $7.78 \cdot 10^{-6}$ )	0.1089 ( $1.41 \cdot 10^{-4}$ )
MODWT <sub>(Haar)</sub>	0.0974 ( $1.54 \cdot 10^{-4}$ )	0.1003 ( $1.85 \cdot 10^{-4}$ )	0.1033 ( $2.82 \cdot 10^{-4}$ )	0.1082 ( $3.32 \cdot 10^{-4}$ )	0.1245 (0.0013)
LLSA <sub>(D4)</sub>	0.0177 ( $4.61 \cdot 10^{-6}$ )	0.0316 ( $1.99 \cdot 10^{-5}$ )	0.0459 ( $2.08 \cdot 10^{-5}$ )	0.0746 ( $1.08 \cdot 10^{-5}$ )	0.1611 ( $4.95 \cdot 10^{-5}$ )
MODWT <sub>(D4)</sub>	0.0922 ( $2.58 \cdot 10^{-4}$ )	0.0949 ( $4.93 \cdot 10^{-4}$ )	0.0982 ( $2.50 \cdot 10^{-4}$ )	0.1045 ( $1.54 \cdot 10^{-4}$ )	0.1232 ( $1.56 \cdot 10^{-4}$ )
LLSA <sub>(LAS)</sub>	0.0321 ( $3.35 \cdot 10^{-5}$ )	0.0443 ( $4.27 \cdot 10^{-5}$ )	0.0561 ( $1.16 \cdot 10^{-4}$ )	0.0789 ( $4.48 \cdot 10^{-5}$ )	0.1474 ( $1.92 \cdot 10^{-4}$ )
MODWT <sub>(LAS)</sub>	0.0923 ( $1.55 \cdot 10^{-4}$ )	0.0957 ( $1.55 \cdot 10^{-4}$ )	0.0991 ( $4.14 \cdot 10^{-4}$ )	0.1062 ( $7.23 \cdot 10^{-4}$ )	0.1259 ( $2.93 \cdot 10^{-4}$ )
Median filter	0.0270 ( $1.94 \cdot 10^{-4}$ )	0.0669 ( $7.51 \cdot 10^{-5}$ )	0.0819 ( $9.79 \cdot 10^{-5}$ )	0.0997 ( $1.22 \cdot 10^{-4}$ )	0.1411 (0.0012)

Table 5.2.: MAE mean and variance, 5 jumps

	$\sigma = 0.01$	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 5$
LLSA <sub>(Haar)</sub>	0.0380 ( $4.85 \cdot 10^{-6}$ )	0.0924 ( $6.06 \cdot 10^{-6}$ )	0.1202 ( $1.47 \cdot 10^{-5}$ )	0.1605 ( $2.13 \cdot 10^{-5}$ )	0.2429 ( $1.07 \cdot 10^{-4}$ )
MODWT <sub>(Haar)</sub>	0.1710 ( $1.42 \cdot 10^{-4}$ )	0.1866 ( $2.85 \cdot 10^{-4}$ )	0.1956 ( $1.95 \cdot 10^{-4}$ )	0.2094 ( $2.33 \cdot 10^{-4}$ )	0.2414 (0.0010)
LLSA <sub>(D4)</sub>	0.0512 ( $1.01 \cdot 10^{-5}$ )	0.1145 ( $3.17 \cdot 10^{-5}$ )	0.1486 ( $4.20 \cdot 10^{-5}$ )	0.1980 ( $2.11 \cdot 10^{-5}$ )	0.2979 ( $5.19 \cdot 10^{-5}$ )
MODWT <sub>(D4)</sub>	0.1676 ( $2.62 \cdot 10^{-4}$ )	0.1814 ( $4.20 \cdot 10^{-4}$ )	0.1918 ( $2.67 \cdot 10^{-4}$ )	0.2075 ( $1.09 \cdot 10^{-4}$ )	0.2430 ( $1.28 \cdot 10^{-4}$ )
LLSA <sub>(LAS)</sub>	0.0821 ( $5.68 \cdot 10^{-5}$ )	0.1300 ( $4.28 \cdot 10^{-5}$ )	0.1576 ( $1.18 \cdot 10^{-4}$ )	0.1976 ( $4.15 \cdot 10^{-5}$ )	0.2804 ( $2.5 \cdot 10^{-4}$ )
MODWT <sub>(LAS)</sub>	0.1728 ( $1.47 \cdot 10^{-4}$ )	0.1859 ( $1.23 \cdot 10^{-4}$ )	0.1960 ( $4.60 \cdot 10^{-4}$ )	0.2120 ( $3.71 \cdot 10^{-4}$ )	0.2481 ( $2.15 \cdot 10^{-4}$ )
Median filter	0.0495 ( $3.51 \cdot 10^{-5}$ )	0.1521 ( $1.15 \cdot 10^{-4}$ )	0.1794 ( $6.76 \cdot 10^{-5}$ )	0.2119 ( $1.17 \cdot 10^{-4}$ )	0.2734 ( $7.94 \cdot 10^{-4}$ )

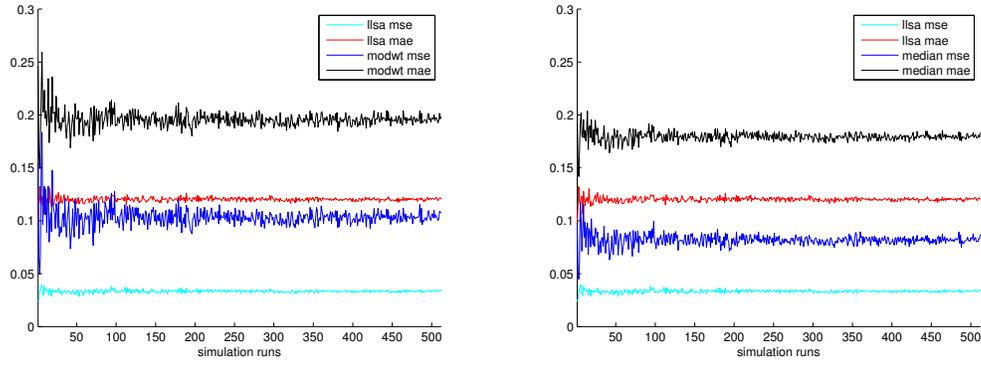


Figure 5.2.: MAE/MSE for the Haar wavelet with 5 jumps and  $\sigma = 1$ .

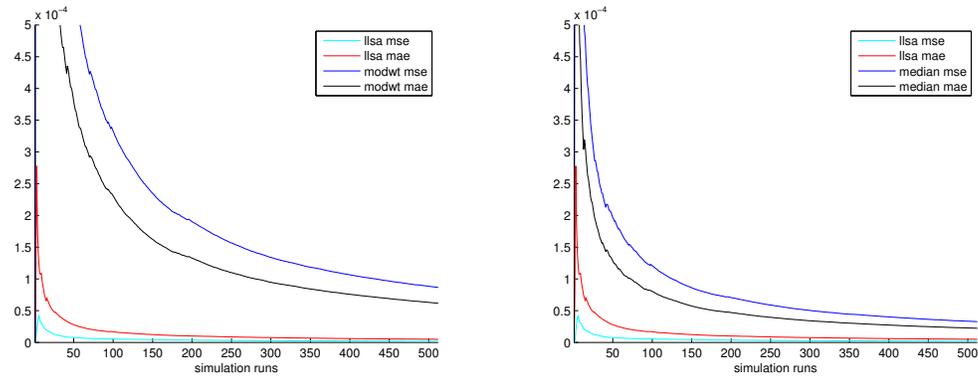


Figure 5.3.: MAE/MSE variances for the Haar wavelet with 5 jumps and  $\sigma = 1$ .

Table 5.3.: MSE mean and variance, 10 jumps,  $\sigma = 1$ 

	$p = 0.3$	$p = 0.5$	$p = 1$
LLSA <sub>(Haar)</sub>	0.0606 ( $1.84 \cdot 10^{-5}$ )	0.0607 ( $3.40 \cdot 10^{-5}$ )	0.1128 (0.0032)
MODWT <sub>(Haar)</sub>	0.2133 ( $5.10 \cdot 10^{-4}$ )	0.2139 ( $7.09 \cdot 10^{-4}$ )	0.2146 ( $5.30 \cdot 10^{-4}$ )
LLSA <sub>(D4)</sub>	0.0829 ( $7.82 \cdot 10^{-5}$ )	0.0712 ( $2.76 \cdot 10^{-5}$ )	0.1318 ( $8.76 \cdot 10^{-4}$ )
MODWT <sub>(D4)</sub>	0.2055 ( $6.94 \cdot 10^{-4}$ )	0.2056 ( $5.16 \cdot 10^{-4}$ )	0.2051 ( $6.06 \cdot 10^{-4}$ )
Median filter	0.1706 ( $3.37 \cdot 10^{-4}$ )	0.1705 ( $4.14 \cdot 10^{-4}$ )	0.1710 ( $3.56 \cdot 10^{-4}$ )

tually occurred. Therefore, additional simulations are run in which  $K$  was allowed to vary, i. e.,  $K$  was chosen to be a uniformly distributed random integer from the interval  $[(1-p)\tilde{K}, (1+p)\tilde{K}]$  with  $p$  the percentaged deviation. For example, for  $p = 1$ , the interval ranges from 0 (thus, being equivalent to the MODWT) up to twice as much as  $\tilde{K}$  being restored. The results are summarized in Tables 5.3 and 5.4. For  $p = 0.3$  and  $p = 0.5$ , it can be observed that the algorithm still performs better than the alternatives, in respect to errors as well as the variance. For the very large deviation  $p = 1$  one finds that in respect to mean errors LLSA still performs better, but this time has a higher variance.

The conclusion that can be drawn is that the LLSA algorithm is stable (i. e., its results are reliable in the way that for a growing number of signals the errors converge to a certain value and the error variances diminish) in every case, though it may be outperformed by alternative algorithms for certain choices of wavelets and extreme (i. e., very high and low) noise levels. Additionally, LLSA remains stable and performs better, even if  $K$  deviates from the actual number of jumps in the signal. It seems that in practical applications, the Haar and the  $D4$  wavelets are the best choice for LLSA.

The above described simulations only comprise signals with either pure jumps or spikes

Table 5.4.: MAE mean and variance, 10 jumps,  $\sigma = 1$ 

	$p = 0.3$	$p = 0.5$	$p = 1$
LLSA <sub>(Haar)</sub>	0.1644 ( $2.35 \cdot 10^{-5}$ )	0.1639 ( $4.91 \cdot 10^{-5}$ )	0.2030 (0.0013)
MODWT <sub>(Haar)</sub>	0.3098 ( $2.58 \cdot 10^{-4}$ )	0.3100 ( $4.48 \cdot 10^{-4}$ )	0.3104 ( $3.55 \cdot 10^{-4}$ )
LLSA <sub>(D4)</sub>	0.1921 ( $7.37 \cdot 10^{-5}$ )	0.1805 ( $3.19 \cdot 10^{-5}$ )	0.2255 ( $4.65 \cdot 10^{-4}$ )
MODWT <sub>(D4)</sub>	0.3017 ( $4.30 \cdot 10^{-4}$ )	0.3016 ( $2.98 \cdot 10^{-4}$ )	0.3014 ( $4.29 \cdot 10^{-4}$ )
Median filter	0.2673 ( $1.69 \cdot 10^{-4}$ )	0.2670 ( $1.97 \cdot 10^{-4}$ )	0.2673 ( $1.72 \cdot 10^{-4}$ )

(in case of two consecutive jumps in different directions). This is due to the fact that steep slopes and valleys are difficult to simulate without predefining a parametric function that represents these features. In order to be able to analyze the LLSA algorithm's performance in respect to these phenomena an empirical study is conducted, which is reported in the next section.

### 5.1.3. Empirical Results

In the empirical study, the performance of the local linear scaling approach with respect to the robustness of the algorithm is investigated. The analysis is performed for the trend extraction of the German DAX stock prices<sup>4</sup> based on four goodness of fit criteria: the Kolmogorov-Smirnov distance, the Anderson-Darling (AD) distance, the Kuiper (K) distance, and the Cramér-von Mises (CVM) distance. In this empirical study, LLSA is tested for homogeneous (i. e., equally spaced) high-frequency time series data aggregated by the linear interpolation method at the sampling frequency of 60 minutes. By applying the algorithm on several moving window subsamples, its robustness is shown and its performance is analyzed by taking into account several statistical measures. Also, statistical significance tests based on bootstrapping are conducted to further emphasize

<sup>4</sup>Obtained from the German Karlsruher Kapitalmarktdatabank (KKMDB).

the above claim.

### The Data

The algorithm is performed for trend extraction with the German DAX stock prices data for the whole year of 2008.<sup>5</sup> The raw data is inhomogeneous, i. e., irregularly spaced. In this thesis, linear interpolation was chosen as a regularizing operation to aggregate the raw data to homogeneous data. The inhomogeneous series with times  $t_i$  is given by  $x(t_i)$ . The target homogeneous time series  $\tilde{x}$  shall be defined at times  $\tau_j := t_0 + j\Delta t$ ,  $j \in \mathbb{N}$ , with  $\Delta t > 0$  fixed. As every regular  $\tau_j$  is bounded by two times of the irregularly spaced series, i. e.,

$$t_{I_j} \leq \tau_j < t_{I_j+1},$$

with

$$I_j := \max\{i \mid t_i \leq \tau_j\},$$

data point  $\tau_j$  is interpolated between  $t_{I_j}$  and  $t_{I_j+1}$  by

$$\tilde{x}(\tau_j) = x_{I_j} + \frac{\tau_j - t_{I_j}}{t_{I_j+1} - t_{I_j}} (x_{I_j+1} - x_{I_j}).$$

### Goodness-of-fit tests

As the real trend for the empirical data series  $x$  is unknown, one cannot measure the MSE/MAE as was done in Section 5.1.2. Instead, this analysis follows [108] and uses the minimum distance estimation approach with the Kolmogorov-Smirnov (KS), Anderson-Darling (AD), Kuiper (K), and the Cramér-von Mises (CVM) distances as estimators and a criterion for the goodness-of-fit testing between the distribution of the original

---

<sup>5</sup>As the EON data was incomplete, it was replaced with the EPCOS stock price data instead

data and the estimated trend. These distances are defined as follows:

$$KS := \sup_{x \in \mathbb{R}} |F_n(x) - F(x)|,$$

$$AD := \sup_{x \in \mathbb{R}} \frac{|F_n(x) - F(x)|}{\sqrt{F(x)(1 - F(x))}},$$

$$K := \sup_{x \in \mathbb{R}} (F_n(x) - F(x)) + \sup_{x \in \mathbb{R}} (F(x) - F_n(x)), \text{ and}$$

$$CVM := \int_{-\infty}^{\infty} (F_n(x) - F(x))^2 dF(x).$$

$F_n(x)$  denotes the empirical sample distribution of  $x$  and  $F(x)$  is the distribution function of the estimated trend. Thus, the smaller the distance, the better the estimated trend on scale  $J$  preserves the distribution of  $x$ .

As both the KS and the K distance focus on deviations around the median of the distribution, they tend to be more sensitive near the center of the distribution than at the tails. On the contrary, the AD distance puts more weight on discrepancies in the tails (see [108] for more details), while CVM measures the sensitivity of dispersion between the empirical data and its trend in respect to the changing LLSA trend estimator. By including these different distances in this robustness study, the reliability of the results will increase.

### **The Methodology**

In this first study, the performance of LLSA is compared against the MODWT. The median filter is not considered, as the filtered output signal for the same scale still contained too many ripples and showed a larger number of fluctuations than to be considered as an appropriate long-term trend. Though the choice of wavelet does not play a major role in the MODWT, yielding always similar smooth curves, significant differences were noted in above simulated robustness analysis for LLSA. Hence, one

wavelet was fixed for the empirical study. The Haar wavelet was chosen due to the following reasons: First, based on the results of the robustness study in 5.1.2, regarding the MSE/MAE as well as the corresponding variances, in most (i. e., non-extreme) cases it outperformed the other tested wavelets. Second, as the Haar wavelet is the one with the smallest support, it also yields the smallest regions around the jumps, which must be considered to be the output from a nonlinear filter also introducing less ripples in the whole trend. Additionally, this minimizes the boundary distortions. Furthermore, several plots (see also the impulse and step response function in Figures 4.2 and 4.3, respectively) indicated that the sections refined by the  $D4$  and  $LA8$  wavelets tend to exhibit the Gibbs phenomenon (see, for example, [61, 105]), especially when there is a real jump in the data and not just a slope.<sup>6</sup>

For the 60 minute data an approximation level  $J = 7$  was selected, as this provides a very smooth trend of the data without almost any ripples. This choice is based on subjective judgment and might have to be changed in practice due to the eventual goal of the analysis. The estimated trend is thus associated with a weighted average of bandwidth of  $2^7 \cdot 60$  minutes, that is, roughly five days. Furthermore, the refinement level was set to  $\Lambda = 2$ . This provides some improvement compared to the MODWT output, while not introducing too many ripples near any occurring jumps. In order to analyze whether any improvement can be expected even when LLSA is configured to provide marginal changes only, for every stock, the algorithm was set to detect one jump, i. e.,  $K = 1$ . This choice of  $(\Lambda, K)$  is certainly not optimal for all stocks at the same time and should be chosen separately dependent on each stock data for practical purposes.

### Empirical Robustness Results

In order to get valid results, the whole time series was filtered not only once, but a window of  $\kappa$  times the filter size was set up, i. e.,  $\kappa \cdot 2^J$ . Furthermore, to achieve a sufficient number of subsamples,  $\kappa$  was set heuristically to  $\kappa \in \{4, \dots, 10\}$ , which was advanced successively over each time series. The smallest value of  $\kappa$  provides a sample

---

<sup>6</sup>This can be explained by the fact that the  $D4$  and  $LA8$  wavelets are associated with higher smoothness spaces and thus, cannot adapt as fast to sudden jumps as, for example, the Haar wavelet.

large enough to contain a sufficient number of data points not affected by boundary distortions (see Section 5.2), while  $\kappa = 10$  ensures enough samples to derive significant results. This choice yields  $N - \kappa \cdot 2^J$  subsamples to validate the robustness of LLSA. In the following, the results for the extreme values  $\kappa = 4$  and  $\kappa = 10$  are analyzed in detail. These tables can be found in Appendix A.1. The tables for the remaining values of  $\kappa$  are in-between and available upon request.

It is observed that the mean as well as the median of LLSA outperforms the MODWT for all distances. Regarding the variance, note that especially for the KS, AD, and K distances around two thirds of the samples are higher for LLSA (and only 2 for CVM). Analyzing the respective data and their plots, this leads to the conclusion that these higher variances are caused by the distances for LLSA dropping to a very low level and changing back afterwards. Also, besides a few exceptions in the K and CVM distance, the minimum and maximum distance values are always lower for LLSA. This holds for all frequencies. By setting the moving windows size to  $\kappa = 10$  it is found that there is no longer a clear improvement of LLSA's performance over the MODWT for all stocks.

Finally, to further emphasize the robustness of the algorithm, since any specific assumptions about the distance measures (like being normally distributed) do not hold and, thus, following [109], further significance tests were conducted by bootstrapping (see [33] for further details about the methods and its applicability). Out of all moving window subsamples, 50.000 samples were drawn and the 99% (i. e.,  $\alpha = 0.01$ ) confidence intervals for the mean difference between LLSA and MODWT were calculated. Therefore, if for both the lower and the upper confidence bounds (LCB and UCB, respectively)  $LCB < 0$  and  $UCB < 0$  holds, there is a significant improvement of LLSA over the MODWT. For  $LCB > 0$  (and thus,  $UCB > 0$ ) the MODWT performs significantly better. In the case of  $LCB \leq 0$  and  $UCB \geq 0$  there is no statistically significant conclusion. The results for  $\kappa \in \{4, \dots, 10\}$  are reported in Tables A.9 to A.15. Analyzing the results of the significance analysis in Table A.15 one sees that the distances most affected are the KS and K distances, followed by one case for CVM. Again, for the AD distance it can be confirmed that LLSA always performs significantly better. However, one must also note a worse performance of LLSA in a number of stocks for the other distances. Besides

the fact that the previous results are shown to be significant in most cases, again it is noted that the highest improvements are listed for the AD and CVM distances. This is reasonable as the AD distance puts more weight on heavy-tailed distributions, while CVM mainly focuses on the fidelity of the estimated trend to the empirical time series around sudden changes in the trend. This is in accordance to the intuition that LLSA filters high-frequency noise while preserving jumps and slopes, which cause heavy-tails in the distribution of the data. Thus, in some particular cases the MODWT is still to be preferred. However, note that LLSA has proven to be significantly stable for all different settings of frequencies, stocks, and moving window sizes.

The above findings confirm that LLSA is not only robust, but also that (in the majority of cases) it outperforms the MODWT when applied on high-frequency data with regime changes (i. e., significant jumps besides the daily fluctuations) in the long-term trends. Note that in this study the input parameters ( $\Lambda$ ,  $K$ ) were not even calibrated separately for every stock data, as is recommended to do in practice (either by a priori or a posteriori analysis). In this case this certainly led to suboptimal results.

## 5.2. Applications

While in the previous Sections 5.1 and 5.1.3 the robustness and consistency of the algorithm itself was analyzed and its performance in relation to the benchmark filtering techniques was shown as a mere byproduct, this section focusses on the benefit of applying LLSA instead of the regular methods for time series' decomposition.

### 5.2.1. General Application and Examples

What can be expected from a more accurate decomposition of a time series into its components? Considering the process depicted in Section 2.1 one would expect that all or at least some of the detrending process succeeding steps improve in accuracy as well, or at least do not perform worse. This, however, cannot be guaranteed. Simple examples can be found (e. g., a step function with additive white Gaussian noise) where the noise distribution estimation succeeding the MODWT yield more accurate results than the ones following the LLSA procedure, though the MODWT delivers an overall

worse estimation of the trend than LLSA. This can be explained by the simple fact that errors in the trend estimation of the former method are symmetric, that is, they even out another in the estimation afterwards. Thus, a more accurate trend estimation does not necessarily lead to a better estimation of seasonalities or the noise distribution.

The application possibilities of LLSA are manifold. They are, however, restricted to homogenous (i. e., regularly spaced) time series, which must always be considered, and, if necessary, the data must be preprocessed. Specifically, considering economic and financial applications in general, a better trend estimation with accurately depicted jumps, steep slopes and valleys provides essential information about leads in any case to a better understanding of the time series long-term development itself. Accurately captured sudden changes help to identify their responsible external explanatory factors and influence of these factors, specifically in case of valleys and slopes where, contrary to jumps, it is initially not evident at which points they exactly begin and end. This information can also be provided by LLSA. On the other hand, an overall smoothness prevents the confusion of short-term influences in the long-term aspect. This depicts the direct advantageous usage of the trend extracted by LLSA. Positive secondary effects can then be expected with methods that use either this trend and/or any of the remaining components. These can be either analysis that are interested the specific components themselves or particular applications like value at risk.

The following example revises the in Section 2.3 given Wikipedia example and shows the improvements that can be achieved by LLSA and the aspects when using different wavelets.

**Example 5.3.** Example 2.4 is revised to analyze, whether one can achieve better results with the MODWT and its LLSA extension. Both methods are applied with the same bandwidth  $2^9$  as in the other example. In Figure 5.4 the effects for different choices of  $K$  are depicted, that is, the aftermost part of the valley around Christmas is partially refined, then the slope around summer holidays, and then the foremost part of the same valley. It can be seen that the last refinement interacts with the first one (due to the wavelets effective filter length, which exceeds the borders determined by LLSA, i. e., Equation (4.11)), thus, improving the overall shape of the whole section.

For the  $D4$  wavelet in Figures 5.5a and 5.5b one sees that one refinement section is sufficient to capture the whole valley, with the second one taking care of the slope, though the result contains significantly more ripples than with the Haar wavelet. The same holds also for the  $LA8$  wavelet, but it can be noticed on first glance that the reconstructed details already contain too many fluctuations that could be contributed to the ordinary trend (see Figure 5.6), which leads to the conclusion that the Haar and  $D4$  wavelets are to be preferred in practice.

In this example the direct benefit is given through the information provided by LLSA at which points the summer holiday and Christmas seasons affect the usage patterns. In addition to a more accurate trend curve, LLSA can directly provide the concrete locations of these regions by  $\Omega_{j,k}^S$ . Furthermore, a better estimation of the weekly and daily seasonalities, as well as the remaining noise can be expected.

The next example illustrates usage of LLSA on empirical financial data that will also be used for an application case in the next section.

**Example 5.4.** The trend of the SAP 60 minute frequency stock prize data is to be extracted. The bandwidth, as in Section 5.1.3 is chosen to be  $2^7$ , and the refinement scale  $\Lambda$  set to 2. In Figure 5.7 the results are depicted for the MODWT and the median filter. It can be seen that the MODWT delivers a slightly smoother trend, while the output of the median filter follows somewhat closer to the trend. The extreme jump between data points 1700 and 1800 is captured much more accurately by the latter method. In this example the median filter performs quite well, as the short-varying fluctuations have only relatively low amplitudes. Based on the results of Example 5.3 the  $LA8$  wavelet is not considered here and LLSA is restricted to usage of the Haar and  $D4$  wavelets.

Figure 5.8 shows that for  $K = 1$ , MODWT lost details are reconstructed so that it can be qualitatively compared to the output of the median filter, yet maintaining a smoother trend otherwise. Application with  $K = 2$  results in a better representation of an adjacent valley.

The results for the  $D4$  wavelet are depicted in Figure 5.9. For  $K = 1$  the reconstructed

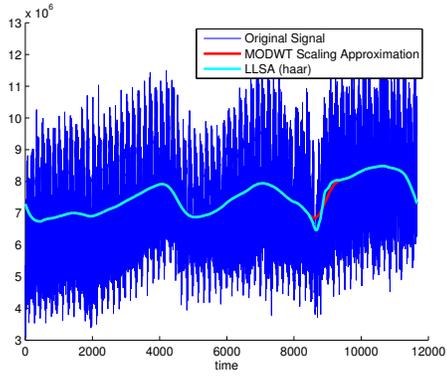
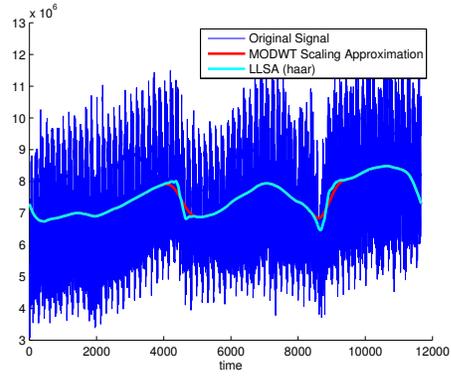
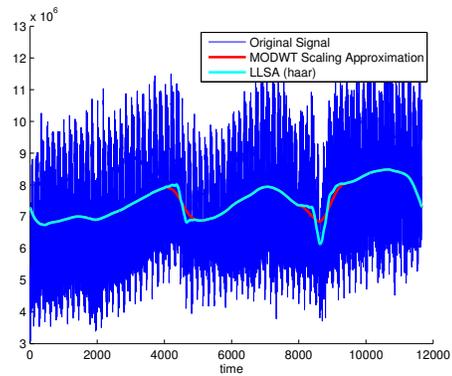
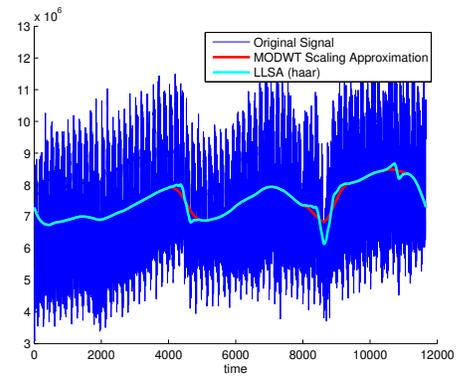
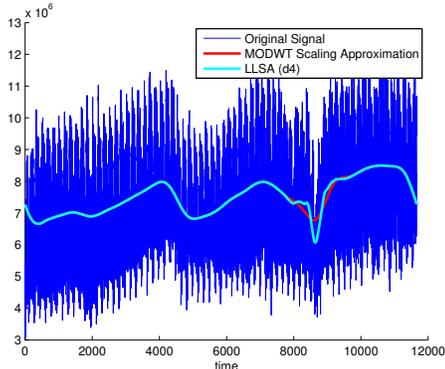
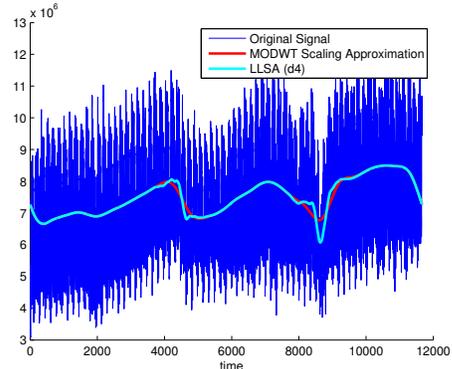
(a)  $K = 1$ (b)  $K = 2$ (c)  $K = 3$ (d)  $K = 4$ 

Figure 5.4.: Wikipedia refinement using the Haar wavelet

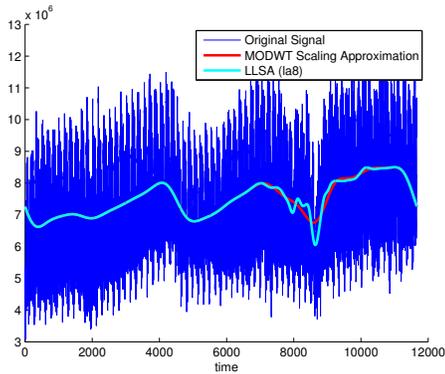


(a)  $K = 1$

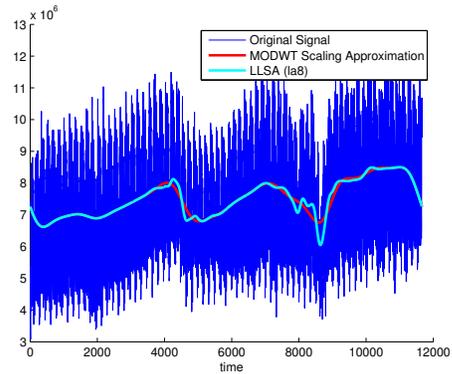


(b)  $K = 2$

Figure 5.5.: Wikipedia refinement with  $D4$  wavelets



(a)  $K = 1$



(b)  $K = 2$

Figure 5.6.: Wikipedia refinement with  $LA8$  wavelets

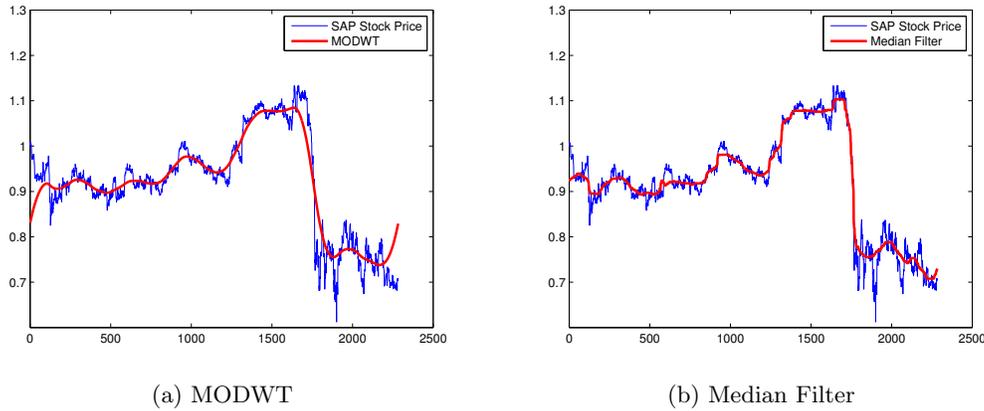


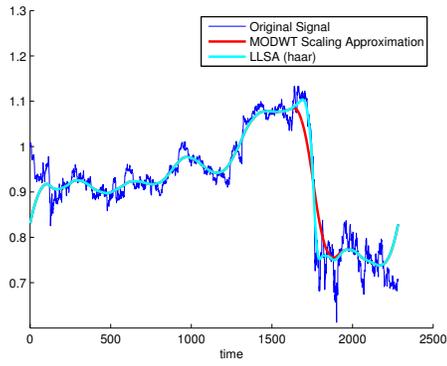
Figure 5.7.: Filtered trend of the SAP stock data

details of the filtered output are comparable to the one for the Haar wavelet with  $K = 2$ . Regarding the output for  $K = 2$  considerable more details are added.

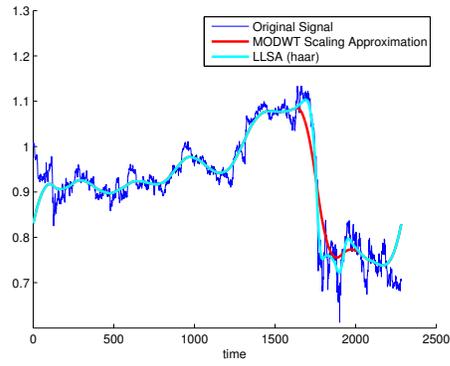
Setting  $K = 3$  for both cases the added details are not that significant, see Figure 5.10. It is noteworthy that the order in which the sections for reconstruction are selected by LLSA do not remain the same for the Haar and  $D4$  wavelet.

### A Note on Wavelets and Forecasting

Though the usefulness and applications exemplarily outlined in the next section will not be anticipated here, the author feels obliged to note what wavelet transforms are *not* appropriate to be used for. Since the advent of wavelets outside the pure mathematical research field, which can be mainly attributed to the pioneering work of Daubechies [32], wavelets have become a popular tool in most research areas that are connected to signal analysis in one form or another. However, considering specifically the general area of forecasting, in the literature there seem to be a discordance whether wavelets are the right tool for this. As outlined in Section 3.2 in order to handle the ends of the signal one has to extend it with the aim to minimize the boundary distortions. Therefore, by choosing this method, an additional bias is involuntarily introduced even before the

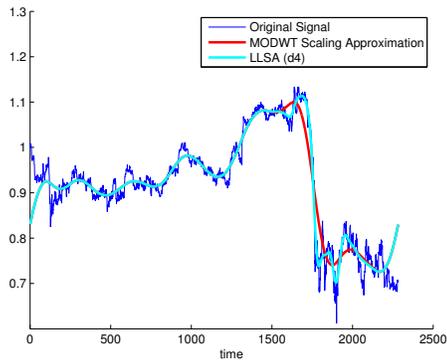


(a)  $K = 1$

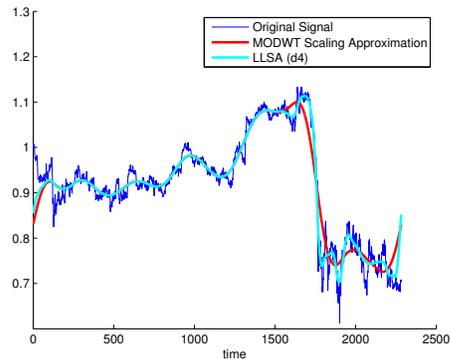


(b)  $K = 2$

Figure 5.8.: LLSA(Haar) filtered trend of the SAP stock data



(a)  $K = 1$



(b)  $K = 2$

Figure 5.9.: LLSA(D4) filtered trend of the SAP stock data

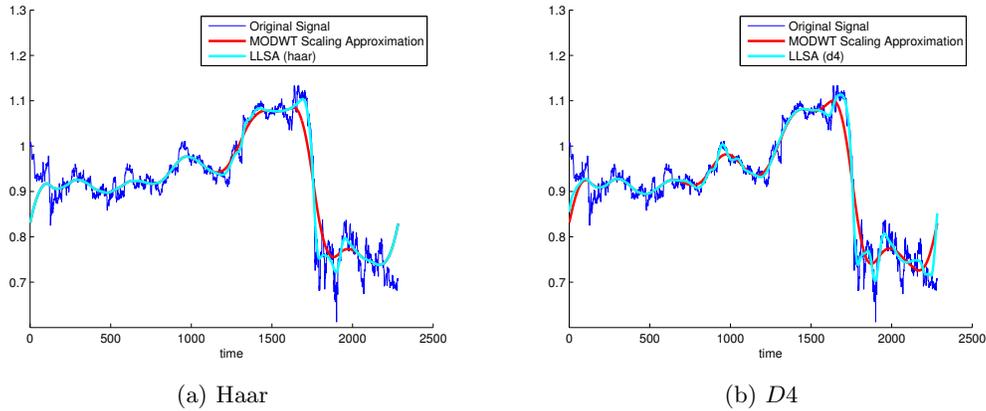


Figure 5.10.: LLSA(Haar and  $D4$ ) filtered trend of the SAP stock data,  $K = 3$

multiresolution decomposition is conducted. As this bias has the largest impact at the very ends of the signal, which at the same time are the most critical points for any forecasting method, it may at least be questioned whether wavelet transforms are an appropriate tool. However, many practitioners seem to ignore or neglect this fact. For example, they perform with either the DWT or the MODWT a multiresolution analysis of a given signal, and perform a forecasting method (e.g., ARIMA) on each of the subseries  $S_J$  and  $D_j$  (or  $V_J$  and  $W_j$ , respectively) to finally derive the prediction through Equation (3.14), see [29, 101, 120]. Even though some authors require stationarity, this does not satisfy the possibility to disregard the boundary aspect completely.

The above argumentation is also strengthened by the fact that not even one of the undoubtedly most important works and reviews about the application of wavelet transforms (see [1, 4, 88]), even mentions the topic of forecasting/prediction. Note that this argumentation only aims at wavelet transforms themselves and not wavelet methods in general. Indeed there exists the possibility to forecast signals with wavelet methods that do not rely on wavelet transforms only, but instead derive a function based on wavelet bases, that is, a wavelet process that mimics the original signals characteristics (see, for example, [44]).

### 5.2.2. Price Volatility Estimation

Now the usefulness of applying LLSA on empirical high-frequency financial time series data is evaluated. The same data as in Section 5.1.3 is used.

#### Evaluation Methodology

Analogously to Section 5.1 the MODWT and the median filter are chosen as benchmarks for the algorithms' performance, though one does not expect the median filter to deliver a sufficient smooth trend, that is, without any ripples. As in Section 5.1.3 this study is conducted using the Haar wavelet only, due to the same reasons. Since the data is the same, the same configurations are carried out, that is, the initial and refinement scale are set to  $J = 7$  and  $\Lambda = 2$ , respectively. However, as this time the performance of LLSA with respect to the number of refinement sections is to be analyzed, the algorithm is configured to detect from one jump only up to three, i. e.,  $K \in \{1, 2, 3\}$ . It is stressed again that this choice of  $(\Lambda, K)$  is certainly not optimal for all stocks and should be chosen separately dependent on each stock data for practical purposes.

In order to increase the statistical significance, the filters were applied on the whole time series not only once, but successively in a window of  $\kappa$  times the filter size, i. e.,  $\kappa \cdot 2^J$ , with  $\kappa = 7$ . The choice of  $\kappa = 7$  was done based on the robustness and performance results of Section 5.1.3, as a moving window filter size too high yielded a worse performance of LLSA in comparison to the MODWT. However, the smallest value  $\kappa = 4$  was not chosen though it would seem the most preferable choice, as in this application scenario it is important to consider only samples not affected by boundary distortions, as noted in Section 5.2.1. Since it is well known that the kind of filters considered here do not perform well at the boundaries of the signal (see also Section 3.2), for each estimated trend in the moving window the first and the last  $2^J$  data points are ignored in order to avoid this phenomenon. Hence, the effective filtering window size equals 28% of the total signal length, and equals an  $5 \cdot 2^J$  sized sample that is unbiased at its ends, that is, the moving window size is the same as in the Section 5.1.3. To compare the performance of the algorithms, an ARMA( $R, M$ ) model is derived to calculate the conditional expected value for the one-step-ahead forecasting. For  $(R, M)$  the feasible

sets  $(R, M) \in [(1, 1), (2, 1)]$  are determined, which are also often used in practice.<sup>7</sup>

Additionally, for each algorithm the remaining historical noise distribution of the detrended signal is used to derive the upper and lower percentile bounds for  $\gamma = 0.05$  for each time spot ( $LB_t$  and  $UB_t$ , respectively) for the next-step forecasting at  $t + 1$ . The mean  $d_B$  width between the percentile bounds is compared, that is, the mean over

$$d_{B,t} = UB_t - LB_t.$$

In this way it can be analyzed which algorithm provides the best base for the combined trend extraction and succeeding noise distribution estimation.

The reader is reminded that in this case the interim forecasting results are only used as an evaluation tool to verify each algorithm's accuracy in trend and variance estimation and in practice would and should not be used for actual forecasting (see Section 5.2.1), due to the boundary distortions which are common among all moving window filters. Note that this kind of procedure can be seen in analogy to the usage of the Black-Scholes formula (see [16]) in practice. While the formula was originally designed to calculate the unique prices for derivatives on financial markets, many traders "misuse" the formula to reversely derive from the empirical prices the stochastic volatility of the underlying price processes, which normally have to be estimated (see [57]). In this way, the forecasting is misused, that is, the conditional mean and the deviations from the percentile bounds, which however, only delivers reliable results when not affected by boundary distortions. This is in contrast to the procedure in Section 5.1.3, where, though it could also have been applied in the same manner, was not essential to validate the results. This was due to the reason that the boundary distortions affected the several goodness-of-fit tests in the same way and, hence, the robustness of the method was shown incorporating these distortions. The reader is reminded that this procedure does not mean to generally exclude LLSA's or the benchmarks' application in forecasting scenarios, but that the necessary boundary

---

<sup>7</sup>Other settings like  $(R, M) = (1, 2)$  were also considered, but led constantly to errors in the calculation, as for the detrended series they did not fit and no solution was available. The above chosen parameter settings for  $(R, M)$  are also in agreement with the autocorrelation plots analyzed for different samples, which exhibited a high autocorrelation.

handling methods need to be chosen carefully, as they depend critically on the concrete setting. Therefore, in this thesis, the above procedure is followed in order to avoid the introduction of any additional biases.

## Results

The results for the conditional mean forecasting for  $K \in \{1, 2, 3\}$  are reported in Tables B.1 to B.3. In the first four columns the mean of  $d_t$  is stated for all three filtering methods and ARMA(1,1) itself, and in the subsequent four columns the analog results for ARMA(2,1). It can be seen that for the majority of the stock data for higher  $K$  the LLSA algorithm performs better than the ARMA and MODWT filter models. Also, by comparing the results column by column, one notices that there is no significant difference between the two conditional mean models. Therefore, in Tables B.4 to B.6 using the same bootstrapping method as in Section 5.1.3 with 50.000 samples uniformly drawn from the empirical results, and  $\alpha = 0.05$ <sup>8</sup>. For  $K = 1$  there is  $LCB \leq 0$  and  $UCB \geq 0$  for almost all stocks and, thus, cannot conclude on any significant improvement nor worse performance of LLSA. For  $K \in \{2, 3\}$  note that the number of stocks for which there is a significant improvement increases, while the few stocks for which LLSA performs significantly worse is reduced to three (the BAS, EPC, and SDF stocks filtered by the median filter). Analyzing these particular stocks it is found that the reason is that these stocks exhibit one or more extremely high pure jumps, which are captured better by the median filter. To reach a better performance of LLSA, the algorithm would have to be set up with a larger  $\Lambda$  that could capture this edge but would lead inevitably to more overall ripples and a Gibbs phenomenon like effect near the jumps.

In Tables B.7 to B.9 in the first three columns the total amount of percentile deviations (i. e., exceedances) is stated, that is, how often for each algorithm's estimated distribution the real value is outside the  $[\gamma, 1 - \gamma]$  percentile bounds. Though in several cases the MODWT and median filter have a smaller total amount than LLSA and are nearer to the expected 5% mark, the next columns show that particularly with higher  $K$  for almost all of the tested data LLSA provides percentile bounds with a narrower width. Analog as above the bootstrapping method is used to show the significance of these results and

---

<sup>8</sup>This parameter was set to be in concordance with  $\gamma$ .

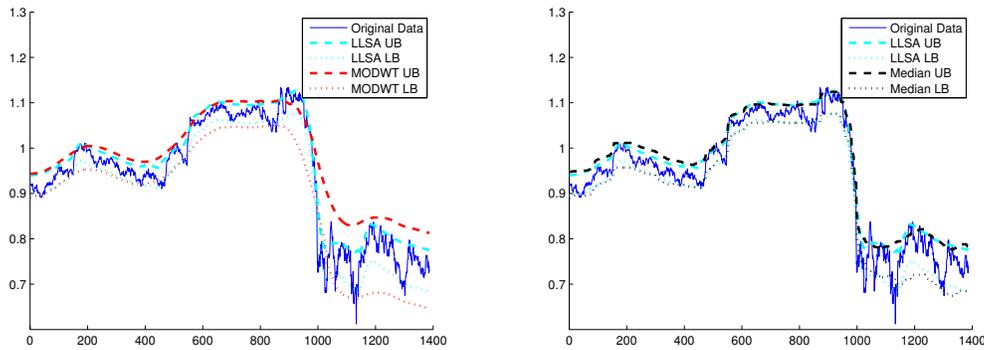
Table 5.5.: LLSA inferior performance amount in percentile exceedances

	MODWT		Median	
$K = 1$	13	(43%)	16	(53%)
$K = 2$	12	(40%)	16	(53%)
$K = 3$	8	(27%)	12	(40%)

report them in Table B.10. The total amount over all stocks of how often LLSA was outperformed by the MODWT and the median filter is given in Table 5.5. One notices an overall better performance for a higher choice of  $K$ , with LLSA outperforming the two alternative algorithms in at least 60% of all analyzed stock price time series for  $K = 3$ . Though the overall results clearly state that a better overall performance of LLSA can be achieved by raising  $K$ , one must take care as this must not lead to any misleading conclusions. By analyzing Tables B.7 to B.9 in detail, it can be seen that for a higher choice of  $K$ , LLSA's performance for some stocks improves, while it gets worse for others. This can be explained by the fact that in case the number of expected sudden changes  $K$  is set or estimated too high, LLSA begins to reconstruct details in areas where no such outstanding phenomena actually occurred. Therefore, a careful choice of  $K$  (and also  $\Lambda$ ) is essential in order to get the best results.

Again, for the SAP stock data, this is exemplarily depicted in Figure 5.11. It can be seen that particularly after a significant jump occurred (e. g., around data point 1000) the MODWT percentile estimations are much larger than necessary.

The interpretation of these results is straightforward. While the conditional mean states that in more than 50% the algorithm estimates a better overall distribution, the remaining columns show that it particularly estimates the distribution better at its tails. It is important to note that LLSA achieves a smaller number of exceptions beyond the percentile while estimating these percentiles with a higher accuracy, that is, a smaller mean distance  $d_B$ . This is in concordance with the intuition that LLSA focuses on estimating jumps and steep slopes that are usually located at the end of each distribution and can lead there to an overestimation, if not removed previously.

Figure 5.11.: SAP stock price percentile deviations,  $K = 2$ .

### 5.2.3. Estimating Value at Risk of High-Frequency Data

While in the previous section the benefits of LLSA's application for volatility estimation of stock price data were discussed, in this section the focus lies on the results one can achieve for value at risk (VaR). The proceedings in this section follow closely [38], which is recommended for reference and discussion of further details omitted here. Value at risk denotes essentially a risk measure that provides a worst case estimation of how much can be expected to be lost inside a certain time interval up to a specific confidence level. Therefore, as was done in for the volatility estimation, percentiles are calculated using the empirical data set as in the previous sections. However, VaR requires to work on the profit/loss data, which can easily be derived via  $P/L_t = P_t - P_{t-1}$ , where  $P/L_t$  denotes the profit/loss at time  $t$ , and  $P_t$  the respective stock price.<sup>9</sup>

#### Evaluation Methodology

The same evaluation methodology as in Section 5.2.2 is followed, that is, a moving window of size  $7 \cdot 2^J$  is used, while the boundary regions at its ends are disregarded. The stock price data was detrended by the respective algorithms (i. e., LLSA, MODWT, and the median filter) and the remaining series transformed into P/L data. Though the

<sup>9</sup>Interim payments are ignored in this case.

detrending process results in "negative" prices, this aspect does not effect the eventual P/L series which is different for every algorithm.<sup>10</sup> Using this series the 5th and 95th percentiles were calculated for the one-step-ahead VaR estimation. This study focusses on nonparametric VaR estimation models, since due to remaining irregularities no parametric model for the P/L data can reasonably be assumed.

In this work two independent VaR measures were calculated. As the ordinary VaR measure (i. e., the lower percentile bound) was criticized due to its simplicity and probable misleading implications (see, for example, [56,110,111]), the authors in [7,8] argue to use a more coherent approach, that is, the risk measure is subject to an additional set of axioms, like monotonicity and subadditivity. The expected shortfall (ES) model proposed by [2,3] is one risk measure that satisfies these stated coherent risk measure properties. The ES model does not simply consider a single quantile, but instead an average over the worst  $100(1 - \gamma)\%$  losses. In this discrete setting, this was done by averaging over  $M$  percentiles, with  $\gamma_M = \gamma - (m - 1) \cdot (0.05/M)$ ,  $m = 1 \dots, M$ . For the here considered data it was found that setting  $M = 50$  was sufficient to yield accurate results, that is, setting  $M > 50$  did not lead to any notable changes or different implications. In addition to the value at risk estimation using the detrended P/L data, also the VaR and ES were calculated using the original P/L series. As was done for the volatility estimation, the number of exceedances is measured. As the percentiles are calculated according to  $\gamma = 0.05$ , any percentaged deviation that lies closer to 5% can be considered as more accurate.

Furthermore, the method of [68] is used to verify the reliability and accuracy of the estimated models. This method, which is widely used in practice, is a two-sided basic frequency (also: binomial) test, that is, the 95% confidence intervals for either the absolute or the percentaged number of exceedances are calculated in order to decide whether to accept or to reject the model.

---

<sup>10</sup>This is also the reason why return series were not used in this approach, since the return calculations cannot handle negative prices directly and would require further transformation of the detrended data.

## Results

The results of the above evaluation for the detrended series are reported in Tables B.11 to B.13 for  $K \in \{1, 2, 3\}$ . First, it can be seen that the exceedances of the ordinary VaR model are always considerably higher than their ES model counterparts. Furthermore, for all  $K$  the performance of the LLSA VaR estimation in relation to the MODWT and the median filter remains nearly the same, that is, in about one third (half, respectively) of the cases, the MODWT (median filter, respectively) performs better. However, for all algorithms the percentaged exceedances are too large as they could be accepted as a viable model. The (analog) results of LLSA's performance in relation to the original VaR model are stated in Table B.14.

For the ES model the amount of percentaged exceedances lies much closer to the expected 5% mark. In fact, using the above named model verification method, all estimated models were accepted but one (LLSA for the LIN stock price data). Analyzing Table B.15, it can be verified that this also holds for the ES model applied on the original (i. e., not detrended) P/L data. In Table 5.6 LLSA's performance with respect to the MODWT and the median filter, as well as the original VaR and ES model, is summarized (i. e., the numbers reported state the amount of cases where LLSA is outperformed by the alternative algorithms). One can note that for  $K = 3$  LLSA achieves the best results, that is, the ES model estimation is more accurate in at least 60% of all cases, which is in complete concordance with the results in the Section 5.2.2. Analog to these results, LLSA's performance on specific stock data improves or worsens dependent on the choice of  $K$ . However, contrary to the results pointed out there, the worst overall results in VaR estimation are achieved for  $K = 2$ . Summarizing these insights, this stresses the importance of selecting  $K$  appropriately and separately for each time series.

## 5.3. Summary

In this chapter an evaluation of LLSA's consistency was given and the benefits of its application was discussed. After an analytical proof the robustness analysis was conducted via different simulation settings (Section 5.1) as well as using empirical data

Table 5.6.: LLSA inferior performance amount in VaR and ES estimation

	VaR						ES					
	VaR		MODWT		Median		ES		MODWT		Median	
$K = 1$	13	(40%)	12	(40%)	15	(50%)	10	(33%)	14	(47%)	12	(40%)
$K = 2$	12	(43%)	10	(33%)	15	(50%)	12	(40%)	17	(57%)	13	(43%)
$K = 3$	10	(33%)	11	(37%)	13	(43%)	9	(30%)	12	(40%)	11	(37%)

with four distance measures (Section 5.1.3). It was concluded that LLSA is robust independently from specific parameter choices and performs significantly better than the linear and nonlinear benchmark filters. In Section 5.2 the limitations and possible benefits of the algorithms application to financial high-frequency data were discussed. LLSA seems to be particularly useful where occasional sharp changes need to be extracted simultaneously along with the trend in order not to falsify any succeeding analysis and estimation methods. In this thesis the better performance of estimating the time series' noise distribution was shown using empirical data.

The reported results prove that the application of the LLSA approach can lead to significant better results in the distribution estimation, particularly at its tails (note that this results is completely consistent with the findings in Section 5.1.3), which is an important issue specifically in the analysis of financial high-frequency data. Please note again that the  $(\Lambda, K)$  tuple should be selected according to the actual data in order to yield optimal results.

It can therefore be concluded that the application of LLSA may be beneficial specifically for the succeeding steps after the trend extraction, that are, the estimation of seasonalities and the noise distribution, as well as any further application using this information. Though there is no guarantee that a better trend estimation will also automatically lead to improvements in these areas (as, for example, errors of bad estimations may balance out each other), this has proven to be the case in many different settings provided in this thesis. Such more accurate results will usually lead to a better understanding of the time series' underlying systems and their development, and is also

advantageous for concrete applications, like value at risk (see [57]). As this method relies heavily on an accurate noise distribution estimation specifically at its tails, the application of LLSA can lead to significant improvements in that particular area. Note again, that in practice LLSA is not suitable for forecasting and other real-time applications without prior selecting an appropriate method (dependent on the specific time series) in order to handle any boundary distortions.

## Chapter 6.

# Conclusion and Outlook

In this chapter, the answers to the research questions raised in the introduction are discussed and the contributions of this thesis are summarized in Section 6.1. Possible future research directions are outlined in Section 6.2. The latter will be subdivided into extensions of the algorithm itself and its potential further applications.

### 6.1. Summary

Having proposed the algorithm and shown its several properties, consistency and application possibilities, it will now be analyzed which of the requirements in Section 1.1 have been met, and how the research questions stated in the same section can be answered. Furthermore, the contributions of this thesis are summarized in Section 6.1.2.

#### 6.1.1. Requirement Satisfaction and Research Questions

##### Requirement Satisfaction

Since LLSA builds on the MODWT and its reconstruction of details is limited to the near proximity of jumps, it can be stated that Requirement R1 is mostly met. Mostly, since a perfect solution may in reality not be available (and has not yet been discovered among the class of nonlinear filters). As the class of linear filters can wholly be excluded to solve this task, LLSA, extending the principles of linear filtering, certainly provides an improvement, as was shown in Sections 5.1 and 5.2. However, while the local linear

output that matches the MODWT is an advantage on one side, it is also a drawback on the other, as outliers still affect the trend in these areas.

Requirement R2 is also met in the way that in the areas where local linearity still holds (i. e., the output of LLSA matches the one of the MODWT) the low-pass filter frequencies are completely determined (i. e., controlled) by the initial scale  $J$ . Additionally, in the near proximities of jumps, where this property does not hold any longer, one still has an explicit control over the frequencies, in terms of that one can state an upper bound (by choosing  $\Lambda$  appropriately) which frequencies shall not pass. Due to the mixture of adjusted wavelet coefficients (i. e., set to zero) no statements can be made about the analytical composition and proportion of frequencies in these areas. Where the cones of influence from different wavelet coefficients (i. e., set to zero or left untouched) do not overlap for different scales (see Theorem 4.4) the output equals to the one of the MODWT, even in the near proximity of jumps. Thus, while LLSA does not provide a complete frequency control throughout the whole signal, but only lower bounds near the proximity of edged frontiers, Requirement R2 is at least partially fulfilled, which can be considered to be a commendable achievement for a nonlinear filter.

Regarding Requirement R3, since LLSA does not rely on any specific assumptions on the time series itself (besides homogeneity) nor its inherent noise structure, it can be stated that this requirement is fully met as with any of the other linear and nonlinear filters and therefore, the novel algorithm is as easily applicable as these. Though many approaches, in order to improve the performance compared to those traditional methods, impose additional assumptions and requirements on the time series and its noise, it was decided against this approach<sup>1</sup> in order not to restrict LLSA's usage. However, this does not deny or exclude the possibility of further enhancing LLSA's performance by including such further assumptions. This is discussed in the Section 6.2, where possible extensions and future work for LLSA are presented, which particularly regard specific noise structures and to these related works in the field of wavelets.

In Corollary 4.3 the computation tractability of LLSA was analyzed, and it was shown that the complexity of the MODWT,  $\mathcal{O}(N \log_2 N)$ , is preserved. Though not as good as

---

<sup>1</sup>In this way, setting  $(K, \Lambda)$  may seem heuristic, but avoids biases for specific samples.

the DWT with  $\mathcal{O}(N)$  it is still on the same level with the fast Fourier transform (FFT) that is oftentimes applied for global frequency analysis, and many other nonlinear filters, which must first order the whole sample located in the moving window (in case they are based on some order or ranking criteria). This satisfies Requirement R4.

In Theorem 5.1 it was proved that a boundary on the error for the trend estimation can be derived and that LLSA converges asymptotically towards the MODWT. As the found boundaries are very generously estimated, the reliability and consistency of the algorithm were extensively examined in simulations and empirical samples in the succeeding Sections 5.1.2 and 5.1.3, respectively. The results show that LLSA is robust and independent of the specific choice of wavelet or  $(K, \Lambda)$ . Thus, Requirement R5 is also satisfied.

Though it was found that LLSA does not perform better in all scenarios<sup>2</sup>, for the purposes it was designed for it performed better in the majority of the cases (though still depending on  $(K, \Lambda)$ ). Though, as pointed out in Section 5.2, this does not lead necessarily to better results, the tests in Sections 5.1 and 5.2 strengthen the argumentation that LLSA is a promising contribution to enrich the class of nonlinear filters and, due to its local linear properties, can be seen as bridging the gap between these two classes of linear and nonlinear filters. A comparison to the alternative methods discussed in Chapter 2 is provided in Table 6.1.

Considering the framework in Section 2.1, LLSA can be characterized as a nonparametric, nonlinear filter used for the approximation of the long-term trends in nonstationary, univariate time series, and can be associated with time as well as with the frequency domain. As the otherwise extracted trend will (by design) exhibit sharp changes it cannot be classified as a traditional smoothing approach, neither is it a pure denoising method, since it also aims at the removal of other (deterministic) components. It is stressed that LLSA *cannot* directly be used for forecasting/prediction, due to the boundary distortions which are common to all moving window filters. Instead, as was shown in Section 5.2.2, one can expect (but not guarantee) LLSA to yield a better signal decomposition, which in turn will lead to more accurate results in the succeeding steps, like volatility es-

---

<sup>2</sup>Any other statement would be dubious, as no single method can serve as a best solution for all cases.

Table 6.1.: Today's algorithms' and LLSA's requirement fulfillment

	R1	R2	R3	R4	R5
Linear filters	○	●	●	●	●
Nonlinear filters	◐	○	●	●	●
Dedicated jump models	●	○	◐	●	●
General least-squares models	◐	○	◐	◐	◐
Smoothing splines & HP filter	◐	○	●	●	●
Kalman filter	●	○	◐	●	◐
LLSA	●	◐	●	●	●

timation, and thus, provide a better understanding of the time series and the underlying system itself. This can be used as an input to, for example, other forecasting methods that do not suffer from these boundary restrictions.

### Research Questions

Taking the above findings into account, the research questions stated at the beginning of this thesis can be answered as follows.

As was seen, since nonlinear filters generally do not account for any a priori frequency control that fulfills Requirement R2, only linear filters provide that desired feature, which designated them to serve as a starting point of the new algorithm's development. They also fulfill Requirement R1 with respect to the smooth trend, which lead to the idea of reconstructing lost details. Of the class of linear filters, wavelets (and their discrete transforms) were chosen due to two facts: First, their excellent localized properties seemed favorable for the task of handling single marked-off phenomena like jumps, and, second, the multiresolution analysis the transforms provide can be exploited in the sense that during the transformation no information is lost, but preserved on different scales. The approach proposed in this thesis is surely not the only feasible one, but was favored here and has proven to work as expected.

As can be seen in Table 6.1 LLSA meets all requirements but R2, which is only partially fulfilled. Though it cannot be excluded that a filter might be developed that enables complete frequency control of the whole signal it is a questionable ambition, due to the contradiction of eliminating and preserving details located in the same frequency range, that differ only in their magnitude. The approach presented in this thesis manages to provide an in-between solution, that is, complete frequency control over the areas of the signal where the trend is smooth, and lower bounds in the proximity of sudden changes. This answers research question RQ 1.

The characteristics and properties of the algorithm were proven in Section 4.2. LLSA has the same computational complexity as the MODWT it is based upon and furthermore can be applied without any restrictions on all time series as comparable moving window filters. The algorithm's robustness was shown in several ways (analytically, via simulations, and by using empirical data) in Section 5.1. In comparison to the MODWT, LLSA additionally needs only the parameter tuple  $(\Lambda, K)$ , that is, the degree of refinement and the number of sections to be refined. Both parameters can either be determined heuristically or, for  $\Lambda$ , by using any additional information about the frequencies contained in the signal, while the optimal choice of  $K$  can be made using any method mentioned in Section 2.4.1, depending on the setting and the information that is available. It was also seen that these additional parameters depend on the choice of wavelet, though, based on the results of Section 5.1, in most cases LLSA is recommended to be used with either the Haar or the  $D4$  wavelet. Along with LLSA's explicitly stated properties, that are, local linearity, the algorithm's impulse and step response, as well as its bounded approximation error and asymptotic convergence towards the MODWT, this answers research question RQ 2.

Several analyses were conducted in this thesis to prove in which cases LLSA performs superior in relation to other benchmark filters. The simulations in Section 5.1.2 have shown that LLSA performs always better in non-extreme settings (i. e., very high or low noise), regardless of the choice of wavelet and even for bad estimations of  $K$ . The succeeding empirical studies clearly demonstrate that LLSA is a valuable addition to today's established filters, also, as it can be thought of being in-between the linear

and nonlinear filter classes. Though, of course, LLSA will not perform superior in all scenarios, for the data sets evaluated in this thesis it did so in a considerable number of cases. As LLSA is based on a moving window filter, as with all filters of that kind, its applicability in forecasting is limited, but not impossible, given the right handling of boundary conditions. The main benefit of this algorithm's application can be seen in that it usually leads to a more accurate estimation of the components remaining in the time series, that are, seasonalities and noise, though no guarantee can be given for that as well. However, the case studies and their evaluations presented in this thesis admit the conclusion that this is in fact the case even if  $(\Lambda, K)$  are not calibrated optimally. Thus, research question RQ 3 is also answered positively.

### 6.1.2. Contributions

Although jump detection can today be considered to be a task well understood and realized by different methods, the LLSA approach goes beyond that. While other approaches focus mainly on the detection and representation of jumps, LLSA incorporates a more general notion, by not considering jumps only, but also extreme regions containing other occurrences like steep slopes, roofs and valleys. While a jump may easily be represented or parametrically modeled (e.g., via indicator functions) this does not hold for the other sudden phenomena, for which it is challenging to assume certain parametric models in general. LLSA's nonparametric detail reconstruction approach treats this task while maintaining the flexibility to be adjustable over different scales. The nonlinear characteristic is mandatory to include high-frequency events in the otherwise low-frequency trend. However, in contrast to other nonlinear filters, the approach proposed in this thesis still preserves the properties of linear filters outside those critical regions and thus, enables the analyst to maintain frequency control over the output. Summarizing, it can be said that LLSA enriches the class of nonlinear filters by providing a bridge from them to their linear counterparts, especially in respect of frequency analysis. The contributions of this thesis can thus be summarized as follows:

- A new algorithm was developed that handles the task of the trend extraction from one-dimensional discrete signals that occasionally exhibit sudden changes. However, this approach is not limited to jumps only, but also takes other phenomena

like steep slopes and valleys into account, which differentiates itself from alternative filtering methods with comparable application requirements. Furthermore, to the author's best knowledge, the LLSA algorithm is today's only available method that fulfills requirements R1 to R5 conjointly to this degree.

- To foster the algorithm's applicability in other scenarios and research fields, its general properties were shown. The algorithm, characterized by these properties, has proven that it is generally neither disadvantaged nor inferior with respect to alternative filtering methods.
- The limitations as well as the benefits that can be expected from LLSA's application in real scenarios have been discussed and its superior performance has been shown in a number of cases, using simulations as well as real data. Though the eventual performance still depends on the analysis and the succeeding methods itself, the algorithm can be considered to be applicable to all kinds of signals in the most diverse research and application areas, as long as its (few) requirements are met.

## 6.2. Future Research Directions

In this section, several possible future research directions are discussed, which were not considered in this work or only touched on. These directions are divided into two subtopics, namely further research considering the LLSA algorithm itself, and application scenarios being enabled by LLSA's usage.

### 6.2.1. Algorithmic Extensions

Another variation of discrete wavelet transforms not explicitly mentioned (as they play no further role in this thesis) in Section 3.2 are the discrete wavelet packet transform (DWPT) and its MODWT counterpart, the maximal overlap discrete wavelet packet transform (MODWPT). The pointed out differences between the DWT and the MODWT hold also for the DWPT and the MODWPT. The basic idea behind these transforms is that through the multiresolution analysis one does not receive only one scaling co-

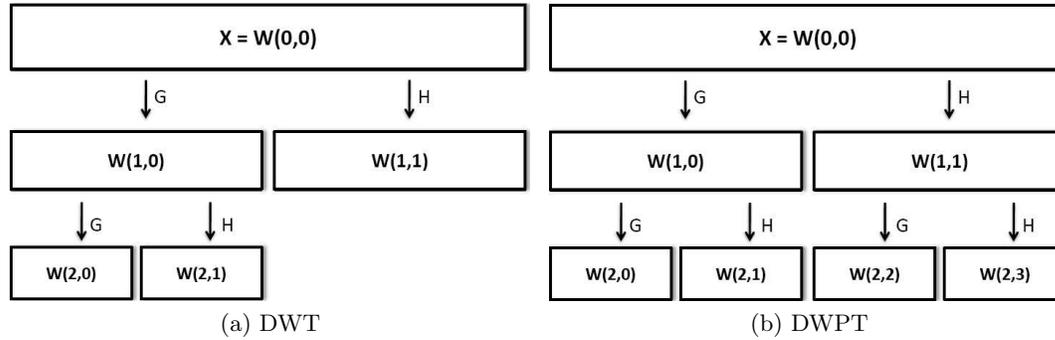


Figure 6.1.: Flow diagram of wavelet transforms

efficient vector  $V_J$  and  $J$  wavelet coefficient vectors, which are related as in Equation (3.15). Instead, one gets a full decomposition, that is, for each  $V_j$  and  $W_j$  there are one succeeding scaling approximation and detail vector each, see Figure 6.1 (adapted from [88]). This by far more redundant decomposition enables time series representation with best basis selection (where *best* depends on the metric used) and thus, provides much more representations of the time series from different viewpoints. In combination with an appropriate basis, these viewpoints would offer additional representation possibilities that provide more flexibility for choosing how to best reconstruct lost details of jumps. That is, the algorithm would not be restricted to the  $W_j$ ,  $1 \leq j \leq J$ , but have access to a variety (i. e.,  $\sum_{j=1}^J 2^j + 1$ ) of different detail combinations. As this new multiresolution decomposition also incorporates the original decomposition vectors of the DWT (or MODWT, respectively), clearly the possibility of an even better trend representation via the additional vectors is given.

### Improvement for White Gaussian Noise

In this work, no information concerning the noise structure was used at all or only implicitly, by setting  $J$  and  $\Lambda$  accordingly. In case of white Gaussian noise, information about its variance  $\sigma$  may be used to improve the performance of the LLSA algorithm:

According to [35] all wavelet coefficients below the threshold

$$\delta_\sigma = \sqrt{2\sigma^2 \log(N)}$$

are dominated by noise and can thus be discarded. This would also affect the definition of  $n_\alpha$  and  $n_\beta$  in Equations (4.4) and (4.5). As for  $|W_j| < \delta_\sigma$  these coefficients hardly contain any information about the signal itself, Equations (4.2) and (4.3) may be redefined to

$$\begin{aligned} l_{\min,j}^s &:= \min \{t \mid |W_{j,t}^s| \geq \delta_\sigma\}, \\ l_{\max,j}^s &:= \max \{t \mid |W_{j,t}^s| \geq \delta_\sigma\}. \end{aligned}$$

This will usually lead to smaller  $n_\alpha$  and  $n_\beta$ . If Equations (4.11), (4.12) and (4.13) are further modified to

$$\Omega_{j,k}^{W,\delta_\sigma} := [\alpha_{j,k}^{\delta_\sigma}, \beta_{j,k}^{\delta_\sigma}],$$

and

$$\begin{aligned} \alpha_{j,k}^{\delta_\sigma} &:= \max \{l \in [\alpha_{j,k}, l_{j,k}^W - 1] \mid \forall t \in [l, l_{j,k}^W - 1] \text{ holds } |W_{j,t}| < \delta_\sigma\}, \\ \beta_{j,k}^{\delta_\sigma} &:= \min \{l \in [l_{j,k}^W + 1, \beta_{j,k}] \mid \forall t \in [l_{j,k}^W + 1, l] \text{ holds } |W_{j,t}| < \delta_\sigma\}, \end{aligned}$$

all wavelet coefficients below the threshold  $\delta_\sigma$  are cut off, yielding an even more localized detail restoration. Furthermore, one can investigate the effect of additionally setting all retained wavelet coefficients to zero that fall below that threshold, that is, for Equation (4.9) one has

$$\widetilde{W}_{j,t}^s = \begin{cases} W_{j,t} & \text{for } t \in \bigcup_{k=1, \dots, K} \bigcup_{j=J-\Lambda, \dots, J} \Omega_{j,k}^W \wedge |W_{j,t}| \geq \delta_\sigma, \\ 0 & \text{otherwise.} \end{cases}$$

The above equations can be associated with *hard thresholding*. The flexibility of LLSA also allows for the application of other thresholding rules (i. e., *soft*, *mid* and *firm*) and scale dependent thresholds, that is,  $\delta_{\sigma,j}$ ,  $\Lambda - J \leq j \leq J$  (see [1] and the references therein for further details).

Furthermore, if there is any information about the noise and its structure, and, thus, one can derive a threshold  $\widehat{\delta}_\sigma$  to determine whether a jump is significant (i. e., whether or not it can be contributed to daily fluctuations, see Section 2.4.1), due to LLSA's basic procedure and convertibility,  $K$  can be determined by

$$\widehat{K} = \max\{k \mid l_{J,k}^W \geq \widehat{\delta}_\sigma\}$$

### Alternative Initial Jump Detection

In Section 4.1.1 a rule was presented for determining  $l_{J,k}^W$ ,  $k = 1, \dots, K$ , that is, the locations of jumps and slopes. By evaluating single filtered time series it was noticed that sometimes the jumps and slopes recognized and restored by LLSA were not the ones that should have been reconstructed in the first place (i. e., they were not the most obvious ones). Hence, this rule may also be replaced or combined with any of the other methods referred to in Section 2.4.1. This can be achieved by setting the initial LLSA jump estimation  $l_{J,k}^W$  in Equation (4.14) according to one of these proposed methods. However, note that Equation (4.15) still remains unchanged.

### Alternative Jump Detection on Different Scales

The formulation in Equation (4.15) ensures that always the same critical section is refined on every scale. Thus, in case there exist several occurrences of minor jumps on the finer scales, it is ensured that once LLSA opts for the refinement of one critical section via the rules in Equations (4.14) and (4.15), on the next lower levels only the details of the very same jump get further reconstructed. This rule is necessary, since the critical sections from the perspective of a particular scale will not be the same as the ones from the next higher or lower one. However, it is possible to state another slightly different formulation

$$l_{j,k}^I := \operatorname{argmax}_t (|W_{j,t}| \mid t \in \Omega_{J,k}^W).$$

In this way, the only restriction is that the refined jump at each scale  $J - \Lambda \leq j < J$  must be detected inside the one from the original perspective at scale  $J$ . The further investigation of these rules and the impact on LLSA and its applications may be the

subject of future works.

### Multivariate Extension

In this work there were only considered time series or discrete univariate signals. But since wavelet transforms are not limited to one dimension, and furthermore denoising and edge preservation are an important part also in multivariate analyses, specifically two-dimensional images, the following paragraph will outline how this thesis' approach can be extended to multiple dimensions (with the focus here on images).

First note that jumps are not unambiguously defined any more as in univariate settings. This becomes specifically important as LLSA is built upon the wavelet coefficient step response structure, see Definition 4.1, which need to be carried over to two dimensions. While an impulse is clearly defined, that is a single nonzero value, for the step response there is either a pure step or an edge like structure (or anything in-between), see Figure 6.2. Additionally, while in the univariate case one could transfer the wavelet coefficient step response onto the measured signal  $X$  by considering its change of signs (see Equations (4.4) and (4.5)), now there must be used contours instead. Furthermore, the multivariate case is not confined to dyadic scaling as in Equation 3.8, but instead has

$$\varphi(t) = \sqrt{|\det(M)|} = \sum_{k \in \mathbb{Z}} \bar{g}_k \varphi(Mt - k),$$

where  $M$  is the scaling matrix, which can take on several forms depending on the actual mesh (e. g., the quincunx or the twin dragon) used for the multiresolution analysis (see [80] for details).

Considering above points, it can be seen that there are still numerous open research questions to further develop LLSA and explore its optimal application scenarios.

#### 6.2.2. Further Applications

There was already depicted one possible application in Section 5.2, that is, a higher accuracy in distribution estimation. It was shown how the usage of LLSA as the very first step of trend extraction can lead to better results. Generally said, the more accurately

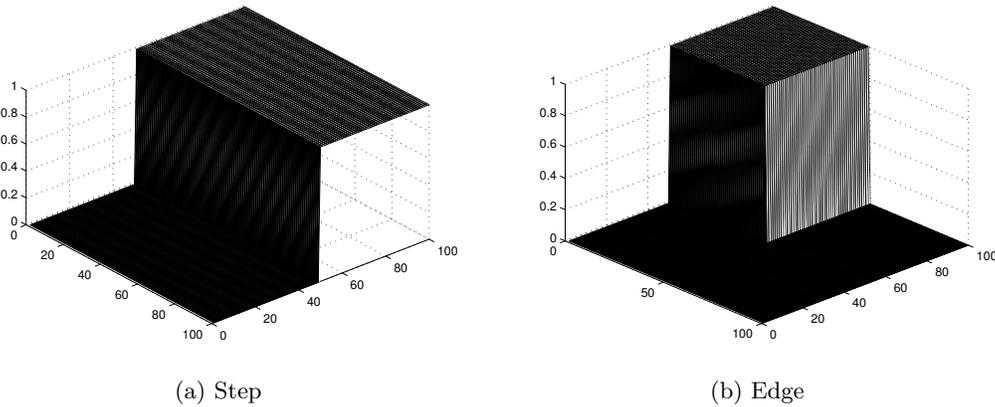


Figure 6.2.: Two-dimensional response structures

the trend is extracted, the more accurately one can expect any succeeding steps to perform. This holds for business cycle and variance estimation, as well as any further analyses and models using this information.

However, the usage of LLSA also provides other advantages in applications from other perspectives. The first issue is a phenomenon called *oversmoothing*, that is, if too many significant details of the trend are lost during the filtering process. This is, of course, related to the main issue of this work that edges and jumps are blurred out, but on a different scale, that is, oversmoothing is generally associated with a wrong choice of the appropriate initial approximation level, i. e.,  $J$  is chosen to high. As [1] notes the choice of  $J$  may be a daunting task, eventually equivalent to the choice of bandwidth for (non) linear filters. While [88] notes that this choice must depend on the data and the noise at hand, for the same measured time series, different choices of  $J$  may be appropriate, dependent on the final aim of the analysis (e. g., extract trends over different periods of length, i. e., mid- and long-term). Though some approaches (like cross-validation) exist to support the analyst in this task (before the actual analysis), they are generally not error free. Though this will not be changed by LLSA, this approach adds an interesting aspect to this. In case  $J$  was chosen too high, LLSA still can provide a very good

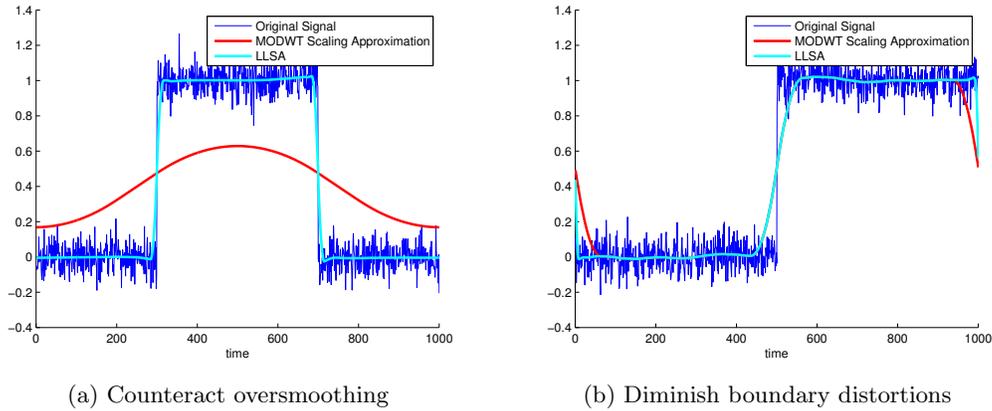


Figure 6.3.: Further applications

approximation of the trend including an accurate resolution of its details, see Figure 6.3a. However, in this case, as the initial coarse resolution of the jump covers a much larger proportion of the filtered signal, the areas that can be considered as the output of a linear filter, will generally be significantly smaller or vanish at all.

Another aspect is LLSA's capability to diminish boundary distortions. As noted in Section 3.2 for all kinds of moving window filters, there are data points missing at the beginning and at the end of the signal, that is, where the filtering window exceeds the measured data points. In order to still conduct the filtering process, missing data points need to be substituted by one of the methods named in Section 3.2. Nevertheless, the areas around the signals boundaries will always be biased, independent of which method is used. As the filtered output in these regions are distorted anyway, one can not take them as a input for further valid statements and uses. However, what can be done intuitively, is to switch to the next lower scale which effectively halves the window length each time. In this way more details are added to the trend at the boundaries, but distortions are avoided. Though this idea is not exclusive to LLSA (it can also be achieved with the MODWT or any other moving window filter) an exemplary application is depicted in Figure 6.3b.



# Appendix A.

## Empirical Robustness Study

### A.1. Statistical Ratios Tables

The tables provided in this appendix show several statistics for our empirical robustness study in Section 5.1.3 for the 60 minute frequency data. Only the tables for the extreme window sizes  $\kappa = 4$  and  $\kappa = 10$  are stated. The tables are to read as follows. In each table caption is given the goodness-of-fit distances (i. e., Kolmogorov-Smirnov, Anderson-Darling, Kuiper, and Cramér-von Mises, that is, one table for each test), together with the multiplier  $\kappa$  of the filtering window size  $J = 2^7$ .

The statistics provided are the mean, median, variance, minimum and maximum of the by LLSA (subscript L) and MODWT (subscript M) of the resulting moving window trend estimation time series.

Table A.1.: Kolmogorov-Smirnov distance statistics,  $\kappa = 4$ 

	mean <sub>L</sub>	mean <sub>M</sub>	median <sub>L</sub>	median <sub>M</sub>	var <sub>L</sub>	var <sub>M</sub>	min <sub>L</sub>	min <sub>M</sub>	max <sub>L</sub>	max <sub>M</sub>
ADS	0.1962	0.2240	0.1914	0.2168	<b>0.0021</b>	<b>0.0017</b>	0.1172	0.1563	0.3281	0.3887
ALV	0.2045	0.2557	0.2109	0.2402	0.0018	0.0022	0.1074	0.1582	0.3281	0.3945
BAS	0.2148	0.2594	0.2090	0.2539	0.0018	0.0025	0.0957	0.1621	0.3105	0.3770
BAY	0.1805	0.2512	0.1797	0.2500	<b>0.0039</b>	<b>0.0025</b>	0.0801	0.1699	0.3184	0.3926
BEI	0.2205	0.2830	0.2168	0.2832	0.0021	0.0042	0.1055	0.1465	0.3281	0.4141
BMW	0.2225	0.2857	0.2266	0.2910	<b>0.0028</b>	<b>0.0025</b>	0.1172	0.1660	0.3965	0.4219
CBK	0.1929	0.2149	0.1914	0.2168	<b>0.0019</b>	<b>0.0009</b>	0.0859	0.1641	0.2891	0.2891
DAI	0.2120	0.2665	0.2148	0.2715	0.0016	0.0020	0.1309	0.1758	0.3125	0.3652
DBK	0.1916	0.2187	0.1875	0.2227	<b>0.0023</b>	<b>0.0021</b>	0.0820	0.1504	0.3027	0.3398
DB1	0.2298	0.2792	0.2344	0.2734	0.0034	0.0039	0.1113	0.1738	0.4082	0.4082
DPB	0.2014	0.2404	0.1914	0.2285	<b>0.0033</b>	<b>0.0024</b>	0.1016	0.1719	0.3418	0.3945
DPW	0.1689	0.2271	0.1641	0.2168	0.0015	0.0023	0.0820	0.1602	0.3203	0.3750
DTE	0.2341	0.2702	0.2217	0.2441	<b>0.0070</b>	<b>0.0048</b>	0.0898	0.1641	0.4141	0.4141
EPC	0.2478	0.2893	0.2422	0.2910	<b>0.0043</b>	<b>0.0042</b>	0.1035	0.1680	0.4336	0.4395
FME	0.2179	0.2607	0.2266	0.2441	<b>0.0048</b>	<b>0.0042</b>	0.0879	0.1777	0.3711	0.4414
LHA	0.2205	0.2595	0.2129	0.2539	0.0023	0.0032	0.1426	0.1523	0.3496	0.3691
HNK	0.1751	0.2512	0.1738	0.2363	0.0022	0.0041	0.0879	0.1289	0.3008	0.3730
IFX	0.1767	0.2236	0.1836	0.2129	<b>0.0026</b>	<b>0.0010</b>	0.0898	0.1699	0.2871	0.3359
SDF	0.2204	0.2711	0.2227	0.2461	0.0018	0.0039	0.1191	0.1875	0.3613	0.4531
LIN	0.2432	0.2680	0.2383	0.2734	<b>0.0023</b>	<b>0.0017</b>	0.1621	0.1680	0.3906	0.3906
MAN	0.2149	0.2426	0.2168	0.2324	<b>0.0035</b>	<b>0.0021</b>	0.1152	0.1641	0.3770	0.3770
MRC	0.2238	0.2625	0.2090	0.2637	<b>0.0048</b>	<b>0.0039</b>	0.1152	0.1660	0.4453	0.4453
MEO	0.2064	0.2689	0.2109	0.2529	0.0025	0.0030	0.0879	0.1777	0.3223	0.3906
MUV	0.2226	0.2782	0.2188	0.2793	<b>0.0058</b>	<b>0.0051</b>	0.0898	0.1563	0.4199	0.4375
RWE	0.2155	0.2497	0.2090	0.2344	0.0022	0.0042	0.1016	0.1699	0.3535	0.4395
SZG	0.2020	0.2478	0.1973	0.2461	0.0027	0.0031	0.1211	0.1758	0.3457	0.4277
SAP	0.2078	0.2783	0.2109	0.2871	<b>0.0046</b>	<b>0.0031</b>	0.0996	0.1660	0.3594	0.3809
SIE	0.1885	0.2354	0.1875	0.2285	<b>0.0021</b>	<b>0.0013</b>	0.1152	0.1797	0.2852	0.3574
TKA	0.2003	0.2357	0.1992	0.2246	<b>0.0022</b>	<b>0.0021</b>	0.0918	0.1699	0.2988	0.3848
VOW	0.2329	0.2809	0.2383	0.2852	<b>0.0045</b>	<b>0.0026</b>	0.0918	0.1973	0.4043	0.4277

Table A.2.: Anderson-Darling distance statistics,  $\kappa = 4$ 

	mean <sub>L</sub>	mean <sub>M</sub>	median <sub>L</sub>	median <sub>M</sub>	var <sub>L</sub>	var <sub>M</sub>	min <sub>L</sub>	min <sub>M</sub>	max <sub>L</sub>	max <sub>M</sub>
ADS	4.3728	5.0268	4.2468	4.8661	<b>1.0748</b>	<b>0.8900</b>	2.6100	3.4063	7.3876	8.7590
ALV	4.4107	5.7474	4.4680	5.3970	<b>1.1723</b>	<b>1.1146</b>	2.2119	3.4063	7.3876	8.8917
BAS	4.5771	5.5015	4.4680	5.5297	<b>1.2633</b>	<b>1.1437</b>	2.1234	3.6275	6.9453	8.4936
BAY	4.0186	5.6449	4.0256	5.6181	<b>1.9988</b>	<b>1.3016</b>	1.7695	3.8044	7.1665	8.8475
BEI	4.5707	6.3642	4.6892	6.3702	1.2130	2.1380	2.3446	3.2736	7.3876	9.3341
BMW	4.6300	6.4250	4.9104	6.5471	<b>1.8544</b>	<b>1.2716</b>	1.9464	3.7159	8.9360	9.5110
CBK	4.3080	4.8116	4.2910	4.8661	<b>1.0360</b>	<b>0.4755</b>	1.5483	3.4505	6.5029	6.5029
DAI	4.6732	5.9950	4.5565	6.1048	0.8464	1.0273	2.8754	3.9371	7.0337	8.2282
DBK	4.2666	4.9085	4.1141	4.9988	<b>1.2208</b>	<b>1.0581</b>	1.8137	3.3620	6.8126	7.6531
DB1	5.0744	6.2789	5.1315	6.1490	1.6852	2.0162	2.4773	3.8929	9.2014	9.2014
DPB	4.4668	5.3995	4.2468	5.1315	<b>1.6159</b>	<b>1.2518</b>	2.2561	3.8487	7.6973	8.8917
DPW	3.6805	5.0969	3.6717	4.8661	0.6191	1.1661	1.8137	3.5832	7.2107	8.4493
DTE	5.2068	6.0723	4.9104	5.4854	<b>3.6170</b>	<b>2.4669</b>	1.9907	3.6717	9.3341	9.3341
EPC	4.7023	6.1521	4.4237	6.1490	1.9315	2.1137	2.3003	3.2736	8.9360	9.9092
FME	4.6422	5.8598	4.7776	5.4854	<b>2.3357</b>	<b>2.1471</b>	1.7695	3.9814	8.3609	9.9534
LHA	4.8847	5.8323	4.6007	5.7066	1.1677	1.6204	3.1851	3.4063	7.8743	8.3166
HNK	3.7805	5.6450	3.4948	5.3085	0.9890	2.1281	1.9464	2.8754	6.7683	8.4051
IFX	3.9458	5.0193	4.1141	4.7776	<b>1.3037</b>	<b>0.5279</b>	1.9907	3.8044	6.4587	7.5646
SDF	4.8044	5.8466	4.6007	5.4854	0.9909	1.4031	2.6542	4.2026	8.1397	10.2188
LIN	5.4175	6.0239	5.3085	6.1490	<b>1.2305</b>	<b>0.8812</b>	3.6275	3.7602	8.8032	8.8032
MAN	4.7831	5.4252	4.7776	5.2200	<b>1.7693</b>	<b>1.1198</b>	2.5658	3.6717	8.4936	8.4936
MRC	4.9799	5.9005	4.6449	5.9278	<b>2.4579</b>	<b>1.9920</b>	2.5658	3.7159	10.0419	10.0419
MEO	4.3344	6.0470	4.1804	5.6845	<b>1.8542</b>	<b>1.5517</b>	1.6810	3.9814	7.2549	8.8032
MUV	4.7897	6.2541	4.5565	6.2817	<b>3.2454</b>	<b>2.6491</b>	1.9907	3.4948	9.4668	9.8649
RWE	4.6534	5.6013	4.6449	5.2643	1.0064	2.1814	2.2561	3.8044	7.9627	9.9092
SZG	4.4842	5.5682	4.4237	5.5297	1.3857	1.5915	2.6985	3.9371	7.7858	9.6438
SAP	4.5258	6.2560	4.4680	6.4587	<b>2.3813</b>	<b>1.6163</b>	2.2119	3.3620	8.0954	8.5821
SIE	4.0871	5.2875	3.8044	5.1315	<b>1.2227</b>	<b>0.6690</b>	2.5658	4.0256	6.4144	8.0512
TKA	4.4622	5.2931	4.3795	5.0431	<b>1.1177</b>	<b>1.0637</b>	2.0349	3.7159	6.7241	8.6705
VOW	4.9097	6.2783	4.9988	6.2375	<b>2.7922</b>	<b>1.3203</b>	1.9907	4.4237	9.1129	9.6438

Table A.3.: Kuiper distance statistics,  $\kappa = 4$ 

	mean <sub>L</sub>	mean <sub>M</sub>	median <sub>L</sub>	median <sub>M</sub>	var <sub>L</sub>	var <sub>M</sub>	min <sub>L</sub>	min <sub>M</sub>	max <sub>L</sub>	max <sub>M</sub>
ADS	0.3318	0.4056	0.3379	0.3887	<b>0.0054</b>	<b>0.0045</b>	0.1875	0.2734	0.4824	0.6328
ALV	0.3372	0.4482	0.3398	0.4473	0.0043	0.0051	0.2070	0.3086	0.5078	0.7031
BAS	0.3633	0.4504	0.3613	0.4316	0.0051	0.0089	0.1641	0.1816	0.5410	0.6660
BAY	0.3040	0.4480	0.2988	0.4512	<b>0.0089</b>	<b>0.0058</b>	0.1328	0.3047	0.5059	0.6172
BEI	0.3748	0.4886	0.3633	0.4805	0.0061	0.0117	0.1797	0.2676	0.5508	0.7305
BMW	0.3807	0.4978	0.3779	0.4844	0.0067	0.0088	0.2188	0.3301	0.6797	0.7891
CBK	0.3303	0.4026	0.3359	0.3945	<b>0.0058</b>	<b>0.0030</b>	0.1563	0.2773	0.5020	0.5527
DAI	0.3548	0.4693	0.3496	0.4629	0.0031	0.0056	0.2461	0.3398	0.5254	0.6543
DBK	0.3232	0.3912	0.3301	0.3984	<b>0.0054</b>	<b>0.0041</b>	0.1641	0.2754	0.4512	0.5293
DB1	0.3910	0.4943	0.3984	0.4785	0.0081	0.0096	0.2051	0.3008	0.6055	0.7441
DPB	0.3380	0.4283	0.3301	0.4258	<b>0.0061</b>	<b>0.0041</b>	0.1875	0.2988	0.4961	0.5918
DPW	0.2930	0.4031	0.2871	0.3867	0.0032	0.0040	0.1563	0.2773	0.5215	0.5742
DTE	0.3977	0.4893	0.3789	0.4463	<b>0.0181</b>	<b>0.0139</b>	0.1621	0.2969	0.6426	0.7188
EPC	0.3905	0.4768	0.3984	0.4902	0.0081	0.0092	0.1777	0.2500	0.6094	0.6660
FME	0.3728	0.4776	0.3594	0.4395	<b>0.0148</b>	<b>0.0147</b>	0.1680	0.3262	0.6387	0.8340
LHA	0.3643	0.4504	0.3594	0.4590	0.0034	0.0048	0.2500	0.2949	0.5391	0.5918
HNK	0.3086	0.4233	0.3047	0.4053	0.0072	0.0079	0.1660	0.2520	0.5156	0.5996
IFX	0.3095	0.4068	0.3027	0.3984	<b>0.0068</b>	<b>0.0030</b>	0.1328	0.2676	0.5059	0.5605
SDF	0.3684	0.4595	0.3594	0.4375	0.0043	0.0072	0.2168	0.2520	0.5137	0.6836
LIN	0.3929	0.4686	0.3984	0.4688	<b>0.0044</b>	<b>0.0033</b>	0.2559	0.3340	0.5469	0.5957
MAN	0.3572	0.4332	0.3535	0.4258	<b>0.0069</b>	<b>0.0040</b>	0.1895	0.2969	0.5645	0.6172
MRC	0.3673	0.4546	0.3652	0.4688	<b>0.0084</b>	<b>0.0076</b>	0.1641	0.2734	0.6133	0.6797
MEO	0.3608	0.4639	0.3633	0.4492	<b>0.0060</b>	<b>0.0059</b>	0.1738	0.3203	0.5332	0.6797
MUV	0.3487	0.4725	0.3672	0.4961	<b>0.0093</b>	<b>0.0072</b>	0.1582	0.2715	0.5898	0.6758
RWE	0.3550	0.4375	0.3555	0.4277	0.0034	0.0067	0.1953	0.2969	0.4902	0.6699
SZG	0.3452	0.4346	0.3398	0.4395	0.0052	0.0055	0.1914	0.2773	0.5586	0.5957
SAP	0.3584	0.4907	0.3535	0.4883	<b>0.0094</b>	<b>0.0092</b>	0.1973	0.3164	0.6328	0.7422
SIE	0.3304	0.4314	0.3066	0.4219	<b>0.0055</b>	<b>0.0032</b>	0.2070	0.2988	0.4922	0.5547
TKA	0.3441	0.4232	0.3320	0.4082	0.0054	0.0058	0.1641	0.2559	0.5488	0.6406
VOW	0.3620	0.4551	0.3633	0.4707	<b>0.0069</b>	<b>0.0059</b>	0.1367	0.2988	0.5313	0.5977

Table A.4.: Cramér-von Mises distance statistics,  $\kappa = 4$ 

	mean <sub>L</sub>	mean <sub>M</sub>	median <sub>L</sub>	median <sub>M</sub>	var <sub>L</sub>	var <sub>M</sub>	min <sub>L</sub>	min <sub>M</sub>	max <sub>L</sub>	max <sub>M</sub>
ADS	1.9900	2.9263	1.9240	2.7096	1.3150	2.0979	0.4096	0.9308	4.6052	8.9510
ALV	2.2924	3.7933	2.0247	3.4767	1.0203	2.3436	0.9035	1.4396	6.5855	10.0982
BAS	3.0622	5.1559	2.7577	4.2652	3.7797	9.7155	0.5551	1.1444	8.0198	11.8742
BAY	2.0292	3.6645	1.7201	3.1484	2.0263	3.2628	0.1698	0.8734	5.8303	7.9427
BEI	2.9864	4.7521	2.6469	4.0014	2.8104	7.8879	0.2663	0.9055	7.7610	11.5922
BMW	3.2591	4.9492	2.9248	4.6733	3.1040	4.6457	0.9487	1.6839	10.5268	12.2826
CBK	1.9459	2.6666	1.9292	2.7669	<b>0.8374</b>	<b>0.7738</b>	0.3393	0.8373	4.4482	4.6790
DAI	2.3984	3.9715	2.3124	3.8662	0.6146	1.5491	0.9151	1.8636	5.0276	7.7329
DBK	1.9260	2.7193	1.9491	2.6215	0.7517	0.9334	0.2702	0.7075	3.6061	4.5656
DB1	3.2885	4.7827	3.0085	3.5228	3.9448	8.4037	0.6810	1.1609	8.6280	10.8132
DPB	2.3378	3.4329	2.0212	3.1424	2.0061	2.7053	0.3698	0.9812	6.3294	7.2588
DPW	1.6011	2.8359	1.4346	2.8982	0.6808	1.3990	0.2179	0.6362	4.8430	6.5450
DTE	3.5237	4.6729	2.4662	3.4324	7.5915	8.8771	0.3016	0.8445	9.5931	10.9201
EPC	3.9282	5.8675	3.4159	4.9286	6.7179	12.7121	0.4060	1.1332	12.0084	15.4798
FME	3.0452	4.4035	2.2603	3.3016	5.0773	9.8786	0.2566	1.1002	9.8324	15.3188
LHA	2.4325	3.6323	2.3730	3.4537	1.2788	2.9123	0.7199	1.0579	6.8867	8.0270
HNK	1.8603	3.1743	1.5039	2.3163	1.6125	4.5147	0.2849	0.5122	5.5362	8.5479
IFX	1.5997	2.6832	1.4412	2.4124	0.8113	1.2863	0.2677	0.7840	3.5972	5.1547
SDF	2.8674	4.9245	2.4860	3.3857	2.7092	14.8020	0.8045	1.4769	9.0228	16.4116
LIN	3.1025	4.1409	3.0251	4.3117	1.5734	1.5906	0.8326	1.3577	6.5804	6.7663
MAN	2.4841	3.3613	2.4237	3.4183	1.6634	1.9588	0.3353	0.9861	6.3633	8.1436
MRC	2.8571	4.1293	2.4160	3.8410	3.6555	4.7198	0.3085	0.7492	9.2614	9.9076
MEO	2.5108	4.1335	2.1746	3.2182	1.7954	4.3799	0.7795	1.4692	6.7141	10.1256
MUV	2.9740	4.5075	2.6191	4.4043	3.9772	5.9180	0.3625	0.8194	9.7890	10.3614
RWE	2.6080	3.5067	2.3967	2.9646	1.9630	5.1336	0.3348	0.8611	6.3691	10.0379
SZG	2.0581	3.4635	1.9780	3.4403	1.2851	2.9043	0.3936	0.8300	6.0652	8.6990
SAP	2.8540	5.3589	2.3501	5.2255	3.1035	5.6411	0.2384	0.9699	7.7538	10.8198
SIE	1.9888	3.3271	1.8373	3.2710	1.0886	1.9108	0.5235	1.2356	4.5689	6.3632
TKA	2.1740	3.3279	2.0225	2.8800	1.4548	3.1846	0.2703	0.8359	5.3359	8.2879
VOW	3.1540	4.8168	3.0321	4.6548	2.7556	4.4986	0.4879	1.4977	8.0505	10.6447

Table A.5.: Kolmogorov-Smirnov distance statistics,  $\kappa = 10$ 

	$KS_{\text{mean}}^{\text{LLSA}}$	$KS_{\text{mean}}^{\text{MODWT}}$	$KS_{\text{med}}^{\text{LLSA}}$	$KS_{\text{med}}^{\text{MODWT}}$	$KS_{\text{var}}^{\text{LLSA}}$	$KS_{\text{var}}^{\text{MODWT}}$	$KS_{\text{min}}^{\text{LLSA}}$	$KS_{\text{min}}^{\text{MODWT}}$	$KS_{\text{max}}^{\text{LLSA}}$	$KS_{\text{max}}^{\text{MODWT}}$
ADS	0.1122	0.1159	0.1090	0.1094	0.0005	0.0005	0.0797	0.0797	0.1719	0.1766
ALV	0.1299	0.1421	0.1289	0.1406	0.0003	0.0004	0.0844	0.1031	0.1773	0.1781
BAS	0.1010	0.1224	0.0992	0.1219	<b>0.0002</b>	<b>0.0001</b>	0.0820	0.0992	0.1430	0.1539
BAY	0.1539	0.1664	0.1617	0.1656	0.0006	0.0008	<b>0.0992</b>	<b>0.0883</b>	0.2000	0.2148
BEI	0.1350	0.1567	0.1297	0.1656	<b>0.0007</b>	<b>0.0003</b>	0.1063	0.1086	0.1773	0.1773
BMW	<b>0.1494</b>	<b>0.1458</b>	0.1410	0.1414	0.0004	0.0012	<b>0.1266</b>	<b>0.0922</b>	0.2164	0.2430
CBK	0.1089	0.1193	0.1070	0.1172	0.0005	0.0005	0.0766	0.0867	0.1711	0.1789
DAI	0.1232	0.1360	0.1266	0.1281	<b>0.0004</b>	<b>0.0003</b>	0.0930	0.1094	0.2008	0.2070
DBK	0.1217	0.1258	0.1203	0.1211	0.0005	0.0005	0.0711	0.0766	0.1906	0.1906
DB1	0.1303	0.1365	0.1320	0.1336	0.0003	0.0005	0.0953	0.0977	0.1859	0.1859
DPB	0.1051	0.1101	0.0961	0.1023	<b>0.0005</b>	<b>0.0005</b>	0.0687	0.0805	0.1680	0.1680
DPW	0.1195	0.1200	0.1188	0.1211	0.0002	0.0005	0.0914	0.0945	0.1539	0.1898
DTE	0.1559	0.1596	0.1594	0.1602	<b>0.0026</b>	<b>0.0024</b>	0.0953	0.0953	0.2508	0.2508
EPC	0.1305	0.1395	0.1266	0.1367	0.0003	0.0003	0.1078	0.1078	0.1734	0.1758
FME	0.1216	0.1226	<b>0.1227</b>	<b>0.1180</b>	<b>0.0003</b>	<b>0.0003</b>	0.0805	0.0930	0.1891	0.1891
LHA	<b>0.1380</b>	<b>0.1338</b>	<b>0.1391</b>	<b>0.1301</b>	0.0009	0.0012	<b>0.0859</b>	<b>0.0828</b>	0.2180	0.2180
HNK	0.1231	0.1458	0.1141	0.1469	<b>0.0008</b>	<b>0.0006</b>	0.0813	0.0859	0.1883	0.2016
IFX	0.1180	0.1251	0.1086	0.1086	0.0005	0.0007	0.0953	0.0953	0.1914	0.1914
SDF	0.0986	0.1271	0.0977	0.1289	0.0003	0.0004	0.0797	0.0797	0.1414	0.1781
LIN	0.1084	0.1123	0.1094	0.1094	<b>0.0002</b>	<b>0.0001</b>	0.0867	0.0930	0.1539	0.1539
MAN	<b>0.1000</b>	<b>0.0998</b>	<b>0.0945</b>	<b>0.0938</b>	<b>0.0003</b>	<b>0.0003</b>	0.0773	0.0773	0.1523	0.1523
MRC	0.1283	0.1292	0.1281	0.1281	<b>0.0002</b>	<b>0.0002</b>	0.0812	0.0813	0.1727	0.1727
MEO	<b>0.1413</b>	<b>0.1405</b>	<b>0.1453</b>	<b>0.1414</b>	0.0005	0.0006	0.0977	0.0977	0.2117	0.2227
MUV	0.1113	0.1126	0.1062	0.1063	0.0004	0.0004	<b>0.0836</b>	<b>0.0797</b>	0.1703	0.1703
RWE	0.1007	0.1036	0.0859	0.0887	<b>0.0010</b>	<b>0.0010</b>	0.0727	0.0727	0.1844	0.1844
SZG	0.1033	0.1080	0.1008	0.1102	0.0002	0.0003	<b>0.0773</b>	<b>0.0742</b>	0.1437	0.1539
SAP	0.1070	0.1273	0.1086	0.1148	<b>0.0011</b>	<b>0.0003</b>	0.0586	0.1086	0.1625	0.1625
SIE	0.1103	0.1427	0.1016	0.1367	0.0005	0.0016	0.0898	0.1016	0.1820	0.2570
TKA	0.1002	0.1051	0.1031	0.1039	0.0001	0.0005	0.0680	0.0695	0.1281	0.1477
VOW	0.1288	0.1315	0.1273	0.1273	0.0003	0.0003	0.0938	0.0938	0.1914	0.1914

Table A.6.: Anderson-Darling distance statistics,  $\kappa = 10$

	$AD_{\text{mean}}^{\text{LLSA}}$	$AD_{\text{mean}}^{\text{MODWT}}$	$AD_{\text{med}}^{\text{LLSA}}$	$AD_{\text{med}}^{\text{MODWT}}$	$AD_{\text{var}}^{\text{LLSA}}$	$AD_{\text{var}}^{\text{MODWT}}$	$AD_{\text{min}}^{\text{LLSA}}$	$AD_{\text{min}}^{\text{MODWT}}$	$AD_{\text{max}}^{\text{LLSA}}$	$AD_{\text{max}}^{\text{MODWT}}$
ADS	2.5760	3.1952	2.3488	3.1177	<b>0.7399</b>	<b>0.6676</b>	1.9014	2.3488	5.3407	5.4525
ALV	3.7212	3.9592	3.6910	3.9146	<b>0.5301</b>	<b>0.4271</b>	2.3208	2.8241	5.7042	5.7601
BAS	3.3448	3.5922	3.3834	3.4393	0.4400	0.5927	2.0132	2.3208	5.0890	5.4805
BAY	4.7216	5.7437	5.3687	5.8999	<b>2.1760</b>	<b>1.2306</b>	2.4886	2.5445	7.0743	7.6615
BEI	4.6340	5.5789	4.4739	5.8999	<b>1.2807</b>	<b>0.4069</b>	2.4606	3.8587	6.3194	6.3194
BMW	4.3886	4.9962	4.2782	4.3341	1.2976	1.7107	2.1531	2.5445	7.4099	8.6682
CBK	3.2847	3.7551	3.0758	3.5511	0.7572	0.8833	2.6004	2.6004	6.0957	6.3753
DAI	3.8982	4.4428	3.8587	4.5578	0.6513	0.8571	2.6284	2.6284	7.1582	7.3819
DBK	3.1774	3.7711	3.0758	3.5791	<b>1.3420</b>	<b>0.9796</b>	1.9014	2.1810	6.7947	6.7947
DB1	3.8932	4.5582	3.6630	4.5857	0.9675	1.1654	2.7403	2.7403	6.6269	6.6269
DPB	3.5326	3.6775	3.2436	3.3834	<b>0.6314</b>	<b>0.4908</b>	2.1810	2.6564	5.9838	5.9838
DPW	3.1674	3.6946	2.9360	3.2156	0.6307	1.2149	2.3488	2.3488	5.3407	6.7667
DTE	3.9811	4.6879	2.8661	3.9426	<b>5.3599</b>	<b>4.4219</b>	2.0132	2.8241	8.9478	8.9478
EPC	3.8125	4.4071	3.7469	4.6556	<b>1.0493</b>	<b>0.8839</b>	2.0132	2.6004	5.8160	6.2634
FME	3.6808	3.9984	3.7748	3.7748	0.5814	0.5967	1.9014	2.6284	6.7388	6.7388
LHA	3.4206	3.6089	2.8801	3.1317	<b>1.3163</b>	<b>1.2273</b>	2.0132	2.3736	7.7734	7.7734
HNK	3.3717	3.8750	3.0199	3.6350	1.3053	1.6451	1.8175	2.1531	6.7108	7.1862
IFX	3.9865	4.3033	3.8587	3.8587	0.7860	0.9203	1.6813	2.5166	6.8227	6.8227
SDF	3.4548	3.8602	3.4113	3.7748	0.3569	0.5850	2.8241	2.8241	5.0331	6.3473
LIN	3.8289	3.9685	3.8867	3.8867	<b>0.2445</b>	<b>0.1387</b>	3.0758	3.1038	5.4805	5.4805
MAN	3.4073	3.4710	3.2995	3.2995	<b>0.4444</b>	<b>0.3846</b>	2.3208	2.7123	5.4246	5.4246
MRC	4.5629	4.5980	4.5578	4.5578	<b>0.2818</b>	<b>0.2457</b>	2.7403	2.8801	6.1516	6.1516
MEO	4.0170	4.2243	3.5232	3.7469	1.2725	1.3644	2.5166	2.5166	7.5497	7.9411
MUV	3.7423	3.8947	3.7189	3.7748	0.4347	0.4806	2.6004	2.6843	5.2568	5.8440
RWE	3.2690	3.3739	2.6843	2.9919	<b>0.7886</b>	<b>0.7311</b>	2.5725	2.5725	6.0677	6.0677
SZG	3.2754	3.5224	3.2995	3.4393	0.4472	0.5663	2.5166	2.5166	5.1170	5.4805
SAP	3.5931	4.5221	3.8587	4.0824	<b>1.7480</b>	<b>0.4245</b>	1.7336	3.8587	5.7881	5.7881
SIE	3.2970	4.9462	3.5232	3.8867	0.4041	2.2411	2.2369	3.3554	5.1450	9.1715
TKA	3.2957	3.7238	3.3275	3.6910	0.3081	0.6615	2.0412	2.4606	4.5578	5.2568
VOW	4.5227	4.6391	4.5298	4.5298	<b>0.4239</b>	<b>0.4125</b>	3.2995	3.3275	6.8227	6.8227

Table A.7.: Kuiper distance statistics,  $\kappa = 10$ 

	$\kappa_{\text{mean}}^{\text{LLSA}}$	$\kappa_{\text{mean}}^{\text{MODWT}}$	$\kappa_{\text{med}}^{\text{LLSA}}$	$\kappa_{\text{med}}^{\text{MODWT}}$	$\kappa_{\text{var}}^{\text{LLSA}}$	$\kappa_{\text{var}}^{\text{MODWT}}$	$\kappa_{\text{min}}^{\text{LLSA}}$	$\kappa_{\text{min}}^{\text{MODWT}}$	$\kappa_{\text{max}}^{\text{LLSA}}$	$\kappa_{\text{max}}^{\text{MODWT}}$
ADS	0.2038	0.2151	0.1977	0.2070	0.0013	0.0016	<b>0.1500</b>	<b>0.1492</b>	0.3055	0.3164
ALV	0.2243	0.2404	0.2180	0.2332	<b>0.0006</b>	<b>0.0005</b>	0.1656	0.2055	<b>0.3367</b>	<b>0.3023</b>
BAS	0.1823	0.2280	0.1820	0.2211	0.0003	0.0005	0.1523	0.1875	0.2273	0.2969
BAY	0.2538	0.2634	0.2625	0.2680	0.0013	0.0014	<b>0.1719</b>	<b>0.1625</b>	0.3242	0.3391
BEI	0.2249	0.2468	0.2125	0.2539	<b>0.0011</b>	<b>0.0006</b>	0.1742	0.1992	0.2781	0.2781
BMW	<b>0.2561</b>	<b>0.2509</b>	0.2547	0.2656	0.0005	0.0019	<b>0.2156</b>	<b>0.1836</b>	0.3117	0.3445
CBK	0.1900	0.2106	0.1895	0.2141	<b>0.0009</b>	<b>0.0009</b>	0.1375	0.1508	0.2742	0.2883
DAI	0.2270	0.2429	<b>0.2430</b>	<b>0.2383</b>	<b>0.0013</b>	<b>0.0007</b>	0.1648	0.1992	<b>0.3242</b>	<b>0.3219</b>
DBK	0.1977	0.2041	0.2031	0.2070	<b>0.0007</b>	<b>0.0006</b>	0.1297	0.1477	0.2578	0.2578
DB1	0.2218	0.2323	0.2191	0.2258	0.0007	0.0010	0.1672	0.1781	0.2875	0.2875
DPB	0.1875	0.1977	0.1781	0.1867	<b>0.0010</b>	<b>0.0010</b>	0.1367	0.1516	0.2547	0.2891
DPW	0.1964	0.2126	0.1930	0.2039	0.0002	0.0007	0.1680	0.1883	0.2414	0.2852
DTE	0.2720	0.2860	0.3000	0.3102	<b>0.0076</b>	<b>0.0071</b>	0.1406	0.1664	0.4172	0.4297
EPC	0.2279	0.2483	0.2242	0.2574	<b>0.0019</b>	<b>0.0018</b>	0.1656	0.1836	0.3133	0.3195
FME	0.2190	0.2229	<b>0.2211</b>	<b>0.2191</b>	<b>0.0005</b>	<b>0.0004</b>	0.1547	0.1625	<b>0.2797</b>	<b>0.2734</b>
LHA	0.2205	0.2219	0.2281	0.2344	0.0009	0.0016	<b>0.1672</b>	<b>0.1570</b>	0.2805	0.2820
HNK	0.2189	0.2553	0.1852	0.2523	<b>0.0032</b>	<b>0.0019</b>	0.1555	0.1648	0.3438	0.3563
IFX	0.2115	0.2190	0.2039	0.2039	0.0011	0.0013	0.1531	0.1531	0.3000	0.3000
SDF	0.1757	0.2088	0.1773	0.2125	0.0004	0.0005	0.1242	0.1422	0.2211	0.2578
LIN	0.1871	0.2016	0.1836	0.2016	<b>0.0006</b>	<b>0.0005</b>	0.1477	0.1602	0.2492	0.2492
MAN	0.1679	0.1806	0.1641	0.1742	0.0004	0.0004	0.1336	0.1539	0.2336	0.2484
MRC	0.2214	0.2241	0.2078	0.2164	0.0012	0.0012	0.1563	0.1594	0.3281	0.3281
MEO	0.2401	0.2584	0.2383	0.2719	0.0013	0.0016	0.1797	0.1797	0.3727	0.3727
MUV	<b>0.1964</b>	<b>0.1959</b>	<b>0.1937</b>	<b>0.1867</b>	0.0011	0.0014	<b>0.1453</b>	<b>0.1422</b>	0.2914	0.3344
RWE	0.1699	0.1852	0.1598	0.1672	<b>0.0020</b>	<b>0.0016</b>	0.1102	0.1406	0.2656	0.2758
SZG	0.1836	0.1938	0.1773	0.1945	0.0005	0.0005	<b>0.1484</b>	<b>0.1461</b>	0.2414	0.2414
SAP	0.1800	0.2251	0.1711	0.2207	<b>0.0023</b>	<b>0.0010</b>	0.1141	0.1656	0.2734	0.2844
SIE	0.1901	0.2352	0.1773	0.2414	0.0015	0.0034	0.1328	0.1625	0.3117	0.4094
TKA	0.1767	0.1831	0.1828	0.1883	0.0004	0.0006	0.1273	0.1297	<b>0.2234</b>	<b>0.2211</b>
VOW	0.2038	0.2105	0.1996	0.2141	0.0014	0.0015	0.1242	0.1344	0.2898	0.3188

Table A.8.: Cramér-von Mises distance statistics,  $\kappa = 10$ 

	$CVM_{\text{mean}}^{\text{LLSA}}$	$CVM_{\text{mean}}^{\text{MODWT}}$	$CVM_{\text{med}}^{\text{LLSA}}$	$CVM_{\text{med}}^{\text{MODWT}}$	$CVM_{\text{var}}^{\text{LLSA}}$	$CVM_{\text{var}}^{\text{MODWT}}$	$CVM_{\text{min}}^{\text{LLSA}}$	$CVM_{\text{min}}^{\text{MODWT}}$	$CVM_{\text{max}}^{\text{LLSA}}$	$CVM_{\text{max}}^{\text{MODWT}}$
ADS	1.6023	1.8725	1.3734	1.5451	0.5301	0.9003	<b>0.8603</b>	<b>0.8175</b>	3.4871	4.1002
ALV	2.2108	2.6050	2.1495	2.5484	0.3016	0.3432	1.5070	1.7115	3.9617	4.0419
BAS	1.4007	2.4928	1.2990	2.2356	0.1317	0.5465	0.9156	1.4335	2.3814	4.4473
BAY	2.9299	3.5374	3.0868	3.6893	0.6462	0.9362	1.0460	1.1650	4.0542	5.0689
BEI	1.5425	1.8889	1.4240	1.8155	<b>0.2619</b>	<b>0.2088</b>	0.8271	1.0070	2.3981	2.5958
BMW	2.4866	2.6261	<b>2.0654</b>	<b>1.9039</b>	0.6870	1.5262	1.5098	1.5445	5.0463	5.7958
CBK	1.5560	2.0001	1.4035	2.0610	0.3130	0.3450	0.6735	0.9874	2.4977	2.9621
DAI	1.8386	1.9640	<b>2.0092</b>	<b>1.9510</b>	<b>0.3907</b>	<b>0.3268</b>	0.6759	0.9955	<b>3.7742</b>	<b>3.5241</b>
DBK	1.6524	1.9174	1.6588	1.7513	0.2158	0.2637	0.7157	1.0852	<b>3.0638</b>	<b>3.0334</b>
DB1	1.9844	2.0972	1.9403	2.1657	<b>0.2236</b>	<b>0.2202</b>	0.9018	1.1662	<b>3.2791</b>	<b>3.2773</b>
DPB	1.4584	1.7228	1.2278	1.3826	0.3540	0.6580	0.7618	0.9326	3.2430	5.4548
DPW	1.2689	1.4792	1.1615	1.3727	<b>0.0833</b>	<b>0.0830</b>	0.7725	0.9956	<b>2.3754</b>	<b>2.0991</b>
DTE	4.0107	4.6933	4.1092	4.8525	8.2181	8.6321	0.8881	1.2169	9.9054	10.5692
EPC	2.2320	2.7803	1.8415	2.3784	0.6069	0.8972	1.4478	1.6411	4.0743	4.8274
FME	1.9651	2.3324	1.9096	2.2998	<b>0.2638</b>	<b>0.2501</b>	0.6542	0.9440	3.1199	3.3070
LHA	2.0948	2.1261	<b>2.3686</b>	<b>2.3334</b>	0.6858	0.9808	<b>0.7587</b>	<b>0.5179</b>	3.7214	4.0386
HNK	1.9493	2.4047	1.6330	2.4064	0.5413	0.5487	0.6396	0.8237	3.3423	3.7620
IFX	1.3766	2.0177	1.1767	1.7449	0.4594	0.5460	0.4566	0.8796	2.8289	3.4133
SDF	1.2396	2.1715	1.2235	2.2826	0.3865	0.6565	0.4014	0.6637	2.2590	3.3071
LIN	1.4346	1.6300	1.4157	1.5258	0.1201	0.1297	0.8874	1.1051	2.0593	2.7268
MAN	0.9525	1.0180	0.8730	0.9292	0.1014	0.1051	0.4752	0.6922	2.2755	2.5951
MRC	<b>1.7641</b>	<b>1.7009</b>	<b>1.5254</b>	<b>1.4165</b>	0.4146	0.7344	<b>1.1562</b>	<b>1.0489</b>	4.2429	4.8630
MEO	1.9812	2.3684	2.0327	2.7488	0.2680	0.6628	<b>1.1732</b>	<b>1.1227</b>	3.8870	4.4996
MUV	1.3975	1.5217	<b>1.2283</b>	<b>1.2181</b>	0.6161	0.9613	0.4821	0.6354	4.2219	4.8244
RWE	1.3506	1.4231	0.9888	1.1321	<b>0.8387</b>	<b>0.8155</b>	0.2019	0.3365	3.0602	3.2511
SZG	1.0258	1.2044	0.9489	1.1415	0.0933	0.1210	0.5076	0.6948	1.6018	1.8991
SAP	1.3345	1.9280	1.2724	1.8617	<b>0.5237</b>	<b>0.3715</b>	0.4621	0.9814	3.1157	3.2203
SIE	1.8317	2.7669	1.4351	2.4768	1.1815	3.5547	0.7454	0.9277	5.1597	8.7686
TKA	0.9706	1.0649	0.9123	1.1290	0.0391	0.1239	<b>0.6076</b>	<b>0.5169</b>	1.5593	1.7836
VOW	1.8550	1.9917	1.8337	1.8789	0.2643	0.4745	0.8096	0.8870	3.2854	4.4124

## **A.2. Bootstrap Confidence Interval Tables**

The following tables state the bootstrap confidence intervals for the mean difference of the LLSA and MODWT result series of Appendix A.1. The 99% confidence intervals are stated, that is,  $\alpha = 0.01$ , achieved by 50000 bootstrapping samples. The tables are stated for all moving window sizes, i. e.,  $\kappa = 4$  to  $\kappa = 10$ .

Table A.9.: Mean difference bootstrap confidence intervals,  $\kappa = 4$ )

	KS	AD	K	CVM
ADS	(-0.0298, -0.0259)	(-0.7036, -0.6086)	(-0.0765, -0.0711)	(-0.9844, -0.8931)
ALV	(-0.0537, -0.0487)	(-1.4060, -1.2666)	(-0.1151, -0.1071)	(-1.5600, -1.4428)
BAS	(-0.0470, -0.0424)	(-0.9885, -0.8635)	(-0.0906, -0.0837)	(-2.1967, -1.9967)
BAY	(-0.0735, -0.0678)	(-1.6909, -1.5610)	(-0.1479, -0.1402)	(-1.6827, -1.5878)
BEI	(-0.0660, -0.0591)	(-1.9035, -1.6879)	(-0.1187, -0.1090)	(-1.8538, -1.6795)
BMW	(-0.0655, -0.0608)	(-1.8675, -1.7225)	(-0.1211, -0.1129)	(-1.7558, -1.6247)
CBK	(-0.0237, -0.0204)	(-0.5435, -0.4668)	(-0.0751, -0.0695)	(-0.7497, -0.6927)
DAI	(-0.0567, -0.0523)	(-1.3783, -1.2667)	(-0.1185, -0.1107)	(-1.6184, -1.5288)
DBK	(-0.0287, -0.0255)	(-0.6795, -0.6050)	(-0.0705, -0.0655)	(-0.8257, -0.7601)
DBI	(-0.0518, -0.0470)	(-1.2675, -1.1428)	(-0.1062, -0.1004)	(-1.5713, -1.4201)
DPB	(-0.0412, -0.0369)	(-0.9901, -0.8798)	(-0.0928, -0.0878)	(-1.1314, -1.0611)
DPW	(-0.0599, -0.0566)	(-1.4625, -1.3708)	(-0.1124, -0.1079)	(-1.2736, -1.1959)
DTE	(-0.0383, -0.0341)	(-0.9206, -0.8134)	(-0.0950, -0.0882)	(-1.2007, -1.1007)
EPC	(-0.0437, -0.0394)	(-1.5305, -1.3708)	(-0.0891, -0.0834)	(-2.0126, -1.8672)
FME	(-0.0454, -0.0402)	(-1.3025, -1.1363)	(-0.1087, -0.1011)	(-1.4496, -1.2726)
LHA	(-0.0408, -0.0373)	(-0.9937, -0.9020)	(-0.0890, -0.0831)	(-1.2612, -1.1389)
HNK	(-0.0788, -0.0733)	(-1.9352, -1.7947)	(-0.1176, -0.1119)	(-1.3773, -1.2528)
IFX	(-0.0497, -0.0443)	(-1.1346, -1.0141)	(-0.1014, -0.0934)	(-1.1262, -1.0411)
SDF	(-0.0534, -0.0480)	(-1.1081, -0.9781)	(-0.0946, -0.0875)	(-2.2143, -1.9124)
LIN	(-0.0268, -0.0229)	(-0.6567, -0.5590)	(-0.0783, -0.0732)	(-1.0861, -0.9918)
MAN	(-0.0298, -0.0257)	(-0.6912, -0.5961)	(-0.0793, -0.0727)	(-0.9229, -0.8314)
MRC	(-0.0412, -0.0362)	(-0.9811, -0.8609)	(-0.0908, -0.0839)	(-1.3315, -1.2163)
MEO	(-0.0654, -0.0596)	(-1.7990, -1.6270)	(-0.1065, -0.0996)	(-1.6897, -1.5593)
MUV	(-0.0590, -0.0524)	(-1.5614, -1.3704)	(-0.1279, -0.1196)	(-1.6180, -1.4519)
RWE	(-0.0372, -0.0314)	(-1.0381, -0.8645)	(-0.0867, -0.0784)	(-0.9772, -0.8260)
SZG	(-0.0480, -0.0437)	(-1.1378, -1.0325)	(-0.0924, -0.0865)	(-1.4562, -1.3557)
SAP	(-0.0727, -0.0683)	(-1.7971, -1.6674)	(-0.1355, -0.1292)	(-2.5810, -2.4284)
SIE	(-0.0496, -0.0445)	(-1.2704, -1.1332)	(-0.1052, -0.0967)	(-1.3889, -1.2898)
TKA	(-0.0373, -0.0335)	(-0.8787, -0.7844)	(-0.0819, -0.0763)	(-1.2057, -1.1037)
VOW	(-0.0511, -0.0449)	(-1.4658, -1.2736)	(-0.0981, -0.0880)	(-1.7342, -1.5900)

Table A.10.: Mean difference bootstrap confidence intervals,  $\kappa = 5$ )

	KS	AD	K	CVM
ADS	(-0.0215, -0.0187)	(-0.6499, -0.5657)	(-0.0452, -0.0399)	(-0.7247, -0.6382)
ALV	(-0.0422, -0.0380)	(-1.1755, -1.0534)	(-0.0884, -0.0824)	(-1.2242, -1.1211)
BAS	(-0.0355, -0.0317)	(-0.8064, -0.6904)	(-0.0716, -0.0655)	(-1.7113, -1.5522)
BAY	(-0.0601, -0.0547)	(-1.7110, -1.5488)	(-0.1206, -0.1128)	(-1.6641, -1.5555)
BEI	(-0.0499, -0.0441)	(-1.8136, -1.5901)	(-0.0781, -0.0703)	(-1.7182, -1.5470)
BMW	(-0.0378, -0.0344)	(-1.2419, -1.1105)	(-0.0774, -0.0709)	(-1.1095, -1.0076)
CBK	(-0.0140, -0.0109)	(-0.3969, -0.3179)	(-0.0509, -0.0467)	(-0.5055, -0.4378)
DAI	(-0.0389, -0.0350)	(-1.0367, -0.9243)	(-0.0779, -0.0722)	(-1.2236, -1.1227)
DBK	(-0.0217, -0.0182)	(-0.6069, -0.5253)	(-0.0479, -0.0427)	(-0.6604, -0.6023)
DBI	(-0.0295, -0.0258)	(-0.9655, -0.8482)	(-0.0567, -0.0509)	(-1.0137, -0.9085)
DPB	(-0.0359, -0.0316)	(-0.9814, -0.8590)	(-0.0686, -0.0639)	(-0.9787, -0.8987)
DPW	(-0.0536, -0.0513)	(-1.5345, -1.4505)	(-0.0912, -0.0877)	(-1.1341, -1.0456)
DTE	(-0.0238, -0.0203)	(-0.6101, -0.5222)	(-0.0622, -0.0556)	(-0.6990, -0.6009)
EPC	(-0.0231, -0.0196)	(-0.9809, -0.8293)	(-0.0539, -0.0488)	(-1.4818, -1.3664)
FME	(-0.0288, -0.0248)	(-0.9682, -0.8189)	(-0.0712, -0.0651)	(-0.9176, -0.7961)
LHA	(-0.0272, -0.0241)	(-0.7931, -0.7011)	(-0.0482, -0.0436)	(-0.7834, -0.7094)
HNK	(-0.0648, -0.0604)	(-1.9274, -1.8018)	(-0.0885, -0.0840)	(-1.0883, -1.0101)
IFX	(-0.0351, -0.0307)	(-0.9384, -0.8179)	(-0.0883, -0.0823)	(-0.8487, -0.7926)
SDF	(-0.0416, -0.0370)	(-0.8314, -0.7191)	(-0.0692, -0.0631)	(-1.8279, -1.5895)
LIN	(-0.0187, -0.0159)	(-0.5293, -0.4438)	(-0.0537, -0.0488)	(-0.6891, -0.6240)
MAN	(-0.0194, -0.0156)	(-0.6115, -0.5048)	(-0.0586, -0.0532)	(-0.6533, -0.5811)
MRC	(-0.0275, -0.0236)	(-0.7255, -0.6191)	(-0.0562, -0.0506)	(-0.7538, -0.6907)
MEO	(-0.0463, -0.0417)	(-1.6868, -1.4852)	(-0.0774, -0.0707)	(-1.3028, -1.1596)
MUV	(-0.0363, -0.0311)	(-1.1307, -0.9611)	(-0.0810, -0.0732)	(-1.0218, -0.9040)
RWE	(-0.0108, -0.0079)	(-0.4422, -0.3279)	(-0.0326, -0.0274)	(-0.4358, -0.3188)
SZG	(-0.0298, -0.0267)	(-0.7927, -0.7080)	(-0.0600, -0.0556)	(-1.0025, -0.9292)
SAP	(-0.0551, -0.0502)	(-1.5135, -1.3710)	(-0.1018, -0.0945)	(-2.0300, -1.9004)
SIE	(-0.0337, -0.0299)	(-1.0075, -0.8894)	(-0.0715, -0.0650)	(-1.1262, -1.0343)
TKA	(-0.0290, -0.0257)	(-0.7751, -0.6848)	(-0.0627, -0.0585)	(-0.8768, -0.8102)
VOW	(-0.0357, -0.0302)	(-1.1604, -0.9767)	(-0.0668, -0.0576)	(-1.0404, -0.9221)

Table A.11.: Mean difference bootstrap confidence intervals,  $\kappa = 6$ )

	KS	AD	K	CVM
ADS	(-0.0181, -0.0156)	(-0.5750, -0.4970)	(-0.0362, -0.0305)	(-0.6214, -0.5385)
ALV	(-0.0319, -0.0287)	(-0.8840, -0.7853)	(-0.0694, -0.0641)	(-0.9610, -0.8873)
BAS	(-0.0246, -0.0212)	(-0.5973, -0.4898)	(-0.0582, -0.0522)	(-1.4226, -1.2868)
BAY	(-0.0458, -0.0413)	(-1.7433, -1.5603)	(-0.0824, -0.0762)	(-1.4612, -1.3398)
BEI	(-0.0419, -0.0366)	(-1.9304, -1.6875)	(-0.0721, -0.0628)	(-1.4247, -1.2825)
BMW	(-0.0236, -0.0204)	(-0.8729, -0.7558)	(-0.0480, -0.0419)	(-0.7831, -0.6629)
CBK	(-0.0075, -0.0059)	(-0.2420, -0.2030)	(-0.0376, -0.0344)	(-0.4019, -0.3641)
DAI	(-0.0280, -0.0247)	(-0.7601, -0.6625)	(-0.0513, -0.0468)	(-0.8418, -0.7672)
DBK	(-0.0146, -0.0124)	(-0.5020, -0.4360)	(-0.0358, -0.0317)	(-0.6787, -0.6070)
DB1	(-0.0149, -0.0113)	(-0.6205, -0.5041)	(-0.0408, -0.0350)	(-0.5648, -0.4776)
DPB	(-0.0218, -0.0187)	(-0.6662, -0.5531)	(-0.0461, -0.0413)	(-0.8594, -0.7603)
DPW	(-0.0424, -0.0398)	(-1.3555, -1.2742)	(-0.0671, -0.0632)	(-0.8988, -0.8287)
DTE	(-0.0218, -0.0187)	(-0.6066, -0.5129)	(-0.0407, -0.0341)	(-0.4730, -0.3821)
EPC	(-0.0165, -0.0136)	(-0.7733, -0.6442)	(-0.0401, -0.0359)	(-1.0703, -0.9987)
FME	(-0.0114, -0.0087)	(-0.5141, -0.3984)	(-0.0317, -0.0269)	(-0.4491, -0.3541)
LHA	(-0.0154, -0.0124)	(-0.5493, -0.4620)	(-0.0249, -0.0209)	(-0.4914, -0.4185)
HNK	(-0.0513, -0.0467)	(-1.6755, -1.5476)	(-0.0780, -0.0719)	(-0.8435, -0.7662)
IFX	(-0.0267, -0.0235)	(-0.9673, -0.8380)	(-0.0629, -0.0580)	(-0.8101, -0.7232)
SDF	(-0.0379, -0.0338)	(-0.7539, -0.6438)	(-0.0592, -0.0539)	(-1.5847, -1.4000)
LIN	(-0.0134, -0.0109)	(-0.4051, -0.3318)	(-0.0419, -0.0375)	(-0.5210, -0.4751)
MAN	(-0.0155, -0.0127)	(-0.5768, -0.4785)	(-0.0440, -0.0402)	(-0.5263, -0.4730)
MRC	(-0.0170, -0.0136)	(-0.5290, -0.4322)	(-0.0384, -0.0328)	(-0.4178, -0.3588)
MEO	(-0.0363, -0.0328)	(-1.6031, -1.3930)	(-0.0599, -0.0544)	(-1.1135, -0.9833)
MUV	(-0.0244, -0.0204)	(-0.9017, -0.7467)	(-0.0512, -0.0442)	(-0.6723, -0.5879)
RWE	(-0.0038, -0.0018)	(-0.2053, -0.1547)	(-0.0177, -0.0142)	(-0.1799, -0.1221)
SZG	(-0.0185, -0.0163)	(-0.5134, -0.4508)	(-0.0375, -0.0341)	(-0.6275, -0.5707)
SAP	(-0.0428, -0.0381)	(-1.3274, -1.1552)	(-0.0766, -0.0684)	(-1.5663, -1.4235)
SIE	(-0.0300, -0.0267)	(-1.1918, -1.0726)	(-0.0568, -0.0514)	(-1.1488, -1.0342)
TKA	(-0.0229, -0.0203)	(-0.6509, -0.5738)	(-0.0485, -0.0446)	(-0.5538, -0.5051)
VOW	(-0.0298, -0.0256)	(-1.0420, -0.8696)	(-0.0472, -0.0397)	(-0.6786, -0.6028)

Table A.12.: Mean difference bootstrap confidence intervals,  $\kappa = 7$ )

	KS	AD	K	CVM
ADS	(-0.0119, -0.0095)	(-0.6228, -0.5231)	(-0.0169, -0.0130)	(-0.4736, -0.4147)
ALV	(-0.0318, -0.0285)	(-1.1493, -1.0108)	(-0.0476, -0.0441)	(-0.8772, -0.8174)
BAS	(-0.0266, -0.0233)	(-0.6160, -0.4979)	(-0.0516, -0.0466)	(-1.2962, -1.1933)
BAY	(-0.0344, -0.0309)	(-1.5354, -1.3662)	(-0.0560, -0.0498)	(-1.3373, -1.2226)
BEI	(-0.0314, -0.0264)	(-1.6704, -1.4113)	(-0.0608, -0.0519)	(-1.0168, -0.8968)
BMW	(-0.0129, -0.0093)	(-0.6700, -0.5705)	(-0.0285, -0.0225)	(-0.5875, -0.4587)
CBK	(-0.0125, -0.0103)	(-0.5355, -0.4400)	(-0.0231, -0.0207)	(-0.5343, -0.4873)
DAI	(-0.0209, -0.0178)	(-0.6495, -0.5599)	(-0.0400, -0.0353)	(-0.5726, -0.5016)
DBK	(-0.0110, -0.0092)	(-0.5651, -0.4727)	(-0.0290, -0.0258)	(-0.6453, -0.5771)
DB1	(-0.0112, -0.0073)	(-0.7732, -0.6374)	(-0.0311, -0.0254)	(-0.3893, -0.3177)
DPB	(-0.0201, -0.0173)	(-0.7835, -0.6570)	(-0.0345, -0.0310)	(-0.7701, -0.6908)
DPW	(-0.0315, -0.0285)	(-1.1344, -1.0305)	(-0.0486, -0.0446)	(-0.6989, -0.6378)
DTE	(-0.0210, -0.0180)	(-0.6936, -0.5716)	(-0.0341, -0.0286)	(-0.6378, -0.5432)
EPC	(-0.0088, -0.0071)	(-0.4668, -0.3829)	(-0.0293, -0.0260)	(-0.8633, -0.8065)
FME	(-0.0087, -0.0063)	(-0.4877, -0.3708)	(-0.0120, -0.0073)	(-0.3282, -0.2344)
LHA	(-0.0068, -0.0040)	(-0.3723, -0.2968)	(-0.0162, -0.0126)	(-0.3322, -0.2661)
HNK	(-0.0380, -0.0335)	(-1.4059, -1.2784)	(-0.0627, -0.0564)	(-0.7763, -0.6923)
IFX	(-0.0256, -0.0219)	(-1.0558, -0.8877)	(-0.0435, -0.0393)	(-0.8003, -0.7143)
SDF	(-0.0338, -0.0298)	(-0.6068, -0.5031)	(-0.0500, -0.0453)	(-1.3781, -1.2317)
LIN	(-0.0163, -0.0140)	(-0.5487, -0.4636)	(-0.0331, -0.0288)	(-0.4197, -0.3793)
MAN	(-0.0100, -0.0083)	(-0.3414, -0.2807)	(-0.0315, -0.0288)	(-0.3269, -0.2969)
MRC	(-0.0142, -0.0111)	(-0.4648, -0.3665)	(-0.0230, -0.0181)	(-0.1848, -0.1275)
MEO	(-0.0292, -0.0257)	(-1.2707, -1.0696)	(-0.0486, -0.0436)	(-0.9271, -0.8128)
MUV	(-0.0190, -0.0155)	(-0.7894, -0.6339)	(-0.0294, -0.0236)	(-0.4381, -0.3681)
RWE	(-0.0118, -0.0089)	(-0.4178, -0.3187)	(-0.0271, -0.0237)	(-0.2871, -0.2356)
SZG	(-0.0170, -0.0144)	(-0.5333, -0.4454)	(-0.0292, -0.0257)	(-0.5151, -0.4648)
SAP	(-0.0272, -0.0232)	(-1.0127, -0.8443)	(-0.0600, -0.0528)	(-1.1021, -0.9996)
SIE	(-0.0272, -0.0237)	(-1.4204, -1.2517)	(-0.0509, -0.0457)	(-1.2821, -1.1430)
TKA	(-0.0207, -0.0175)	(-0.6375, -0.5288)	(-0.0366, -0.0328)	(-0.4169, -0.3803)
VOW	(-0.0243, -0.0205)	(-0.9068, -0.7407)	(-0.0419, -0.0353)	(-0.6004, -0.5264)

Table A.13.: Mean difference bootstrap confidence intervals,  $\kappa = 8$ )

	KS	AD	K	CVM
ADS	(-0.0111, -0.0085)	(-0.7221, -0.5980)	(-0.0134, -0.0101)	(-0.3891, -0.3373)
ALV	(-0.0237, -0.0206)	(-0.9036, -0.7564)	(-0.0329, -0.0297)	(-0.7475, -0.6874)
BAS	(-0.0296, -0.0268)	(-0.5662, -0.4543)	(-0.0491, -0.0448)	(-1.2263, -1.1404)
BAY	(-0.0336, -0.0304)	(-1.5848, -1.3869)	(-0.0446, -0.0396)	(-1.2180, -1.1154)
BEI	(-0.0248, -0.0202)	(-1.2109, -0.9787)	(-0.0402, -0.0330)	(-0.6589, -0.5673)
BMW	(-0.0017, <b>0.0025</b> )	(-0.5353, -0.4349)	(-0.0160, -0.0101)	(-0.4210, -0.2908)
CBK	(-0.0136, -0.0114)	(-0.6639, -0.5578)	(-0.0236, -0.0209)	(-0.5982, -0.5438)
DAI	(-0.0173, -0.0144)	(-0.5840, -0.4953)	(-0.0314, -0.0271)	(-0.3777, -0.3205)
DBK	(-0.0067, -0.0046)	(-0.5005, -0.4114)	(-0.0199, -0.0165)	(-0.4919, -0.4171)
DB1	(-0.0060, -0.0021)	(-0.6836, -0.5670)	(-0.0231, -0.0178)	(-0.2776, -0.2087)
DPB	(-0.0121, -0.0101)	(-0.4548, -0.3776)	(-0.0192, -0.0165)	(-0.5230, -0.4646)
DPW	(-0.0221, -0.0189)	(-0.8599, -0.7484)	(-0.0333, -0.0288)	(-0.4809, -0.4363)
DTE	(-0.0107, -0.0084)	(-0.7266, -0.5857)	(-0.0272, -0.0234)	(-0.8717, -0.8109)
EPC	(-0.0072, -0.0056)	(-0.4965, -0.4007)	(-0.0211, -0.0174)	(-0.7067, -0.6540)
FME	(-0.0032, -0.0007)	(-0.3934, -0.2754)	<b>(0.0027, 0.0066)</b>	(-0.3122, -0.2207)
LHA	<b>(0.0003, 0.0027)</b>	(-0.2373, -0.1758)	(-0.0061, -0.0028)	(-0.1968, -0.1268)
HNK	(-0.0338, -0.0299)	(-1.1651, -1.0400)	(-0.0508, -0.0450)	(-0.6753, -0.6088)
IFX	(-0.0198, -0.0167)	(-0.9284, -0.7611)	(-0.0300, -0.0268)	(-0.9710, -0.9039)
SDF	(-0.0322, -0.0287)	(-0.4709, -0.3659)	(-0.0409, -0.0368)	(-1.2378, -1.1255)
LIN	(-0.0101, -0.0083)	(-0.3534, -0.2928)	(-0.0235, -0.0208)	(-0.3235, -0.2898)
MAN	(-0.0070, -0.0055)	(-0.2709, -0.2166)	(-0.0203, -0.0177)	(-0.1835, -0.1589)
MRC	(-0.0071, -0.0050)	(-0.2520, -0.1797)	(-0.0087, -0.0052)	(-0.0162, <b>0.0292</b> )
MEO	(-0.0206, -0.0176)	(-0.9618, -0.7676)	(-0.0372, -0.0330)	(-0.7671, -0.6621)
MUV	(-0.0108, -0.0080)	(-0.4921, -0.3529)	(-0.0163, -0.0117)	(-0.3081, -0.2420)
RWE	(-0.0140, -0.0115)	(-0.3893, -0.2962)	(-0.0306, -0.0276)	(-0.3478, -0.3070)
SZG	(-0.0177, -0.0147)	(-0.6064, -0.5120)	(-0.0258, -0.0218)	(-0.4178, -0.3686)
SAP	(-0.0231, -0.0195)	(-1.0911, -0.9119)	(-0.0536, -0.0478)	(-0.8317, -0.7590)
SIE	(-0.0306, -0.0263)	(-1.5726, -1.3678)	(-0.0458, -0.0410)	(-1.1055, -0.9762)
TKA	(-0.0122, -0.0099)	(-0.4235, -0.3440)	(-0.0218, -0.0187)	(-0.2666, -0.2374)
VOW	(-0.0159, -0.0129)	(-0.6098, -0.4720)	(-0.0316, -0.0259)	(-0.4701, -0.4056)

Table A.14.: Mean difference bootstrap confidence intervals,  $\kappa = 9$ )

	KS	AD	K	CVM
ADS	(-0.0076, -0.0055)	(-0.6659, -0.5549)	(-0.0130, -0.0102)	(-0.3944, -0.3217)
ALV	(-0.0166, -0.0135)	(-0.6087, -0.4798)	(-0.0225, -0.0191)	(-0.5863, -0.5266)
BAS	(-0.0283, -0.0254)	(-0.4777, -0.3628)	(-0.0474, -0.0433)	(-1.2281, -1.1375)
BAY	(-0.0262, -0.0224)	(-1.4129, -1.1909)	(-0.0268, -0.0216)	(-0.8877, -0.7919)
BEI	(-0.0228, -0.0183)	(-1.0099, -0.8078)	(-0.0270, -0.0220)	(-0.4464, -0.3881)
BMW	<b>(0.0019, 0.0059)</b>	(-0.5041, -0.4027)	(-0.0055, <b>0.0001</b> )	(-0.2766, -0.1664)
CBK	(-0.0089, -0.0079)	(-0.4350, -0.3696)	(-0.0216, -0.0197)	(-0.5338, -0.4896)
DAI	(-0.0157, -0.0128)	(-0.6429, -0.5570)	(-0.0257, -0.0216)	(-0.2461, -0.1916)
DBK	(-0.0053, -0.0030)	(-0.5673, -0.4591)	(-0.0122, -0.0092)	(-0.3860, -0.3163)
DB1	(-0.0069, -0.0027)	(-0.7425, -0.6256)	(-0.0190, -0.0141)	(-0.2080, -0.1469)
DPB	(-0.0097, -0.0077)	(-0.4209, -0.3119)	(-0.0133, -0.0110)	(-0.4652, -0.3754)
DPW	(-0.0116, -0.0079)	(-0.7202, -0.6072)	(-0.0218, -0.0174)	(-0.3194, -0.2879)
DTE	(-0.0063, -0.0043)	(-0.7999, -0.6559)	(-0.0220, -0.0197)	(-0.7785, -0.7440)
EPC	(-0.0093, -0.0076)	(-0.5012, -0.4082)	(-0.0221, -0.0194)	(-0.6211, -0.5784)
FME	<b>(0.0024, 0.0044)</b>	(-0.2353, -0.1669)	<b>(0.0054, 0.0082)</b>	(-0.3342, -0.2553)
LHA	<b>(0.0031, 0.0054)</b>	(-0.2004, -0.1377)	(-0.0036, -0.0005)	(-0.0860, -0.0360)
HNK	(-0.0278, -0.0240)	(-0.7429, -0.6421)	(-0.0427, -0.0368)	(-0.5420, -0.4925)
IFX	(-0.0154, -0.0125)	(-0.5769, -0.4533)	(-0.0182, -0.0150)	(-0.8526, -0.8029)
SDF	(-0.0315, -0.0283)	(-0.5238, -0.4120)	(-0.0392, -0.0358)	(-1.1033, -1.0235)
LIN	(-0.0040, -0.0027)	(-0.1474, -0.1049)	(-0.0189, -0.0169)	(-0.2103, -0.1890)
MAN	(-0.0040, -0.0027)	(-0.2070, -0.1547)	(-0.0165, -0.0145)	(-0.1217, -0.1007)
MRC	(-0.0019, -0.0013)	(-0.0677, -0.0449)	(-0.0042, -0.0018)	<b>(0.0468, 0.0904)</b>
MEO	(-0.0122, -0.0092)	(-0.7039, -0.5329)	(-0.0290, -0.0257)	(-0.6254, -0.5323)
MUV	(-0.0027, -0.0012)	(-0.1779, -0.1048)	(-0.0035, -0.0004)	(-0.2377, -0.1764)
RWE	(-0.0076, -0.0059)	(-0.2421, -0.1726)	(-0.0210, -0.0185)	(-0.2098, -0.1728)
SZG	(-0.0075, -0.0058)	(-0.3038, -0.2638)	(-0.0129, -0.0106)	(-0.2463, -0.2025)
SAP	(-0.0215, -0.0181)	(-1.0864, -0.9026)	(-0.0485, -0.0437)	(-0.7371, -0.6689)
SIE	(-0.0337, -0.0288)	(-1.7014, -1.4690)	(-0.0494, -0.0433)	(-1.0040, -0.8756)
TKA	(-0.0074, -0.0049)	(-0.4285, -0.3459)	(-0.0114, -0.0077)	(-0.1681, -0.1299)
VOW	(-0.0084, -0.0061)	(-0.3768, -0.2701)	(-0.0179, -0.0137)	(-0.3218, -0.2635)

Table A.15.: Mean difference bootstrap confidence intervals,  $\kappa = 10$ )

	KS	AD	K	CVM
ADS	(-0.0044, -0.0032)	(-0.6755, -0.5674)	(-0.0126, -0.0100)	(-0.2962, -0.2440)
ALV	(-0.0136, -0.0108)	(-0.2770, -0.2043)	(-0.0177, -0.0144)	(-0.4223, -0.3649)
BAS	(-0.0225, -0.0202)	(-0.2928, -0.2077)	(-0.0472, -0.0443)	(-1.1348, -1.0512)
BAY	(-0.0139, -0.0111)	(-1.1433, -0.9061)	(-0.0121, -0.0073)	(-0.6503, -0.5629)
BEI	(-0.0239, -0.0194)	(-1.0453, -0.8487)	(-0.0243, -0.0197)	(-0.3688, -0.3248)
BMW	<b>(0.0018, 0.0052)</b>	(-0.6626, -0.5546)	<b>(0.0026, 0.0078)</b>	(-0.1898, -0.0908)
CBK	(-0.0113, -0.0096)	(-0.5231, -0.4239)	(-0.0217, -0.0195)	(-0.4664, -0.4233)
DAI	(-0.0143, -0.0113)	(-0.5953, -0.4942)	(-0.0178, -0.0141)	(-0.1513, -0.1008)
DBK	(-0.0051, -0.0029)	(-0.6512, -0.5401)	(-0.0078, -0.0050)	(-0.2957, -0.2328)
DB1	(-0.0080, -0.0043)	(-0.7362, -0.5983)	(-0.0128, -0.0081)	(-0.1399, -0.0857)
DPB	(-0.0057, -0.0044)	(-0.1647, -0.1270)	(-0.0113, -0.0093)	(-0.2911, -0.2427)
DPW	(-0.0023, <b>0.0012</b> )	(-0.5807, -0.4758)	(-0.0184, -0.0140)	(-0.2208, -0.1989)
DTE	(-0.0046, -0.0030)	(-0.7758, -0.6406)	(-0.0154, -0.0126)	(-0.7153, -0.6549)
EPC	(-0.0097, -0.0082)	(-0.6603, -0.5336)	(-0.0217, -0.0192)	(-0.5649, -0.5314)
FME	(-0.0021, <b>0.0001</b> )	(-0.3617, -0.2775)	(-0.0053, -0.0024)	(-0.4113, -0.3240)
LHA	<b>(0.0027, 0.0057)</b>	(-0.2242, -0.1575)	(-0.0033, <b>0.0003</b> )	(-0.0620, -0.0001)
HNK	(-0.0244, -0.0210)	(-0.5426, -0.4657)	(-0.0391, -0.0336)	(-0.4749, -0.4347)
IFX	(-0.0082, -0.0061)	(-0.3796, -0.2632)	(-0.0087, -0.0066)	(-0.6541, -0.6280)
SDF	(-0.0300, -0.0271)	(-0.4666, -0.3505)	(-0.0343, -0.0319)	(-0.9545, -0.9080)
LIN	(-0.0046, -0.0033)	(-0.1645, -0.1174)	(-0.0155, -0.0136)	(-0.2117, -0.1807)
MAN	(-0.0000, <b>0.0003</b> )	(-0.0805, -0.0500)	(-0.0133, -0.0120)	(-0.0741, -0.0573)
MRC	(-0.0012, -0.0007)	(-0.0443, -0.0272)	(-0.0035, -0.0018)	<b>(0.0399, 0.0835)</b>
MEO	<b>(0.0001, 0.0013)</b>	(-0.2416, -0.1775)	(-0.0197, -0.0168)	(-0.4218, -0.3533)
MUV	(-0.0017, -0.0008)	(-0.1856, -0.1246)	(-0.0009, <b>0.0020</b> )	(-0.1478, -0.1025)
RWE	(-0.0033, -0.0025)	(-0.1193, -0.0919)	(-0.0164, -0.0141)	(-0.0865, -0.0589)
SZG	(-0.0056, -0.0038)	(-0.2691, -0.2260)	(-0.0114, -0.0090)	(-0.1973, -0.1589)
SAP	(-0.0221, -0.0185)	(-1.0128, -0.8478)	(-0.0473, -0.0430)	(-0.6137, -0.5730)
SIE	(-0.0352, -0.0296)	(-1.7842, -1.5185)	(-0.0485, -0.0419)	(-1.0133, -0.8618)
TKA	(-0.0066, -0.0032)	(-0.4857, -0.3746)	(-0.0087, -0.0040)	(-0.1169, -0.0716)
VOW	(-0.0033, -0.0022)	(-0.1403, -0.0958)	(-0.0082, -0.0052)	(-0.1598, -0.1164)



## Appendix B.

# Empirical One-step-ahead Forecasting

### B.1. Conditional Mean One-step-ahead Forecasting

The tables in this appendix denote the results for the one-step-ahead forecasting with a moving window for 60 minutes frequency data. They are to read as follows. The first (second) four columns denote the conditional mean calculated according to the ARMA(1,1) (ARMA(2,1)) model for LLSA, ARMA itself, the MODWT, and the median filter. All values are stated for the unit  $10^{-3}$ . In Tables B.4 to B.6 the 5% (i. e.,  $\alpha = 0.05$ ) bootstrapped confidence intervals for the ARMA(1,1) model are stated for  $K \in \{1, 2, 3\}$ , respectively.

Table B.1.: Conditional mean one-step-ahead price forecasting,  $K = 1$

	ARMA(1,1)				ARMA(2,1)			
	LLSA	ARMA	MODWT	Median	LLSA	ARMA	MODWT	Median
ADS	4.6480	4.7065	4.6686	4.6559	4.6605	4.7184	4.6802	4.6674
ALV	5.5858	5.6365	<b>5.5802</b>	<b>5.5726</b>	5.5914	5.6449	<b>5.5856</b>	<b>5.5801</b>
BAS	3.2772	<b>3.1121</b>	<b>3.2536</b>	<b>2.9046</b>	3.3228	<b>3.1405</b>	<b>3.2907</b>	<b>2.9110</b>
BAY	5.5455	5.5681	<b>5.5118</b>	<b>5.5217</b>	5.5544	5.5851	<b>5.5196</b>	<b>5.5316</b>
BEI	4.8267	4.8995	4.8577	4.8363	4.8346	4.9071	4.8632	4.8384
BMW	5.3135	5.3926	5.3154	5.3137	5.3277	5.4022	5.3307	<b>5.3244</b>
CBK	6.5162	6.5465	<b>6.5074</b>	6.5227	6.5199	6.5489	<b>6.5082</b>	6.5239
DAI	4.4306	4.4670	<b>4.4181</b>	<b>4.4288</b>	4.4450	4.4804	<b>4.4324</b>	4.4488
DBK	5.0453	5.0580	5.0493	<b>5.0333</b>	5.0662	5.0764	5.0702	<b>5.0548</b>
DB1	4.8311	4.8664	4.8323	<b>4.8112</b>	4.8345	4.8677	4.8357	<b>4.8161</b>
DPB	5.7655	5.8179	5.7769	5.7774	5.7870	5.8355	5.7988	5.8021
DPW	4.2808	4.3376	4.2913	4.2812	4.2877	4.3453	4.2951	4.2882
DTE	4.1490	4.1589	<b>4.1439</b>	<b>4.1392</b>	4.1417	4.1497	<b>4.1352</b>	<b>4.1298</b>
EPC	6.5854	<b>6.3896</b>	<b>6.4594</b>	<b>6.2202</b>	6.6043	<b>6.3953</b>	<b>6.4948</b>	<b>6.2495</b>
FME	5.3449	5.4701	5.3830	5.3974	5.3460	5.4727	5.3839	5.3994
LHA	5.5580	5.6198	5.5626	<b>5.5545</b>	5.5668	5.6302	5.5725	<b>5.5647</b>
HNK	4.5328	4.6289	4.5526	4.5415	4.5386	4.6329	4.5590	4.5452
IFX	6.1535	6.1612	<b>6.1283</b>	<b>6.1133</b>	6.1791	<b>6.1735</b>	<b>6.1496</b>	<b>6.1371</b>
SDF	10.784	<b>10.568</b>	10.787	<b>10.084</b>	10.841	<b>10.571</b>	<b>10.818</b>	<b>10.107</b>
LIN	5.1221	5.2036	5.1392	5.1296	5.1300	5.2106	5.1460	5.1388
MAN	4.5561	4.6030	<b>4.5460</b>	<b>4.5337</b>	4.5609	4.6072	<b>4.5522</b>	<b>4.5408</b>
MRC	4.6282	4.6606	<b>4.6142</b>	<b>4.6035</b>	4.6497	4.6892	<b>4.6431</b>	<b>4.6274</b>
MEO	4.1639	4.1858	<b>4.1553</b>	<b>4.1426</b>	4.1696	4.1936	<b>4.1637</b>	<b>4.1477</b>
MUV	5.4272	5.4674	5.4321	5.4307	5.4363	5.4871	5.4441	5.4411
RWE	4.3271	4.3702	4.3530	4.3428	4.3364	4.3779	4.3606	4.3510
SZG	7.4782	7.5078	<b>7.4401</b>	<b>7.4290</b>	7.4594	7.4984	<b>7.4139</b>	<b>7.4085</b>
SAP	5.4023	5.4742	5.4308	<b>5.3974</b>	5.4040	5.4765	5.4360	<b>5.4035</b>
SIE	4.4264	4.4748	4.4356	<b>4.4179</b>	4.4405	4.4883	4.4511	<b>4.4277</b>
TKA	6.0097	6.0612	<b>6.0066</b>	<b>5.9927</b>	6.0142	6.0605	<b>6.0081</b>	<b>5.9933</b>
VOW	29.335	<b>28.638</b>	<b>29.075</b>	<b>29.029</b>	30.139	<b>29.644</b>	<b>30.078</b>	<b>29.987</b>

Table B.2.: Conditional mean one-step-ahead price forecasting,  $K = 2$

	ARMA(1,1)				ARMA(2,1)			
	LLSA	ARMA	MODWT	Median	LLSA	ARMA	MODWT	Median
ADS	4.6336	4.7065	4.6686	4.6559	4.6463	4.7184	4.6802	4.6674
ALV	5.5592	5.6365	5.5802	5.5726	5.5634	5.6449	5.5856	5.5801
BAS	3.2277	<b>3.1121</b>	3.2536	<b>2.9046</b>	3.2689	<b>3.1405</b>	3.2907	<b>2.9110</b>
BAY	5.5284	5.5681	<b>5.5118</b>	<b>5.5217</b>	5.5354	5.5851	<b>5.5196</b>	<b>5.5316</b>
BEI	4.8253	4.8995	4.8577	4.8363	4.8318	4.9071	4.8632	4.8384
BMW	5.3064	5.3926	5.3154	5.3137	5.3225	5.4022	5.3307	5.3244
CBK	6.4548	6.5465	6.5074	6.5227	6.4616	6.5489	6.5082	6.5239
DAI	4.4341	4.4670	<b>4.4181</b>	<b>4.4288</b>	4.4541	4.4804	<b>4.4324</b>	<b>4.4488</b>
DBK	5.0156	5.0580	5.0493	5.0333	5.0349	5.0764	5.0702	5.0548
DB1	4.8133	4.8664	4.8323	<b>4.8112</b>	4.8173	4.8677	4.8357	<b>4.8161</b>
DPB	5.7766	5.8179	5.7769	5.7774	5.7978	5.8355	5.7988	5.8021
DPW	4.2821	4.3376	4.2913	<b>4.2812</b>	4.2891	4.3453	4.2951	<b>4.2882</b>
DTE	4.1344	4.1589	4.1439	4.1392	4.1276	4.1497	4.1352	4.1298
EPC	6.4799	<b>6.3896</b>	<b>6.4594</b>	<b>6.2202</b>	6.4943	<b>6.3953</b>	6.4948	<b>6.2495</b>
FME	5.3449	5.4701	5.3830	5.3974	5.3463	5.4727	5.3839	5.3994
LHA	5.5176	5.6198	5.5626	5.5545	5.5257	5.6302	5.5725	5.5647
HNK	4.5224	4.6289	4.5526	4.5415	4.5265	4.6329	4.5590	4.5452
IFX	6.1300	6.1612	<b>6.1283</b>	<b>6.1133</b>	6.1569	6.1735	<b>6.1496</b>	<b>6.1371</b>
SDF	10.887	<b>10.568</b>	<b>10.787</b>	<b>10.084</b>	10.934	<b>10.571</b>	<b>10.818</b>	<b>1.0107</b>
LIN	5.1198	5.2036	5.1392	5.1296	5.1274	5.2106	5.1460	5.1388
MAN	4.4896	4.6030	4.5460	4.5337	4.4928	4.6072	4.5522	4.5408
MRC	4.6040	4.6606	4.6142	<b>4.6035</b>	4.6247	4.6892	4.6431	4.6274
MEO	4.1452	4.1858	4.1553	<b>4.1426</b>	4.1510	4.1936	4.1637	<b>4.1477</b>
MUV	5.4027	5.4674	5.4321	5.4307	5.4129	5.4871	5.4441	5.4411
RWE	4.3368	4.3702	4.3530	4.3428	4.3448	4.3779	4.3606	4.3510
SZG	7.3843	7.5078	7.4401	7.4290	7.3606	7.4984	7.4139	7.4085
SAP	5.4033	5.4742	5.4308	<b>5.3974</b>	5.3990	5.4765	5.4360	5.4035
SIE	4.4273	4.4748	4.4356	<b>4.4179</b>	4.4404	4.4883	4.4511	<b>4.4277</b>
TKA	5.9789	6.0612	6.0066	5.9927	5.9836	6.0605	6.0081	5.9933
VOW	29.042	<b>28.638</b>	29.075	<b>29.029</b>	29.912	<b>29.644</b>	30.078	29.987

Table B.3.: Conditional mean one-step-ahead price forecasting,  $K = 3$

	ARMA(1,1)				ARMA(2,1)			
	LLSA	ARMA	MODWT	Median	LLSA	ARMA	MODWT	Median
ADS	4.6521	4.7065	4.6686	4.6559	4.6678	4.7184	4.6802	<b>4.6674</b>
ALV	5.5344	5.6365	5.5802	5.5726	5.5366	5.6449	5.5856	5.5801
BAS	3.2150	<b>3.1121</b>	3.2536	<b>2.9046</b>	3.2557	<b>3.1405</b>	3.2907	<b>2.9110</b>
BAY	5.5213	5.5681	<b>5.5118</b>	5.5217	5.5284	5.5851	<b>5.5196</b>	5.5316
BEI	4.8154	4.8995	4.8577	4.8363	4.8211	4.9071	4.8632	4.8384
BMW	5.2897	5.3926	5.3154	5.3137	5.3079	5.4022	5.3307	5.3244
CBK	6.4670	6.5465	6.5074	6.5227	6.4751	6.5489	6.5082	6.5239
DAI	4.3993	4.4670	4.4181	4.4288	4.4185	4.4804	4.4324	4.4488
DBK	5.0036	5.0580	5.0493	5.0333	5.0243	5.0764	5.0702	5.0548
DB1	4.8197	4.8664	4.8323	<b>4.8112</b>	4.8250	4.8677	4.8357	<b>4.8161</b>
DPB	5.7720	5.8179	5.7769	5.7774	5.7937	5.8355	5.7988	5.8021
DPW	4.2488	4.3376	4.2913	4.2812	4.2584	4.3453	4.2951	4.2882
DTE	4.1188	4.1589	4.1439	4.1392	4.1102	4.1497	4.1352	4.1298
EPC	6.4219	<b>6.3896</b>	6.4594	<b>6.2202</b>	6.4333	<b>6.3953</b>	6.4948	<b>6.2495</b>
FME	5.3275	5.4701	5.3830	5.3974	5.3294	5.4727	5.3839	5.3994
LHA	5.5098	5.6198	5.5626	5.5545	5.5172	5.6302	5.5725	5.5647
HNK	4.5181	4.6289	4.5526	4.5415	4.5221	4.6329	4.5590	4.5452
IFX	6.0622	6.1612	6.1283	6.1133	6.0820	6.1735	6.1496	6.1371
SDF	10.942	<b>10.568</b>	<b>10.787</b>	<b>10.084</b>	10.981	<b>10.571</b>	<b>10.818</b>	<b>10.107</b>
LIN	5.0932	5.2036	5.1392	5.1296	5.0982	5.2106	5.1460	5.1388
MAN	4.4920	4.6030	4.5460	4.5337	4.4967	4.6072	4.5522	4.5408
MRC	4.5936	4.6606	4.6142	4.6035	4.6144	4.6892	4.6431	4.6274
MEO	4.1041	4.1858	4.1553	4.1426	4.1095	4.1936	4.1637	4.1477
MUV	5.4008	5.4674	5.4321	5.4307	5.4081	5.4871	5.4441	5.4411
RWE	4.3429	4.3702	4.3530	<b>4.3428</b>	4.3499	4.3779	4.3606	4.3510
SZG	7.4048	7.5078	7.4401	7.4290	7.3803	7.4984	7.4139	7.4085
SAP	5.3560	5.4742	5.4308	5.3974	5.3501	5.4765	5.4360	5.4035
SIE	4.3988	4.4748	4.4356	4.4179	4.4081	4.4883	4.4511	4.4277
TKA	5.9659	6.0612	6.0066	5.9927	5.9675	6.0605	6.0081	5.9933
VOW	28.897	<b>28.638</b>	29.075	29.029	29.702	<b>29.644</b>	30.078	29.987

Table B.4.: Conditional mean confidence intervals,  $K = 1$

	ARMA(1,1)	MODWT	Median
ADS	$(-1.0987 \cdot 10^{-4}, -6.5164 \cdot 10^{-6})$	$(-4.1559 \cdot 10^{-5}, -1.2934 \cdot 10^{-7})$	$(-3.2547 \cdot 10^{-5}, \mathbf{1.6430} \cdot 10^{-5})$
ALV	$(-1.3359 \cdot 10^{-4}, \mathbf{3.2454} \cdot 10^{-5})$	$(-2.5553 \cdot 10^{-5}, \mathbf{3.6524} \cdot 10^{-5})$	$(-1.8620 \cdot 10^{-5}, \mathbf{4.4960} \cdot 10^{-5})$
BAS	$(\mathbf{2.5376} \cdot 10^{-5}, \mathbf{3.3885} \cdot 10^{-4})$	$(-7.0152 \cdot 10^{-5}, \mathbf{1.3313} \cdot 10^{-4})$	$(\mathbf{2.5139} \cdot 10^{-4}, \mathbf{5.7444} \cdot 10^{-4})$
BAY	$(-1.1038 \cdot 10^{-4}, \mathbf{6.5452} \cdot 10^{-5})$	$(-4.3983 \cdot 10^{-6}, \mathbf{7.1443} \cdot 10^{-5})$	$(-3.1509 \cdot 10^{-6}, \mathbf{5.1136} \cdot 10^{-5})$
BEI	$(-1.2687 \cdot 10^{-4}, -1.8431 \cdot 10^{-5})$	$(-5.7376 \cdot 10^{-5}, -4.5630 \cdot 10^{-6})$	$(-3.5125 \cdot 10^{-5}, \mathbf{1.5630} \cdot 10^{-5})$
BMW	$(-1.4541 \cdot 10^{-4}, -1.3716 \cdot 10^{-5})$	$(-2.4248 \cdot 10^{-5}, \mathbf{2.0962} \cdot 10^{-5})$	$(-2.8320 \cdot 10^{-5}, \mathbf{2.8312} \cdot 10^{-5})$
CBK	$(-1.1719 \cdot 10^{-4}, \mathbf{5.6821} \cdot 10^{-5})$	$(-1.7524 \cdot 10^{-5}, \mathbf{3.5674} \cdot 10^{-5})$	$(-3.8866 \cdot 10^{-5}, \mathbf{2.5397} \cdot 10^{-5})$
DAI	$(-9.3305 \cdot 10^{-5}, \mathbf{1.9730} \cdot 10^{-5})$	$(-1.2928 \cdot 10^{-5}, \mathbf{3.7345} \cdot 10^{-5})$	$(-1.9004 \cdot 10^{-5}, \mathbf{2.2793} \cdot 10^{-5})$
DBK	$(-8.0700 \cdot 10^{-5}, \mathbf{5.5135} \cdot 10^{-5})$	$(-3.2218 \cdot 10^{-5}, \mathbf{2.3894} \cdot 10^{-5})$	$(-1.5829 \cdot 10^{-5}, \mathbf{3.9539} \cdot 10^{-5})$
DB1	$(-9.0143 \cdot 10^{-5}, \mathbf{1.8789} \cdot 10^{-5})$	$(-1.9851 \cdot 10^{-5}, \mathbf{1.7063} \cdot 10^{-5})$	$(-5.4207 \cdot 10^{-6}, \mathbf{4.5244} \cdot 10^{-5})$
DPB	$(-1.3217 \cdot 10^{-4}, \mathbf{2.6619} \cdot 10^{-5})$	$(-4.2457 \cdot 10^{-5}, \mathbf{1.9140} \cdot 10^{-5})$	$(-4.0607 \cdot 10^{-5}, \mathbf{1.7043} \cdot 10^{-5})$
DPW	$(-1.1923 \cdot 10^{-4}, \mathbf{5.5089} \cdot 10^{-6})$	$(-3.8453 \cdot 10^{-5}, \mathbf{1.6935} \cdot 10^{-5})$	$(-3.0063 \cdot 10^{-5}, \mathbf{2.8159} \cdot 10^{-5})$
DTE	$(-4.6306 \cdot 10^{-5}, \mathbf{2.6422} \cdot 10^{-5})$	$(-1.0307 \cdot 10^{-5}, \mathbf{2.0793} \cdot 10^{-5})$	$(-1.5194 \cdot 10^{-5}, \mathbf{3.5176} \cdot 10^{-5})$
EPC	$(\mathbf{9.4795} \cdot 10^{-5}, \mathbf{3.0593} \cdot 10^{-4})$	$(\mathbf{5.4875} \cdot 10^{-5}, \mathbf{2.0219} \cdot 10^{-4})$	$(\mathbf{2.6255} \cdot 10^{-4}, \mathbf{4.7722} \cdot 10^{-4})$
FME	$(-1.7817 \cdot 10^{-4}, -7.2412 \cdot 10^{-5})$	$(-5.8829 \cdot 10^{-5}, -1.7668 \cdot 10^{-5})$	$(-7.8320 \cdot 10^{-5}, -2.7410 \cdot 10^{-5})$
LHA	$(-1.2644 \cdot 10^{-4}, \mathbf{2.4499} \cdot 10^{-6})$	$(-3.2310 \cdot 10^{-5}, \mathbf{2.2760} \cdot 10^{-5})$	$(-2.4077 \cdot 10^{-5}, \mathbf{3.0585} \cdot 10^{-5})$
HNK	$(-1.4939 \cdot 10^{-4}, -4.2150 \cdot 10^{-5})$	$(-3.8802 \cdot 10^{-5}, -3.6817 \cdot 10^{-7})$	$(-3.1628 \cdot 10^{-5}, \mathbf{1.3786} \cdot 10^{-5})$
IFX	$(-8.1251 \cdot 10^{-5}, \mathbf{6.6116} \cdot 10^{-5})$	$(-6.0892 \cdot 10^{-6}, \mathbf{5.6608} \cdot 10^{-5})$	$(-2.6405 \cdot 10^{-6}, \mathbf{8.2660} \cdot 10^{-5})$
SDF	$(-8.2175 \cdot 10^{-5}, \mathbf{5.8055} \cdot 10^{-4})$	$(-2.1120 \cdot 10^{-4}, \mathbf{2.4278} \cdot 10^{-4})$	$(\mathbf{4.5159} \cdot 10^{-4}, \mathbf{1.0509} \cdot 10^{-3})$
LIN	$(-1.4931 \cdot 10^{-4}, -1.4012 \cdot 10^{-5})$	$(-4.8114 \cdot 10^{-5}, \mathbf{1.4089} \cdot 10^{-5})$	$(-2.9499 \cdot 10^{-5}, \mathbf{1.4585} \cdot 10^{-5})$
MAN	$(-1.1425 \cdot 10^{-4}, \mathbf{2.0143} \cdot 10^{-5})$	$(-1.6058 \cdot 10^{-5}, \mathbf{3.6291} \cdot 10^{-5})$	$(-4.5776 \cdot 10^{-6}, \mathbf{4.8784} \cdot 10^{-5})$
MRC	$(-1.0185 \cdot 10^{-4}, \mathbf{3.5153} \cdot 10^{-5})$	$(-1.6093 \cdot 10^{-5}, \mathbf{4.4479} \cdot 10^{-5})$	$(-5.9270 \cdot 10^{-7}, \mathbf{5.0203} \cdot 10^{-5})$
MEO	$(-7.8359 \cdot 10^{-5}, \mathbf{3.3172} \cdot 10^{-5})$	$(-1.4676 \cdot 10^{-5}, \mathbf{3.1336} \cdot 10^{-5})$	$(-1.1219 \cdot 10^{-5}, \mathbf{5.3704} \cdot 10^{-5})$
MUV	$(-1.0055 \cdot 10^{-4}, \mathbf{1.9968} \cdot 10^{-5})$	$(-3.2201 \cdot 10^{-5}, \mathbf{2.2138} \cdot 10^{-5})$	$(-3.5350 \cdot 10^{-5}, \mathbf{2.8628} \cdot 10^{-5})$
RWE	$(-8.6084 \cdot 10^{-5}, -4.0651 \cdot 10^{-7})$	$(-4.7550 \cdot 10^{-5}, -4.4687 \cdot 10^{-6})$	$(-4.0077 \cdot 10^{-5}, \mathbf{8.4678} \cdot 10^{-6})$
SZG	$(-1.3650 \cdot 10^{-4}, \mathbf{7.5183} \cdot 10^{-5})$	$(-1.1580 \cdot 10^{-5}, \mathbf{8.8742} \cdot 10^{-5})$	$(-1.2955 \cdot 10^{-5}, \mathbf{1.1100} \cdot 10^{-4})$
SAP	$(-1.5777 \cdot 10^{-4}, \mathbf{1.3552} \cdot 10^{-5})$	$(-8.2440 \cdot 10^{-5}, \mathbf{2.6255} \cdot 10^{-5})$	$(-2.5559 \cdot 10^{-5}, \mathbf{3.4329} \cdot 10^{-5})$
SIE	$(-1.0978 \cdot 10^{-4}, \mathbf{1.3043} \cdot 10^{-5})$	$(-3.9026 \cdot 10^{-5}, \mathbf{2.1138} \cdot 10^{-5})$	$(-2.0722 \cdot 10^{-5}, \mathbf{3.7655} \cdot 10^{-5})$
TKA	$(-1.3202 \cdot 10^{-4}, \mathbf{2.9975} \cdot 10^{-5})$	$(-3.1074 \cdot 10^{-5}, \mathbf{3.6934} \cdot 10^{-5})$	$(-2.5284 \cdot 10^{-5}, \mathbf{5.8494} \cdot 10^{-5})$
VOW	$(-8.4255 \cdot 10^{-5}, \mathbf{1.5600} \cdot 10^{-3})$	$(-2.4172 \cdot 10^{-4}, \mathbf{6.8169} \cdot 10^{-4})$	$(-2.7627 \cdot 10^{-4}, \mathbf{7.9332} \cdot 10^{-4})$

Table B.5.: Conditional mean confidence intervals,  $K = 2$

	ARMA(1,1)	MODWT	Median
ADS	$(-1.2901 \cdot 10^{-4}, -1.7187 \cdot 10^{-5})$	$(-6.3628 \cdot 10^{-5}, -6.7372 \cdot 10^{-6})$	$(-4.6527 \cdot 10^{-5}, \mathbf{1.9998} \cdot 10^{-6})$
ALV	$(-1.5831 \cdot 10^{-4}, \mathbf{5.8871} \cdot 10^{-6})$	$(-5.6367 \cdot 10^{-5}, \mathbf{1.4802} \cdot 10^{-5})$	$(-4.4056 \cdot 10^{-5}, \mathbf{1.6845} \cdot 10^{-5})$
BAS	$(-2.4795 \cdot 10^{-5}, \mathbf{2.9426} \cdot 10^{-4})$	$(-1.2298 \cdot 10^{-4}, \mathbf{9.2152} \cdot 10^{-5})$	$(\mathbf{1.9922} \cdot 10^{-4}, \mathbf{5.2513} \cdot 10^{-4})$
BAY	$(-1.2768 \cdot 10^{-4}, \mathbf{4.8181} \cdot 10^{-5})$	$(-2.6490 \cdot 10^{-5}, \mathbf{5.9989} \cdot 10^{-5})$	$(-2.5152 \cdot 10^{-5}, \mathbf{3.8733} \cdot 10^{-5})$
BEI	$(-1.3993 \cdot 10^{-4}, -6.9474 \cdot 10^{-6})$	$(-6.8522 \cdot 10^{-5}, \mathbf{4.4309} \cdot 10^{-6})$	$(-4.1185 \cdot 10^{-5}, \mathbf{1.9323} \cdot 10^{-5})$
BMW	$(-1.5599 \cdot 10^{-4}, -1.7052 \cdot 10^{-5})$	$(-4.2676 \cdot 10^{-5}, \mathbf{2.5501} \cdot 10^{-5})$	$(-4.3495 \cdot 10^{-5}, \mathbf{2.8980} \cdot 10^{-5})$
CBK	$(-1.7824 \cdot 10^{-4}, -3.5578 \cdot 10^{-6})$	$(-8.6374 \cdot 10^{-5}, -1.8677 \cdot 10^{-5})$	$(-1.0155 \cdot 10^{-4}, -3.4527 \cdot 10^{-5})$
DAI	$(-9.9344 \cdot 10^{-5}, \mathbf{3.3239} \cdot 10^{-5})$	$(-1.8374 \cdot 10^{-5}, \mathbf{5.0088} \cdot 10^{-5})$	$(-1.9681 \cdot 10^{-5}, \mathbf{3.0650} \cdot 10^{-5})$
DBK	$(-1.1392 \cdot 10^{-4}, \mathbf{2.8494} \cdot 10^{-5})$	$(-6.7330 \cdot 10^{-5}, -5.2833 \cdot 10^{-7})$	$(-4.7471 \cdot 10^{-5}, \mathbf{1.2256} \cdot 10^{-5})$
DB1	$(-1.1430 \cdot 10^{-4}, \mathbf{8.0454} \cdot 10^{-6})$	$(-5.1366 \cdot 10^{-5}, \mathbf{1.3334} \cdot 10^{-5})$	$(-3.0087 \cdot 10^{-5}, \mathbf{3.4580} \cdot 10^{-5})$
DPB	$(-1.3102 \cdot 10^{-4}, \mathbf{4.7933} \cdot 10^{-5})$	$(-3.7751 \cdot 10^{-5}, \mathbf{3.7234} \cdot 10^{-5})$	$(-3.1467 \cdot 10^{-5}, \mathbf{2.9999} \cdot 10^{-5})$
DPW	$(-1.2579 \cdot 10^{-4}, \mathbf{1.3486} \cdot 10^{-5})$	$(-4.8438 \cdot 10^{-5}, \mathbf{2.9342} \cdot 10^{-5})$	$(-3.4648 \cdot 10^{-5}, \mathbf{3.6546} \cdot 10^{-5})$
DTE	$(-6.6895 \cdot 10^{-5}, \mathbf{1.7962} \cdot 10^{-5})$	$(-3.4109 \cdot 10^{-5}, \mathbf{1.5143} \cdot 10^{-5})$	$(-3.1393 \cdot 10^{-5}, \mathbf{2.1597} \cdot 10^{-5})$
EPC	$(-2.6542 \cdot 10^{-5}, \mathbf{2.1546} \cdot 10^{-4})$	$(-6.4562 \cdot 10^{-5}, \mathbf{1.0899} \cdot 10^{-4})$	$(\mathbf{1.4517} \cdot 10^{-4}, \mathbf{3.8364} \cdot 10^{-4})$
FME	$(-1.9148 \cdot 10^{-4}, -5.8355 \cdot 10^{-5})$	$(-7.0977 \cdot 10^{-5}, -4.9842 \cdot 10^{-6})$	$(-8.6573 \cdot 10^{-5}, -1.8174 \cdot 10^{-5})$
LHA	$(-1.6710 \cdot 10^{-4}, -3.7589 \cdot 10^{-5})$	$(-7.6565 \cdot 10^{-5}, -1.2670 \cdot 10^{-5})$	$(-6.6562 \cdot 10^{-5}, -6.6079 \cdot 10^{-6})$
HNK	$(-1.6629 \cdot 10^{-4}, -4.7915 \cdot 10^{-5})$	$(-5.9203 \cdot 10^{-5}, -1.2718 \cdot 10^{-6})$	$(-4.6882 \cdot 10^{-5}, \mathbf{8.2688} \cdot 10^{-6})$
IFX	$(-1.1417 \cdot 10^{-4}, \mathbf{5.1968} \cdot 10^{-5})$	$(-4.3721 \cdot 10^{-5}, \mathbf{4.7448} \cdot 10^{-5})$	$(-1.6581 \cdot 10^{-5}, \mathbf{5.0290} \cdot 10^{-5})$
SDF	$(-2.1101 \cdot 10^{-5}, \mathbf{7.5377} \cdot 10^{-4})$	$(-1.3111 \cdot 10^{-4}, \mathbf{3.9320} \cdot 10^{-4})$	$(\mathbf{5.1990} \cdot 10^{-4}, \mathbf{1.2027} \cdot 10^{-3})$
LIN	$(-1.5575 \cdot 10^{-4}, -1.0381 \cdot 10^{-5})$	$(-5.7122 \cdot 10^{-5}, \mathbf{1.8089} \cdot 10^{-5})$	$(-4.1054 \cdot 10^{-5}, \mathbf{2.1127} \cdot 10^{-5})$
MAN	$(-1.7815 \cdot 10^{-4}, -4.9085 \cdot 10^{-5})$	$(-8.9263 \cdot 10^{-5}, -2.3597 \cdot 10^{-5})$	$(-7.3823 \cdot 10^{-5}, -1.4497 \cdot 10^{-5})$
MRC	$(-1.2598 \cdot 10^{-4}, \mathbf{1.2731} \cdot 10^{-5})$	$(-4.4113 \cdot 10^{-5}, \mathbf{2.3793} \cdot 10^{-5})$	$(-2.5309 \cdot 10^{-5}, \mathbf{2.6327} \cdot 10^{-5})$
MEO	$(-9.8458 \cdot 10^{-5}, \mathbf{1.6971} \cdot 10^{-5})$	$(-4.1664 \cdot 10^{-5}, \mathbf{2.0853} \cdot 10^{-5})$	$(-3.5342 \cdot 10^{-5}, \mathbf{3.9627} \cdot 10^{-5})$
MUV	$(-1.3490 \cdot 10^{-4}, \mathbf{5.7960} \cdot 10^{-6})$	$(-6.7602 \cdot 10^{-5}, \mathbf{8.2486} \cdot 10^{-6})$	$(-6.5081 \cdot 10^{-5}, \mathbf{8.9499} \cdot 10^{-6})$
RWE	$(-9.1539 \cdot 10^{-5}, \mathbf{2.3843} \cdot 10^{-5})$	$(-4.8356 \cdot 10^{-5}, \mathbf{1.5472} \cdot 10^{-5})$	$(-3.3179 \cdot 10^{-5}, \mathbf{2.1259} \cdot 10^{-5})$
SZG	$(-2.3448 \cdot 10^{-4}, -1.1867 \cdot 10^{-5})$	$(-1.1558 \cdot 10^{-4}, \mathbf{4.4985} \cdot 10^{-6})$	$(-9.4075 \cdot 10^{-5}, \mathbf{3.7149} \cdot 10^{-6})$
SAP	$(-1.6353 \cdot 10^{-4}, \mathbf{1.9296} \cdot 10^{-5})$	$(-8.8755 \cdot 10^{-5}, \mathbf{3.3352} \cdot 10^{-5})$	$(-3.1377 \cdot 10^{-5}, \mathbf{4.2362} \cdot 10^{-5})$
SIE	$(-1.2074 \cdot 10^{-4}, \mathbf{2.4902} \cdot 10^{-5})$	$(-4.6889 \cdot 10^{-5}, \mathbf{3.0838} \cdot 10^{-5})$	$(-1.9475 \cdot 10^{-5}, \mathbf{3.7896} \cdot 10^{-5})$
TKA	$(-1.7755 \cdot 10^{-4}, \mathbf{1.2147} \cdot 10^{-5})$	$(-7.7639 \cdot 10^{-5}, \mathbf{2.1766} \cdot 10^{-5})$	$(-4.9404 \cdot 10^{-5}, \mathbf{2.1442} \cdot 10^{-5})$
VOW	$(-4.0769 \cdot 10^{-4}, \mathbf{1.2410} \cdot 10^{-3})$	$(-5.9620 \cdot 10^{-4}, \mathbf{4.1000} \cdot 10^{-4})$	$(-6.3425 \cdot 10^{-4}, \mathbf{5.3453} \cdot 10^{-4})$

Table B.6.: Conditional mean confidence intervals,  $K = 3$

	ARMA(1,1)	MODWT	Median
ADS	$(-1.2017 \cdot 10^{-4}, \mathbf{1.0971} \cdot 10^{-5})$	$(-5.3445 \cdot 10^{-5}, \mathbf{2.0884} \cdot 10^{-5})$	$(-3.3611 \cdot 10^{-5}, \mathbf{2.6052} \cdot 10^{-5})$
ALV	$(-1.8529 \cdot 10^{-4}, -1.8372 \cdot 10^{-5})$	$(-8.9334 \cdot 10^{-5}, -2.2265 \cdot 10^{-6})$	$(-7.3317 \cdot 10^{-5}, -3.6709 \cdot 10^{-6})$
BAS	$(-4.1194 \cdot 10^{-5}, \mathbf{2.8400} \cdot 10^{-4})$	$(-1.3991 \cdot 10^{-4}, \mathbf{8.6170} \cdot 10^{-5})$	$(\mathbf{1.8395} \cdot 10^{-4}, \mathbf{5.1887} \cdot 10^{-4})$
BAY	$(-1.4357 \cdot 10^{-4}, \mathbf{4.8931} \cdot 10^{-5})$	$(-4.3382 \cdot 10^{-5}, \mathbf{6.1821} \cdot 10^{-5})$	$(-3.9913 \cdot 10^{-5}, \mathbf{3.9318} \cdot 10^{-5})$
BEI	$(-1.5214 \cdot 10^{-4}, -1.6000 \cdot 10^{-5})$	$(-8.0347 \cdot 10^{-5}, -3.9298 \cdot 10^{-6})$	$(-5.1379 \cdot 10^{-5}, \mathbf{8.7515} \cdot 10^{-6})$
BMW	$(-1.7774 \cdot 10^{-4}, -2.7971 \cdot 10^{-5})$	$(-6.2053 \cdot 10^{-5}, \mathbf{1.1071} \cdot 10^{-5})$	$(-5.8537 \cdot 10^{-5}, \mathbf{1.0461} \cdot 10^{-5})$
CBK	$(-1.7097 \cdot 10^{-4}, \mathbf{1.3516} \cdot 10^{-5})$	$(-8.2126 \cdot 10^{-5}, \mathbf{8.0914} \cdot 10^{-7})$	$(-9.5471 \cdot 10^{-5}, -1.6079 \cdot 10^{-5})$
DAI	$(-1.3600 \cdot 10^{-4}, -9.9539 \cdot 10^{-8})$	$(-5.7396 \cdot 10^{-5}, \mathbf{1.9246} \cdot 10^{-5})$	$(-6.0499 \cdot 10^{-5}, \mathbf{2.1392} \cdot 10^{-6})$
DBK	$(-1.2648 \cdot 10^{-4}, \mathbf{1.9365} \cdot 10^{-5})$	$(-8.1058 \cdot 10^{-5}, -1.0216 \cdot 10^{-5})$	$(-5.9613 \cdot 10^{-5}, \mathbf{3.1555} \cdot 10^{-7})$
DB1	$(-1.1547 \cdot 10^{-4}, \mathbf{2.2676} \cdot 10^{-5})$	$(-5.3390 \cdot 10^{-5}, \mathbf{2.8307} \cdot 10^{-5})$	$(-2.7245 \cdot 10^{-5}, \mathbf{4.4681} \cdot 10^{-5})$
DPB	$(-1.4067 \cdot 10^{-4}, \mathbf{4.9191} \cdot 10^{-5})$	$(-5.1885 \cdot 10^{-5}, \mathbf{4.1175} \cdot 10^{-5})$	$(-4.3524 \cdot 10^{-5}, \mathbf{3.3529} \cdot 10^{-5})$
DPW	$(-1.5606 \cdot 10^{-4}, -2.1025 \cdot 10^{-5})$	$(-8.2517 \cdot 10^{-5}, -1.9029 \cdot 10^{-6})$	$(-6.8018 \cdot 10^{-5}, \mathbf{2.7543} \cdot 10^{-6})$
DTE	$(-8.4991 \cdot 10^{-5}, \mathbf{5.5073} \cdot 10^{-6})$	$(-5.3948 \cdot 10^{-5}, \mathbf{4.3354} \cdot 10^{-6})$	$(-4.7837 \cdot 10^{-5}, \mathbf{6.6054} \cdot 10^{-6})$
EPC	$(-1.0477 \cdot 10^{-4}, \mathbf{1.8097} \cdot 10^{-4})$	$(-1.3599 \cdot 10^{-4}, \mathbf{6.6042} \cdot 10^{-5})$	$(\mathbf{7.4811} \cdot 10^{-5}, \mathbf{3.3871} \cdot 10^{-4})$
FME	$(-2.1460 \cdot 10^{-4}, -7.0072 \cdot 10^{-5})$	$(-9.7120 \cdot 10^{-5}, -1.4143 \cdot 10^{-5})$	$(-1.0929 \cdot 10^{-4}, -2.9558 \cdot 10^{-5})$
LHA	$(-1.8132 \cdot 10^{-4}, -4.0410 \cdot 10^{-5})$	$(-8.8052 \cdot 10^{-5}, -1.7291 \cdot 10^{-5})$	$(-7.6597 \cdot 10^{-5}, -1.3048 \cdot 10^{-5})$
HNK	$(-1.8036 \cdot 10^{-4}, -4.1685 \cdot 10^{-5})$	$(-7.1299 \cdot 10^{-5}, \mathbf{2.6617} \cdot 10^{-6})$	$(-5.4211 \cdot 10^{-5}, \mathbf{7.3353} \cdot 10^{-6})$
IFX	$(-1.8686 \cdot 10^{-4}, -1.1909 \cdot 10^{-5})$	$(-1.1906 \cdot 10^{-4}, -1.3058 \cdot 10^{-5})$	$(-9.1969 \cdot 10^{-5}, -9.8860 \cdot 10^{-6})$
SDF	$(\mathbf{4.1617} \cdot 10^{-6}, \mathbf{8.4910} \cdot 10^{-4})$	$(-1.1921 \cdot 10^{-4}, \mathbf{4.9759} \cdot 10^{-4})$	$(\mathbf{5.4606} \cdot 10^{-4}, \mathbf{1.3099} \cdot 10^{-3})$
LIN	$(-1.8333 \cdot 10^{-4}, -3.8088 \cdot 10^{-5})$	$(-8.7956 \cdot 10^{-5}, -4.1482 \cdot 10^{-6})$	$(-7.2376 \cdot 10^{-5}, -4.8773 \cdot 10^{-7})$
MAN	$(-1.8051 \cdot 10^{-4}, -4.1856 \cdot 10^{-5})$	$(-9.1908 \cdot 10^{-5}, -1.6247 \cdot 10^{-5})$	$(-7.5299 \cdot 10^{-5}, -8.7700 \cdot 10^{-6})$
MRC	$(-1.4299 \cdot 10^{-4}, \mathbf{8.3117} \cdot 10^{-6})$	$(-6.2323 \cdot 10^{-5}, \mathbf{2.1227} \cdot 10^{-5})$	$(-3.8067 \cdot 10^{-5}, \mathbf{1.8146} \cdot 10^{-5})$
MEO	$(-1.4245 \cdot 10^{-4}, -2.0231 \cdot 10^{-5})$	$(-8.7598 \cdot 10^{-5}, -1.5892 \cdot 10^{-5})$	$(-7.4360 \cdot 10^{-5}, -3.5839 \cdot 10^{-6})$
MUV	$(-1.4063 \cdot 10^{-4}, \mathbf{6.5612} \cdot 10^{-6})$	$(-7.6464 \cdot 10^{-5}, \mathbf{1.3664} \cdot 10^{-5})$	$(-6.6590 \cdot 10^{-5}, \mathbf{7.3064} \cdot 10^{-6})$
RWE	$(-9.6742 \cdot 10^{-5}, \mathbf{4.2168} \cdot 10^{-5})$	$(-5.4034 \cdot 10^{-5}, \mathbf{3.3013} \cdot 10^{-5})$	$(-3.3719 \cdot 10^{-5}, \mathbf{3.3891} \cdot 10^{-5})$
SZG	$(-2.2680 \cdot 10^{-4}, \mathbf{1.9869} \cdot 10^{-5})$	$(-1.1316 \cdot 10^{-4}, \mathbf{4.2228} \cdot 10^{-5})$	$(-7.6106 \cdot 10^{-5}, \mathbf{2.6795} \cdot 10^{-5})$
SAP	$(-2.1526 \cdot 10^{-4}, -2.2932 \cdot 10^{-5})$	$(-1.4205 \cdot 10^{-4}, -7.0666 \cdot 10^{-6})$	$(-8.1043 \cdot 10^{-5}, -7.6877 \cdot 10^{-7})$
SIE	$(-1.5697 \cdot 10^{-4}, \mathbf{4.7665} \cdot 10^{-6})$	$(-8.3742 \cdot 10^{-5}, \mathbf{1.1349} \cdot 10^{-5})$	$(-5.4912 \cdot 10^{-5}, \mathbf{1.6464} \cdot 10^{-5})$
TKA	$(-1.9739 \cdot 10^{-4}, \mathbf{6.1221} \cdot 10^{-6})$	$(-1.0163 \cdot 10^{-4}, \mathbf{1.8512} \cdot 10^{-5})$	$(-6.5287 \cdot 10^{-5}, \mathbf{1.1408} \cdot 10^{-5})$
VOW	$(-6.1696 \cdot 10^{-4}, \mathbf{1.1954} \cdot 10^{-3})$	$(-8.2940 \cdot 10^{-4}, \mathbf{3.4982} \cdot 10^{-4})$	$(-8.7023 \cdot 10^{-4}, \mathbf{4.7163} \cdot 10^{-4})$

## B.2. Empirical Percentiles

In the following Tables the 5% (i. e.,  $\gamma = 0.05$ ) percentile deviation results are reported. They are calculated from the historical distribution of the via LLSA, MODWT, and median filter detrended series. In Tables B.7 to B.9 the first three columns denote the absolute number of deviations, followed by the respective percentage (in brackets). Due to above choice of  $\gamma$ , a value closer to 5% means a more accurate model. Columns 4 to 6 state the mean width  $d_B$  of the estimated percentile range. In Table B.10 the 5% (i. e.,  $\alpha = 0.05$ ) bootstrapped confidence intervals are given for these mean widths. Tables B.11 to B.13 contain the percentile exceedances, including their percentaged amount of deviations, for the detrended VaR and ES models. The results for the comparison of LLSA to the VaR and ES estimation on the original P/L data are reported in Tables B.14 and B.15.

Table B.7.: Empirical percentiles (Price),  $K = 1$ 

	Exceedances						Mean percentile width $d_B$		
	LLSA		MODWT		Median		LLSA	MODWT	Median
ADS	165	(11.90%)	199	(14.35%)	221	(15.93%)	0.0584	0.0655	<b>0.0533</b>
ALV	232	(16.73%)	<b>198</b>	<b>(14.28%)</b>	<b>200</b>	<b>(14.42%)</b>	0.0641	0.0787	0.0673
BAS	235	(16.94%)	248	(17.88%)	<b>153</b>	<b>(11.03%)</b>	0.0342	0.1351	<b>0.0286</b>
BAY	245	(17.66%)	<b>188</b>	<b>(13.55%)</b>	<b>213</b>	<b>(15.36%)</b>	0.0516	0.0622	<b>0.0509</b>
BEI	168	(12.11%)	205	(14.78%)	196	(14.13%)	0.0520	0.0588	0.0555
CBK	174	(12.55%)	<b>148</b>	<b>(10.67%)</b>	<b>160</b>	<b>(11.54%)</b>	0.0919	0.1027	0.1084
DAI	189	(13.63%)	204	(14.71%)	193	(13.91%)	0.0629	0.0710	<b>0.0592</b>
DBK	235	(16.94%)	<b>225</b>	<b>(16.22%)</b>	<b>198</b>	<b>(14.28%)</b>	0.0621	0.0709	0.0662
DB1	197	(14.20%)	<b>186</b>	<b>(13.41%)</b>	<b>189</b>	<b>(13.63%)</b>	0.0657	0.0718	<b>0.0626</b>
DPB	176	(12.69%)	<b>159</b>	<b>(11.46%)</b>	180	(12.98%)	0.0693	0.0820	<b>0.0692</b>
DPW	186	(13.41%)	199	(14.35%)	<b>185</b>	<b>(13.34%)</b>	0.0500	0.0596	0.0505
DTE	245	(17.66%)	<b>211</b>	<b>(15.21%)</b>	<b>232</b>	<b>(16.73%)</b>	0.0495	0.0564	0.0497
EPC	189	(13.63%)	238	(17.16%)	<b>149</b>	<b>(10.74%)</b>	0.0899	0.1506	<b>0.0686</b>
FME	178	(12.83%)	213	(15.36%)	238	(17.16%)	0.0519	0.0570	0.0541
LHA	150	(10.81%)	170	(12.26%)	177	(12.76%)	0.0719	0.0810	0.0744
HNK	186	(13.41%)	202	(14.56%)	201	(14.49%)	0.0557	0.0631	<b>0.0542</b>
IFX	179	(12.91%)	<b>154</b>	<b>(11.10%)</b>	<b>158</b>	<b>(11.39%)</b>	0.0877	0.1049	<b>0.0866</b>
SDF	99	(7.14%)	192	(13.84%)	112	(8.07%)	0.1897	0.5268	<b>0.1623</b>
LIN	226	(16.29%)	267	(19.25%)	232	(16.73%)	0.0561	0.0674	0.0599
MAN	227	(16.37%)	<b>176</b>	<b>(12.69%)</b>	<b>166</b>	<b>(11.97%)</b>	0.0543	0.0659	0.0569
MRC	217	(15.65%)	<b>197</b>	<b>(14.20%)</b>	<b>209</b>	<b>(15.07%)</b>	0.0569	0.0683	<b>0.0514</b>
MEO	226	(16.29%)	<b>222</b>	<b>(16.01%)</b>	<b>169</b>	<b>(12.18%)</b>	0.0517	0.0641	0.0528
MUV	200	(14.42%)	207	(14.92%)	<b>193</b>	<b>(13.91%)</b>	0.0613	0.0709	0.0633
RWE	156	(11.25%)	233	(16.80%)	222	(16.01%)	0.0462	0.0535	<b>0.0422</b>
SZG	241	(17.38%)	<b>187</b>	<b>(13.48%)</b>	<b>184</b>	<b>(13.27%)</b>	0.0963	0.1152	<b>0.0869</b>
SAP	188	(13.55%)	236	(17.02%)	210	(15.14%)	0.0607	0.0857	<b>0.0583</b>
SIE	181	(13.05%)	224	(16.15%)	205	(14.78%)	0.0489	0.0615	<b>0.0463</b>
TKA	165	(11.90%)	<b>163</b>	<b>(11.75%)</b>	176	(12.69%)	0.0776	0.0915	<b>0.0662</b>
VOW	319	(23.00%)	331	(23.86%)	<b>317</b>	<b>(22.86%)</b>	0.2209	0.3641	0.3407

Table B.8.: Empirical percentiles (Price),  $K = 2$

	Exceedances						Mean percentile width $d_B$		
	LLSA		MODWT		Median		LLSA	MODWT	Median
ADS	158	(11.39%)	199	(14.35%)	221	(15.93%)	0.0509	0.0655	0.0533
ALV	210	(15.14%)	<b>198</b>	<b>(14.28%)</b>	<b>200</b>	<b>(14.42%)</b>	0.0597	0.0787	0.0673
BAS	186	(13.41%)	248	(17.88%)	<b>153</b>	<b>(11.03%)</b>	0.0288	0.1351	<b>0.0286</b>
BAY	212	(15.28%)	<b>188</b>	<b>(13.55%)</b>	213	(15.36%)	0.0477	0.0622	0.0509
BEI	212	(15.28%)	<b>205</b>	<b>(14.78%)</b>	<b>196</b>	<b>(14.13%)</b>	0.0456	0.0588	0.0555
CBK	159	(11.46%)	<b>148</b>	<b>(10.67%)</b>	160	(11.54%)	0.0847	0.1027	0.1084
DAI	207	(14.92%)	<b>204</b>	<b>(14.71%)</b>	<b>193</b>	<b>(13.91%)</b>	0.0538	0.0710	0.0592
DBK	208	(15.00%)	225	(16.22%)	<b>198</b>	<b>(14.28%)</b>	0.0567	0.0709	0.0662
DB1	187	(13.48%)	<b>186</b>	<b>(13.41%)</b>	189	(13.63%)	0.0598	0.0718	0.0626
DPB	208	(15.00%)	<b>159</b>	<b>(11.46%)</b>	<b>180</b>	<b>(12.98%)</b>	0.0643	0.0820	0.0692
DPW	222	(16.01%)	<b>199</b>	<b>(14.35%)</b>	<b>185</b>	<b>(13.34%)</b>	0.0441	0.0596	0.0505
DTE	243	(17.52%)	<b>211</b>	<b>(15.21%)</b>	<b>232</b>	<b>(16.73%)</b>	0.0434	0.0564	0.0497
EPC	182	(13.12%)	238	(17.16%)	<b>149</b>	<b>(10.74%)</b>	0.0740	0.1506	<b>0.0686</b>
FME	259	(18.67%)	<b>213</b>	<b>(15.36%)</b>	<b>238</b>	<b>(17.16%)</b>	0.0443	0.0570	0.0541
LHA	106	(7.64%)	170	(12.26%)	177	(12.76%)	0.0666	0.0810	0.0744
HNK	172	(12.40%)	202	(14.56%)	201	(14.49%)	0.0502	0.0631	0.0542
IFX	178	(12.83%)	<b>154</b>	<b>(11.10%)</b>	<b>158</b>	<b>(11.39%)</b>	0.0794	0.1049	0.0866
SDF	127	(9.16%)	192	(13.84%)	<b>112</b>	<b>(8.07%)</b>	0.1617	0.5268	0.1623
LIN	235	(16.94%)	267	(19.25%)	<b>232</b>	<b>(16.73%)</b>	0.0507	0.0674	0.0599
MAN	125	(9.01%)	176	(12.69%)	166	(11.97%)	0.0496	0.0659	0.0569
MRC	185	(13.34%)	197	(14.20%)	209	(15.07%)	0.0507	0.0683	0.0514
MEO	185	(13.34%)	222	(16.01%)	<b>169</b>	<b>(12.18%)</b>	0.0460	0.0641	0.0528
MUV	162	(11.68%)	207	(14.92%)	193	(13.91%)	0.0550	0.0709	0.0633
RWE	229	(16.51%)	233	(16.80%)	<b>222</b>	<b>(16.01%)</b>	0.0399	0.0535	0.0422
SZG	171	(12.33%)	187	(13.48%)	184	(13.27%)	0.0832	0.1152	0.0869
SAP	224	(16.15%)	236	(17.02%)	<b>210</b>	<b>(15.14%)</b>	0.0519	0.0857	0.0583
SIE	194	(13.99%)	224	(16.15%)	205	(14.78%)	0.0456	0.0615	0.0463
TKA	173	(12.47%)	<b>163</b>	<b>(11.75%)</b>	176	(12.69%)	0.0655	0.0915	0.0662
VOW	312	(22.49%)	331	(23.86%)	317	(22.86%)	0.1912	0.3641	0.3407

Table B.9.: Empirical percentiles (Price),  $K = 3$ 

	Exceedances						Mean percentile width $d_B$		
	LLSA		MODWT		Median		LLSA	MODWT	Median
ADS	220	(15.86%)	<b>199</b>	<b>(14.35%)</b>	221	(15.93%)	0.0453	0.0655	0.0533
ALV	175	(12.62%)	198	(14.28%)	200	(14.42%)	0.0567	0.0787	0.0673
BAS	162	(11.68%)	248	(17.88%)	<b>153</b>	<b>(11.03%)</b>	0.0264	0.1351	0.0286
BAY	226	(16.29%)	<b>188</b>	<b>(13.55%)</b>	<b>213</b>	<b>(15.36%)</b>	0.0447	0.0622	0.0509
BEI	186	(13.41%)	205	(14.78%)	196	(14.13%)	0.0433	0.0588	0.0555
CBK	183	(13.19%)	<b>148</b>	<b>(10.67%)</b>	<b>160</b>	<b>(11.54%)</b>	0.0795	0.1027	0.1084
DAI	203	(14.64%)	204	(14.71%)	<b>193</b>	<b>(13.91%)</b>	0.0458	0.0710	0.0592
DBK	164	(11.82%)	225	(16.22%)	198	(14.28%)	0.0538	0.0709	0.0662
DB1	183	(13.19%)	186	(13.41%)	189	(13.63%)	0.0558	0.0718	0.0626
DPB	202	(14.56%)	<b>159</b>	<b>(11.46%)</b>	<b>180</b>	<b>(12.98%)</b>	0.0612	0.0820	0.0692
DPW	170	(12.26%)	199	(14.35%)	185	(13.34%)	0.0402	0.0596	0.0505
DTE	191	(13.77%)	211	(15.21%)	232	(16.73%)	0.0407	0.0564	0.0497
EPC	202	(14.56%)	238	(17.16%)	<b>149</b>	<b>(10.74%)</b>	0.0652	0.1506	0.0686
FME	264	(19.03%)	<b>213</b>	<b>(15.36%)</b>	<b>238</b>	<b>(17.16%)</b>	0.0404	0.0570	0.0541
LHA	126	(9.08%)	170	(12.26%)	177	(12.76%)	0.0605	0.0810	0.0744
HNK	180	(12.98%)	202	(14.56%)	201	(14.49%)	0.0447	0.0631	0.0542
IFX	160	(11.54%)	<b>154</b>	<b>(11.10%)</b>	<b>158</b>	<b>(11.39%)</b>	0.0707	0.1049	0.0866
SDF	122	(8.80%)	192	(13.84%)	<b>112</b>	<b>(8.07%)</b>	0.1442	0.5268	0.1623
LIN	208	(15.00%)	267	(19.25%)	232	(16.73%)	0.0467	0.0674	0.0599
MAN	145	(10.45%)	176	(12.69%)	166	(11.97%)	0.0464	0.0659	0.0569
MRC	193	(13.91%)	197	(14.20%)	209	(15.07%)	0.0459	0.0683	0.0514
MEO	165	(11.90%)	222	(16.01%)	169	(12.18%)	0.0407	0.0641	0.0528
MUV	174	(12.55%)	207	(14.92%)	193	(13.91%)	0.0502	0.0709	0.0633
RWE	237	(17.09%)	<b>233</b>	<b>(16.80%)</b>	<b>222</b>	<b>(16.01%)</b>	0.0363	0.0535	0.0422
SZG	186	(13.41%)	187	(13.48%)	<b>184</b>	<b>(13.27%)</b>	0.0748	0.1152	0.0869
SAP	191	(13.77%)	236	(17.02%)	210	(15.14%)	0.0493	0.0857	0.0583
SIE	205	(14.78%)	224	(16.15%)	205	(14.78%)	0.0410	0.0615	0.0463
TKA	183	(13.19%)	<b>163</b>	<b>(11.75%)</b>	<b>176</b>	<b>(12.69%)</b>	0.0586	0.0915	0.0662
VOW	274	(19.75%)	331	(23.86%)	317	(22.86%)	0.1769	0.3641	0.3407

Table B.10.: Mean percentile width confidence intervals

	$K = 1$		$K = 2$		$K = 3$	
	MODWT	Median	MODWT	Median	MODWT	Median
ADS	(-0.0073, -0.0069)	<b>(0.0048, 0.0056)</b>	(-0.0150, -0.0143)	(-0.0028, -0.0019)	(-0.0205, -0.0200)	(-0.0083, -0.0076)
ALV	(-0.0151, -0.0141)	(-0.0036, -0.0026)	(-0.0196, -0.0185)	(-0.0079, -0.0072)	(-0.0226, -0.0215)	(-0.0110, -0.0102)
BAS	(-0.1060, -0.0957)	<b>(0.0052, 0.0060)</b>	(-0.1114, -0.1012)	(-0.0002, <b>0.0007</b> )	(-0.1139, -0.1036)	(-0.0026, -0.0018)
BAY	(-0.0111, -0.0102)	<b>(0.0004, 0.0008)</b>	(-0.0149, -0.0140)	(-0.0034, -0.0030)	(-0.0180, -0.0171)	(-0.0065, -0.0061)
BEI	(-0.0071, -0.0065)	(-0.0036, -0.0033)	(-0.0136, -0.0128)	(-0.0100, -0.0096)	(-0.0160, -0.0151)	(-0.0124, -0.0120)
BMW	(-0.0065, -0.0060)	(-0.0041, -0.0031)	(-0.0146, -0.0137)	(-0.0122, -0.0109)	(-0.0222, -0.0214)	(-0.0198, -0.0185)
CBK	(-0.0113, -0.0105)	(-0.0169, -0.0161)	(-0.0184, -0.0177)	(-0.0242, -0.0232)	(-0.0238, -0.0227)	(-0.0295, -0.0283)
DAI	(-0.0084, -0.0078)	<b>(0.0032, 0.0040)</b>	(-0.0176, -0.0167)	(-0.0059, -0.0049)	(-0.0256, -0.0246)	(-0.0140, -0.0129)
DBK	(-0.0092, -0.0085)	(-0.0045, -0.0037)	(-0.0146, -0.0137)	(-0.0099, -0.0090)	(-0.0176, -0.0167)	(-0.0128, -0.0120)
DBI	(-0.0064, -0.0058)	<b>(0.0028, 0.0033)</b>	(-0.0124, -0.0117)	(-0.0031, -0.0026)	(-0.0163, -0.0157)	(-0.0070, -0.0066)
DPB	(-0.0131, -0.0123)	(-0.0004, <b>0.0005</b> )	(-0.0181, -0.0174)	(-0.0052, -0.0046)	(-0.0213, -0.0204)	(-0.0083, -0.0077)
DPW	(-0.0099, -0.0093)	(-0.0008, -0.0001)	(-0.0159, -0.0152)	(-0.0068, -0.0060)	(-0.0198, -0.0191)	(-0.0106, -0.0101)
DTE	(-0.0071, -0.0067)	(-0.0008, <b>0.0004</b> )	(-0.0134, -0.0127)	(-0.0071, -0.0056)	(-0.0160, -0.0154)	(-0.0098, -0.0083)
EPC	(-0.0635, -0.0580)	<b>(0.0203, 0.0222)</b>	(-0.0793, -0.0739)	<b>(0.0045, 0.0063)</b>	(-0.0879, -0.0828)	(-0.0045, -0.0024)
FME	(-0.0053, -0.0049)	(-0.0025, -0.0019)	(-0.0131, -0.0123)	(-0.0104, -0.0092)	(-0.0170, -0.0163)	(-0.0143, -0.0132)
LHA	(-0.0095, -0.0088)	(-0.0028, -0.0022)	(-0.0148, -0.0141)	(-0.0082, -0.0075)	(-0.0209, -0.0201)	(-0.0142, -0.0136)
HNK	(-0.0076, -0.0073)	<b>(0.0012, 0.0018)</b>	(-0.0132, -0.0126)	(-0.0043, -0.0036)	(-0.0187, -0.0182)	(-0.0099, -0.0091)
IFX	(-0.0175, -0.0169)	<b>(0.0007, 0.0015)</b>	(-0.0258, -0.0252)	(-0.0077, -0.0067)	(-0.0346, -0.0338)	(-0.0164, -0.0155)
SDF	(-0.3548, -0.3200)	<b>(0.0262, 0.0286)</b>	(-0.3827, -0.3474)	(-0.0013, <b>0.0003</b> )	(-0.4007, -0.3647)	(-0.0188, -0.0174)
LIN	(-0.0118, -0.0107)	(-0.0040, -0.0036)	(-0.0173, -0.0161)	(-0.0095, -0.0090)	(-0.0214, -0.0198)	(-0.0136, -0.0127)
MAN	(-0.0119, -0.0113)	(-0.0030, -0.0023)	(-0.0166, -0.0160)	(-0.0077, -0.0069)	(-0.0199, -0.0193)	(-0.0111, -0.0101)
MRC	(-0.0118, -0.0111)	<b>(0.0051, 0.0058)</b>	(-0.0180, -0.0174)	(-0.0011, -0.0005)	(-0.0227, -0.0221)	(-0.0058, -0.0052)
MEO	(-0.0129, -0.0120)	(-0.0016, -0.0006)	(-0.0188, -0.0175)	(-0.0075, -0.0062)	(-0.0241, -0.0228)	(-0.0128, -0.0114)
MUV	(-0.0099, -0.0091)	(-0.0022, -0.0017)	(-0.0163, -0.0155)	(-0.0086, -0.0081)	(-0.0211, -0.0202)	(-0.0136, -0.0127)
RWE	(-0.0076, -0.0069)	<b>(0.0037, 0.0044)</b>	(-0.0140, -0.0132)	(-0.0027, -0.0019)	(-0.0177, -0.0168)	(-0.0063, -0.0056)
SZG	(-0.0195, -0.0183)	<b>(0.0087, 0.0101)</b>	(-0.0325, -0.0315)	(-0.0042, -0.0031)	(-0.0408, -0.0400)	(-0.0126, -0.0115)
SAP	(-0.0266, -0.0235)	<b>(0.0021, 0.0028)</b>	(-0.0354, -0.0321)	(-0.0068, -0.0059)	(-0.0381, -0.0347)	(-0.0094, -0.0085)
SIE	(-0.0132, -0.0120)	<b>(0.0023, 0.0029)</b>	(-0.0165, -0.0154)	(-0.0009, -0.0005)	(-0.0211, -0.0200)	(-0.0056, -0.0050)
TKA	(-0.0142, -0.0135)	<b>(0.0109, 0.0120)</b>	(-0.0262, -0.0257)	(-0.0011, -0.0003)	(-0.0332, -0.0326)	(-0.0079, -0.0074)
VOW	(-0.1565, -0.1307)	(-0.1339, -0.1065)	(-0.1873, -0.1595)	(-0.1647, -0.1351)	(-0.2025, -0.1729)	(-0.1797, -0.1485)

Table B.11.: Empirical percentiles (P/L),  $K = 1$ 

	Exceedances (VaR)						Exceedances (ES)					
	LLSA		MODWT		Median		LLSA		MODWT		Median	
ADS	173	(12.47%)	179	(12.91%)	185	(13.34%)	75	(5.41%)	77	(5.55%)	78	(5.62%)
ALV	179	(12.91%)	181	(13.05%)	184	(13.27%)	80	(5.77%)	<b>79</b>	<b>(5.70%)</b>	80	(5.77%)
BAS	154	(11.10%)	<b>143</b>	<b>(10.31%)</b>	<b>145</b>	<b>(10.45%)</b>	51	(3.68%)	44	(3.17%)	<b>52</b>	<b>(3.75%)</b>
BAY	187	(13.48%)	188	(13.55%)	191	(13.77%)	77	(5.55%)	<b>74</b>	<b>(5.34%)</b>	<b>75</b>	<b>(5.41%)</b>
BEI	168	(12.11%)	<b>166</b>	<b>(11.97%)</b>	<b>163</b>	<b>(11.75%)</b>	76	(5.48%)	77	(5.55%)	76	(5.48%)
BMW	174	(12.55%)	177	(12.76%)	<b>172</b>	<b>(12.40%)</b>	69	(4.97%)	70	(5.05%)	67	(4.83%)
CBK	164	(11.82%)	164	(11.82%)	165	(11.90%)	68	(4.90%)	66	(4.76%)	64	(4.61%)
DAI	187	(13.48%)	<b>183</b>	<b>(13.19%)</b>	<b>181</b>	<b>(13.05%)</b>	72	(5.19%)	<b>71</b>	<b>(5.12%)</b>	<b>71</b>	<b>(5.12%)</b>
DBK	170	(12.26%)	<b>165</b>	<b>(11.90%)</b>	175	(12.62%)	69	(4.97%)	69	(4.97%)	70	(5.05%)
DBI	153	(11.03%)	<b>152</b>	<b>(10.96%)</b>	<b>149</b>	<b>(10.74%)</b>	51	(3.68%)	50	(3.60%)	<b>52</b>	<b>(3.75%)</b>
DPB	143	(10.31%)	146	(10.53%)	149	(10.74%)	60	(4.33%)	<b>61</b>	<b>(4.40%)</b>	58	(4.18%)
DPW	178	(12.83%)	178	(12.83%)	180	(12.98%)	61	(4.40%)	<b>64</b>	<b>(4.61%)</b>	<b>69</b>	<b>(4.97%)</b>
DTE	193	(13.91%)	<b>186</b>	<b>(13.41%)</b>	<b>186</b>	<b>(13.41%)</b>	76	(5.48%)	<b>75</b>	<b>(5.41%)</b>	81	(5.84%)
EPC	205	(14.78%)	215	(15.50%)	207	(14.92%)	76	(5.48%)	81	(5.84%)	79	(5.70%)
FME	211	(15.21%)	<b>209</b>	<b>(15.07%)</b>	212	(15.28%)	93	(6.71%)	<b>91</b>	<b>(6.56%)</b>	<b>91</b>	<b>(6.56%)</b>
LHA	160	(11.54%)	<b>156</b>	<b>(11.25%)</b>	160	(11.54%)	61	(4.40%)	59	(4.25%)	61	(4.40%)
HNK	177	(12.76%)	178	(12.83%)	<b>173</b>	<b>(12.47%)</b>	79	(5.70%)	86	(6.20%)	84	(6.06%)
IFX	158	(11.39%)	158	(11.39%)	<b>150</b>	<b>(10.81%)</b>	59	(4.25%)	59	(4.25%)	58	(4.18%)
SDF	118	(8.51%)	<b>107</b>	<b>(7.71%)</b>	<b>113</b>	<b>(8.15%)</b>	41	(2.96%)	<b>43</b>	<b>(3.10%)</b>	<b>43</b>	<b>(3.10%)</b>
LIN	193	(13.91%)	196	(14.13%)	193	(13.91%)	87	(6.27%)	<b>86</b>	<b>(6.20%)</b>	<b>84</b>	<b>(6.06%)</b>
MAN	150	(10.81%)	154	(11.10%)	150	(10.81%)	56	(4.04%)	<b>58</b>	<b>(4.18%)</b>	<b>60</b>	<b>(4.33%)</b>
MRC	172	(12.40%)	<b>171</b>	<b>(12.33%)</b>	<b>170</b>	<b>(12.26%)</b>	67	(4.83%)	66	(4.76%)	<b>69</b>	<b>(4.97%)</b>
MEO	168	(12.11%)	174	(12.55%)	173	(12.47%)	73	(5.26%)	76	(5.48%)	<b>70</b>	<b>(5.05%)</b>
MUV	217	(15.65%)	<b>215</b>	<b>(15.50%)</b>	<b>211</b>	<b>(15.21%)</b>	78	(5.62%)	80	(5.77%)	81	(5.84%)
RWE	209	(15.07%)	216	(15.57%)	<b>206</b>	<b>(14.85%)</b>	83	(5.98%)	<b>81</b>	<b>(5.84%)</b>	84	(6.06%)
SZG	146	(10.53%)	<b>143</b>	<b>(10.31%)</b>	<b>142</b>	<b>(10.24%)</b>	60	(4.33%)	60	(4.33%)	60	(4.33%)
SAP	193	(13.91%)	194	(13.99%)	196	(14.13%)	78	(5.62%)	<b>76</b>	<b>(5.48%)</b>	79	(5.70%)
SIE	186	(13.41%)	189	(13.63%)	<b>180</b>	<b>(12.98%)</b>	70	(5.05%)	66	(4.76%)	71	(5.12%)
TKA	152	(10.96%)	156	(11.25%)	<b>150</b>	<b>(10.81%)</b>	66	(4.76%)	<b>69</b>	<b>(4.97%)</b>	<b>69</b>	<b>(4.97%)</b>
VOW	288	(20.76%)	292	(21.05%)	292	(21.05%)	130	(9.37%)	<b>129</b>	<b>(9.30%)</b>	130	(9.37%)

Table B.12.: Empirical percentiles (P/L),  $K = 2$ 

	Exceedances (VaR)						Exceedances (ES)					
	LLSA		MODWT		Median		LLSA		MODWT		Median	
ADS	179	(12.91%)	179	(12.91%)	185	(13.34%)	73	(5.26%)	77	(5.55%)	78	(5.62%)
ALV	180	(12.98%)	181	(13.05%)	184	(13.27%)	81	(5.84%)	<b>79</b>	<b>(5.70%)</b>	<b>80</b>	<b>(5.77%)</b>
BAS	150	(10.81%)	<b>143</b>	<b>(10.31%)</b>	<b>145</b>	<b>(10.45%)</b>	53	(3.82%)	44	(3.17%)	52	(3.75%)
BAY	192	(13.84%)	<b>188</b>	<b>(13.55%)</b>	<b>191</b>	<b>(13.77%)</b>	76	(5.48%)	<b>74</b>	<b>(5.34%)</b>	<b>75</b>	<b>(5.41%)</b>
BEI	168	(12.11%)	<b>166</b>	<b>(11.97%)</b>	<b>163</b>	<b>(11.75%)</b>	75	(5.41%)	77	(5.55%)	76	(5.48%)
BMW	176	(12.69%)	177	(12.76%)	<b>172</b>	<b>(12.40%)</b>	71	(5.12%)	<b>70</b>	<b>(5.05%)</b>	67	(4.83%)
CBK	163	(11.75%)	164	(11.82%)	165	(11.90%)	69	(4.97%)	66	(4.76%)	64	(4.61%)
DAI	186	(13.41%)	<b>183</b>	<b>(13.19%)</b>	<b>181</b>	<b>(13.05%)</b>	72	(5.19%)	<b>71</b>	<b>(5.12%)</b>	<b>71</b>	<b>(5.12%)</b>
DBK	168	(12.11%)	<b>165</b>	<b>(11.90%)</b>	175	(12.62%)	69	(4.97%)	69	(4.97%)	70	(5.05%)
DB1	154	(11.10%)	<b>152</b>	<b>(10.96%)</b>	<b>149</b>	<b>(10.74%)</b>	52	(3.75%)	50	(3.60%)	52	(3.75%)
DPB	143	(10.31%)	146	(10.53%)	149	(10.74%)	58	(4.18%)	<b>61</b>	<b>(4.40%)</b>	58	(4.18%)
DPW	177	(12.76%)	178	(12.83%)	180	(12.98%)	62	(4.47%)	<b>64</b>	<b>(4.61%)</b>	<b>69</b>	<b>(4.97%)</b>
DTE	190	(13.70%)	<b>186</b>	<b>(13.41%)</b>	<b>186</b>	<b>(13.41%)</b>	76	(5.48%)	<b>75</b>	<b>(5.41%)</b>	81	(5.84%)
EPC	201	(14.49%)	215	(15.50%)	207	(14.92%)	77	(5.55%)	81	(5.84%)	79	(5.70%)
FME	210	(15.14%)	<b>209</b>	<b>(15.07%)</b>	212	(15.28%)	93	(6.71%)	<b>91</b>	<b>(6.56%)</b>	<b>91</b>	<b>(6.56%)</b>
LHA	159	(11.46%)	<b>156</b>	<b>(11.25%)</b>	160	(11.54%)	58	(4.18%)	<b>59</b>	<b>(4.25%)</b>	<b>61</b>	<b>(4.40%)</b>
HNK	178	(12.83%)	178	(12.83%)	<b>173</b>	<b>(12.47%)</b>	80	(5.77%)	86	(6.20%)	84	(6.06%)
IFX	151	(10.89%)	158	(11.39%)	<b>150</b>	<b>(10.81%)</b>	62	(4.47%)	59	(4.25%)	58	(4.18%)
SDF	118	(8.51%)	<b>107</b>	<b>(7.71%)</b>	<b>113</b>	<b>(8.15%)</b>	41	(2.96%)	<b>43</b>	<b>(3.10%)</b>	<b>43</b>	<b>(3.10%)</b>
LIN	192	(13.84%)	196	(14.13%)	193	(13.91%)	87	(6.27%)	<b>86</b>	<b>(6.20%)</b>	<b>84</b>	<b>(6.06%)</b>
MAN	148	(10.67%)	154	(11.10%)	150	(10.81%)	56	(4.04%)	<b>58</b>	<b>(4.18%)</b>	<b>60</b>	<b>(4.33%)</b>
MRC	170	(12.26%)	171	(12.33%)	170	(12.26%)	66	(4.76%)	66	(4.76%)	<b>69</b>	<b>(4.97%)</b>
MEO	172	(12.40%)	174	(12.55%)	173	(12.47%)	74	(5.34%)	76	(5.48%)	<b>70</b>	<b>(5.05%)</b>
MUV	213	(15.36%)	215	(15.50%)	<b>211</b>	<b>(15.21%)</b>	78	(5.62%)	80	(5.77%)	81	(5.84%)
RWE	208	(15.00%)	216	(15.57%)	<b>206</b>	<b>(14.85%)</b>	84	(6.06%)	<b>81</b>	<b>(5.84%)</b>	84	(6.06%)
SZG	143	(10.31%)	143	(10.31%)	<b>142</b>	<b>(10.24%)</b>	59	(4.25%)	<b>60</b>	<b>(4.33%)</b>	<b>60</b>	<b>(4.33%)</b>
SAP	190	(13.70%)	194	(13.99%)	196	(14.13%)	78	(5.62%)	<b>76</b>	<b>(5.48%)</b>	79	(5.70%)
SIE	184	(13.27%)	189	(13.63%)	<b>180</b>	<b>(12.98%)</b>	69	(4.97%)	66	(4.76%)	71	(5.12%)
TKA	153	(11.03%)	156	(11.25%)	<b>150</b>	<b>(10.81%)</b>	65	(4.69%)	<b>69</b>	<b>(4.97%)</b>	<b>69</b>	<b>(4.97%)</b>
VOW	290	(20.91%)	292	(21.05%)	292	(21.05%)	130	(9.37%)	<b>129</b>	<b>(9.30%)</b>	130	(9.37%)

Table B.13.: Empirical percentiles (P/L),  $K = 3$ 

	Exceedances (VaR)						Exceedances (ES)					
	LLSA		MODWT		Median		LLSA		MODWT		Median	
ADS	182	(13.12%)	<b>179</b>	<b>(12.91%)</b>	185	(13.34%)	73	(5.26%)	77	(5.55%)	78	(5.62%)
ALV	181	(13.05%)	181	(13.05%)	184	(13.27%)	79	(5.70%)	79	(5.70%)	80	(5.77%)
BAS	155	(11.18%)	<b>143</b>	<b>(10.31%)</b>	<b>145</b>	<b>(10.45%)</b>	53	(3.82%)	44	(3.17%)	52	(3.75%)
BAY	192	(13.84%)	<b>188</b>	<b>(13.55%)</b>	<b>191</b>	<b>(13.77%)</b>	76	(5.48%)	<b>74</b>	<b>(5.34%)</b>	<b>75</b>	<b>(5.41%)</b>
BEI	168	(12.11%)	<b>166</b>	<b>(11.97%)</b>	<b>163</b>	<b>(11.75%)</b>	76	(5.48%)	77	(5.55%)	76	(5.48%)
BMW	175	(12.62%)	177	(12.76%)	<b>172</b>	<b>(12.40%)</b>	71	(5.12%)	<b>70</b>	<b>(5.05%)</b>	67	(4.83%)
CBK	166	(11.97%)	<b>164</b>	<b>(11.82%)</b>	<b>165</b>	<b>(11.90%)</b>	70	(5.05%)	66	(4.76%)	64	(4.61%)
DAI	188	(13.55%)	<b>183</b>	<b>(13.19%)</b>	<b>181</b>	<b>(13.05%)</b>	70	(5.05%)	71	(5.12%)	71	(5.12%)
DBK	167	(12.04%)	<b>165</b>	<b>(11.90%)</b>	175	(12.62%)	67	(4.83%)	<b>69</b>	<b>(4.97%)</b>	<b>70</b>	<b>(5.05%)</b>
DBI	152	(10.96%)	152	(10.96%)	<b>149</b>	<b>(10.74%)</b>	52	(3.75%)	50	(3.60%)	52	(3.75%)
DPB	142	(10.24%)	146	(10.53%)	149	(10.74%)	58	(4.18%)	<b>61</b>	<b>(4.40%)</b>	58	(4.18%)
DPW	177	(12.76%)	178	(12.83%)	180	(12.98%)	62	(4.47%)	<b>64</b>	<b>(4.61%)</b>	<b>69</b>	<b>(4.97%)</b>
DTE	188	(13.55%)	<b>186</b>	<b>(13.41%)</b>	<b>186</b>	<b>(13.41%)</b>	73	(5.26%)	75	(5.41%)	81	(5.84%)
EPC	201	(14.49%)	215	(15.50%)	207	(14.92%)	78	(5.62%)	81	(5.84%)	79	(5.70%)
FME	207	(14.92%)	209	(15.07%)	212	(15.28%)	90	(6.49%)	91	(6.56%)	91	(6.56%)
LHA	159	(11.46%)	<b>156</b>	<b>(11.25%)</b>	160	(11.54%)	60	(4.33%)	59	(4.25%)	<b>61</b>	<b>(4.40%)</b>
HNK	177	(12.76%)	178	(12.83%)	<b>173</b>	<b>(12.47%)</b>	81	(5.84%)	86	(6.20%)	84	(6.06%)
IFX	155	(11.18%)	158	(11.39%)	<b>150</b>	<b>(10.81%)</b>	61	(4.40%)	59	(4.25%)	58	(4.18%)
SDF	118	(8.51%)	<b>107</b>	<b>(7.71%)</b>	<b>113</b>	<b>(8.15%)</b>	41	(2.96%)	<b>43</b>	<b>(3.10%)</b>	<b>43</b>	<b>(3.10%)</b>
LIN	189	(13.63%)	196	(14.13%)	193	(13.91%)	87	(6.27%)	<b>86</b>	<b>(6.20%)</b>	<b>84</b>	<b>(6.06%)</b>
MAN	149	(10.74%)	154	(11.10%)	150	(10.81%)	56	(4.04%)	<b>58</b>	<b>(4.18%)</b>	<b>60</b>	<b>(4.33%)</b>
MRC	167	(12.04%)	171	(12.33%)	170	(12.26%)	66	(4.76%)	66	(4.76%)	<b>69</b>	<b>(4.97%)</b>
MEO	169	(12.18%)	174	(12.55%)	173	(12.47%)	74	(5.34%)	76	(5.48%)	<b>70</b>	<b>(5.05%)</b>
MUV	212	(15.28%)	215	(15.50%)	<b>211</b>	<b>(15.21%)</b>	79	(5.70%)	80	(5.77%)	81	(5.84%)
RWE	212	(15.28%)	216	(15.57%)	<b>206</b>	<b>(14.85%)</b>	84	(6.06%)	<b>81</b>	<b>(5.84%)</b>	84	(6.06%)
SZG	140	(10.09%)	143	(10.31%)	142	(10.24%)	59	(4.25%)	<b>60</b>	<b>(4.33%)</b>	<b>60</b>	<b>(4.33%)</b>
SAP	191	(13.77%)	194	(13.99%)	196	(14.13%)	76	(5.48%)	76	(5.48%)	79	(5.70%)
SIE	184	(13.27%)	189	(13.63%)	<b>180</b>	<b>(12.98%)</b>	69	(4.97%)	66	(4.76%)	71	(5.12%)
TKA	150	(10.81%)	156	(11.25%)	150	(10.81%)	66	(4.76%)	<b>69</b>	<b>(4.97%)</b>	<b>69</b>	<b>(4.97%)</b>
VOW	293	(21.12%)	<b>292</b>	<b>(21.05%)</b>	<b>292</b>	<b>(21.05%)</b>	130	(9.37%)	<b>129</b>	<b>(9.30%)</b>	130	(9.37%)

Table B.14.: VaR percentile exceedances (P/L)

	$K = 1$		$K = 2$		$K = 3$	
	VaR	LLSA	VaR	LLSA	VaR	LLSA
ADS	182 (13.12%)	173 (12.47%)	182 (13.12%)	179 (12.91%)	182 (13.12%)	182 (13.12%)
ALV	<b>178 (12.83%)</b>	179 (12.91%)	<b>178 (12.83%)</b>	180 (12.98%)	<b>178 (12.83%)</b>	181 (13.05%)
BAS	<b>141 (10.17%)</b>	154 (11.10%)	<b>141 (10.17%)</b>	150 (10.81%)	<b>141 (10.17%)</b>	155 (11.18%)
BAY	<b>186 (13.41%)</b>	187 (13.48%)	<b>186 (13.41%)</b>	192 (13.84%)	<b>186 (13.41%)</b>	192 (13.84%)
BEI	<b>166 (11.97%)</b>	168 (12.11%)	<b>166 (11.97%)</b>	168 (12.11%)	<b>166 (11.97%)</b>	168 (12.11%)
BMW	177 (12.76%)	174 (12.55%)	177 (12.76%)	176 (12.69%)	177 (12.76%)	175 (12.62%)
CBK	<b>162 (11.68%)</b>	164 (11.82%)	<b>162 (11.68%)</b>	163 (11.75%)	<b>162 (11.68%)</b>	166 (11.97%)
DAI	<b>183 (13.19%)</b>	187 (13.48%)	<b>183 (13.19%)</b>	186 (13.41%)	<b>183 (13.19%)</b>	188 (13.55%)
DBK	<b>169 (12.18%)</b>	170 (12.26%)	169 (12.18%)	168 (12.11%)	169 (12.18%)	167 (12.04%)
DB1	153 (11.03%)	153 (11.03%)	<b>153 (11.03%)</b>	154 (11.10%)	153 (11.03%)	152 (10.96%)
DPB	146 (10.53%)	143 (10.31%)	146 (10.53%)	143 (10.31%)	146 (10.53%)	142 (10.24%)
DPW	181 (13.05%)	178 (12.83%)	181 (13.05%)	177 (12.76%)	181 (13.05%)	177 (12.76%)
DTE	<b>187 (13.48%)</b>	193 (13.91%)	<b>187 (13.48%)</b>	190 (13.70%)	<b>187 (13.48%)</b>	188 (13.55%)
EPC	208 (15.00%)	205 (14.78%)	208 (15.00%)	201 (14.49%)	208 (15.00%)	201 (14.49%)
FME	<b>207 (14.92%)</b>	211 (15.21%)	<b>207 (14.92%)</b>	210 (15.14%)	207 (14.92%)	207 (14.92%)
LHA	<b>159 (11.46%)</b>	160 (11.54%)	159 (11.46%)	159 (11.46%)	159 (11.46%)	159 (11.46%)
HNK	179 (12.91%)	177 (12.76%)	179 (12.91%)	178 (12.83%)	179 (12.91%)	177 (12.76%)
IFX	163 (11.75%)	158 (11.39%)	163 (11.75%)	151 (10.89%)	163 (11.75%)	155 (11.18%)
SDF	<b>112 (8.07%)</b>	118 (8.51%)	<b>112 (8.07%)</b>	118 (8.51%)	<b>112 (8.07%)</b>	118 (8.51%)
LIN	194 (13.99%)	193 (13.91%)	194 (13.99%)	192 (13.84%)	194 (13.99%)	189 (13.63%)
MAN	150 (10.81%)	150 (10.81%)	150 (10.81%)	148 (10.67%)	150 (10.81%)	149 (10.74%)
MRC	172 (12.40%)	172 (12.40%)	172 (12.40%)	170 (12.26%)	172 (12.40%)	167 (12.04%)
MEO	172 (12.40%)	168 (12.11%)	172 (12.40%)	172 (12.40%)	172 (12.40%)	169 (12.18%)
MUV	<b>215 (15.50%)</b>	217 (15.65%)	215 (15.50%)	213 (15.36%)	215 (15.50%)	212 (15.28%)
RWE	209 (15.07%)	209 (15.07%)	209 (15.07%)	208 (15.00%)	<b>209 (15.07%)</b>	212 (15.28%)
SZG	151 (10.89%)	146 (10.53%)	151 (10.89%)	143 (10.31%)	151 (10.89%)	140 (10.09%)
SAP	197 (14.20%)	193 (13.91%)	197 (14.20%)	190 (13.70%)	197 (14.20%)	191 (13.77%)
SIE	191 (13.77%)	186 (13.41%)	191 (13.77%)	184 (13.27%)	191 (13.77%)	184 (13.27%)
TKA	152 (10.96%)	152 (10.96%)	<b>152 (10.96%)</b>	153 (11.03%)	152 (10.96%)	150 (10.81%)
VOW	<b>274 (19.75%)</b>	288 (20.76%)	<b>274 (19.75%)</b>	290 (20.91%)	<b>274 (19.75%)</b>	293 (21.12%)

Table B.15.: ES percentile exceedances (P/L)

	$K = 1$		$K = 2$		$K = 3$	
	ES	LLSA	ES	LLSA	ES	LLSA
ADS	76 (5.48%)	75 (5.41%)	76 (5.48%)	73 (5.26%)	76 (5.48%)	73 (5.26%)
ALV	<b>79 (5.70%)</b>	80 (5.77%)	<b>79 (5.70%)</b>	81 (5.84%)	79 (5.70%)	79 (5.70%)
BAS	46 (3.32%)	51 (3.68%)	46 (3.32%)	53 (3.82%)	46 (3.32%)	53 (3.82%)
BAY	<b>76 (5.48%)</b>	77 (5.55%)	76 (5.48%)	76 (5.48%)	76 (5.48%)	76 (5.48%)
BEI	76 (5.48%)	76 (5.48%)	76 (5.48%)	75 (5.41%)	76 (5.48%)	76 (5.48%)
BMW	67 (4.83%)	69 (4.97%)	67 (4.83%)	71 (5.12%)	67 (4.83%)	71 (5.12%)
CBK	66 (4.76%)	68 (4.90%)	66 (4.76%)	69 (4.97%)	66 (4.76%)	70 (5.05%)
DAI	75 (5.41%)	72 (5.19%)	75 (5.41%)	72 (5.19%)	75 (5.41%)	70 (5.05%)
DBK	72 (5.19%)	69 (4.97%)	72 (5.19%)	69 (4.97%)	72 (5.19%)	67 (4.83%)
DBI	50 (3.60%)	51 (3.68%)	50 (3.60%)	52 (3.75%)	50 (3.60%)	52 (3.75%)
DPB	58 (4.18%)	60 (4.33%)	58 (4.18%)	58 (4.18%)	58 (4.18%)	58 (4.18%)
DPW	<b>66 (4.76%)</b>	61 (4.40%)	<b>66 (4.76%)</b>	62 (4.47%)	<b>66 (4.76%)</b>	62 (4.47%)
DTE	78 (5.62%)	76 (5.48%)	78 (5.62%)	76 (5.48%)	78 (5.62%)	73 (5.26%)
EPC	76 (5.48%)	76 (5.48%)	<b>76 (5.48%)</b>	77 (5.55%)	<b>76 (5.48%)</b>	78 (5.62%)
FME	<b>91 (6.56%)</b>	93 (6.71%)	<b>91 (6.56%)</b>	93 (6.71%)	91 (6.56%)	90 (6.49%)
LHA	61 (4.40%)	61 (4.40%)	<b>61 (4.40%)</b>	58 (4.18%)	<b>61 (4.40%)</b>	60 (4.33%)
HNK	79 (5.70%)	79 (5.70%)	<b>79 (5.70%)</b>	80 (5.77%)	<b>79 (5.70%)</b>	81 (5.84%)
IFX	<b>60 (4.33%)</b>	59 (4.25%)	60 (4.33%)	62 (4.47%)	60 (4.33%)	61 (4.40%)
SDF	40 (2.88%)	41 (2.96%)	40 (2.88%)	41 (2.96%)	40 (2.88%)	41 (2.96%)
LIN	<b>82 (5.91%)</b>	87 (6.27%)	<b>82 (5.91%)</b>	87 (6.27%)	<b>82 (5.91%)</b>	87 (6.27%)
MAN	<b>59 (4.25%)</b>	56 (4.04%)	<b>59 (4.25%)</b>	56 (4.04%)	<b>59 (4.25%)</b>	56 (4.04%)
MRC	62 (4.47%)	67 (4.83%)	62 (4.47%)	66 (4.76%)	62 (4.47%)	66 (4.76%)
MEO	75 (5.41%)	73 (5.26%)	75 (5.41%)	74 (5.34%)	75 (5.41%)	74 (5.34%)
MUV	83 (5.98%)	78 (5.62%)	83 (5.98%)	78 (5.62%)	83 (5.98%)	79 (5.70%)
RWE	83 (5.98%)	83 (5.98%)	<b>83 (5.98%)</b>	84 (6.06%)	<b>83 (5.98%)</b>	84 (6.06%)
SZG	<b>61 (4.40%)</b>	60 (4.33%)	<b>61 (4.40%)</b>	59 (4.25%)	<b>61 (4.40%)</b>	59 (4.25%)
SAP	<b>76 (5.48%)</b>	78 (5.62%)	<b>76 (5.48%)</b>	78 (5.62%)	76 (5.48%)	76 (5.48%)
SIE	67 (4.83%)	70 (5.05%)	67 (4.83%)	69 (4.97%)	67 (4.83%)	69 (4.97%)
TKA	64 (4.61%)	66 (4.76%)	64 (4.61%)	65 (4.69%)	64 (4.61%)	66 (4.76%)
VOW	<b>128 (9.23%)</b>	130 (9.37%)	<b>128 (9.23%)</b>	130 (9.37%)	<b>128 (9.23%)</b>	130 (9.37%)



# Bibliography

- [1] F. Abramovich, T.C. Bailey, and T. Sapatinas. Wavelet analysis and its statistical applications. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 49(1):1–29, 2000.
- [2] C. Acerbi. Coherent representations of subjective risk-aversion. *Risk measures for the 21st century*, pages 147–207, 2004.
- [3] C. Acerbi and D. Tasche. Expected Shortfall: a natural coherent alternative to Value at Risk. *Economic Notes*, 31(2):379–388, 2002.
- [4] A. Antoniadis. Wavelets in statistics: a review. *Statistical Methods and Applications*, 6(2):97–130, 1997.
- [5] A. Antoniadis, J. Bigot, and T. Sapatinas. Wavelet estimators in nonparametric regression: a comparative simulation study. *Journal of Statistical Software*, 6(6):1–83, 2001.
- [6] A. Antoniadis, D. Leporini, and J.C. Pesquet. Wavelet thresholding for some classes of non-Gaussian noise. *Statistica neerlandica*, 56(4):434–453, 2002.
- [7] P. Artzner, F. Delbaen, J.M. Eber, and D. Heath. Thinking coherently. *Risk*, 10(11):68–71, 1997.
- [8] P. Artzner, F. Delbaen, J.M. Eber, and D. Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.
- [9] J. Astola and P. Kuosmanen. *Fundamentals of nonlinear filtering*. CRC Press, 1997.

- [10] R. Averkamp and C. Houdré. Wavelet thresholding for non-necessarily Gaussian noise: idealism. *Annals of Statistics*, 31(1):110–151, 2003.
- [11] R. Averkamp and C. Houdré. Wavelet thresholding for non-necessarily Gaussian noise: functionality. *Annals of Statistics*, 33(5):2164–2193, 2005.
- [12] G. Bianchi and R. Sorrentino. *Electronic filter simulation & design*. McGraw-Hill, 2007.
- [13] M. Bianchi, M. Boyle, and D. Hollingsworth. A comparison of methods for trend estimation. *Applied Economics Letters*, 6(2):103–109, 1999.
- [14] T. Bjoerk. *Arbitrage theory in continuous time*. Oxford Universal Press, 2009.
- [15] F. Black. The pricing of commodity contracts. *Journal of financial economics*, 3(1-2):167–179, 1976.
- [16] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3), 1973.
- [17] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.
- [18] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time series analysis: forecasting and control*. Holden-day San Francisco, 1970.
- [19] E.O. Brigham and RE Morrow. The fast Fourier transform. *Spectrum, IEEE*, 4(12):63–70, 2009.
- [20] D.R. Brillinger. *Time series: data analysis and theory*. Society for Industrial Mathematics, 2001.
- [21] P.J. Brockwell and R.A. Davis. *Introduction to time series and forecasting*. Springer Verlag, 2002.
- [22] P.J. Brockwell and R.A. Davis. *Time series: theory and methods*. Springer Verlag, 2009.

- 
- [23] C. Brooks. *Introductory econometrics for finance*. Cambridge University Press, 2008.
- [24] A. Bruce, H.Y. Gao, and A. Bruce. *Applied wavelet analysis with S-plus*. Springer New York, 1996.
- [25] E. Cantoni and E. Ronchetti. Resistant selection of the smoothing parameter for smoothing splines. *Statistics and Computing*, 11(2):141–146, 2001.
- [26] E. Carlstein, H.G. Mueller, and D. Siegmund. *Change-point problems*. Institute of Mathematical Statistics Hayward, CA, 1994.
- [27] C. Chatfield. Apples, oranges and mean square error. *International Journal of Forecasting*, 4(4):515–518, 1988.
- [28] C. Chatfield. *The analysis of time series: an introduction*. CRC press, 2004.
- [29] A.J. Conejo, M.A. Plazas, R. Espinola, and A.B. Molina. Day-ahead electricity price forecasting using the wavelet transform and ARIMA models. *Power Systems, IEEE Transactions on*, 20(2):1035–1042, 2005.
- [30] J. Connor and R. Rossiter. Wavelet transforms and commodity prices. *Studies in Nonlinear Dynamics & Econometrics*, 9(1):433–465, 2005.
- [31] M.M. Dacorogna, R. Gençay, U. Mueller, R.B. Olsen, and O.V. Pictet. *An introduction to high-frequency finance*. Academic Press, 2001.
- [32] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial Mathematics, 1992.
- [33] A.C. Davison and D.V. Hinkley. *Bootstrap methods and their application*. Cambridge University Press, 1999.
- [34] A. Dermoune, B. Djehiche, and N. Rahmania. Consistent estimators of the smoothing parameter in the hodrick-prescott filter. *Journal of the Japan Statistical Society*, 38:225–241, 2008.
- [35] D.L. Donoho and I.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

- [36] D.L. Donoho and I.M. Johnstone. Adapting to Unknown Smoothness Via Wavelet Shrinkage. *Journal of the American Statistical Association*, 90(432), 1995.
- [37] D.L. Donoho and I.M. Johnstone. Minimax estimation via wavelet shrinkage. *Annals of Statistics*, pages 879–921, 1998.
- [38] K. Dowd. *Measuring market risk*. John Wiley & Sons Inc, 2002.
- [39] R.F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the Econometric Society*, 50(4):987–1007, 1982.
- [40] R.L. Eubank. *Spline smoothing and nonparametric regression*. M. Dekker New York, 1988.
- [41] J. Fan and I. Gijbels. *Local polynomial modelling and its applications*. Chapman & Hall/CRC, 1996.
- [42] J. Fan and Y. Wang. Multi-scale jump and volatility analysis for high-frequency financial data. *Journal of the American Statistical Association*, 102(480):1349–1362, 2007.
- [43] J. Fan and Q. Yao. *Nonlinear time series: Nonparametric and parametric methods*. Springer Verlag, 2003.
- [44] P. Fryzlewicz, S. Van Bellegem, and R. Von Sachs. Forecasting non-stationary time series by wavelet process modelling. *Annals of the Institute of Statistical Mathematics*, 55(4):737–764, 2003.
- [45] I. Gijbels, A. Lambert, and P. Qiu. Jump-preserving regression and smoothing using local linear fitting: a compromise. *Annals of the Institute of Statistical Mathematics*, 59(2):235–272, 2007.
- [46] P.J. Green and BW Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach*. Chapman & Hall/CRC, 1994.
- [47] M.S. Grewal, A.P. Andrews, and Ebooks Corporation. *Kalman filtering: theory*

- 
- and practice using MATLAB*. Wiley Online Library, 2001.
- [48] H. Guo. *Theory and applications of the shift-invariant, time-varying and undecimated wavelet transforms*. PhD thesis, Citeseer, 1995.
- [49] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.
- [50] J.D. Hamilton. *Time series analysis*. Princeton University Press, 1994.
- [51] M.E. Hamrita, B. Abdallah, B. Ammou, et al. The Multi-Scale Interaction between Interest Rate, Exchange Rate and Stock Price. 2009.
- [52] A. Harvey and T. Trimbur. Trend estimation and the Hodrick-Prescott filter. *Journal of the Japan Statistical Society*, 38(1):41–49, 2008.
- [53] C. Harvey, Andrew. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.
- [54] M. Higo and S.K. Nakada. How can we extract a fundamental trend from an economic time-series? *Monetary and Economic Studies*, 16(2):61–111, 1998.
- [55] R.J. Hodrick and E.C. Prescott. Postwar US Business Cycles: An Empirical Investigation. *Journal of Money, Credit & Banking*, 29(1), 1997.
- [56] R. Hoppe. VaR and the unreal world. *Risk*, 11(7):45–50, 1998.
- [57] J. Hull. *Options, futures and other derivatives*. Pearson Prentice Hall, 2009.
- [58] C.M. Hurvich, J.S. Simonoff, and C.L. Tsai. Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):271–293, 1998.
- [59] R.J. Hyndman and A.B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.
- [60] R.A. Irizarry. Choosing smoothness parameters for smoothing splines by minimiz-

- ing an estimate of risk. *Johns Hopkins University, Dept. of Biostatistics Working Papers*, page 30, 2004.
- [61] A.J. Jerri. *The Gibbs phenomenon in Fourier analysis, splines, and wavelet approximations*. Springer, 1998.
- [62] F.T. Johnsen, T. Hafsoe, C. Griwodz, and P. Halvorsen. Workload characterization for news-on-demand streaming services. In *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE International*, pages 314–323. IEEE, 2007.
- [63] I.M. Johnstone and B.W. Silverman. Wavelet threshold estimators for data with correlated noise. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(2):319–351, 1997.
- [64] J.H. Joo and P. Qiu. Jump detection in a regression curve and its derivative. *Technometrics*, 51(3):289–305, 2009.
- [65] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, volume 3, page 26. Citeseer, 1997.
- [66] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [67] C.S. Kim and J.S. Marron. SiZer for jump detection. *Journal of nonparametric statistics*, 18(1):13–20, 2006.
- [68] P.H. Kupiec and Board of Governors of the Federal Reserve System (US). Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, 3(2):73–84, 1995.
- [69] T. Lee. Smoothing parameter selection for smoothing splines: a simulation study. *Computational statistics & Data analysis*, 42(1-2):139–148, 2003.
- [70] CEV Leser. A simple method of trend construction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 23(1):91–107, 1961.

- [71] A.W. Lo. Long-term memory in stock market prices. *Econometrica: Journal of the Econometric Society*, 59(5):1279–1313, 1991.
- [72] J.J. Lucia and E.S. Schwartz. Electricity prices and power derivatives: Evidence from the nordic power exchange. *Review of Derivatives Research*, 5(1):5–50, 2002.
- [73] H. Luetkepohl and F. Xu. The role of the log transformation in forecasting economic variables. Technical report, 2009.
- [74] S.G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.
- [75] S.G. Mallat and W.L. Hwang. Singularity detection and processing with wavelets. *IEEE transactions on information theory*, 38(2):617–643, 1992.
- [76] Y. Meyer. Wavelets-Algorithms and applications. *Applied Mathematics*, 1993.
- [77] Y. Meyer and D.H. Salinger. *Wavelets and operators*. Cambridge University Press, 1995.
- [78] S.G. Mohinder and P.A. Angus. *Kalman filtering theory and practice*. Prentice hall, 1993.
- [79] M.F. Mohr. A trend-cycle (-season) filter. *Working Paper Series European Central Bank*, (499), 2005.
- [80] A. Mojsilovic, M. Popovic, S. Markovic, and M. Krstic. Characterization of visually similar diffuse diseases from B-scan liver images using nonseparable wavelet transform. *Medical Imaging, IEEE Transactions on*, 17(4):541–549, 2002.
- [81] J. Morlet. Sampling theory and wave propagation. *Issues in acoustic signal/image processing and recognition*, 1:233–261, 1983.
- [82] R. Morton, E.L. Kang, and B.L. Henderson. Smoothing splines for trend estimation and prediction in time series. *Environmetrics*, 20(3):249–259, 2009.
- [83] GP Nason. Wavelet shrinkage using cross-validation. *Journal of the Royal Statis-*

- tical Society. Series B (Methodological)*, 58(2):463–479, 1996.
- [84] GP Nason, T. Sapatinas, and A. Sawczenko. Statistical modelling of time series using non-decimated wavelet representations. *Preprint, University of Bristol*, 1999.
- [85] M.H. Neumann and R. Von Sachs. Wavelet thresholding: beyond the Gaussian iid situation. In *Lect. Notes Statist.* Citeseer, 1995.
- [86] S. Peltonen, M. Gabbouj, and J. Astola. Nonlinear filter design: methodologies and challenges. In *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, 2001. ISPA 2001*, pages 102–107, 2001.
- [87] D.B. Percival and H.O. Mofjeld. Analysis of Subtidal Coastal Sea Level Fluctuations Using Wavelets. *Journal of the American Statistical Association*, 92(439), 1997.
- [88] D.B. Percival and A.T. Walden. *Wavelet methods for time series analysis*. Cambridge University Press, 2006.
- [89] D.S.G. Pollock. Trend estimation and de-trending via rational square-wave filters. *Journal of Econometrics*, 99(2):317–334, 2000.
- [90] D.S.G. Pollock. Methodology for trend estimation. *Economic Modelling*, 18(1):75–96, 2001.
- [91] S. Pollock and I.L. Cascio. Non-Dyadic Wavelet Analysis. *Optimisation, Econometric and Financial Analysis*, pages 167–203, 2007.
- [92] P. Qiu. A jump-preserving curve fitting procedure based on local piecewise-linear kernel estimation. *Journal of Nonparametric Statistics*, 15(4):437–453, 2003.
- [93] P. Qiu and B. Yandell. A local polynomial jump-detection algorithm in nonparametric regression. *Technometrics*, 40(2):141–152, 1998.
- [94] S.T. Rachev. *Handbook of heavy tailed distributions in finance*. North-Holland, 2003.
- [95] S.T. Rachev and S. Mittnik. *Stable Paretian models in finance*. John Wiley &

- Sons Inc, 2000.
- [96] M. Raimondo. Minimax estimation of sharp change points. *Annals of Statistics*, 26(4):1379–1397, 1998.
- [97] M. Raimondo and N. Tajvidi. A peaks over threshold model for change-point detection by wavelets. *Statistica Sinica*, 14(2):395–412, 2004.
- [98] J.B. Ramsey. The contribution of wavelets to the analysis of economic and financial data. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 357(1760):2593–2606, 1999.
- [99] J.B. Ramsey. Wavelets in economics and finance: Past and future. *Studies in Nonlinear Dynamics & Econometrics*, 6(3):1–27, 2002.
- [100] M.O. Ravn and H. Uhlig. On adjusting the Hodrick-Prescott filter for the frequency of observations. *Review of Economics and Statistics*, 84(2):371–376, 2002.
- [101] O. Renaud, F. Murtagh, and J.L. Starck. *Wavelet-based forecasting of short and long memory time series*. Université de Genève, Faculté des sciences économiques et sociales, Département d'économétrie, 2002.
- [102] E. Schlicht. Estimating the smoothing parameter in the so-called Hodrick-Prescott filter. *Journal of the Japan Statistical Society*, 35(1):99–119, 2005.
- [103] M.A. Schulze. An edge-enhancing nonlinear filter for reducing multiplicative noise. In *Proc. SPIE*, volume 3026, pages 46–56. Citeseer, 1997.
- [104] J. Seifert and M. Uhrig-Homburg. Modelling jumps in electricity prices: theory and empirical evidence. *Review of Derivatives Research*, 10(1):59–85, 2007.
- [105] H.T. Shim and H. Volkmer. On the Gibbs phenomenon for wavelet expansions. *Journal of approximation theory*, 84(1):74–95, 1996.
- [106] C.M. Stein. Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9(6):1135–1151, 1981.
- [107] J. Sun and P. Qiu. Jump detection in regression surfaces using both first-order

- and second-order derivatives. *Journal of Computational and Graphical Statistics*, 16(2):289–311, 2007.
- [108] W. Sun, S. Rachev, and F.J. Fabozzi. Fractals or IID: evidence of long-range dependence and heavy tailedness from modeling German equity market returns. *Journal of Economics and Business*, 59(6):575–595, 2007.
- [109] W. Sun, S. Rachev, F.J. Fabozzi, and P.S. Kalem. Fractals in trade duration: capturing long-range dependence and heavy tailedness in modeling trade duration. *Annals of Finance*, 4(2):217–241, 2008.
- [110] N. Taleb. Against VaR. *Derivatives Strategy*, 2:21–26, 1997.
- [111] N. Taleb. The world according to Nassim Taleb. *Derivatives Strategy*, 1, 1997.
- [112] S.J. Taylor. *Modelling financial time series*. World Scientific Publishing Co., Inc, 2008.
- [113] R.S. Tsay. *Analysis of financial time series*. Wiley-Interscience, 2005.
- [114] G. Wahba. *Spline models for observational data*. Society for Industrial Mathematics, 1990.
- [115] Y. Wang. Jump and sharp cusp detection by wavelets. *Biometrika*, 82(2):385–397, 1995.
- [116] Y. Wang. Change Curve Estimation Via Wavelets. *Journal of the American Statistical Association*, 93(441):163–172, 1998.
- [117] G. Welch and G. Bishop. An introduction to the Kalman filter. *University of North Carolina at Chapel Hill, Chapel Hill, NC*, 1995.
- [118] P. Wojtaszczyk. *A mathematical introduction to wavelets*. Cambridge University Press, 1997.
- [119] X. Xiong, X.T. Zhang, W. Zhang, and C.Y. Li. Wavelet-based beta estimation of China stock market. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 6, pages 3501–3505. IEEE, 2005.

- [120] S. Yousefi, I. Weinreich, and D. Reinarz. Wavelet-based prediction of oil prices. *Chaos, Solitons & Fractals*, 25(2):265–275, 2005.
- [121] H. Yu, D. Zheng, B.Y. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. *ACM SIGOPS Operating Systems Review*, 40(4):333–344, 2006.
- [122] L. Yuan and X. Zhongjie. The wavelet detection of the jump and cusp points of a regression function. *Acta Mathematicae Applicatae Sinica (English Series)*, 16(3):283–291, 2000.



## Affidavit

I hereby affirm truthfully that this thesis has been written only by the undersigned and without any assistance from third parties. Furthermore, I confirm that no resources have been used in the preparation of this thesis other than those indicated in the thesis itself, including quoted and adapted contents of publications from other authors and myself.

## Eidesstattliche Erklärung

Ich versichere wahrheitsgemäß, die Dissertation bis auf die in der Abhandlung angegebene Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und genau kenntlich gemacht zu haben, was aus Arbeiten anderer und aus eigenen Veröffentlichungen unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, January 2011

---

(Thomas Meinl)