



NTNU – Trondheim
Norwegian University of
Science and Technology

Wavelets and irregular time series

Børge Solli Andreassen

Master of Science in Physics and Mathematics

Submission date: June 2012

Supervisor: Yurii Lyubarskii, MATH

Co-supervisor: Pål Navestad, ConocoPhillips

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

In this thesis we study time series containing pressure measurements from a three phase flow pipeline at the Ekofisk oil field. The pipeline transports a mixture of oil, water and gas from 15 wells for approximately 2.5km to a production facility. Our aim is to develop techniques that allow the selection and (to some extent) prediction of "non-standard" behavior in the system (sharp pressure changes and other type of instabilities).

To advice this aim we perform a scalewise decomposition of the input signal/time series and investigate the behavior of each scale separately. We introduce the Sliding Window Wavelet Transform (SWWT) method. The method evaluate the variability on different scales within the time interval of a characteristic length (a window) and then trace these characteristics as the window slides in time.

We use the discrete wavelet transform (DWT) in order to obtain the scalewise decomposition within the window. Using orthonormal discrete wavelets, we show that the variability of such sequences can be decomposed into their corresponding scales.

Based on this, a thresholding algorithm is applied, characterizing the state of the system any given time. The results we find are promising and we show that different parameters in the thresholding algorithm extracts different types of special events. We also show that in some cases, this approach allows to predict special events before they really occur.

While we investigate one particular system in this thesis, the procedures developed can be applied to other complicated systems where instability in system parameters is important.

Sammendrag

I denne oppgaven studerer vi tidsrekker med trykkmålinger fra Ekofiskfeltet i Nordsjøen. Målingene kommer fra transport (av en blanding med olje, gass og vann) fra 15 brønner, gjennom en 2,5 kilometer lang rørledning. Rørledningen ender på en produksjonsplattform, hvor olje, vann og gass separeres. Målet med denne oppgaven er å utvikle teknikker som kan oppdage og (til en viss grad) forutsi systemoppørsel som avviker fra det normale (raske trykkforandringer og annen ustabil oppførsel).

Vi dekomponerer måledataene i komponenter på ulike skalaer for å undersøke disse hver for seg. Vi introduserer så sliding window wavelet transform (SWWT). Metoden måler variabiliteten på de ulike skalaene innenfor et tidsintervall av en karakteristisk lengde (et vindu) og analyserer hvordan den endrer seg over tid.

Vi bruker discrete wavelet transform (DWT) for å dekomponere signalet avgrenset innenfor et slikt vindu. Ved å bruke ortonormale wavelet-funksjoner viser vi at variansen til det avgrensede signalet kan dekomponeres med hensyn på skala.

Ved hjelp av disse redskapene definerer vi en thresholding-algoritme som karakteriserer tidsintervaller hvor signalet er ustabil. Vi oppnår gode resultater. Blant annet viser vi at ulike parametere i algoritmen muliggjør karakterisering av ulike systemoppørsel. Vi viser også at det i noen tilfeller, er det mulig å forutsi hendelser før de inntreffer.

Selv om vi i oppgaven kun analyserer måledata fra et system, kan metodene vi utvikler anvendes på andre systemer hvor variabilitet av måledata er viktig.

Preface

This thesis is submitted in fulfillment of my master's degree in Industrial Mathematics at the Department of Mathematical Sciences at the Norwegian University of Science and Technology, NTNU. The thesis concludes a five year program in mathematics and physics, with a specialization in applied mathematics. This final work is done in cooperation with ConocoPhillips with the purpose of solving a real life industry problem.

I would like to thank my supervisors, professor Yuriy Lyubarskii, NTNU, and Pål Navestad, ConocoPhillips.

Yurii: Thanks for the many hours spent helping me to understand the theoretical background and for helping me to pace out the right direction. Through guidance, discussions and corrections your help has been invaluable for this text and I really enjoyed the process!

Pål: Thank you for all guidance and your explanations around the workings of the system. The workshop in Stavanger was also valuable for the progress of the thesis. Further, thanks for commenting on and suggesting improvements to the draft of this thesis.

I would also like to thank Tatyana Kholodna, Capgemini, for helping me whenever I needed answers during the semester. I really enjoyed working with you, Pål and Yurii during the semester.

I owe a lot of gratitude to my family and my girlfriend for support and patience during the hectic periods of writing this thesis, thank you!

Lastly I would like to thank my friends, especially my friends at the Department for Mathematical Sciences for making my stay at NTNU the greatest time of my life.

Trondheim, June 2012

Børge Solli Andreassen

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Available data | 2 |
| 1.2 | The system | 2 |
| 1.2.1 | Ekofisk | 2 |
| 1.2.2 | Description of the system | 4 |
| 1.2.3 | Behavior of the system | 5 |
| 1.3 | Datasets | 9 |
| 1.3.1 | Example intervals | 11 |
| 1.4 | Thesis structure | 13 |
| 1.4.1 | A brief overview | 14 |
| 1.4.2 | Description of the methods | 15 |
| 2 | Mathematical background | 17 |
| 2.1 | The continuous wavelet transform | 18 |
| 2.2 | The discrete wavelet transform | 21 |
| 2.2.1 | Orthonormal discrete wavelet | 23 |
| 2.2.2 | Filter bank algorithm | 30 |
| 2.3 | Dealing with finite signals | 31 |
| 2.3.1 | Periodization of the transform | 32 |
| 2.3.2 | Periodized cascading algorithm | 37 |

| | | |
|----------|---|-----------|
| 2.3.3 | Initialization | 38 |
| 2.3.4 | Estimate of error | 40 |
| 2.4 | Wavelet variance | 45 |
| 2.5 | Sliding Window Wavelet Transform | 48 |
| 2.5.1 | Sliding window wavelet transform | 49 |
| 2.5.2 | Smoothing SWWT | 49 |
| 2.5.3 | Position of the sliding window | 50 |
| 2.5.4 | Choice of scales | 51 |
| 2.5.5 | SWWT's effect on error | 52 |
| 2.6 | Thresholding algorithm | 54 |
| 2.6.1 | Algorithm parameters | 54 |
| 2.7 | Time series preparation | 57 |
| 3 | Results | 59 |
| 3.1 | Detection and signal characterization | 59 |
| 3.2 | Prediction | 66 |
| 4 | Concluding remarks | 71 |
| 4.1 | Further work | 71 |
| | Bibliography | 74 |

Notation

| | |
|---|--|
| $\mathcal{W}f(u, s)$ | Wavelet transform of f |
| $\mathcal{W}f(j, n)$ | Discrete wavelet transform of f : $s = 2^j$ and $u_j = 2^j n$ |
| $\hat{f}(\omega)$ | Fourier transform |
| $\mathcal{S}_{J,L}$ | Sliding window wavelet transform (SWWT) |
| $\mathcal{S}_{J,L}^{\mathcal{B}}$ | SWWT convolved with window \mathcal{B} |
| $\mathcal{T}_{\alpha,\delta}$ | Thresholding test with weighing α and threshold δ |
| \mathbf{V}_j | Coarse scale j , see section 2.2.1 |
| \mathbf{W}_j | Detailed scale j , see section 2.2.1 |
| ψ, ψ^p | Wavelet function and its periodization over $[0, 1]$ |
| ϕ, ϕ^p | Scaling function and its periodization over $[0, 1]$ |
| a_j, d_j | Scaling and wavelet coefficients of scale j |
| a_j^p, d_j^p | Periodized scaling and wavelet coefficients of scale j , obtained respectively with ϕ_j^p and ψ_j^p |
| ν_j^2 and $\tilde{\nu}_j^2$ | Wavelet and scaling variance on scale j |
| $b \in \ell^2$ | Sequence of length $N = 2^{-J}$ containing system measurements, $b[n] \approx f(t_0 + n)$, $n = 1, \dots, N$, for some initial time, t_0 |
| $g(\tau) = f(t_0 + N\tau)$ | Change of variable, obtaining $\tau \in [0, 1]$ from $t \in [t_0, t_0 + N]$, used when relating b to scaling coefficient of a function |
| $X^{t_0} \equiv (b^{t_m})^p$ | Notation used for periodized b , location in time is specified |
| $V_j^{t_0}, W_j^{t_0}$ | a_j^p and d_j^p with t_0 explicitly given s.t. $b[n] \approx f(t_0 + n)$ |
| $\langle f_1, f_2 \rangle_{\mathbf{A}}$ | Inner product on \mathbf{A} , \mathbf{A} can be $\{L^2(\mathbb{R}), L^2[0, 1], \ell^2\}$ |
| $f_1 * f_2$ | Convolution between f_1 and f_2 |
| $f_1 \otimes f_2$ | Circular convolution between f_1 and f_2 |
| f^p | Periodization of $f \in L^2(\mathbb{R})$ to the interval $[0, 1]$ |
| f_p | Periodic extension of $f \in L^2[0, 1]$ to \mathbb{R} |

Chapter 1

Introduction

In this thesis, we study time series containing measurements from a complicated system. The main purpose is to detect and perhaps predict time intervals with "non-standard" behavior of the system.

The particular system we will be analyzing is related to one of the pipe lines at the *Ekofisk* oil field in the North Sea. In this system, "non-standard", or special events, are periods of time where the system is not operating with a stable pressure. We will come back to this later in the text.

Being able to automatically detect special system behavior in general can enable system operators to act faster when problems arise. As a data analysis tool, system characterization based on time series could also contribute to developing a better understanding about the problems and the system as a whole. It could also give valuable insight on how to avoid the problem in the future system designs.

Clearly, different types of systems demands a different approach, implying that at least an informal understanding about how the particular system functions is necessary. When a system has a certain level of complexity, it can be virtually impossible to model it precisely. In such cases, a statistical approach can be of great help. With such an approach

it is natural to face the problem with a combination of empirical and theoretical methods.

While this thesis investigate one particular system, the procedures developed can be applied to other complicated systems where instability in system parameters is important.

1.1 Available data

The amount of available data for analysis is growing rapidly in the oil industry. However, the available data is not always clear cut, and there is a present need for techniques that could extract vital information from the huge amount of information.

The technology used in the North Sea is, and will for a long time remain, a mix of new and old installations and wells, where the sophistication of the instrumentation and the quality of measurements varies. This combination of new and old systems makes the extraction of useful information harder.

Since most process data have some degree of noise, any good automatic and self learning method that can detect state changes would be beneficial for the mathematical and statistical analyses of the systems.

1.2 The system

As mentioned above, the data we will analyze in this text comes from the Ekofisk oil field. This section describes in more detail the system our datasets are measured from.

1.2.1 Ekofisk

Ekofisk is an oil field, operated by ConocoPhillips, in the southern part of the North Sea. The production at the Ekofisk field began in 1971, making it the oldest oil field in production on the Norwegian continental shelf. Since then, the field has been under constant development.

The operative parts of the *Ekofisk Centre* consists of several separated facilities, see Figure 1.1 for an illustration. Our datasets come from the pipeline routing production from the *Ekofisk B* platform, top left in the figure, to the processing and production facility *Ekofisk M*, which is located to the right of the "cluster" of facilities. For more information about Ekofisk, see, e.g., [12].

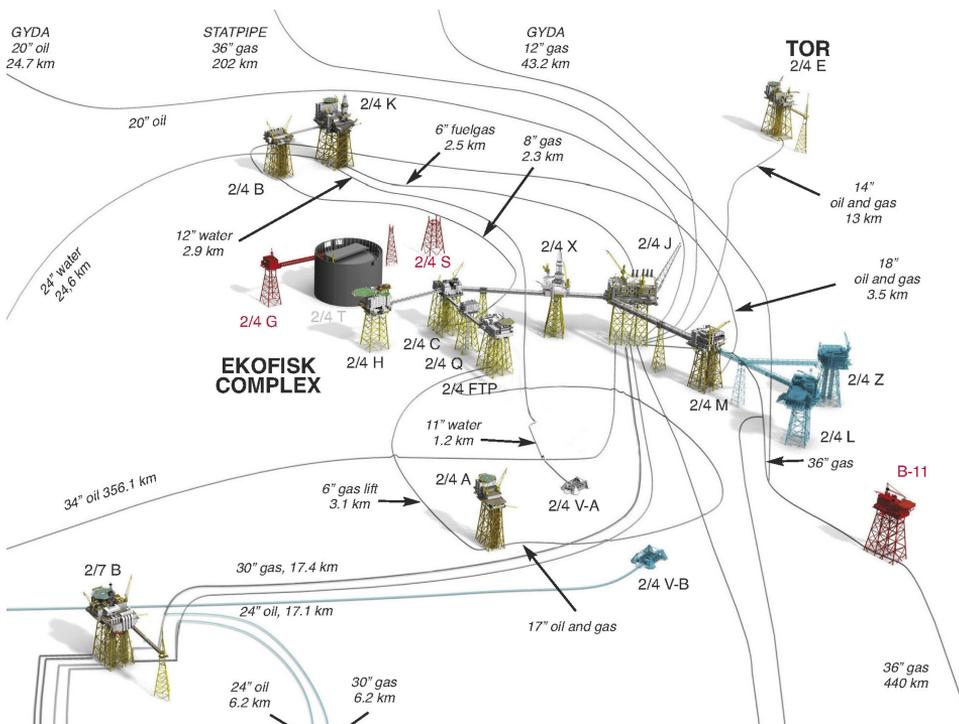


Figure 1.1: Illustration of the Ekofisk complex. The Ekofisk B platform is the top left platform of the illustration, while Ekofisk M is located on the right hand side of the central cluster. The blue platforms are planned installations, while the red are third party structures.

1.2.2 Description of the system

The pipeline

The physical process our data is measured from is a three phase flow inside a pipeline. Oil, water and gas enters the pipeline from 15 different wells of Ekofisk B, and is transported approximately 2.5km inside this pipeline, to the Ekofisk M facility, where the pipeline enters a high pressure (HP) separator.

In the end of the pipeline between Ekofisk B and Ekofisk M, there is located a *slug control valve*. The purpose of this valve is to limit the amount of an unwanted phenomena called slugging. Slugs in the system result in unstable flow. For a brief description of slugging, see section 1.2.3. We will not go into any discussion about the slug control valve beyond its purpose. Figure 1.2 shows a sketch of the system.

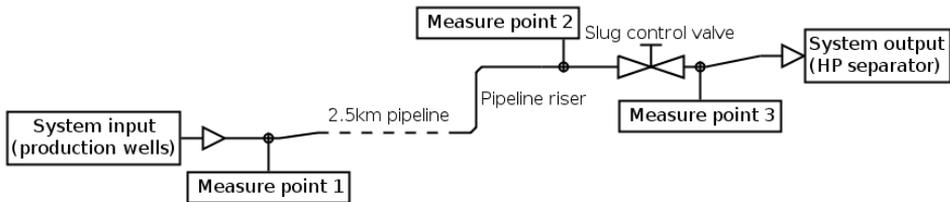


Figure 1.2: Sketch of the system our data comes from. The system output comes from 15 wells at Ekofisk B. The three phase flow is transported to Ekofisk M, through a 2.5km pipeline. At Ekofisk M, the flow is transported up to the platform through the pipeline riser. Before the HP separator, the pipeline goes through a slug control valve. See section 1.3 for a discussion on the measure points.

Phase separation

The purpose of the HP separator is to separate the water, oil and gas that is fed into it from the pipeline. This is an important process. The amount of water in the processed oil is crucial and should be as low as possible.

Water in the oil is unwanted, mainly because it can lead to corrosion and other problems in subsequent processing systems. Because of this, it is important that the separator functions optimally.

For the HP separator to work well, it is beneficial with a steady input flow. An even input flow is unfortunately not the case in real life. As mentioned above, the input from Ekofisk B originates from 15 wells. All of these are usually in production. With 2.5km of unstable three phase flow it is obvious that we are dealing with a system that is very hard to model directly. In addition to the wells from Ekofisk B, wells of the Ekofisk M platform are also being processed in the same separator, making a complete analysis of the pipeline problems even harder.

1.2.3 Behavior of the system

We will now take a look at some of the different behaviors of the system. Irregular flow is likely to reduce both the production quality and quantity, and can in extreme cases cause system shutdowns [6]. Since irregular flow, reflected by more rapid changes in pressure, is more likely to cause the system to operate badly, we will consider such behavior *special events* in the system. Figure 1.3 shows four examples of this occurring.

On the contrary, as already mentioned, the separation and production process works best under stable input flow. We therefore say that on intervals of time where we, loosely speaking, only observe small fluctuations the system is in its *ideal state*. Figure 1.4 shows an example of this.

In reality, it is off course not this simple. There may be several different *good* and *bad* states of a system and also different degrees of faulty behavior. However, with the data we are given, we can only qualitatively distinguish the different states and describe them with the information we actually do have.

Slugging and special events

Since the system is quite complex, there can be several different causes leading to unstable behavior. We will limit this discussion to one. We have

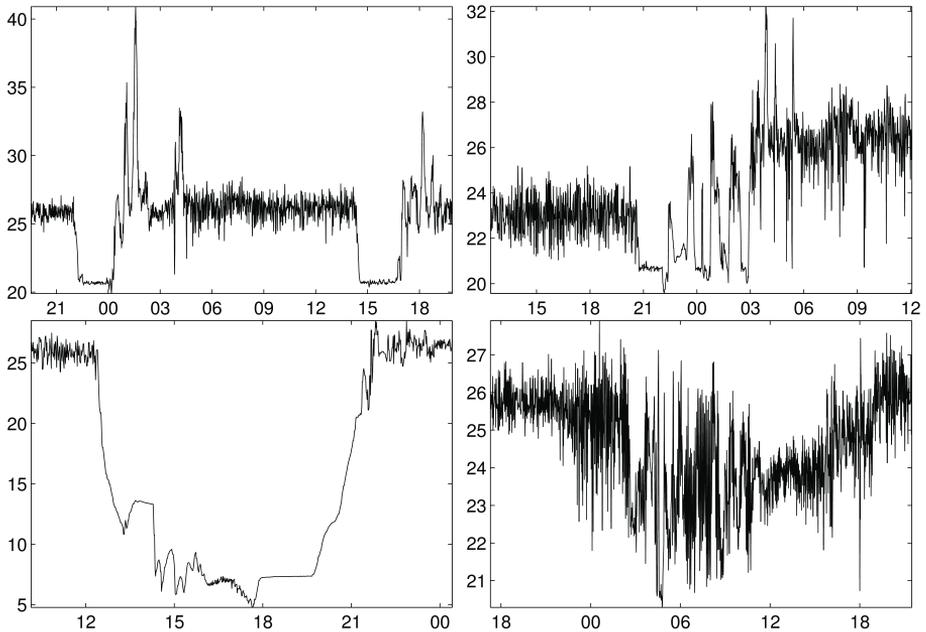


Figure 1.3: Four examples of events in the data. All examples come from the second dataset. The datasets are described in more detail in section 1.3. Top left: 25-26 September, top right: 6-7 November, bottom left: 27-28 November, bottom right: 17-18 March, all in 2011.

already mentioned the phenomena above, namely slugging.

Slugging is one of the special events that regularly occur [6]. The phenomena is characterized by a liquid blockage formed inside the pipeline. This blocks the gas flow and leads to a build-up in pressure behind the *slug*. When the pressure becomes sufficiently high, the gas forces itself past the liquid block, creating instabilities in the system. Figure 1.5 illustrates the phenomena. There are several different types of slugging, e.g. riser slugging and terrain induced slugging. While the reason of their occurrence differs slightly, the symptoms are pretty much the same.

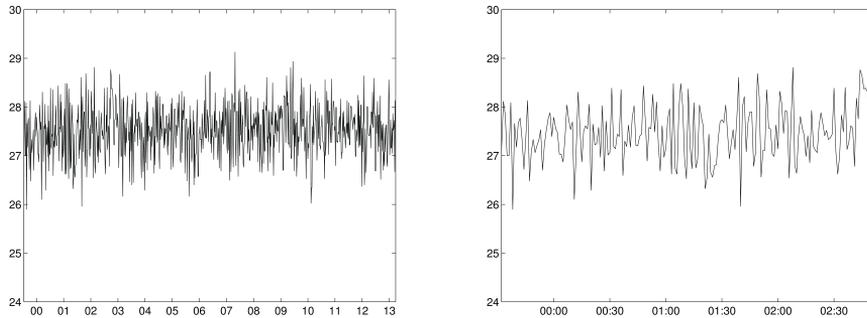


Figure 1.4: *Left: Thirteen hour extract of the second dataset, see 1.3, where the system is operating nicely. Right: The first few hours of the same signal. Signal comes from 17 August 2011.*

According to the company Neftemer - a company making products that aims to reduce slugging - the current total loss caused by slugging worldwide is approximately £6 billion per year. Neftemer also claim that slugging can reduce production by up to 10% [11].

The platform operators do, in real time, notice the effects of slugs. Even so, the occurrence of the phenomena is presently not recorded, since there is no automatic procedure of detection. Being able to automatically detect and also predict slugging in the system before it actually happens, could therefore be valuable.

In the system discussed in this thesis there are not any good countermeasures available for the operator today. A short term remedy is to temporarily close down parts of the production. Although the system have some tolerance for bad behavior, the resulting cost is unknown.

However, there are indications that actively responding to the system could reduce the amount of slugging in the system quite a bit. Techniques like automatic adjustments of the topside choke valve opening have been implemented, yielding a positive impact [4, 5]. This further indicates the possible value of our endeavor.

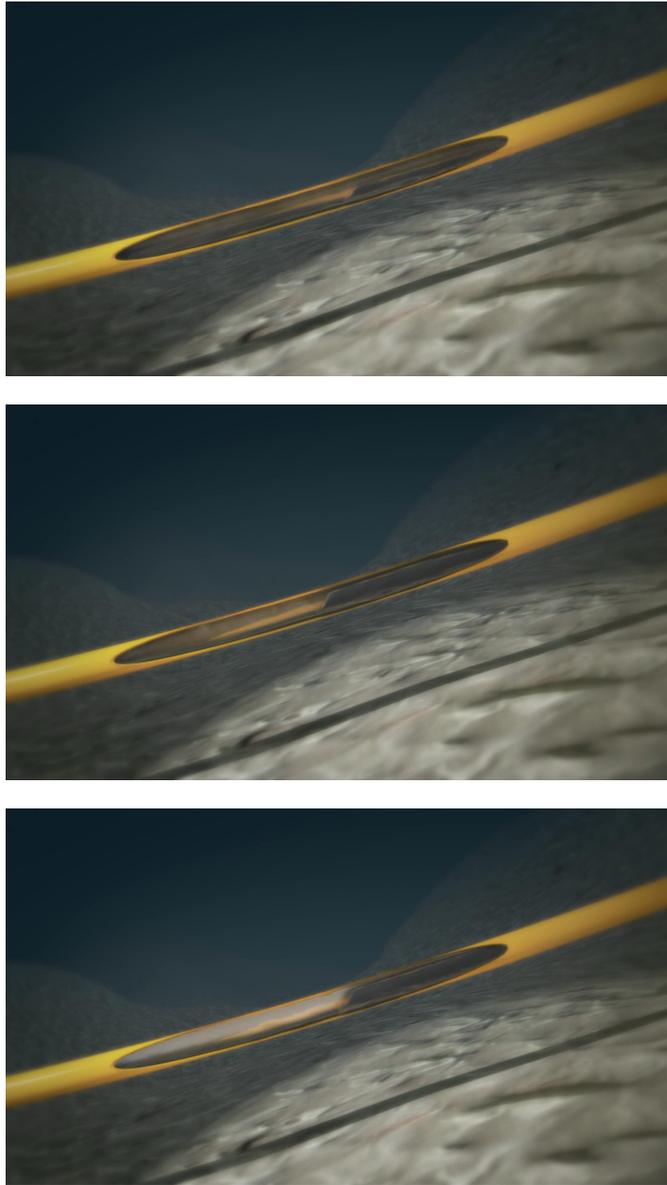


Figure 1.5: *Illustration of the occurrence of slugging in three phase system. The dark colored liquid is water, the yellow liquid is oil and the white "fog" is gas. The first image shows normal flow. In the second, a slug is formed and the pressure behind it start increasing. In the last image the gas pressure behind the slug is high and just about to force itself through. The images are extracted from the short documentary "Laboratory Values" [14], made by Sintef.*

1.3 Datasets

As mentioned earlier in this chapter, there exists a huge amount of information available for analysis. The data which is most readily available in general are pressure data from various points in the process. In addition to the availability of these measurements, experience has shown that the pressure measurements of our system is highly correlated with the performance of the separation process. With these facts in mind, applying our analysis on datasets of pressure measurements makes sense.

We are given three datasets, containing pressure measurements at the points of the pipeline indicated in Figure 1.2. The three datasets are named:

1. EKOB pipeline pressure U S ESD
2. EKOM U S Slug EkoB riser vlv p
3. EKOM D S Slug EkoB riser vlv p

We will, for the most part, refer to these as respectively the first, second and third dataset. The first dataset, *EKOB pipeline pressure U S ESD*, is the pressure at the beginning of the pipeline leaving the Ekofisk B rig. The second and third datasets, are the pressures in the same pipeline, arriving at Ekofisk M, but before the HP separator. The second set, *EKOM U S Slug EkoB riser vlv p*, consists of measurements just before the slug control valve, briefly mentioned above, while the third set, *EKOM D S Slug EkoB riser vlv p* is measured a few meters away from the second, but after the slug control valve. Figure 1.6 shows an interval extracted from the three datasets.

The average pressure at the first measure point is higher than at the two others, while the second measure point has the second largest average pressure. The average value of the three datasets are, respectively, 29.9, 25.7 and 19.7 barg (this is the average of all the available data). The fall in pressure is off course expected. The fall between the first and second dataset is mostly due to friction in the pipeline, while the difference between the second and third dataset is due to the slug control valve. The difference in pressure can be seen in Figure 1.6.

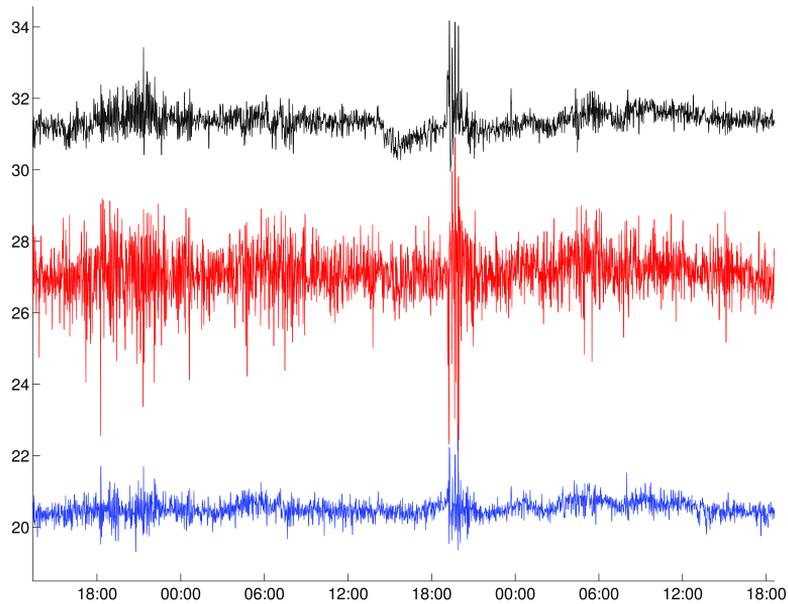


Figure 1.6: *Typical 48 hour extract of the three time series. First dataset is black, the second dataset red, and the third dataset blue. Note the difference in mean pressure between the three signals. The signal was recorded between 15 and 17 February 2012.*

Qualitatively, the second dataset has a higher amount of average variation, or instability, while the third is usually more stable than the others. Also this can be observed in the signal extracts in Figure 1.6.

Remark. Note that the above observations are based on the typical trend of the signal overall, not on the extract of Figure 1.6.

Apart from the information in the three datasets, we do not have information about any particular events. This means that we manually

have to scan the datasets, looking for special events that can be of interest.

With the datasets spanning from 1 January 2010 to 6 March 2012, we have more than 1.1 million data points for each of the three datasets. The typical intervals we will be analyzing are on the timespan of between a few hours to a few days. Further, the events occurring usually last between a few minutes to a few hours. Keeping this in mind, we have an overwhelming amount of data available. As we later note in section 2.6, this opens the possibility for statistical optimization of the parameters in our algorithm (we do not however carry out such an analysis in this thesis).

The sample frequency of the three data types is quite high with respect to the scale we are interested in. Measurements are being done approximately every second. However, only a fraction of this data is stored and thus available for analysis. This is due to limitations in the current control system. The stored data has an average, but not uniform, sampling rate of approximately 40 seconds. In this text, we will convert the data to minutely intervals. We come back to this preparation of the datasets in section 2.7.

1.3.1 Example intervals

Different states

In Figure 1.7 we can clearly see three different behaviors, or states, of the system. On roughly the first third of the interval, the system has a relatively high variation, but does not contain big jumps or extreme values.

Then, the central part of the interval, the system seems to be behaving in a nice and steady state. This behavior is what we associate with the term ideal state.

At between 20:00 to 22:30 towards the end of the interval, we have a slowly oscillating instability. We can observe big variation in pressure changing slowly during this time. After, the system seems to jump back to the state of the first part of the signal. We will get back to this extract of the datasets later in the text.

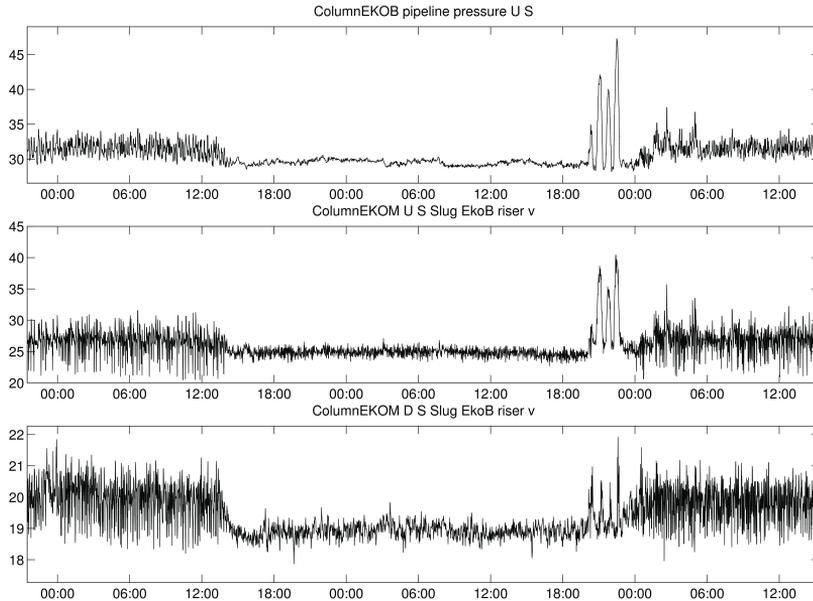


Figure 1.7: *Time interval where the measurements vary from quite noisy, to quite nice, before a big event occurs at around 20:00 to 22:30 on the last day of the interval. Note that the scales along the y-axes are not equal. Extract from 10 to 12 October 2010*

The slug control valve

We now briefly discuss the effect of the slug control valve. As mentioned, the valve is situated between our second and third dataset and its purpose is to reduce slugging and irregular behavior.

The effect of the valve can clearly be seen in both Figure 1.6 and 1.7. The valve seems to be attenuating the oscillations of the second dataset. We can still spot the unstable intervals in the third dataset, but it is not as clear. Figure 1.8 shows an example where the unstable interval is merely visible after the slug control valve.

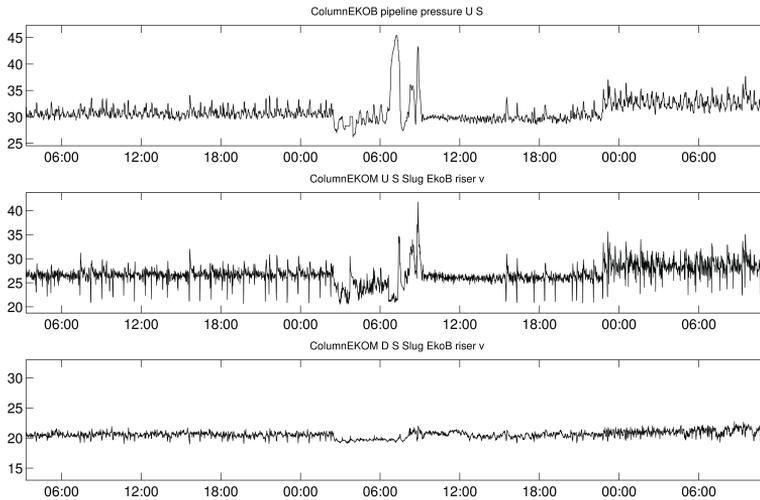


Figure 1.8: *The irregularity on the centre of the interval is clearly reduced through the valve. Irregularities are not always dampened as efficiently as in this case. Extract from 7 to 9 January 2010.*

1.4 Thesis structure

One of the goals of this thesis is to learn how to identify and analyze different states of the system. Doing so, would give increased knowledge of the system itself and could also make it possible to partly counteract the unwanted changes in the system. Thereafter, the intention is to investigate the possibility of predicting special events based on the newest data available.

As mentioned, to pinpoint the cause of events is not our aim. We do not, for instance, distinguish the effects caused by someone "pushing a button" from the effects of special events.

In real life, these "false detections" would easily be associated with changes done by the operator and thus safely overlooked. Therefore we

only care about detecting that something is going on, and characterize the behavior.

1.4.1 A brief overview

Our approach is based on the on the following understanding: In order to detect special events within the signal, one has to analyze its components, on various scales, rather than the signal as a whole. There are two types of relevant time scales in our case: the scales corresponding to local and global behavior.

The *local* scales will determine the characteristics of the time-frequency decomposition, while the *global* scale determines the size of the window, i.e., the size of the time-interval in which this decomposition is done. For our particular problem, the global scale is approximately two hours, while the local scales range from a couple minutes to roughly three quarters of an hour.

With this approach we develop a Sliding Window Wavelet Transform (SWWT) method that enables us to monitor the local variability of multiscale coefficients, taken within a typical time interval (the global scale from above). We evaluate the scalewise variance of these coefficients, while the window moves along the time axis. The values of these variances are used in order to indicate special events.

For convenience, we will distinguish between two cases: causal, when the window location is attributed to the rightmost point of the window, say t_0 , and non-causal, with the window centered around t_0 . In the causal case, the transform does not, per definition, contain any "future information", i.e., information occurring after t_0 . This is appropriate for dealing with prediction of events. The non-causal alternative will be used for characterizing past events from recored data.

1.4.2 Description of the methods

Calculations within one window

In order to develop a multiscale decomposition within a window, we use DWT. It is well adjusted to the discrete structure of the data, besides, the orthogonality property simplifies the calculation of variances.

Additional advantage comes from the fact that one can use a variety of numerical packages developed for DWT. We use the MATLAB package *WaveLab 850* for basic DWT operations. A collection of MATLAB functions and scripts (built upon *WaveLab 850*) that implement the techniques used in this thesis is also implemented.

We use the periodized wavelet transform with finite resolution and the cascading databank algorithm to calculate DWT of a sequence. Once DWT of a window is calculated, we use the orthogonality of the transform to calculate the scalewise variance.

The sliding window

The SWWT truncates the dataset corresponding to the sliding window and applies the the calculations described above in every window. From this method, we obtain a trace of the localized variability on different scales along the dataset, which we will base our analysis on.

Chapter 2

Mathematical background

In this chapter, we describe the mathematical background and also develop necessary techniques. The chapter splits naturally into three parts. In the first we describe the continuous and discrete wavelet transform for functions $f \in L^2(\mathbb{R})$. In this part, we present the general theory. This part consist of sections 2.1 and 2.2, presenting the continuous and discrete wavelet transform.

The second part consists of section 2.3 and 2.4. Section 2.3 deals with periodizing DWT in order to work with functions, $f \in L^2[0, 1]$, with finite domain. We associate a finite sequence of coefficients with a projection of a function from $L^2[0, 1]$ onto a space of finite resolution. In section 2.4 we then define the wavelet variance of such sequences.

In the third part, section 2.5 to the end of the chapter, we define the Sliding Window Wavelet Transform (SWWT). Based on this, we can to do a time dependent analysis of variance on different scales along the datasets. We finally define a thresholding algorithm to enable us to characterize special behavior of the system.

We begin by describing wavelet and the discrete wavelet transform (DWT) which is at the very heart of our algorithm, before introducing discrete orthonormal wavelets in section 2.2.1. After periodizing the DWT

and filter bank algorithm in section 2.3.1 and 2.3.2, we show how to associate a finite sequence with the DWT of a continuous function. This is done in section 2.3.3. Using these properties, we show that the variance of a sequence can be precisely decomposed into information on the different scales, using the wavelet variance, see section 2.4.

Section 2.5 introduce the time dependent SWWT, which is essentially obtained by calculating DWT of signal extracts inside a moving window. Section 2.6 finally introduce the thresholding algorithm, which we use in order to characterize the signal.

2.1 The continuous wavelet transform

Wavelets is a relatively new tool in mathematical signal processing. During the last couple decades, wavelet theory has been applied in many different fields and have been proven to be very powerful. Some of its applications worth mentioning are signal compression [16], singularity detection [8] and denoising [2]. It has also been found to be useful in statistical characterization of time series, e.g [13].

In contrast to Fourier analysis, where signals are decomposed into a trigonometric basis, wavelet analysis decompose signals into a basis where the basis elements are concentrated in time. While there only exists one basis for Fourier analysis, there are multiple candidates in wavelet analysis.

Definition 2.1.1. A *wavelet* is any function $\psi(t) \in L^2(\mathbb{R})$, centered around $t = 0$, satisfying

$$C_\psi = \int_0^\infty \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty. \quad (2.1)$$

This is usually called the admissibility condition for the wavelet, see, e.g., [7, 17]. Note that (2.1) implies

$$\int_{-\infty}^{\infty} \psi(t) dt = 0.$$

We also assume that ψ is normalized, i.e.,

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1.$$

Once a wavelet ψ is chosen, a wavelet basis is created by dilating and translating ψ , yielding the following system of *time-frequency atoms*:

$$\{\psi_{s,u}\}_{u \in \mathbb{R}, s \in \mathbb{R}^+}, \psi_{s,u}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right).$$

The parameter s dilates the wavelet, while u translates it along the time domain. The simplest example of a wavelet is the *Haar wavelet*, which is defined by:

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

Which gives us

$$\psi_{s,u}(t) = \begin{cases} -1/\sqrt{s}, & \text{if } -s/2 + u \leq t < u \\ 1/\sqrt{s} & \text{if } u \leq t < s/2 + u \\ 0 & \text{otherwise.} \end{cases}$$

Three elements of the resulting Haar wavelet basis $\{\psi_{s,u}\}$ are shown in Figure 2.1. Figure 2.2 shows two other examples of wavelet functions. These are of the form

$$\psi(t) = (-1)^n \theta^{(n)}(t), \tag{2.2}$$

where $\theta^{(n)}$ is the n^{th} derivative of the Gaussian. Note that these wavelets does not in fact have compact support. They do however fulfill the admissibility condition (2.1), and are therefore proper wavelets. The wavelets defined by (2.2) are for instance very useful, for classifying signal irregularities, see, i.e., [9]. We only consider compactly supported wavelets from this point on.

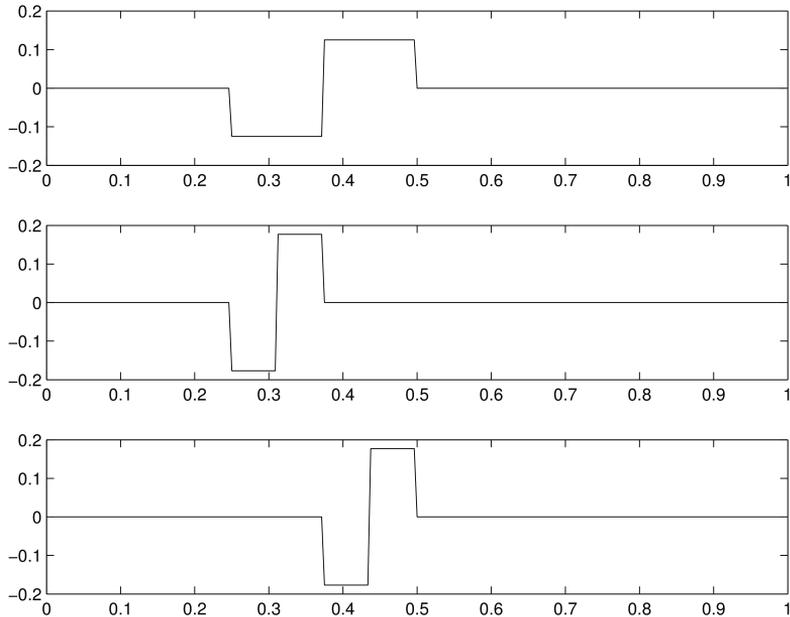


Figure 2.1: *Illustration of the translation and dilation of the Haar wavelet. The second and third plot shows a dilated, or compressed, version of the first. The third subplot is a translation of the second.*

Given a wavelet ψ and signal f , the continuous wavelet transform (CWT) is given by:

$$\mathcal{W}_\psi f(s, u) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) dt = \int_{-\infty}^{\infty} f(t) \psi_{s,u}(t) dt.$$

The continuous wavelet transform has a well defined inverse and preserves the L^2 -norm and is therefore stable, see e.g. [7], section 4.3. Figure 2.3 shows an example of CWT calculated on an extract from the first dataset,

EKOB pipeline pressure U S ESD.

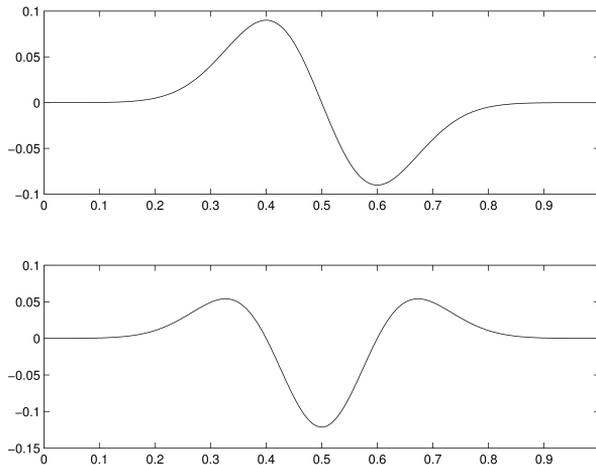
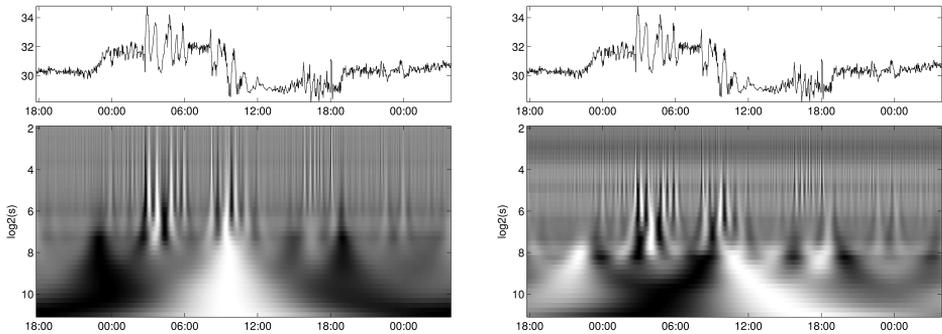


Figure 2.2: *Wavelets on the form of (2.2). Top plot: The first derivative of the Gaussian, Bottom plot: Second derivative of the Gaussian. The latter is often called the mexican hat wavelet, because of its characteristic shape.*

Applying the wavelet transform in any practical application forces us to discretize the parameters s and t . When calculating the CWT of Figure 2.3, we restricted the scales to $s = 2^{j/v}$ with $v = 4$ and the time to one minute intervals, corresponding to the sampling rate of our data. In other words, for every j , 2^{11} values of the transform are calculated, and for every *octave*, that is $s \in [2^k, 2^{k+1})$, we calculate $v = 4$ scales. In the next section, we introduce another discretization that yields the discrete wavelet transform.

2.2 The discrete wavelet transform

The discretization used in the calculation of Figure 2.3 is common when numerically calculating CWT of a function. The resulting transform is



(a) CWT using the first derivative of the Gaussian (b) CWT using the second derivative of the Gaussian

Figure 2.3: Examples of CWT calculated on an interval of EKOB pipeline pressure U S ESD. Black indicates positive coefficients, white negative and grey corresponds to zero, or close to zero values. The signal was measured between 17 to 19 March 2011.

highly redundant. In this section, we investigate a particular discretization which enables an orthonormal representation of functions in $L^2(\mathbb{R})$.

If we choose to discretize using dyadic scales $s = 2^j$ in scale and $u_j = 2^j n$, $n \in \mathbb{Z}$ in time, we get what is usually called the *discrete wavelet transform* (DWT) [1, 7]:

$$\mathcal{W}_\psi f(j, n) = \int_{t=-\infty}^{\infty} f(t) \frac{1}{\sqrt{2^j}} \psi \left(\frac{t - 2^j n}{2^j} \right) dt. \quad (2.3)$$

Remark. Notice the difference between this discretization and the one we used to approximate CWT. Here, the distance in time between the location of $\mathcal{W}f(j, n)$ and $\mathcal{W}f(j, n+1)$. This distance is greater on a scale $j+1$, then on the scale j .

2.2.1 Orthonormal discrete wavelet

We now show how to find wavelets ψ that create an orthonormal basis of $L^2(\mathbb{R})$. We later see that this orthonormality is a very useful property for our application. We follow the procedure of Mallat, [7], in this and the next sections.

Definition 2.2.1. A *multiresolution analysis* (MRA) is a sequence of subsets $\{\mathbf{V}_j\}_{j \in \mathbb{Z}} \subset L^2(\mathbb{R})$, satisfying

$$\begin{aligned}
 \forall (j, k) \in \mathbb{Z}^2, f(t) \in \mathbf{V}_j &\iff f(t - 2^j k) \in \mathbf{V}_j, \\
 \forall j \in \mathbb{Z}, \mathbf{V}_{j+1} &\subset \mathbf{V}_j, \\
 \forall j \in \mathbb{Z}, f(t) \in \mathbf{V}_j &\iff f\left(\frac{t}{2}\right) \in \mathbf{V}_{j+1}, \\
 \lim_{j \rightarrow \infty} \mathbf{V}_j &= \bigcap_{j=-\infty}^{\infty} \mathbf{V}_j = \{0\}, \\
 \lim_{j \rightarrow -\infty} \mathbf{V}_j &= \text{Closure}\left(\bigcup_{i=-\infty}^{\infty} \mathbf{V}_i\right) = L^2(\mathbb{R}).
 \end{aligned} \tag{2.4}$$

In addition there exists a function ϕ , called a *scaling function* s.t. $\{\phi(t - n)\}_{n \in \mathbb{Z}}$ is an orthonormal basis of \mathbf{V}_0 .

Remark. The above definition can be generalized by requiring a Riesz basis instead of an orthonormal basis. It is possible to create an orthonormal basis from the Riesz basis (see for instance section 7.1 of [7]), so we assume an orthonormal basis for convenience.

Given a MRA $\{\mathbf{V}_j\}_{j \in \mathbb{Z}}$ with scaling function ϕ we get that $\{\phi_{j,n}\}_{n \in \mathbb{Z}}$, where

$$\phi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{t - 2^j n}{2^j}\right),$$

is an orthonormal basis of \mathbf{V}_j for all $j \in \mathbb{Z}$. Based on this, we get the orthonormal projection of f onto any \mathbf{V}_j , given by: $P_{\mathbf{V}_j} f =$

$\sum_{n=-\infty}^{\infty} \langle f, \phi_{j,n} \rangle \phi_{j,n}$. We introduce the notation: $a_j[n] = \langle f, \phi_{j,n} \rangle$. A change of variables yields:

$$a_j[n] = f * \bar{\phi}_j(2^j n), \text{ where } \bar{\phi}_j(n) = \frac{1}{\sqrt{2^j}} \phi(-2^{-j} n).$$

The projection onto \mathbf{V}_j becomes:

$$P_{\mathbf{V}_j} f = \sum_{n=-\infty}^{\infty} a_j[n] \phi_{j,n},$$

where $a_j[n]$ is a convolution. Since the convolution of two signals can be computed quickly, using the fast Fourier transform (FFT), this is convenient.

Quadrature mirror filter

We want to find a condition on ϕ that guarantees that the conditions of (2.4) are satisfied. Observe that the third property of (2.4) implies that $2^{-1/2} \phi(t/2) \in \mathbf{V}_1$. The second property of the MRA also tells us that $\mathbf{V}_{j+1} \subset \mathbf{V}_j$, in particular $\mathbf{V}_1 \subset \mathbf{V}_0$. Hence, we can express $2^{-1/2} \phi(t/2) \in \mathbf{V}_1$, using the basis elements $\{\phi(t-n)\}_{n \in \mathbb{Z}}$ of \mathbf{V}_0 :

$$\frac{1}{\sqrt{2}} \phi(t/2) = \sum_{n=-\infty}^{\infty} \left\langle \frac{1}{\sqrt{2}} \phi(t/2), \phi(t-n) \right\rangle \phi(t-n)$$

This is called the *scaling relation* and is really a cornerstone of the theory that follows. We define $h[n] = \langle \frac{1}{\sqrt{2}} \phi(t/2), \phi(t-n) \rangle$ and observe that

$$\hat{\phi}(2\omega) = \frac{1}{\sqrt{2}} \hat{h}(\omega) \hat{\phi}(\omega). \quad (2.5)$$

Assuming that the $\hat{\phi}$ is continuous in the origin we obtain:

$$\hat{\phi}(\omega) = \prod_{p=1}^{\infty} \frac{\hat{h}(2^{-p}\omega)}{\sqrt{2}} \hat{\phi}(0), \quad (2.6)$$

by recursively substituting (2.5) into itself. A particular choice of h guarantees that (2.6) is the Fourier transform of a scaling function.

Definition 2.2.2. A **quadrature mirror filter** is a 2π -periodic function h satisfying

$$\forall \omega \in \mathbb{R}, |\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2.$$

From the above definition we see that quadrature mirror filters are 2π periodic. The following theorem from section 7.1 in [7] shows the importance of such filters.

Theorem 2.2.1. *Let $\phi \in L^2(\mathbb{R})$ be a scaling function. Then the Fourier series of $h[n] = \langle 2^{-1/2}\phi(t/2), \phi(t-n) \rangle$ satisfies*

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2, \quad \forall \omega \in \mathbb{R}$$

and

$$\hat{h}(0) = \sqrt{2}.$$

Conversely, if $\hat{h}(\omega)$ satisfies (2.7) and (2.7), is periodic and continuously differentiable in a neighborhood of $\omega = 0$, then:

$$\hat{\phi}(\omega) = \prod_{p=1}^{\infty} \frac{\hat{h}(2^{-p})}{\sqrt{2}}$$

is the Fourier transform of a scaling function $\phi \in L^2(\mathbb{R})$. If in addition $|\hat{h}(\omega)|$ is strictly positive for $\omega \in [-\pi/2, \pi/2]$, then the union of \mathbf{V}_j is dense in $L^2(\mathbb{R})$.

It follows from the scaling relation that any scaling function is characterized by a quadrature mirror filter h . The above also shows that $\hat{\phi}(0) = 1$, indicating that

$$\hat{\phi}(0) = \int_{-\infty}^{\infty} \phi(t) dt = 1. \quad (2.7)$$

As earlier mentioned we only consider finitely supported wavelets in this thesis. The following theorem, regarding the support of ϕ and ψ is used later in the text:

Theorem 2.2.2. *The scaling function ϕ has a compact support if and only if h , the quadrature mirror filter, has a compact support. In this case, their supports coincide. If the support of g and ϕ is $[N_1, N_2]$, the support of ψ is $[(N_1 - N_2 + 1)/2, (N_2 - N_1 + 1)/2]$.*

The proof can be found in section 7.2 of [7].

Detail spaces

From (2.4) we know that \mathbf{V}_j is contained in \mathbf{V}_{j-1} . We now define *detail spaces* by $\mathbf{W}_j = \mathbf{V}_{j-1} \ominus \mathbf{V}_j$. This implies:

$$\mathbf{V}_{j-1} = \mathbf{V}_L \oplus \left(\bigoplus_{k=j}^L \mathbf{W}_k \right), \quad (2.8)$$

for all $J \leq L$.

Remark. In the decomposition $\mathbf{V}_{j-1} = \mathbf{V}_j \oplus \mathbf{W}_j$, the space \mathbf{V}_j contains the more "coarse" information, while \mathbf{W}_j corresponds to the "finer details". Therefore, equation (2.8) is a decomposition of the space \mathbf{V}_{j-1} into a coarse scale \mathbf{V}_L and multiple detail spaces, \mathbf{W}_k with $k \in [j, \dots, L]$, each containing details on scale 2^k , justifying their name.

Orthonormal wavelet basis

Now we need bases $\{\psi_{j,n}\}_{n \in \mathbb{Z}}$ which span \mathbf{W}_j for all j . Given a scaling function ϕ and its corresponding quadrature mirror filter h we define ψ as the function with Fourier transform

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{2}} \hat{g}\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right), \quad (2.9)$$

$$\text{where } \hat{g}(\omega) = e^{-i\omega} \hat{h}^*(\omega + \pi).$$

The inverse Fourier transform of \hat{g} yields $g[n] = (-1)^{1-n} h[1-n]$. It can then be shown (see, e.g., [7]) that

$$\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - 2^j n}{2^j}\right), \quad n \in \mathbb{Z}$$

form an orthonormal basis $\{\psi_{j,n}\}_{n \in \mathbb{Z}}$ of \mathbf{W}_j , for every scale j .

The following result from section 7.1 in [7] gives a necessary and sufficient condition for \hat{g} to correspond to orthonormal wavelets ψ .

Theorem 2.2.3. *$\{\psi_{j,n}\}_{n \in \mathbb{Z}}$ created from (2.9) is an orthonormal basis of \mathbf{W}_j for all scales j if and only if h and g satisfies:*

$$\begin{aligned} |\hat{g}(\omega)|^2 + |\hat{g}(\omega + \pi)|^2 &= 2, \\ \hat{g}(\omega)\hat{h}^*(\omega) + \hat{g}(\omega + \pi)\hat{h}^*(\omega + \pi) &= 0. \end{aligned} \tag{2.10}$$

We now have the necessary tools to decompose any function $f \in L^2(\mathbb{R})$ in orthonormal components of ψ .

Orthonormal representation

Given ψ , satisfying (2.9) with h and g satisfying (2.10), we now have orthonormal bases for \mathbf{V}_j and \mathbf{W}_j , for all j . If we denote the coefficients of the projections onto \mathbf{V}_j and \mathbf{W}_j by respectively

$$a_j[n] = \langle f, \phi_{j,n} \rangle, \text{ and } d_j[n] = \langle f, \psi_{j,n} \rangle,$$

the projections onto \mathbf{V}_j and \mathbf{W}_j are:

$$\begin{aligned} P_{\mathbf{V}_j} f &= \sum_{n=-\infty}^{\infty} \langle f, \phi_{j,n} \rangle \phi_{j,n} = \sum_{n=-\infty}^{\infty} a_j[n] \phi_{j,n}, \\ P_{\mathbf{W}_j} f &= \sum_{n=-\infty}^{\infty} \langle f, \psi_{j,n} \rangle \psi_{j,n} = \sum_{n=-\infty}^{\infty} d_j[n] \psi_{j,n}. \end{aligned}$$

We see that the coefficients $a_j[n]$ and $d_j[n]$ characterize the orthonormal projection of f onto respectively \mathbf{V}_j and \mathbf{W}_j . Observe also that $\|P_{\mathbf{V}_j} f\|_{L^2(\mathbb{R})}^2 = \|a_j\|_{\ell^2}^2$, and similarly $\|P_{\mathbf{W}_j} f\|_{L^2(\mathbb{R})}^2 = \|d_j\|_{\ell^2}^2$.

Remark. We simply denote the sequences of coefficients $a_j[n]$ and $d_j[n]$ by a_j and d_j , respectively, when discussing the coefficients of a projection to the spaces \mathbf{V}_j or \mathbf{W}_j .

Since the projections are orthonormal, the last condition of (2.4), combined with (2.8) yields the following decomposition of energy:

$$\begin{aligned} \|f\|_{L^2(\mathbb{R})}^2 &= \sum_{j=-\infty}^L \|P_{W_j} f\|_{L^2(\mathbb{R})}^2 + \|P_{V_L} f\|_{L^2(\mathbb{R})}^2 \\ &= \sum_{j=-\infty}^L \|d_j\|_{\ell^2}^2 + \|a_L\|_{\ell^2}^2, \end{aligned} \tag{2.11}$$

for any $f \in L^2(\mathbb{R})$. This is the decomposition of the energy of functions f from $L^2(\mathbb{R})$. Since we in applications we are working with finite signals, we need to consider the corresponding decomposition of $f \in L^2[0, 1]$. This is done in section 2.3.

Choice of orthonormal discrete wavelets

Different quadrature mirror filters lead to a different multiresolution analysis and thus different wavelet functions. In this section, we briefly discuss some of the possible properties these can have.

In contrast to the case of the continuous wavelet transform, the creation of orthonormal discrete wavelet basis demands a very specific construction. We use known filters, h and g , corresponding to orthonormal wavelet functions in this thesis. For details on the construction of quadrature mirror filters corresponding to orthonormal wavelet functions, see for instance section 7.2 of [7].

A good choice of wavelet function, ψ , depends on the application in which it is to be applied. The wavelet and scaling functions can have several useful properties and characteristics. An example of this is wavelet functions that yields sparse a representation of a signals of a particular type, i.e., efficiently representing a function with relatively few coefficients. This can for instance be exploited in signal compression and is one of the main principles behind the still image compression standard *jpeg 2000* [10]. A sparse transformation indicates that the chosen wavelet is able to extract the characteristics of the signal.

In section 2.3.4 we see that the support of ϕ , therefore the support of h (by Theorem 2.2.2), affects the error incurred by assumptions in our algorithm. Briefly summarized, the size of support affects the number of biased wavelet coefficients. Therefore, we should not select a scaling filter with large support.

Apart from the size of the support of the wavelet, the actual choice is not crucial for the applications of this thesis. It is off course an advantage that the shape of the wavelet can capture the characteristics of the signal on the different scales. Keeping this in mind, we use the symmlet family of wavelets. In particular we select the symmlet 4 wavelet and scaling function, shown in Figure 2.4 (the number 4 indicates the order of ψ and ϕ , see [7] for details).

Remark. Daubechies has shown, chapter 8 of [1], that the only symmetric/antisymmetric discrete orthonormal wavelet is the Haar wavelet, shown in Figure 2.1. The symmlet family of wavelets are constructed to be as close to symmetric as possible. Because of this they are also known as the least asymmetric wavelets.

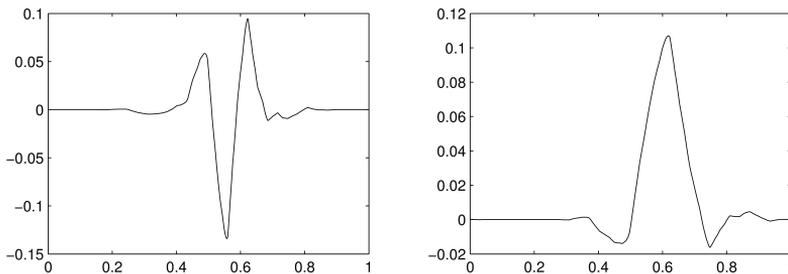


Figure 2.4: *Left plot: Symmlet 4 wavelet ψ , right plot: Symmlet 4 scaling function . Note that the functions are indeed close to being symmetric.*

2.2.2 Filter bank algorithm

We now have an orthonormal basis for both \mathbf{V}_j and \mathbf{W}_j on every scale j . Remember that the spaces \mathbf{W}_j are constructed, so that (2.8) always holds. In this section we look at how $P_{\mathbf{V}_{j+1}}f$ and $P_{\mathbf{W}_{j+1}}f$ can be calculated from $P_{\mathbf{V}_j}f$, in other words how to decompose a_j into a_{j+1} and d_{j+1} .

Decomposition

The following theorem, by Mallat, shows how the coefficients of a_j can be used to calculate the coefficients of a_{j+1} and d_{j+1} , so that:

$$P_{\mathbf{V}_j}f = \sum_{n=-\infty}^{\infty} a_j[n]\phi_{j,n} = \sum_{q=-\infty}^{\infty} a_{j+1}[q]\phi_{j+1,q} + \sum_{q=-\infty}^{\infty} d_{j+1}[q]\psi_{j+1,q}.$$

Theorem 2.2.4. *The sequences a_{j+1} and d_{j+1} are found by:*

$$\begin{aligned} a_{j+1}[q] &= \sum_{n=-\infty}^{\infty} h[n-2q]a_j[n] = a_j * \bar{h}[2q], \\ d_{j+1}[q] &= \sum_{n=-\infty}^{\infty} g[n-2q]a_j[n] = a_j * \bar{g}[2q]. \end{aligned} \tag{2.12}$$

Where $\bar{x}[n] = x[-n]$. Also, given a_{j+1} and d_{j+1} , we can reconstruct a_j by:

$$a_j[n] = \sum_{q=-\infty}^{\infty} h[n-2q]a_{j+1}[q] + \sum_{q=-\infty}^{\infty} g[n-2q]d_{j+1}[q]. \tag{2.13}$$

The proof of the theorem can be found in section 7.3 of [7]. The theorem states that we can calculate a_{j+1} and d_{j+1} , simply by taking every second value of the convolution of a_j with respectively \bar{h} and \bar{g} . What really happens in equation (2.12) is a downsampling of a_j , into two smaller signals. One, d_{j+1} , containing the high frequencies of a_j , the other, a_{j+1} , containing the low frequencies.

Equation (2.12) and (2.13), of Theorem 2.2.4, together make up what is called the *fast orthogonal wavelet transform*, or, the *cascade databank algorithm* [15].

When we have done the decomposition (2.12) once, the decomposition can be applied again, this time on a_{j+1} . Repeating this process, we end up with the following decomposition:

$$\begin{aligned} P_{V_J} f &= \sum_{j=J+1}^L \sum_{n=-\infty}^{\infty} d_j[n] \psi_{j,n} + \sum_{n=-\infty}^{\infty} a_L[n] \psi_{L,n} \\ &= \sum_{j=J+1}^L P_{W_j} f + P_{V_L} f \end{aligned} \tag{2.14}$$

for any initial projection of f onto V_J and choice of integer, L .

Remark. Notice that the above equation has a finite resolution. In other words: a finite number of scales j , $J \leq j \leq L$. For a representation of $f \in L^2(\mathbb{R})$, let $J \rightarrow -\infty$ and $L \rightarrow \infty$.

Figure 2.5 shows the projections, (2.14), of a test signal f from V_{-8} , onto V_j , $j = -8, \dots, -3$, and W_j , $j = -7, \dots, -3$. More precisely, the figure depicts the periodized projections of a signal with domain $[0, 1]$. We discuss the details of the procedure in section 2.3. The decomposition of functions from $L^2(\mathbb{R})$ would look principally the same, along all of \mathbb{R} . The wavelet coefficients of Figure 2.5, on scales $j = -7, \dots, -3$ and scaling coefficients, on the most coarse scale L , are shown in Figure 2.6.

2.3 Dealing with finite signals

Above, we have been considering signals $f \in L^2(\mathbb{R})$. In each step of our calculations, we wish to deal only with the part of the signal located within the given window. Thus we restrict to functions of finite domain.

Doing so cause problems related to the influence of the boundary region. There are several ways to deal with this. The approach of this thesis is periodization of the signal. In section 2.3.1 we describe the periodization procedure and show that we can obtain an orthonormal basis of $L^2[0, 1]$. Continuing, we see how the cascade databank algorithm should be appropriately modified in section 2.3.2.

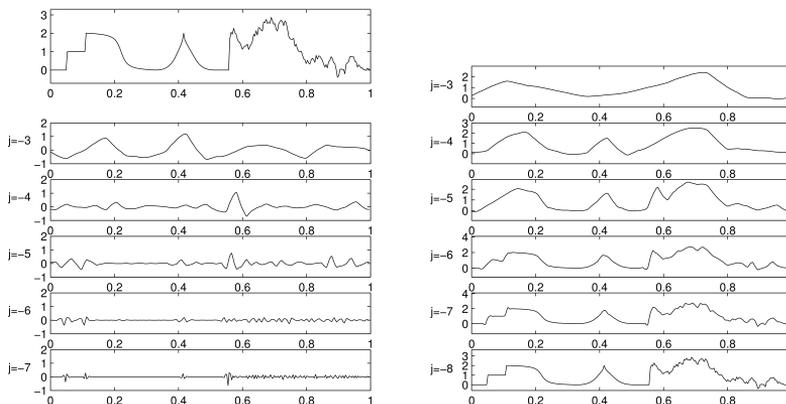


Figure 2.5: *Projection of a signal from V_{-8} onto subspaces V_j and W_j as in (2.14). Top left: The original signal, top right: coarse approximation, left column: projections onto detail spaces W_j , right column: projections onto coarse spaces V_j . A projection in V_j is the sum of the projections in V_{j+1} and W_{j+1} . Since the signal comes from V_{-8} , the bottom right plot contains the original signal. The test signal is available in Wavelab .850 as `msignal` and was originally used in [9]. Symmlet 4 is used for the calculations.*

In numerical calculations, we usually have is a finite sequence of coefficients that contain samples of a signal. Since the wavelet transform operate on functions, we need to associate this finite sequence with the projection of a function to a space of appropriate scale. We do this in section 2.3.3. In section 2.3.4 we qualitatively discuss the error in numerical calculations, due to the truncating (when initializing) and periodization.

2.3.1 Periodization of the transform

In applications we deal with functions defined on a finite interval, say $[0, 1]$ for definiteness. Yet it is still convenient to prolongate it to the whole

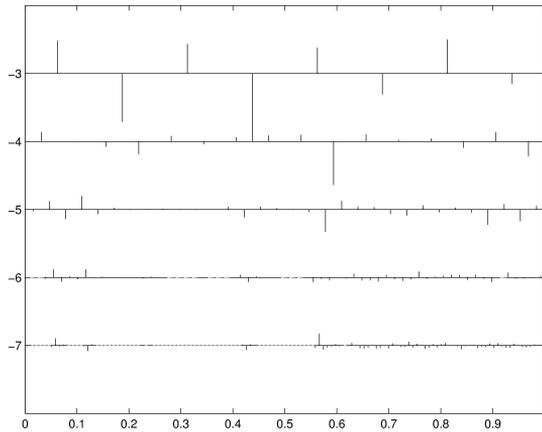
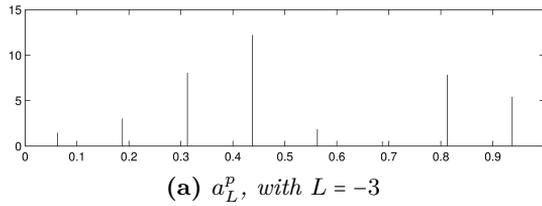


Figure 2.6: The wavelet and scaling coefficients of *msignal*, shown in Figure 2.5. The coefficients correspond to d_j^p , $j = -7, \dots, -3$ and a_L^p , $L = -3$ of (2.14). a_j^p and d_j^p are periodized scaling and wavelet coefficients and are formally defined in section 2.3.2. The Symmlet 4 wavelet and scaling filter was used for the calculation. The wavelet coefficients are scaled so that $\max_{j,k}(a_j[k]) = 1$.

real axis, by taking the *periodization*. We need to transform the wavelet basis elements $\{\psi_{j,n}\}_{(j,n)\in\mathbb{Z}^2}$ from $L^2(\mathbb{R})$ into $L^2[0,1]$. Given $\psi \in L^2(\mathbb{R})$, we periodize $\psi_{j,n}$ in the natural way, see, e.g., [7]:

$$\psi_{j,n}^p(t) \equiv \sum_{k=-\infty}^{\infty} \psi_{j,n}(t+k) = \frac{1}{\sqrt{2^j}} \sum_{k=-\infty}^{\infty} \psi\left(\frac{t-2^j n+k}{2^j}\right).$$

For every $j \leq L \leq 0$ we then have 2^{-j} basis elements in \mathbf{W}_j . Since $\psi_{j,n}^p$ is only defined for $j \leq L \leq 0$, we also need the periodized scaling functions, $\phi_{L,n}$, on the most coarse scale L . We define $\phi_{L,n}^p$ to be the periodization of $\phi_{L,n}$, defined like $\psi_{j,n}^p$ above.

The *periodized discrete wavelet transform* is:

$$\mathcal{W}_\psi f(j, n) = \int_0^1 f(t) \psi_{j,n}^p(t) dt, \quad (2.15)$$

where $f \in L^2[0,1]$ and the limits for j and n are given above. The periodization of $\psi_{j,n}$ is equivalent to periodically extending f over $L^2(\mathbb{R})$. To see this, we define f_p to be the 1 periodic extension of f , that is, $f_p(t) \equiv f(\tilde{t})$, $\tilde{t} = t \bmod 1$, we then get:

$$\begin{aligned} \mathcal{W}_\psi f(j, n) &= \int_0^1 f(t) \psi_{j,n}^p(t) dt = \sum_{k=-\infty}^{\infty} \int_0^1 f(t) \psi_{j,n}(t+k) dt \\ &= \sum_{k=-\infty}^{\infty} \int_k^{k+1} f(u-k) \psi_{j,n}(u) du \\ &= \sum_{k=-\infty}^{\infty} \int_k^{k+1} f_p(u) \psi_{j,n}(u) du = \int_{-\infty}^{\infty} f_p(u) \psi_{j,n}(u) du. \end{aligned} \quad (2.16)$$

Remark. If $f(0)$ is not equal to $f(1)$, f_p will clearly be discontinuous at the boundary of the interval. This means that the periodized wavelet coefficients $\psi_{j,n}^p$ with support outside $[0,1]$ will be affected by the potential

discontinuity. Figure 2.7 shows one such periodized wavelet $\psi_{j,n}^p$ with support outside $[0, 1]$. In fact, unless the function f and its derivatives are periodic, the wavelet transform can detect an irregularity close to the boundaries.

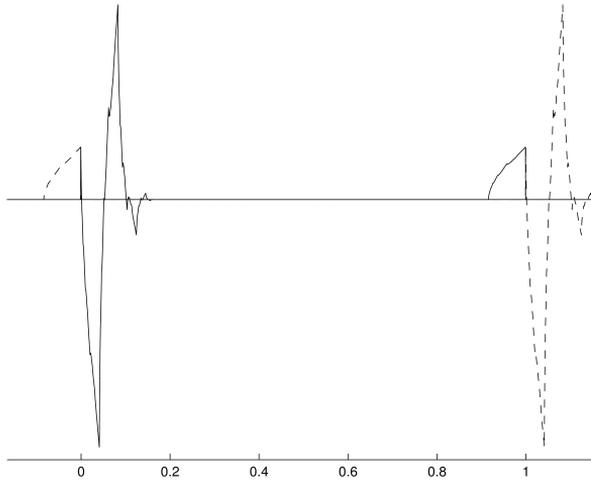


Figure 2.7: Example of a wavelet periodized over $[0, 1]$. The wavelet ψ depicted is the Daubechies 2 wavelet. The wavelet was created using `Wavelab .850` in `MATLAB`. The dashed line means that the wavelet is outside $[0, 1]$ (and thus continues on the opposite side of the interval).

The following theorem, from section 7.5 of [7], states that periodized wavelets, together with periodized scaling functions $\phi_{j,n}^p$, generate an orthogonal basis of $L^2[0, 1]$:

Theorem 2.3.1. Given $\{\psi_{j,n}\}_{(j,k) \in \mathbb{Z}^2}$, orthonormal wavelet basis of $L^2(\mathbb{R})$. For any $L \leq 0$,

$$\left[\{\psi_{j,n}^p\}_{-\infty < j \leq L, 0 < n \leq 2^{-j}}, \{\phi_{L,n}^p\}_{0 < n \leq 2^{-L}} \right]$$

is an orthogonal basis of $L^2[0, 1]$.

The above theorem states that the periodized discrete wavelet transform (2.15) yields an orthonormal representation of functions $f \in L^2[0, 1]$.

The projection of $f \in L^2[0, 1]$ onto \mathbf{V}_J , using periodized wavelets becomes

$$\begin{aligned} P_{\mathbf{V}_j} f_p(\tau) &= \sum_{n=1}^{2^{-j}} a_j^p[n] \phi_{j,n}^p(\tau), \\ P_{\mathbf{W}_j} f_p(\tau) &= \sum_{n=1}^{2^{-j}} d_j^p[n] \psi_{j,n}^p(\tau), \end{aligned} \tag{2.17}$$

where a_j^p and d_j^p are the *periodic wavelet coefficients* $f \in L^2[0, 1]$,

$$a_j^p[n] = \langle f, \phi_{j,n}^p \rangle, \text{ and } d_j^p[n] = \langle f, \psi_{j,n}^p \rangle.$$

The use of subscripted p 's, e.g., $P_{\mathbf{V}_j} f_p$ indicates the use of the periodized DWT applied on a function $f \in L^2[0, 1]$.

In section 2.3.3, we project the functions to some finite scale resolution, corresponding to the nature of the sampled data that is available. We therefore consider such projections of f in the equations that follow. The scale decomposition of periodized functions is:

$$\begin{aligned} P_{\mathbf{V}_J} f_p &= \sum_{j=J+1}^L \sum_{n=1}^{2^{-j}} d_j^p[n] \psi_{j,n} + \sum_{n=1}^{2^{-j}} a_L^p[n] \psi_{L,n} \\ &= \sum_{j=J+1}^L P_{\mathbf{W}_j} f_p + P_{\mathbf{V}_L} f_p, \end{aligned} \tag{2.18}$$

for an integer $L \leq 0$. The energy decomposition of the projection of $f \in L^2[0, 1]$ onto \mathbf{V}_J is

$$\begin{aligned} \|P_{\mathbf{V}_J} f_p\|_{L^2[0,1]}^2 &= \sum_{j=J+1}^L \|P_{\mathbf{W}_j} f_p\|_{L^2[0,1]}^2 + \|P_{\mathbf{V}_L} f_p\|_{L^2[0,1]}^2 \\ &= \sum_{j=J+1}^L \|d_j^p\|_{\ell^2}^2 + \|a_L^p\|_{\ell^2}^2. \end{aligned} \tag{2.19}$$

The equations corresponding to (2.18) and (2.19), for functions from $L^2(\mathbb{R})$ are (2.14) and (2.11).

2.3.2 Periodized cascading algorithm

We now look at how the cascading algorithm can be rewritten for periodized functions and their corresponding periodized wavelet coefficients. Observe that a_j^p and d_j^p are now discrete, periodic filters of periodic length 2^{-j} . As in the above section, $j \leq L \leq 0$. By the same argument used in (2.16), we can rewrite the coefficients in the following way:

$$\begin{aligned} a_j^p[n] &= \langle f, \phi_{j,n}^p \rangle_{L^2[0,1]} = \langle f_p, \phi_{j,n} \rangle_{L^2(\mathbb{R})}, \\ d_j^p[n] &= \langle f, \psi_{j,n}^p \rangle_{L^2[0,1]} = \langle f_p, \psi_{j,n} \rangle_{L^2(\mathbb{R})}. \end{aligned}$$

Remind that f_p is a periodic extension of $f \in L^2[0,1]$ to the real line. Inserting the sequences a_j^p and d_j^p into (2.12) of Theorem 2.2.4, changing to circular convolution, we see that the theorem still holds and:

$$\begin{aligned} a_{j+1}^p[q] &= \sum_{n=1}^{2^{-j}} h[n-2q] a_j^p[n] = a_j^p \otimes \bar{h}[2q], \\ d_{j+1}^p[q] &= \sum_{n=1}^{2^{-j}} g[n-2q] a_j^p[n] = a_j^p \otimes \bar{g}[2q]. \end{aligned} \tag{2.20}$$

Remark. Here, and in what follows, both the periodization and circular convolution on the j 'th scale will always correspond to a period of length 2^{-j} .

For the periodized reconstruction, corresponding to (2.13), we first define, for any sequence $x \in \ell^2$:

$$\check{x}[q] = \begin{cases} x[m] & \text{if } q=2m \\ 0 & \text{if } q=2m+1. \end{cases}$$

The above is simply an upsampling of x , inserting a zero between every two samples, and doubling its length. Again, we change to circular convolution,

obtaining:

$$\begin{aligned} a_j[n] &= \sum_{q=1}^{2^{-(j+1)}} h[n-2q]a_{j+1}^p[q] + \sum_{q=1}^{2^{-(j+1)}} g[n-2q]d_{j+1}^p[n] \\ &= \check{a}_{j+1}^p \otimes h[q] + \check{d}_{j+1}^p \otimes g[q]. \end{aligned} \quad (2.21)$$

Implementations of the equations (2.20) and (2.21) are the main components of computing DWT of a signal.

Remarks. 1) The calculation of DWT of a finite signal, using (2.20), is very efficient. In fact, the computation of the transform is quicker than the computation of FFT.

2) Other more complex, and actually more accurate, techniques for dealing with the boundary issues are known, see, i.e., section 7.5 of [7]. While other techniques can yield more accurate coefficients near the boundaries, they come with a cost: They are both harder to implement and have higher computational complexity, i.e., they are more computationally expensive to calculate. As we later argue; the error due to periodization is relatively small. More importantly, the sliding window, which we introduce in section 2.5, further reduces the effect of the error.

2.3.3 Initialization

The last few sections showed how to calculate DWT of functions with finite support in time, by applying a periodization procedure. As we pointed out above, what we usually have when analyzing a signal is a sequence of signal samples. In order to calculate the discrete wavelet transform based on these coefficients, in any meaningful way, we need to associate these sequences with a function, g , of which the DWT will be calculated. Note that we will not need the function explicitly, but it is a formal necessity.

We will denote the sequence of sample data by b , such that $b[n]$, $n = 1, \dots, N$ denote the measured samples. Working with DWT, we have to limit N to be of dyadic length, i.e., $N = 2^{-J}$, for some negative integer J .

We consider the periodized sequence

$$b^p = \{b^p[n]\}, \quad b^p[n] = b[n(\bmod N)],$$

and directly associate this sequence with a function $g \in L^2[0, 1]$, by

$$P_{V_J} g_p(\tau) = \sum_{n=1}^{2^{-J}} b^p[n] \phi_{J,n}^p(\tau), \quad \tau \in [0, 1]. \quad (2.22)$$

By comparison with the first equation of (2.17) we see that b^p now equals a_J^p , the scaling coefficients on scale J , of the function g . This means that calculating the DWT coefficients is a matter of applying the cascading algorithm (2.20) on $b^p = a_J^p$. Doing this recursively yields the wavelet and scaling coefficients of g on the scales $j = J + 1, \dots, L$. We denote DWT of $P_{V_J} g$ as a vector:

$$\mathcal{W}_\psi P_{V_J} g(j, n) = [a_L \mid d_L \mid d_{L-1} \mid \dots \mid d_{J+2} \mid d_{J+1}]. \quad (2.23)$$

The intended interpretation of the above equation is explained in the remark below.

Remark. In software libraries used for numerical computations of DWT, the coefficients are often saved as a vector of length 2^{-J} , equal to the signal length. One such way, used for instance in *Wavelab .850* [3], saves V_L in the first 2^{-L} elements, $[1, \dots, 2^{-L}]$. Thereafter W_j , $j \in \{L, \dots, J + 1\}$, is saved in the elements $[1 + 2^{-j}, 2^{-j+1}]$. Remind that $J \leq L \leq 0$ for finite signals. Figure 2.8 shows the vector containing the coefficients from Figure 2.6.

This is (one of) the standard ways to numerically calculate the wavelet transform. The above initialization, where we associate the elements of b with the periodized scaling coefficients of a finite function, leads to additional error. We discuss the error due to initialization after considering the error due to truncation in the next section.

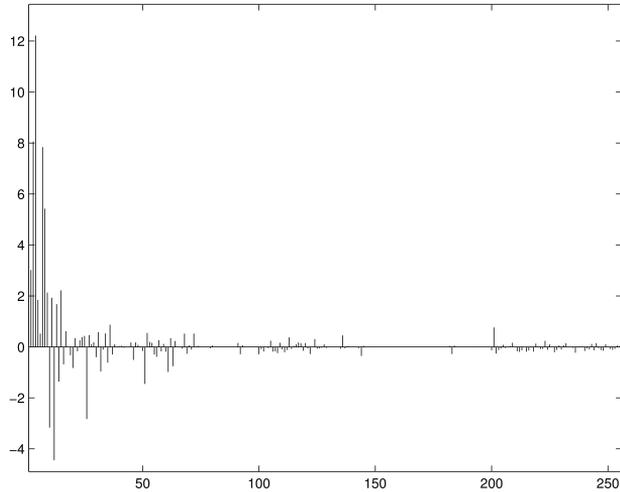


Figure 2.8: *The same scaling and wavelet coefficients as in Figure 2.6 stored in vector form. See (2.23) and the remark below it for explanation.*

2.3.4 Estimate of error

In this section, we discuss the error related to performing DWT within a single fixed window. Note that the effect of the error discussed in this section is reduced in our final algorithm. We discuss this in section 2.5.5.

There are two main sources of error in our calculation of DWT. The first comes from truncating the sequence $a_J = \{a_J[n]\}_{n=-\infty}^{\infty}$ by considering the subsequence $\{a_J[n]\}_{n=1}^{2^{-J}}$, for some negative integer, J . The second error comes from periodizing this truncated sequence, yielding a_J^p . This is essentially what we implicitly do in section 2.3.3.

Since we are considering the error induced by our method and not the error of the measurements, we ignore measurement and interpolation error of the original data.

Error due to truncating

We remind that the samples corresponds to a discrete sampling of a "real" signal, in our case the pressure measured at some point of the system. In order understand the numerical calculation of DWT, it is convenient to think of the *originating function* that the data is sampled from. Given the sequence, b of length $N = 2^{-J}$, we can assume

$$b[n] \approx f(t_0 + n), \quad n = 1, \dots, N,$$

where t_0 is some initial time of the physical system. The function f have to be scaled, so that $f(t)$ and $f(t+n)$ are located n sample units from each other. Since our data has minutely measurements, a change in time of n units simply corresponds to n minutes.

If we fix k , we know that the sample $b[k]$ is an averaging of f around $t_0 + k$, so we can assume

$$b[k] \approx \int_{-\infty}^{\infty} f(t) \phi(t - t_0 - k) dt,$$

where ϕ is a normalized *masking function* localized around zero. We assume that the mask function generates MRA, see (2.4), in other words that ϕ is a scaling function. Since scaling functions satisfy (2.7), this is an appropriate assumption.

We rescale the original function so that the interval containing the samples, $[t_0, t_0 + N]$, is scaled to $[0, 1]$:

$$g(\tau) = \sqrt{2^{-J}} f(t_0 + N\tau).$$

Then

$$b[n] \approx \int_{-\infty}^{\infty} g(\tau) \phi_{J,n}(\tau) d\tau = a_J[n], \quad n = 1, \dots, N, \quad (2.24)$$

where

$$\phi_{J,n}(\tau) = \frac{1}{\sqrt{2^J}} \phi\left(\frac{\tau - 2^J n}{2^J}\right).$$

Having only a finite sequence of coefficients, we will deal with the *truncated projection*

$$\tilde{P}_{\mathbf{V}_J}g(\tau) = \sum_{n=1}^N a_J[n]\phi_{J,n}(\tau) \approx \sum_{n=1}^N b[n]\phi_{J,n}(\tau), \quad (2.25)$$

which is concentrated mainly on $[0, 1]$. Note that $\tilde{P}_{\mathbf{V}_J}g$ is not equal to the projection $P_{\mathbf{V}_J}g$ close to $\tau = 0$ and $\tau = 1$, but yields an error. To see this, remind that the projection is given by

$$P_{\mathbf{V}_J}g(\tau) = \sum_{n=-\infty}^{\infty} a_J[n]\phi_{J,n}(\tau).$$

Assuming $\text{supp } \phi \subseteq [-C, C]$, yields

$$\text{supp } \phi_{J,n} = \left[\frac{n-C}{N}, \frac{n+C}{N} \right]. \quad (2.26)$$

We are only interested in the projection on $\tau \in [0, 1]$. We see that the coefficients affecting $P_{\mathbf{V}_J}g|_{\tau \in [0,1]}$ are $a_J[k]$ where k satisfies $-C \leq k < N+C$. This yields

$$\begin{aligned} P_{\mathbf{V}_J}g(\tau) &= \sum_{n=-C}^{N+C-1} a_J[n]\phi_{J,n}(\tau) \\ &\approx \sum_{n=1}^N b[n]\phi_{J,n}(\tau) \\ &\quad + \sum_{m=-C}^0 \sum_{m=N+1}^{N+C-1} a_J[m]\phi_{J,m}(\tau), \quad \tau \in [0, 1]. \end{aligned}$$

The error from truncating the sequence is

$$\begin{aligned} \mathcal{E}_t &= \|P_{\mathbf{V}_J}g - \tilde{P}_{\mathbf{V}_J}g\|_{L^2[0,1]} \\ &\approx \left\| \sum_{k \in \mathcal{A}_t} a_J[k]\phi_{J,k} \right\|_{L^2[0,1]} \\ &< \left\| \sum_{k \in \mathcal{A}_t} a_J[k] \right\|_{\ell^2} \end{aligned} \quad (2.27)$$

where $\mathcal{A}_t = \{-C, -C+1, \dots, 0, N+1, \dots, N+C-1\}$ is the set of k satisfying $\text{supp } \phi_{J,k} \cap [0, 1] \neq \emptyset$. The inequality of the last line is due to the fact that parts of $\phi_{J,k}$ lie outside $[0, 1]$. In fact, a relatively big part of the basis elements lie outside the interval, implying that the last estimate is quite rough.

As an example, we use the Symmlet 4 scaling function, see Figure 2.4, and set $J = -7$, i.e., $N = 2^7$. The quadrature mirror filter of the Symmlet 4, shown in Figure 2.9, has eight elements. By Theorem 2.2.2 we know that $C = 4$, yielding that \mathcal{A}_t has 8 elements that contribute to the error. From the above discussion, we know that only a portion of their energy contributes to the error. With a total of $N = 128$ coefficients, the relative error is quite small.

Remark. It can off course occur that the coefficients $a_J[k]$, $k \in \mathcal{A}_t$ are big compared to the coefficients that lie inside $[0, 1]$. This would off course increase the error due to the truncation. The sliding window, mentioned earlier in the text, will decrease the effect of this error.

Error due to periodization

We now consider error due to periodizing the truncated sequence. In other words, the error introduced by using (2.22) to approximate (2.25).

Like in the above discussion, we consider $f \in L^2(\mathbb{R})$ where $\tau \in [t_0, t_0 + 2^{-J}]$ is the area of interest. Scaling f , as we did in (??), we obtain g . We compare the projections $\tilde{P}_{V_J}g$ and $P_{V_J}g_p$:

$$\mathcal{E}_p = \left\| \sum_{n=1}^{2^{-J}} a_j[n] \phi_{j,n} - a_J^p[n] \phi_{J,n}^p \right\|_{L^2[0,1]}^2 = \left\| \sum_{n=1}^{2^{-J}} a_j[n] - a_J^p[n] \right\|_{\ell^2}^2. \quad (2.28)$$

Clearly $\text{supp } \phi_{J,k} \subset [0, 1]$ implies $\phi_{J,k}(\tau) = \phi_{J,k}^p(\tau)$, $\tau \in [0, 1]$. If we assume $\text{supp } \phi \subset [-C, C]$ (like in the discussion of truncation error), (2.26) yields that

$$a_J[k] = a_J^p[k], \quad k = C+1, \dots, N-C.$$

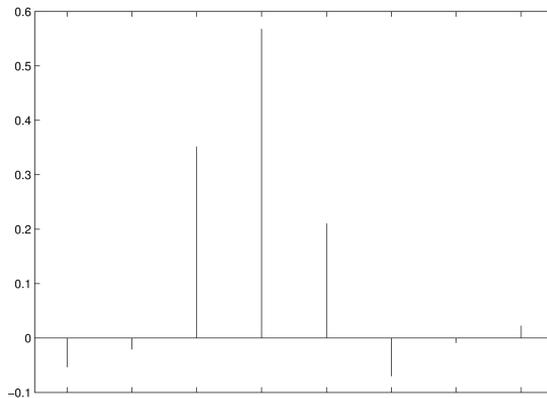


Figure 2.9: The eight quadrature mirror filter coefficients, h , for the Symmlet 4 scaling function. The corresponding scaling function is shown in Figure 2.4.

Equation (2.28) then simplifies to:

$$\mathcal{E}_p = \left\| \sum_{k \in \mathcal{A}_p} a_j[k] - a_j^p[k] \right\|_{\ell^2}^2. \quad (2.29)$$

Above, $\mathcal{A}_p = \{1, 2, \dots, C, N - C + 1, N - C + 2, \dots, N\}$ is the set of indices, k , where $a_J[k] = a_j^p[k]$. This means that $2C$ elements are biased and contribute to the error. If $N = 2^{-J}$ is sufficiently big, the relative number of coefficients that does not coincide is small.

As an example, we again use the Symmlet 4 scaling function, see Figure 2.4. The quadrature mirror filter for this scaling function has 8 coefficients, see Figure 2.9. This means that $C = 4$, implying that only 8 out of 128 coefficients, that is 6.25%, are biased.

Remark. Studying Figure 2.9, we see that 3 out of the 8 coefficients are dominant. When the smaller coefficients lie outside the boundaries of the sequence, the error they yield is relatively small. This improves the estimate further.

Total error

We now discuss the total error due to the truncation and periodization. Selecting b from a larger dataset is essentially equivalent to truncating an infinite sequence of measurements. By (2.24), the coefficients $b[k]$ are approximately $a_J[k]$, for $k = 1, \dots, N$. Thus the error due to truncation is \mathcal{E}_t . Similarly, having the finite sequence b , the error due to periodizing it is \mathcal{E}_p .

The total error can be estimated by comparing the original function, necessarily projected onto \mathbf{V}_J , $P_{\mathbf{V}_J}g$, with the function we base the calculation of DWT on, $P_{\mathbf{V}_J}g_p$, on the interval $[0, 1]$. The resulting error is:

$$\begin{aligned} \mathcal{E} &= \|P_{\mathbf{V}_J}g_p - P_{\mathbf{V}_J}g\|_{L^2[0,1]} \\ &\leq \|P_{\mathbf{V}_J}g_p - \tilde{P}_{\mathbf{V}_J}g\|_{L^2[0,1]} + \|\tilde{P}_{\mathbf{V}_J}g - P_{\mathbf{V}_J}g\|_{L^2[0,1]} \\ &= \mathcal{E}_p + \mathcal{E}_t, \end{aligned}$$

where \mathcal{E}_t and \mathcal{E}_p are given by (2.27) and (2.29).

2.4 Wavelet variance

In this section, we will show how the above theory can be used to decompose the variance of a sampled signal into a scalewise variance. Using orthonormal wavelet and scaling functions, we show that the variance can be decomposed into components corresponding to the scales of DWT.

Signal variance

In what follows we do not distinguish between the finite sequence X and its natural periodization. Remind that the estimated variance of a discrete sequence X , of length N , is

$$\text{Var}(X) = \frac{1}{N} \sum_{n=1}^N (X[n] - \bar{X})^2 = \frac{1}{N} \|X\|_{\ell^2}^2 - \bar{X}^2, \quad (2.30)$$

where \bar{X} denotes the mean of the sequence. Having the sequence of samples $b[n] = f(t_0 + n)$, $n = 1, \dots, N$, $N = 2^{-J}$ for some initial point t_0 , we fix $X[n] = b^p[n]$.

Like in section 2.3.3, we associate b^p with a function $g \in L^2[0, 1]$ by (2.22). Then b^p corresponds to the periodized scaling coefficients of g on scale J . Using the periodic cascading algorithm, we calculate the wavelet and scaling coefficients on scales $J+1, \dots, L$ with $L \leq 0$. By (2.19) we have

$$\|X\|_{\ell^2}^2 = \|b^p\|_{\ell^2}^2 = \sum_{j=J+1}^L \|d_j^p\|_{\ell^2}^2 + \|a_L^p\|_{\ell^2}^2, \quad (2.31)$$

which inserted into (2.30) yields

$$\text{Var}(X) = \frac{1}{2^{-J}} \left(\sum_{j=J+1}^L \|d_j^p\|_{\ell^2}^2 + \|a_L^p\|_{\ell^2}^2 \right) - \bar{X}^2. \quad (2.32)$$

This motivates the following definition of the *wavelet variance*, on scale j , of a signal of dyadic length $N = 2^{-J}$:

$$\nu_j^2 = \frac{1}{2^{-J}} \|d_j^p\|_{\ell^2}^2,$$

and the corresponding *scaling variance*:

$$\tilde{\nu}_j^2 = \frac{1}{2^{-J}} \|a_j^p\|_{\ell^2}^2.$$

The above definition of wavelet variance corresponds to the definition of Percival, see, e.g., [13]. Inserting the two expressions into (2.32), yields the wavelet decomposition of variance:

$$\text{Var}(X) = \sum_{j=J+1}^L \nu_j^2 + \tilde{\nu}_L^2 - \bar{X}^2.$$

We can now decompose the variance of a sequence into the components corresponding to behavior of different frequencies.

Figure 2.10 shows an example of this on two artificial signals. While the signals are clearly of a different nature, they have the same signal variance. In the figure, we see that the wavelet variance captures the different nature of the signals.

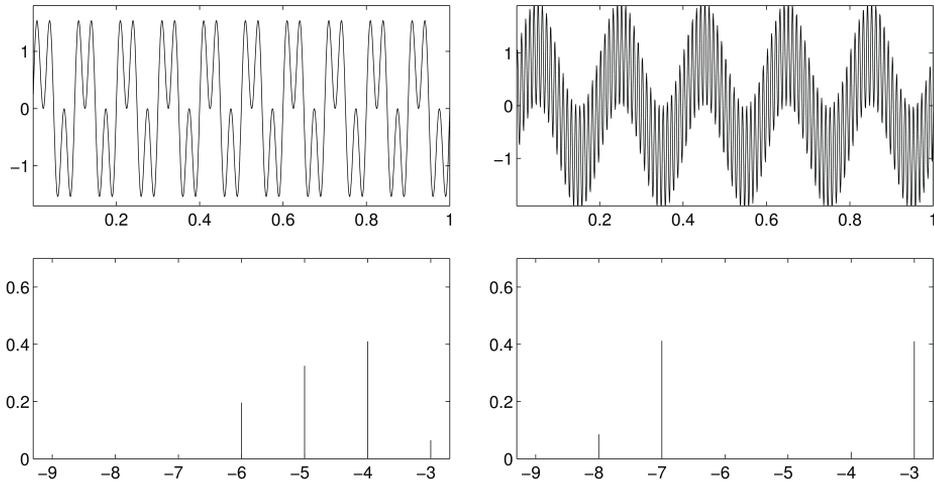


Figure 2.10: *Example of the decomposition of wavelet variance on two artificial signals. The signals are the sums of two sine functions with different frequencies. The sines of the first signal have 10 and 30 oscillations on $[0,1]$, while the second have 5 and 100. For both signals $N = 2^{10}$, implying $J = -10$, and the variance of the two signals are the same. The two bottom plots shows the wavelet variances of the signals on scales j , $J = -10 < j \leq L = -3$. The different variance captures the different oscillations well. The scaling variance is of little interest to us and is therefore not included in the figure.*

2.5 Sliding Window Wavelet Transform

In the last section, we dealt with signals restricted to a single, large scale window. We obtained local characteristics (wavelet variance) of the window content. We now wish to investigate the state of the system with respect to time. In order to do this, we introduce a sliding window and observe how these local characteristics change in time.

We fix $J \leq L \leq 0$ and consider

$$b^{t_m}[n] \approx f(t_m + n), \quad n = 1, \dots, 2^{-J},$$

with $t_m = t_0 + m$. We explicitly indicate the location of the window since the position in time now is quite central. We denote the periodization of b^{t_m} by:

$$X^{t_m} = (b^{t_m})^p. \quad (2.33)$$

As in section 2.4, we do not distinguish between X^{t_m} and its natural periodization. We can now apply the techniques from section 2.3.3 on X^{t_m} for all m . To denote the wavelet and scaling coefficients of X^{t_m} , we introduce the notation

$$\begin{aligned} W_j^{t_m}[n] &= d_j^p[n], \quad j = J + 1, \dots, L \\ V_L^{t_m}[n] &= a_L^p[n], \end{aligned}$$

where d_j^p , $j = J + 1, \dots, L$ and a_L^p are calculated using (2.20) with the initial coefficients $X^{t_0} = (b^{t_0})^p$. This leads to the time dependent wavelet and scaling variance

$$\nu_{j,t_0}^2 = \frac{1}{2^{-J}} \|W_j^{t_0}\|_{\ell^2}^2, \quad \text{and} \quad \tilde{\nu}_{j,t_0}^2 = \frac{1}{2^{-J}} \|V_j^{t_0}\|_{\ell^2}^2.$$

The decomposed signal variance of X^{t_m} is then

$$\text{Var}(X^{t_m}) = \sum_{j=J+1}^L \nu_{j,t_m}^2 + \tilde{\nu}_{L,t_m}^2 - (\bar{X}^{t_m})^2.$$

2.5.1 Sliding window wavelet transform

Fix $J \leq L \leq 0$ and some initial position in time, t_0 . Given $\{X^{t_m}\}_{m=0}^M$, for some integer M , where X^{t_m} is given by (2.33), the *sliding window wavelet transform* (SWWT), $\mathcal{S}_{J,L} : \mathbb{R}^{2^{-J}} \rightarrow \mathbb{R}^{L-J}$, is

$$\mathcal{S}_{J,L} X^{t_m} = (\nu_{J+1,t_m}^2, \nu_{J+2,t_m}^2, \dots, \nu_{L,t_m}^2), m = 0, \dots, M.$$

Note that $\tilde{\nu}_{L,t_m}^2$ is not included in $\mathcal{S}_{J,L}$. This is because we are not interested in scales more coarse than L . When J and L are fixed, we will omit them in the notation.

Figure 2.11 shows the result of applying the sliding window operator on the data from the first dataset of Figure 1.6. In the figure, $J = -7, L = -3$. Notice that the green and yellow scales (the two most detailed) "notice" something at 21:00 in the beginning of the interval, while the higher scales do not. On the big oscillation between 18:00 and 21:00, in the middle of the interval, also the two most coarse scales feel something going on. We use this phenomena to characterize different events in section 2.6.

2.5.2 Smoothing SWWT

The values of $\mathcal{S} X^{t_m}$ will often fluctuate around a local trend as the window moves in time. This makes the thresholding algorithm, defined in section 2.6, unstable. To obtain more stable values we apply a scalewise smoothing on the output of \mathcal{S} . Let \mathcal{B} be a sequence representing a rectangular window of length l with $\sum_{n=1}^l \mathcal{B}[n] = 1$. The *smoothened sliding window wavelet transform* is:

$$\mathcal{S}^{\mathcal{B}} X^{t_m} = (\nu_{J+1,t_m}^2 \star \mathcal{B}, \dots, \nu_{L,t_m}^2 \star \mathcal{B}),$$

where we only use the central part of of the convolution (of size M). We will always apply this convolution below. The smoothing of $\mathcal{S} X^{t_m}$ in Figure 2.11 is shown in Figure 2.12 (the first two plots).

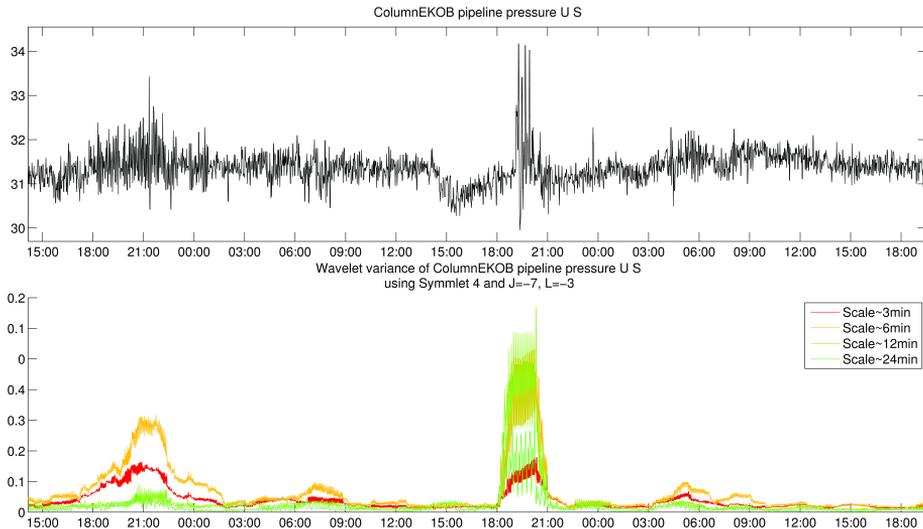


Figure 2.11: *Top subplot: The first dataset of the extract shown in Figure 1.6, Bottom subplot: SWWT of the above signal. Notice that the values of the wavelet variance fluctuates quite a bit from one point to the next, especially within the chaotic area between 18:00 to 21:00 towards the end of the signal. We smoothen these lines, obtaining a more stable, localized, trend in section 2.5.2, see Figure 2.12*

2.5.3 Position of the sliding window

When we defined X^{t_m} above, we related the coefficients to f by $X^{t_m}[n] = f(t_m + n)$, $n = 1, \dots, N$. In other words the window is located to the right of t_m . In numerical applications, other conventions are more convenient. The window is moved appropriately by substituting t_m with $t_m + \Delta_J$, for a suitable integer Δ_J .

When we wish to characterize events, we center the window around t_m , so that $\mathcal{S}^{\mathcal{B}}X^{t_m}$ is based on a symmetric window around t_0 . When we investigate the possibility of predicting events, we locate the window to the

left of t_m , obtaining a causal transform (indicating that the transform only depends on past and present data to calculate $\mathcal{S}^B X^{t_m}$).

2.5.4 Choice of scales

This far, we have not discussed the relationship between the scales of the wavelet transform and the physical scale it corresponds to. The parameters affecting the relationship are the sampling rate of the measurements, the window size $N = 2^{-J}$ and choice of wavelet/scaling function. We consider our system in particular in the discussion below.

Fix k and J . Then $X^{t_k} \approx a_j^{t_k}$, by (2.24). The scaling coefficients, $a_j^{t_k}$, are obtained with: $\phi_{J,n}$, $n = 1, \dots, N$. Minutely sampling rate and the symmlet 4 scaling function yields that these scaling coefficients have a main support ~ 3 minutes (by the arguments of section 2.3.4).

Applying the cascade databank algorithm, X^{t_k} is split in two: $W_{J+1}^{t_k}$ and $V_{J+1}^{t_k}$. The former contains the details of X^{t_k} on scales ~ 3 minutes, while the latter contains scales ~ 6 minutes and all more coarse scales. Repeating the procedure, $W_{J+2}^{t_k}$ contains the details on ~ 6 minute scales, $V_{J+2}^{t_k}$ contains ~ 12 minutes and everything more coarse and so on. Thus the wavelet coefficients on scale j , $J < j \leq L$ contain details on the scale $\sim 3 \cdot 2^{j-J-1}$ minutes.

We add the restriction $L < 0$. There are several arguments for this, one being that the biased interval grows as we increase j (while, however the relative error on the biased interval declines). This is outside the scope of this thesis and we do not discuss it any further and set a provisional upper boundary of L to be -3 . This is possibly artificially strict.

The choice of J decides the the global scale, e.g., the number of coefficients inside the window. The relationship between wavelet scale and global scale is independent of the numerical value, but the size of the window is, off course, affected. With a large number of elements in X_m^t , SWWT will become less sensitive to changes in time. On the other hand, a short short window will limit the number of scales in the analysis and will lack the wanted averaging (needed to obtain the localized trend), and thereby also be very sensitive to small signal changes.

Figure 2.12 shows a comparison of $\mathcal{S}_{J,L}^{\mathcal{B}} X^{t_m}$ with $J = -7, L = -3$ and $J = -6, L = -2$, using the same extract, X^{t_m} , as in Figure 2.11. The latter do seem to give a more localized characterization and also captures the same behavior as the former. However, it violates our upper boundary for L . With $J = -8$ we get a global scale of over four hours and the resulting SWWT gives less information than with $J = -7$. Unless anything else is specified, we use $J = -7, L = -3$ in what follows.

2.5.5 SWWT's effect on error

Fix k . The calculation of $\mathcal{S}^{\mathcal{B}} X^{t_k}$ then corresponds to calculating the wavelet variance of a single sequence X^{t_k} . In section 2.3.4 we discussed the error due to the truncation and periodization of X^{t_k} . Naturally, this error yields an error in wavelet variance. This is true for every k , indicating that $\mathcal{S}^{\mathcal{B}} X^{t_m}$ is somewhat biased for all m .

However, we assume that the spread around the average error is small. This implies that even if the computed values are not exact, they are quantitatively correct. In other words: when moving along in time, the effect of the error efficiently becomes smaller.

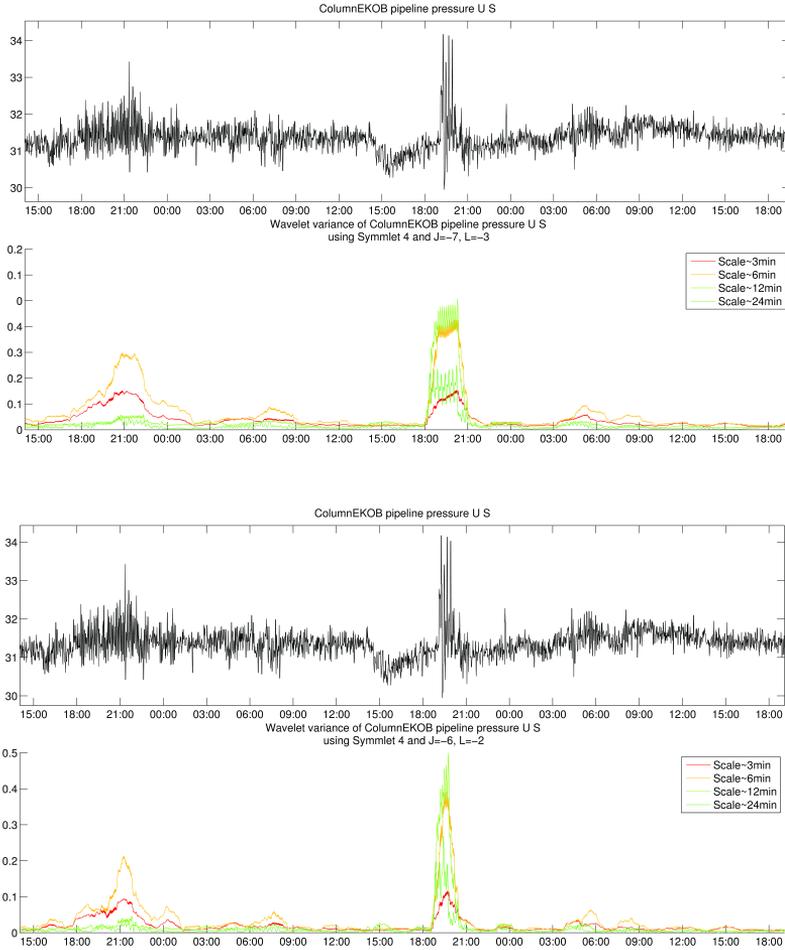


Figure 2.12: SWWT after smoothing with two different choices of J and L . The top pair shows the result when using $J = -7, L = -3$, the bottom pair shows $J = -6, L = -2$, i.e., we are considering the same physical scales. Notice that the detected events of the latter are better localized, but has $L = -2$. The signal extract is the first dataset of Figure 1.6.

2.6 Thresholding algorithm

Using SWWT, defined in the last section, we can now define a thresholding algorithm, indicating whether the signal is stable or not.

Given a dataset containing measurements from the system on the interval of interest, fix $J \leq L < 0$. Fix an initial point t_0 and an end point $t_M = t_0 + M$. Using the notation from the last section, X^{t_m} contains 2^{-J} measurements related to t_m in one of the ways discussed in section 2.5.3 (either with t_m in the middle of end of the window).

We denote the output of the (smoothened) SWWT at time t_m , i.e., $\mathcal{S}^B X^{t_m}$, by the column vector $\mathbf{S}^m = (\nu_{J+1,t^m}^2, \nu_{J+2,t^m}^2, \dots, \nu_{L,t^m}^2)^T$. With *weighting coefficients* $\boldsymbol{\alpha} = (\alpha_{J+1}, \alpha_{J+2}, \dots, \alpha_L)$, the *thresholding coefficients* are:

$$\boldsymbol{\alpha} \cdot \mathbf{S}^m = \sum_{j=J+1}^L \alpha_j \nu_{j,t^m}^2, \quad m = 0, \dots, M.$$

The *threshold value* $\delta \in \mathbb{R}$, leads to the *thresholding test*

$$\mathcal{T}_{\boldsymbol{\alpha},\delta}[m] = \mathcal{T}_{\boldsymbol{\alpha},\delta}(\mathbf{S}^m) = \begin{cases} 0, & \text{if } \boldsymbol{\alpha} \cdot \mathbf{S}^m < \delta \\ 1, & \text{if } \boldsymbol{\alpha} \cdot \mathbf{S}^m \geq \delta. \end{cases}$$

The thresholding test indicates whether a weighted sum of the local variability is higher than some prescribed value, δ .

Remark. The developed thresholding algorithm is appropriate for real time surveillance of a system. Assume that we use a causal window and that $\mathcal{T}_{\boldsymbol{\alpha},\delta}[m]$ is known for $0 \leq m < M$ before the value at time t_m is measured. When we know t_m , we only need to calculate $\mathcal{T}_{\boldsymbol{\alpha},\delta}[M]$, using X^{t_M} , which is then added to the already known measurements.

2.6.1 Algorithm parameters

The idea is that by choosing appropriately parameters $\boldsymbol{\alpha}$ and thresholding value δ , different characteristics of the signal can be detected. More specifically, we can characterize the signal based on its local variability on different scales.

Example

Figure 2.13 shows $\mathcal{T}_{\alpha,\delta}$ of the signal extract shown in Figure 2.12, with two different α . We set $J = -3, L = -3$. In the figure, $\alpha_1 = [1, 1, 0, 0]$, and $\alpha_2 = [0, 0, 1, 1]$. In other words, the first figure extracts details on a small scale, the second on a larger scale. The threshold levels are the same in both cases: $\delta = 0.05, \delta = 0.1, \delta = 0.25$. Note that the first result successfully captures the arcs that have high frequent oscillations and the second only captures the event with the larger time scale. An explanation of the plots follow.

Explanation of the thresholding test plot

The first subplot shows the input signal. The second subplot of the figures show the value of $\alpha \cdot \mathbf{S}^m$ (solid line) and it's components:

$$\alpha_{J+1}\nu_{J+1,t_m}^2, \alpha_{J+2}\nu_{J+2,t_m}^2, \dots, \alpha_L\nu_{L,t_m}^2$$

(the grey dashed lines) with respect to t_m . The horizontal lines represent threshold values δ .

The third subplot of the figures plots $\mathcal{T}_{\alpha,\delta}[m]$ with respect to m for the values δ . The colors in the third plot correspond to the colors of the vertical lines on the second. When $\mathcal{T}_{\alpha,\delta}[m] \geq 0$, it is indicated with a filled box. The higher boxes correspond to higher thresholds. The threshold values δ in the results are denoted by T in the legend box.

Parameter values

An appropriate choice of α and δ depends on the nature of the input signal. All three datasets analyzed in this thesis require different parameters (due to the differences in their variability). In addition, the average overall variability of the three time series is not constant over time, further complicating the matter. However this change of average variability occurs only a few times over a timespan of more than two years, so it is likely that system changes are the cause of this.

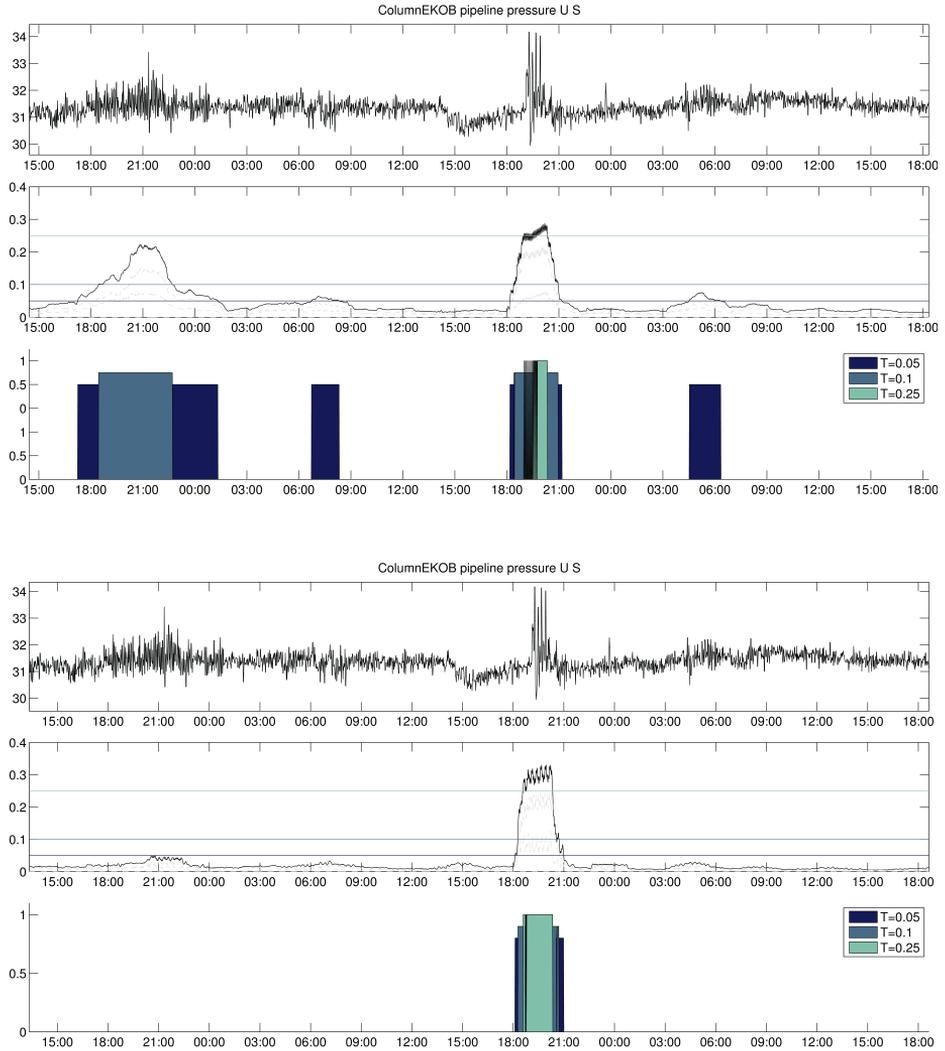


Figure 2.13: The result of the thresholding algorithm applied on the extract in Figure 1.6, first dataset, with $J = -7, L = -3$. For the first calculation $\alpha_1 = [1, 1, 0, 0]$, while for the second $\alpha_2 = [0, 0, 1, 1]$. The threshold values δ are denoted by T in the plot. An explanation of the plots is given in the text.

In the results presented in this thesis, the parameters are chosen manually, based on SWWT on the relevant interval of data. An automatic process for choosing these parameters, for instance based on statistical analysis, could be investigated.

2.7 Time series preparation

In section 1.3 we mentioned that the three datasets from our system, while being sampled every second, is stored roughly every forty second. The sampling rate of this data is not uniform, and large gaps in the data is not uncommon. The signal is fitted minutely intervals in the following way:

1. Every measurement is *fitted* to whole minutes. If there are multiple measurements within one minute, the average value is used
2. If there is no measurement on a given minute, the last available value is used

When the control system receives several data points where the change in the data is very small, its compression algorithm ignores the data. This justifies the second rule above. However, it can happen that data is actually missing over large intervals in time. We have tried to avoid such intervals in our examples.

The first rule in our dataset preparation can clearly yield signal samples that are not correct, but with no better data available, we are not left with much choice.

Remarks. 1) The data received from ConocoPhillips have already been fitted to minutely observations.

2) As we have mentioned, the limited sampling rate of the stored data is due to limitations of the control system. A newer control system, with an increased bandwidth, will increase this rate. At Ekofisk, there are plans to replace the control system, during 2013. This could improve the resolution of our algorithm, enabling analysis on finer scales.

Chapter 3

Results

In this chapter we present some examples that illustrate typical results achieved using the methods developed in this thesis.

All calculations are done on the Ekofisk time series described in section 1.3 and use the Symmlet 4 wavelet and scaling filter.

3.1 Detection and signal characterization

The characterization of different irregular behavior of the system, based on an analysis of the time series, was the initial goal of this thesis. The example shown in section 2.6.1 indicates that our method achieves this. In what follows, we consider a few more examples that indicate that our approach is successful. When we calculate SSWT in this section, we use a centering of the window, i.e., the samples in X^{t_m} are centered around t_m . The parameters α and δ are chosen manually, based on SWWT of the signal extract.

Example 1

In section 1.2.3 we pointed out that Figure 1.7 contains three obviously different types of behavior. We now try to extract the time intervals containing the different behaviors. We apply SWWT and the thresholding algorithm to the second dataset.

We use $J = -7$ and $L = -3$ and calculate $\mathcal{S}^{\mathcal{B}}X^{t_m}$, shown in Figure 3.1. Observe that the most coarse scale practically only detects the event

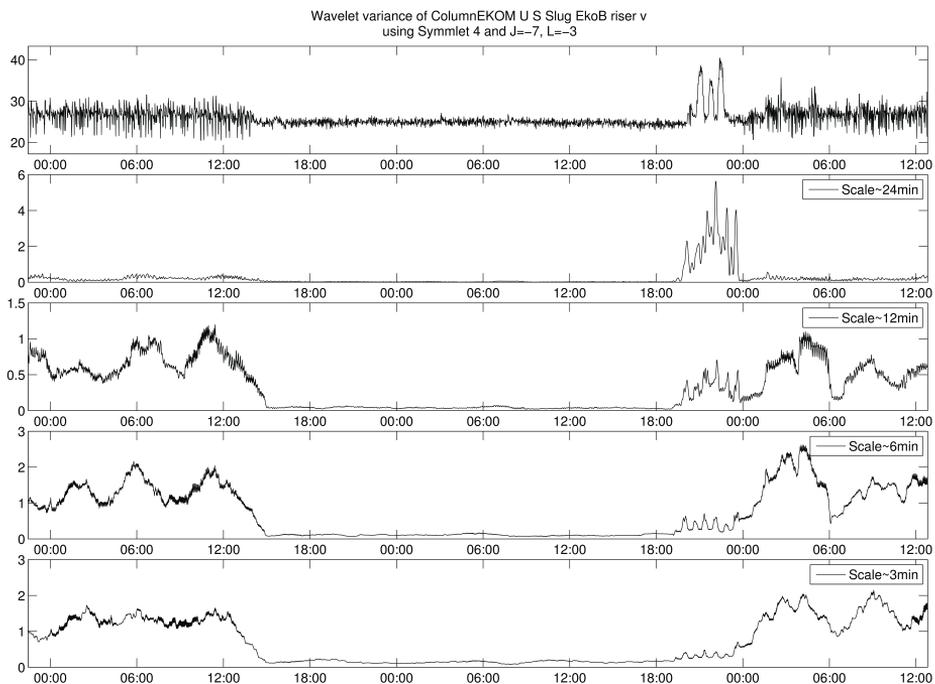


Figure 3.1: *SWWT of the second dataset on the interval shown in Figure 1.7. The scales are put in separate subplots in order to study them independently. The calculations use $J = -7$ and $L = -3$. The first plot shows the signal, while the rest represent the wavelet variance of the different scales, along the signal.*

between roughly 20:00 and 22:00, while the chaotic behavior in the very beginning and end of the signal is captured in the more detailed scale. We calculate $\mathcal{T}_{\alpha,\delta}$ with parameters $\alpha_1 = [1, 1, 0.5, 0]$ and $\alpha_2 = [0, 0, 0, 2]$ and $\delta = \{1, 2, 3\}$ (the latter are indicated by T in the figure). The result is shown in Figure 3.2. The algorithm successfully extracts the two wanted types of behavior we expected it to. Observe that the localization of the events is quite good.

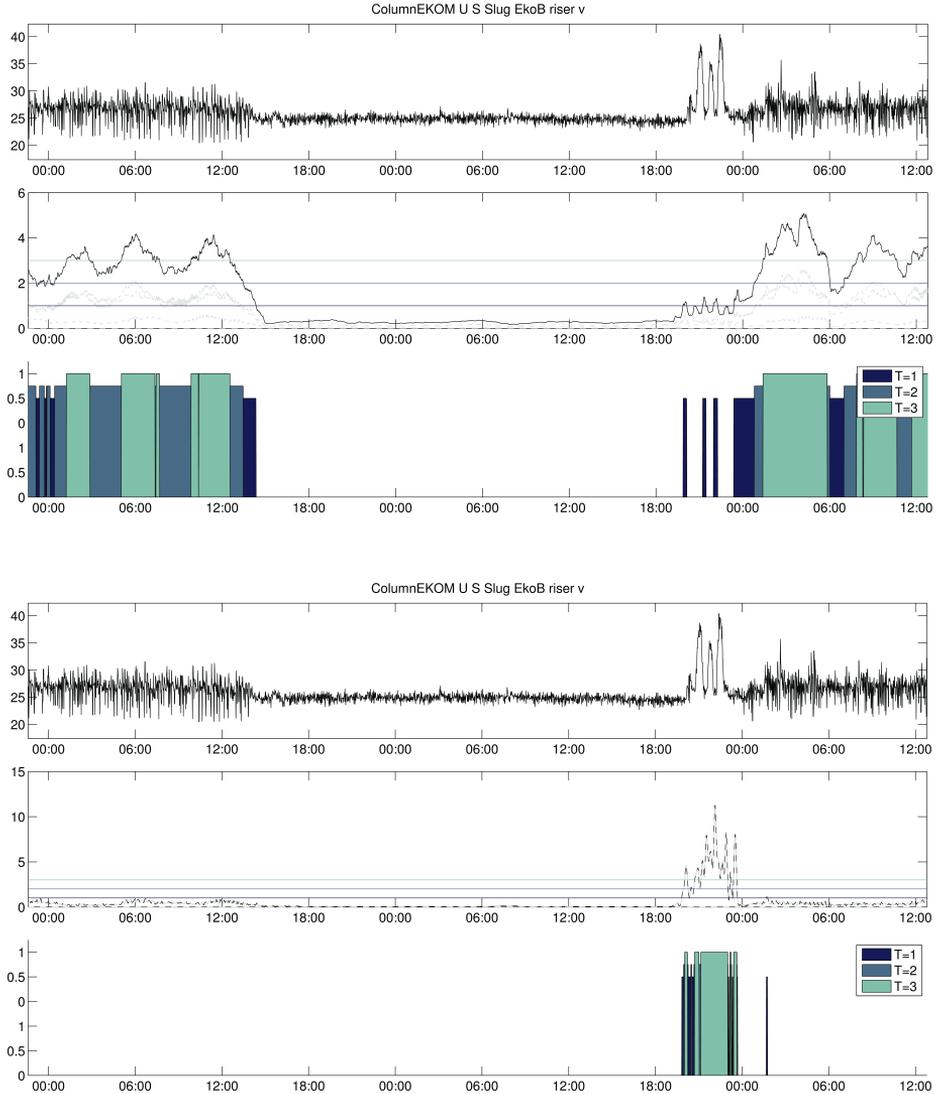


Figure 3.2: Results of the thresholding algorithm applied on the extract of Figure 1.7 (second dataset). The parameters of the first results are $\alpha_1 = [1, 1, 0.5, 0]$ and the second $\alpha_2 = [0, 0, 0, 2]$. Threshold values are $\delta = \{1, 2, 3\}$ (indicated by T in the figures). As expected, variability on high frequencies, i.e., small scales, are detected in the first, and variability on low frequencies, large scale, in the second.

Example 2

From experimenting with different time intervals, it generally seems that the the third dataset is harder to analyze than the first and second. We are however still able to gain some information from it.

The following example applies our procedure on the third dataset of the signal extract of Figure 3.3 (the same extract, second dataset, is shown in Figure 1.3, section 1.2.3). The information from the thresholding algorithm with $J = -7$ is in this case very limited. We therefore try the procedure with $J = -6$ and $L = -2$. The result is shown in Figure 3.4.

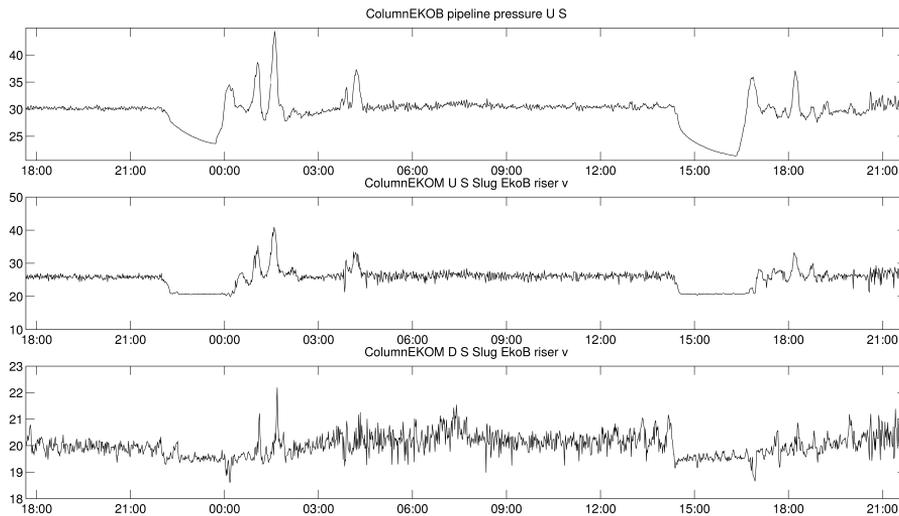


Figure 3.3: Dataset used in example 2. Signal extracted from 25 to 26 September 2011. This is the interval of the first example in Figure 1.3.

We choose to analyze the most detailed and the most coarse scales: $\alpha_1 = [10, 0, 0, 0]$ and $\alpha_2 = [0, 0, 0, 10]$ and $\delta = \{0.2, 0.4, .7\}$ (notice the high values of the weighting parameters). The results of $\mathcal{T}_{\alpha, \delta}$ are shown in Figure 3.5.

From the first result shown in Figure 3.5, we observe that the slightly

more chaotic interval of the extract, between 03:00 and 14:00 is marked. In the second result we also observe that the areas that seem to have a slightly higher instability (on a coarse scale) are detected. The main events from the first and second dataset are marked, in addition to a few other parts of the signal.

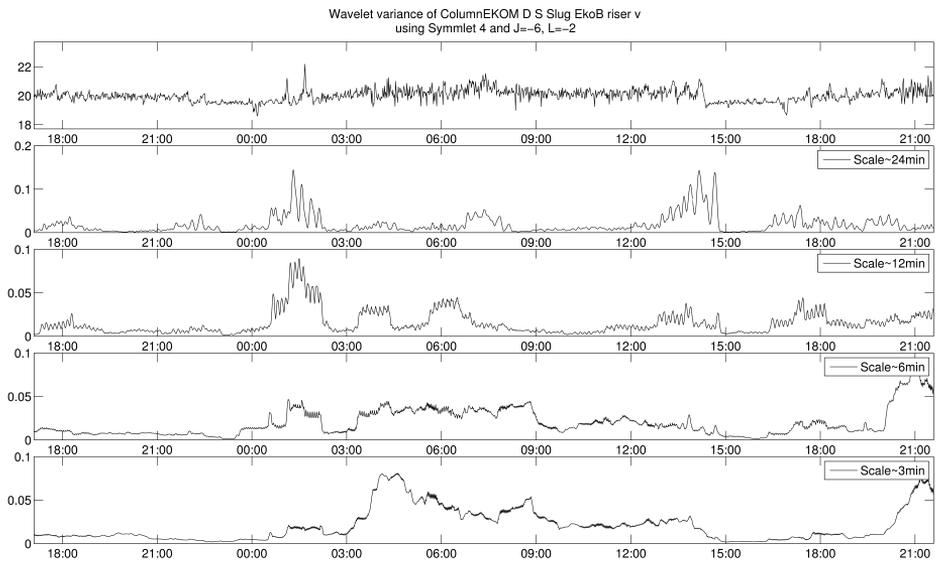


Figure 3.4: *SWWT of the third dataset on the interval shown in Figure 3.3. The extracted signal spans the time interval between the 25 to the 26 September 2011. Note that in this case $J = -6$ and $L = -2$.*

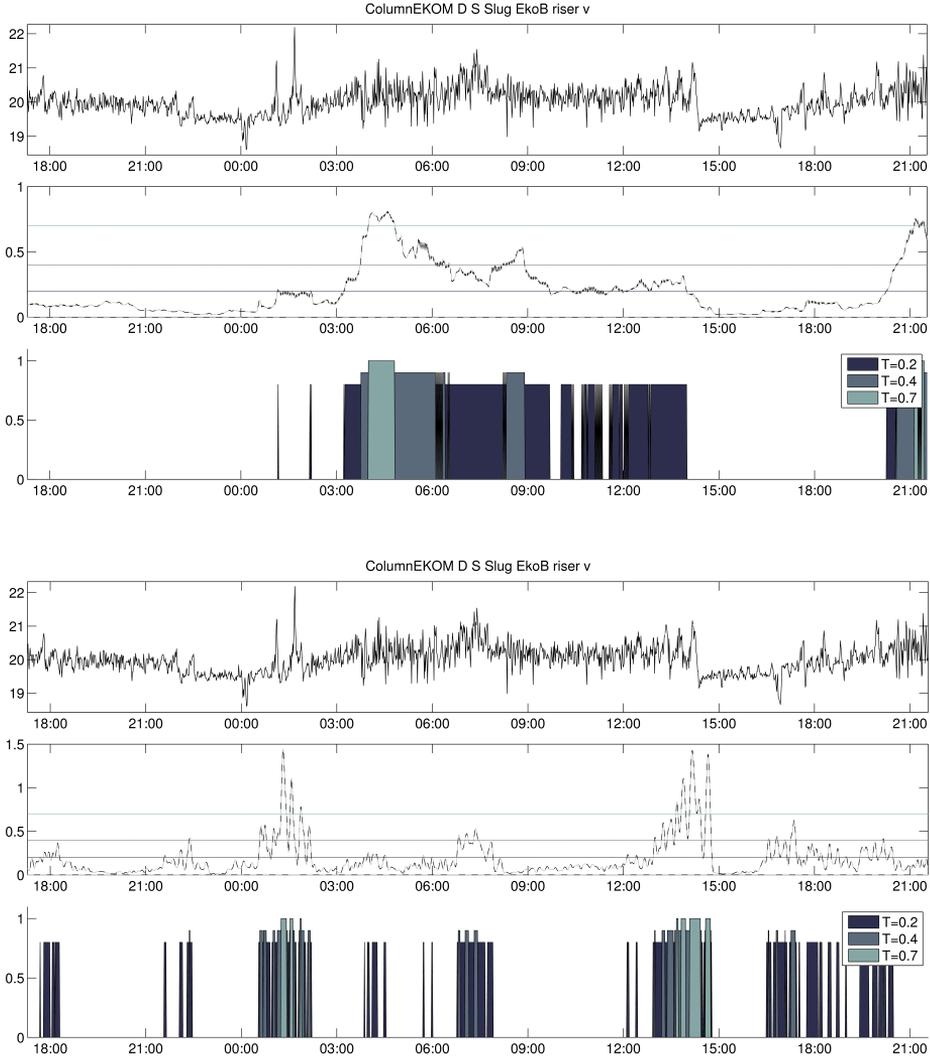


Figure 3.5: The result of the thresholding algorithm applied on the third dataset of the extract in Figure 3.3. For the first calculation $\alpha_1 = [10, 0, 0, 0]$, while for the second $\alpha_2 = [0, 0, 0, 10]$. The large values relatively low variation in the data. The threshold values are $\delta = \{0.2, 0.4, .7\}$.

3.2 Prediction

The idea that we could be able to predict events before they actually occur is related to the physical system itself. Instabilities on a small scale could slowly build up energy that eventually is released, leading to what we classify as unwanted events.

We will use a window X^{t_m} located to the left of t_m in this section, yielding a causal system, as discussed section 2.5.3. An increase in the variability on a scale at a point in time will then only depend on past values.

In this section we show some results of SWWT that indicate that the prediction of unwanted events is plausible. The example below use $J = -7$ and $L = -3$.

The first example is shown in Figure 3.6. The signal extract comes from the second dataset between the 28 and 29 June 2010. The increased variability in both the scale of ~ 3 and ~ 6 minutes could represent the cause of the sudden pressure change at around 04:00.

Similar behavior can be seen both in Figure 3.7 and in Figure 3.8. In Figure 3.7 the most detailed scale, ~ 3 minutes, increase, followed by the ~ 6 minute scale before the average pressure of the entire system drops several barg, leading to a decrease in both scales.

The same can be seen in Figure 3.8, where we observe an increase in the variability on the scale ~ 3 minutes a few hours before the mean pressure is suddenly lowered several barg.

We can naturally not draw conclusions solely from examples like the ones shown in this section. Our results do however indicate that our hypothesis about the possibility of developing a prediction system should be investigated further.

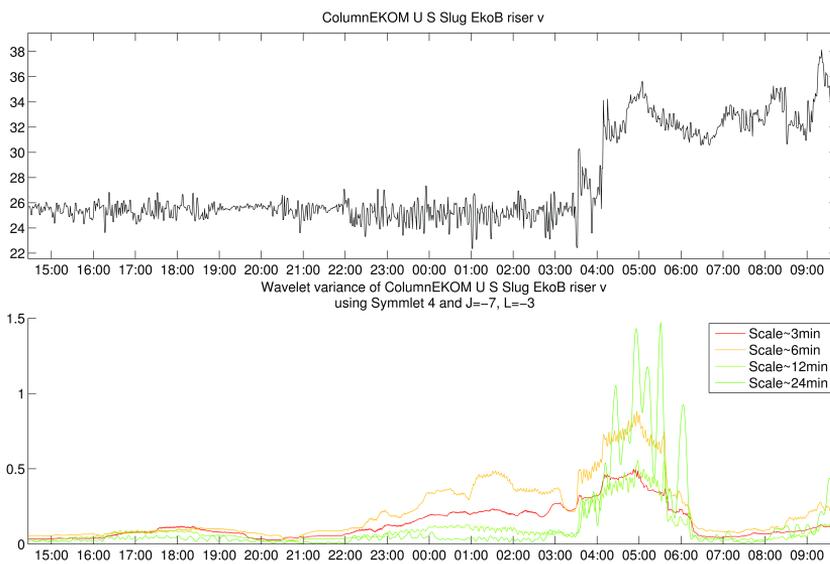


Figure 3.6: *An example where the variability on the scales of ~ 3 and ~ 6 minutes seems to be building up before an event on a larger scale occurs. From 28 to 29 June 2010.*

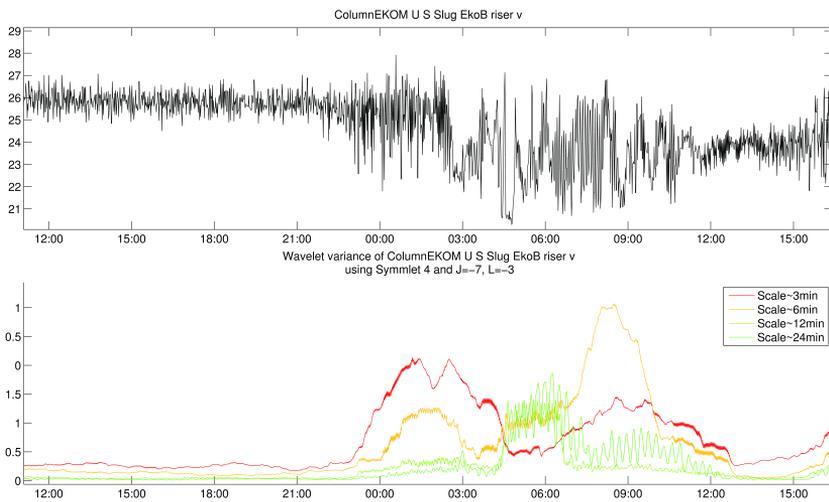


Figure 3.7: *An increase of variability on the two most detailed scales before the average pressure drops. This extract comes from the second dataset. Extract from 16 to 19 March 2011.*

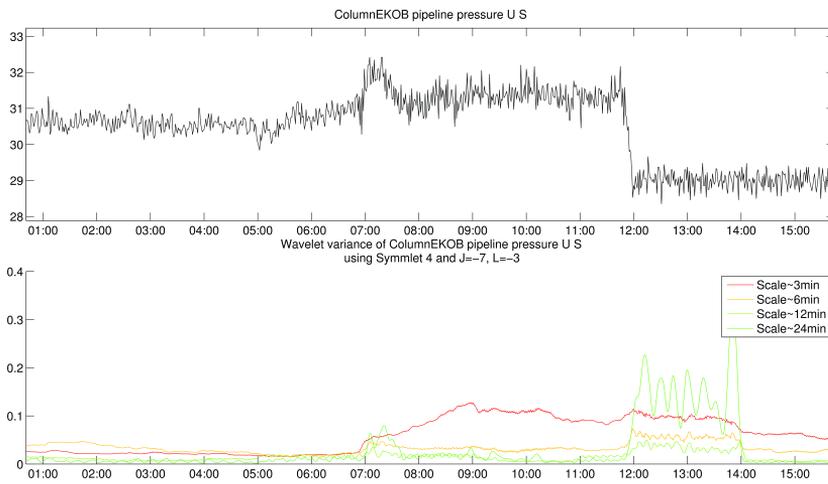


Figure 3.8: Like in both Figure 3.6 and 3.7, the variability on a small scale increase before a change on a larger scale occurs. This extract comes from the first dataset between 14 to 16 April 2011.

Chapter 4

Concluding remarks

In this thesis we have developed procedures that enable the characterization of certain complex system. In particular, we have analyzed a system where variability on different scales are of interest.

With the sliding window wavelet transform (SWWT) we created a time dependent scalewise decomposition of variance. By weighting SWWT appropriately and thresholding the output, we showed how to detect different types of special events.

We also found indications that SWWT could be applied to predict future events in the system, before they do occur, by detecting a build-up of variability on small scales that precedes special events in the measurements of the signal.

While the numerical examples of this thesis were based on pressure measurements in a pipeline with three phase flow, the techniques applied would also work on systems where instability of the system is relevant.

4.1 Further work

There are multiple ways the theory and techniques developed in this thesis could, and should, be developed further.

A statistical analysis for optimizing the parameters of the thresholding algorithm should be carried out. This would be necessary in order to apply the method in the industrially. The large amount of available data should enable this to be done.

Further, as mentioned in the introduction of the thesis, the amount of available data is growing fast in the oil industry. In the particular system our datasets come from, there are several other types of available data, one of which is the water content of the oil leaving the HP separator. This is one of the key factors in the system. Relating this, and possibly other, data with the results from our method would be a very interesting and would quite possibly give promising results.

Our hypothesis about prediction of events is also quite interesting and deserves further investigation.

Both the case shown in Figure 2.12 and the discussion around the second example of section 3.1 indicates a shorter window yields a more localized, but still qualitatively similar SWWT. This indicates that a survey regarding the, provisional, upper boundary of L should be carried out.

The above are just a few ideas, out of the many possibilities.

Bibliography

- [1] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM: Society for Industrial and Applied Mathematics, 1992.
- [2] S. Durand and J. Froment. Artifact free signal denoising with wavelets. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 6, pages 3685 –3688 vol.6, 2001.
- [3] David Donoho et.al. Wavelab .850, matlab library for numerical calculations of signals. <http://www-stat.stanford.edu/wavelab/>, May 2012.
- [4] John-Morten Godhavn, Mehrdad P. Fard, and Per H. Fuchs. New slug control strategies, tuning rules and experimental results. *Journal of Process Control*, 15(5):547 – 557, 2005.
- [5] Kjetil Havre and Morten Dalsmo. Active feedback control as the solution to severe slugging. Conference paper, Jan 2001.
- [6] Henrik Stray Kjetil Havre, Karl Ole Stomes. Taming slug flow in pipelines. *ABB Review 4*, pages 55–63, 2000.
- [7] S.G. Mallat. *A wavelet tour of signal processing: the sparse way*. Elsevier /Academic Press, 2009.

- [8] Stephane Mallat and Wen Liang Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, pages 617–643, 1992.
- [9] Stephane Mallat and Sifen Zhong. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 710 –732, 1992.
- [10] M.W. Marcellin, M.J. Gormish, A. Bilgin, and M.P. Boliek. An overview of jpeg-2000. In *Data Compression Conference, 2000. Proceedings. DCC 2000*, pages 523 –541, 2000.
- [11] Nefitemer. Active slug control technology for the oil industry. <http://www.nefitemer.com/slugging-control.html>, April 2012.
- [12] Norwegian Ministry of Petroleum, Energy, and Norwegian Petroleum Directorate. Facts 2012 - the norwegian petroleum sector, 2012.
- [13] Donald B. Percival and Andrew T. Walden. *Wavelet Methods for Time Series Analysis (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, 2000.
- [14] Sintef. Laboratory values - 25 years of multiphase flow transportation. <http://www.sintef.no/home/SINTEF-Petroleum-Research/Wellstream-Technology/Multiphase-flow/Laboratory-Values-movie/>, April 2010.
- [15] G. Strang. Eigenvalues of $(\text{darr};2)h$ and convergence of the cascade algorithm. *Signal Processing, IEEE Transactions on*, 44(2):233 –238, feb 1996.
- [16] M. Vetterli. Wavelets, approximation, and compression. *Signal Processing Magazine, IEEE*, 18(5):59 –73, sep 2001.
- [17] M. Vetterli and J. Kovačević. *Wavelets and subband coding*. Prentice Hall PTR Upper Saddle River, NJ, 1995.