



ONE-DIMENSIONAL REAL-TIME SIGNAL DENOISING USING  
WAVELET-BASED KALMAN FILTERING

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT DURMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

MAY 2007

Approval of the Graduate School of Natural and Applied Sciences.

---

Prof. Dr. Canan Özgen  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

---

Assoc. Prof. Dr. H. Şebnem  
DÜZGÜN  
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

---

Assoc. Prof. Dr. Mahmut  
Onur KARSLIOĞLU  
Supervisor

### **Examining Committee Members**

Prof. Dr. Volkan ATALAY (METU, CENG) \_\_\_\_\_

Assoc. Prof. Dr. M. Onur KARSLIOĞLU (METU, CE) \_\_\_\_\_

Prof. Dr. Vedat TOPRAK (METU, GEOE) \_\_\_\_\_

Assoc. Prof. Dr. Sadık BAKIR (METU, CE) \_\_\_\_\_

Dr. Uğur LELOĞLU (TUBİTAK-UZAY) \_\_\_\_\_

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: Murat DURMAZ

Signature :

# ABSTRACT

## ONE-DIMENSIONAL REAL-TIME SIGNAL DENOISING USING WAVELET-BASED KALMAN FILTERING

DURMAZ, Murat

M.S., Department of Geodetic and Geographic Information Technologies

Supervisor : Assoc. Prof. Dr. Mahmut Onur KARSLIOĞLU

May 2007, 74 pages

Denoising signals is an important task of digital signal processing. Many linear and non-linear methods for signal denoising have been developed. Wavelet based denoising is the most famous nonlinear denoising method lately. In the linear case, Kalman filter is famous for its easy implementation and real-time nature. Wavelet-Kalman filter developed lately is an important improvement over Kalman filter, in which the Kalman filter operates in the wavelet domain, filtering the wavelet coefficients, and resulting in the filtered wavelet transform of the signal in real-time. The real-time filtering and multiresolution representation is a powerful feature for many real world applications.

This study explains in detail the derivation and implementation of Real-Time Wavelet-Kalman Filter method to remove noise from signals in real-time. The filter is enhanced to use different wavelet types than the Haar wavelet, and also it is improved to operate on higher block sizes than two. Wavelet shrinkage is integrated to the filter and it is shown that by utilizing this integration more noise suppression is obtainable. A user friendly application is developed to import, filter and export signals in Java programming language. And finally, the applicability of the proposed method to suppress noise from seismic waves coming from earthquakes and to enhance spontaneous potentials measured from groundwater wells is also shown.

Keywords: Wavelet-Kalman Filter, Kalman Filtering, Wavelets, Wavelet Shrinkage,  
Noise removal

# ÖZ

## TEK BOYUTLU SİNYALLERDE DALGACIK TABANLI KALMAN FİLTRESİ KULLANARAK GERÇEK ZAMANLI GÜRÜLTÜ BASTIRILMASI

DURMAZ, Murat

Yüksek Lisans, Jeodezi ve Coğrafi Bilgi Teknolojileri Bölümü

Tez Yöneticisi : Doç. Dr. Mahmut Onur KARSLIOĞLU

Mayıs 2007, 74 sayfa

Sayısal sinyal işlemenin önemli bir görevi sinyallerdeki gürültüyü bastırmaktır. Sinyallerdeki gürültünün bastırılması için bir çok doğrusal ve doğrusal olmayan teknik geliştirilmiştir. Dalgacık tabanlı gürültü bastırma son zamanların en ünlü doğrusal olmayan tekniğidir. Kalman filtresi ise doğrusal olanlar içinde basitçe uygulanabilir olması ve gerçek zamanlı kullanılabilir olmasından dolayı tanınır. Son zamanlarda geliştirilen Dalgacık-Kalman filtresi normal Kalman filtresi üzerine önemli bir ilave- dir ki bu filtre Dalgacık tanım kümesinde sinyalin dalgacık katsayılarını filtreler ve sonuç olarak filtrelenmiş sinyalin Dalgacık dönüşümünü gerçek zamanlı olarak hesaplar. Gerçek-zamanlı filtreleme ve çoklu çözünürlüklü gösterim bir çok gerçek sinyal işleme problemi için çok önemlidir.

Bu çalışma sinyaller içindeki gürültüyü bastırmak için, Gerçek-Zamanlı Dalgacık-Kalman filtresi metodunun türetilmesini ve geliştirilmesini ayrıntılı biçimde anlatır. Filtre Haar dalgacık fonksiyonu dışındaki diğer fonksiyonları da kullanabilecek biçimde geliştirilmiştir. Filtre aynı zamanda daha yüksek blok büyüklüklerinde de çalışacak şekilde genişletilmiştir. İlave olarak, metoda Dalgacık sıkıştırma tekniği entegre edilerek daha fazla gürültü bastırılabilceği gösterilmiştir. Bu çalışmada ayrıca, sinyallerin açılması, işlenmesi ve saklanması amacıyla kullanıcı dostu bir bilgisayar programı geliştirilmiştir. Son olarak, metodun sismik dalgalardaki gürültünün

batırılması ve yeraltı kuyularında ölçülen doğal potansiyel sinyalinin düzeltilmesi amacıyla kullanılabilirliği gösterilmiştir.

Anahtar Kelimeler: Dalgacık-Kalman, Dalgacık, Kalman Filtresi, Dalgacık Sıkıştırma, Sismik, Gürültü bastırma

To my family

## ACKNOWLEDGMENTS

I am very grateful to Assoc. Prof. Dr. Mahmut Onur KARSLIOĞLU for his guidance suggestions, explanations and encouragement throughout my study.

I am also grateful to Prof. Dr. Volkan ATALAY, Prof. Dr. Vedat TOPRAK, Assoc. Prof. Dr. Sadık BAKIR and Dr. Uğur LELOĞLU for their valuable comments and guidance.

Special thanks to my wife Melek Eda and my son Can for their endless motivation, understanding and encouragement.

I would like to thank BİLGİ G.I.S. family for their support, understanding and motivation. I also would like to thank to hydrogeology engineer Dr. Ahmet APAYDIN, geophysical engineer Erol BOZ and geophysical engineer İbrahim Görgünoğlu(MS) from DSİ (Devlet Su İşleri) for providing me the well logs and valuable information about measurement devices. Last but not least, I am also grateful to Earthquake Engineering Research Center METU for providing the prefiltered and original seismic signals of Gönen earthquake.

And finally, I would like to thank all of my teachers that lead my education life, without whom I could not establish a base of understanding and knowledge of science.



	2.3.2	Derivation . . . . .	13
	2.3.3	Recursive Parameter Estimation . . . . .	14
2.4		Kalman Filter . . . . .	16
	2.4.1	Discrete Kalman Filter . . . . .	16
		2.4.1.1 Derivation . . . . .	16
	2.4.2	Example: Estimating a Constant . . . . .	21
3		MULTI RESOLUTION REPRESENTATION AND THRESHOLDING .	23
	3.1	Fourier Transform . . . . .	23
		3.1.1 Windowed Fourier Transform . . . . .	25
	3.2	Wavelets and Continuous Wavelet Transform . . . . .	26
		3.2.1 Multiresolution Representation . . . . .	28
		3.2.2 Wavelet Shrinkage . . . . .	34
4		WAVELET-KALMAN FILTER . . . . .	36
	4.1	Block Equations . . . . .	37
	4.2	Wavelet Decomposition . . . . .	40
	4.3	Integrating The Wavelet Shrinkage . . . . .	42
	4.4	Monte-Carlo Simulations and Results . . . . .	42
5		APPLICATIONS . . . . .	49
	5.1	Application to Seismic Waves . . . . .	50
	5.2	Application to Well Logs . . . . .	54
6		CONCLUSION . . . . .	60
		REFERENCES . . . . .	62
		APPENDIX	
	A	THE WAVELET-KALMAN EQUATIONS FOR 8-BLOCK . . . . .	64
		A.1 Process noise covariance matrix $Q$ . . . . .	64
		A.2 The System linearmodel matrix $A$ . . . . .	66
	B	SOFTWARE . . . . .	70
		B.1 Definition . . . . .	70
		B.2 Design and Algorithms . . . . .	70

## LIST OF TABLES

### TABLES

Table 4.1	Monte-Carlo Performance of filters of level two . . . . .	45
Table 4.2	Monte-Carlo Performance of filters of level four . . . . .	46
Table 4.3	Monte-Carlo Performance of filters of level six . . . . .	46
Table 4.4	Monte-Carlo Performance of filters of level six, with thresholding based on MAD variance estimation . . . . .	47
Table 5.1	2 level Wavelet-Kalman Performance Comparisons on seismic signal	53
Table 5.2	4 level Wavelet-Kalman Performance Comparisons on seismic signal	54
Table 5.3	6 level Wavelet-Kalman Performance Comparisons on seismic signal	54
Table 5.4	Change of SNR with process variance . . . . .	58
Table 5.5	Change of SNR with Wavelet-Kalman filter level . . . . .	59
Table 5.6	Change of SNR with MAF kernel width . . . . .	59

## LIST OF FIGURES

### FIGURES

Figure 2.1 Kalman Filter execution diagram . . . . .	20
Figure 2.2 Measurements from a constant process . . . . .	21
Figure 2.3 Different process variances and their effect on the results of the Kalman Filter. . . . .	22
Figure 3.1 Representation of a point in 3-D cartesian coordinate system (a) and representation of a function in frequency domain (b) . . . . .	24
Figure 3.2 Daubechies Wavelet and scaling function . . . . .	27
Figure 3.3 Forward Filter Bank . . . . .	30
Figure 3.4 Inverse Filter Bank . . . . .	30
Figure 3.5 Sine wave and its wavelet transform . . . . .	31
Figure 3.6 Noisy sine wave and its wavelet transform . . . . .	34
Figure 4.1 Wavelet-Kalman execution simple diagram . . . . .	37
Figure 4.2 Monte-Carlo simulation original Signal . . . . .	44
Figure 4.3 Monte-Carlo simulation noisy Signal . . . . .	44
Figure 4.4 Monte-Carlo simulation Wavelet-Kalman Filter Result . . . . .	45
Figure 4.5 Monte-Carlo simulation Kalman Filter Result . . . . .	46
Figure 4.6 Thresholded Wavelet Coefficients . . . . .	47
Figure 4.7 Reconstructed signal after thresholding . . . . .	48
Figure 5.1 GSR-16 accelerometer . . . . .	50
Figure 5.2 Location of a sample GSR-16 observation site (a) and its inside (b) (Afet-İşleri, 2007) . . . . .	50
Figure 5.3 Gönen Earthquake 20/10/2006 . . . . .	51
Figure 5.4 Frequency response of Gönen Earthquake and Buttleworth filter . . . . .	52
Figure 5.5 Reconstructed signal after Butterworth Filtering . . . . .	52
Figure 5.6 Reconstructed signal after Wavelet-Kalman Filtering . . . . .	53
Figure 5.7 Simplified Well Logging Schema . . . . .	55
Figure 5.8 Sample SP Log . . . . .	56
Figure 5.9 A Noisy Well Log . . . . .	57
Figure 5.10 MAF Filtered Well Log with kernel=5 . . . . .	57
Figure 5.11 Wavelet-Kalman Filtered Well Log with level 4 . . . . .	58
Figure B.1 Wavelet UML Diagram . . . . .	71
Figure B.2 Wavelet UML Diagram . . . . .	71
Figure B.3 Screenshot of Torpu (the software developed) . . . . .	73
Figure B.4 another screenshot of Torpu (the software developed) . . . . .	74

# CHAPTER 1

## INTRODUCTION

We sense our environment with observations made by measuring devices. Although the technology have developed new high quality measuring devices, there still exists uncertainty in our measurements. Therefore removal of the noise (the unwanted signal parts in general) from the observed signal is a very important part of any signal processing task. In this sense, also a parameter estimation procedure can be regarded as a noise reduction method in an optimal way.

The first formal method of finding an optimal estimate from noisy data is the method of least squares, which is introduced by Gauss in 1809 (Grival and Andrews, 2001). Wiener (1964) proposed a new time-invariant filter that uses the signal-to-noise ratio (SNR) of the signal to find the solution for the least-mean-squared prediction error of the signal (Grival and Andrews, 2001).

The Wiener Filter is useful in predicting the signal in future in the presence of noisy measurements. However, it only works for time-invariant (stationary) signals. Moreover it is hard to find a recursive, computationally efficient discrete algorithm for it (Minkler and Minkler, 1993). In 1960s Kalman introduced a new sequential and recursive discrete-time filter for time-varying or time-invariant signals which can easily be implemented by computers (Minkler and Minkler, 1993). His work and other achievements in the area is explained in detail in Chapter 2.

The above methods are based on linear systems theory. They use linear systems of equations to describe the signal. On the other hand, the Fourier Transform have been used by scientists and engineers to represent signals in frequency domain since 1822. By the invention of FFT(Fast Fourier Transform) and improvements in computational power, other types of noise removal techniques have been introduced. These noise removal techniques use the frequency response of white noise in the frequency

domain. Since white noise has a flat spectrum in frequency domain, a thresholding filter can be implemented by a prior knowledge of the noise characteristics (Petrou and Bosdogianni, 1999). These kind of filters can also be implemented as convolution filter in time domain which makes it computationally efficient. They are called low pass filters since they only allow low frequency components in the resulting signal. Although the Fast Fourier Transform gives a computationally fast orthogonal transform for signal processing, it does not have time resolution, which makes it difficult to analyse the noise content of the signal in time domain (Mallat, 1998). To overcome this problem Gabor (1946) introduced Windowed Fourier Transform. Windowed Fourier Transform is a transform where a Gabor atom is convolved with the signal resulting in frequency variations of the signal over time (Mallat, 1998).

Grossman and Morlet (1984) showed that scaling and dilating a single function “wavelet” can be used to construct the so called Continuous Wavelet Transform to analyse a given signal in both frequency and time domain. Haar (1910) introduced the simplest form of wavelet function called Haar Wavelet.

The Continuous Wavelet Transform is simple to implement, but it is not computationally efficient. Mallat (1989) showed that the continuous wavelet transform can be implemented in a computationally efficient form with a dyadic multiresolution analysis. Instead of scaling and translating the wavelet function over continuous scales and time, dyadic scaling and translating functions are introduced. Moreover a filterbank for transforming dyadic wavelet transform is also introduced. In Addition, he showed a way to systematically construct wavelet basis functions (Mallat, 1989). After the theoretical and practical basis for wavelet transform is described, different types of wavelet basis functions have been introduced among which the Daubechies wavelets (Daubechies, 1988) are the most used ones since they are compactly supported and differentiable. A compactly supported wavelet function is defined only on an interval in time domain, which is an important property to have a good time localization.

Donoho and Johnstone (1994) showed that simple thresholding in an appropriate basis can be a nearly optimal non-linear estimator of a signal contaminated with white noise. Moreover They showed that a universal threshold can be computed by the length of the signal and  $\sigma$ , which is the variance of the noise content in the signal (Donoho and Johnstone, 1995). Different kind of basis and threshold choosing

mechanizms have been developed since then.

Hong et al. (1998) introduced a new agorithm called Wavelet-Kalman Filter which is a special kind of the Kalman Filter that decomposes a signal to its wavelet coefficients. Wavelet-Kalman Filter decomposes the signal to its wavelet coefficients in real-time while filtering the signal with Kalman Filter. With this algorithm the two different areas of signal estimation methods meet in one algorithm. They also proved that this new algorithm gives more accurate estimations than the standard Kalman Filter (Hong et al., 1998).

In the second chapter of this study the Kalman Filter will be explained in detail and it's relation to the least squares method will be shown in detail. While the third chapter will explain the wavelet based multiresolution analysis and wavelet shrinkage, the forth chapter will give the Wavelet-Kalman filter derivation and implementation in detail. Also Monte-Carlo simulation results of the implementation will be listed in this chapter. The fifth chapter will show the application of the Wavelet-Kalman Filter to well logs and seismic waves and comparisons with existing filtering methods. Finally, the sixth chapter will list the conclusions of the study and potential future research areas.

The main objective of the thesis is to study if the method developed by (Hong et al., 1998) can be used in denoising of some signals related to geosciences. The algorithm introduced by Hong et al. (1998) will be implemented. Throughout the thesis the method introduced by (Hong et al., 1998) will be developed further to include other wavelet functions and to work with levels higher than two. Moreover, wavelet based thresholding (wavelet shrinkage) will be integrated into the Wavelet-Kalman Filter to introduce even more smoothing of the signal. The algorithm will be implemented in Java programming language and will be bundled in a software with simple to use graphical user interface.

## CHAPTER 2

### KALMAN FILTERING

#### 2.1 Stochastic Processes

Suppose a surveyor is holding a GPS receiver and measuring the coordinates of his position in five minutes intervals. Because of, among others, the GPS satellites ephemeris errors and other environmental conditions along the signal path, he will measure different coordinates of his position in five minute intervals. Which is the exact position of the surveyor? The answer is not straightforward. The measurements are simply random variables. The process of measuring the position of the surveyor in five minutes time, standing at the same position, can be identified as a stochastic process or time series.

##### 2.1.1 One Dimensional Random Processes

###### 2.1.1.1 Definition

A signal  $f(t)$  whose values can be determined for any time instance  $t$  of interest is called a deterministic signal. The exact value of the signal can be predicted by the associated formulation of  $f(t)$  (Brown and Hwang, 1991). For example for a deterministic signal

$$f(t) = 10\sin 2\pi t \quad (2.1)$$

the exact value of  $f(t)$  for any time instance  $t$  can be predicted exactly. However, suppose that  $f(t)$  is formulated as :

$$f(t) = 10\sin(2\pi t + \theta) \quad (2.2)$$

where  $\theta$  is an independent random variable with known distribution. Then the value of the signal  $f(t)$  can not be predicted in deterministic sense and it will be a random variable. A one dimensional process  $f(t)$ , whose value is a random variable for any time instance  $t$ , is called one dimensional random process (Therrien, 1992). If the process is defined on discrete time intervals then it is called discrete one dimensional random process and represented as  $f[k]$  or  $f_k$ , where  $k$  is the discrete time interval. In this study one dimensional random processes are referred as random processes for short.

### 2.1.1.2 Expected Value and Variance

The expected value for a discrete one dimensional stochastic process is defined as (Grival and Andrews, 2001):

$$E\{\mathbf{x}\} = \sum_{k=0}^n \mathbf{x}_k p(\mathbf{x}_k) \quad (2.3)$$

Where  $E$  means the expected value and  $p$  is the probability density function of  $\mathbf{x}$ . If the occurrence of random variables are of equal probability, then  $E\{\mathbf{x}\}$  is simply the average of the series. The mean is also called the first statistical moment of  $\mathbf{x}$ .

The variance of a stochastic process is defined as:

$$\sigma^2 = \sum_{k=0}^n (\mathbf{x}_k - E\{\mathbf{x}\})^2 p(\mathbf{x}_k) \quad (2.4)$$

Where  $\sigma^2$  is called the variance (square of standard deviation from mean) and  $p$  is the probability function of  $\mathbf{x}$ . If the occurrence of random variables are of equal probability, then variance can be written in the form of

$$\sigma^2 = \frac{1}{n} \sum_{k=0}^n (\mathbf{x}_k - E\{\mathbf{x}\})^2 \quad (2.5)$$

The variance in this case is called the average mean squared deviation from mean. The variance is a measure of clustering property of a random variable around its

mean. The variance is smaller, the more strong clustering property there is (Grival and Andrews, 2001).

### 2.1.1.3 Autocovariance, Autocorrelation and Crosscorrelation Functions

The autocovariance function of a one dimensional random process  $\mathbf{x}(t)$  is defined as:

$$\mathbf{R}_{cov(\mathbf{x})}(i, j) = \mathbf{E}\{[\mathbf{x}(i) - \mathbf{m}_{\mathbf{x}}(i)][\mathbf{x}(j) - \mathbf{m}_{\mathbf{x}}(j)]\} \quad (2.6)$$

where  $i$  and  $j$  are two arbitrary time samples,  $\mathbf{m}_{\mathbf{x}}$  is the mean (expected value) of the process (Brown and Hwang, 1991). A special case of the autocovariance function for zero mean random processes is called autocorrelation function. Autocorrelation function is defined as :

$$\mathbf{R}_{\mathbf{x}}(i, j) = \mathbf{E}\{x(i)x(j)\} \quad (2.7)$$

Autocorrelation function shows how the random process  $\mathbf{x}(t)$  is correlated with itself at two different time samples. If the random process is stationary (described in Sub-section 2.1.1.4), then autocorrelation function depends only on the time difference  $\tau$ .

$$\mathbf{R}_{\mathbf{x}}(\tau) = \mathbf{E}\{\mathbf{x}(t)\mathbf{x}(t + \tau)\} \quad (2.8)$$

where  $i = t$  and  $j = t + \tau$  substituted into Equation 2.7 (Brown and Hwang, 1991). In the case of a discrete random process  $x[k]$ , the autocorrelation and autocovariance functions can be represented as matrices.

$$\mathbf{R}_{cov(\mathbf{x})} = \begin{bmatrix} E\{[x_1 - m_1]^2\} & \dots & E\{[x_1 - m_1][x_k - m_k]\} \\ \vdots & \ddots & \vdots \\ E\{[x_k - m_k][x_1 - m_1]\} & \dots & E\{[x_k - m_k]^2\} \end{bmatrix} \quad (2.9)$$

When the expected value of the random process is zero then the autocovariance and autocorrelation matrices are equal (Brown and Hwang, 1991):

$$\mathbf{R}_{cov(x)} = \mathbf{R}_x = \begin{bmatrix} E\{x_1^2\} & \dots & E\{x_1x_k\} \\ \vdots & \ddots & \vdots \\ E\{x_kx_1\} & \dots & E\{x_k^2\} \end{bmatrix} \quad (2.10)$$

Autocorrelation function gives valuable information about the random process some of which are (Brown and Hwang, 1991):

1.  $\mathbf{R}_x(\mathbf{0})$  is the mean square value (variance) of the process  $x(t)$ .
2.  $\mathbf{R}_x$  is a symetric matrix in the scase of a stationary process  $x(t)$ .
3.  $\mathbf{R}_x(t) \leq \mathbf{R}_x(\mathbf{0})$ . In the case of stationary processes they are equal.
4. If  $x(t)$  is a periodic signal with period  $T$ , then  $\mathbf{R}_x(t)$  is also periodic with the same period.
5. If  $x(t)$  does not contain any periodic components, then  $\lim_{t \rightarrow \infty} \mathbf{R}_x(t) = 0$ . This implies zero mean for the process  $x(t)$ .

Crosscorrelation of two random processes  $x(t)$  and  $y(t)$  is defined as (Brown and Hwang, 1991):

$$\mathbf{R}_{xy}(i, j) = E\{x(i)y(j)\} \quad (2.11)$$

As in autocorrelation, if the random processes are stationary then the crosscorrelation function depends only on the time difference  $\tau$ .

$$\mathbf{R}_{xy}(\tau) = E\{x(t)y(t + \tau)\} \quad (2.12)$$

where  $i = t$  and  $j = t + \tau$ . The crosscorrelation of two random processes gives information about how the two random processes correlate with each other. The subscripts in  $R_{xy}$  is important since

$$\mathbf{R}_{xy}(\tau) = \mathbf{R}_{yx}(-\tau) \quad (2.13)$$

#### 2.1.1.4 Stationarity and Ergodicity

A random process  $x(t)$  is called strict-sense stationary if all of its statistics (mean, variance, etc..) are independent of time. At which time you observe the process the mean, and variance are the same (Grival and Andrews, 2001). Wide-sense stationary processes are random processes whose mean is constant and covariance is dependent only on the time tifference of the observation as in Equation 2.8.

A random process is called ergodic if it's statistics (mean, variance, etc..) are the same as the statistics of any ensambled processes.(Grival and Andrews, 2001). Thus, any sample of the random process contains all statistical variations of it (Brown and Hwang, 1991).

#### 2.1.1.5 Power Spectral Density Function

The autocorrelation function  $R_x(\tau)$  defined in Equation 2.8 gives important clues about the frequency content of the random process (Brown and Hwang, 1991). If the autocorrelation function  $R_x(\tau)$  decreases rapidly with  $\tau$ , then it means that the random process varies rapidly with time. On the other hand, if the random process changes slowly with time, then  $R_x(\tau)$  decreases slowly with  $\tau$  (Brown and Hwang, 1991). For stationary signals, the Wiener-Khinchine relation is defined as :

$$S_x(j\omega) = \mathfrak{F}[R_x(\tau)] = \int_{-\infty}^{\infty} R_x(\tau)e^{-j\omega\tau} d\tau \quad (2.14)$$

where  $S_x(j\omega)$  is the power spectral density function of the random process  $x(t)$ ,  $\mathfrak{F}[R_x(\tau)]$  is the Fourier Transform (define in Equation 3.1) of the autocorrelation function of  $x(t)$ . Power spectral density function and the autocorrelation function contains the basic information about the process. They can be converted to each other by means of the Fourier Transform.  $R_x(0)$  is the power (mean square value) of the random process and has the relation with power spectral density function as (Brown and Hwang, 1991):

$$R_x(0) = E\{x^2(t)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_x(jw)dw \quad (2.15)$$

where  $E\{x^2(t)\}$  is the expected value for the power of the random process  $x(t)$ . Most

of the time it is called mean square value of the random process (Brown and Hwang, 1991).

## 2.2 Some of Special Random Processes

### 2.2.1 White noise

White noise is a random process whose power spectral density, as the Fourier Transform of the autocorrelation function, is constant (Gelb, 1974).

$$S_{wn}(j\omega) = \mathfrak{F}[R_{wn}(\tau)] = A \quad (2.16)$$

where  $A$  is a constant value,  $S_{wn}(j\omega)$  is the power spectral density function of the white noise,  $R_{wn}(\tau)$  is the autocorrelation function. Having a constant power spectral density function, white noise covers all the frequency spectrum. The term *white* comes from the white light, which is the sum of all frequencies from visible light spectrum (Brown and Hwang, 1991). The pure white noise has infinite power :

$$P_{wn} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{wn}(j\omega) d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} A d\omega \quad (2.17)$$

where  $P_{wn}$  is the power of pure white noise. Thus, it is not physically plausible (Brown and Hwang, 1991). To achieve a finite variance white noise, *bandlimited white noise* is introduced. It has a band limited power spectral density function defined as:

$$S_{bwn}(j\omega) = \begin{cases} A, & |\omega| \leq 2\pi W \\ 0, & |\omega| > 2\pi W \end{cases} \quad (2.18)$$

where  $W$  is the physical frequency bandwidth. The corresponding autocorrelation function is (Brown and Hwang, 1991) :

$$R_{bwn}(\tau) = 2WA \left( \frac{\sin(2\pi W\tau)}{2\pi W\tau} \right) \quad (2.19)$$

It is obvious from the equation that the power of band limited white noise is finite

and has the value  $R_{bwn}(0) = 2WA$ . Discrete white noise is usually referred as zero-mean uncorrelated random variable with fixed variance (Brown and Hwang, 1991).

### 2.2.2 Gauss-Markov Processes

A random process  $x(t)$  is called Gaussian (normal) if its probability density function  $p$  is a Gaussian function defined below (Grival and Andrews, 2001). The Gaussian probability density function for single valued process  $x$  is :

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-E\{x\})^2}{2\sigma^2}} \quad (2.20)$$

where  $\sigma$  is the standard deviation of  $x$ . For a vector valued  $x$  the Gaussian probability density function is of the form:

$$p(x) = \frac{1}{\sqrt{(2\pi)^n |C_x|}} e^{\frac{1}{2}(\mathbf{x}-E\{\mathbf{x}\})^T C_x^{-1} (\mathbf{x}-E\{\mathbf{x}\})} \quad (2.21)$$

Where  $|C_x|$  is the determinant of the covariance matrix  $C_x$ ,  $n$  is the dimension (or the number of variates) in the column vector  $x$ .

A stationary Gaussian process  $x(t)$  whose autocorrelation function is exponential is called Gauss-Markov process. The autocorrelation function for Gauss-Markov process is defined as (Brown and Hwang, 1991) :

$$R_x(\tau) = \sigma^2 e^{-\beta|\tau|} \quad (2.22)$$

where  $\sigma^2$  is the mean-square value of the process and  $1/\beta$  is the time constant. The process is nondeterministic, uncorrelated and looks like noise (Brown and Hwang, 1991). As time increases the autocorrelation function  $R_x$  approaches to zero, thus the process has zero mean. Brown and Hwang (1991) states that Gauss-Markov process is an important process because, it seems to fit a large number of physical processes with good accuracy and has a relatively simple mathematical formulation.

### 2.2.3 ARMA Processes

A discrete random process which is modelled by :

$$y[n + 1] = \sum_{k=0}^K a_k y[n - k] + \sum_{m=0}^M b_m w[n - m] \quad (2.23)$$

where  $y$  is the random process, which is modeled by a difference equation of its previous values,  $w$  is a white noise sequence with zero mean and unit variance, is called an ARMA (Autoregressive Moving Averages) model of the random process  $y$  (Brown and Hwang, 1991). Generally, ARMA models are known as shaping filters of white noise to desired colored noise. Colored noise is a model for correlated random processes. Both sides of the Equation 2.23 can be z-transformed and arranged as :

$$A(z)Y(z) = B(z)W(z) \quad (2.24)$$

where  $Y(z)$  and  $W(z)$  are the z-transforms of the process and white noise respectively,  $A(z)$  and  $B(z)$  are the coefficients of the z-transforms of the process and white noise. The z-transform or complex spectral density function is defined as (Therrien, 1992):

$$H(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k} \quad (2.25)$$

where  $z = e^{j\omega}$ ,  $H(z)$  is the z-transform of the process  $x[k]$ . The resulting power spectral density function for the ARMA process is then (Therrien, 1992)

$$S_y(j\omega) = \frac{|B(z)|^2}{|A(z)|^2} \quad (2.26)$$

where  $S_y$  is the desired power spectral density function for the model. The system is said to be stationary if the roots of  $A(z)$  lies inside the unit circle in the z-plane, otherwise it is nonstationary (Brown and Hwang, 1991).

#### 2.2.4 Wiener or Brownian Motion Process

A Random process which is an integral of white noise is called a Wiener or Brownian motion process (Brown and Hwang, 1991).

$$x(t) = \int_0^t F(u)du \quad (2.27)$$

where  $F(u)$  is a white noise process. The expected value of the process  $x(t)$  is  $E\{x(t)\} = 0$  and its variance is  $E\{x^2(t)\} = t$  (Brown and Hwang, 1991). The autocorrelation function of  $x(t)$  is defined as:

$$\mathbf{R}_x(t_1, t_2) = \begin{cases} t_2, & t_1 \geq t_2 \\ t_1, & t_1 < t_2 \end{cases} \quad (2.28)$$

where  $t_1$  and  $t_2$  are two time samples of the brownian motion process. The ARMA model for the brownian motion can be defined as :

$$x[t + 1] = x[t] + w[t] \quad (2.29)$$

where  $t$  is the time instance and  $w[t]$  is the realization of the white noise at time instance  $t$ .

## 2.3 Recursive Parameter Estimation

Recursive parameter estimation is based on the Gauss-Markoff model (not to be confused with Gauss-Markov process), which leads to the best unbiased estimation of the parameters by using the method of least squares.

### 2.3.1 Gauss-Markoff Model

#### 2.3.1.1 Definition

A Gauss-Markof model is defined as:

$$E\{\mathbf{y}\} = \mathbf{X}\boldsymbol{\beta}, D(\mathbf{y}) = \sigma^2 \mathbf{P}^{-1} \quad (2.30)$$

where  $\mathbf{y}$  is the observation (measurements) vector,  $D(\mathbf{y})$  is the variance covariance matrix of the observations.  $\mathbf{P}$  is a positive definite weight matrix.  $\mathbf{X}$  is the linear

(or linearized) coefficient matrix, which contains the partial derivatives of the observations with respect to parameters and  $\beta$  is the unknown, fixed system parameters (Koch, 1999). According to the model expected values of the measurements ( $E\{\mathbf{y}\}$ ) are a linear function of the system parameters  $\beta$ . This linear system model is sometimes a linearized model by using physical or mathematical laws (Koch, 1999).

### 2.3.2 Derivation

Let  $s$  be an estimator of the expected values of measurements ( $E(y) = s(\beta)$ ), and observations have a positive definite covariance function  $\Sigma$ . If the estimator  $s$  minimizes the quadratic form

$$\mathbf{f}(\beta) = [\mathbf{y} - \mathbf{s}(\beta)]^T \Sigma^{-1} [\mathbf{y} - \mathbf{s}(\beta)] \quad (2.31)$$

where  $\mathbf{f}(\beta)$  is the quadratic form to be minimized and superscript  $T$  is the transpose sign, then  $s$  is called the least squares estimator. Least squares estimators minimize the sum of squares of the residuals which are the corrections to the observations. This can be treated as finding the best fit for estimation parameters geometrically (Koch, 1999).

In the case of Gauss-Markoff model the  $\Sigma$  is known as  $\Sigma = \sigma^2 \mathbf{P}^{-1}$  where  $\sigma$  is the variance of unit weight. Taking  $\mathbf{P} = \mathbf{I}$ , which means the observations have the same quality, the quadratic form becomes (Koch, 1999):

$$\mathbf{f}(\beta) = [\mathbf{y} - \mathbf{X}\beta]^T \frac{1}{\sigma^2} \mathbf{P} [\mathbf{y} - \mathbf{X}\beta] \quad (2.32)$$

which has an extreme value by

$$\frac{\partial \mathbf{f}}{\partial \beta} = \frac{\partial (\frac{1}{\sigma^2} (\mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{X}\beta + \beta^T \mathbf{X}^T \mathbf{X}\beta))}{\partial \beta} = 0 \quad (2.33)$$

The roots are the estimated parameters  $\hat{\beta}$  which can be calculated by

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (2.34)$$

In the case of non unit weight variance  $P \neq I$ , the estimated parameters become

$$\hat{\beta} = (X^T P X)^{-1} X^T P y \quad (2.35)$$

and

$$D(\hat{\beta}) = \sigma^2 (X^T P X)^{-1} \quad (2.36)$$

where  $D(\hat{\beta})$  is the covariance matrix of the estimated parameters. The resulting estimate of  $E\{y\}$  is given by

$$E\{y\} = \hat{y} = X \hat{\beta} \quad (2.37)$$

The estimated residual vector is given by  $\hat{e} = \hat{y} - y$ . The solution of the Gauss-Markoff model can be treated as a noise reduction method if the residuals  $\hat{e}$  are treated as the unknown white noise component in the measurements  $y$ . Then the variance of the residuals is the variance of the white noise removed.

$$D(\hat{e}) = \frac{\hat{e}^T \hat{e}}{n - u} \quad (2.38)$$

where  $n$  is the number of measurements,  $u$  is the number of parameters and  $D(\hat{e})$  is the estimated variance of the residuals.

### 2.3.3 Recursive Parameter Estimation

Gauss-Markoff model assumes that all the observations are available prior to the parameter estimation. Suppose that the measurements can be separated to uncorrelated vectors  $y_m$  and  $y_{m-1}$  measured at different times. Then the model equations can be written as (Koch, 1999):

$$X = \begin{bmatrix} X_{m-1} \\ X_m \end{bmatrix}, y = \begin{bmatrix} y_{m-1} \\ y_m \end{bmatrix} \text{ and } D(y) = \sigma^2 \begin{bmatrix} P_{m-1}^{-1} & 0 \\ 0 & P_m^{-1} \end{bmatrix} \quad (2.39)$$

By using Equation 2.35, the estimation for the parameters  $\beta$  can be obtained by (Koch, 1999):

$$\hat{\beta} = (X_{m-1}^T P_{m-1} X_{m-1} + X_m^T P_m X_m)^{-1} (X_{m-1}^T P_{m-1} y_{m-1} + X_m^T P_m y_m) \quad (2.40)$$

If  $\hat{\beta}_{m-1}$  denotes the estimators of the parameters based on the measurements at time  $m - 1$  and  $D(\hat{\beta}) = \sigma^2 \Sigma_{m-1}$  is their covariance matrix, then the estimators of the parameters at time  $m$  can be written as:

$$\hat{\beta}_m = (\Sigma_{m-1}^{-1} + X_m^T P_m X_m)^{-1} (\Sigma_{m-1}^{-1} \hat{\beta}_{m-1} + X_m^T P_m y_m) \quad (2.41)$$

with  $D(\hat{\beta}_m) = \sigma^2 \Sigma_m$  (Koch, 1999). Using Equation 2.36  $\Sigma_m$  can be written as:

$$\Sigma_m = \Sigma_{m-1} - F_m X_m \Sigma_{m-1} \quad (2.42)$$

where

$$F_m = \Sigma_{m-1} X_m^T (P_m^{-1} + X_m \Sigma_{m-1} X_m^T)^{-1} \quad (2.43)$$

Substituting  $F_m$  into Equation 2.41 yields (Koch, 1999):

$$\hat{\beta}_m = \hat{\beta}_{m-1} + F_m (y_m - X_m \hat{\beta}_{m-1}) \quad (2.44)$$

This defines a recursive relationship between the estimators of the parameters at time  $m$  with the estimators of the parameters at time  $m - 1$ . The recursive algorithm starts with normal Gauss-Markoff model to obtain the estimators of the parameters  $\hat{\beta}_1$  by using Equation 2.35 or 2.34. If the parameters  $\beta$  are assumed to be the state variables of a dynamic system which are linearly transformed from time  $t_{m-1}$  to  $t_m$  by addition of random disturbances (white noise), then The Equation 2.44 turns out to be a Kalman-Bucy filter (Koch, 1999).

## 2.4 Kalman Filter

Kalman filter is an optimal recursive estimation algorithm introduced by Kalman (1960). The word optimal is a tricky keyword since it depends on the chosen criteria. However, Maybeck (1979) states that Kalman Filter is optimal in virtually any criterion that makes sense under the assumptions of noise characteristics and system model in its definition. Kalman Filter uses every information available for it to estimate the state of a given system. Maybeck (1979) expresses that this knowledge can be listed as :

1. Knowledge of system deterministic model or dynamics.
2. Knowledge of measurement device dynamics.
3. Statistics of the measurement noise.
4. Statistics of the system noise.
5. Uncertainty of the system model.
6. Initial conditions.
7. Control inputs to the system.

Moreover Kalman filter does not depend on previously processed data to be reprocessed when new measurements are available.

### 2.4.1 Discrete Kalman Filter

Discrete Kalman filter is a state space approach to parameter estimation. The parameters of the system are treated as states of a random process. The Discrete Kalman Filter estimates the states of the given system that minimizes the state estimate error covariance (Welch and Bishop, 2001).

#### 2.4.1.1 Derivation

Assume the system is modeled by the following linear difference equation:

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k \quad (2.45)$$

and the system is observed by measuring the outputs of the system which has a linear relationship with the system state as:

$$z_k = C_k x_k + v_k \quad (2.46)$$

where,

$x_k$  is an (nx1) process state vector at time instance  $t_k$ .

$A_k$  is (nxn) matrix that relates the state  $x_k$  to the state  $x_{k+1}$ , if the random process is a continues process it is called the state transition matrix Brown and Hwang (1991).

$w_k$  is a (nx1) vector which is assumed to be a white noise sequence with known covariance.  $w_k$  is called the process noise.

$z_k$  is (mx1) measurement vector at time instance  $t_k$ .

$C_k$  is a (mxn) matrix, which is the ideal (noiseless) linear relation between the measurements an the state vector  $x_k$ .

$v_k$  is (mx1) measurement error vector. It is assumed to be independent of process noise  $w_k$  and having a known covariance structure (Brown and Hwang, 1991).

White noise sequences  $w_k$  and  $v_k$  has the following covariance matrices:

$$E\{w_k(w_i)^T\} = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \quad (2.47)$$

$$E\{v_k(v_i)^T\} = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \quad (2.48)$$

$$E\{w_k v_i^T\} = 0 \text{ for all } k \text{ and } i \quad (2.49)$$

Assume that we have a priori estimate of the system state  $x_k$  at time  $t_k$  as  $\hat{x}_k^-$ . And also assume that the error covariance matrix associated to the prior estimate is known. Estimation error of the known prior estimate is defined as:

$$e_k^- = x_k - \hat{x}_k^- \quad (2.50)$$

and the associated process error covariance matrix is

$$\mathbf{P}_k^- = E\{e_k^- e_k^{-T}\} \quad (2.51)$$

With the assumption of the a priori estimate  $\hat{\mathbf{x}}_k^-$ , measurements  $z_k$  at time instance  $t_k$  is used to improve the quality of the prior estimate by using a blending matrix  $\mathbf{K}_k$ .

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(z_k - C_k \hat{\mathbf{x}}_k^-) \quad (2.52)$$

where  $\hat{\mathbf{x}}_k$  is the updated estimate of the state and  $\mathbf{K}_k$  is the blending factor that gives and optimal update to the state in least squares sense. The error covariance matrix for the updated estimate  $\hat{\mathbf{x}}_k$  is defined as

$$\mathbf{P}_k = E\{e_k e_k^T\} = E\{[\mathbf{x}_k - \hat{\mathbf{x}}_k][\mathbf{x}_k - \hat{\mathbf{x}}_k]^T\} \quad (2.53)$$

Substituting the Equations 2.51 and 2.52 into the equation of  $\mathbf{P}_k$  one gets

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k C_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k C_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (2.54)$$

and setting the derivative of  $\mathbf{P}_k$  with respect to the blending factor  $\mathbf{K}_k$  we get the *Kalman Gain* matrix  $\mathbf{K}_k$  that minimizes the mean square error in estimated state variables.

$$\mathbf{K}_k = \mathbf{P}_k^- C_k^T (C_k \mathbf{P}_k^- C_k^T + \mathbf{R}_k)^{-1} \quad (2.55)$$

The updated process error covariance matrix  $\mathbf{P}_k$  in the case of optimal kalman gain  $\mathbf{K}_k$  can be computed by substituting Equation 2.55 into 2.54 as

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k C_k) \mathbf{P}_k^- \quad (2.56)$$

The updated optimal estimate  $\hat{\mathbf{x}}_k$  can be easily projected ahead to find the a priori estimate at time instance  $t_{k+1}$  as

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_k \quad (2.57)$$

and the associated error covariance matrix is

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k \quad (2.58)$$

By using Equations 2.58, 2.57, 2.55 and 2.56, the recursive Kalman Filter can be written as a prediction correction style algorithm by assuming the system and measurement noise is normally distributed white noise with variance matrices  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  respectively. The graphical representation of the Algorithm 1 is given in Figure 2.1.

---

**Algorithm 1** Kalman Filter

---

- 1:  $\mathbf{P}_k^- \Leftarrow \mathbf{P}_0$
  - 2:  $\mathbf{x}_k^- \Leftarrow \mathbf{x}_0$
  - 3:  $k \Leftarrow 0$
  - 4: **while** measurement is collected **do**
  - 5:    $\mathbf{R}_k \Leftarrow$  covariance of new measurements //  $\mathbf{R}_k$  and  $\mathbf{Q}_k$  can be dependent on time.
  - 6:    $\mathbf{Q}_k \Leftarrow$  covariance of current system state variables
  - 7:    $z_k \Leftarrow$  measurements
  - 8:    $\mathbf{K}_k \Leftarrow \mathbf{P}_k^- \mathbf{C}^T (\mathbf{C} \mathbf{P}_k^- \mathbf{C}^T + \mathbf{R}_k)^{-1}$  // Find Kalman gain matrix  $\mathbf{K}_k$
  - 9:    $\mathbf{P}_k \Leftarrow (\mathbf{I} - \mathbf{K}_k \mathbf{C}) \mathbf{P}_k^-$
  - 10:    $\hat{\mathbf{x}}_k \Leftarrow \hat{\mathbf{x}}_k^- + \mathbf{K}_k (z_k - \mathbf{C} \hat{\mathbf{x}}_k^-)$  // Apply correction to the predicted state variables by taking the measurements into account
  - 11:    $\mathbf{P}_{k+1}^- \Leftarrow \mathbf{A} \mathbf{P}_k \mathbf{A}^T + \mathbf{Q}_k$  // Predict Future error covariance of the state matrix
  - 12:    $\hat{\mathbf{x}}_{k+1}^- \Leftarrow \mathbf{A} \hat{\mathbf{x}}_k$  // Predict future version of  $\hat{\mathbf{x}}_k$
  - 13:    $k \Leftarrow k + 1$
-

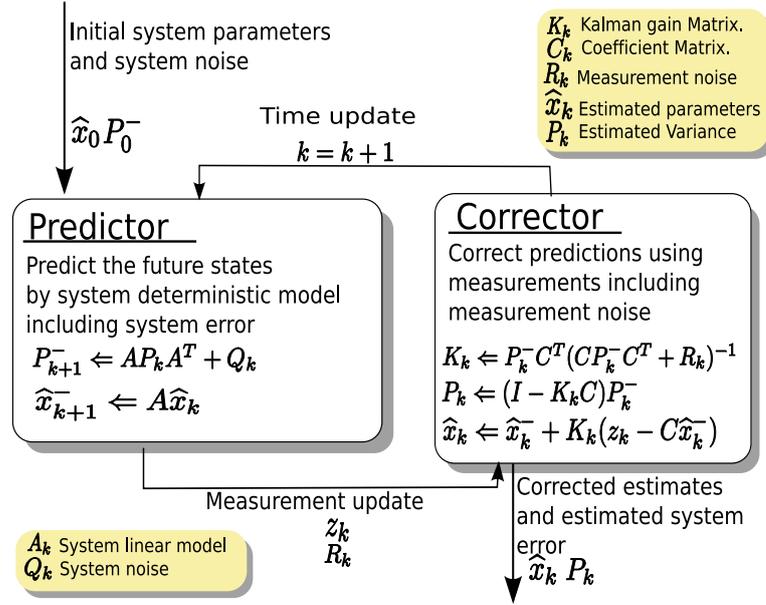


Figure 2.1: Discrete Kalman filter execution diagram

Kalman gain is the most important part of the algorithm it resembles the  $F_m$  matrix of the recursive parameter estimation in Equation 2.44.

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R_k}$$

As it is seen from the formula, it gives the filter an adaptive nature since the filter relies on measurements as the measurement variance decreases (measurement quality increases) and relies more on system prediction otherwise. Moreover, the state transition matrix  $A$  and the observation matrix  $C$  can also be dependent on time which strengthens the adaptive nature of the Kalman filter. The adaptive nature of the Kalman Filter introduces a tuning phase for the Kalman Filter. The variances  $Q$ (process noise variance) and  $R$ (measurement noise variance) may not be known exactly. That is why, the filtering algorithm is tuned by adjusting  $Q$  and  $R$  to obtain the desired process error covariance  $P_k$ . Welch and Bishop (2001) states that, since the error variance of the estimates converges very rapidly to some constant matrix, if the measurement noise variance and process noise variance do not change with time, the Kalman gain matrix  $K_k$  can be taken as a constant matrix, which reduces the computation cost dramatically.

### 2.4.2 Example: Estimating a Constant

Assume that we are measuring a constant value such as voltage where the measurement device has a known error covariance. Then the system state equation is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{w}_k \quad (2.59)$$

and the measurement equation is

$$z_k = \mathbf{x}_k + \mathbf{v}_k \quad (2.60)$$

Also assume that the true system state is 5 volts. According to the system description, the state transition matrix  $\mathbf{A}_k$  is constant and equal to 1, the measurement model matrix  $\mathbf{C}_k$  is constant and equal to 1. The measurement error is a zero mean uncorrelated sequence with fixed variance  $\mathbf{R} = 3$  and the process noise variance is unknown. The Figure 2.2 shows the measurements.

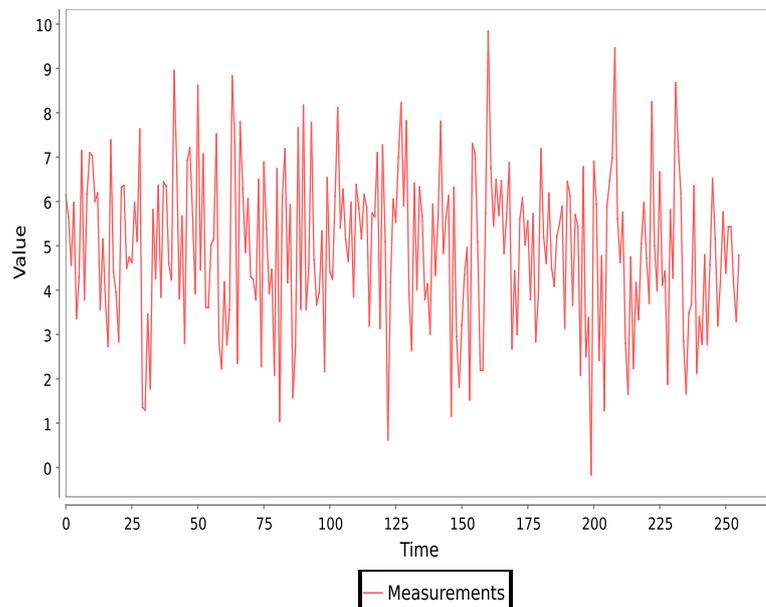
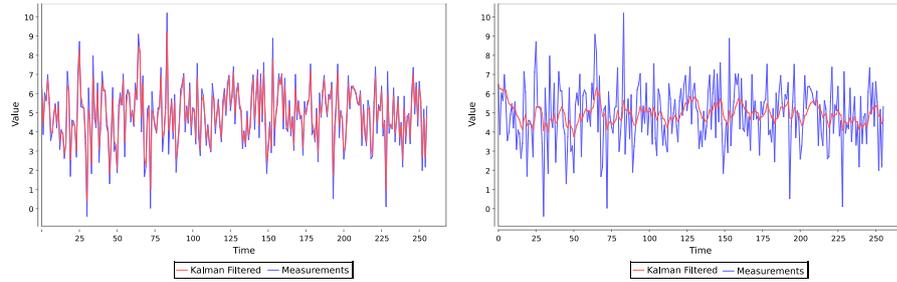
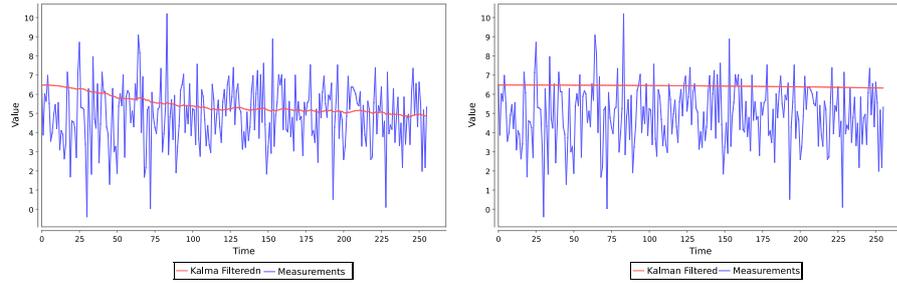


Figure 2.2: Realization of the process with constant value of 5 and zero mean measurement noise with variance of 3.

The process noise variance is unknown and yet to be determined by tuning the Kalman filter to get the desired error covariance matrix  $\mathbf{P}_k$  of Equation 2.56. This Figure 2.3 shows the effects of choosing different process variances.



(a) Kalman filter result when process noise variance set to 10,  $P_k = 2.4162$  (b) Kalman Filter result when process noise variance set to 0.1,  $P_k = 0.5$



(c) Kalman Filter result when process noise variance set to 0.001,  $P_k = 0.0543$  (d) Kalman Filter result when process noise variance set to 0.00001,  $P_k = 0.0024$

Figure 2.3: Different process variances and their effect on the results of the Kalman Filter.

The tuning phase of the Kalman filter is an offline process where different distinct Kalman filters run on a sample measured input to find the best process noise variance  $Q$  and the measurement noise variance  $R$  (Welch and Bishop, 2001). After tuning the noise variances the filter can be run online (rela-time) to estimate the state of the desired process.

## CHAPTER 3

# MULTI RESOLUTION REPRESENTATION AND THRESHOLDING

### 3.1 Fourier Transform

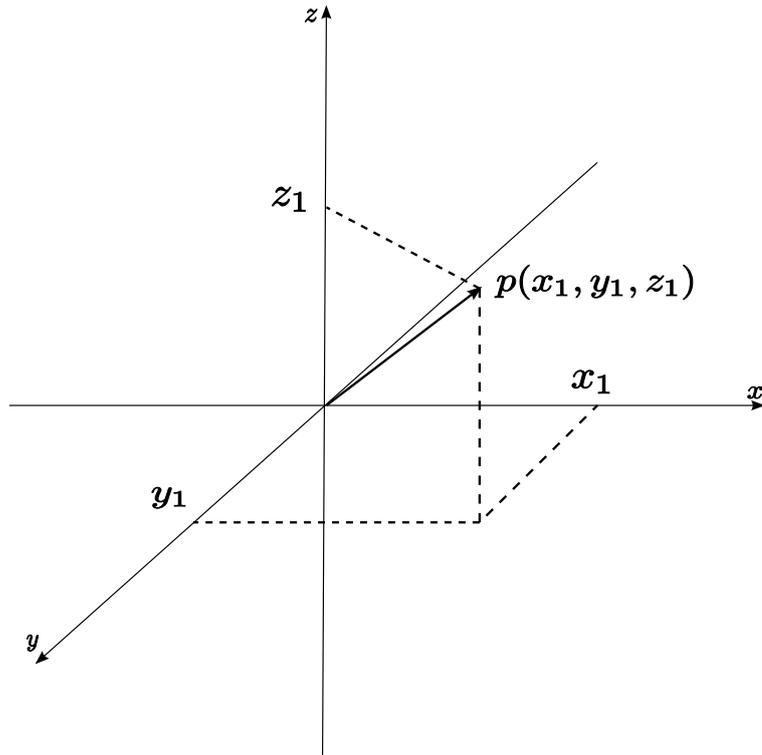
Signals can be treated as functions of time or space. For example, the seismic waves coming from an earthquake can be represented as a function of time having magnitude of the acceleration ( $cm/s^2$ ), as *magnitude* =  $f(t)$  (Simav, 1996). In some cases, the signal itself is a superposition of other signals. Fourier Transform is the most famous signal representation technique that decomposes the signal into superposition of trigonometric functions of different frequencies (Debnath and Mikusinski, 1999).

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (3.1)$$

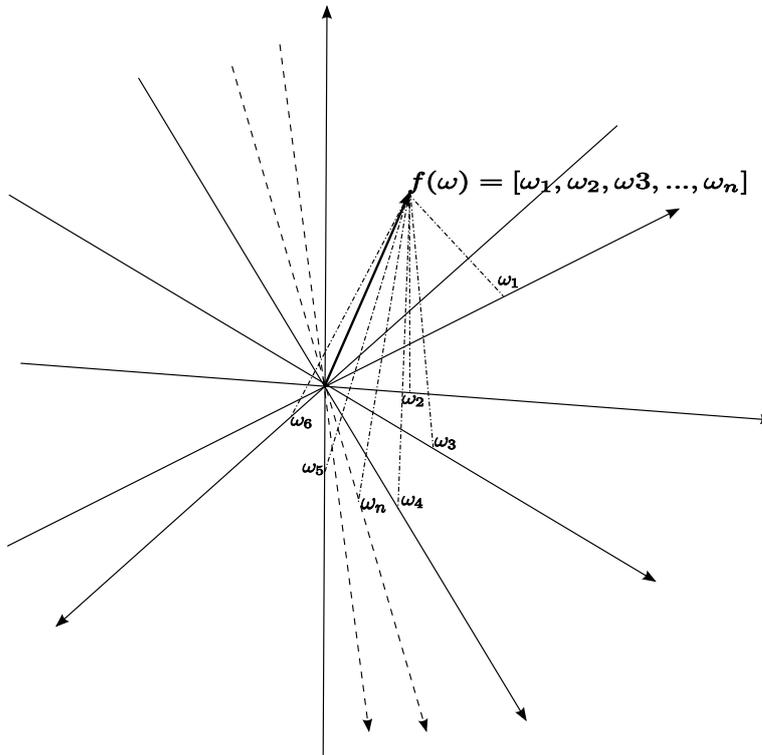
Where  $\omega$  is frequency and  $\hat{f}(\omega)$  is the signal representation in frequency domain. Fourier transform finds how much oscillations are there for a given frequency of  $\omega$  Mallat (1998).

Figure 3.1 shows a similarity between representing points in 3-D euclidian space and representations of functions in frequency domain. Debnath and Mikusinski (1999) states that every periodic finite energy function can be represented in frequency domain by using the Fourier Transform. The Fourier Transform can be interpreted as finding the frequency distribution of the given function in the frequency domain.

The Inverse Fourier Transform converts the frequency coefficients into the time domain. The inverse transform equation is (Debnath and Mikusinski, 1999):



(a) Representation of a point in 3-D cartesian coordinate system with coordinates  $(x_1, y_1, z_1)$



(b) Fourier Representation of a function  $f(t)$  in frequency domain with coefficients  $\omega_1, \omega_2, \omega_3, \dots, \omega_n$  of frequencies

Figure 3.1: Representation of a point in 3-D cartesian coordinate system (a) and representation of a function in frequency domain (b)

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(w) e^{iwt} dw \quad (3.2)$$

Fourier Transform is used in diverse application areas of science and engineering ranging from quantum physics to solution of partial differential equations, from image processing to data compression. Simple noise removal or smoothing filters can be implemented by setting the frequency coefficients of the high frequency components to zero (Mallat, 1998). This simple algorithm is called low-pass filtering since it allows only the low frequency components of the signal in Fourier domain. However, this kind of algorithms assume that the noise content of the signal is stationary (noise characteristics do not change with time) and noise in frequency domain is represented by high frequency components. The Fourier transform does not have good time/space localization (knowledge of which frequencies appear at which time), because the building blocks of the transform are trigonometric functions which do not have finite (compact) support (Mallat, 1998). A function with finite support is defined on a finite interval and zero outside that interval.

### 3.1.1 Windowed Fourier Transform

Although the Fourier Transform is a very useful method to represent the signal content in frequency domain, it cannot reveal the frequency localization of the signal. Suppose that only a subset of the signal content is contaminated with noise, the fourier transform will only show the frequency content of the whole signal. To overcome the frequency localization problem of the fourier transform, Gabor (1946) showed that windowed fourier transform can be implemented by convolving the signal with the gabor windows  $g_{u,\xi}(t)$ . Gabor windows are modulated and translated versions of simple symmetric window functions (Mallat, 1998).

Gabor window is defined as :

$$g_{u,\xi}(t) = e^{i\xi t} g(t - u) \quad (3.3)$$

Where  $u$  is the time translation and  $\xi$  is the frequency modulation parameters and  $t$  is time.  $g(t)$  is a windowing function. Blackman, Hamming Gaussian and Han-

ning window functions are mostly used window functions. The Windowed Fourier Transform (Short time Fourier Transform) can be formulated as (Mallat, 1998):

$$Sf(u, \xi) = \int_{-\infty}^{\infty} f(t)g(t - u)e^{-i\xi t} dt \quad (3.4)$$

where  $u$  is the locallization and  $\xi$  is the frequency modulation parameters. Note that  $Sf(u, \xi)$  in Equation 3.4 is a two dimensional signal rather than one dimensional result of standard Fourier Transform.  $u$  is the localization dimension, and  $\xi$  is the frequency parameter. The reconstruction formula for the Windowed fourier transform can be written as (Mallat, 1998):

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Sf(u, \xi)g(t - u)e^{i\xi t} d\xi du \quad (3.5)$$

Although Windowed Fourier Transform gives the frequency localization of the signal, It is a redundant transform and its time frequency resolution is constant. The time frequency resolution can be arranged by scaling the windowing function  $g(t)$ (Mallat, 1998).

### 3.2 Wavelets and Continuous Wavelet Transform

Wavelets are simple wave-like functions whose scaled and translated versions form an orthonormal basis for finite energy signals (Mallat, 1998). Although the theoretical basis for wavelets and wavelet transform was given by (Mallat, 1989), (Haar, 1910) had used the term *wavelet* in his thesis for simple functions of finite support. In 1960s through 1980s Guido Weiss and Ronald Coifman and (Grossman and Morlet, 1984) used scaling and dilating simple functions (atoms or wavelets) to analyse signals. After Mallat's theoretical foundation (Mallat, 1989) many researchers and scientists studied different families of wavelet functions. (Daubechies, 1988) has given the formulations of widely used finite support and p-times differentible wavelet functions known as Daubechies family of wavelets. Recently, wavelet like functions for multiresolution analysis of signals lead to orther kinds of wavelets like geometric wavelets, which tries to find geometrical wavelets forming a geometric subdivision

of an image to remove noise (Alani et al., 2007) and spherical wavelets, which tries to implement wavelets on a spherical surface (McEwen et al., 2007).

Formally, a wavelet is a function  $\psi(t)$  whose average is zero,

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (3.6)$$

Its dilations and translations form an orthonormal base for finite energy signals ( $\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty$ ) (Mallat, 1998). Figure 3.2 shows Daubechies wavelet and its scaling function.

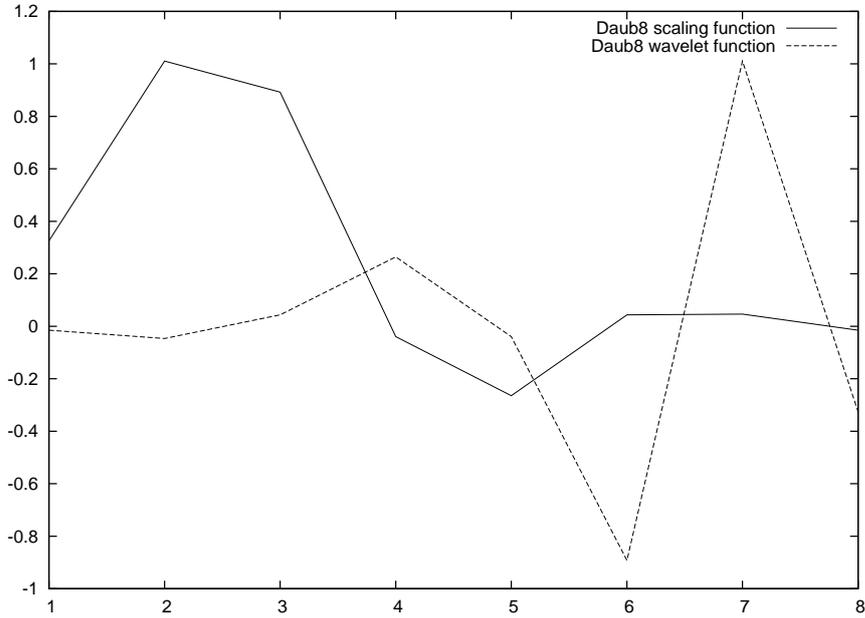


Figure 3.2: Daubechies 8 wavelet and its associated scaling function.

scaled and dilated version of the mother wavelet can be written as:

$$\psi_{u,s} = \frac{1}{\sqrt{|s|}} \psi\left(\frac{t-u}{s}\right) \quad (3.7)$$

Where  $u$  is the translation and  $s$  is the scale factor. Then any finite energy signal can be represented as (Mallat, 1998):

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(u,s) \psi_{u,s}(t) dt \frac{ds}{s^2} \quad (3.8)$$

Where  $C_\psi$  is the admissibility constant defined as :

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\rho)|^2}{|\rho|} d\rho \quad (3.9)$$

where  $\hat{\psi}(\rho)$  is the Fourier transform of  $\psi(t)$ . The success of the inverse transform depends on the condition  $0 < C_\psi < \infty$ . The admissibility condition implies that  $\hat{\psi}(0) = \int_{-\infty}^{\infty} \psi(t) dt = 0$ , which is identical to Equation 3.6 (Mallat, 1998).  $c(u, s)$  is the wavelet transform of  $f(t)$  which defines the coefficients of the wavelet functions at different scales and times.

$$c(u, s) = \int_{-\infty}^{\infty} f(t) \psi_{u,s} dt \quad (3.10)$$

Although Continuous Wavelet Transform gives a multiresolution representation of the signal at different scales and times, it is not computationally efficient and redundant like Windowed Fourier Transform (Mallat, 1998). Discrete wavelet transform can be computed efficiently by using a Dyadic Wavelet Transform (Mallat, 1989). Instead of using continuous scales, one can use dyadic scales (scales of power of two coming from Nyquist-Shannon Sampling theorem), and use orthogonal wavelets to eliminate redundant coefficients.

### 3.2.1 Multiresolution Representation

A Multiresolution representation of a signal is a sparse representation by its wavelet and scaling coefficients, forming a resolution pyramid of the signal. At every scale (pyramid level) a coefficient of the scaling function ( $\phi(t)$ ) represent the low pass filtered version of the original signal, whereas coefficients of the wavelet function ( $\psi(t)$ ) represent the high pass filtered version of the original signal at that scale. The top of the pyramid gives the coarse representation of the signal (mostly the average of the signal) whereas the bottom of the pyramid represents the fine details. The direct sum of the coefficients of the pyramid gives the perfect reconstruction of the original signal (Mallat, 1998). The frequency resolution increases in the direction of the top of the pyramid, however the time resolution decreases to the top of the pyramid.

The  $J$  level multiresolution representation of a signal  $f(t)$  can be written as:

$$f(t) = \sum_{u=-\infty}^{\infty} a_{u,J} \phi_{u,J}(t) + \sum_{u=-\infty}^{\infty} \sum_{s=J}^0 w_{u,s} \psi_{u,s}(t) \quad (3.11)$$

Where  $a_{u,J}$  is the scaling function coefficients at scale  $J$ ,  $w_{u,s}$  is the wavelet function coefficients at scale  $s$  and time  $u$ , The scaling function is defined as :

$$\phi_{u,J} = \frac{1}{\sqrt{2^J}} \phi\left(\frac{t - 2^J u}{2^J}\right) \quad (3.12)$$

and the wavelet function is defined as :

$$\psi_{u,s} = \frac{1}{\sqrt{2^s}} \psi\left(\frac{t - 2^s u}{2^s}\right) \quad (3.13)$$

Scaling function  $\phi(t)$  itself can be represented as a sum of wavelet functions:

$$\phi(t) = \sum c_{u,s} \psi_{u,s}(t) \quad (3.14)$$

That means, at the coarsest scale it represents the remaining wavelet coefficients of the signal as well. The admissibility condition for scaling function is  $\int \phi(t) dt = \hat{\phi}(0) = 1$  ( $\hat{\phi}$  is the Fourier Transform of  $\phi$ ). This also means that the scaling function is a low pass filter (Mallat, 1998).

The wavelet transform can be calculated by a series of convolutions and down-sampling operations performed on a discrete signal  $f[t]$  (Mallat, 1989). This series can be setup easily as a conjugate mirror filter or filter banks. Wavelet based filterbank splits the signal into two components one of which represents the high pass content and the other represents the low pass content. The low pass content is computed by convolving the signal with the associated scaling function  $\phi[t]$ . The high pass content is computed by convolving the signal by associated wavelet function  $\psi[t]$ . Then the low frequency part is subsampled again to find the high and low frequency content of the signal at the next coarse level (Mallat, 1998). The resulting signal representation is the scaling function and wavelet function coefficients in a pyramidwise fashion. The graphical representation of the algorithm is given in Figure 3.3.

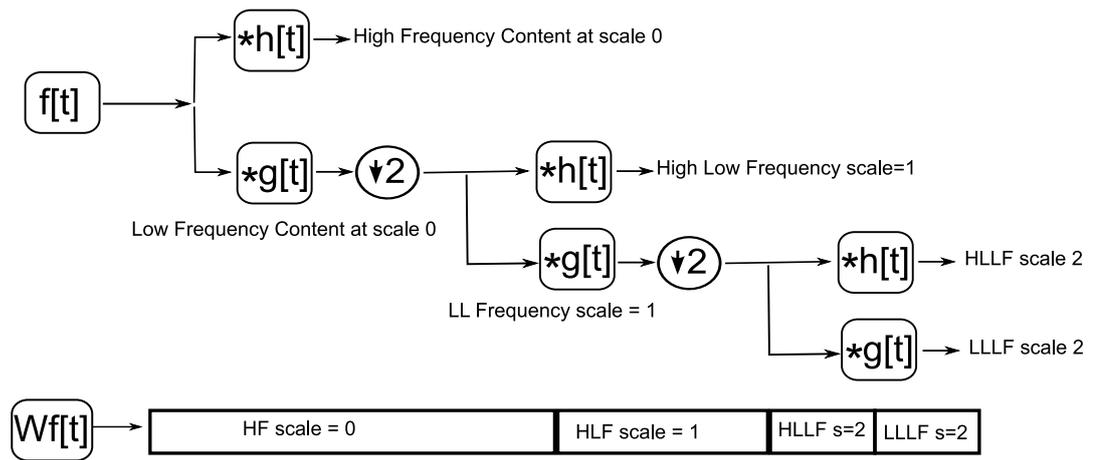


Figure 3.3: Graphical representation of a filter bank.  $Wf[t]$  represents the wavelet transform of  $f[t]$ .  $g[t]$  is the scaling function,  $h[t]$  is the wavelet function.  $*$  represents the convolution operator.

Inverse wavelet transform can be computed by inverting the filter bank. Summing up and upsampling the lower scales to find the finest scale representation of the signal itself. The graphical representation of the inverse filter bank is shown in Figure 3.4.

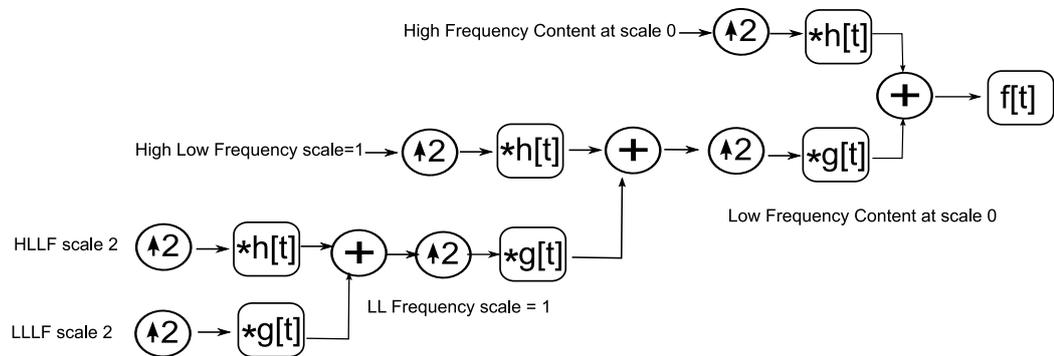


Figure 3.4: Graphical representation of an inverse filter bank.  $g[t]$  is the scaling function,  $h[t]$  is the wavelet function.  $*$  represents the convolution operator.

Multiresolution representation gives a sparse representation, where most of the coefficients are zero or close to zero (Mallat, 1998). Sparse representation of the signal is a very important property for signal compression and implementing fast signal processing algorithms. The sparsity property comes from the orthogonal property of the wavelets used. It removes the redundant information from the signal content. The choice of the wavelet function according to the signal characteristics is very im-

portant since redundancy depends on wavelet function coefficients itself. The Figure 3.5 shows a simple sine wave function and its wavelet coefficients.

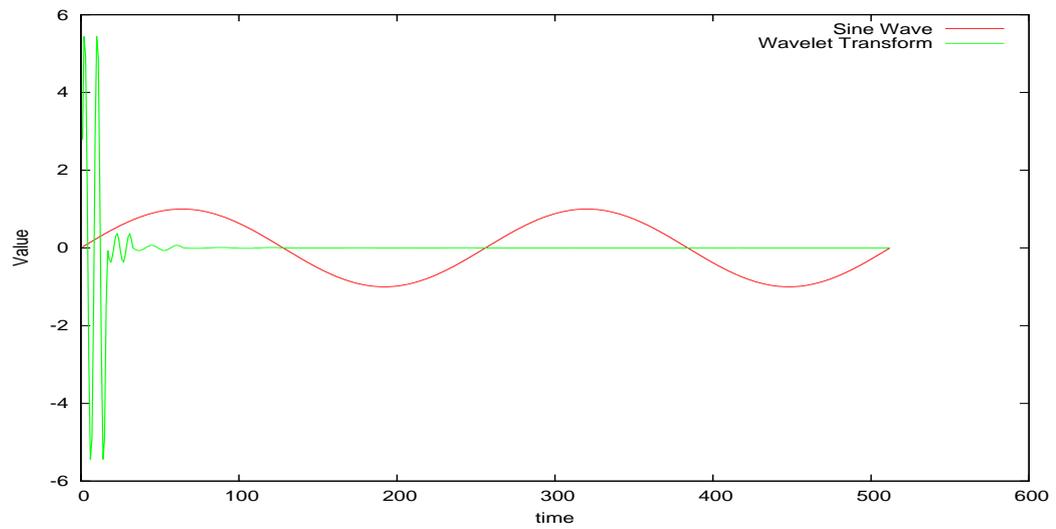


Figure 3.5: Sine wave function and its wavelet transform with Daubechies 4 wavelet. The left side of the wavelet coefficients hold the low frequency content of the signal (coarse representation), the right hand side holds the fine details.

Conjugate mirror filter implementation of the wavelet transform is computationally efficient and simple to implement. Wavelet transform can also be implemented as a matrix operator  $T_{l,s}$  where  $l$  is the input size and  $s$  is the level of the transform. Inputsize is the rowdimension of the column matrix that is to be transformed. Haar wavelet is defined by its wavelet and scaling function as:

$$\mathbf{h} = \left[ \frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}} \right] \quad (3.15)$$

$$\mathbf{g} = \left[ \frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right] \quad (3.16)$$

Where  $\mathbf{h}$  is the wavelet function and  $\mathbf{g}$  is the scaling function of the Haar wavelet. Then the one level wavelet transform matrix,  $\mathbf{T}_{2,1}$  to transform inputsize of two can be written as:

$$\mathbf{T}_{2,1} = \begin{bmatrix} \mathbf{g} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \quad (3.17)$$

The resulting matrix  $\mathbf{T}_{2,1}$  is an orthonormal matrix. It's inverse is it's transpose  $\mathbf{T}_{2,1}(\mathbf{T}_{2,1})^T = \mathbf{I}$  and  $\det(\mathbf{T}_{2,1}) = 1$ . This matrix can be used to project column vectors of length two to the wavelet domain simply by left multiplying the column vector. Let  $\mathbf{x} = [5, 3]^T$ , then the wavelet transform of  $\mathbf{x}$  is  $\mathbf{T}\mathbf{x} = [5.6569, -1.4142]^T$ . For higher input column vector sizes, the matrix  $\mathbf{T}_{l,s}$  must be extended to include translated versions of the scaling and wavelet function. The following matrix is a one level wavelet transform matrix for input size of four.

$$\mathbf{T}_{4,1} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \quad (3.18)$$

Note that the first two rows of the matrix  $\mathbf{T}_{4,1}$  in Equation 3.18 contains the translated versions of the scaling function. This is identical to downsampling and convolving the signal as in filter banks. And the next two rows contain the translated version of the wavelet function  $\mathbf{h}$  again just like in filterbanks.

The matrix  $\mathbf{T}_{4,1}$  is a one level wavelet transform matrix. To develop a two level wavelet transform matrix, the matrix  $\mathbf{T}_{2,1}$  can be used.

$$\mathbf{T}_4 = \begin{bmatrix} \mathbf{T}_{2,1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \mathbf{T}_{4,1} \quad (3.19)$$

The wavelet transform matrix for eight valued input data can be written as:

$$\mathbf{T}_8 = \begin{bmatrix} \mathbf{T}_{2,1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{4,1} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \mathbf{T}_{8,1} \quad (3.20)$$

Higher matrix dimensions can be reached by implementing a recursive algorithm. The recursive algorithm only depends on the one level wavelet transform in appropriate scale level. Haar wavelet is a very simple case of a wavelet function. It has a support of two, which leads to proper matrix formulations. However, for the case of wavelets having support of more than two, care must be taken into account in the formulation, because edge overlapping occurs. For instance, Daub4 wavelet has a support of four defined by function coefficients  $\mathbf{h} = [h1, h2, h3, h4]$  and  $\mathbf{g} = [g1, g2, g3, g4]$ . The one level wavelet transform for four valued input data is:

$$\mathbf{T}_{4,1} = \begin{bmatrix} g1 & g2 & g3 & g4 & & & & \\ 0 & 0 & g1 & g2 & g3 & g4 & & \\ h1 & h2 & h3 & h4 & & & & \\ 0 & 0 & h1 & h2 & h3 & h4 & & \end{bmatrix} \quad (3.21)$$

As it is seen from the matrix layout in Equation 3.21, the second and fourth row contain overlaps. This overlapping behaviour can be eliminated by assuming the signal to be transformed is periodic. Then the matrix  $\mathbf{T}_{4,1}$  can be written in the form:

$$\mathbf{T}_{4,1} = \begin{bmatrix} g1 & g2 & g3 & g4 \\ \mathbf{g3} & \mathbf{g4} & g1 & g2 \\ h1 & h2 & h3 & h4 \\ \mathbf{h3} & \mathbf{h4} & h1 & h2 \end{bmatrix} \quad (3.22)$$

Arranging the matrix does not effect its orthogonal property  $\mathbf{T}_{4,1}(\mathbf{T}_{4,1})^T = \mathbf{I}_{4 \times 4}$ . Pe-

riodic signal assumption is required because the wavelet transform is implemented as a circular convolution operator.

### 3.2.2 Wavelet Shrinkage

Wavelet transform is an orthogonal projection on to the wavelet domain. Thresholding the wavelet coefficients is equivalent to estimating the signal with less wavelet components(Mallat, 1998). White noise in wavelet domain produces high frequency components. Figure 3.6 shows the sinewave example with added zero mean gaussian noise of variance 0.1.

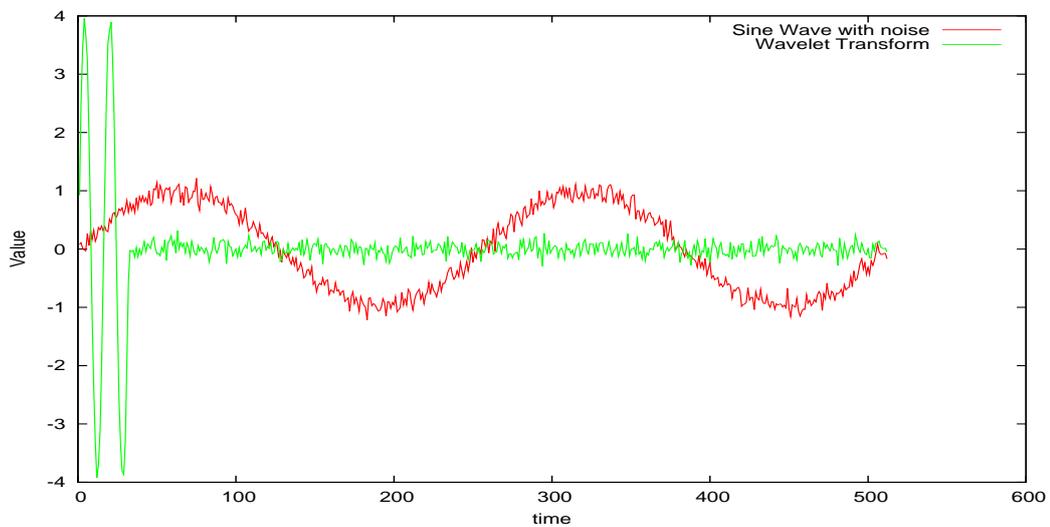


Figure 3.6: A noisy Sine wave function with zero mean gaussian noise of variance 0.01, and its 5 level wavelet transform with Daubechies 4 wavelet. Observe that the left hand side of the wavelet transform shows the trend significantly.

Thresholding the amplitudes of the wavelet coefficients which are closer to zero removes the frequency content added by white noise. Most commonly used thresholding techniques are soft and hard thresholding.

Hard thresholding can be implemented as :

$$\mathbf{W}f[i] = \begin{cases} \mathbf{W}f[i] & \text{if } |\mathbf{W}f[i]| > T \\ 0 & \text{if } |\mathbf{W}f[i]| \leq T \end{cases} \quad (3.23)$$

Where  $\mathbf{W}f[i]$  is the wavelet coefficients in the finest scale,  $T$  is the threshold value.

Similarly soft thresholding can be implemented as :

$$\mathbf{W}f[i] = \begin{cases} \mathbf{W}f[i] - T & \text{if } \mathbf{W}f[i] > T \\ \mathbf{W}f[i] + T & \text{if } \mathbf{W}f[i] < -T \\ 0 & \text{if } |\mathbf{W}f[i]| \leq T \end{cases} \quad (3.24)$$

The threshold value  $T$  is very important, since the wrong value of a threshold may remove signal content when trying to remove noise, or it may not denoise well. Donoho and Johnstone (1995) states that such an optimal threshold exists and can be calculated by

$$T = \sigma \sqrt{2 \log_e N} \quad (3.25)$$

Where  $T$  is the threshold value,  $\sigma$  is the noise standard deviation,  $N$  is the signal length. In most cases,  $\sigma$  is not known directly, and must be calculated by the signal content. Donoho and Johnstone (1994) shows that unknown signal variance can be calculated with the finest scale wavelet coefficients.

$$\hat{\sigma} = \frac{MAD(\mathbf{W}f[i])}{0.6745} = \frac{\text{median}_i(|\mathbf{W}f[i]|)}{0.6745} \quad (3.26)$$

where  $MAD$  is Median of Absolute Deviation and  $\mathbf{W}f[i]$  are the wavelet coefficients at the finest scale. Median is calculated as follows: Absolute values of the wavelet coefficients in the finest scale are sorted. Then, the median value is the middle value in the sorted array if the data length is odd, otherwise it is the mean of the two middle values.

## CHAPTER 4

### WAVELET-KALMAN FILTER

Kalman filter is a recursive optimal (linear, adaptive, unbiased and minimum error variance) filtering technique (Maybeck, 1979). Wavelet-Kalman filter, on the other hand, is a specialized Kalman filtering algorithm to make use of Kalman filter's optimality as well as to decompose the signal to its wavelet coefficients in real-time (Hong et al., 1998).

Standard Kalman filter operates on signal parameters, continuously adjusting the state variables by using measurements, such that the error covariance of the estimated state variable is minimum. Whereas, Wavelet-Kalman Filter operates on blocks of information where the block content is wavelet transform of the signal state variables. Thus, the algorithm operates in the wavelet domain in order to estimate the wavelet transform of the signal, minimizing the error variance of the estimated wavelet transform coefficients. Another advantage of this algorithm is, it computes the total wavelet transform of the filtered signal in real-time due to real-time characteristics of the Kalman filter.(Hong et al., 1998). In order to use wavelet coefficients of the signal in state vector of the Kalman filter, the signal should be processed block by block. The block size depends on the desired decomposition level of the signal. Since the wavelet decomposition relies on dyadic filter bank, the blocksize must be in powers of two. Figure 4.1 shows a simple diagram showing the filter in action.

In order to use blocks as state variables instead of the variables themselves, the system model equations, observation equation and process and measurement noise variances should be rearranged. In subsection 4.1 this rearrangement for blocksize of four is shown. And in subsection 4.2 transforming the block equations into the Wavelet-Kalman filter equations is presented.



Figure 4.1: Wavelet-Kalman filter in action. The left hand side of the figure shows the block level filtered wavelet transform of the signal, the right hand side shows the continuous feeding of the filter by measurement blocks.

## 4.1 Block Equations

In most cases, system state in Kalman filter is modelled by differential equations and continuously refined by measurements as a function of the state variables. Assume a random signal  $\mathbf{x}_k$  which is governed by the following discrete equation:

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k \quad (4.1)$$

which is observed by :

$$\mathbf{z}_k = \mathbf{C}_k \mathbf{x}_k + \mathbf{v}_k \quad (4.2)$$

where  $\mathbf{z}_k$  is measurement at time instant  $k$ ,  $\mathbf{w}_k$  is the model noise and  $\mathbf{v}_k$  is the measurement noise,  $\mathbf{A}_k$  is the state transition matrix,  $\mathbf{x}_k$  is the state vector and  $\mathbf{C}_k$  is the measurement equation that relates the the state vector to the measurements. Mean and variances as expected values of the white noise processes in terms of covariance matrices are known as follows:

$$E\{\mathbf{w}_k\} = 0 \quad (4.3)$$

$$E\{\mathbf{w}_k(\mathbf{w}_i)^T\} = \begin{cases} \mathbf{Q}_k & i = k \\ 0 & i \neq k \end{cases} \quad (4.4)$$

$$E\{\mathbf{v}_k\} = 0 \quad (4.5)$$

$$E\{\mathbf{v}_k(\mathbf{v}_i)^T\} = \begin{cases} \mathbf{R}_k & i = k \\ 0 & i \neq k \end{cases} \quad (4.6)$$

$$(4.7)$$

Since wavelet transform is performed in dyadic scales the blocksize is in powers of two and can be calculated by  $2^{level}$ . If the desired level of wavelet decomposition is two then a blocksize of four should be used. In order to run the filter, it should be fed with four measurements per block ( $[z_{k-3}, z_{k-2}, z_{k-1}, z_k]^T$ ) up to time instant  $k$ . The state vector of the system should be a vector of four elements ( $[x_{k-3}, x_{k-2}, x_{k-1}, x_k]^T$ ) and the system equations Equation 4.1, Equation 4.2 should be rearranged as follows.

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}_{k+2} \\ \mathbf{x}_{k+3} \\ \mathbf{x}_{k+4} \end{bmatrix} = \tilde{\mathbf{A}}_k \begin{bmatrix} \mathbf{x}_{k-3} \\ \mathbf{x}_{k-2} \\ \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{bmatrix} + \tilde{\mathbf{w}}_k \quad (4.8)$$

which is observed by:

$$\begin{bmatrix} z_{k-3} \\ z_{k-2} \\ z_{k-1} \\ z_k \end{bmatrix} = \tilde{\mathbf{C}}_k \begin{bmatrix} \mathbf{x}_{k-3} \\ \mathbf{x}_{k-2} \\ \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{bmatrix} + \tilde{\mathbf{v}}_k \quad (4.9)$$

Where  $\tilde{\mathbf{A}}_k$  is the matrix that relates the block at  $k$  to block at  $k + 4$ ,  $\tilde{\mathbf{C}}_k$  is the matrix that relates the state variable block to the measurement block,  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are the new system and measurement noise processes. By using the relation between  $\mathbf{x}_{k+1}$  and  $\mathbf{x}_k$ , in Equation 4.1,  $\mathbf{x}_{k+1}$  can be represented with respect to  $\mathbf{x}_{k-3}, \mathbf{x}_{k-2}, \mathbf{x}_{k-1}, \mathbf{x}_k$  by :

1.  $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + w_k$
2.  $\mathbf{x}_{k+1} = \mathbf{A}(\mathbf{A}\mathbf{x}_{k-1} + w_{k-1}) + w_k$
3.  $\mathbf{x}_{k+1} = \mathbf{A}^3\mathbf{x}_{k-2} + \mathbf{A}^2w_{k-2} + \mathbf{A}w_{k-1} + w_k$
4.  $\mathbf{x}_{k+1} = \mathbf{A}^4\mathbf{x}_{k-3} + \mathbf{A}^3w_{k-3} + \mathbf{A}^2w_{k-2} + \mathbf{A}w_{k-1} + w_k$

By averaging both sides of the above equations the relationship of  $\mathbf{x}_{k+1}$  with  $\mathbf{x}_{k-3}, \mathbf{x}_{k-2}, \mathbf{x}_{k-1}$  and  $\mathbf{x}_k$  is obtained as following:

$$\mathbf{x}_{k+1} = \frac{1}{4}[\mathbf{A}^4 \mathbf{A}^3 \mathbf{A}^2 \mathbf{A}] \begin{bmatrix} \mathbf{x}_{k-3} \\ \mathbf{x}_{k-2} \\ \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{bmatrix} + (\tilde{w}_{k+1} = w_k + \frac{3}{4}\mathbf{A}w_{k-1} + \frac{2}{4}\mathbf{A}^2w_{k-2} + \frac{1}{4}\mathbf{A}^3w_{k-3})$$

where  $\tilde{w}_{k+1}$  is the total propagated error in the calculation of  $\mathbf{x}_{k+1}$ . The equations for  $\mathbf{x}_{k+2}$ ,  $\mathbf{x}_{k+3}$  and  $\mathbf{x}_{k+4}$  can be written similarly. After all the equations are set  $\tilde{\mathbf{A}}_k$  and  $\tilde{\mathbf{w}}_k$  matrices can be written as :

$$\tilde{\mathbf{A}}_k = \begin{bmatrix} \frac{1}{4}\mathbf{A}^4 & \frac{1}{4}\mathbf{A}^3 & \frac{1}{4}\mathbf{A}^2 & \frac{1}{4}\mathbf{A} \\ 0 & \frac{1}{3}\mathbf{A}^4 & \frac{1}{3}\mathbf{A}^3 & \frac{1}{3}\mathbf{A}^2 \\ 0 & 0 & \frac{1}{2}\mathbf{A}^4 & \frac{1}{2}\mathbf{A}^3 \\ 0 & 0 & 0 & \mathbf{A}^4 \end{bmatrix}$$

$$\tilde{\mathbf{w}}_k = \begin{bmatrix} w_k + \frac{3}{4}\mathbf{A}w_{k-1} + \frac{2}{4}\mathbf{A}^2w_{k-2} + \frac{1}{4}\mathbf{A}^3w_{k-3} \\ w_{k+1} + \mathbf{A}w_k + \frac{2}{3}\mathbf{A}^2w_{k-1} + \frac{1}{3}\mathbf{A}^3w_{k-2} \\ w_{k+2} + \mathbf{A}w_{k+1} + \mathbf{A}^2w_k + \frac{1}{2}\mathbf{A}^3w_{k-1} \\ w_{k+3} + \mathbf{A}w_{k+2} + \mathbf{A}^2w_{k+1} + \mathbf{A}^3w_k \end{bmatrix}$$

The observation matrix  $\tilde{\mathbf{C}}_k$  in Equation 4.9 can be written as  $diag(\mathbf{C}, \mathbf{C}, \mathbf{C}, \mathbf{C})$ , assuming  $\mathbf{C}_k$  is a constant matrix  $\mathbf{C}$ . Transferring the equations in to the block form results in a new set of process equations which is directly related to the original process defined by Equation 4.1 and Equation 4.2 as:

$$E\{\tilde{w}_k\} = 0 \quad (4.10)$$

$$E\{\tilde{v}_k\} = 0 \quad (4.11)$$

since  $\tilde{w}_k$  and  $\tilde{v}_k$  are linear combinations of  $w_k$  and  $v_k$  respectively. The new process and measurement variances are

$$E\{\tilde{v}_k(\tilde{v}_k)^T\} = diag(\mathbf{R}, \mathbf{R}, \mathbf{R}, \mathbf{R}) \quad (4.12)$$

$$E\{\tilde{w}_k(\tilde{w}_k)^T\} = \tilde{\mathbf{Q}} \quad (4.13)$$

The symmetric matrix  $\tilde{\mathbf{Q}}$  is the variance-covariance matrix of the new system associ-

ated with the system noise. The matrix  $\tilde{Q}$  is not diagonal since inter block relations result in correlations. The matrix content is given below element-wise . Note that the equations are simplified by using the statistics of process noise  $w_k$  ( $E\{w_k w_j\} = 0, k \neq j$  and  $E\{w_k w_k^T\} = Q$ ).

$$\tilde{Q}_{11} = Q(1/16A^6 + 1/4A^4 + 9/16A^2 + 1)$$

$$\tilde{Q}_{12} = Q(1/6A^5 + 1/2A^3 + A)$$

$$\tilde{Q}_{13} = Q(3/8A^4 + A^2)$$

$$\tilde{Q}_{14} = QA^3$$

$$\tilde{Q}_{21} = \tilde{Q}_{12}$$

$$\tilde{Q}_{22} = Q(1/9A^6 + 4/9A^4 + A^2 + 1)$$

$$\tilde{Q}_{23} = Q(1/3A^5 + A^3 + A)$$

$$\tilde{Q}_{24} = Q(A^4 + A^2)$$

$$\tilde{Q}_{31} = \tilde{Q}_{13}$$

$$\tilde{Q}_{32} = \tilde{Q}_{23}$$

$$\tilde{Q}_{33} = Q(1/4A^6 + A^4 + A^2 + 1)$$

$$\tilde{Q}_{34} = Q(A^5 + A^3 + A)$$

$$\tilde{Q}_{41} = \tilde{Q}_{14}$$

$$\tilde{Q}_{42} = \tilde{Q}_{24}$$

$$\tilde{Q}_{43} = \tilde{Q}_{34}$$

$$\tilde{Q}_{44} = Q(A^6 + A^4 + A^2 + 1)$$

## 4.2 Wavelet Decomposition

The block equations of the previous chapter can be projected onto the wavelet domain by using a projection matrix  $T$ , which can be calculated by the algorithm defined in Chapter 3. Representing wavelet projection in matrix form is important since the resultant wavelet domain equations can be used with existing Kalman filter implementations. The block state variables can be represented in a matrix  $X_k$  as :

$$\mathbf{X}_k = \begin{bmatrix} x_{k-3} \\ x_{k-2} \\ x_{k-1} \\ x_k \end{bmatrix} \quad (4.14)$$

The state matrix  $\mathbf{X}_k$ , which contains the block state variables, can be transformed into the wavelet domain by :

$$\bar{\mathbf{X}}_k = \mathbf{T} \mathbf{X}_k \quad (4.15)$$

Where  $\bar{\mathbf{X}}_k$  contains the wavelet coefficients of the state variables. Substituting the wavelet coefficients into the Equation 4.8 and Equation 4.9 Wavelet domain process equations can be written as :

$$\bar{\mathbf{X}}_{k+1} = \bar{\mathbf{A}} \bar{\mathbf{X}}_k + \bar{\mathbf{W}}_k \quad (4.16)$$

Which is observed by :

$$\mathbf{Z}_k = \bar{\mathbf{C}} \bar{\mathbf{X}}_k + v_k \quad (4.17)$$

Where  $\bar{\mathbf{A}} = \mathbf{T} \tilde{\mathbf{A}} \mathbf{T}^T$ ,  $\bar{\mathbf{C}} = \tilde{\mathbf{C}} \mathbf{T}^T$  and  $\bar{\mathbf{W}}_k = \mathbf{T} \tilde{w}_k$ . The mean and variance of the new process noise is  $E\{\bar{\mathbf{W}}_k\} = 0$  and  $E\{\bar{\mathbf{W}}_k (\bar{\mathbf{W}}_k)^T\} = \mathbf{T} \tilde{\mathbf{Q}} \mathbf{T}^T$  respectively. The Standard Kalman filter can be directly applied to the system to obtain the estimates of the wavelet coefficients.

At every Kalman step the wavelet coefficients  $\bar{\mathbf{X}}_k$  is corrected by using the measurement block  $\mathbf{Z}_k$  and the measurement equation 4.17. The output of the correction step can be the wavelet transform of the block  $\bar{\mathbf{X}}_k$ , or the signal representation  $\hat{\mathbf{X}}_k$ . The signal representation of the block is calculated by applying inverse wavelet transform to the block.

$$\hat{\mathbf{X}}_k = \mathbf{T}^T \bar{\mathbf{X}}_k \quad (4.18)$$

After the correction step, next Wavelet-Kalman block is predicted by using the adopted system stochastic model equation 4.16.

### 4.3 Integrating The Wavelet Shrinkage

Wavelet-Kalman filter is a linear filter that estimates the wavelet coefficients in terms of the state variables of the given system. Non-linear wavelet thresholding, as described in Section 3.2.2, may be used to improve noise reduction in the Wavelet-Kalman correction step. After correcting the Wavelet-Kalman state variables soft or hard thresholding can be applied to the state variable.

The threshold value can either be calculated by Equation 3.26 or the state error covariance matrix of the estimated state. System state error covariance matrix  $\mathbf{P}_k$  in Equation 2.56 contains the error covariance matrix of the estimated wavelet coefficients. However,  $\sigma$  needed in the threshold value in Equation 3.25 is the standard deviation of the system state variables not the coefficients in the wavelet domain. The system error covariance matrix  $\mathbf{P}_k^s$  can be calculated by utilizing the law of error propagation.

$$\mathbf{P}_k^s = \mathbf{T}^T \mathbf{P}_k \mathbf{T} \quad (4.19)$$

Then the system standard deviation estimate  $\hat{\sigma}$  can be calculated by taking the square root of the average of the diagonal elements.

$$\hat{\sigma} = \sqrt{\frac{\sum_i \mathbf{P}_k^s[l, i]}{\dim(\mathbf{P}_k^s)}} \quad (4.20)$$

Where  $\mathbf{P}_k^s[i][i]$  is the  $i^{th}$  diagonal element, and  $\dim(\mathbf{P}_k^s)$  is the diagonal dimension of the matrix  $\mathbf{P}_k^s$ .

### 4.4 Monte-Carlo Simulations and Results

Filtering performance of the Wavelet-kalman filter is tested with monte-carlo simulations on a simple signal of length 512 with additive noise of variance 1. The signal can be treated as a recording of a Brownian Motion which is described by its system equation :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{w}_k \quad (4.21)$$

which is observed by :

$$z_k = x_k + v_k \quad (4.22)$$

Where  $z_k$  is the signal content,  $x_k$  is the system state variable,  $w_k$  and  $v_k$  are process and measurement noise respectively with

$$E\{w_k\} = 0 \quad (4.23)$$

$$E\{v_k\} = 0 \quad (4.24)$$

$$E\{w_k(w_i)^T\} = \begin{cases} 0.1 & i = k \\ 0 & i \neq k \end{cases} \quad (4.25)$$

$$E\{v_k(v_i)^T\} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases} \quad (4.26)$$

Filter performance is tested with calculating the MSE (Mean Square Error). The MSE is calculated as:

$$mse = \sqrt{\frac{\sum_i (f[i] - \hat{f}[i])^2}{N - 1}} \quad (4.27)$$

Where  $f[i]$  is the original signal,  $\hat{f}[i]$  is the filtered signal and  $N$  is the signal length. Smaller mse value means better estimation of the signal.

Figure 4.2 shows the original signal before adding noise.

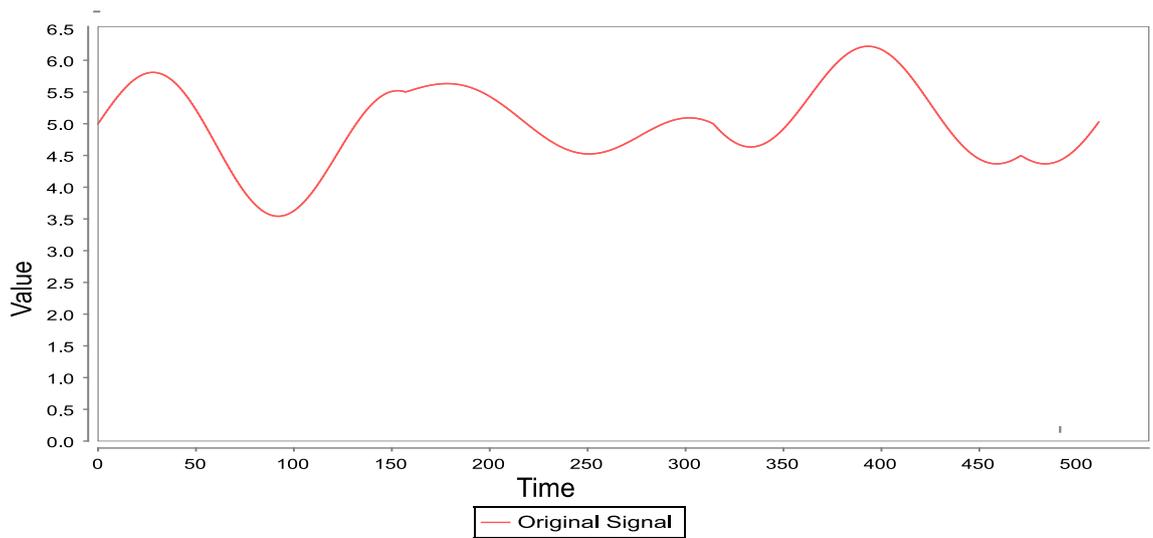


Figure 4.2: Original signal generated by  $f(t) = 5 + |\sin(t/50.0)|\cos(t/30.0) + 1/2.0\sin(t/20.0)$

The noisy version of the signal looks like as in Figure 4.3.

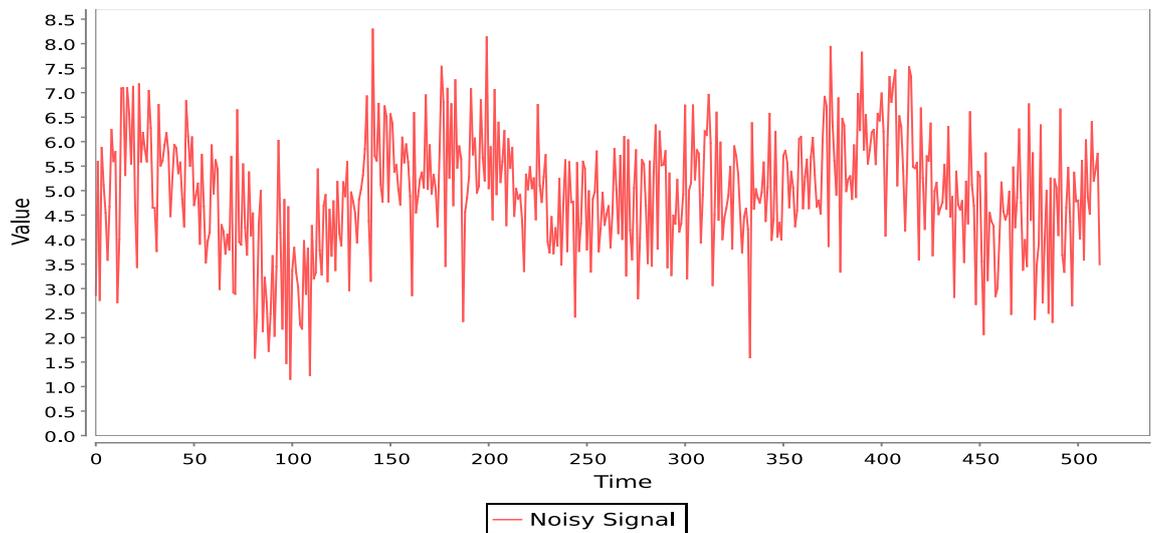


Figure 4.3: Original signal with additive Gaussian white noise with  $\sigma=1$  and  $\mu=0$

Standard Kalman filter and Wavelet-Kalman filter is applied to the noisy signal 200 times. At each iteration a new signal is generated from the original signal by adding Gaussian white noise of variance 1 and mean 0. The process noise variance is set to 0.1.

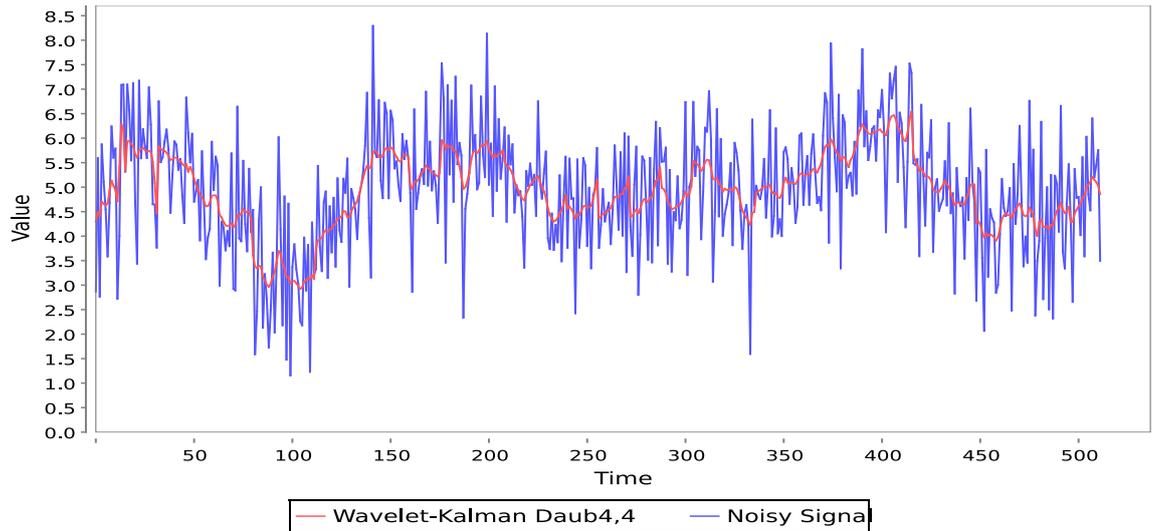


Figure 4.4: Wavelet Kalman Filter result with level=4, Daub4 wavelet and process noise variance set to 0.1. mse(Mean Square Error)= 0.015083.

The mean square errors for both thresholding enabled and disabled versions are collected and their averages are shown in Table 4.1 for performance comparison of 2 level Wavelet-Kalman filter.

Table 4.1: Monte-Carlo Performance of filters of level two

Filter	mse(mean)	mse(mean,thresholded)
WK(Haar)	0.0115403	0.0113227
WK(Daub4)	0.0115403	0.0112438
WK(Daub6)	0.0115403	0.0110092
WK(Coiflet6)	0.0115403	0.0110708
Kalman	0.0125757	

According to Table 4.1, Wavelet-Kalman filter performs better than the Standard Kalman filter. Moreover, it is obvious that the choice of wavelet function does not change the resulting mean square error. This is due to the orthonormal projection property of the wavelet transform, which means perfect (lossless) reconstruction of a signal from it's wavelet coefficients. Thresholding the coefficients of the state variables improves the resulting mean square error. In the case of thresholding, Daub6 wavelet performs better than other wavelets.

Table 4.2 shows the results of the filters at decomposition level of four. The performance of the Wavelet-Kalman filter is increasing as the decomposition level increase. Daub6 wavelet performs better than other wavelet types. This shows that wavelet

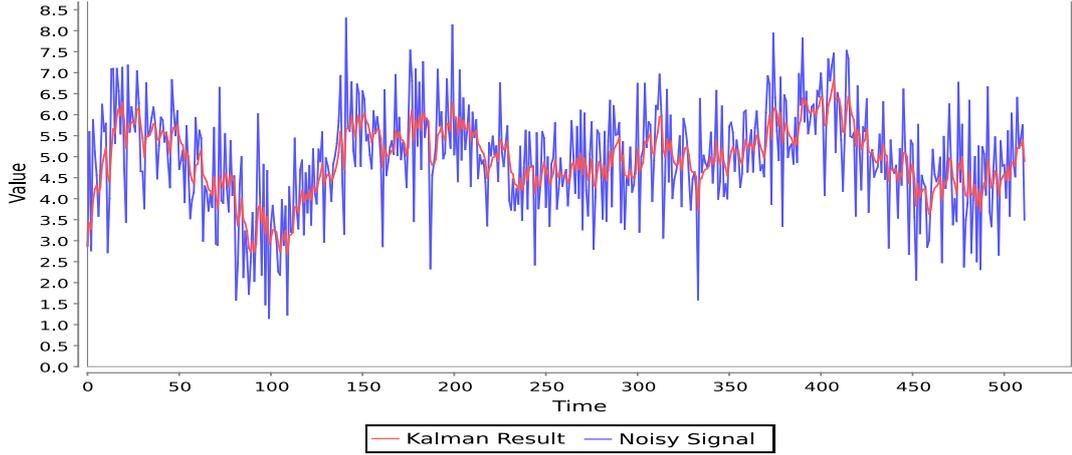


Figure 4.5: Standard Kalman Filter result with process noise variance set to 0.1.  $mse(\text{Mean Square Error})=0.017433$ .

Table 4.2: Monte-Carlo Performance of filters of level four

Filter	mse(mean)	mse(mean,thresholded)
WK(Haar)	0.0105139	0.0103911
WK(Daub4)	0.0105139	0.0103984
WK(Daub6)	0.0105139	0.0103374
WK(Coiflet6)	0.0105139	0.0103428
Kalman	0.0126197	

thresholding performance is related to the signal content and the input size.

Table 4.3: Monte-Carlo Performance of filters of level six

Filter	mse(mean)	mse(mean,thresholded)
WK(Haar)	0.0094232	0.0093188
WK(Daub4)	0.0094232	0.0094184
WK(Daub6)	0.0094232	0.0094697
WK(Coiflet6)	0.0094232	0.009442
Kalman	0.0126048	

Table 4.3 proves that as the decomposition level increases, the Wavelet-Kalman filter performance increases. However, the wavelet thresholding does not improve the results as the decomposition level increases. This can be due to improved performance of Wavelet-Kalman filter which removes high frequencies from the signal content leading to less thresholding. Table 4.4 shows the result of thresholding where the standard deviation  $\hat{\sigma}$  is calculated with the equation 3.26.

According to Table 4.4, when variance, estimated by Equation 3.26, is used to

Table 4.4: Monte-Carlo Performance of filters of level six, with thresholding based on MAD variance estimation

Filter	mse(mean)	mse(mean,thresholded)
WK(Haar)	0.0094146	0.0093091
WK(Daub4)	0.0094146	0.0094109
WK(Daub6)	0.0094146	0.0094644
WK(Coiflet6)	0.0094146	0.0094332
Kalman	0.0125667	

estimate thresholding, Wavelet-Kalman Filter is not improved further.

Figure 4.6 shows the wavelet coefficients of Daub4 wavelet after applying wavelet thresholding. Wavelet threshold is chosen with the Equation 3.25 taking  $\sigma = 1$  and  $N = 512$ . Figure 4.7 shows the reconstructed signal from thresholded coefficients. The mean square error for simple wavelet hard thresholding (0.46274) is much higher than the Wavelet Kalman Results (0.0094146).

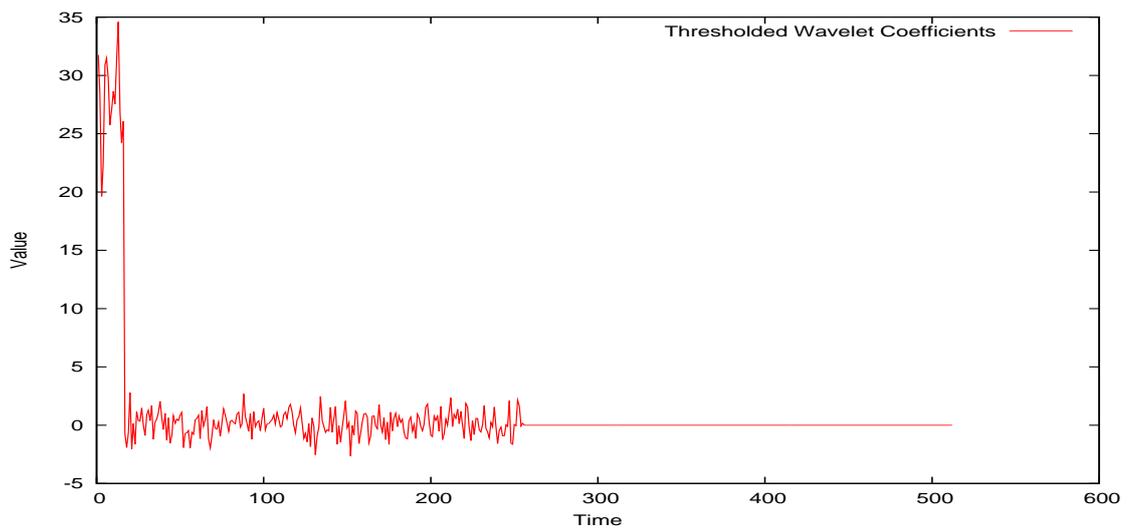


Figure 4.6: Thresholded wavelet coefficients with 4 level wavelet transform with Daub4 wavelet.  $\sigma = 1$ ,  $N = 512$ , threshold is calculated according to Equation 3.25.

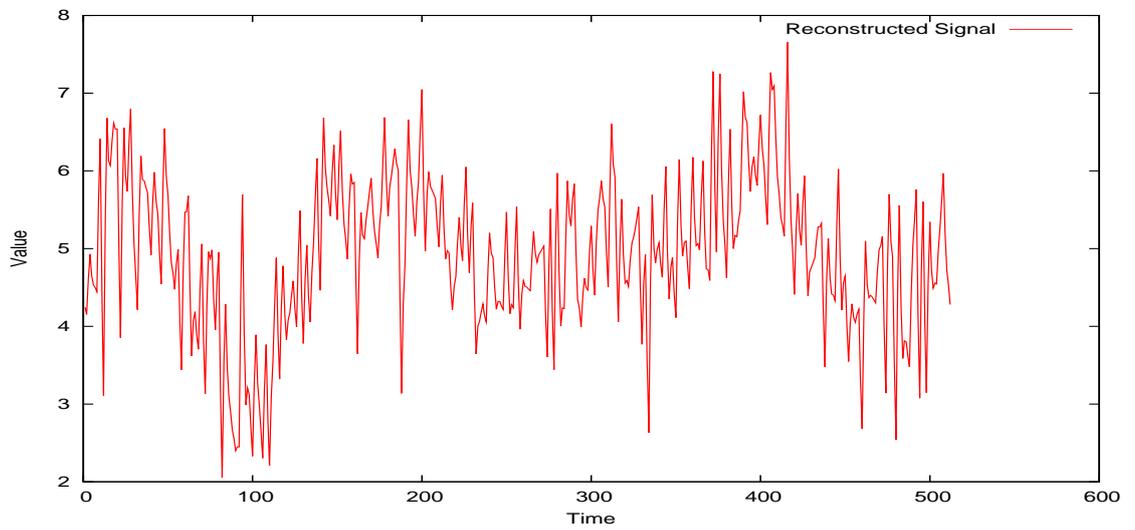


Figure 4.7: Reconstructed signal after wavelet thresholding, mse=0.46274.

## CHAPTER 5

### APPLICATIONS

Wavelet-Kalman Filter can be applied to a diversity of signal processing applications, especially to those that require real-time processing. However, the filter can also be applied to offline (not real-time) data after the data is collected. This chapter shows how Wavelet-Kalman filter derived in Chapter 4 can be used to filter real signals and show the comparison of the filter with certain existing filtering applications. Two different types of earth related signals are studied. The first one is the seismic signals from Gönen/Balıkesir earthquake recorded on 20<sup>th</sup> of November 2006. The second type of the signal is the well logs of spotaneos potentials from DSİ (Devlet Su İşleri - General Directorate of State Hydraulic Works, Turkey) well logging devices.

Wavelet-Kalman Filter results are compared to the existing filtering results by means of a signal to noise ratio. The signal to noise shows how well the filter suppressed the noise power while the signal power is preserved. it is usually measured in decibells (dB) and its is a logarithmic scale measure. There are different types of SNR calculations in the literature (Mallat, 1998; Petrou and Bosdogianni, 1999), since the true signal is not known in this case an approximate SNR can be calculated as :

$$SNR = 10\log_{10}\left(\frac{\sum_i \hat{f}[i]^2}{\sum_i (f[i] - \hat{f}[i])^2}\right) \quad (5.1)$$

where  $\hat{f}[i]$  is the filtered signal,  $f[i]$  is the measured signal. The SNR increases as the estimation error decreases. Although the SNR is a quantitative measure of filter quality, visual quality check is also required to compare the results.

Wavelet-Kalman filter and Kalman Filter is applied to the real signals with the Brownian Motion system model which approximates the system model as it is given in Equation 4.21 and 4.22. Process noise variance resulting from this uncertainty is tuned according to application needs by changing process noise variance in the filter.

## 5.1 Application to Seismic Waves

Seismic waves coming from earthquakes are recorded by strong motion accelerometers. Figure 5.1 shows a strong motion accelerometer.

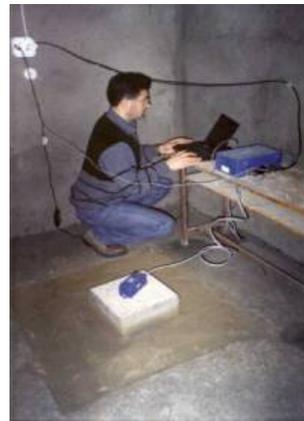


Figure 5.1: GSR-16 accelerometer with a dynamic range of 72/96 dB,  $SNR > 102\text{dB}$  (Geosig, 2007)

Accelerometers are located on ground in a fixed position in public buildings or in small observatories as in figure 5.2 (Afet-İşleri, 2007).



(a) An observation site with GSR-16 accelerometer.



(b) Inside of the observation site

Figure 5.2: Location of a sample GSR-16 observation site (a) and its inside (b) (Afet-İşleri, 2007)

The Figure 5.3 shows acceleration record of Gönen/Balıkesir Earthquake occurred on November 20, 2006 at 18:15 pm local time in the north to south direction.

The signal was prefiltered with Butterworth bandpass filter with low frequency set to 0.08 hz, and high frequency is set to 25 hz to remove high frequency com-

## Gönen Earthquake 20/10/2006

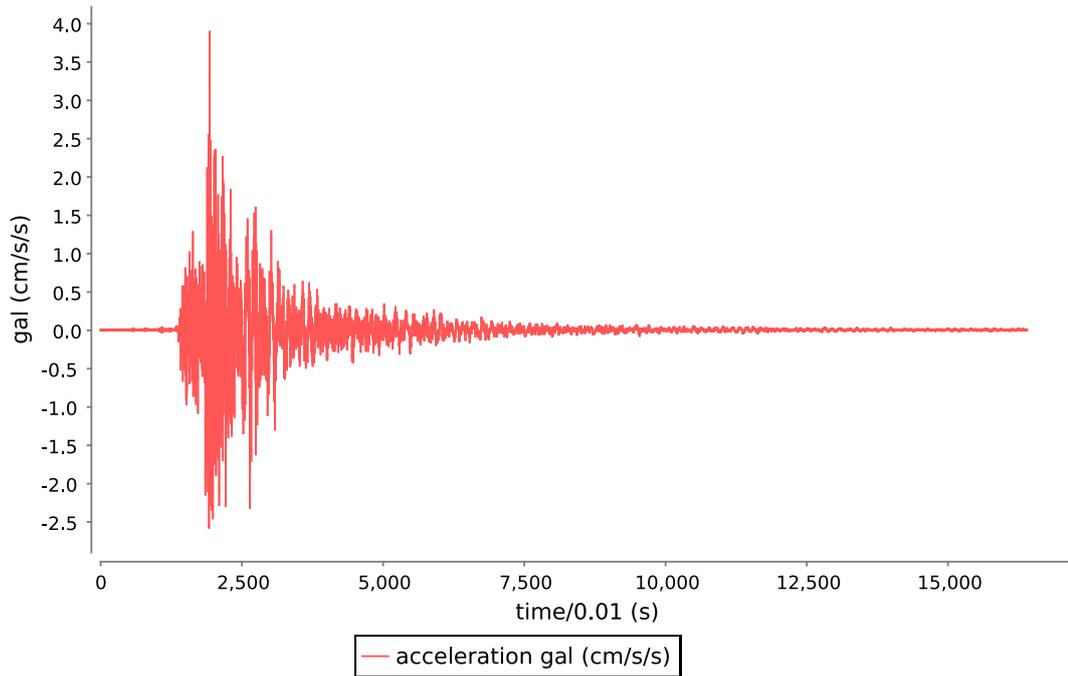


Figure 5.3: Gönen/Balıkesir Earthquake occurred on 20/10/2006 18:15 pm. localtime

ponents from the signal. The frequency response of the earthquake signal and the butterworth filter is shown in Figure 5.4

The reconstructed version of the Butterworth filtered signal zoomed to [2500-2600] in the time axis is shown in Figure 5.5. This reconstruction produces a signal to noise ratio of 67.154 dB, calculated using Equation 5.1. Figure 5.6 shows the same signal reconstructed after Wavelet-Kalman filter.

To apply the Wavelet-Kalman filter to the signal based on the system model of brownian motion, the signal noise variance and process noise variance are needed. The approximate value for signal noise variance can be estimated from mean square difference of the butterworth-filtered signal and the original signal.

$$\hat{\sigma}^2 = \frac{\sum_i (f[i] - f_b[i])^2}{N - 1}$$

where  $\hat{\sigma}^2$  is the estimated noise variance,  $f[]$  is the original recorded signal,  $f_b[]$  is the filtered signal with butterworth filter,  $N$  is the record length. The estimated variance for the signal is found to be 0.000006361. The process variance is the unknown variance value and it is to be found by experimenting the filter to find the best match. This is the so called tuning phase of the Kalman filter (Maybeck, 1979). The best

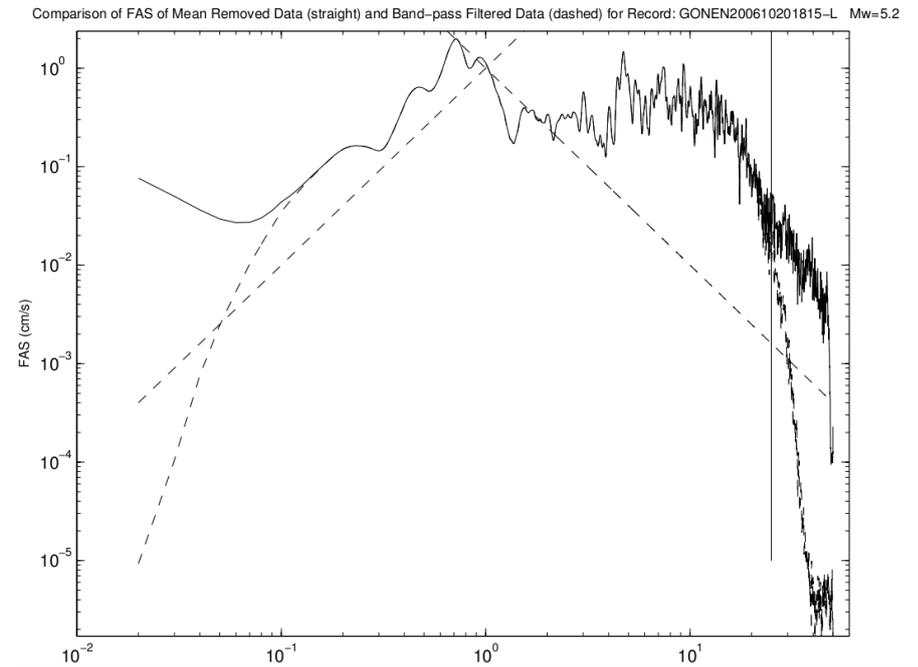


Figure 5.4: Butterworth filter and the frequency response (Fourier Transform) of the Gönen Earthquake record.

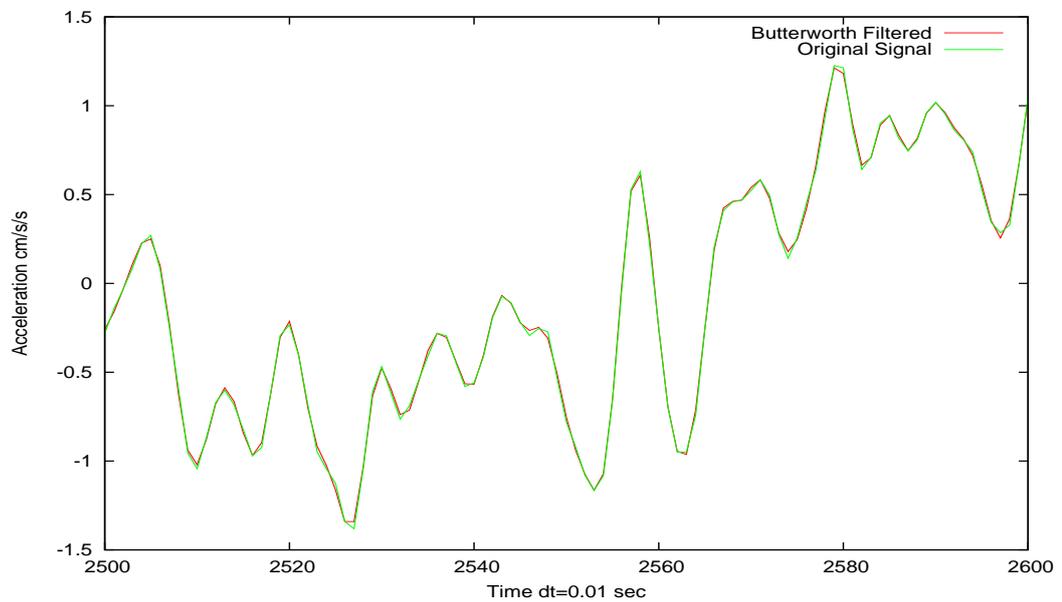


Figure 5.5: Reconstructed and original signal after butterworth filter zoomed to [2500-2600] in time axis. The signal to noise ratio is 67.154 dB

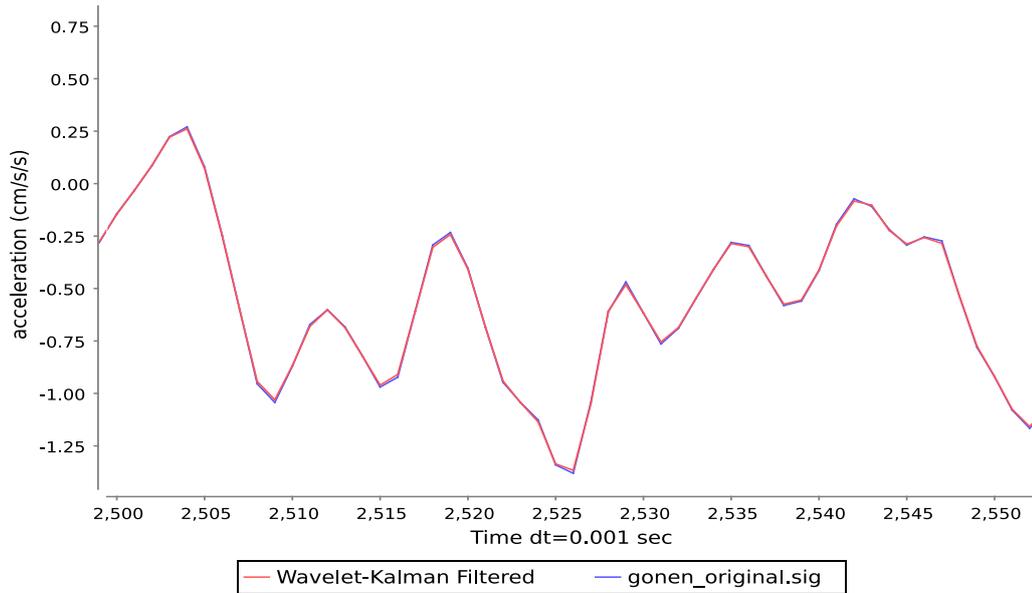


Figure 5.6: Reconstructed and original signal after Wavelet-Kalman filter. Wavelet-Kalman level is 4 and process noise variance is set to 0.00012722. The graph is zoomed to [2500-2550] in time axis. The resulting signal to noise ratio is 68.848072 dB

match is decided to be the process noise variance that minimizes the mean square error between the wavelet-kalman filtered signal and the butterworth filtered signal. The best matched process noise variance is found to be 0.00012722. The comparisons of Wavelet-Kalman filters and Kalman filters are shown in table 5.1, mse values are calculated between the Wavelet-Kalman Filtered signal and the butterworth filtered signal.

Table 5.1: 2 level Wavelet-Kalman Filter performance comparisons on seismic signal filtering.

Filter	SNR(dB)	mse	SNR(dB,threshold)	mse(threshold)
WK(Haar)	66.9673663	0.0000827	66.6932399	0.000083
WK(Daub4)	66.9673663	0.0000827	66.70236	0.000083
WK(Daub6)	66.9673666	0.0000827	66.7030922	0.000083
WK(Coiflet6)	66.9673663	0.0000827	66.7042513	0.000083
Kalman	64.9747688	0.0000842		

According to Table 5.1 Wavelet Kalman filter provides better signal to noise ratio for the best match process noise variance than the Butterworth-Filtered signal. Wavelet-Kalman filter gives better performance at any rate than the standard Kalman filter. A drop is detected in the SNR when the thresholding is enabled. This is referred

to the smoothing effect of the wavelet shrinkage which decreases the signal power. Coiflet6 wavelet seems to perform better than other wavelets in filtering the seismic signals with Wavelet-Kalman filter of level two.

Table 5.2: 4 level Wavelet-Kalman Filter performance comparisons on seismic signal filtering.

Filter	SNR(dB)	mse	SNR(dB,threshold)	mse(threshold)
WK(Haar)	69.4309269	0.0000807	68.8420142	0.0000813
WK(Daub4)	69.4309269	0.0000807	68.9551482	0.0000812
WK(Daub6)	69.4309269	0.0000807	69.0048833	0.0000812
WK(Coiflet6)	69.4309269	0.0000807	68.9649212	0.0000812
Kalman	64.9747688	0.0000842		

Table 5.3: 6 level Wavelet-Kalman Filter performance comparisons on seismic signal filtering.

Filter	SNR(dB)	rmse	SNR(dB,threshold)	rmse(threshold)
WK(Haar)	69.5676505	0.0000808	68.7896925	0.0000815
WK(Daub4)	69.5676505	0.0000808	68.9596389	0.0000814
WK(Daub6)	69.5676505	0.0000808	69.0316703	0.0000813
WK(Coiflet6)	69.5676505	0.0000808	68.9801525	0.0000813
Kalman	64.9747688	0.0000842		

Tables 5.2 and 5.3 show that as the level of decomposition increase the SNR value increases as well. When thresholding is enabled in the Wavelet-Kalman Filter, Daub6 wavelet turned out to be more suitable for seismic signal denoising than other wavelet types. This should be because of the polynomial approximation property of Daubechies wavelets (Daubechies, 1988). When compared to Butterworth filtered signal, Wavelet-Kalman filters produce maximum SNR value at 6 level decomposition as 69.5676505 dB which is better than 67.154 dB value of Butterworth filter.

## 5.2 Application to Well Logs

Well logging is a procedure to find information about the earth's crust while drilling for different purposes like oil/gas or groundwater. Well logging is the process of measuring some electrical or physical properties of rocks (Jorden and Campbell, 1986). The mostly used measurements are spontaneous potential (SP) and resistiv-

ity (RT). SP is the measured potential differences in the borehole caused by different chemical properties of rocks. Resistivity is related to the electrical currents in the borehole. The interpretation of the measurements give clues about:

- Formation fluid saturation.
- Net reservoir thickness.
- Formation porosity.
- Formation water resistivity.
- Structure of the strata.

Therefore it's an important part of formation evaluation (Jorden and Campbell, 1986). The signals used in this study are SP signals measured by DSI (General Directorate of State Hydraulic Works, Turkey) well log measuring devices with measurement quality of 1 millivolts. SP is the measure of potential difference between different strata through the borehole. Spontaneous potentials are formed by electrochemical potentials, electrokinetic potentials and redox potentials. Different diffusion speeds of ions in the solutions and boundaries of junctions develop electrochemical potential. Electrokinetic potential forms on porous material when an ionic solution moves through it (Jorden and Campbell, 1986). Figure 5.7 shows a simplified well logging schema.

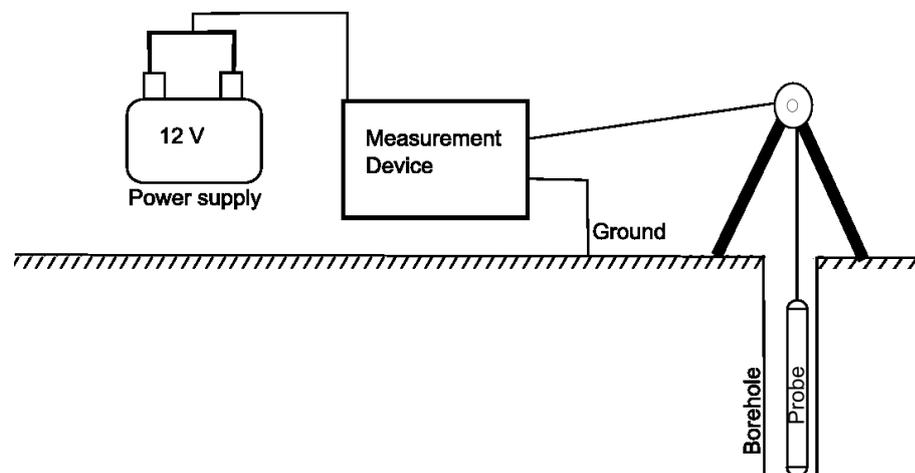


Figure 5.7: Simplified Well logging schema, measurement device logs the SP and RT through the borehole.

The signal is collected with a computer attached to the measurement device via a serial port. The measurement device measures the potentials and convert the po-

tential values to digital values in millivolts to be read and recorded by the computer. Moreover, it amplifies the measured signal to adjust them to the dynamic range of the analog to digital converter. Figure 5.8 shows a sample SP log of a well log.

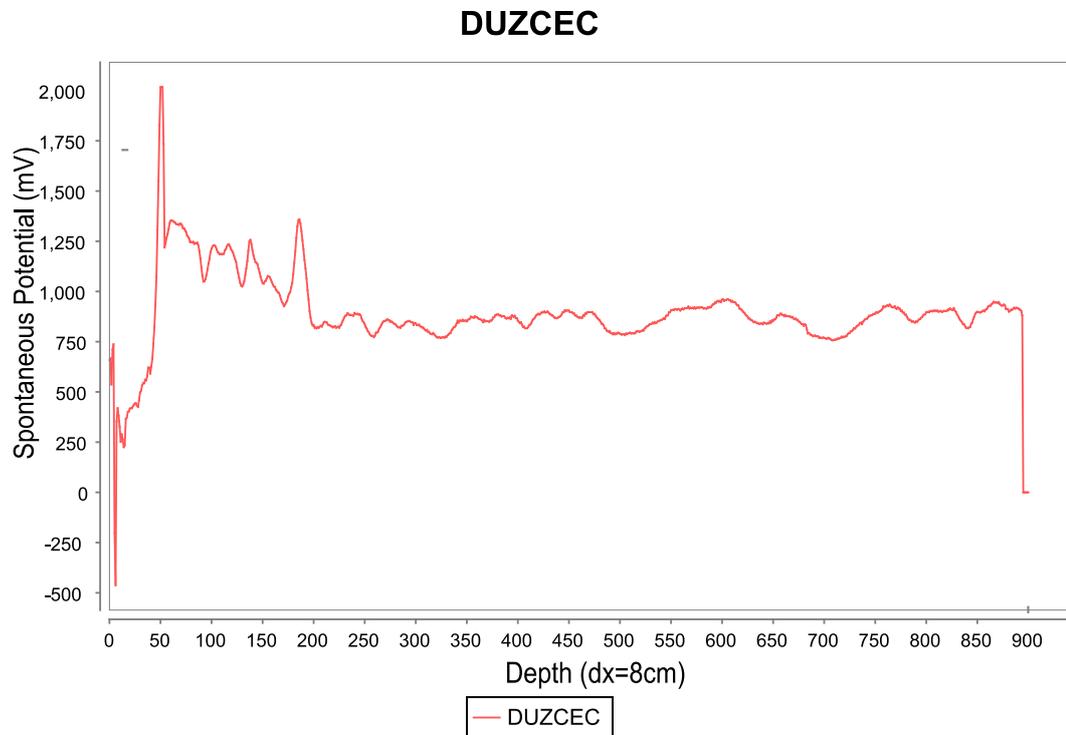


Figure 5.8: Sample spontaneous potential log plot.

The measurement device has a precision of 1 mV. However, due to amplification of the signal the resulting precision is the multiple of amplification. The signal noise is removed by a simple moving averages filter (MAF). Moving averages filter is a simple averaging convolution operator. The width of the convolution kernel is controlled by user. Figure 5.9 shows a simple noisy well log signal and Figure 5.10 shows the MAF filtered output with kernel width of 5. The graph is zoomed for better visibility of filter results. Figure 5.11 shows the same signal filtered with Wavelet-Kalman Filter of level 4 and process variance of 20, again the graph is zoomed in for better visibility.

Figures 5.10 and 5.11 show that Wavelet-Kalman filter can be used to eliminate noise from the SP signals. Moreover, the performance becomes better for properly chosen process noise variance. The choice of process noise variance is dependent on the resulting smoothness of the filtered output. The bigger the process noise vari-

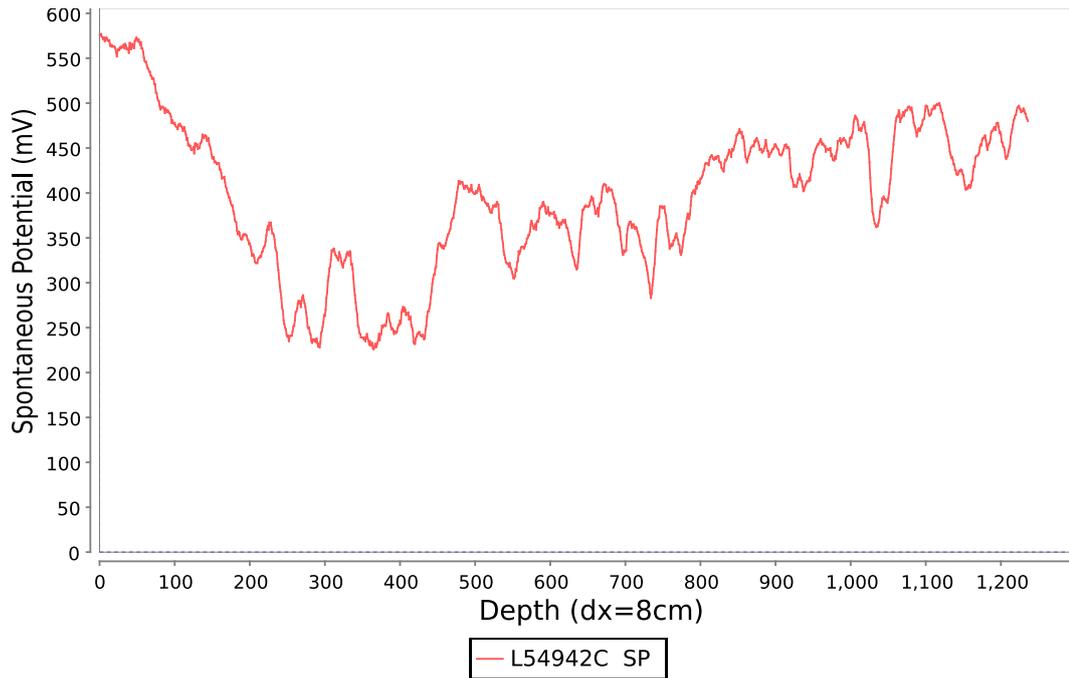


Figure 5.9: A noisy SP log of a well where the amplification is 20.

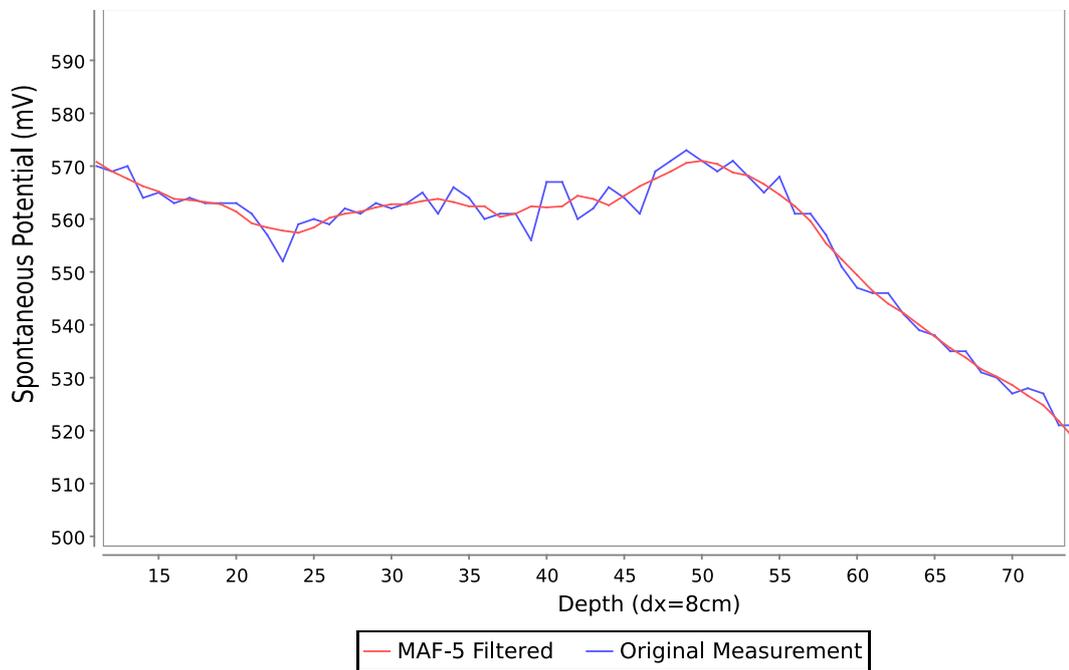


Figure 5.10: Moving averages filtered version of the signal in Figure 5.9. The MAF kernel is 5 and resulting SNR is 90.7634124 dB

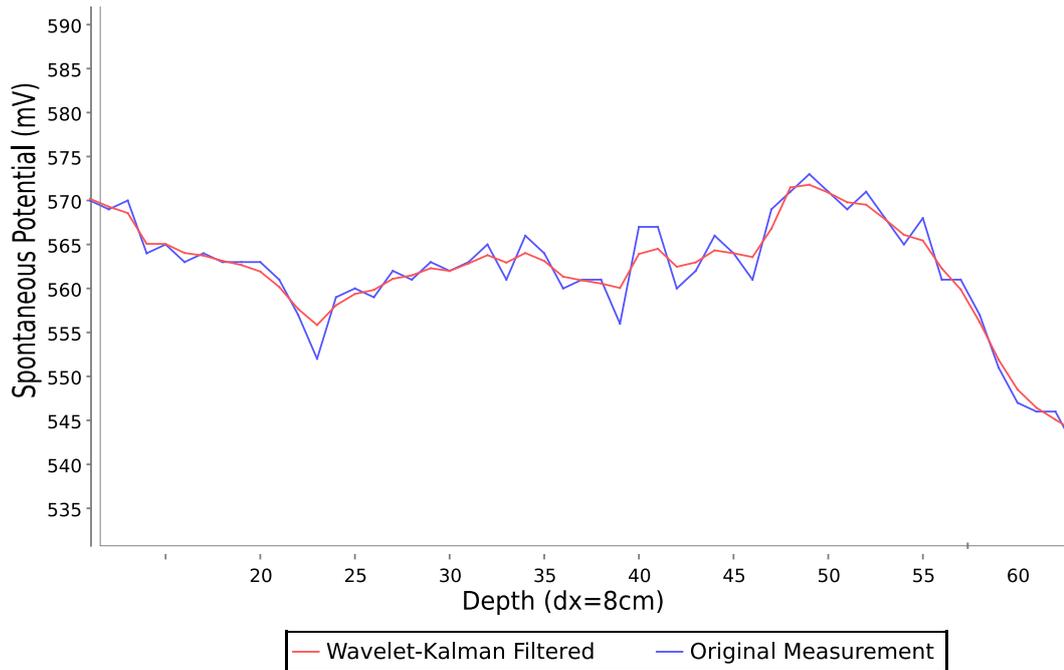


Figure 5.11: Wavelet-Kalman filtered version of the SP signal in Figure 5.9. Level is 4 and process variance is set to 20. The resulting SNR is 96.53214549.

ance the bigger the resulting SNR. However, this results in less noise content removal. So the choice of the process noise variance should be user controlled like the kernel within MAF filter. Table 5.4 shows the relationship between SNR and process noise variance. It is obvious that as process noise variance increases the SNR value increases also. This shows that, increasing the process noise variance decreases the removed noise power.

Table 5.4: The relation between process variance and SNR

Process Variance	SNR(dB)
1	72.17045
5	85.31541
10	90.88055
20	96.53214
30	100.03057

Table 5.5 shows the relation between the level of the Wavelet-Kalman filter and the SNR value with the process variance set to constant value of 20. As the level increases the SNR value increase also, resulting better estimation with the same process variance. Table 5.6 shows the relation of MAF filter kernel width and SNR. The

Table 5.5: The relation between Wavelet-Kalman level and SNR

level	SNR(dB)
2	94.48818
4	96.53214
6	96.92265

Table 5.6: The relation between MAF kernel width and SNR.

level	SNR(dB)
3	95.84879
5	90.76341
7	86.73729

SNR drops rapidly as the kernel width of the MAF filter increases.

Tables (5.6, 5.5, 5.4) shows that using MAF filters with higher kernel widths may result in significant signal content loss. Choice of process variance and level of Wavelet-Kalman filter gives the ability to the user to control the level of noise removal better than MAF filter.

## CHAPTER 6

### CONCLUSION

It results from the Monte-Carlo simulations that proposed algorithm shows better performance than the Standard Kalman filter. It performs at the same time the wavelet decomposition of the signal in real-time, which is very useful to analyse the signal in multiresolutions in real-time. Additionally, real-time wavelet thresholding was integrated in the algorithm which improves the filter performance.

Also the experiments on case studies showed that Wavelet based Kalman Filtering algorithm suppresses noise more than the standard Kalman Filter by using real, earth related signals such as well logs and seismic waves. Comparisons are accomplished with some certain existing filtering applications in terms of resulting signal to noise ratios. Real-time characteristics and the performance of the filter makes it a useful method for real-time signal analysis and control. It turned out that Brownian motion system model combined with Wavelet-Kalman filter is a very useful tool to model signals whose deterministic part is not known directly. The choice of process variance gives the user the ability to control the smoothness of the resulting signal.

It is also observed that using different kind of wavelet functions inside the proposed algorithm gives the same noise removal level in the Wavelet-Kalman Filter. This means that the Wavelet-Kalman filter without wavelet thresholding is independent of the wavelet function chosen. On the other hand, integrating wavelet based thresholding in the algorithm gives better results for different wavelet types. Moreover, introducing higher block sizes with or without thresholding increases the filter performance.

The algorithm is also applied to some real world signals of earth sciences. Application to the seismic signals showed that if properly tuned, the proposed algorithm could replace the existing filtering technique of Butterworth filter. It is also

observed that Daubechies6 wavelet gives better results than other wavelets when thresholding is enabled. Another application area is the filtering of well logs, where Wavelet-Kalman filter gives better results and control over the existing moving averages based filter when the filter is tuned by changing the process noise variance.

Although this study gives a deep understanding of the proposed method, further research is still necessary for observing the applicability of the method in different kind of problems of geosciences, like satellite orbit predictions, satellite attitude determination. It is also reasonable to expand proposed methodology into higher dimensions for adoption to wider applications areas. Real-time characteristics of the signal will make it useful for application to deformation monitoring of civil structures such as dams or bridges (installed in the accelerometer chip).

## REFERENCES

- Afet-İşleri (2007). <http://angora.deprem.gov.tr/depreminiome.htm>. Last accessed April 10, 2007.
- Alani, D., A. Averbuch, and S. Dekel (2007, January). Image coding with geometric wavelets. *IEEE Transactions On Image Processing* 16(1), 69–77.
- Brown, R. G. and P. Y. C. Hwang (1991). *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley Sons.
- Daubechies, I. (1988, November). Orthonormal basis of compactly supported wavelets. *Communications on Pure and Applied Mathematics* (41), 909–996.
- Debnath, L. and P. Mikusinski (1999). *Introduction to Hilbert Spaces with Applications*. Acedemic Press.
- Donoho, D. L. and I. M. Johnstone (1994). Ideal spatial adaptation via wavelet shrinkage. *Biometrika* (81), 425–455.
- Donoho, D. L. and I. M. Johnstone (1995). Adopting to unknown smoothness via wavelet shrinkage. *Journal of American Statistic. Ass.* (90), 1200–1224.
- Gabor, D. (1946). Theory of communications. *J. IEE London* 93(26), 429–457.
- Gelb, A. (1974). *Applied Optimal Estimation*. MIT Press.
- Geosig (2007). <http://www.geosig.com>. Last accessed April 10, 2007.
- Grival, M. S. and A. P. Andrews (2001). *Kalman Filtering Theory and Practice Using MATLAB*. John Willey and Sons.
- Grossman, A. and J. Morlet (1984). Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal Of Mathematical Analysis* 15, 723.
- Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematical Analysis* (69), 331–371.
- Hong, L., G. Chen, and C. K. Chui (1998, February). A filter-bank-based kalman filtering technique for wavelet estimation and decomposition of random signals. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS* 45(2), 237–241.
- JAMA (2007). <http://math.nist.gov/javanumerics/jama/>. Last accessed March 13, 2007.
- JFreeChart (2007). <http://www.jfree.org/jfreechart/>. Last accessed March 13, 2007.
- Jorden, J. R. and F. L. Campbell (1986). *Well Logging II - Electric and Acoustic Logging*. Society of Petroleum Engineers New York.

- Kalman, R. E. (1960, March). A new approach to linear prediction and filtering problems. *Transaction of the ASME, Journal of Basic Engineering*, 35–45.
- Koch, K.-R. (1999). *Parameter Estimation and Hypothesis Testing in Linear Model*. Springer-Verlag.
- Mallat, S. (1989). A theory of multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7), 674–693.
- Mallat, S. (1998). *A Wavelet Tour of Signal Processing*. Academic Press.
- Maybeck, P. S. (1979). *Stochastic models, estimation, and control*, Volume I. Academic Press.
- McEwen, J. D., M. P. Hobson, D. J. Mortlock, and A. N. Lasenby (2007, February). Fast directional continuous spherical wavelet transform algorithms. *IEEE Transactions on Signal Processing* 55(2), 520–529.
- Minkler, G. and J. Minkler (1993). *Theory and Application of Kalman Filtering*. Magellan Book Company.
- Petrou, M. and P. Bosdogianni (1999). *Image Processing The Fundamentals*. John Wiley and Sons.
- Simav, D. (1996). The wavelet technique and its applications to some problems in solid mechanics : Seismic data. Master's thesis, The Graduate School of Natural And Applied Sciences of METU.
- Therrien, C. W. (1992). *Discrete Random Signals and Statistical Signal Processing*. Prentice Hall.
- Welch, G. and G. Bishop (2001, August). Course 8: An introduction to the kalman filter. *SIGGRAPH*.
- Wiener, N. (1964, March). *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. MIT Press.

## APPENDIX A

### THE WAVELET-KALMAN EQUATIONS FOR 8-BLOCK

For three level wavelet-kalman filter a block of size eight is needed. The block equations and variance covariance matrices for blocksize of eight are given below, which are calculated similar to the blocksize of four calculations in Chapter 4.

#### A.1 Process noise covariance matrix $Q$

The process noise covariance matrix  $Q$  is given elementwise below.

$$Q_{11} = \frac{1}{64}A^14Q + \frac{1}{16}A^12Q + \frac{9}{64}A^{10}Q + \frac{1}{4}A^8Q + \frac{25}{64}A^6Q + \frac{9}{16}A^4Q + \frac{49}{64}A^2Q + Q$$

$$Q_{12} = \frac{1}{28}A^13Q + \frac{3}{28}A^11Q + \frac{3}{14}A^9Q + \frac{5}{14}A^7Q + \frac{15}{28}A^5Q + \frac{3}{4}A^3Q + AQ$$

$$Q_{13} = \frac{1}{16}A^12Q + \frac{1}{6}A^10Q + \frac{15}{48}A^8Q + \frac{1}{2}A^6Q + \frac{35}{48}A^4Q + A^2Q$$

$$Q_{14} = \frac{1}{10}A^11Q + \frac{1}{4}A^9Q + \frac{18}{40}A^7Q + \frac{7}{10}A^5Q + A^3Q$$

$$Q_{15} = \frac{5}{32}A^10Q + \frac{3}{8}A^8Q + \frac{21}{32}A^6Q + A^4Q$$

$$Q_{16} = \frac{1}{4}A^9Q + \frac{7}{12}A^7Q + A^5Q$$

$$Q_{17} = \frac{7}{16}A^8Q + A^6Q$$

$$Q_{18} = A^7Q$$

$$Q_{21} = Q_{12}$$

$$Q_{22} = \frac{1}{49}A^14Q + \frac{4}{49}A^12Q + \frac{9}{49}A^{10}Q + \frac{16}{49}A^8Q + \frac{25}{49}A^6Q + \frac{36}{49}A^4Q + A^2Q + AQ$$

$$Q_{23} = \frac{1}{21}A^13Q + \frac{1}{7}A^11Q + \frac{2}{7}A^9Q + \frac{10}{21}A^7Q + \frac{5}{7}A^5Q + A^3Q + AQ$$

$$Q_{24} = \frac{3}{35}A^12Q + \frac{8}{35}A^10Q + \frac{3}{7}A^8Q + \frac{24}{35}A^6Q + A^4Q + A^2Q$$

$$Q_{25} = \frac{1}{7}A^11Q + \frac{10}{28}A^9Q + \frac{18}{28}A^7Q + A^5Q + A^3Q$$

$$Q_{26} = \frac{5}{21}A^10Q + \frac{4}{7}A^8Q + A^6Q + A^4Q$$

$$Q_{27} = \frac{3}{7}A^9Q + A^7Q + A^5Q$$

$$Q_{28} = A^8Q + A^6Q$$

$$Q_{31} = Q_{13}$$

$$Q_{32} = Q_{23}$$

$$Q_{33} = \frac{1}{36}A^14Q + \frac{1}{9}A^12Q + \frac{1}{4}A^10Q + \frac{4}{9}A^8Q + \frac{25}{36}A^6Q + A^4Q + A^2Q + Q$$

$$Q_{34} = \frac{1}{15}A^13Q + \frac{1}{5}A^11Q + \frac{2}{5}A^9Q + \frac{2}{3}A^7Q + A^3Q + AQ$$

$$Q_{35} = \frac{3}{24}A^12Q + \frac{1}{3}A^10Q + \frac{5}{8}A^8Q + A^6Q + A^4Q + A^2Q$$

$$Q_{36} = \frac{2}{9}A^11Q + \frac{10}{18}A^9Q + A^7Q + A^5Q + A^3Q$$

$$Q_{37} = \frac{5}{12}A^10Q + A^8Q + A^6Q + A^4Q$$

$$Q_{38} = A^9Q + A^7Q + A^5Q$$

$$Q_{41} = Q_{14}$$

$$Q_{42} = Q_{24}$$

$$Q_{43} = Q_{34}$$

$$Q_{44} = \frac{1}{25}A^14Q + \frac{4}{25}A^12Q + \frac{9}{25}A^10Q + \frac{16}{25}A^8Q + A^6Q + A^4Q + A^2Q + Q$$

$$Q_{45} = \frac{20}{20}A^13Q + \frac{6}{20}A^11Q + \frac{3}{5}A^9Q + A^7Q + A^5Q + A^3Q + AQ$$

$$Q_{46} = \frac{1}{5}A^12Q + \frac{8}{15}A^10Q + A^8Q + A^6Q + A^4Q + A^2Q$$

$$Q_{47} = \frac{2}{5}A^11Q + A^9Q + A^7Q + A^5Q + A^3Q$$

$$Q_{48} = A^10Q + A^8Q + A^6Q + A^4Q$$

$$Q_{51} = Q_{15}$$

$$Q_{52} = Q_{25}$$

$$Q_{53} = Q_{35}$$

$$Q_{54} = Q_{45}$$

$$Q_{55} = \frac{1}{16}A^14Q + \frac{1}{4}A^12Q + \frac{9}{16}A^10Q + A^8Q + A^6Q + A^4Q + A^2Q + Q$$

$$Q_{56} = \frac{1}{6}A^13Q + \frac{1}{2}A^11Q + A^9Q + A^7Q + A^5Q + A^3Q + AQ$$

$$Q_{57} = \frac{3}{8}A^12Q + A^10Q + A^8Q + A^6Q + A^4Q + A^2Q$$

$$Q_{58} = A^11Q + A^9Q + A^7Q + A^5Q + A^3Q$$

$$Q_{61} = Q_{16}$$

$$Q_{62} = Q_{26}$$

$$Q_{63} = Q_{36}$$

$$Q_{64} = Q_{46}$$

$$Q_{65} = Q_{56}$$

$$Q_{66} = \frac{1}{9}A^14Q + \frac{4}{9}A^12Q + A^10Q + A^8Q + A^6Q + A^4Q + A^2Q + Q$$

$$Q_{67} = \frac{1}{3}A^13Q + A^11Q + A^9Q + A^7Q + A^5Q + A^3Q + AQ$$

$$Q_{68} = A^12Q + A^10Q + A^8Q + A^6Q + A^4Q + A^2Q$$

$$Q_{71} = Q_{17}$$

$$Q_{72} = Q_{27}$$

$$Q_{73} = Q_{37}$$

$$Q_{74} = Q_{47}$$

$$Q_{75} = Q_{57}$$

$$Q_{76} = Q_{67}$$

$$Q_{77} = \frac{1}{4}A^14Q + A^12Q + A^10Q + A^8Q + A^6Q + A^4Q + A^2Q + Q$$

$$Q_{78} = A^13Q + A^11Q + A^9Q + A^7Q + A^5Q + A^3Q + AQ$$

$$Q_{81} = Q_{18}$$

$$Q_{82} = Q_{28}$$

$$Q_{83} = Q_{38}$$

$$Q_{84} = Q_{48}$$

$$Q_{85} = Q_{58}$$

$$Q_{86} = Q_{68}$$

$$Q_{87} = Q_{78}$$

$$Q_{88} = A^14Q + A^12Q + A^10Q + A^8Q + A^6Q + A^4Q + A^2Q + Q$$

## A.2 The System linearmodel matrix A

The associated  $A$  matrix is given below

$$A = \begin{bmatrix} \frac{1}{8}A^8 & \frac{1}{8}A^7 & \frac{1}{8}A^6 & \frac{1}{8}A^5 & \frac{1}{8}A^4 & \frac{1}{8}A^3 & \frac{1}{8}A^2 & \frac{1}{8}A \\ 0 & \frac{1}{7}A^8 & \frac{1}{7}A^7 & \frac{1}{7}A^6 & \frac{1}{7}A^5 & \frac{1}{7}A^4 & \frac{1}{7}A^3 & \frac{1}{7}A^2 \\ 0 & 0 & \frac{1}{6}A^8 & \frac{1}{6}A^7 & \frac{1}{6}A^6 & \frac{1}{6}A^5 & \frac{1}{6}A^4 & \frac{1}{6}A^3 \\ 0 & 0 & 0 & \frac{1}{5}A^8 & \frac{1}{5}A^7 & \frac{1}{5}A^6 & \frac{1}{5}A^5 & \frac{1}{5}A^4 \\ 0 & 0 & 0 & 0 & \frac{1}{4}A^8 & \frac{1}{4}A^7 & \frac{1}{4}A^6 & \frac{1}{4}A^5 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3}A^8 & \frac{1}{3}A^7 & \frac{1}{3}A^6 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}A^8 & \frac{1}{2}A^7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A^8 \end{bmatrix}$$

Note that both in process noise covariance matrix  $Q$  and the system linear model matrix  $A$  has a pattern similar to the matrices computed in Chapter 4. A generalized  $Q$  and  $A$  matrix can be calculated by utilizing this pattern. The following code segments show how this matrices can be computed in Java programming language.

The following code segment lists how the covariance matrix  $Q$  can be represented as a class in Java programming language.

```
package edu.metu.ggit.signal.torpu.kalman;

import Jama.Matrix;

public class GeneralizedQMatrix extends Matrix {
    //level of decomposition
    private int level;

    public GeneralizedQMatrix(int level,double a,double q) {
        super(1<<level,1<<level);
        this.level = level;
        init(a,q);
    }

    //initialize the Q matrix according to fixed value
    //a and process noise variance q
    //use the symmetry property of the covariance matrix.
    private void init(double a, double q) {
        int nData=1<<level;
        int colCount = nData+nData-1;
        Matrix D = new Matrix(nData,colCount);

        for (int row = 0; row < nData; row++) {
            for (int col=nData-row-1;col<colCount-row;col++) {
                if (col == nData-row-1) {
                    D.set(row, col, 1);
                }else if ( col <= nData-1){
                    D.set(row, col, Math.pow(a, col-(nData-row-1)));
                } else {
                    double coef = (double)(2*nData-row-col-1)/(nData-row);
                    if (2*nData-row-col-1 <1) coef=1.0/(nData-row);
                    D.set(row, col, coef*Math.pow(a, col-(nData-row-1)));
                }
            }
        }
    }
}
```

```
}  
Matrix X = D.times(q).times(D.transpose());  
setMatrix(0, nData-1, 0, nData-1, X);  
}  
  
}
```

The code segment below shows how the generic form of the  $A$  matrix can be represented as a class in Java programming language.

```
package edu.metu.ggit.signal.torpu.kalman;

import Jama.Matrix;

public class BlockAMatrix extends Matrix {

    public BlockAMatrix(int level, double a) {
        super(1<<level, 1<<level);
        int nDataBlock = 1<<level;
        for (int k = 0; k < nDataBlock; k++) {
            for (int i = k; i < nDataBlock; i++) {
                double ap = Math.pow(a, nDataBlock * i + k);
                set(k, i, ap / (nDataBlock * k));
            }
        }
    }

    public static void main(String[] args) {
        new BlockAMatrix(2, 1).print(1, 4);
    }
}
```

## APPENDIX B

### SOFTWARE

#### B.1 Definition

The wavelet-Kalman filter is implemented as a software program with a graphical user interface in Java programming language. The software is called “Törpu” which is the Turkish word for raspa. The software also includes conversion routines for seismic data and well log data. An image denoising method based on Wavelet-Kalman filter also included for investigating the applicability of the algorithm to two dimensions.

#### B.2 Design and Algorithms

The software is designed in an extensible way to make it easy to adopt new filters and wavelets into the Wavelet-Kalman Filter and the main Graphical User Interface (GUI). Kalman and Wavelet-Kalman filters are designed to be classes that use interfaces to read the input signal and write the output signal. Signals are modeled by interface classes to enable easy adoption of existing signals.

The external packages used in the software is the JAMA(Java Matrix Algebra package) which contains the Matrix class and other utility functions needed to implement the Kalman Filters (JAMA, 2007). And JFreeChart is used to present the signal and filter results in graphs. The graphs in this study are generated by JFreeChart (JFreeChart, 2007).

IWavelet interface is the base interface for all WaveletFunction implementations. AbstractWavelet implements the matrix representation of the wavelet transform in a generic way. Figure B.1 shows the UML diagram for wavelets.

Wavelet-Kalman filter and Kalman Filter implements the base interface IKalmanFilter which defines the necessary operation for running a Kalman Filter. Figure B.2 shows the UML diagram of Wavelet-Kalman filter and Kalman Filter and their relations.

The Wavelet-Kalman filter algorithm is given in Algorithm 2. and Some sample screenshots from the software called Törpü is listed in Figures B.3 and B.4.

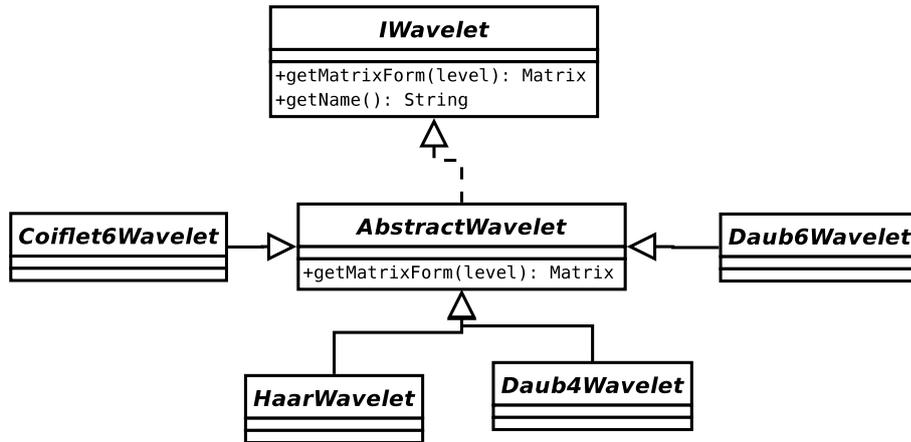


Figure B.1: Iwavelet Interface and its implementing Wavelet classes. getMatrixForm() function returns the matrix form of the associated wavelet.

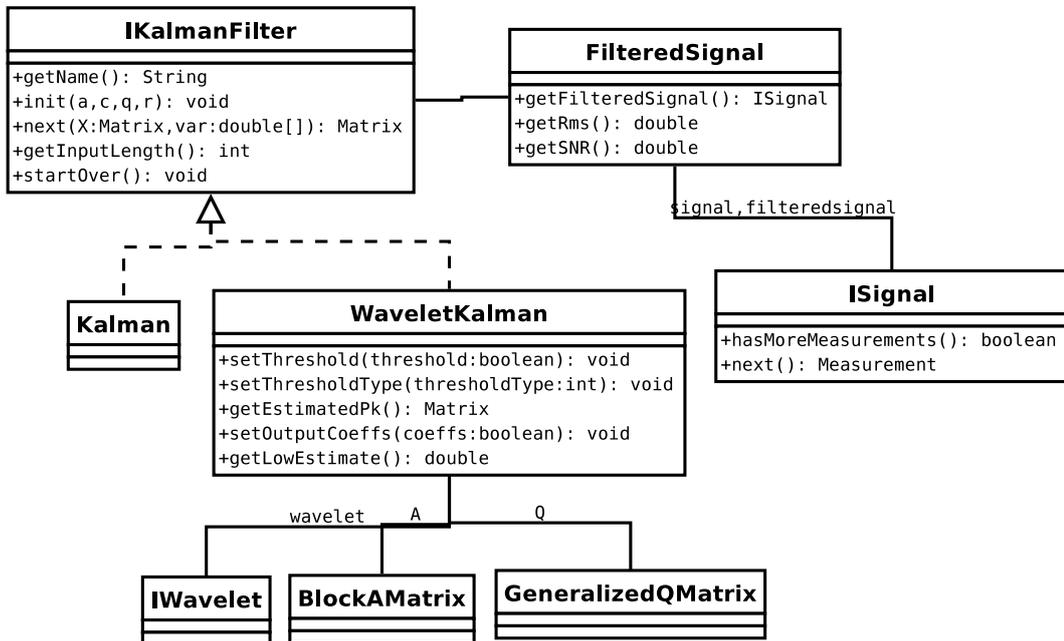


Figure B.2: IKalmanFilter Interface and its implementing Wavelet-Kalman and Kalman Filters.

---

**Algorithm 2** Wavelet-Kalman Filter

---

```
1:  $\mathbf{T} \leftarrow$  wavelet transform matrix
2:  $\mathbf{A} \leftarrow$  BlockAMatrix()
3:  $\mathbf{Q} \leftarrow$  GeneralizedQMatrix()
4:  $\mathbf{A} \leftarrow \mathbf{TAT}^T$ 
5:  $\mathbf{Q} \leftarrow \mathbf{TQT}^T$ 
6:  $\mathbf{C} \leftarrow \mathbf{TCT}^T$ 
7:  $\mathbf{P}_k^- \leftarrow \mathbf{P}_0, \mathbf{x}_k^- \leftarrow \mathbf{T}\mathbf{x}_0$  //  $\mathbf{x}_0$  is the a priori block estimate at  $k$ 
8:  $k \leftarrow 0$ 
9: while measurement is collected in blocks do
10:    $\mathbf{R}_k \leftarrow$  variance of new measurement block
11:    $\mathbf{z}_k \leftarrow$  measurement block
12:    $\mathbf{z}_k \leftarrow \mathbf{T}\mathbf{z}_k$  // Transform measurements into wavelet domain.
13:    $\mathbf{R}_k \leftarrow \mathbf{T}\mathbf{R}_k\mathbf{T}^T$  // Transform the measurement variance into wavelet domain
14:    $\mathbf{K}_k \leftarrow \mathbf{P}_k^- \mathbf{C}^T (\mathbf{C}\mathbf{P}_k^- \mathbf{C}^T + \mathbf{R}_k)^{-1}$  // Find Kalman gain matrix  $\mathbf{K}_k$ 
15:    $\mathbf{P}_k \leftarrow (\mathbf{I} - \mathbf{K}_k \mathbf{C}) \mathbf{P}_k^-$ 
16:    $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_k^-)$  // Apply correction to the predicted state variables by
   taking the measurements into account
17:   if thresholding enabled then
18:     applyThreshold( $\hat{\mathbf{x}}_k$ )
19:    $\mathbf{P}_{k+1}^- \leftarrow \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{Q}$  // Predict Future error covariance of the state matrix
20:    $\hat{\mathbf{x}}_{k+1}^- \leftarrow \mathbf{A}\hat{\mathbf{x}}_k$  // Predict future version of  $\hat{\mathbf{x}}_k$ 
21:   if output coefficients then
22:     output  $\hat{\mathbf{x}}_k$ 
23:   else
24:     output  $\mathbf{T}^T \hat{\mathbf{x}}_k$  // Output the estimated state in system state variable domain
25:    $k \leftarrow k + 1$ 
```

---

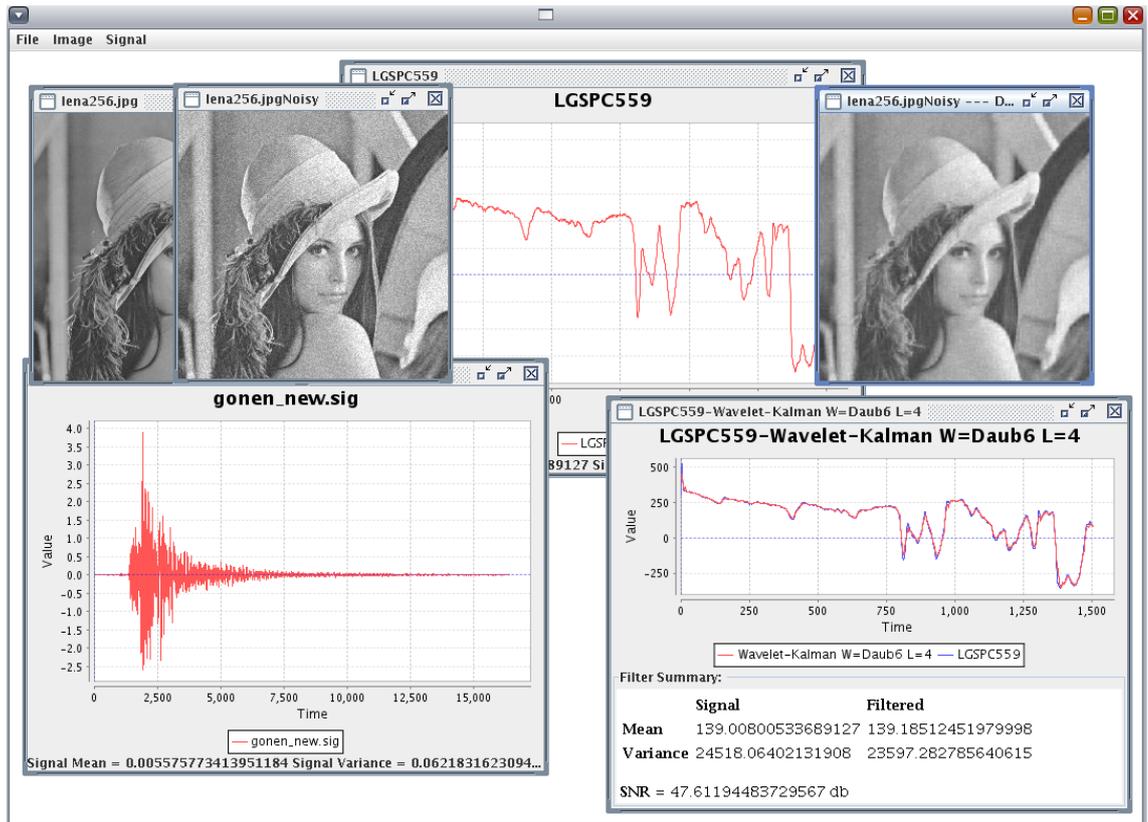


Figure B.3: Shows a screenshot of the software with seismic signal, well log and images.

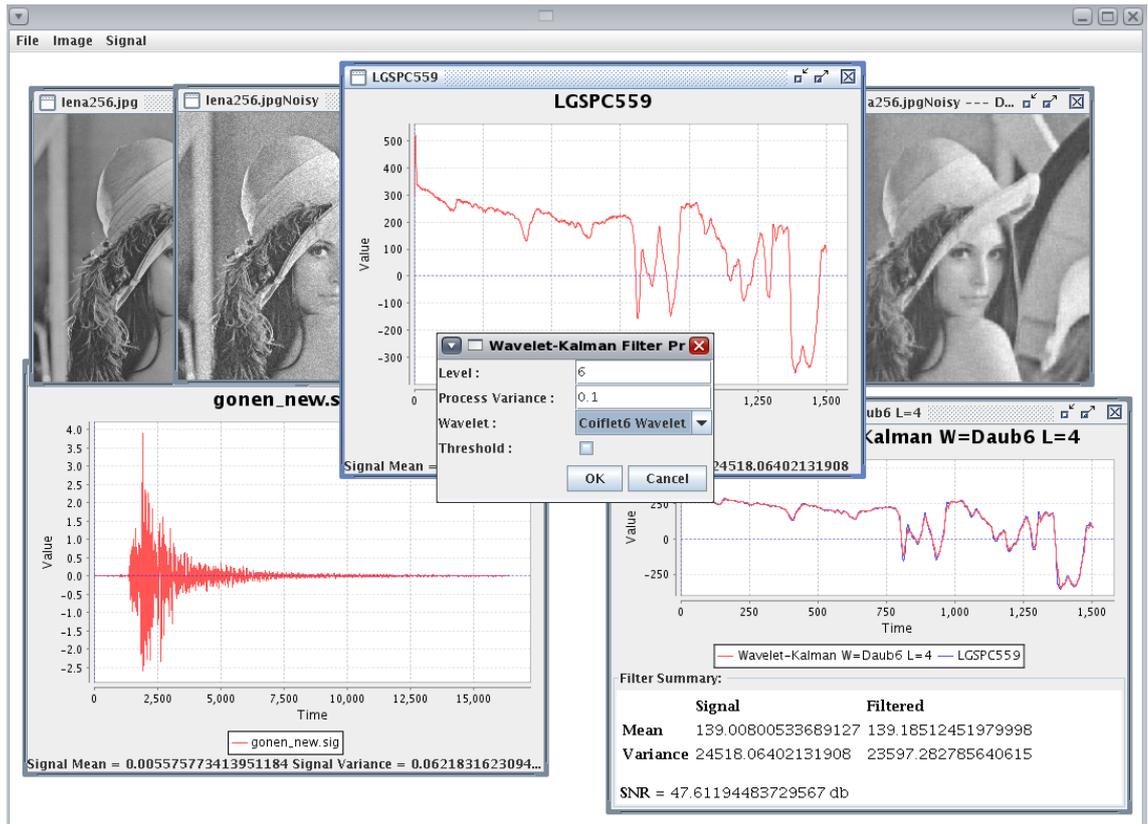


Figure B.4: Shows a screenshot of the software with seismic signal, well log and images with the parameter dialog for wavelet-kalman filter.