LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
FACULTY OF TECHNOLOGY
DEPARTMENT OF TECHNOMATHEMATICS AND PHYSICS

Kalman Filtering in Online Paper Quality Assessment

The topic for this thesis was approved by the faculty of technology on June, 2009.

Supervisors: Dr. Heikki Haario
Examiners: Dr. Tuomo Kauranne

Lappeenranta 22 June 2009

_____

Ville Manninen

Abstract

Lappeenranta University of Technology

Faculty of Technology

Department of Technomathematics and Physics

Ville Manninen

Kalman Filtering in Online Paper Quality Assessment

Thesis for the Degree of Master of Science in Technomathematics and Physics

2009

52 pages, 21 figures and 5 tables

Examiners: Doctor of philosophy, Heikki Haario

Doctor of philosophy, Tuomo Kauranne

Keywords: Kalman filter, Fourier transform, data assimilation, mathematical modelling, quality control system, quality variable

Controlling the quality variables (such as basis weight, moisture etc.) is a vital part of making top quality paper or board. In this thesis, an advanced data assimilation tool is applied to the quality control system (QCS) of a paper or board machine. The functionality of the QCS is based on quality observations that are measured with a traversing scanner making a zigzag path.

The basic idea is the following: The measured quality variable has to be separated into its machine direction (MD) and cross direction (CD) variations due to the fact that the QCS works separately in MD and CD. Traditionally this is done simply by assuming one scan of the zigzag path to be the CD profile and its mean value to be one point of the MD trend. In this thesis, a more advanced method is introduced.

The fundamental idea is to use the signals' frequency components to represent the variation in both CD and MD. To be able to get to the frequency domain, the Fourier transform is utilized. The frequency domain, that is, the Fourier components are then used as a state vector in a Kalman filter. The Kalman filter is a widely used data assimilation tool to combine noisy observations with a model. The observations here refer to the quality measurements and the model to the Fourier frequency components.

By implementing the two dimensional Fourier transform into the Kalman filter, we get an advanced tool for the separation of CD and MD components in total variation or, to be more general, for data assimilation. A piece of a paper roll is analyzed and this tool is applied to model the dataset. As a result, it is clear that the Kalman filter algorithm is able to reconstruct the main features of the dataset from a zigzag path. Although the results are made with a very short sample of paper roll, it seems that this method has great potential to be used later on as a part of the quality control system.

Tiivistelmä

Lappeenrannan teknillinen yliopisto

Tekniikan tiedekunta

Matematiikan ja fysiikan laitos

Ville Manninen

Kalman Filtering in Online Paper Quality Assessment

Diplomityö, 2009

52 sivua, 21 kuvaa ja 5 taulukkoa.

Tarkastajat: Tuomo Kauranne, Heikki Haario

Hakusanat: Kalman suodin, Fourier-muunnos, data-assimilaatio, matemaattinen mallinnus, laatusuure, laatusäätöjärjestelmä

Paperi- tai kartonkikoneen laatusäätöjärjestelmä on tärkeä osa hyvälaatuisen paperin tai kartongin valmistusprosessia. Laatusäätöjärjestelmä säätää tärkeimpiä suureita, kuten ominaispainoa, kosteutta jne., perustuen mittausdataan. Mittausdataa keräävät radan yli poikkisuuntaan sahaavat skannerit, muodostaen näin siksak-tyyppisen polun. Laatusäätöjärjestelmä koostuu kahdesta osasta jotka säätävät paperirainaa erikseen poikki- ja konesuunnassa. Siksak- tyyppisestä mittausdatasta on näin ollen kyettävä erottelemaan poikki- ja konesuuntaiset komponentit.

Nykyisin tämä erottelu tehdään yksinkertaisesti ottamalla yksi poikkisuuntainen skannaus vastaamaan poikkisuuntaista profiilia, ja tämän keskiarvo vastaamaan yhtä pistettä konesuuntaisessa profiilissa. Ongelmaksi muodostuu se, että paperirainan kulkiessa kovaa vauhtia yksi skannaus sisältää myös paljon konesuuntaista vaihtelua.

Kalman suodin on tapa, jolla kohinainen ja mahdollisesti harvaan mitattu mittausdata saadaan tilastollisessa mielessä optimaalisesti yhdistettyä prosessia kuvaavan mallin kanssa. Mittausdatana tässä työssä on kartonkikoneesta saatava ominaispainodata. Ominaispainodataa mallinnetaan kaksiulotteisella Fourier muunnoksella, jonka taajuuskertoimia käytetään Kalman suotimessa tilavektorina.

Tässä työssä esitellään uuden tyyppinen algoritmi kone- ja poikkisuuntaisen vaihtelun erottamiseen mittausdatasta. Algoritmi perustuu Kalman suotimen sekä Fourier muunnoksen yhdistämiseen laatusuureiden vaihtelun mallinnuksessa. Tekemällä mittausdatalle kaksiulotteinen Fourier muunnos, saadaan suoraan sekä kone- että poikkisuuntaiset taajuuskomponentit. Näitä komponentteja käytetään Kalman suotimen tilavektorina.

Tällä kaksiulotteiseen Fourier muunnokseen pohjautuvalla Kalman suotimella on tässä työssä analysoitu aitoa, mutta suhteellisen lyhyttä pätkää kartonkikoneesta saatua paperirainaa. Tulosten perusteella voidaan todeta että kehitetty algoritmi kykenee erottelemaan mittausdatasta aitoja poikki- ja konesuuntaisia vaihteluja. Vaikka datajoukko onkin suhteellisen pieni, on tässä algoritmissa potentiaalia olla tulevaisuudessa osa paperi- tai kartonkikoneen laatusäätöjärjestelmää.

Acknowledgements

Firstly, I want thank Matti Heiliö and Merja Mäkelä, without them this thesis would not have been started in the first place. Merja has also been a major editorial help, and an expert concerning the paper industry.

I want express my appreciation for Tuomo Kauranne who initially came up with the fundamental idea of this thesis and also helped greatly with the technical and editorial issues.

This thesis has been carried out as a part of NoTes project, funded by Tekes. Thanks to Risto Ritala for his guidance during this time.

Also thanks to my fellow colleagues Piotr Ptak and Miika Tolonen for their support.

Respect to myself for being able to finally finish this. J

Contents

Abbreviations

2D	2-Dimensional
3D	3-Dimensional
ATPA	Advanced Total Paper Analyzer (paper analyzing device by Stora Enso)
ARMA	Auto-Regressive Moving Average
CD	Cross Direction
CFT	Continuous Fourier Transform
DFT	Discrete Fourier Transform
EKF	Extended Kalman Filter
FFT	Fast Fourier Transform
FT	Fourier Transform
LQG	Linear Quadratic Gaussian
MD	Machine Direction
QCS	Quality Control System

Symbols

a,b	Coefficients of the Fourier series
c	Complex coefficient of the Fourier series
f	Real-valued periodic function
f_j	Discrete signal vector
F_k	Discrete complex Fourier components system
N,M	Length of a vector or size of a matrix
W	Fourier transform matrix
w_N	Element in the Fourier matrix
$freq$	Frequency
wl	Wavelength
t,i,j,k	(time) Indices
x	Variable in Fourier transform or state vector on Kalman filter
E	Expectation value
H	Measurement sensitivity matrix (Kalman filter)
A	Model matrix (Kalman filter)
K	Kalman gain (Kalman filter)
R	Measurement covariance matrix (Kalman filter)
Q	Process covariance matrix (Kalman filter)
v,w	Gaussian noise parameters (Kalman filter)
B	Matrix for the fixed-lag Kalman smoother
y	Measurement data
P	Error covariance matrix (Kalman filter)
M	Weighting matrix

1 Introduction

Making paper, in general, is a very complex process. It is very hard, if not impossible, to model or analyze the whole process from pulp to a final product. In this thesis we concentrate on a part at the end of this process, controlling the quality of the final product. The type of the final product can vary from regular office paper or high class magazine paper to different kinds of board paper types. In order to control the quality of the final product, a quality control system (QCS) is introduced.

In a paper or board machine, the direction that paper is moving towards is called the machine direction (MD), and the perpendicular direction the cross direction (CD). QCS of a paper or board machine is based on observations directly measured from the product. The scanner providing those observations moves only in CD, thus making a zigzag measurement path while travelling back and forth. The most significant types of observations the scanner measures are called quality variables, such as basis weight, moisture, caliper etc. The term QCS refers to these variables, which have the most effect on the quality of the paper or board. /13/

For QCS in a paper or board machine, it is important to separate the measurements into CD and MD components; the CD and MD variations are controlled differently. In a paper or board machine, the CD/MD separation is done usually in a way that one scan of a traversing scanner is considered as a single CD profile and a mean value of that is considered as one point of an MD trend. The ultimate goal of this thesis is to introduce, define, develop and test a CD/MD separation method which is based on an advanced mathematical model.

The problem of the CD/MD separation is old, and several methods have been developed over the years, see for example /4, 9, 10, 11/. The measurement data of the traversing scanner in a paper or board machine is neither cross nor machine directional and that makes the impact of the control actions hard to predict.

The CD/MD separation method introduced in this thesis is based on a Kalman filter. A Kalman filter has been previously used for the separation and analysis of quality variables /4/. However, the Kalman filter used in this thesis is based on the Fourier transform in a way that we use the Fourier components in both CD and MD as a state vector in the Kalman filter. The Fourier components represent the quality variable variation in CD and MD. The idea of using Fourier components as a basis of the Kalman filter in a separation method is novel.

On the mathematical point of view, we are defining a data assimilation method. We mathematically model the 2-dimensional quality variable dataset based on sparse observations. A tool for analysing a 2D dataset is introduced.

The structure of this thesis is the following: In the second chapter the discrete Fourier transform is introduced from a theoretical point of view and the formulas for doing the transform are defined. The Fourier transform is applied to the CD/MD separation of the variation in chapter three. In the fourth chapter, the Kalman filter is introduced and Kalman equations are derived. Then, the way of using the Kalman equations in paper web analysis is introduced in the chapter 5. Finally the results of this data assimilation method with the Kalman filter are shown in the sixth chapter.

The following Figure 1.1 shows the big picture of the QCS loop with CD/MD separation. The scanner measures the quality measurement data *y*, which is sent to the CD/MD separation model. The CD/MD separation model returns the estimates for CD and MD variations. Those variations are sent to the control model, separately for CD and MD. The control model then controls the ongoing product based on the estimates.

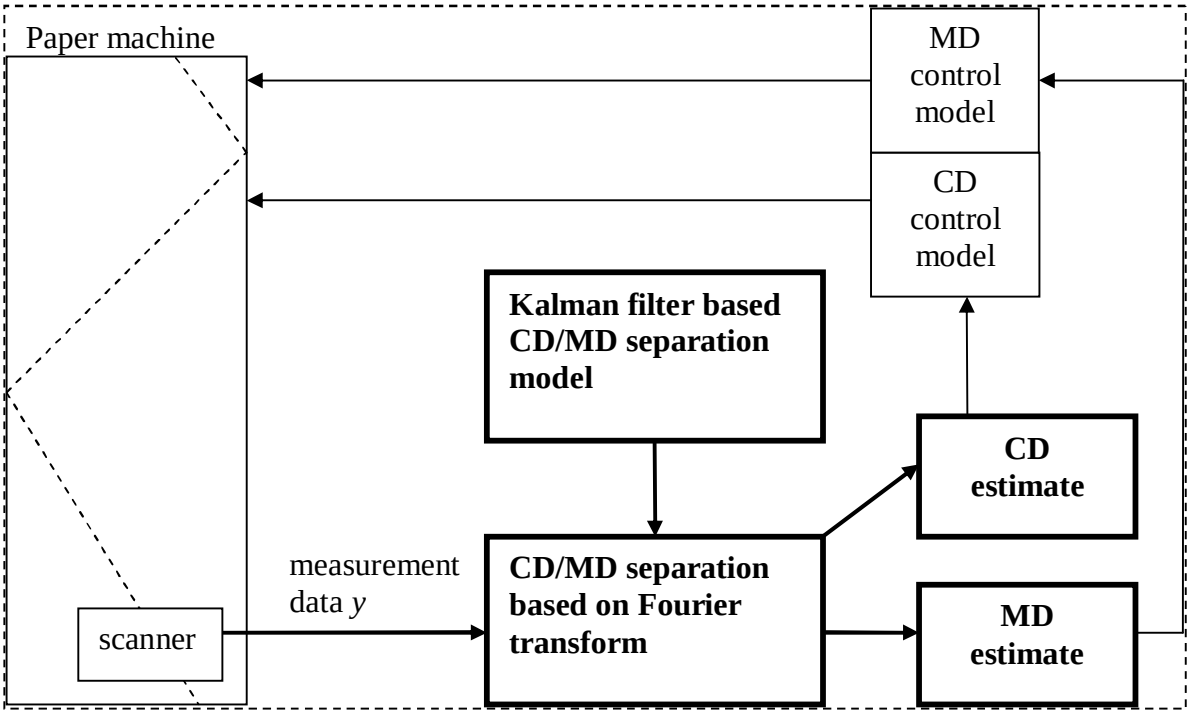


Figure 1.1: The big picture of how the CD/MD separation is related to the control systems in a paper or board machine.

2 Basics of the Discrete Fourier Transform (DFT)

Fourier series and Fourier transform are named after the French mathematician and physicist Jean Baptiste Joseph Fourier (1768-1830). Fourier came up with an idea that any function can be approximated arbitrary well by a series of sine and cosine functions. The Fourier series is a way to break an arbitrary periodic function into simple terms, solve the initial problem and then reconstruct the function. This idea has long been widely used as a helpful tool in at least signal and image processing, acoustics and data compression. /7,8/ The nature of the quality variable data is periodical (as to be seen in the forthcoming chapters), which makes the usage of the Fourier series reasonable.

Let us begin by introducing the idea of the Fourier series: Let $f(x)$ be any real-valued periodic function with a period of $2L$. The following equation is called the Fourier series of such a function

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos \frac{n\pi}{L} x + b_n \sin \frac{n\pi}{L} x), \qquad \textbf{2.1}$$

where

$$b_n = \frac{1}{L} \int_{-L}^{L} f(x) \sin \frac{n\pi}{L} x dx,$$

$n = 1, 2, 3, ...,$

$x, L \in \Re, L \neq 0.$

Now when n goes towards infinity, the Fourier series approximates arbitrary closely the original function f. The larger n grows, the more accurate approximation we are defining.

By applying the well known Euler equations

$$e^{inx} = \cos nx + i \sin nx$$
$$e^{-inx} = \cos nx - i \sin nx$$

to the Fourier series equation 2.1, the Fourier series can be expressed as a series of complex components

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{in\pi x / L}, \qquad \textbf{2.2}$$

where

$$c_n = \frac{1}{2L} \int_{-L}^{L} f(x) e^{-in\pi x / L} dx.$$

The following figure shows an example of the Fourier series; at first, a periodic wave has been generated, that is, the sawtooth wave. Then a few first components of the Fourier series have been calculated and plotted as a red curve. After this, a few more of the components are calculated and plotted as the green, and similarly the black curve with quite many components already. It is easy to see how the Fourier representation approaches the original function as the number of components increases.

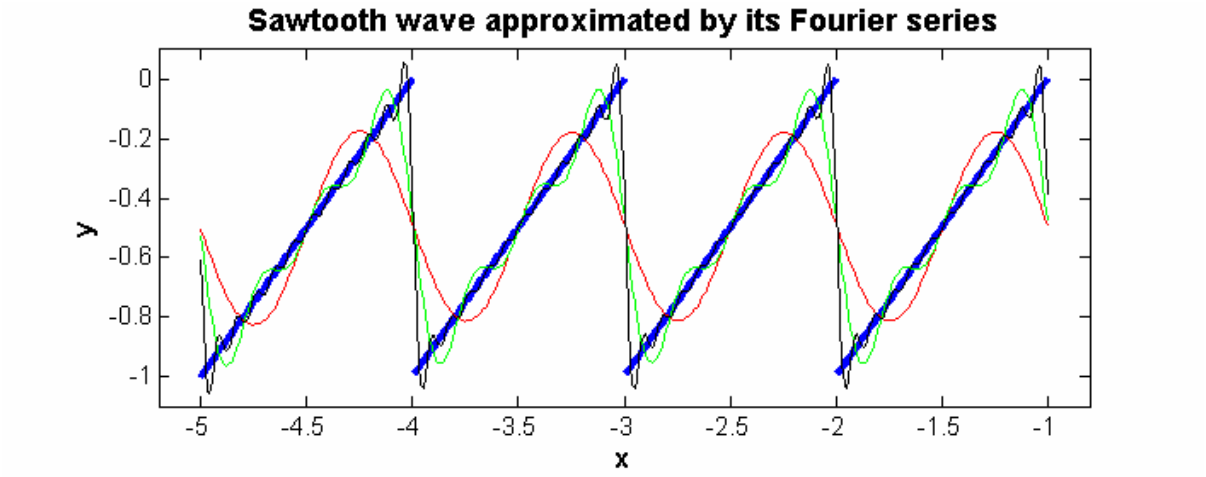


Figure 2.1: The Fourier series as an approximation of a periodic function: A sawtooth wave (blue) is approximated with only a few (red), several (green), and a lot (black) of the components of the Fourier series

2.1 1-Dimensional Discrete Fourier Transform

The Fourier transform is a generalization of the complex Fourier series. It is used to find what kind of periodicity exists in a given signal. The discrete Fourier transform (DFT) means transforming a signal into its frequency domain. Similarly, the inverse DFT means transforming the frequency domain back to a signal. The advantage of using the DFT is that

it provides an easy method of finding and measuring the periodicity in a given signal. In real cases, a signal to be transformed is almost always discrete. In our case, the quality variables are measured in a paper or board machine from a discrete number of time steps. That is why we are concentrating on the discrete Fourier transform instead of the continuous one. /7, 8/

The CFT (Continuous Fourier Transform) can be derived from the complex Fourier series equation 2.2. The DFT is a result of the discretization of the CFT. The DFT to a real discrete function f_j is defined

$$F_k = \sum_{j=0}^{N-1} f_j e^{-2\pi i k j / N}, \qquad \textbf{2.3}$$

where

f_j = a discrete signal vector
N = the length of f_j
F_k = a vector of the Fourier components.

Now F_k is a discrete complex vector with a length N. Each component in F represents a periodicity in the signal having a frequency of k/N. From now on we call these periodicities waves of different frequencies. F_k is usually represented by the amplitude of the waves, that is $F_p = |F_k|$. F_p is called the power spectrum of F_k. /7, 8/

To get back to the signal vector, the Inverse DFT to a complex vector F_k is defined

$$f_j = \frac{1}{N} \sum_{k=0}^{N-1} F_k e^{2\pi i j k / N}. \qquad \textbf{2.4}$$

The DFT can also be calculated as a matrix operation (due to the fact that it is a linear operator). Let's define the Fourier transform matrix

$$W = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N^{1\cdot 1} & w_N^{1\cdot 2} & \cdots & w_N^{1\cdot(N-1)} \\ 1 & w_N^{2\cdot 1} & w_N^{2\cdot 2} & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \\ 1 & w_N^{(N-1)\cdot 1} & & & w_N^{(N-1)\cdot(N-1)} \end{bmatrix}, \qquad \textbf{2.5}$$

where

$$w_N = e^{-2\pi i / N}.$$

Now we can redefine the DFT in matrix notation

$$F = Wf_j, \qquad \textbf{2.6}$$

where f_j is a (N x 1) signal vector.

Now F is a N x 1 complex vector each component representing one frequency in f. The inverse DFT can be calculated using the Fourier transform matrix W as

$$f_j = \frac{1}{N} W^T F \text{ . /8/} \qquad \textbf{2.7}$$

We can see from the matrix notation of the DFT, equation 2.6, that the resulting vector F has a real value as its first component, that is, actually just a sum of the signal vector f. Moreover, it can be seen that the second component is the complex conjugate of the last one, then the third is the complex conjugate of the second to last one and so on. This phenomenon is called the complex conjugate phenomenon. It means in practice that only one half of the Fourier components are needed to reconstruct the original signal completely.

MATLAB Fourier functions use the FFT (Fast Fourier Transform) algorithm to speed up the transform. Normally the DFT to a signal vector with a length N requires N^2 calculations. By taking into account the complex conjugate phenomenon, it reduces computational time to $\frac{1}{2}N^2$. That's not significant for big values of N. The FFT algorithm is required to speed the operation up. By rearranging the components in the Fourier matrices, the number of required calculations reduces to $N\log_2 N$. This is a significant benefit for large values of N. /6/

2.1.1 Phase Shift

In the discrete Fourier component space, a phase shift means shifting the Fourier coefficients a discrete number of time steps forward or backward. In other words, it means moving the analyzed function or dataset forward or backward. This is useful when analysing a longer dataset and, to reduce computational costs, we don't want our Fourier window to offset the whole set of data. By using a phase shift, we can make our Fourier window smaller and let the window move along the dataset as the analysis goes on. An example of a phase shift can be seen in chapter 2.1.2 and it is later on used in practice in chapters 5 and 6.

A phase shift to the Fourier coefficients is carried out in the following way: Let f_j be a discrete function with length N to which we have computed a DFT. Let F_k be the complex

Fourier coefficients of *f*, where *k* refers to the element with frequency *k/N*. If we shift the function f_j *a* steps backward, then the corresponding Fourier coefficient, f_{j-a}, is then given by $F_k e^{-2\pi ika/N}$. As we can now see, the phase shift does not change complex coefficients' modulus but just rotates them in the complex plane by $-2\pi ika$ radian units.

2.1.2 An Example of DFT and Phase Shift in MATLAB

The MATLAB function *fft* is based on FFT algorithm introduced in previous chapter and uses the following equation to carry out the discrete Fourier transform to a finite artificial real signal vector f_j, $j = 1,\ldots,N$:

$$F_k = \sum_{j=1}^{N} f_j w_N^{(j-1)(k-1)}, \; w_N = e^{(-2\pi i)/N} \qquad \textbf{2.8}$$

F_k is now a complex vector with length *N*. Each element *k* in *F* represents a frequency of *k/N* Hz in the signal. We can say that the signal vector *f* is separated into waves of different frequencies. In practice *F* is represented by the complex modulus of its components, that is, the power spectrum $F_p = |F_k|$. F_p is then a symmetric vector, its modulus being the same for a complex number and its conjugate.

MATLAB calculates the inverse discrete Fourier transform for *F* with the function *ifft* using the following equation:

$$f_j = \frac{1}{N} \sum_{k=1}^{N} F_k w_N^{-(j-1)(k-1)} \qquad \textbf{2.9}$$

.

The following figures show an example of the DFT and a phase shift. The figures have been produced in the following way:

- An artificial signal has been generated with a certain noise level (Figure 2.2).
- The DFT has been calculated to the signal and its power spectrum is shown (Figure 2.2).

- 15 strongest Fourier components have been selected from the power spectrum and the inverse DFT has been calculated to them. As a result, a Fourier filtered curve is shown (Figure 2.3).
- A phase shift by 40 units has been done to the selected Fourier components to show how the phase shift literally shifts the signal. (Figure 2.3).

Note: The signal presented in the Figure 2.2 is the cross direction profile and the signal vector used in the Figure 2.3 is a part of the machine direction profile from the sample dataset of a quality variable used in the forthcoming chapters.

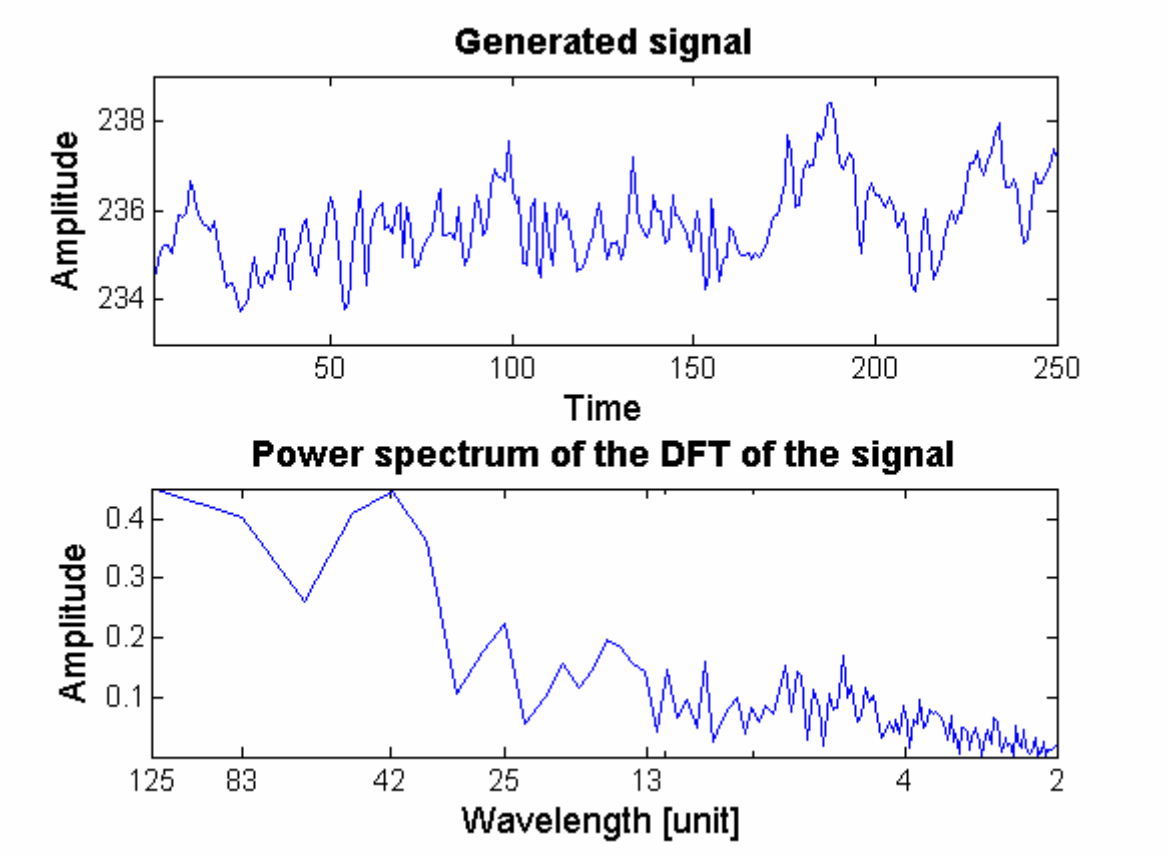


Figure 2.2: An example of the DFT. The upper figure is the original signal; the lower figure is the DFT of that signal, expressed by its power spectrum.

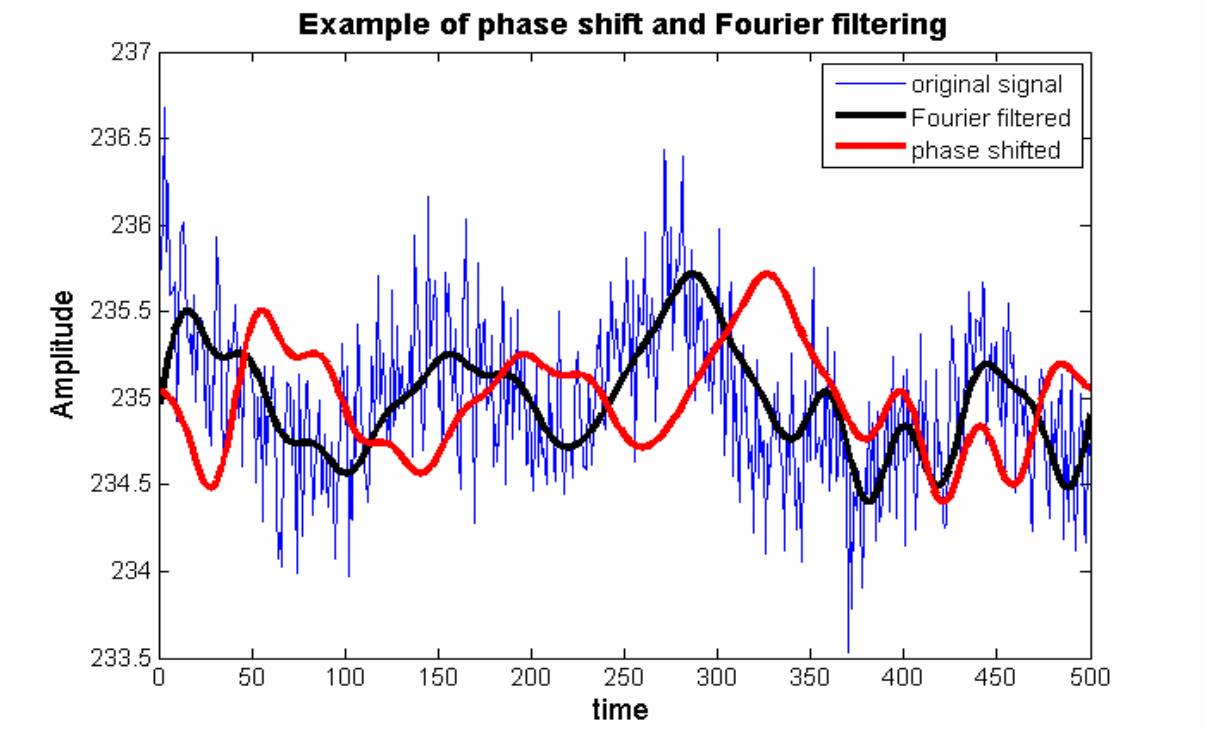


Figure 2.3: An example of Fourier filtering and a phase shift applied to a signal vector. The Blue signal is the original signal we are analysing. The Bold black line is an inverse transform of the 15 first Fourier components of the signal. The Red line is an inverse transform of the phase shifted Fourier components.

2.2 2-Dimensional Discrete Fourier Transform

In this chapter, the Fourier transform is extended to be 2-dimensional in order to analyze a 2D dataset. Fourier transforms used in the forthcoming chapters are all 2D although only certain components may have been taken into account. The 2D Fourier transform is very similar to the 1D one. In practice, the main difference is that by transforming a 2D dataset, the components in the frequency domain have also an angle in addition to the frequency. In theory, the Fourier transform being a linear operator, it's rather easy to extend it to 2D (and even more dimensions if needed).

The 2D Fourier transform to a 2D discrete function f with a size N x M is defined by the following equation

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) e^{-2\pi i \left(\frac{xu}{N} + \frac{yv}{M}\right)} . \qquad \textbf{2.10}$$

Now F is a matrix with a size of N x M each component representing a wave with certain frequency and angle in f. Let's consider a Fourier component having coordinates (x, y), $-N/2 < x < N/2, -M/2 < y < M/2$, in the frequency domain F. According to the Pythagorean Theorem the corresponding wave has then a frequency of

$$freq = \sqrt{\left(\frac{x}{N}\right)^2 + \left(\frac{y}{M}\right)^2} \qquad \textbf{2.11}$$

and similarly has a wavelength of

$$wl = \sqrt{\left(\frac{N}{x}\right)^2 + \left(\frac{M}{y}\right)^2} . \qquad \textbf{2.12}$$

This wave has the same angle as the vector (x, y) (Figure 2.5).

2D inverse discrete Fourier transform to a discrete complex Fourier component system F is carried out by the following equation

$$f(x,y) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} F(u,v) e^{2\pi i \left(\frac{xu}{N} + \frac{yv}{M}\right)} \qquad \textbf{2.13}$$

It is also possible to define a matrix notation of the 2D DFT. Let's define the DFT matrix W as in equation 2.5. Now the 2D DFT can be calculated by the following equation

$$f(x,y) = \frac{1}{NM}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} F(u,v)e^{2\pi i\left(\frac{xu}{N}+\frac{yv}{M}\right)} \qquad \textbf{2.14}$$

It is also possible to define a matrix notation of the 2D DFT. Let's define the DFT matrix W as in equation 2.5. Now the 2D DFT can be calculated from the following equation

$$F = (Wf)W \qquad \textbf{2.15}$$

and similarly the inverse 2D DFT by

$$f = \frac{1}{NM} W^T F W^T . \qquad \textbf{2.16}$$

2.2.1 An example of 2D DFT in MATLAB

MATLAB uses two inner fft – functions (Chapter 2.1.2) to calculate the 2D DFT, that is, calculating the FFT along the *x*- and the *y*-direction recursively. That generates the same result as equations 2.10 and 2.14 in the previous chapter. The idea of this chapter is to show how the 2D DFT works in practice. A sample dataset is created, it is Fourier transformed and the 2D power spectrum has been calculated (Figure 2.4). Then a few components have been extracted from the Fourier component matrix and an inverse transform has been calculated from those particular components only (Figure 2.5). The components are chosen in a way that they represent certain diagonal waves. Now, looking at the left figure in Figure 2.4, it is much harder to detect the presence of the diagonal waves.

In the 2D DFT power spectrum figures, the origin is in the middle. This is a common way of showing the power spectrum, the other way having the origin in the left bottom corner. In MATLAB, the fftshift function has been used to shift the Fourier components in that way.

Note: The 2D signal used here is half of the quality variable sample dataset introduced in the forthcoming chapter 3.

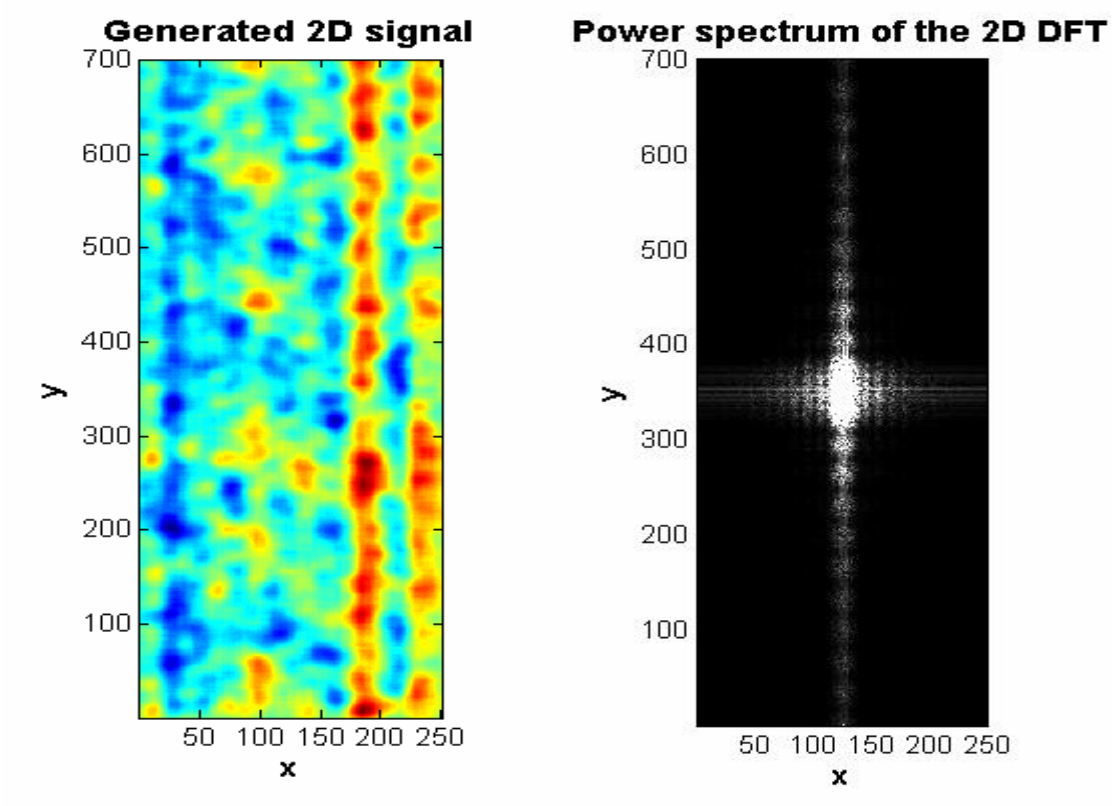


Figure 2.4: A Sample 2D dataset (left) has been Fourier transformed and its power spectrum is shown (right). The whiter the spot, the higher the amplitude.

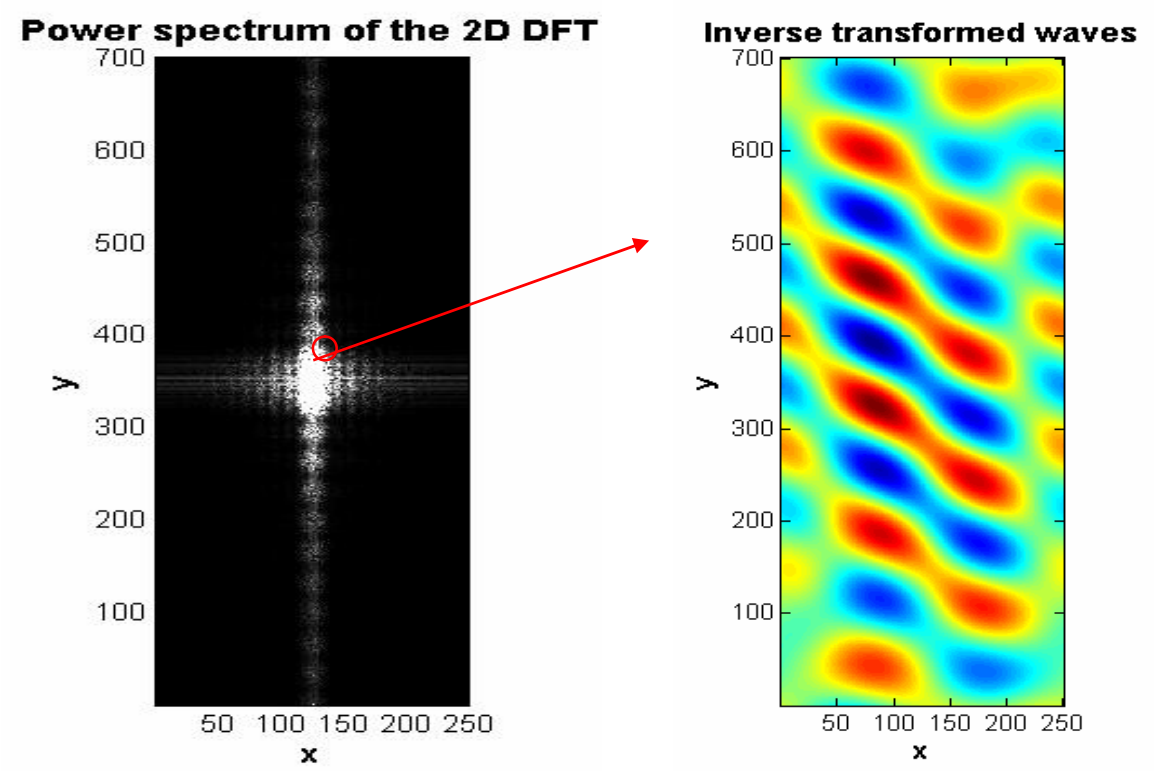


Figure 2.5: A few components of the power spectrum (left) have been inverse transformed back to a signal (right). The components have been chosen in a way that they represent diagonal waves only.

3 Fourier Transform in CD/MD Separation

In a paper or board machine, different kinds of quality variables are measured from the paper web /13/. In this chapter, we are using the discrete Fourier transform (DFT) to separate different kinds of variations out from the given quality variable dataset. The quality variable analyzed in this chapter is the basis weight of the paper web that has been measured offline with the paper web analyzer Advanced Total Paper Analyzer (ATPA) in Falun, Sweden. The paper roll from which the data is measured is from a board machine in Kaukopää mills in Imatra, Finland.

The offline measurements are measured more frequently than online, making the online measurements a sparse subset. The benefit from this is, naturally, that we get a more accurate picture of paper web variation. The ATPA data here is considered as a true 2D quality variable web sample and online data represent estimation to this, constructed from the zigzag path of the ATPA.

The x-dimension of the ATPA data is called the cross direction (CD) and the y-dimension the machine direction (MD), illustrating the direction that the paper or board is moving in a paper or board machine. The ATPA analyzer measures basis weight of paper with a traversing scanning path so that one unit (scan) to MD is 14 cm and one unit to CD is 1 cm. The size of the dataset is 250 x 1454 units thus making the total area of the paper web analyzed in this chapter 2.5 m x 202 m.

In Figure 3.1 one can see the quality variable dataset after the filtration with a low-pass Gaussian filter. The filtration is done by averaging the data with a 20 x 20 Gaussian window.

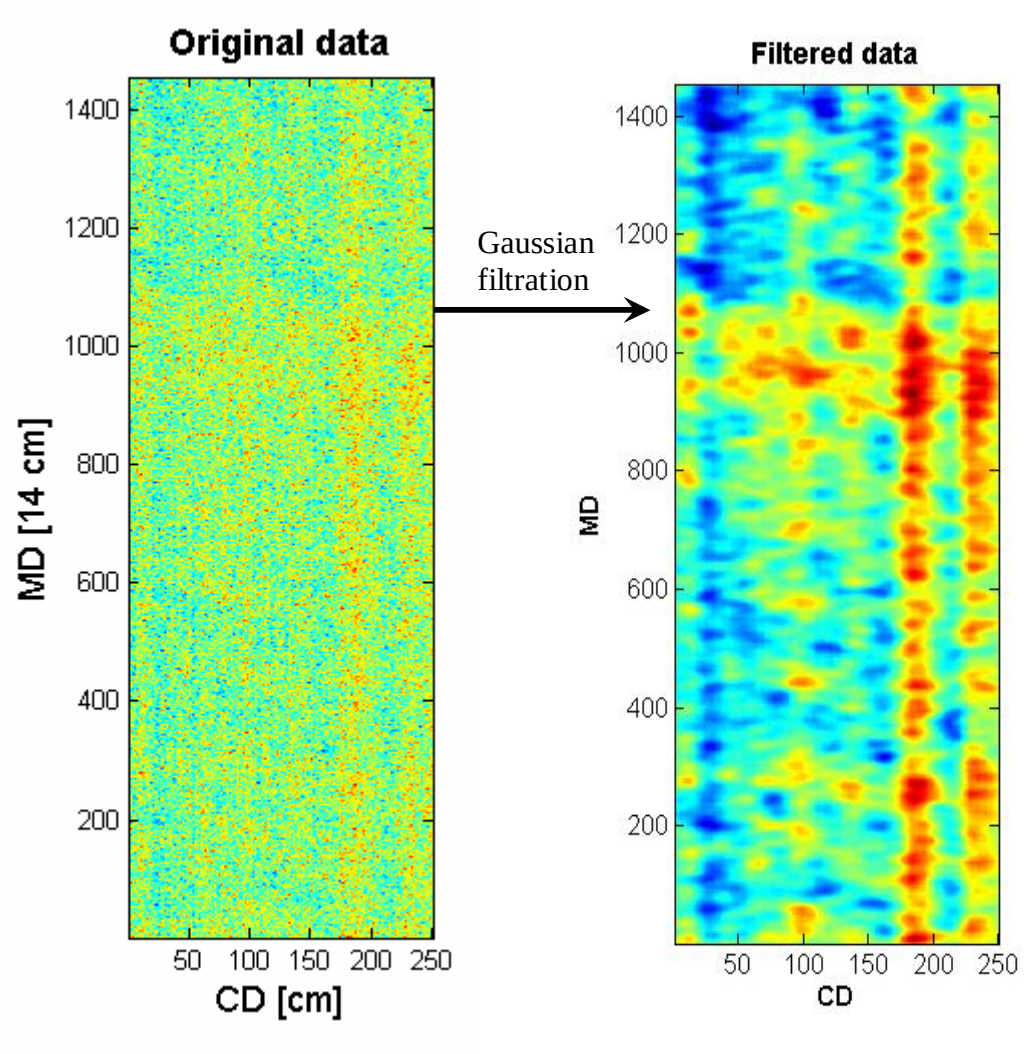


Figure 3.1: 2D figures of the sample offline data analyzed in this chapter. The left figure is the original data. The right figure is the same data after the filtration

3.1 Fourier Separation Method

The 2D discrete Fourier transform is applied here to the ATPA data. As the dataset is relatively small, the size of the Fourier window is set to be the same as the data matrix. So the dimensions of the Fourier matrix are 250 * 1454. Thus the wavelength range of this Fourier transform is 1 cm – 125 cm in CD and 0.14 m – 101 m in MD. The lowest wavelength in both directions is one unit and the highest half of the length of the data due to the Nyquist phenomenon.

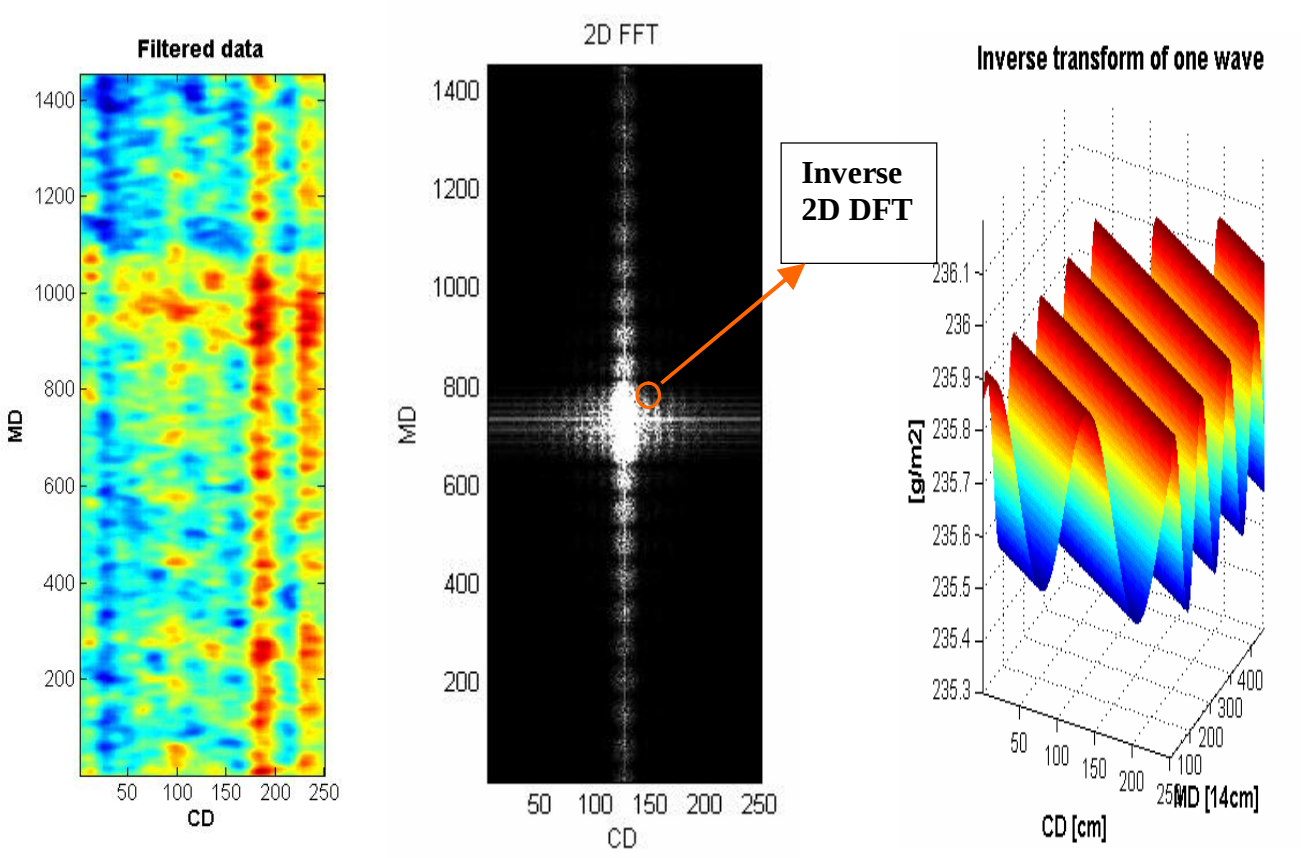


Figure 3.2: An example of a (inverse) Fourier transform: first the data, then its Fourier power spectrum, and finally inverse transform of one Fourier peak

This example shows how the 2D DFT (discrete Fourier transform) works. At first, the transform has been made to the data (left) and its power spectrum is shown in the middle. Then, to clarify what each coefficient in the Fourier power spectrum matrix means, the inverse Fourier transform has been made to one Fourier coefficient which coordinates are (CD, MD) = (3, 12). Other coefficients in the Fourier matrix are set to zero. The resulting

wave should thus be parallel to the vector (3, 12) and have a wavelength of (according to the equation 2.12)

$$wl = \sqrt{(\frac{2{,}50}{3})^2 + (\frac{1454}{12} * 0{,}14)^2} = 22.6m\,. \qquad \textbf{3.1}$$

The 0.14 factor in the equation comes from the fact that one unit in MD equals 14 cm. The wave can be seen in Figure 3.2.

3.2 CD/MD Separation

Since the quality control system works separately for CD and MD variations, it is important to separate the measured 2-dimensional data into its CD and MD components. In online systems this is done by considering one scan of a traversing scanner to be a CD profile of the paper and by averaging that to get one point for the MD profile. The speed of paper in a paper machine is high compared to the scanning speed: during one scan the paper can move for example 400 m in MD. Since the width of the paper is approximately 5 m, we can easily see that one scan is far from the “real” CD profile. The main difference comparing online data to ATPA data is that ATPA data is much denser and one scan is much closer to the real CD profile. The sparse online measurement path is compared to the dense web analyzer data (Figure 3.3).

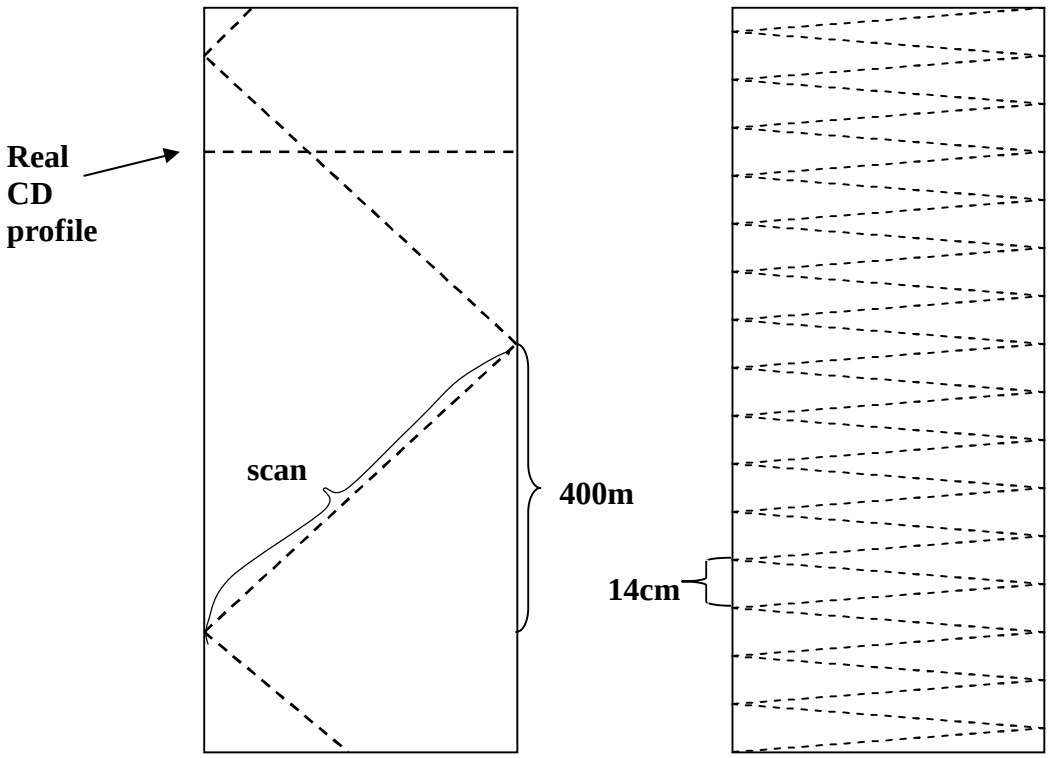


Figure 3.3: Comparison of online and ATPA scanning paths: The online path (left) and the ATPA path (right)

By using the same Fourier technique presented in earlier Chapter 2, the CD and MD variations can be separated from the ATPA data. In the 2D Fourier component matrix, the first row represents the waves directly in the CD (the MD component is zero) and correspondingly the first column the waves in MD. Thus we can easily separate the CD variation by setting everything else to zero but the 1st row and then calculating an inverse transform. The MD variation can be separated in a similar way. The following Figure 3.4 shows the result of this technique.

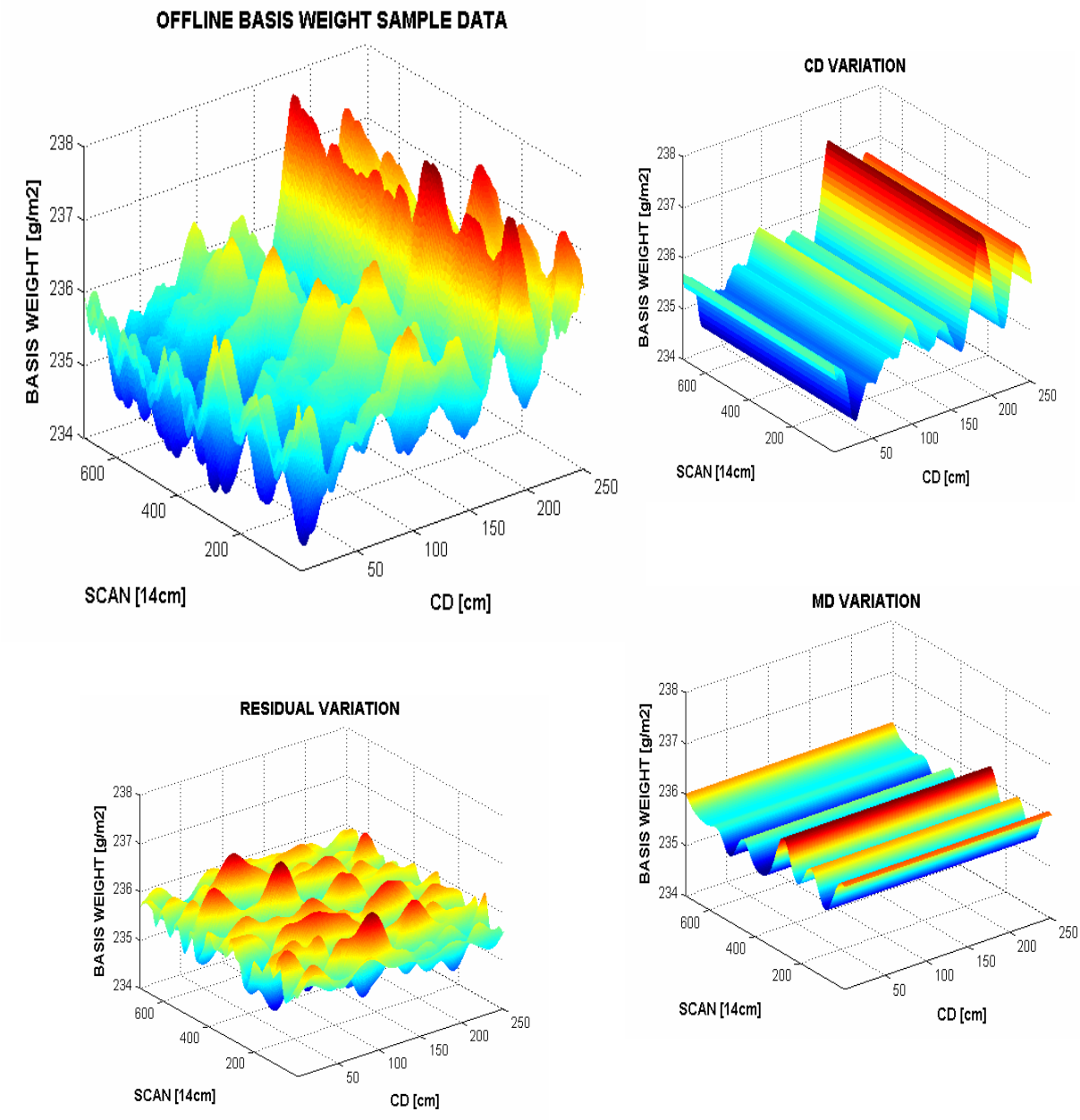


Figure 3.4: Offline basis weight sample data separated into its CD, MD and residual variation

In Figure 3.4 we can see a sample of the basis weight dataset 3-dimensionally. To make the figures clearer, only one half of the scans (rows) are shown here and the filtered data has been used. The CD and MD variation has been extracted from the data using the 2D DFT. The CD variation is an inverse DFT of the first row of the Fourier component matrix. Similarly, the MD variation is an inverse transform of the first column. Residual variation refers to the variation what is left after removing the CD and MD variations from the sample data. That means in practice an inverse transform of all the components left in the Fourier transform matrix. Note that since there are only a few big waves in both directions, only eight strongest Fourier components have been taken into account when calculating CD or MD variations.

From this figure, we can see how the CD variation is stronger than the MD variation. That can also be seen directly from the Fourier components; the amplitudes of the biggest CD components are 20 – 80% bigger than the biggest MD components. The strongest wavelengths, which were used in the Figure 3.4, range from 25 cm to 125 cm in CD and from 10 m to 100 m in MD.

4 Mathematical Background of the Kalman Filter

Theoretically the Kalman filter is an estimator for what is called the linear-quadratic problem. The linear-quadratic problem is the problem of estimating the instantaneous state of a linear dynamic system perturbed by white noise. The Kalman filter also uses measurements linearly related to the state but corrupted by white noise. The resulting estimator is statistically optimal with respect to any quadratic function of estimation error.

As far as practical applications of signal analysis are concerned, the Kalman filter is one of the greatest discoveries in the history of statistical estimation theory and possibly the greatest discovery in the twentieth century. It has been applied to various complex dynamic systems, such as continuous manufacturing processes, aircrafts, ships, weather forecasting /3/ and spacecraft. For these applications, it is not always possible or desirable to measure every variable that you want to control, and the Kalman filter provides a means for inferring the missing information from indirect and noisy measurements. The Kalman filter is also used for predicting the likely future courses of dynamic systems that people are not likely to control. /1/

From a practical point of view, there are several perspectives to look at the Kalman filter:

- *As a mathematical tool*. It does not solve any problem all by itself, although it can make it easier for an expert to do it.
- *As a computer program*. It has been called “ideally suited to digital computer implementation”, in part because it uses a finite representation of the estimation problem and also a finite number of variables.
- *As a complete statistical characterization of an estimation problem*. It is more than an estimator because it propagates the entire probability distribution of the variables it is asked to estimate. It takes into account also the influence of all the past measurements.
- In a limited context, *the Kalman filter is also a learning method*. It uses a model of the estimation problem that distinguishes between phenomena (what one is able to observe), noumena (what is really going on), and the state of knowledge about the noumena that one can deduce from the phenomena. That state of knowledge is represented by probability distributions. To the extent that those probability

distributions represent knowledge of the real world and the cumulative processing of knowledge is learning, this is a learning process. /1/

Kalman filter is named after Rudolf Emil Kalman who was born in Budapest, Hungary on May 19, 1930. Kalman received his Bachelor's and Master's degrees in electrical engineering at MIT (Massachusetts Institute of Technology) in 1953 and 1954, respectively. His master thesis topic was the behaviour of second-order difference equations. It was suspected that second-order difference equations might be modelled by something analogous to the describing functions used for second-order differential equations. Kalman discovered that this was not true, in fact, difference equations were found to exhibit chaotic behaviour.

The algebraic nature of systems theory first became of interest to Kalman in 1953. After realising that linear discrete-time systems can be solved in a similar way than continuous-time systems, an idea occurred to him that there is no fundamental difference between continuous and discrete linear systems. The two must be equivalent in a sense even though the solutions of the linear differential equations cannot go to zero and stay there. In 1954 he began studying the issue of controllability, which is the question of whether there exists an input (control) function to a dynamic system that will drive the state of the system to zero.

In 1958 an idea occurred to Kalman: Why not apply the notion of state variables to the Wiener filtering problem? By implementing probability theory and calculating expectation values Kalman derived the Wiener filter that we now call the Kalman filter. /1, 5/

4.1 Kalman Filter Equations

A Kalman filter is a way of estimating the state of a stochastic system by using observations that are functions of the state. It is based on a few equations, which are derived in this chapter. The Kalman filter used in this thesis is linear, although it can be extended to apply the nonlinear models as well, case it is called the Extended Kalman Filter (EKF). Also, we concentrate on the discrete Kalman filter because of the discrete nature of our estimation problem of the quality of paper web.

Now let us define a discrete stochastic system by the types of plant and measurement models, as represented in Table 4.1 (Equations 4.1-4.4) /1/. Δ stands for the Kronecker delta function. The system is assumed to have n variables and l different measurement types.

Table 4.1: Linear plant and measurement models

Name	Equation in discrete time	Eq number
Plant	$x_k = A_{k-1}x_{k-1} + w_{k-1}$	**4.1**
Measurement	$y_k = H_k x_k + v_k$	**4.2**
Plant noise	$E\langle w(k)\rangle = 0$ $E\langle w_k w_i^T\rangle = \Delta(k-i)Q_k$	**4.3**
Observation noise	$E\langle v(k)\rangle = 0$ $E\langle v_k v_i^T\rangle = \Delta(k-i)R_k$	**4.4**

Table 4.2: Dimensions of vectors and matrices in a linear model

Symbol	Dimensions	Symbol	Dimensions
x, *w*	*n* x 1	*A*, *Q*	*n* x *n*
y, *v*	*l* x 1	*H*	*l* x *n*
R	*l* x *l*	Δ	scalar

Table 4.3: Explanation of the model and plant symbols

Symbol	Explanation
x	system state vector
A	linear model matrix
w	plant noise, Gaussian
v	measurement noise, Gaussian
y	measurement
H	measurement sensitivity matrix
t	time (discrete)
Q	covariance matrix of w_k
R	covariance matrix of v_k
E	expectation value

We assume that the measurement and plant noise v_k and w_k are uncorrelated zero-mean Gaussian processes. The objective is to find an estimate of the state vector x_k, represented by

$\hat{x}_k$, a linear function of the measurements $y_i,\ldots, y_k$ that minimizes the weighted mean-square error

$$E\left(\left[x_k - \hat{x}_k\right]^T M\left[x_k - \hat{x}_k\right]\right), \qquad \textbf{4.5}$$

where E is referring to the expected value and M is any symmetric nonnegative-definite weighting matrix. That makes our estimation a linear quadratic Gaussian (LQG) problem. The dynamic systems are linear, the performance cost functions are quadratic and the random processes are Gaussian. /1/

Suppose now that a measurement y has been made at time t_k, and according to that the estimate of the state x should be updated. The measurement y is linearly related to the state by the equation 4.2, where H is the measurement sensitivity matrix and v_k the measurement noise. The new estimate of $\hat{x}_k$ is a linear function of the a priori estimate and the measurement y:

$$\hat{x}_k(+) = K_k^1 \hat{x}_k(-) + \overline{K}_k y_k, \qquad \textbf{4.6}$$

where $\hat{x}_k$ (-) is the a priori value of the estimate of x_k (the measurement hasn't been taken into account) and $\hat{x}_k$ (+) is the a posteriori value of the estimate (after the measurement has been taken into account). The weighting matrices K_k^1 and $\overline{K}_k$ are unknown. For deriving optimal values of those matrices, we start by using the orthogonality condition:

$$E\left\langle\left[x_k - \hat{x}_k(+)\right]y_i^T\right\rangle = 0, i = 1,2,\ldots,k-1, \qquad \textbf{4.7}$$

$$E\left\langle\left[x_k - \hat{x}_k(+)\right]y_k^T\right\rangle = 0. \qquad \textbf{4.8}$$

By applying the equation 4.1 (in Table 4.1) for x_k and the equation 4.6 for $\hat{x}_k(+)$ to the equation 4.7 and taking into account the fact that v_k and w_k are uncorrelated, that is $Ew_k y_i^T = 0$ for $1 \le i \le k$, the following equation can be derived:

$$E\left|\left\langle A_{k-1}x_{k-1} + w_{k-1} - K_k^1(-) - K_k y_k\right\rangle\right| = 0, i = 1,\ldots k-1, \qquad \textbf{4.9}$$

then, by applying $y_k = H_k x_k + v_k$, we get

$$E\left[A_{k-1}x_{k-1} - K_k^1 x_k(-) - K_k H_k x_k - K_k v_k\right]y_i^T = 0. \qquad \textbf{4.10}$$

Also equation 4.7 holds for the previous equation and, by v_k and w_k being uncorrelated, we get that $Ev_k y_i^T = 0$. By implementing these two arguments into equation 4.9, the following form is derived:

$$A_{k-1}Ex_{k-1}y_i^T - K_k^1 Ex_k^{\wedge}(-)y_i^T - \overline{K}_k H_k A_{k-1}Ex_{k-1}y_i^T - \overline{K}_k Ev_k y_i^T = 0,$$
$$A_{k-1}Ex_{k-1}y_i^T - K_k^1 Ex_k^{\wedge}(-)y_i^T - \overline{K}_k H_k A_{k-1}Ex_{k-1}y_i^T = 0,$$
$$E\left\langle \left[x_k - \overline{K}_k H_k x_k - K_k^1 x_k\right] - K_k^1(\hat{x}_k(-) - x_k)\right\rangle y_i^T = 0,$$

$$\left[I - K_k^1 - \overline{K}_k H_k\right]Ex_k y_i^T = 0. \qquad \textbf{4.11}$$

Now the previous equation 4.11 can be satisfied for any x_k if

$$K_k^1 = I - \overline{K}_k H_k. \qquad \textbf{4.12}$$

This choice of K_k^1 makes the equation 4.6 to satisfy the orthogonality condition given in the equation 4.7. The next step is to derive the formula for $\overline{K}_k$.

Let's start from the errors

$$\tilde{x}_k(+) \triangleq \hat{x}_k(+) - x_k, \qquad \textbf{4.13}$$
$$\tilde{x}_k(-) \triangleq \hat{x}_k(-) - x_k, \qquad \textbf{4.14}$$
$$\tilde{y}_k \triangleq \hat{y}_k(-) - y_k = H_k\hat{x}_k(-) - y_k, \qquad \textbf{4.15}$$

where $\tilde{x}_k$(-) and $\tilde{x}_k$(+) are the estimation error before and after updates, respectively. Now the estimation parameter $\hat{x}_k$ depends linearly on x_k, which depends linearly on y_k. Therefore, from equation 4.8

$$E\left\langle \left[x_k - \hat{x}_k(+)\right]y_k^T(-)\right\rangle = 0 \qquad \textbf{4.16}$$

and also

$$E\left\langle \left[x_k - \hat{x}_k(+)\right]\tilde{y}_k^T\right\rangle = 0. \qquad \textbf{4.17}$$

Substitute x_k, $\hat{x}_k$ and $\tilde{y}_k$ from equations 4.1, 4.6 and 4.15, respectively, to get the following:

$$E\left[A_{k-1}x_{k-1} + w_{k-1} - K_k^1 x_k(-) - \overline{K}_k z_k\right]\left[H_k\hat{x}_k(-) - z_k\right]^T = 0, \qquad \textbf{4.18}$$

however, we know that $Ew_k z_k^T = Ew_k\hat{x}_k^T = 0$ as mentioned earlier and the equation 4.18 simplifies to

$$E\left[A_{k-1}x_{k-1} - K_k^1 x_k(-) - \overline{K}_k z_k\right]\left[H_k\hat{x}_k(-) - z_k\right]^T = 0. \qquad \textbf{4.19}$$

Now we can substitute K_k^1 from equation 4.12, y_k from equation 4.2, $\tilde{x}_k$ from equation 4.14 and use once again the fact that $E\tilde{x}_k(-)v_k^T = 0$ to finally approach our goal of determining the Kalman gain matrix:

$$\begin{aligned} &E\left\langle \left[A_{k-1}x_{k-1} - \hat{x}_k(-) + \overline{K}_k H_k \hat{x}_k(-) - \overline{K}_k H_k x_k - \overline{K}_k v_k\right]\left[H_k \hat{x}_k(-) - H_k x_k - v_k\right]^T \right\rangle = 0, \\ &E\left\langle \left[x_k - \hat{x}_k(-) - \overline{K}_k H_k (x_k - \hat{x}_k(-)) - \overline{K}_k v_k\right]\left[H_k \tilde{x}_k(-) - v_k\right]^T \right\rangle = 0, \\ &E\left\langle \left[\hat{x}_k(-) + \overline{K}_k H_k \hat{x}_k(-) - \overline{K}_k v_k\right]\left[H_k \tilde{x}_k(-) - v_k\right]^T \right\rangle = 0. \end{aligned} \qquad \textbf{4.20}$$

By definition, the a priori covariance matrix is

$$P_k(-) = E\left\langle \tilde{x}_k(-)\tilde{x}_k{}^T(-)\right\rangle. \qquad \textbf{4.21}$$

It satisfies the equation

$$\left[I - \overline{K}_k H_k\right]P_k(-)H_k^T - \overline{K}_k R_k = 0, \qquad \textbf{4.22}$$

from which we can solve the Kalman gain matrix as a function of the a priori covariance

$$\overline{K}_k = P_k(-)H_k^T\left[H_k P_k(-)H_k^T + R_k\right]^{-1}. \qquad \textbf{4.23}$$

For deriving an equation for the a posteriori covariance, we start from an equation similar to the equation 4.21:

$$P_k(+) = E\left\langle \tilde{x}_k(+)\tilde{x}_k{}^T(+)\right\rangle. \qquad \textbf{4.24}$$

By substituting the equation 4.12 into the equation 4.6, one can derive the observational update for the state estimate as follows

$$\hat{x}_k(+) = (I - \overline{K}_k H_k)\hat{x}_k(-) + \overline{K}_k y_k$$

$$\hat{x}_k(+) = \hat{x}_k(-) + \overline{K}_k[y_k - H_k \hat{x}_k(-)]. \qquad \textbf{4.25}$$

By setting y_k according to equation 4.2 and extracting x_k from both sides of the equation 4.25, we get the following

$$\tilde{x}_k(+) = (I - \overline{K}_k H_k)\tilde{x}_k(-) + \overline{K}_k v_k. \qquad \textbf{4.26}$$

Now we can substitute this to the equation 4.24 and obtain the following (note again that $E\tilde{x}_k(-)v_k^T = 0$)

$$P_k(+) = E\{[I - \overline{K}_k H_k]\tilde{x}_k(-)\tilde{x}_k^T(-)[I - \overline{K}_k H_k]^T + \overline{K}_k v_k v_k^T \overline{K}_k^T\}$$

$$P_k(+) = [I - \overline{K}_k H_k]P_k(-)[I - \overline{K}_k H_k]^T + \overline{K}_k R_k \overline{K}_k^T \qquad \textbf{4.27}$$

Now we have derived an equation for the a posteriori error covariance matrix as a function of the a priori covariance and the Kalman gain matrix. That version of the equation is called the "Joseph-form" of the covariance update, named after P.D.Joseph, who derived it in the first place. The following form of the latest equation can be obtained by substituting $\overline{K}_k$ from the equation 4.23 (see /1/ for a more detailed derivation):

$$P_k(+) = [I - \overline{K}_k H_k]P_k(-). \qquad \textbf{4.28}$$

This equation is later on used in the Kalman filter algorithm.

Our next step is to derive an equation for the error covariance extrapolation. It models the effects of time on the estimation uncertainty. We start from the basic equations of the a priori covariance (4.21) and the state estimate

$$\hat{x}_k(-) = A_{k-1}\hat{x}_{k-1}(+).$$

Now, subtract x_k from both sides to obtain

$$\begin{aligned} \hat{x}_k(-) - x_k &= A_{k-1}\hat{x}_{k-1}(+) - x_k \\ \tilde{x}_k(-) &= A_{k-1}[\hat{x}_{k-1}(+) - x_{k-1}] - w_{k-1}, \qquad \textbf{4.29} \\ \tilde{x}_k(-) &= A_{k-1}\tilde{x}_{k-1}(+) - w_{k-1} \end{aligned}$$

for the update of the estimation error. Starting from the definition of the a priori covariance and using the latest equation multiplied by $\tilde{x}_k^T(-)$ from both sides and taking expected values, we get:

$$\begin{aligned} P_k(-) &= E\left\langle \tilde{x}_k(-)\tilde{x}_k{}^T(-)\right\rangle \\ &= A_{k-1}E[\tilde{x}_{k-1}(+)\tilde{x}_{k-1}{}^T(+)]A_{k-1}^T + E[w_{k-1}w_{k-1}^T] \end{aligned}$$

$$P_k(-) = A_{k-1}P_{k-1}^{(+)}A_{k-1}^T + Q_{k-1}. \qquad \textbf{4.30}$$

Now we have the a priori covariance matrix as a function of the model matrix and the a posteriori covariance from the previous time step. /1/

To make the understanding of the forthcoming Kalman filter based algorithms easier; we summarize these previously derived Kalman equations. The following basic steps are needed when computing the discrete Kalman filter procedure:

- *Initializing*
  - Define the noise levels for measurements and the model:
    - $w_k \sim N(0, Q)$
      $v_k \sim N(0, R)$
  - Define the initial conditions for the state and the error covariance
    - $E\langle x_0 \rangle = \hat{x}_0$
      $E\langle \tilde{x}_0 \tilde{x}_0^T \rangle = P_0$
- *Extrapolating*
  - Extrapolate the state estimate and the error covariance (equations 4.1 and 4.30)
    - $x_k = A_{k-1} x_{k-1} + w_{k-1}$
      $P_k(-) = A_{k-1} P_{k-1}^{(+)} A_{k-1}^T + Q_{k-1}$
  - Calculate the Kalman gain (equation 4.23)
    - $\overline{K}_k = P_k(-) H_k^T \left[H_k P_k(-) H_k^T + R_k \right]^{-1}$
- *Updating*
  - Update the error covariance and the state estimate (equations 4.28 and 4.25)
    - $P_k(+) = [I - \overline{K}_k H_k] P_k(-)$
      $\hat{x}_k(+) = \hat{x}_k(-) + \overline{K}_k [y_k - H_k \hat{x}_k(-)].$

4.2 **The Fixed-Lag Kalman Smoother**

A smoother, in general, is a type of estimator that uses not only current and past but also the future time step observations to estimate the state at the current time step. The Kalman filter introduced above uses only the current and all the past observations (that is why it is called a

'filter'). To be able to use future observations, the state of the system is set to be behind the real current time step. Smoothers are used to make the estimation more stable as more observations are taken into account. There are three basic kinds of smoothers, fixed-interval, fixed-point and fixed-lag (see for example /1/). A fixed-lag smoother is the one that has been successfully used in similar situations earlier on /3/.

A fixed-lag Kalman smoother uses a fixed number of time steps beyond the current time step of the estimator. This number l is called a fixed-lag. So we are estimating the system state at the time step t_{k-l}, where t_k is the current time step. Now we are able to take into account also l time steps beyond the time step we are estimating. This can be implemented into the Kalman equations using the following techniques.

For the first l steps we initialize the smoother using the equations of a fixed point smoother /1/ since we cannot yet use future time steps. The initial value for B is the identity matrix I. The following equation is used to obtain a smoothed state estimate at the first steps, $t_i < t_k$:

$$x_{[s]i|k} = x_{[s]i|k-1} + B_k \overline{K}_k (y_k - H\hat{x}_k(-)), \quad \textbf{4.31}$$

$$B_k = B_{k-1} P_{k-1} A_{k-1}^T P_k^{-1}(-), \quad \textbf{4.32}$$

where the subscript notation $[s]i|k$ refers to the smoothed estimate of the state at the time t_i, when measurements up to point t_k are known.

We have to save the values of x, P, A and Q for the first steps and then, after step l, implement the following to get the smoothed state estimate /1/:

$$x_{[s]k+1-l} = A_{k-l} x_{[s]k-l} + Q_{k-l} A_{k-l}^T P_{k-l} (x_{[s]k-l} - x_{k-l}) + B_{k+1} K_{k+1} (z_{k+1} - H_{k+1} A_k x_k) \quad \textbf{4.33}$$

$$B_{k+1} = B_k P_k A_k^T P_{k+1}^{-1} \quad \textbf{4.34}$$

5 The Kalman Filter in Paper Web Analysis

In the third chapter we used the Fourier transform to extract different kinds of waves from paper web data. The idea of using Fourier coefficients to represent CD and MD variation was presented. Now we shall show how the DFT (Discrete Fourier Transform) and the problem of a traversing scanner can be applied to the Kalman filter.

In weather forecasting, the situation is very similar to paper web analysis from a mathematical point of view. A traversing satellite produces very sparse measurement data which is then used to predict weather over the earth's entire atmosphere. In a paper machine, the traversing scanner produces sparse measurement data which is then used to predict the variation over the entire paper web. Different kinds of Kalman filter methods have been used rather successfully for weather forecasting /3/. This led to an idea of modelling the paper web by using the Kalman filter with the Fourier coefficients as state variables. Even though the Kalman filter has been previously used in paper web analysis /4/, this idea of using the Kalman filter with Fourier components in both CD and MD is novel.

The CD/MD separation method by using the Fourier transform was presented in chapter 3. Now we extend that idea to be a part of a general CD/MD separation model. That separation model is based on the Kalman filter, which uses Fourier coefficients as a state vector x. Separation is necessary because the QCS of a paper or board machine works separately for CD and MD variations. That is why measurement data y, which comes from the scanners in a paper or board machine, needs to be separated into its CD and MD components (Figure 5.1).

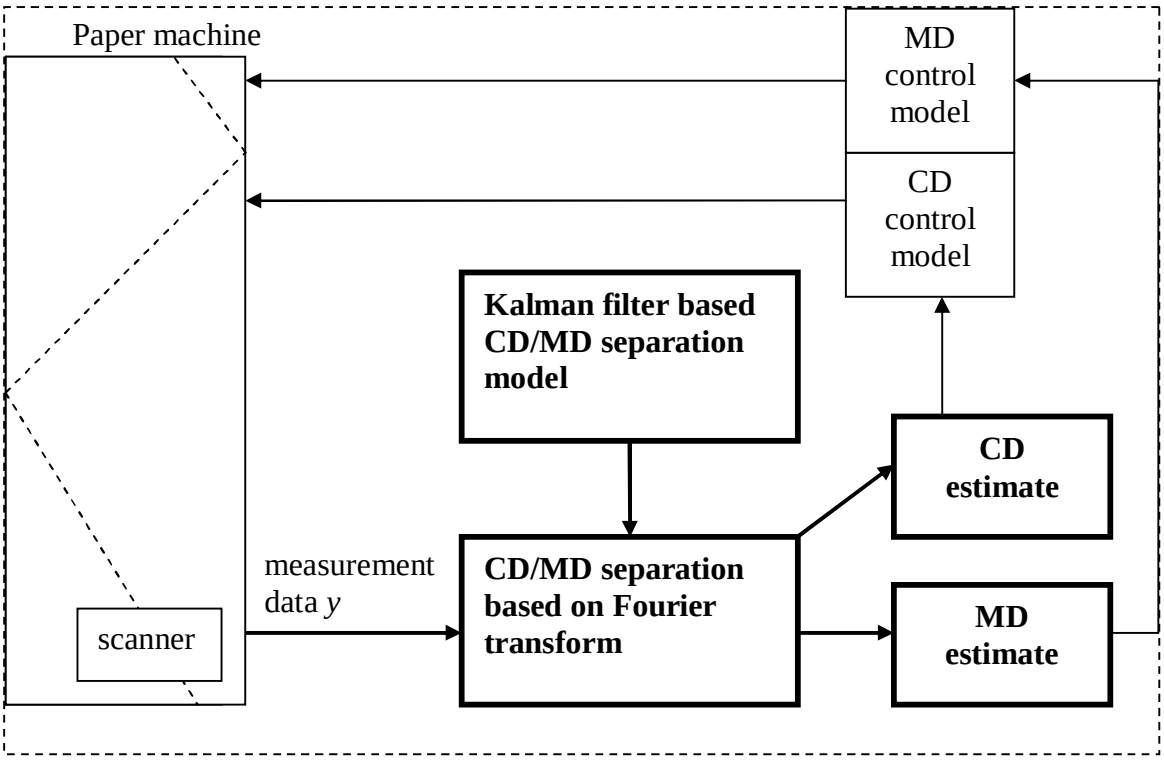


Figure 5.1: Paper machine and QCS relation to the separation model with Kalman filter

5.1 Kalman equations in paper web analysis

In the separation model with the Kalman filter, the following variables and equations, as shown in Table 5.1 and Table 5.2, are used.

Table 5.1: Variables and explanations of the Kalman filter in paper web analysis

Variable	Explanation
$x = [C_1, C_2, ..., C_n]$	Complex state vector with the Fourier coefficients as state variables C_1, C_2, …, C_n. Each of them represents one wave in the data either in CD or MD. Initially set to a zero vector. Note: This is later extended to include diagonal waves as well.

$Q = \begin{bmatrix} \mathrm{var}(C_1) & 0 & \cdots & 0 \\ 0 & \mathrm{var}(C_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \mathrm{var}(C_n) \end{bmatrix}$	Covariance of the plant e.g. covariance of the state vector. Note that the variance of a complex variable C with mean γ is defined by $\mathrm{var}(C) = E\left[(C-\gamma)(C-\gamma)^*\right]$ where * refers to complex conjugate and it is a real number.
$R = \mathrm{var}(y)$	Variance of the basis weight measurement data. Can be extended to covariance matrix if different kinds of measurements are observed. Initial value can be calculated from the dataset.
$P = QI$	Error covariance matrix. Initially the identity matrix multiplied by Q. Updated during the Kalman loop.
$A = e^{-2\pi i a n / T} I$	The phase shift operator matrix or model matrix. The identity matrix when no phase shift used. See chapter 2.1.2 for details of the phase shift.
$K = [k_1, k_2, ..., k_n]^T$	The Kalman gain matrix. Calculated during the Kalman loop, no initial value.
$H = [w_1, w_2, \cdots, w_n]$	Measurement sensitivity matrix. The inverse discrete Fourier transform operator corresponding to the current measurement data point e.g. one row of the inverse Fourier transform matrix introduced in chapter 2.
y	Basis weight measurement data point. In this case y is scalar but can be easily extended to a vector if there are different kinds of measurement types (moisture, caliper etc.) to be analyzed.

Table 5.2: Equations and explanations used in Kalman filter loop

Equation	Explanation
$x_k = Ax_{k-1}$	State vector extrapolation by the model one time step forward.
$H = W(k,:)$	One row of the inverse Fourier matrix W

	corresponding to the current time step *k*. See Chapter 2 for details of Fourier transform matrix.
$P_k = AP_{k-1}A^T + Q$	Error covariance extrapolation by the model one time step forward.
$K_k = P_k H_k^T \left[H_k P_k H_k^T + R\right]^{-1}$	Kalman gain matrix. Optimal weights to the state variables.
$P_k = \left[I - K_k H_k\right] P_k$	Error covariance matrix update. "Take new Kalman gain into account."
$x_{k+1} = x_k + K_k \left[y_i - H_k x_k\right]$	State vector update. "Take the measurement *y* into account."

5.2 CD/MD separation model

Variables and equations introduced in the previous chapter are now used in the Kalman filter based CD/MD separation model. The separation model uses the equations in a so called Kalman loop, which is a base loop of the Kalman filter. The basic idea of this model is to get the CD and MD variations out of the measurements coming from the paper or board machine scanners.

The routine of the separation model is the following. First, the measurement data *y* (in this case, basis weight) comes from the scanners as an input to the Kalman filter. Kalman filter initializes its variables as mentioned in Table 5.1, and starts the first round of the Kalman loop. The loop goes through every point of the input vector *y*, updating and extrapolating its variables as introduced in Table 5.2. Finally, after every element of the input vector *y* has been taken into account, the Kalman loop returns the final estimate of the state vector *x*.

State variables in *x* are defined in a way that each of them represents a Fourier coefficient corresponding either to CD or to MD variation. The next step is to split the state vector x into its CD and MD components regarding to the corresponding variation. Now we have Fourier coefficients for both CD and MD estimates and we get the actual estimates by using the inverse Fourier transform. The following Figure 5.2 explains the idea (see Figure 5.1 for the big picture).

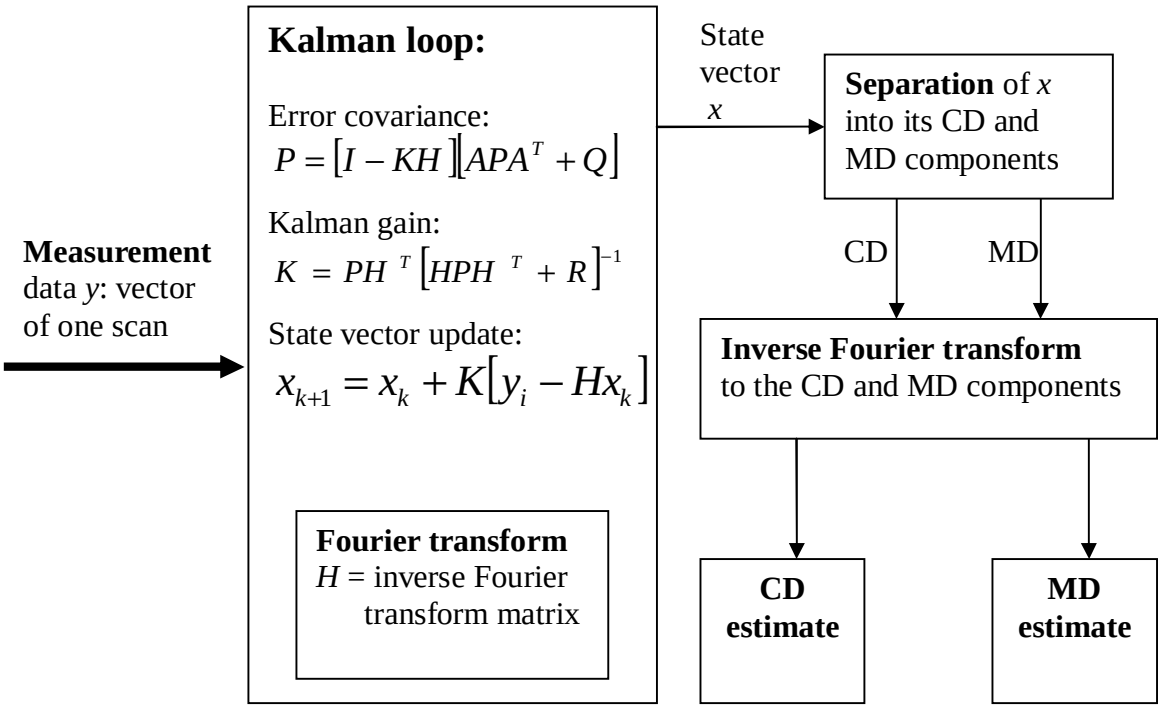


Figure 5.2: The CD/MD separation model with Kalman filter

6 Results of data assimilation with the Kalman filter

Data assimilation, in general, means mathematical modelling in a way that we merge observational data into a model. This merging makes very accurate and realistic estimations possible if the error levels of the observations and the model are known. An important aspect of data assimilation is the estimation and modelling of the error. The observational error arises from various sources: for example instrumental noise, sampling, measurements interpretation. The Kalman filter is a very widely used data assimilation method since it has built-in error estimation via the error covariance matrix.

In this chapter, a data assimilation method, the CD/MD separation model based on the Kalman filtering is applied to a 2D quality variable dataset. The separation model is also extended to be a 2-dimensional mathematical model, and the 2D figures are shown. These are applied to the basis weight sample data as introduced in chapter 3. The resulting figures and explanations are presented.

We are analysing the same basis weight sample dataset as in chapter 3, so we have 2,5 m * 200 m sized sample of basis weight paper web measured by the ATPA machine. The Kalman equations and variables, as introduced in chapter 5, are applied to the sample data. There are two perspectives of how to approach the results presented here. From a mathematical point of view, we are modelling the given paper property according to a 2D DFT based Kalman filtering model. On the other hand, we are separating the CD and MD variations out of the given sample data of basis weight. The idea here is to test whether the Kalman filter is able to reconstruct the most significant phenomena in the dataset according to sparse measurement data. Those phenomena are represented by the strongest Fourier components in both CD and MD.

The Kalman filter algorithm used for the data assimilation works in the following way:

- *Collecting measurements from the measurement path*. The idea here is to imitate an online scanner as well as possible with this short amount of data.
- *Defining the Fourier transform matrices according to the selected Fourier window*. The window size in this case equals the size of the dataset, which is 250 * 1454 units. One unit being 1 cm in CD and 14 cm in MD.

- *Defining wavelengths of the Fourier components*. In this case, we are using wavelengths ranging from 25 cm to 125 cm in CD and from 10 m to 100 m in MD.
- *Defining noise parameters for the Kalman filter*. That is, the measurement and process noise levels. In practice this means estimating how much we trust our model and the measurements. The higher the noise level of the measurements, the lower is its effect on the model and vice versa. The higher the noise level of the process, the higher is the effect of the measurements. Moreover, the process noise is separated into two components, the CD noise and the MD noise, in order to be able to deal with them separately.
- *Running the Kalman filter loop according to the measurement path*. The loop returns the updated state vector of Fourier components for both CD and MD variations (Figure 5.2).
- *Calculating the inverse Fourier transform* to the state vector to get final estimates for CD and MD profiles and an estimated 2D surface of the quality variable web.

6.1 The CD/MD Separation with the Kalman Filter Algorithm

The following sets of figures are presented as a result of the Kalman filter algorithm. The idea here is to demonstrate the Kalman filter's ability to recognise the CD and MD variations correctly from the dataset according to a sparse traversing measurement path. The traversing measurement paths are made with different angles in order to test whether it makes a difference for the Kalman filter. We start from a thick measurement pattern and then move on to more and more sparse ones. In each case, the measurement pattern superposed on the dataset, the estimated 2D surface and the estimated CD and MD profiles are presented (Figure 6.1, Figure 6.3 and Figure 6.2).

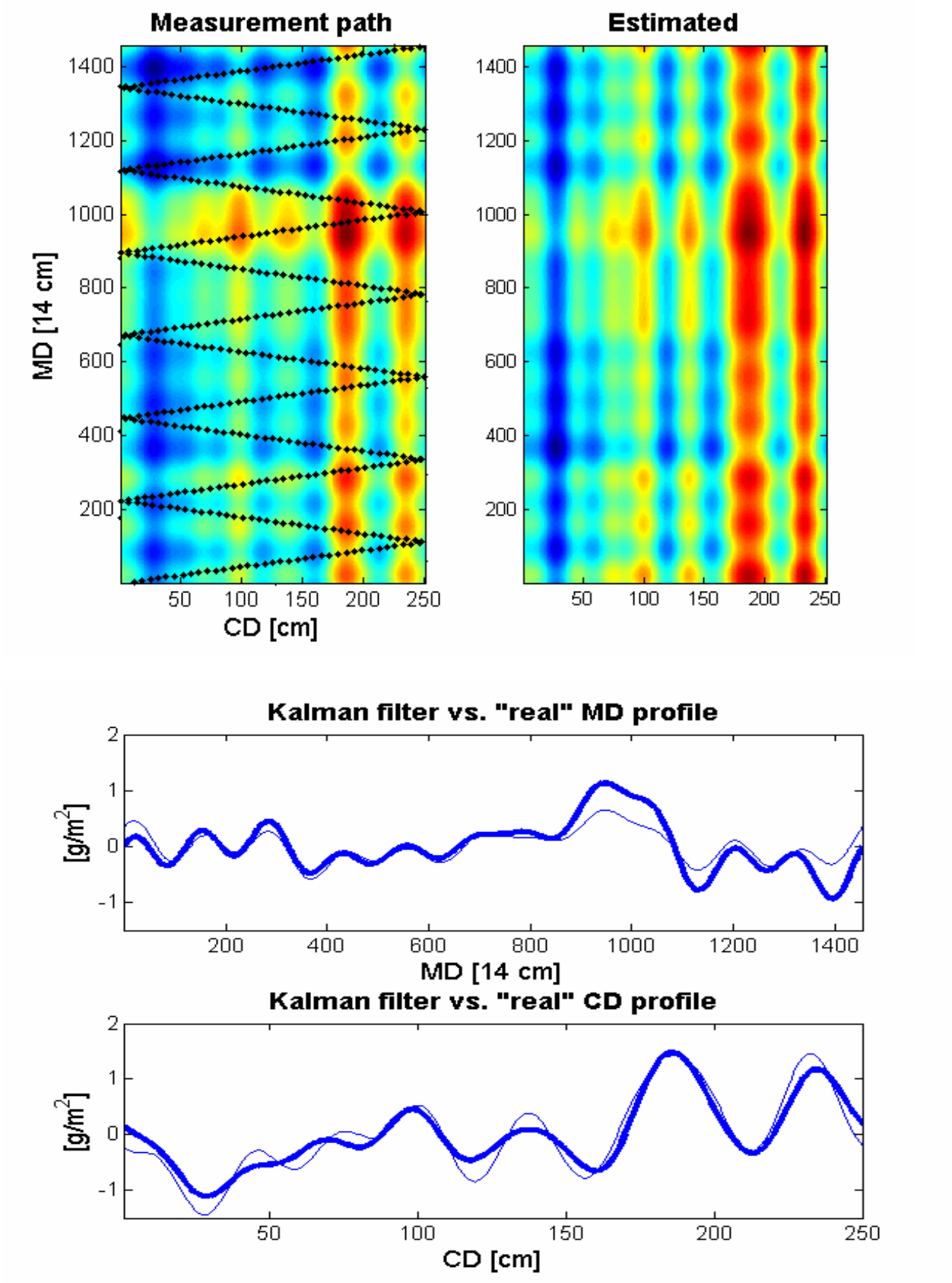


Figure 6.1: In the upper left figure, the original (filtered) surface is presented with a thick measurement path superposed onto it. In the right upper figure, the estimated 2D surface is presented. In the lower figure, the real (bold) and estimated (thin) CD/MD profiles are presented.

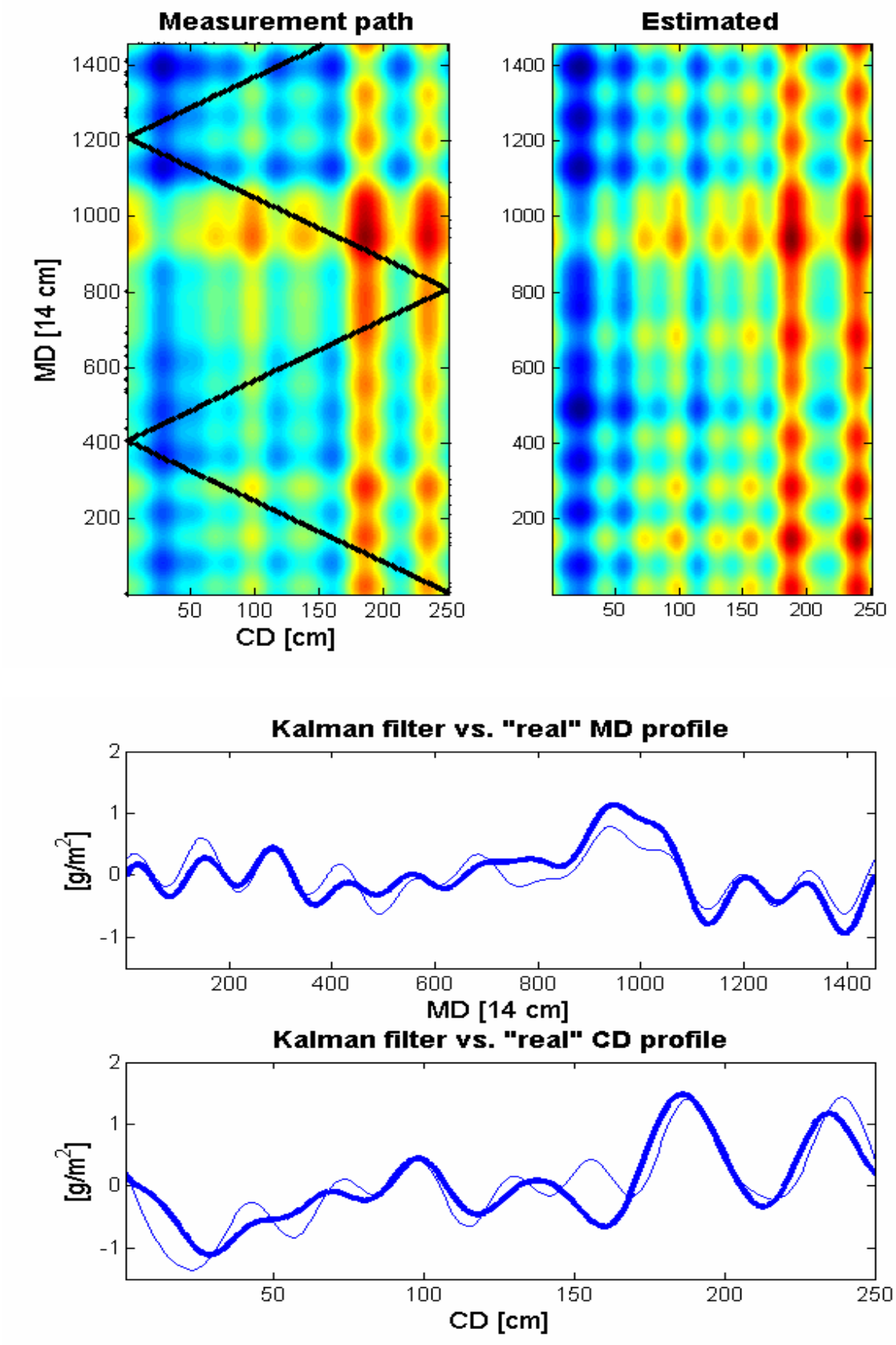


Figure 6.2: In the upper left figure, the original (filtered) surface is presented with a sparse measurement path superposed onto it. In the right upper figure, the estimated 2D surface is presented. In the lower figure, the real (bold) and estimated (thin) CD/MD profiles are presented.

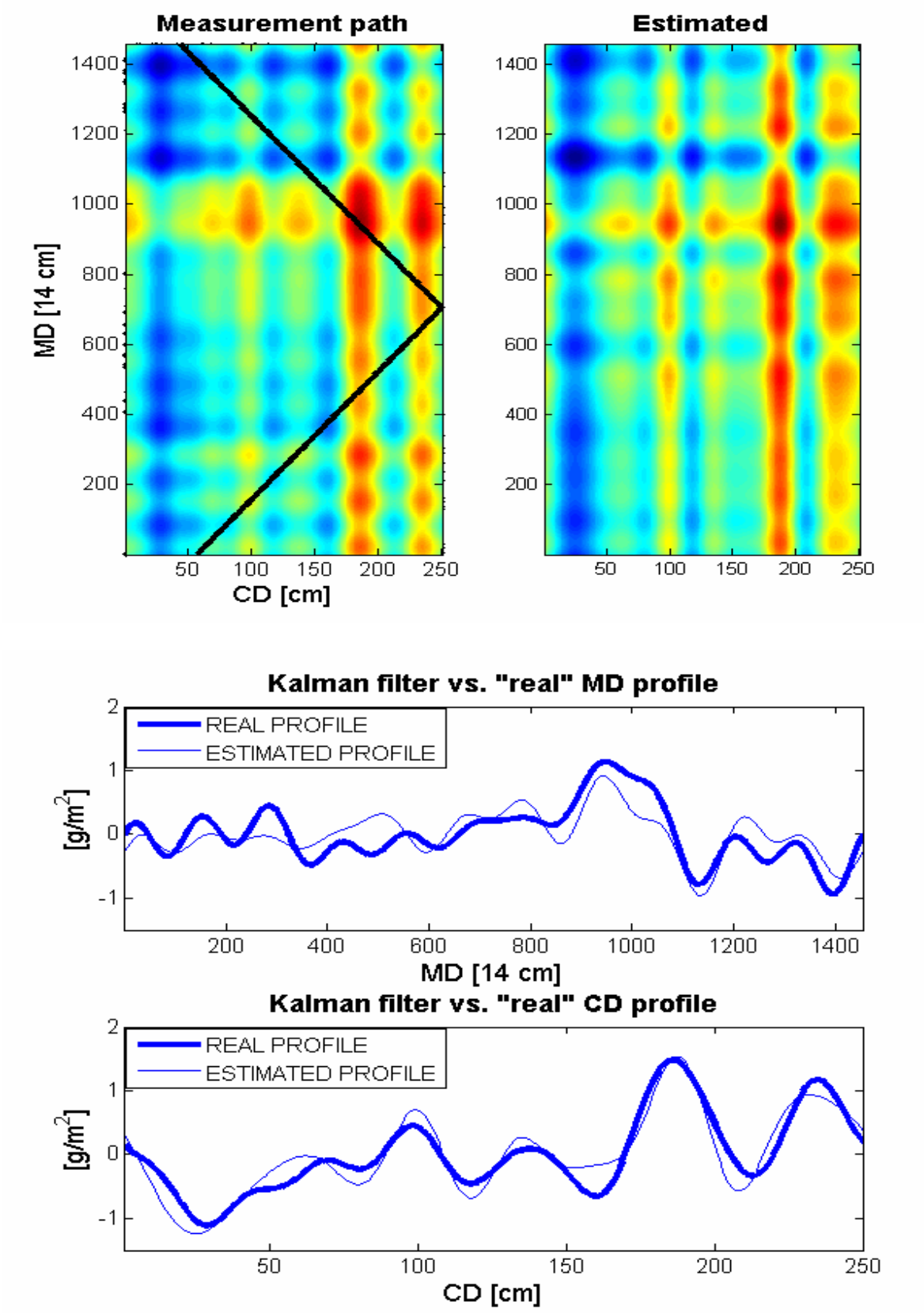


Figure 6.3: In the upper left figure, the original (filtered) surface is presented with a very sparse measurement path superposed onto it. In the right upper figure, the estimated 2D surface is presented. In the lower figure, the real (bold) and estimated (thin) CD/MD profiles are presented.

All these runs were made in the following way: We start from the zero, i.e. the Fourier components are set to be equal to zero. Then we feed the Kalman algorithm with measurements taken from different types of measurement routes: tight, sparse and very sparse. The resulting CD/MD profiles and constructed 2D surface are then presented. Note that even though the upper left figure is filtered for visualizing purposes, the measurements are always taken from the original, non-filtered data.

It can be seen from these figures that the Kalman filter is able to reconstruct the most significant variations even with the most sparse measurement path. It is however visible that the sparser the measurement pattern, the worse the result. Especially the MD profile seems to get worse with sparser path. That is probably due to the fact that CD variation is higher and thus 'easier' to extract. Also one noticeable point is that the high frequency MD variation is not of principal interest here since the QCS is not able to react fast enough in MD, making those waves uncontrollable. On the other hand, the result is not supposed to be 100% accurate, the most important aspect being the reconstructing of the main variations, which can be achieved even with a very sparse measurement data.

6.2 Analysis of the Kalman Filter Variables

In this chapter, we are doing some deeper analysis of what is going on inside the Kalman filter. As introduced earlier, the variables in the Kalman filter are defined so that each of them represents one wave in the dataset. The waves are either CD or MD with the wavelengths ranging 25 cm – 125 cm in CD and 10 m – 100 m in MD. One interesting thing here is to find out how fast each of the amplitudes of the waves is rising. Our second goal is to specify the most significant waves and, on the other hand, to check whether there are any meaningless variables. Thirdly, we want to test the Kalman filter's ability to recognize sudden changes in the data, as if there was a control effect or a malfunction in a paper machine.

All of the following figures and analysis are done by running the algorithm in a similar way: We start by running it in the very same way as before, but this time, since we need to run longer samples to be able to see the stabilization of the variables, we run the algorithm loop

several times with different measurement paths with the same angle of the scanner. The main benefit from this is, besides the study of stability, the ability to add some artificial changes into the data. The angle of the path is chosen in a way that one scan represents 56 m of MD distance. That is still way shorter than scans' lengths on real paper machines where the length of one scan in MD usually exceeds 400 m, but with this short amount of data, this approximation has to be done. From now on we call these online scans. The following Figure 6.5 shows an example of the measurement path used later on for analysis:

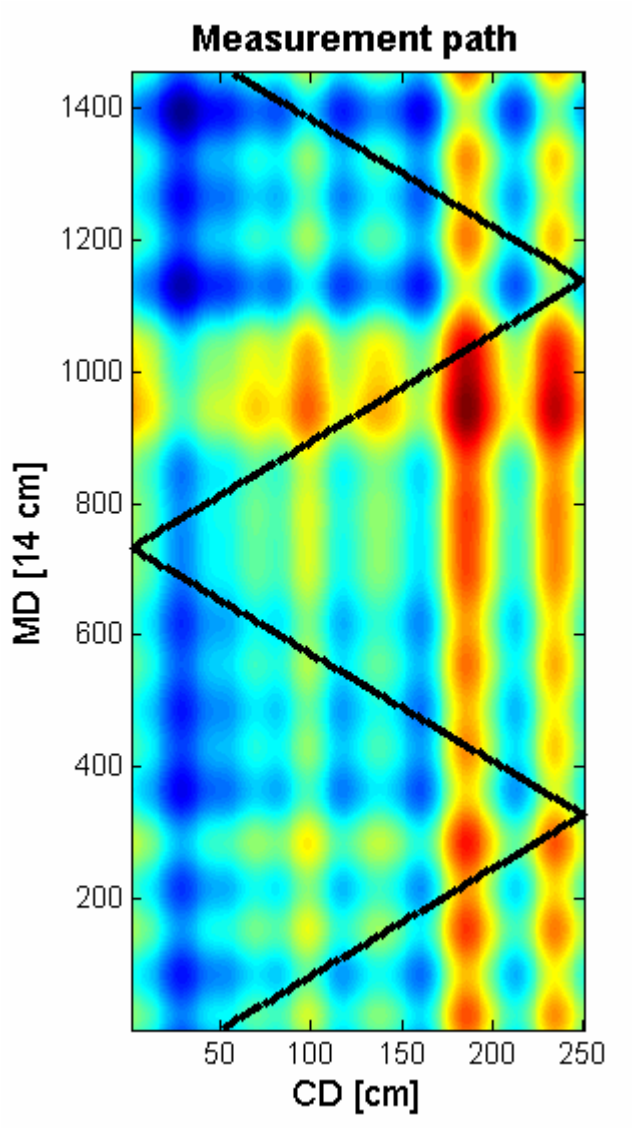


Figure 6.4: One loop of the measurement path used to create the forthcoming results.

The first set of figures shows how the mean value of the amplitudes of the waves behaves. This shows clearly how fast the Kalman algorithm is able to find the waves from the data and how fast the amplitudes stabilize. We use 25 loops to create the following results and since one loop represents 1454 ATPA scans in MD; the total length of the dataset is 25 *1454 = 36350 scans, that is, 5089 meters.

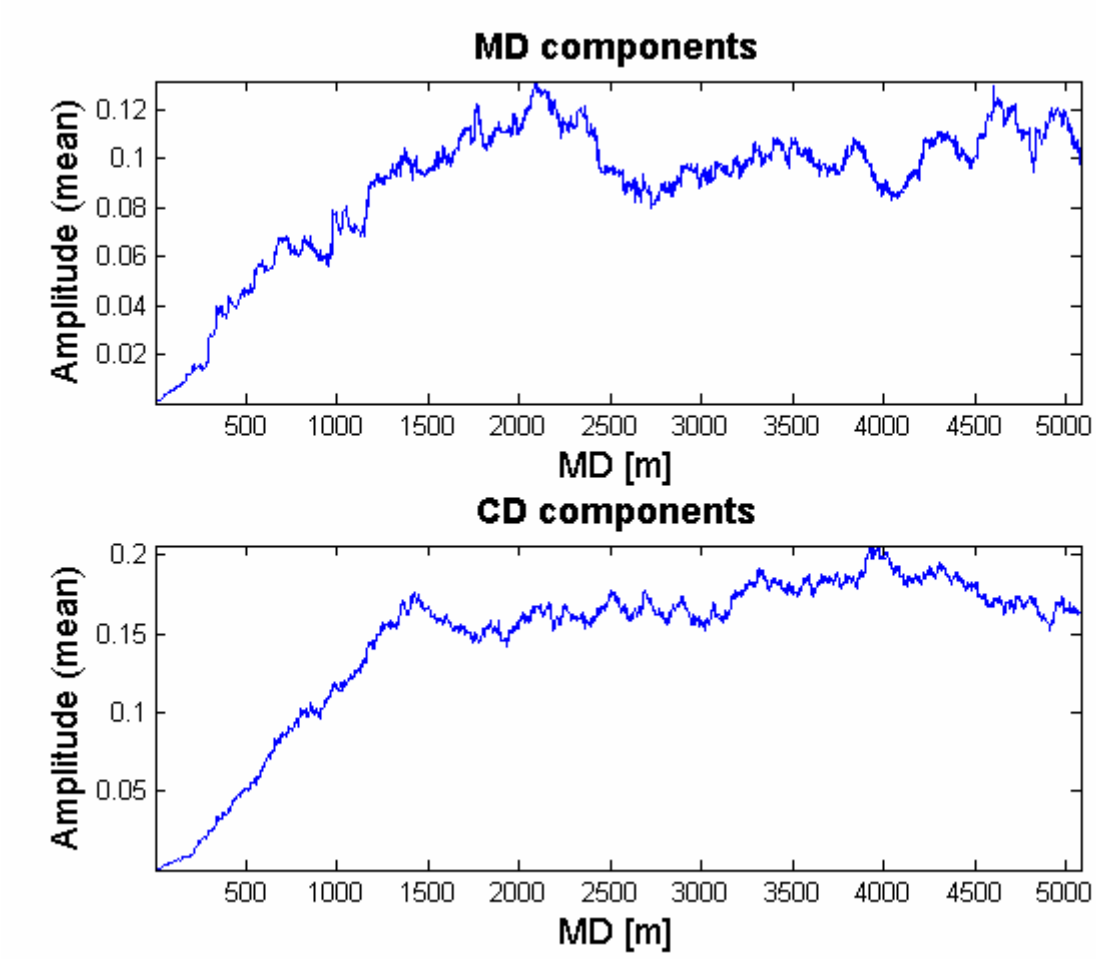


Figure 6.5: The mean value of the CD and MD Fourier components during 5 km of traversing data.

From these figures, we are able to approximate the stabilization speed of our Kalman algorithm, that is, how long does it take for the algorithm to find the most important variations. As mentioned before, we start the analysis by setting all the components to zero. Then, after running the Kalman algorithm loop 25 times, we get to see how fast the algorithm updates the variables. It can be seen that it takes around 1500 m in CD (that is, 26 online

scans) and a bit longer in the MD for Kalman algorithm to stabilize. After this, the mean value of the amplitudes varies between 0.15 – 0.2 for CD and around 0.1 for MD.

The stabilization speed also depends on the noise level of the measurements. As mentioned earlier, the lower the noise level, the more we trust our measurements and the more effect they have on the model. So, by setting the noise level lower we are able to reach the real profiles faster. The downside is, however, that the algorithm becomes unstable. Instability emerges naturally because of the fact that the noise level is too low, and the white noise part of the data is affecting the behavior of the algorithm, and it starts to oscillate easily. The noise levels used in this chapter are chosen in a way that the algorithm is as stable as possible, at the expense of the reaction speed.

Now, after seeing the behavior of the mean value of the 8 components in both directions, let's concentrate on the components individually. How does each of the components behave? The Figure 6.6 presents the amplitude of each of the components individually during the same ~5 km run as in the previous Figure 6.5 of the mean values. The data may look a bit messy at first, but some phenomena can still be observed.

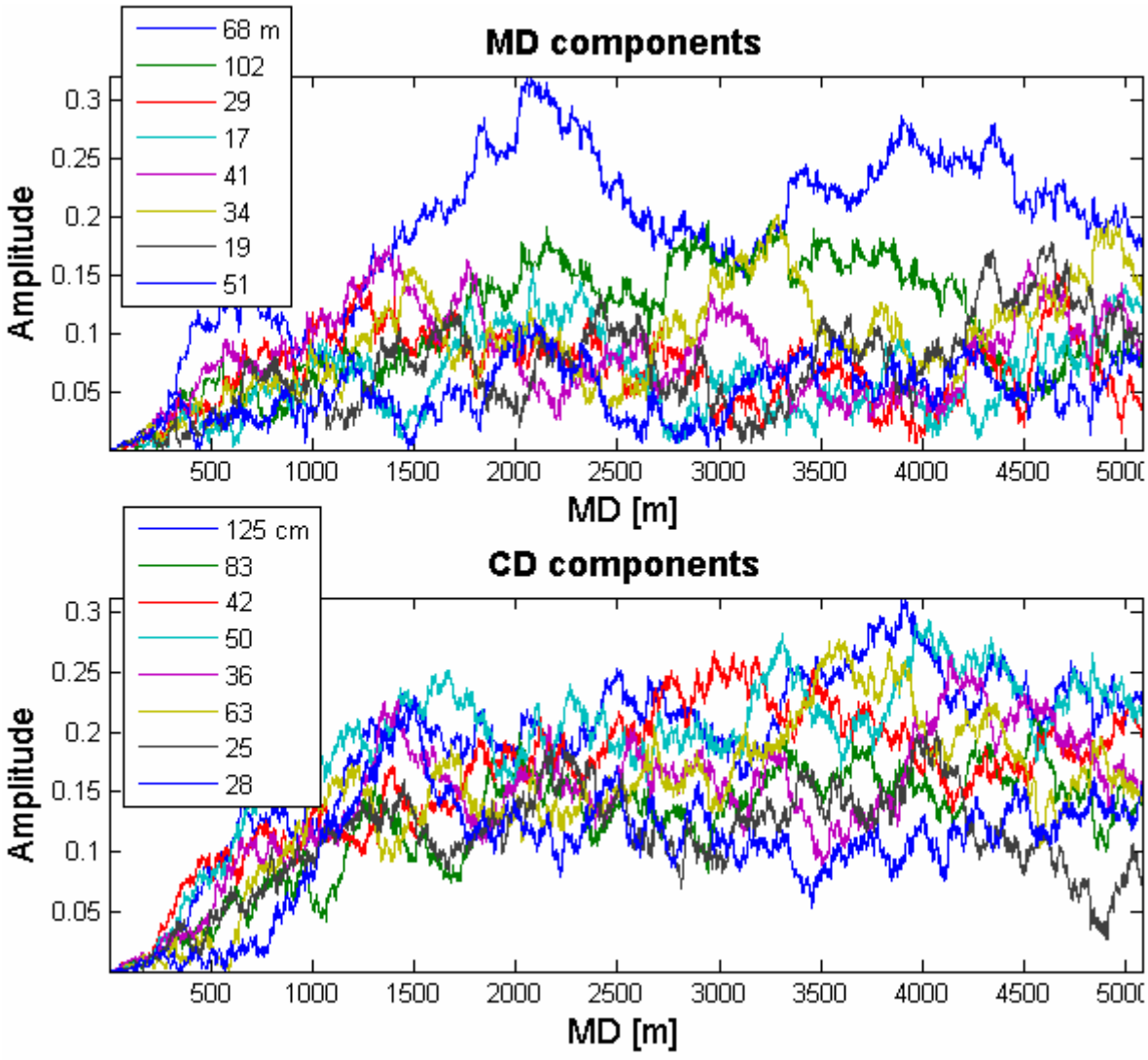


Figure 6.6: The amplitudes of the Fourier components in CD and MD during 5 km of traversing data. In the upper figure, the upper blue line represents the 68 m wavelength. Similarly in the lower figure, the upper blue line represents the 125 cm wavelength.

From this Figure 6.6 we can see, first of all, that the variance of a single component is high. One component can vary even 50% of its absolute value. That makes the controlling and applying the algorithm more difficult. From the QCS point of view, it makes the controlling of the quality variable challenging. The challenge here is that how do we know which kind of variation of the Fourier components is significant for the control system? Some kind of additional filtering is needed for this. This filter would tell the control system whether the variation is in the range of the normal variance or a control action has to take place. On the other hand, the mean value of the components is rather stable, so maybe that would be a more reliable way of recognizing the significance of the changes in the quality variable measurement data.

Another thing worth mentioning from the Figure 6.6 is the accuracy of the stabilized value of each of the components compared to its real value. The real values are calculated directly from the dataset using the Fourier transform. The highest amplitudes seem to belong to the wavelengths 125, 50, and 42 cm in CD and 102, 68, and 34 m in MD. For comparison, the highest real amplitudes are 125, 83, 50, 42 for CD, and 102, 68, 29, 17 for MD. So it seems that we have a match, at least two of the most notable waves are the same, and the others very close. This is also verified in the following chapter, where the mean value of 10 runs of the algorithm is shown to reduce the noise. This confirms our earlier conclusion that the Kalman algorithm is indeed able to recognize the most significant variation correctly.

We have so far concentrated only on stationary quality variable data that stays the same all the time, but what happens when we modify the data suddenly? Is the algorithm able to detect the change? One interesting feature of the Kalman filter algorithm is its ability to react to sudden changes in the data. That is now tested by implementing a cross directional artificial wave into the data in the middle of the run. In real life this could represent for example a control effect, which actually doesn't happen this fast, but for testing purposes we are close enough. This wave has a known wavelength of 50 cm (Figure 6.7) which will allows us to check whether the algorithm is able to react to the artificial wave correctly. The correct way here means that the Kalman filter should be able to modify the corresponding Fourier component (with the wavelength of 50 cm) accordingly.

The following set of figures is made in the following way: First we run the algorithm long enough (that is, around 2.5 km in MD) to stabilize the Fourier components. Then the artificial CD wave is added into the data and the algorithm is run until 5 km. We follow especially the

corresponding component with 50 cm wavelength. To reduce the high variance of the components and to make the figures clearer, this procedure has been carried out 10 times and the mean value of ten runs is shown.

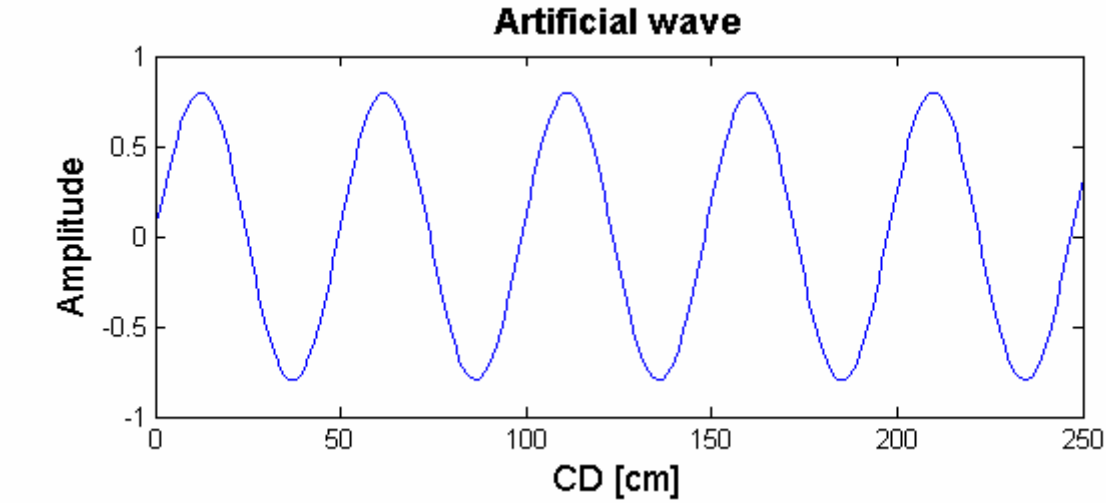
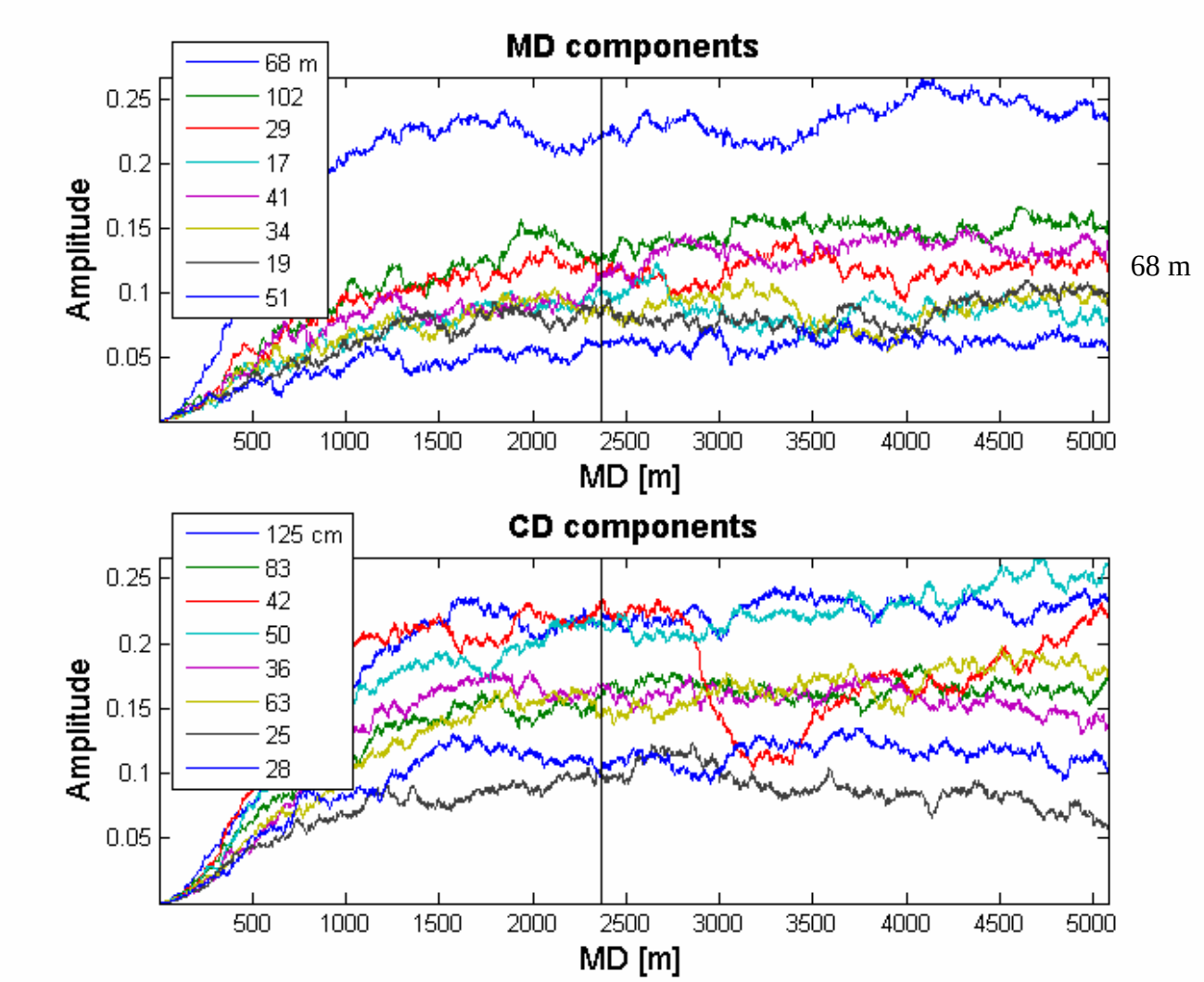


Figure 6.7: The artificial CD wave added to the quality variable dataset in the middle of the 5 km run.



68 m

Figure 6.8: The mean value of the amplitudes of the Fourier components during ten times of the 5 km run of the traversing scan. The black vertical line in the middle refers to the point where the artificial wave has been added. Note the drop of the red line in the CD component figure.

The Figure 6.8 shows the mean value of 10 consecutive 5 km runs of the Kalman filter algorithm. According to the Figure 6.8, the artificial CD component added to the data does appear directly in the Fourier component space: The red line, representing the wavelength of 42 cm, drops dramatically a few scans after the alteration. The drop is clearly due to the adding the artificial wave and the wavelength is very close to the correct 50 cm. The red line returns to its original level after a few kilometers. This is probably due to the calibration process to the new CD profile (Figure 6.9). The MD components or the other CD components have not altered above the noise level. Thus the Kalman filter algorithm seems to be able to recognize the artificial wave correctly although one cannot see the effect anymore after a few kilometers.

Also, as seen in the following Figure 6.9, the algorithm calibrates itself according to this new CD profile. So, even though the componentwise analysis (Figure 6.6) may look messy, the algorithm works as it's supposed to, and the quality control system should not face any problems due to the high variance of a single component.

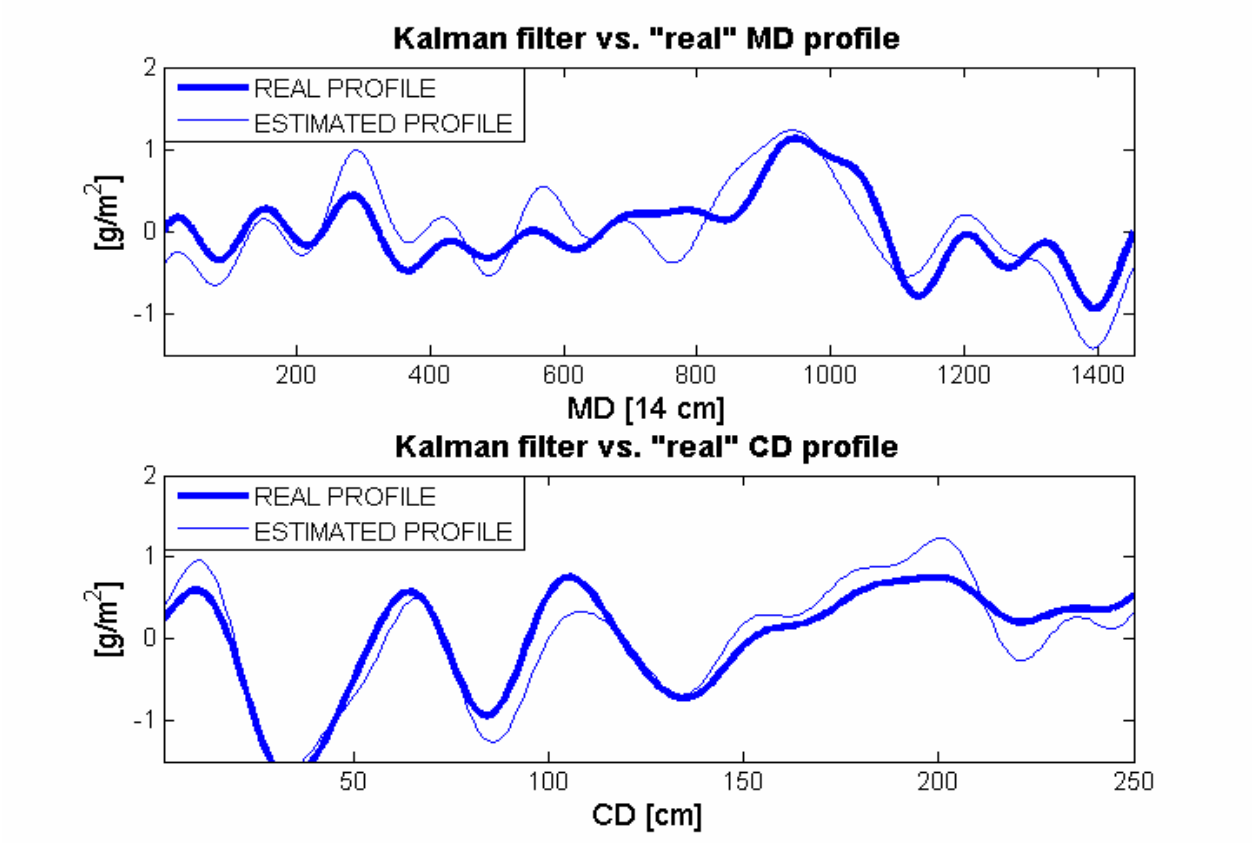


Figure 6.9: The estimated profiles after the implementation of the artificial wave. Notice the altered CD profile compared to previous ones.

7 Conclusions

The goal of this thesis was to build up a new method for data assimilation and the separation of CD and MD of quality variable data. We obtained a short amount of real quality variable data, which was used to test and tune the algorithm. We imitated an online scanner by collecting the measurements from a zigzag path in a dataset trying to reconstruct the original data's CD an MD variations based on those measurements.

We utilized a separation algorithm based on the Kalman filter with two dimensional Fourier frequency components as the state. This is a novel way to do the CD/MD separation. The algorithm was run through the short amount of data with different measurement paths.

The results were very promising. The most significant CD and MD variations were reconstructed even with a very sparse measurement path. The sparser the measurement path, though, the worse is the accuracy of the separation. Also the reaction speed of the algorithm was tested. It was concluded that it requires tens of online scans for the algorithm to stabilize onto a certain level when starting from zero. Analyzing the frequency components individually, it was noticed that the variance of a single component is very high, although the mean value of all the components was stable.

There were some downsides as well. A big issue was the lack of data; 200 m of paper web is not even close enough to carry out a waterproof analysis. Also the stability of the algorithm was causing problems, although this was reduced by setting the noise level of the measurements high enough. Too low noise levels very easily caused the algorithm to oscillate.

For future work, it would be necessary to test the algorithm with longer samples of real data and with more realistic control effects.

All in all, the Kalman-Fourier algorithm is an interesting newcomer to be applied to the old challenge of the CD/MD separation. As shown in this thesis, it has great potential of being later on a part of the quality control system of a paper or board machine.

References

1. Grewal, Mohinder S., Kalman filtering: theory and practice using MATLAB, Wiley, cop. 2001. ISBN 0-471-39254-5
2. Kreyszig, E, Advanced Engineering Mathematics, 8th edition
3. Auvinen H., Haario H., Kauranne T., Optimal Approximation of Kalman Filtering With Temporally Local 4D-Var In Operational Weather Forecasting; article in Realizations of Teracomputing (Kreitz and Zwiefelhofer, eds.)World Scientific 2005
4. Wang, Dumont, Davies: Modelling and Identification of Basis Weight Variations in Paper Machines, IEEE transactions and control systems technology, vol 1 no 4 December 1993
5. Welch, Bishop: http://www.cs.unc.edu/~welch/kalman/ , 30.09.2006
6. Fourier Methods, Pearson Education limited, 2006. ISBN-13 978-1-84479-595-6. 3 chapters from Advanced Modern Engineering Mathematics by Glyn James, ISBN 0-13-0454287
7. Mathworld: http://mathworld.wolfram.com/ , 21.11.2006
8. Wikipedia: http://en.wikipedia.org/wiki/Fourier_series , 27.12.2006

   http://en.wikipedia.org/wiki/Discrete_Fourier_transform , 27.12.2006
9. Ptak, Piotr: Methods for the analysis of two dimensional measurement data of paper web, Diploma Thesis, Lappeenranta University of Technology, 2006
10. Dumont, G., Ball, J., Emmond, G., Guillemette, M., Improved CD control Using Wavelet-based MD/CD Separation, In proceedings Control Systems February 6-10 2004, Quebec City, Canada
11. Gopaluni, B., Dumont, G., Davies, M., Loewen, P., MD-CD Separation, Mapping and Model Identification in a Paper Machine: A discussion Paper, Control Systems June 6-8 2006, Tampere, Finland. ISBN 952-5183-26-2
12. Mäkelä, M., Manninen, V., Heiliö, M., Myller, T., Performance Assessment of Automatic Quality Control in Mill Operations, Control Systems June 6-8 2006, Tampere, Finland. ISBN 952-5183-26-2
13. Mäkelä, M., Paperin laatusuureiden mittaus ja säätö, Suomen Automaatioseura, Helsinki 2003. ISBN 952-5183-20-3
14. Leiviskä, K., Papermaking Science and Technology, Fabet Oy Helsinki 1999. ISBN 952-5216-14-4