Estimating Paper Sheet Process Variations from Scanned Data Using Compressive Sensing

by

Parisa Towfighi

B.Sc., The University of British Columbia, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2011

© Parisa Towfighi 2011
Abstract

During paper fabrication, system actuators are used to control paper properties over the entire sheet on the basis of a restricted set of measurements taken from the moving sheet. From these measurements it is necessary to estimate the properties of the full paper sheet. The axis perpendicular to the direction of motion of the paper through the machine is referred to as the cross direction (CD), and the axis of the sheet motion itself is the machine direction (MD). Low pass filtering is commonly used in industrial practice to separate the slow vibrations in the cross direction from the typically faster variations in machine direction. Exponential low pass filtering can reconstruct the actual variations if uniform sampling has been carried out at a sampling rate that is at least twice the bandwidth of the original signal. Such conditions are almost never met in practice. To overcome this limitation, we propose here a novel algorithm based on the well-established theory of compressive sensing.

The bandwidth constraints of conventional sampling can be avoided if certain general characteristics of the unknown signal are assumed to be known, and if a few computational conditions are satisfied. Compressive sensing can then estimate the signal with impressive accuracy from a minimal number of samples.
Abstract

This new technique requires that samples are collected in a random fashion. In this thesis, compressive sampling is applied to paper machine data. The data representation is optimized in an l1 basis and its resulting performance is evaluated using both industrial and simulated data. It should be noted that this approach has broad potential industrial application in situations where process constraints dictate the timing and location of available process data that is to be used for control and monitoring purposes.
Table of Contents

Abstract ............................................................. ii

Table of Contents .................................................. iv

List of Tables ........................................................ viii

List of Figures ........................................................ ix

Acknowledgements ..................................................... xii

Dedication ............................................................... xiii

1 Introducing the Thesis ............................................. 1
  1.1 Problem Statement ............................................. 1
  1.2 Motivation ...................................................... 3
  1.3 Overview of the Paper Machine ............................... 5
    1.3.1 Sensing ................................................... 5
    1.3.2 Actuators ................................................ 7
  1.4 Paper profile Estimation Techniques ........................ 9
    1.4.1 First Order Low Pass Exponential Filtering .......... 10
    1.4.2 Wavelet Based Methods ................................. 15
# Table of Contents

1.5 Contributions ................................................. 19
1.6 Thesis Overview ............................................. 20

2 CS Theory ....................................................... 21
  2.1 An overview of Compressive Sensing ......................... 21
    2.1.1 Coherence ............................................ 23
    2.1.2 Restricted Isometry Property .......................... 24
    2.1.3 Minimum Number of Samples ........................... 25
    2.1.4 Basis Pursuit Without Noise ......................... 27
    2.1.5 Basis Pursuit With Denoising, LASSO ................. 28
  2.2 Compressive Sensing and the Paper Sheet ...................... 30
    2.2.1 Simulating a Paper Profile ........................... 31
    2.2.2 Signal Reconstruction Error Calculation ............... 34

3 Compressive Sensing Implementation ............................ 36
  3.1 Outline of Implementation Steps ............................ 36
  3.2 Simulating Paper Properties ............................... 40
  3.3 Random Projection Matrix .................................. 41
  3.4 Sparco and Spgl1 Operations ................................ 43
  3.5 Machine Direction and Cross Direction Variations .......... 44
  3.6 Simulated Data ............................................ 46
    3.6.1 Minimum Number of Sensors Needed .................... 48
    3.6.2 Overcoming the Nyquist Rate by Compressive Sensing 50
    3.6.3 Noise Performance ................................... 56
    3.6.4 Random Speed Sensors vs Random Sampling ............ 62
  3.7 Industrial Data ............................................ 68
Table of Contents

3.7.1 Honeywell Basis Weight and Moisture Data ............. 68
3.7.2 Industrial Moisture Data from METSO ............... 74
3.8 Advantages and Disadvantages of Compressive Sensing ... 79
  3.8.1 Advantages ............................................ 79
  3.8.2 Disadvantages ........................................... 79

4 Compressive Sensing in Practice ............................... 81
  4.1 Profile Reconstruction in Realtime Using a Solver ....... 81
  4.2 Future Work: Implementing Random Sampling ............. 83
    4.2.1 Stationary Array of Sensors ....................... 83
    4.2.2 Very Rapid Scanning Sensors ..................... 83
    4.2.3 Multiple Random Speed Sensors ................. 83
  4.3 Future Applications ....................................... 85

5 Conclusion .................................................... 86

Bibliography ..................................................... 88

Appendices

Appendix A - Programs ........................................ 94
  A. 1 METSO Automation ..................................... 94
  A. 2 Exponential Filter ..................................... 97
  A. 3 Scanner .................................................. 97
  A. 4 Wavelet ................................................ 98
  A. 5 Simulated Paper ........................................ 99
Table of Contents

A. 5.1 Simulated Signal with Random Speed Scanner . . . . . . . . . . . . 99
A. 5.2 Simulated Data ........................................... 103
A. 5.3 Simulated Data with Varying Sigma .................. 107
A. 6 Honeywell Industrial Data ................................... 111
  A. 6.1 Basis Weight .............................................. 111
  A. 6.2 Moisture .................................................. 114
  A. 6.3 Moisture Data in a Loop ............................... 117

Appendix B - Least Squares Method .......................... 122
  B. 1 Least Squares Method ................................. 122
List of Tables

3.1 Distribution of MD frequencies (3) . . . . . . . . . . . . . . 44
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Simplified sketch of a paper machine, ([26])</td>
<td>11</td>
</tr>
<tr>
<td>1.2</td>
<td>Scanner traverses the rolling sheet and gathers data in a zigzag path, ([1])</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>Low pass filter</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Original simulated paper sheet, test case 1</td>
<td>46</td>
</tr>
<tr>
<td>3.2</td>
<td>Randomly selecting samples at the marked locations.</td>
<td>47</td>
</tr>
<tr>
<td>3.3</td>
<td>Error with a varying sample size, test case 1</td>
<td>48</td>
</tr>
<tr>
<td>3.4</td>
<td>Original simulated paper sheet with noise, case 2 and case 3</td>
<td>49</td>
</tr>
<tr>
<td>3.5</td>
<td>Error, noisy signal with the Basis Pursuit method, case 2</td>
<td>50</td>
</tr>
<tr>
<td>3.6</td>
<td>Error, noisy signal using BPDN, case 3</td>
<td>51</td>
</tr>
<tr>
<td>3.7</td>
<td>Simulated paper sheet when the Nyquist condition is not met</td>
<td>52</td>
</tr>
<tr>
<td>3.8</td>
<td>Simulated paper sheet when the Nyquist condition is not met with noise, SNR=28.89</td>
<td>53</td>
</tr>
<tr>
<td>3.9</td>
<td>Relative error when the Nyquist condition is NOT met using exponential filtering and compressive sensing in the Fourier basis</td>
<td>54</td>
</tr>
<tr>
<td>3.10</td>
<td>Simulated paper sheet when the Nyquist condition is met</td>
<td>55</td>
</tr>
</tbody>
</table>
List of Figures

3.11 Relative error when the Nyquist condition is met using exponential filtering and compressive sensing in the Fourier basis, SNR of noisy signal is 26.1222 ........................................ 55
3.12 Original noisy signal ............................................. 56
3.13 Recovered signal, using Fourier basis with and without de-noising ...................................................... 58
3.14 Error with increasing sigma values ............................. 59
3.15 The noisy signal with 2 periodic signals ......................... 59
3.16 Error with increasing sigma ...................................... 60
3.17 The MD and CD are separated from the recovered and original signals .................................................. 61
3.18 Original Signal .......................................................... 62
3.19 Path of two random speed sensors .............................. 63
3.20 Error with 184 samples and 2 random speed sensors ...... 64
3.21 Error with 184 completely random samples ................. 65
3.22 CD and MD variation in the recovered Signal with two random speed scanners ..................................................... 66
3.23 CD and MD variation in the recovered Signal, completely random samples ....................................................... 67
3.24 Basis weight measurements are collected at 4096 grid points. 69
3.25 Error calculated comparing the recovered and industrial basis weight data ...................................................... 70
3.26 CD recovered from industrial basis weight data ............... 71
3.27 Moisture measurements are collected at 4096 grid points ... 72
List of Figures

3.28 Error calculated comparing the recovered and real moisture data .......................... 72
3.29 CD recovered from industrial moisture data ................................. 73
3.30 Moisture values of the paper signal, 256 × 256, normalized ....... 74
3.31 recovered signal, 256 × 256 .................................................. 75
3.32 Moisture values of the paper, 256 × 512, normalized ........... 76
3.33 recovered signal, 256 × 512 .................................................. 77
3.34 recovered MD and CD ......................................................... 78

B. 1 Paper with ones in the middle and x where samples are taken 122
B. 2 Paper Profile ............................................................... 125
B. 3 reconstructedPaper ....................................................... 125
B. 4 reconstructedPaper3d ..................................................... 126
I would like to sincerely express my gratitude to my professors, Dr. Guy Dumont and Dr. Michael Davies for their support and guidance. I would also like to thank my committee members Dr. Bhushan Gopaluni and Dr. Philip Loewen for their valuable feedback and advice.

I would like to acknowledge and thank my colleagues Angshul Majumdar, Tanaya Guha and Mustafa Fanaswala for pointing me in the right direction. I would like to thank Dr. Herrmann for lectures on image reconstruction. This work would not have been possible without data collected by Danlei Chu and Johan Backstrom from Honeywell Process Solutions. I would like to thank Dr. Calvin Fu from Metso Automation for guidance and data on random sampling. I would like to thank the pulp and paper center and the department of electrical and computer engineering of UBC as well as my office members Xin Mei, Masita and Jaime for their encouragement.
Dedication

I would like to dedicate this work to the people who raised me since childhood, my parents, Arezou Pouria and Saeed Towfighi, and my grandparents, especially my grandmother, Batul Razavi, also known as Iran Pouria, may she rest in peace. No words can describe how much they have cared for me. I would also like to thank my husband, Paulo J.S. Pereira and my brothers, Siyavash and Sohrab, for bringing joy into my life.
Chapter 1

Introduction and Background

1.1 Problem Statement

A paper machine produces a paper sheet of about 6 to 10 m wide at speeds as high as 120 km/hr. The sheet is wound onto a large roll and later split into smaller rolls of dimensions appropriate for printing or other purposes.

A simplified schematic diagram of a paper machine is shown in Fig. 1.1.
1.1. Problem Statement

As the paper is being produced, its quality must be monitored and control adjusted to maintain the main properties within their specific requirements.

Among the properties of the sheet that affect its quality significantly are moisture, weight and thickness. To control these quantities, their values need to be measured. Industrial sensors are available which can measure these properties on the rapidly moving sheet. Considering the sheet of paper as a two dimensional grid, with one dimension being the width of the sheet (cross direction) and the other perpendicular to it (machine direction), we can assign a numerical value corresponding to a paper property, such as weight, to each cell in the grid. Each cell in the grid is therefore corresponding a data point. However, due to the high cost of industrial sensors, and the large size of the manufactured sheet, it is not feasible to collect data from all the grid points on a sheet. To overcome this problem, a limited number of points are measured, and from the measurements, property values at the unknown grid points are estimated. There are various techniques available for doing this type of estimation and interpolation. In this thesis, compressive sensing is used - a novel technique that can estimate an unknown signal from relatively few samples as long as certain general properties of that signal are known, and a few conditions are satisfied.
1.2 Motivation

In brief, the main motivation for proposing compressive sensing is the hope of reconstructing the paper profile from a small sample size with increased accuracy over the already available methods. Other estimation methods rely on the signal being of limited bandwidth. It is well known that in order to successfully represent a continuous function by a uniformly sampled set of values, the rate at which the sampling occurs must be at least twice the highest frequency in the spectrum of the original signal. If the highest frequency in the spectrum is denoted by $f_s$, then $2f_s$ is the Nyquist frequency. If the Nyquist rate limit is not satisfied, aliasing will occur and the spectrum of the sampled signal will include frequency components not present in the original signal. Aliasing makes it impossible to discern the spectrum of the original signal solely from the samples. In other words, there is a minimum sampling rate for successful reconstruction.

In practice however, process variable signals are not bandlimited. A filtering scheme is thus necessary in practice to bandlimit the signal and to establish an upper bound for frequencies of significance. Practical filters are not capable of completely discarding higher frequencies and some high frequency components will remain. Also, filtering and wavelet methods separate lower CD frequency components via various thresholding methods, and typically average the MD variations. These methods return estimates with the same dimensions as the measured data. However, compressive sensing estimates the entire sheet which usually has a larger dimension than the measured points. If the threshold value is not set properly or changes during
1.2. Motivation

manufacturing, it becomes difficult to estimate the properties of paper.

With new intelligent random speed scanners, it is important to use an estimation algorithm that utilizes random sampling. Random sampling minimizes MD and CD cross over [25] and compressive sensing effectively exploits the benefits of random sampling.

The compressive sensing technique being proposed in this thesis is able to reconstruct signals with fewer samples than required by the Nyquist rule, if the original signal can be represented using only a small number of non-zero coefficients (for example Fourier coefficients) in some arbitrary function space. Also, a minimum number of samples need to be collected randomly and not at uniform intervals. These requirements of compressive sensing will be explored in detail in chapter 2. The compressive sensing method makes signal reconstruction from a smaller sample size feasible and thus avoids the Nyquist limit.
1.3 Overview of the Paper Machine

Modern paper machines must maintain rigorous control of the quality of the manufactured sheet. A dilute mixture of water and wood fibres, enters the paper machine at the headbox as shown in Fig. 1.1.

The headbox processes the pulp slurry so that the wood fibers are uniformly distributed across the machine [31]. The pulp is carried by a wire mesh and by means of gravity, heat and vacuum, its moisture is partially extracted. The moisture content is further reduced in the press and dryer sections by heated rollers and other drainage elements. Sensors for estimating the sheet properties are normally mounted near the end of the dryer section. For detailed discussion of the paper machine, please refer to [10],[38].

1.3.1 Sensing

Properties of papers are normally measured by a scanning sensor that traverses the moving sheet at up to 2000 points ([31]). The data gathered by the sensors are used in controlling and monitoring the quality of the paper being produced. The success of the feedback or feedforward control at each stage of paper production depends on multiple factors including the ability of the control system to track the control variables accurately. Other factors include control delay, measurement accuracy and accuracy of the system model ([31]).

It is not practical to measure paper properties over the entire sheet. However, for quality control, paper properties of the entire sheet need to be estimated. The more information is available about the paper properties,
the better the control action becomes. The process variables over the entire sheet must be estimated based on the points which were actually sampled by the sensor.

Typically sensors take about 20 seconds to traverse the sheet. Since the paper is moving forward while the sensor is traversing the width of the sheet, measurements are gathered along a zig-zag path over the sheet (Fig. 1.2). Note that the actual path of the measurements on a stationary sheet would be at a very shallow angle (typically less than 1 deg) with the machine direction. Based on the speed of the roll and its size, CD measurements are much closer to each other in distance than MD measurements. Only the points that lie on the scanner path are measured, all other locations need to be estimated.

The main properties measured and to be controlled are basis weight (mass of fiber per unit area), moisture (percentage of water in the overall mass of paper sheet) and caliper (thickness) ([34],[10]). After they are taken along
the above-mentioned path, the sensor measurements must be mapped into both CD and MD directions ([23]) before being used by CD and MD controllers that manipulate either equally spaced actuators distributed in the CD direction for CD control, or MD actuators such as the stock valve for weight or the dryer steam pressure for moisture. Mapping, or alignment, of CD actuators to CD data measurement points is also an issue. In certain situations, such as when the sheet wanders, or undergoes shrinkage at the sides, the location of the CD data point corresponding to a CD actuator shifts. To correct this, there are estimators that calculate the error as discussed in [1].

1.3.2 Actuators

CD basis weight is often adjusted by changing the flow of fibre from the headbox by using actuators that manipulate the slice lip. The pulp slurry flow that is between 1-6 cm deep passes through a slice (gap) with an adjustable top lip. This top lip can be bent up or down via CD actuators. The larger the actuator opening, the more fibers are passed through, resulting in a heavier sheet at the corresponding CD actuator location.

A second approach to CD weight control adjusts the fibre distribution by diluting the flow from the headbox at various CD positions.

Dilution actuators alter the consistency of pulp and when fiber concentration goes down, basis weight goes down. The dilution actuators can have a very small actuator spacing (35mm) and a narrower spatial response than slice lip actuators.

Sheet moisture content is controlled by steam boxes, rewet showers, or
1.3. Overview of the Paper Machine

infrared heaters.

At the start of the wet end of the paper machine, the mixture contains 99% water and 1 percent fiber. The mixture leaving the wet end contains 40% fibre and 60% moisture. At the end of the dryer section, the sheet typically is made up of 95% fiber and 5% water.
1.4 Paper profile Estimation Techniques

There have been many studies of paper profile estimation techniques over the past several decades. Kalman filtering to track MD variation, along with a model of the system has been studied in [44], [33] and [16]. Exponential low pass first order filtering, is the common industrial practice and will be discussed shortly. Wavelet reconstruction techniques have also been examined as noted in [34] and [46], that rely on different thresholding methods to denoise profile variations, including soft and hard thresholding. Hard thresholding sets all coefficients below a threshold value to zero. Soft thresholding also sets the coefficients less than the threshold to zero, but it also shrinks the other coefficients by the threshold value. There are many other thresholding methods that can be used for denoising as well.

The discrete cosine transform with coefficient thresholding has been implemented in [29]. The DCT method uses the assumption that CD variations are smaller than MD variations and are twice the scanning frequency. The lower coefficients separate the lower frequencies which are expected to be in the CD direction. The drawback of these thresholding methods is that they are not able to detect any possible small variations in MD.

All methods represent attempts to infer complete CD and MD variations from a set of sparse and irregular sampled data points. The method of uniform sampling is commonly applied but seldom justified.
1.4. Paper profile Estimation Techniques

1.4.1 First Order Low Pass Exponential Filtering

Underlying the decomposition of the scanner measurements into orthogonal CD and MD variations is the assumption that the CD profile is nearly time invariant. Current industrial practice is to use exponential low pass filtering along the MD for each CD location ([23]) in the hope of separating the relatively steady CD value from a rapidly changing MD variation.

The output of the filter is then taken as the CD profile estimate while typically the scan average is used as the MD estimate. This is far from ideal since there is no information available about the points which were not scanned.

Despite a number of studies that have shown that exponential filtering is suboptimal ([28], [17], [27]), and have proposed alternative schemes ([29] and [20]), exponential filtering is still the industrial norm.

A major problem with the use of conventional filtering methods on the scanned data is that the sampling frequency must be at least twice the highest frequency in the bandwidth of the original signal. However, due to practical issues such as measurement failure, sensor calibration and machine failure, sampling may not be uniform. Also, real signals are not perfectly bandlimited.

Indeed, when MD variations contain a frequency equal to the regular scanning frequency, it becomes impossible to properly separate CD from MD variation ([3]) as will be illustrated in the next chapter. This may have dire consequences as wrong CD and MD estimates can then be used as inputs to CD and MD controllers, and possibly result in control instability.
The low pass first order exponential filter is given as follows in the Laplace domain.

\[
\frac{Y(s)}{U(s)} = \frac{K}{1 + s\tau}
\]  

(1.1)

Here, \(K\) is the filter gain and \(\tau\) is the filter time constant. We use figure 1.3 to derive a recursive formula for a low pass RC exponential filter.

It is important to remember a few basic properties of capacitors in order to understand how an RC circuit can act as a low pass filter. The impedance of the capacitor is given by \(\frac{1}{sC}\) in the Laplace domain. By substituting \(jw\) for \(s\), with \(j\) being \(\sqrt{-1}\) and \(w\) being the frequency in radians, the impedance becomes \(\frac{1}{jw}\).

At frequencies close to zero, the impedance of a capacitor is infinite and it acts as an open circuit, \(V_{out}(s) = V_{in}(s)\). As the frequency increases, the impedance of a capacitor decreases and the current starts to flow through it. At very large frequencies, the impedance of the capacitor is almost zero and it acts as a short circuit [35] with \(V_{out}\) being grounded. The series RC circuit in figure 1.3 is a low pass filter.

Based on Ohm’s law, the difference between the input voltage, \(V_{in}(t)\) and the output voltage, \(V_{out}\), is the energy dissipated by the current, \(I(t)\) in the resistor, \(R\).

\[V_{in}(t) - V_{out}(t) = RI(t)\]  

(1.2)

The current through the capacitor in the time domain is \(I(t)\) and is given by

\[I(t) = C \frac{dV_{out}}{dt}\].  

(1.3)
1.4. Paper profile Estimation Techniques

By combining the above two equations, the voltage difference becomes

\[ Vin(t) - Vout(t) = RC \frac{dVout}{dt}. \]  

This equation can be written in the discrete form, with \( n \) as the discretization index.

\[ Vin_n - Vout_n = RC \frac{Vout_n - Vout_{n-1}}{\Delta T}. \]  

By defining the filter time constant, \( \tau \) as \( RC \), the filter cutoff frequency, \( w_c \) is the reciprocal of the time constant and is equal to \( 1/RC \).

\[ RC = \tau \]  

The output voltage can be written in terms of the input contribution and
1.4. Paper profile Estimation Techniques

inertia from the previous input.

\[ V_{out_n} = V_{in_n}(\frac{\Delta T}{\tau + \Delta T}) + V_{out_{n-1}}(\frac{\tau}{\tau + \Delta T}) \]  

(1.7)

The resulting recursive equation of a low pass first order exponential filter is

\[ v_{out_n} = \alpha V_{in_n} + (1 - \alpha)V_{out_{n-1}}, \text{ where } \alpha = \frac{\Delta T}{\tau + \Delta T} \text{ and } 0 \leq \alpha \leq 1. \]  

(1.8)

\[ \tau = \Delta T\left(\frac{1 - \alpha}{\alpha}\right) \]  

(1.9)

Now we have a recursive equation for a low pass filter. In what follows, it is assumed that scanning is centered and regular. In practice, there may be irregular pauses between scans due to calibration of sensors or process interruptions. All of the samples that lie on the scanner path are fed to a low pass filter. For measurement points that lie on the center line of the sheet, there is equal spacing between measurement points. All other scanner paths have unequal spacing between consecutive measurement points. This is because of the zigzag nature of the sensor’s path over the sheet, please refer to figure 1.2. You will notice that samples obtained close to the edges arrive in pairs and do not allow reconstruction with the precision available from samples on the center line [4]. To correct this, we modify \( \alpha \) to vary depending on the distance of the scanner path to the center line. The new \( \alpha \) would be defined as \( e^{-T_{wc}} \), where \( T \) is the scan time and \( w_c \) is the cutoff bandwidth. The scan time \( T \) is varying to account for the varying
time interval between measurements. The filter factor changes between two values FF1 and FF2 at measurement points along the scanner path. Each filter factor has its own scan time.

\[ FF = e^{-Tw_c} \quad \text{FilterFactor} \quad (1.10) \]

For the even numbered scans:

\[ R_n(k) = FF_1.S_n(k) + (1 - FF_1)R_n(k - 1) \quad (1.11) \]

For the odd numbered scans:

\[ R_n(k) = FF_2.S_n(k) + (1 - FF_2)R_n(k - 1) \quad (1.12) \]

For more information on recursive filtering, please refer to [4].

The filter is good at constructing variation in the cross direction. All of the points along the scanner path (all samples with the same CD coordinate number) are fed to an exponential low pass filter.

When the frequency in the machine direction is equal to or greater than the sampling frequency, the low pass filter fails to reconstruct variations in the machine direction. This failure can result in a halt in paper production [10]. To overcome these restrictions, a new algorithm based on compressive sensing is used in this thesis to reconstruct the entire paper profile.
1.4. Paper profile Estimation Techniques

1.4.2 Wavelet Based Methods

The classic Fourier transform has been used to decompose signals based on sine and cosine frequencies.

There are many different Fourier transforms, such as the Discrete Fourier Transform, DFT, the fast Fourier transform, FFT, and the Discrete Cosine Transform, DCT. The Discrete Cosine Transform, which builds on the Fourier transform has been implemented in [29] and applied to the variations on a paper sheet. The coefficients were computed and then thresholded to separate MD and CD variations. The wavelet transform is similar to the Fourier transform in the sense that it allows us to represent a signal based on coefficients that show the contribution of various wavelet basis functions. While Fourier is only about the frequency domain, wavelets are localized in both time and frequency domains. Wavelets are designed to gather information over different resolutions. Also, a wavelet transform produces fewer coefficients around a discontinuity than the Fourier transform. The following equations are taken from [45]. A wavelet is defined by

$$w_{(j,k)}(t) = 2^{-j/2}w(2^{-j}t - k).$$ (1.13)

The function $w(t)$ is the mother wavelet and the translation is done by $k$, $-\infty < k < \infty$, within the wavelet function. The set $w_{(j,k)}(t)$ denotes the baby wavelets, which are scaled and time shifted. In this case, $j$, $-\infty < j < \infty$, is the scaling factor and can be any integer. The wavelet basis, containing $w(2^{-j}t - k)$ is an orthogonal basis. The wavelet function has a zero mean and a 2-norm of one.
The wavelet coefficients can be obtained from the discrete wavelet transform.

\[ c_{j,k} = \int_{-\infty}^{\infty} x(t) \phi_{j,k}(t) dt. \]  

(1.14)

From the coefficients, the signal can be recovered by the inverse discrete wavelet transform.

\[ x(t) = \sum_j \sum_k c_{j,k} w_{j,k}(t) \]  

(1.15)

Define a space \( W_j \) to be a set of all signals, \( x(t) \), which can be synthesized from the wavelets \( w_{j,k}(t) \). The signal \( x_j(t) \) is in the space \( W_j \).

\[ x(t) = \sum_{j=-\infty}^{\infty} x_j(t), \text{ where } x_j(t) = \sum_{k=-\infty}^{\infty} c_{j,k} w_{j,k}(t). \]  

(1.16)

Similarly, consider another wavelet, \( \phi(t) \), given by

\[ \phi_{(j,k)}(t) = 2^{j/2} \phi(2^j t - k). \]  

(1.17)

The scale is \( 1/2^j \) and the position is \( k/2^j \). The signal \( x_j(t) \), synthesized from \( \phi_{j,k}(t) \), is in the space \( V_j \).

\[ x(t) = \sum_{j=-\infty}^{\infty} x_j(t), \text{ where } x_j(t) = \sum_{k=-\infty}^{\infty} c_{j,k} \phi_{j,k}(t). \]  

(1.18)

Also, we can define \( V_j \) to be the set of all signals, \( x(t) \), which can be syn-
1.4. Paper profile Estimation Techniques

thesized from the baby wavelets $w_{i,k}$ where $i < j$ and $-\infty < k < \infty$.

$$x(t) = \sum_{i=-\infty}^{j-1} \sum_{k} c_{i,k} w_{i,k}(t)$$ (1.19)

The spaces $V_j$ are nested and $V_{j+1} = V_j + W_j$, or

$$x(t) = \sum_{i=-\infty}^{j} \sum_{k} c_{i,k} w_{i,k}(t) = \sum_{i=-\infty}^{j-1} \sum_{k} c_{i,k} w_{i,k}(t) + \sum_{k} c_{j,k} w_{j,k}(t).$$ (1.20)

We can also write $V_j = W_{j-1} + V_{j-1}$.

$$x(t) = \sum_{m} cA_0(m) \phi_{j,m}(t) = \sum_{k} cA_1(k) \phi_{j-1,k}(t) + \sum_{k} cD_1(k) w_{j-1,k}(t)$$ (1.21)

The wavelet signal can be approximated by $A_1(t)$ and $D_1(t)$, approximation and detail coefficients at first level. $A_1(t)$ can be further decomposed into $A_2(t)$ and $D_2(t)$.

$$x(t) = A_1(t) + D_1(t) = A_2(t) + D_2(t) + D_1(t)$$ (1.22)

The $D_i(t)$, in $W_{-i}$, is detail at level $i$, and $A_i(t)$, in the space $V_{-i}$, is the approximation at that level.

We can write the approximation and detail coefficients in terms of the wavelet basis functions.

$$cA_1(k) = \sum_{n} \langle \phi_{j-1,k}(t), \phi_{j,n}(t) \rangle cA_0(n)$$ (1.23)
1.4. Paper profile Estimation Techniques

\[ cD_1(k) = \sum_n \langle w_{j-1,k}(t), \phi_{j,n}(t) \rangle cA_0(n) \] (1.24)

The wavelet analysis method has been used to separate the smaller coefficients from the larger ones in order to separate the typically smaller variations in CD from the larger MD variations. The method has been used for denoising too by discarding the coefficients corresponding to high frequency noise. The wavelet method has many applications in data and image compression. For more information, the interested reader is encouraged to refer to [45].
1.5 Contributions

The significant contributions of this thesis are:

1. The compressive sensing method is applied to the paper machine to estimate paper process variables from a sampled data set.

2. The accuracy of the compressive sensing method is assessed and expected error values are calculated with simulated data.

3. The compressive sensing technique is compared to the common industrial filtering practice and it is shown that compressive sampling is potentially an improvement over Nyquist sampling.

4. The performance of the compressive sensing technique is assessed with industrial data as well. Different sampling schemes are discussed.

5. New techniques for recursively implementing the compressive sensing method in real time are examined.
1.6 Thesis Overview

Compressive sensing is the technique being proposed in this thesis as a possible alternative to current industrial practice. The paper making process, the nature of the problem, and the paper model were discussed in this chapter. The theory of compressive sensing and the conditions that make reliable reconstruction feasible will be explained in detail in chapter 2 and its performance will be validated using both computer simulated and industrial data in chapter 3. The common industrial practice which is exponential low pass filtering will be compared to the compressive sensing method.

The limitations of the compressive sensing method, as well as its advantages, will be assessed. An overview of the signal reconstruction process will be given. Ideas on how to implement the algorithm in real time at a reasonable speed will also be presented. Finally, a conclusion and suggestions for future work are included.
Chapter 2

Compressive Sensing Theory

2.1 An overview of Compressive Sensing

The Shannon theorem states that for accurate reconstruction, the sampling rate must be at least twice the highest frequency component of the signal for its effective reconstruction. Twice the highest frequency component is known as the Nyquist rate. Compressive sensing theory claims that if we know some general properties of the signal such as sparsity, we can reconstruct the signal from fewer measurements and have a reconstruction better than what was promised by the Shannon theorem. To use compressive sensing successfully, we need to meet two fundamental conditions: sparsity and incoherent sampling ([11]). Compressive sensing exploits the property that many signals are sparse or compressible in a certain basis. Consider a signal $s$ in the form of an $n \times 1$ matrix column vector:

$$s = \Psi x. \quad (2.1)$$

Here, $x$ is the $n \times 1$ matrix column vector of weighing coefficients and the $n$ columns of matrix $\Psi$, namely, the vectors $\Psi_1, \Psi_2, ..., \Psi_n$, are orthonormal.
2.1. An overview of Compressive Sensing

At every index $i$, the coefficient $x$ is given by $x_i$. In more detail,

$$ s = \sum_{i=1}^{n} x_i \psi_i. $$

Here $x$ is the coefficient sequence of $s$, given by

$$ x_i = \langle s, \psi_i \rangle $$

The notation $\langle ., . \rangle$ is an inner product. A signal $s$ may be sparse in some orthonormal representation basis $\psi$, such as Fourier or wavelets. The signal $s$ is $K$-sparse if it has $K$ nonzero coefficients out of $n$ coefficients. It is compressible if the representation has just a few large coefficients and many small coefficients.

Consider the signal $s$ being sampled via a $p$ by $n$ measurement matrix $\Phi$.

$$ y = \Phi s $$

Here, $p$ is the number of samples and $n$ is the dimension of $s$, in other words, the total number of points that can be measured. The unknown signal $s$ is a column vector of $n$ elements and the data set is also a column vector, of $p$ elements. The problem is about reconstructing the unknown signal $s$ from a set of samples. Basic linear algebra states that for successful reconstruction we need $p \geq n$. However, in practice, many times we have an ill-posed problem, meaning that $p < n$. The compressive sensing theory states that even when $p < n$, if $s$ is sparse, it is possible to seek a solution to the problem.
2.1. An overview of Compressive Sensing

One example of a sparsifying representation basis $\Psi$ is the Fourier transform, which allows us to sparsify signals of a sinusoidal nature and other periodic signal. Other transforms, such as wavelet transforms, are suitable for sparsifying different signals. When the Fourier transform is applied to the signal in the time domain, we obtain the coefficients in the frequency domain (36). The goal is to reconstruct the signal $s$, given only $p$ samples in the Fourier domain

$$y_p = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t e^{-i2\pi w_p t/n},$$

(2.5)

here the measured frequencies $w_p$ are the known subset of all frequencies $0, ..., n - 1$. The $n \times n$ discrete Fourier transform matrix is the basis function matrix, $\Psi$, [12].

The implication of sparsity is that we can ignore the small coefficients and still be able to approximately reconstruct the signal. However, there are still some questions that need to be considered. What conditions should the measurement matrix satisfy? Can the solution be reliable? What happens in the presence of noise?

2.1.1 Coherence

The coherence, $\mu$, is one of the conditions of compressive sensing and is given by the equation below, measures the maximum correlation between any two columns of the basis matrix, $\psi$, and the measurement matrix, $\phi$.

$$\mu(\Phi, \Psi) = \sqrt{n} \max_{k,j} |\langle \phi_k, \psi_j \rangle|$$

(2.6)
2.1. An overview of Compressive Sensing

When $\Phi$ and $\Psi$ are orthonormal, the minimum coherence is 1 and the maximum coherence is $\sqrt{n}$. One pair with maximal incoherence involves the spike basis given by $\psi_k(t) = \delta(t - k)$ and the Fourier basis, $\phi_j(t) = n^{-1/2}e^{2\pi j t/n}$ in which case $\mu(\Phi, \Psi) = 1$ ([15]).

Compressive sampling is interested in incoherent basis pairs, in other words, pairs with minimal coherence. A random measurement matrix $\phi$ is usually incoherent with any sparsifying basis $\psi$, [15]. Incoherence allows for more information to be captured by a smaller sample size about the underlying nature of the signal to be reconstructed. Consider a signal $s$ being sampled by a projection matrix $\Phi$, and corrupted with additive noise:

$$y = \Phi s + w.$$  \hspace{1cm} (2.7)

Here $y$ is the measurement vector, $w$ is the noise, and $\Phi$ is the projection matrix. The projection matrix contains information about how the samples were gathered in the domain of the original signal $s$. Since we do not know the position of the sparse coefficients, it can be shown that a sufficient condition for recovery is the restricted isometry property.

2.1.2 Restricted Isometry Property

For each integer $K = 1, 2, ..., n$ define the isometry constant $\delta_K \in (0, 1)$ of a $p$ by $n$ matrix, $\Phi$ (with orthonormal columns) as the smallest number such that

$$(1 - \delta_K)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_K)\|x\|_2^2 \quad (2.8)$$
holds for all $K$-sparse vectors $x$. We say that a matrix $\Phi$ has the restricted isometry property (RIP) of order $K$ if $\delta_K$ is not too close to 1 (typically less than 0.5). Generally, one method of obtaining a matrix $\Phi$ that satisfies the RIP is to use a random matrix. Another condition necessary for satisfying the RIP is having enough samples, see [13]. We use the notation $\Phi$ in $RIP(K, \delta_K)$ to denote $\Phi$ satisfies the RIP of order $K$ with $\delta_K$. When this property is satisfied, the $K$-sparse vectors cannot be in the null space of $\Phi$. The RIP states that the columns of $\Phi$ are nearly orthogonal. The columns cannot be exactly orthogonal since there are more columns than rows [15].

Depending on the sparsity of the signal in a given basis, the number of measurements that is likely to suffice for an acceptable reconstruction varies. However, in many practical situations, it is impossible to know the sparsity of a signal from just a few samples with certainty and as a result, it becomes difficult to know how many samples are needed.

### 2.1.3 Minimum Number of Samples

In theory, see [7], the number of measurements needed is $M$, where

$$M \gg c\gamma \log \frac{n}{\gamma}.$$  \hspace{1cm} (2.9)

Here, $\gamma$ is the sparsity level of the signal. The total number of points available in the original signal is denoted by $n$. The oversampling factor is denoted by $c$ and $c > 1$. For more information on how to select a proper oversampling factor, please refer to [9]. Equation (2.10) implies that the sparser the signal, the fewer measurements are needed for successful recon-
2.1. An overview of Compressive Sensing

In certain situations the sparsity rate varies with time and it becomes difficult to prove mathematically that the RIP has been met. Here are some mathematical theorems about conditions that help us meet RIP with high probability \([8], [37]\).

**Theorem 2.1.1.** Independent and identically distributed matrices: Let \( \Phi \) be a measurement matrix with entries that are independent and identically distributed from the following zero mean distributions with variance \( 1/p \) \([8]\),

\[
\phi_{i,j} \overset{i.i.d}{\sim} N(0, 1/p)
\]

\[
\phi_{i,j} \overset{i.i.d}{\sim} \begin{cases}
1/\sqrt{p} \text{ with probability } 1/2 \\
-1/\sqrt{p} \text{ with probability } 1/2 \\
\sqrt{3/p} \text{ with probability } 1/6 \\
0 \text{ with probability } 2/3 \\
-\sqrt{3/p} \text{ with probability } 1/6
\end{cases}
\]

For any \( K, \delta_K \in (0, 1) \), and \( c_1 < \delta_K^2 (3 - \delta_K)/48 \).

Set \( c_2 = \frac{192 \log(12/\delta_K)}{36\delta_K^2 - \delta_K - 48c_1} \) Then whenever \( p \geq c_2 K \log n \), \( \Phi \in \text{RIP}(k, \delta_k) \) with probability exceeding \( 1 - \exp(-c_1 p) \).

Both the RIP and the incoherent basis pair condition can be met with high probability by selecting a random projection matrix. Let us define a matrix \( H \) \([2]\) as the product of the projection matrix and the sparsifying representation basis and call it a regression matrix:

\[
H = \Phi \Psi.
\]
2.1. An overview of Compressive Sensing

If the projection matrix contains independent and identically distributed random variables from a Gaussian probability density function with zero mean and variance of $1/p$, with $p$ being the number of samples, and if there are enough samples available, the regression matrix $H$ satisfies the RIP ([7]).

With this notation, any noisy measurements can be expressed as follows:

$$y = \Phi s + w = \Phi \Psi x + w = Hx + w$$ (2.11)

If we know the sampled data set $y$, and the projection matrix $\Phi$, we can use a representation matrix $\Psi$ to obtain $H$ and then recover the signal coefficients $x$. From the coefficients we can then reconstruct the original signal $s$. This task can be accomplished via multiple algorithms. $L^1$ optimization is suitable for signals that are sparse. Let us introduce the basis pursuit method, which is a simple exact recovery algorithm.

2.1.4 Basis Pursuit Without Noise

The basis pursuit method minimizes the coefficients $x$ such that the condition $Hx = y$ holds.

$$\min \|x\|_1 \quad s. t. \quad Hx = y$$ (2.12)

where $\|x\|_1 = \sum_{i=1}^{n} |x_i|$. This problem is convex and can be solved as a linear program. It can be shown that the basis pursuit method is robust to measurement noise and discretization error as long as the oversampling rate, $c$, is large enough ([9]). For more information about the basis pursuit linear convex problem and the conditions under which useful reconstruction
2.1. An overview of Compressive Sensing

is feasible, please refer to [6]. The following theorem applies to signals that are not exactly sparse, but compressible.

**Theorem 2.1.2.** Noiseless Recovery theorem: *Suppose the columns of the matrix $H$ have unit 2-norms and $H \in \text{RIP}(2K, 0.3)$. Then the solution vector $x_1$ obtained via basis pursuit satisfies*

$$\|x_1 - s\|_2^2 \leq \frac{c_0}{K} \|x - x_K\|_1^2$$

(2.13)

here, $x_K$ is the vector formed by setting all but the $K$ largest entries of $x$ to zero and $c_0$ is a constant given by $c_0 = 4(1 + \delta_{2K})^2$, see [6], [14]. There are solvers available which use recursive linear optimization algorithms ([43]) to find the coefficients $x$ given the regression matrix and the sampled data.

2.1.5 Basis Pursuit With Denoising, LASSO

We will now consider what happens to the compressed sensing algorithm in the presence of noise in the measured signal. The noisy situation is very similar to the noise free situation except that exact recovery is not possible ([6]).

If there is a known level of noise in the signal, it is possible to have basis pursuit with denoising (BPDN). In this case a new parameter called $\sigma$ is introduced that corresponds to the standard deviation in the noise.

$$\min \|x\|_1 \quad s. t. \quad \|Hx - y\|_2 \leq \sigma$$

(2.14)

BPDN is a quadratically constrained linear program. The same problem
can be formulated differently when there is an estimate of how much noise is present in the coefficients of the original signal. In this case, one can include a tolerance parameter called $\tau$ to minimize the coefficients while accounting for noise. This new method is called the Least Absolute Shrinkage and Selection Operator (LASSO), see [40].

$$\min \| y - Hx \|_2^2 \quad s. t. \quad \| x \|_1 \leq \tau \quad (2.15)$$

LASSO is a quadratic program. Convex analysis can show that a solution to this problem is a minimizer of $x$. In practice it can be difficult to know the optimum value of the denoising parameter.

The compressive sampling problem is normally solved in a 1-norm because this norm is most likely to find a sparse unique solution with the highest reliability. The 1-norm is the sum of all coefficients and a good estimator of the sparsity, which in turn makes solving the optimization problem computationally possible. The 2-norm has also been used as a way to solve the compressive sensing problem. However, it is not as reliable in finding a minimized solution ([6]).

The solver used in this work is spgl1 ([41]) employed in conjunction with Sparco ([42]). Sparco is a toolbox for creating linear operators from which new sparse recovery problems can be assembled. It acts as an interface to the Spgl1 solver. Both of these tools were developed at The University of British Columbia.
2.2 Compressive Sensing and the Paper Sheet

By thinking of a sheet of paper as a grid and recording a scalar paper characteristic such as weight at each grid point, we get a matrix. Let us denote this matrix by $s$. We can reshape this matrix to become a vector.

In the paper mill, only certain data points on the sheet can be sampled and the unsampled points need to be estimated for quality control. The known elements of $s$, are the gathered measurements and are denoted by $y$. The MD and CD location of the sampled data is known and collected in a projection matrix, $\Phi$, leading to the relationship

$$\Phi s = y. \quad (2.16)$$

The matrix $\Phi$ is the projection, also called the measurement matrix, and contains the location of the samples. When the projection matrix $\Phi$ is applied to the column vector $s$, the sampled set is obtained. To clarify this process, a very simple example is presented here. Let us assume that there are 3 data points in the MD and 2 data points in the CD.

$$s_{\text{matrix}} = \begin{bmatrix} .1 & .2 & .3 \\ .4 & .5 & .6 \end{bmatrix}$$

There are a total of 6 elements in the original signal $s$. For this example, suppose 2 samples are needed. Based on the number of samples needed, we generate random numbers in a row vector called $M$. The minimum value of these numbers is 1 and the maximum value is equal to the total number of elements in the original column vector $s$, in this case 6. In this case, suppose $M = [5, 2]$. We initialize a projection matrix $\Phi$ with as many rows
2.2. Compressive Sensing and the Paper Sheet

as the number of samples needed. For every sample, we set one element in every row of the projection matrix $\Phi$ equal to 1. The column index of that element is taken from the row vector $M$.

Suppose that we need 2 samples from our example signal at random. We reshape the paper properties matrix to become a column vector $s$ and sample it.

\[
\begin{bmatrix}
0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0
\end{bmatrix} \times
\begin{bmatrix}
.1 \\
.2 \\
.3 \\
.4 \\
.5 \\
.6
\end{bmatrix} =
\begin{bmatrix}
.5 \\
.2
\end{bmatrix}
\]

This example illustrated how the projection matrix represents the location of sampled points.

2.2.1 Simulating a Paper Profile

To verify the performance of the compressive sensing algorithm, a simulated paper two dimensional signal needs to be generated. The discrete model of the process is given below ([19], [21], [39]).

\[
s(z) = G(z)u(z) + d(z)
\]  \hspace{1cm} (2.17)

Here, $s(z)$ is the paper variation signal in terms of deviations from steady-state, $u(z)$ is the actuator set-points in terms of deviations from
steady-state, $d(z)$ is a coloured noise and $z$ is the transform variable. $G(z)$ models the dynamics associated with processing, actuation and sensing. $G(z)$ is defined as

$$G(z) = G_0 V(z).$$  \hspace{1cm} (2.18)

$$V(z) = \text{diag}(v_i(z))_{i \times i}, \quad v_i(z) = \frac{z^{-T_d}}{1 - a_i z^{-1}}$$  \hspace{1cm} (2.19)

$v_i(z)$ is the transform of the dynamic response of the process; $T_d$ is the time-delay; and $a_i$ is the open-loop pole. $G_0$ is an $i \times r$ matrix containing the actuator responses at spatial frequency zero and describes the interactions between the inputs and outputs. $i$ corresponds the number of scanned data boxes and $r$ is the number of CD actuators. Since the actuators are identical by design, $G_0$ is a band-diagonal symmetric Toeplitz matrix.

$$G_0 = \begin{pmatrix}
g_1 & g_m & \cdots & g_q & 0 & \cdots & \cdots & \cdots & \cdots & 0 
g_2 & g_1 & g_2 & \cdots & g_q & 0 & \cdots & \cdots & \cdots & \cdots 
\vdots & g_2 & g_1 & g_2 & \cdots & g_q & 0 & \cdots & \cdots & \cdots 
g_q & \vdots & g_2 & g_1 & g_2 & \cdots & g_q & 0 & \cdots & \cdots 
0 & g_q & \cdots & g_2 & \cdots & \cdots & g_q & 0 & \cdots & \cdots 
\vdots & 0 & g_q & \cdots & g_2 & \cdots & g_q & 0 & \cdots & \cdots 
\vdots & \vdots & \vdots & \vdots & g_2 & g_1 & g_2 & \cdots & g_q & \vdots 
\vdots & \vdots & \vdots & \vdots & \vdots & g_2 & g_1 & g_2 & \cdots & \vdots 
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & g_q & \cdots & g_2 & g_1 & g_2 
0 & \cdots & \cdots & \cdots & 0 & g_q & \cdots & g_2 & g_1 & g_2 
\end{pmatrix} \in \mathbb{R}^{r \times r}
$$

The parameters, $g_1, \ldots, g_q$, describe the spatial response to the $i^{th}$ ac-
2.2. Compressive Sensing and the Paper Sheet

tuator. Here, \( q \) corresponds to the half-width of an actuator’s response and \( m \) denotes the number of data boxes between two adjacent actuators. For more information about the paper model, please refer to [19]. \( V(z) \) is the discrete time model with a first order time delay \( T_d \). This model ignores shrinkage at the edges of the roll.

After modeling and generating the paper property signal, we need to sample it. Compressive sensing requires random samples to satisfy the RIP. To gather random samples from our signal, we need to set up the paper model represented by a matrix as a column vector and denote it by \( s \). The total number of elements in this vector \( s \) is equal to the number of MD data points multiplied by the number of CD data points. We need to sample data points at random locations from this column vector \( s \). The traditional low pass exponential filtering method works with samples collected along a zigzag path. After sampling the signal, we need to find a family of basis functions that sparsifies the signal \( s \). The act of sparsifying makes computations easier and makes it possible to estimate the signal using compressive sensing.

The basis functions tested here are Fourier, and three wavelet families: Daubechies, Haar and Symlet ([30]). We define the regression matrix \( H \) as the product of the projection matrix in the spatial Dirac basis and another sparsifying basis such as Fourier or wavelet. Assuming that the measurements on a sheet of paper can be compactly represented using a set of basis functions, the problem of reconstructing the paper variation signal from a small sample set can be formulated as minimizing the 1-norm of the sparse coefficients, \( x \), such that the regression matrix, \( H \), applied to \( x \), yields the
2.2. Compressive Sensing and the Paper Sheet

measured data vector $y$. The regression matrix, projection matrix and the sparsifying basis are all passed to Sparco and wrapped as matrix operators.

With the regression matrix $H$ and the measured data $y$ being known, we can minimize the 1-norm of the coefficient vector $x$. The regression matrix operator and the gathered samples are passed to the Spgl1 solver and the coefficients of the original signal are estimated. These coefficients represent the sheet of paper in a sparsifying basis such as Fourier or wavelet and are then transformed to obtain the paper sheet in the spatial domain.

2.2.2 Signal Reconstruction Error Calculation

The error in the compressive sensing method is calculated using all the tested basis functions in a test situation. To calculate the error inherent in the estimation, the original signal needs to be known, this is not the case in practice. Samples taken from the original signal are used to estimate the entire signal. Then, the estimated signal is compared against the original signal. We calculate the relative error using the following formula.

$$\text{error} = \frac{\|s - s_r\|}{\|s\|} \quad (2.20)$$

Here, $s$ is the original signal and $s_r$ is the recovered signal. We will evaluate the performance and accuracy of the compressive sensing technique using both simulated and industrial data shortly.
The 1-norm and 2-norm of the error are given by the equations below.

\[ \|s - s_r\|_2 = \sqrt{\sum_{i=1}^{n} (s(i) - s_r(i))^2} \]  \hspace{1cm} (2.21)

\[ \|s - s_r\|_1 = \sum_{i=1}^{n} (s(i) - s_r(i)) \]  \hspace{1cm} (2.22)

The 2-norm error is a measure of the distance between the original and recovered signals.
Chapter 3

Compressive Sensing

Implementation

In this chapter, the compressive sensing algorithm will be implemented and verified. The results of applying the method to simulated and industrial data will then be presented.

3.1 Outline of Implementation Steps

In this section, the code that performs compressive sensing on industrial data is broken down into small steps.

Please note that in practice we start from step 3, described in this section, since the entire paper variation signal is unknown. In a practical situation, information about the sampling points is stored in the form of a measurement matrix and measured samples are analyzed to estimate the entire paper variation signal. Also, it is not possible to calculate the reconstruction error since the original signal is not known and can only be estimated from the samples.

The compressive sensing solver used in this work is Spgl1 [41] in conjunction with Sparco [42]. Spgl1 is an optimization solver, using an $l_1$ basis and solves
the compressive sensing problem using different algorithms. Sparco needs
to be used in conjunction with Spgl1 as a way to interface with the solver.
Sparco has been developed to solve sparse optimization problems. Both
Spgl1 and Sparco are MATLAB toolbox libraries, which were developed at
the University of British Columbia and are available for public use.

1. **Generate measurement data/Upload a paper signal:** Define
the number of databoxes in both the MD and CD. Either load the
data already available, or generate a paper variation signal. Noise can
be added to the signal if desired. Refer to subsection 3.2, below, for
information on how to generate a paper signal.

2. **Reshape the paper:** Stack the columns in the paper matrix on top
of each other to obtain a column vector. In this case, the paper matrix
is the signal to be estimated and can be denoted by $s$.

3. **Create a measurement matrix:** The measurement matrix, also
called the projection matrix, has one row for each sampled data point.
Its number of columns equals the dimension of the paper properties
column vector. To generate a measurement matrix, start with a zero
matrix, write a loop which sets a single number to 1 in each row of
the measurement matrix. For more information, please refer to the
section titled random projection matrix. It can be denoted by $\Phi$.

4. **Obtain measurements:** Multiply the measurement matrix by the
original loaded data column vector.

$$y = \Phi s + w$$
3.1. Outline of Implementation Steps

In this situation, the data set, $y$, is obtained by applying the measurement matrix to the paper matrix and $w$ denotes the presence of any noise. In reality, $s$ is the unknown signal to be estimated and any possible noise needs to be filtered out.

5. **Basis function operators:** Create the Fourier transform and wavelet transforms. The basis functions are denoted by $\Psi$.

6. **Regression matrix operator:** Form a product operator between the measurement matrix and the basis function. The result is the regression matrix.

   $$ H = \Phi \Psi $$

7. **Solver:** Run Spgl1 with a desired method. Choose from basis pursuit, LASSO and BPDN. The solver returns the coefficients of the estimated signal.

8. **Spatial basis:** Transform the recovered coefficients to the original basis using the basis function transform.

   $$ s = \Psi x $$

9. **Reshape:** Reshape the recovered signal to match the original 2 dimensional shape.

10. **Error:** Calculate reconstruction error. Find the difference between the original and recovered signals and normalize the difference in a Frobenius manner. Divide your finding by the Frobenius norm of the original signal.
3.1. Outline of Implementation Steps

11. **Separate MD-CD variations:** If required by the controller, use any separation method on the recovered signal to separate the MD and CD variations.

12. **Plot:** Generate all desired figures (optional).
3.2 Simulating Paper Properties

Define three Toeplitz matrices as $E$, $Bt$ and $H$. Both matrices $E$ and $H$ are to be used in creating a white random noise matrix. The matrix $Bt$ corresponds to the actuator dynamics and it will be applied on the actuator setpoint matrix called $U$. We assume 5 actuators with their own dynamics in this simulation. Generate at least one periodic sinusoidal function for the MD at a frequency appropriate for MD variation ($\sin md$). Generate at least one periodic sinusoidal function for the CD at a low frequency appropriate for CD variation ($\sin cd$).

The paper variation signal, $Y$, includes the actuator dynamics applied to the actuator setpoints, as well as a time delay and some noise.

It also contains at least two periodic sinusoidal functions: one in the MD and one in the CD. More sinusoids can be added to both dimensions. The code for generating the paper profile is given in Appendix 1.
3.3 Random Projection Matrix

It is important to gather random samples when using compressive sensing. To satisfy the necessary condition of RIP with high probability, data needs to be collected randomly. Also, there is a minimum number of samples for successful recovery. In certain practical situations, it is not possible to implement random sampling. For example, when uniform speed scanners are used, all the gathered data points fall on a zigzag path and are not randomly collected. It is not advantageous to use random points from each scan, because using all of the collected points always results in a better approximation. To create a random projection matrix, start by creating an array containing random numbers between 1 and $MD \times CD$. Initialize the measurement matrix $MM$ and allocate memory to it.

```matlab
for ii=1:numberofsamples
    MM(ii,M(ii))=1;
end
bpure= MM*original;
```

The vector $M$ contains as many elements as the total number of samples. The value of each element is a random number between 1 and $MD \times CD$, the total number of points in the original paper variation signal. The matrix $MM$ is the projection matrix, also called the measurement matrix. Its number of rows is equal to the number of samples and its number of columns is equal to the total data points available on the sheet. We reshape the original paper variation signal to become a column vector and apply the measurement matrix to it.
3.3. Random Projection Matrix

\[ b = M M * y \]

In this case, \( b \) denotes the measurements, \( M M \) is the measurement or projection matrix and \( y \) is the paper variation signal.

In the section on compressive sensing theory, the measurement matrix was denoted by \( \Phi \), the sampled points by \( y \) and the paper signal by \( s \). By multiplying the MM with the paper signal column vector, we can obtain a sample set.
3.4 Sparco and Spgl1 Operations

In this section, Sparco and Spgl1 operations necessary to solve the compressive sensing problem will be described. The opHaar2d, opWavelet and opFFT2d operators are used to create basis functions. The opWavelet operator is used to create daubechies and symlet operators. The opHaar2d operator is used to create a two dimensional Haar basis function. The opFFT2d operator is the two dimensional fast Fourier transform operator. The basis function operators are converted into a matrix object using ClassOp. The opMatrix is used to turn a measurement or projection matrix into a projection object.

\[
\text{opWW} = \text{classOp}(\text{WW}); \\
\text{AW} = \text{opFoG}(\text{MM}, \text{WW}); \quad % \text{AW} = \text{MM} \cdot \text{WW} \\
[\text{recoveredD}, R, G, \text{INFO}] = \text{spg_bpdn}(\text{AW}, \text{bpure}, 0, \text{opts}); \\
\text{recoveredDd} = \text{opWW} \cdot \text{recoveredD};
\]

The basis function and projection matrix objects are multiplied together via the opFoG operator and the resulting regression matrix object is passed to Spgl1 along with the data set. The solver returns the coefficients of the unknown signal in the domain of the basis function. The coefficients are transformed by the basis function matrix object to yield the signal in the spatial domain. The one dimensional signal is reshaped to be two dimensional.
3.5 Machine Direction and Cross Direction Variations

Due to the zigzag motion of the sensor, data initially contains a mixture of CD and MD measurements that need to be processed for use in control.

For any given MD, the set of the values that constitute the CD variations is collectively known as the profile. The profile is assumed to have zero average value at any time. It is understood that the variations in CD are typically slower than variations in machine direction. They can both be represented as sinusoidal functions of various frequencies. We are mostly interested in the CD estimates, since only 15% of MD frequencies are controllable ([3]).

<table>
<thead>
<tr>
<th>Category</th>
<th>Frequency (Hz)</th>
<th>Period(s)</th>
<th>Percent of total variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25-250</td>
<td>0.004-0.04</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>2.5-25</td>
<td>0.04-0.4</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>0.25-2.5</td>
<td>0.4-4.0</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>0.006-0.25</td>
<td>4.0-167</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>0.0016-0.006</td>
<td>167-600</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 3.1: Distribution of MD frequencies ([3])

Table 3.1 shows an estimated spectral distribution of variation in MD for a typical machine.

CD variation frequencies are usually lower than MD variation frequencies. Since MD actuators and CD actuators are separate, the MD and CD variations obtained from any measurement scheme need to be separated.

There have been many studies done on separating MD and CD variations
3.5. Machine Direction and Cross Direction Variations

and any technique may be used to achieve this task. As was illustrated in [24], using sensors with randomly varying speed reduces the need for MD/CD separation.

A novel method of analysing variation components and residual noise built upon TAPPI T 545 [32] is presented in [24]. After removing average CD and MD variations, data is decomposed into principal variation components and random two-dimensional noise. This method used in conjunction with a variable-speed scanning sensor, removes the aliasing of fast MD components into the slower MD and CD variations. For more information, please refer to [32] and [24].

In this thesis, the compressive sensing method is used to estimate the two dimensional sheet. The recovered signal is simply averaged in both dimensions to separate the MD and CD values.

\[
MD_j = \sum_{i=1}^{i=cd} s_{i,j}
\]

(3.1)

\[
CD_i = \sum_{j=1}^{j=md} s_{i,j}
\]

(3.2)

Here, \(s\) is the paper properties signal. The index \(i\) runs in the cross direction between 1 and the total number of CD points, while the index \(j\) runs in the machine direction between 1 and the total number of MD points. The CD estimate is given as the average of the rows and the MD estimate is the average of the column values.
3.6 Simulated Data

To test the compressive sensing method, a two dimensional array of paper properties is simulated. The simulated paper properties are obtained as a sum of sinusoids; two sinusoids in the machine direction and one in the cross direction. The paper model was simulated based on information described in [22].

The two dimensional paper profile is then reshaped to a row vector. Samples are collected at random and put in a column vector.

Simulated data was tested with basis pursuit and basis pursuit with denoising.

The first test case is a noise free signal with basis pursuit. The signal generated for this test case is illustrated in Fig. 3.1 with 4096 points.

![Figure 3.1: Original simulated paper sheet, test case 1](image)

The signal is sampled randomly at the locations shown in Fig. 3.2.
3.6. Simulated Data

To separate the CD variation from the MD variation, any technique can be used including low pass filtering. A comparison can then be made between the accuracy of compressive sampling and that of other methods.

The second and third test cases both start with a noisy sinusoidal signal. The second test case uses basis pursuit and the third test case uses basis pursuit with denoising.

For various basis functions and randomly collected samples, the entire sheet variation is estimated and the error between the reconstruction and the true signal is calculated. From the error plots (Fig. 3.3, Fig. 3.5, Fig. 3.6), one can conclude that, not surprisingly, increasing the number of measurements results in a better approximation. Out of the different basis functions, Fourier yields the lowest error. This is to be expected, since the signal is mainly sinusoidal and therefore, most sparse in a Fourier basis.

Figure 3.2: Randomly selecting samples at the marked locations.
3.6. Simulated Data

Figure 3.3: Error with a varying sample size, test case 1

3.6.1 Minimum Number of Sensors Needed

The minimum number of sensors needed for a useful reconstruction via the compressive sensing method can be obtained from equation 2.10 which states

\[ M \gg cK \log \frac{n}{K}. \]

The minimum number of sensors for test case 1 described in the previous section may be estimated. In this case, the total number of points in the original signal is \(64 \times 64\). Thus, \(n = 4096\). Based on the paper model, there are two frequencies in the machine direction and one frequency in the cross direction. Therefore, the sparsity number \(K\) is 3. To find a reasonable number for the oversampling factor, we refer to a look-up graph on page 12 of [9]. In this case, the oversampling factor \(c\) can be safely chosen as 9 based on the sample size and according to the look-up graph.

Width the variables \(c\), \(K\) and \(n\), it is possible to use equation 2.10 to
3.6. Simulated Data

Figure 3.4: Original simulated paper sheet with noise, case 2 and case 3

find the minimum number of measurements needed. The result is 195. Since there are 64 data points in the machine direction, with 2 or 3 scanned points per MD location, it is possible to successfully reconstruct the signal for this special test case.

Gathering two or three measurements per MD location can be achieved by using an array of sensors or by mounting three slower scanning sensors. The speed of a scanning sensor relative to the speed at which the paper sheet moves, determines how many measurements can be gathered per MD location. At zero speed, only MD data is gathered. As already been stated, the speed of the paper through the machine is 120 km/hr. The width of the sheet is 10 m. A sensor gathers about 2000 measurements per scan in 20 s. As a result, CD measurements are normally much closer to each other in distance than MD measurements. Increasing the scanner speed can reduce
3.6. Simulated Data

Figure 3.5: error, noisy signal with the Basis Pursuit method, case 2

the distance between consecutive MD points.

3.6.2 Overcoming the Nyquist Rate by Compressive Sensing

As noted above, current industrial practice is to use a scanning sensor to gather sheet data and to feed the scanned data to an exponential filter in order to extract CD and MD variations. The most common filter is a first order filter that separates out the lower frequency CD variation. The mean value of the data obtained in each scan is taken as the estimated MD value. When the sheet is being scanned in a zigzag manner, for measurement points that lie on the center line of the sheet, there is regular equal spacing between successive measurement points. All other measurement points have unequal but periodic spacing between consecutive measurements.
3.6. Simulated Data

Figure 3.6: error, noisy signal using BPDN, case 3

To account for the varying distance between consecutive points, variable weighting coefficients are used. For more information about this adaptive exponential filter, refer to [3]. An alternative low pass filter with a weighting function is discussed in [18].

In this section, compressive sensing is compared with exponential filtering. The paper variation is represented at 512 points in the machine direction and 32 points in the cross direction. Based on this sampling frequency two separate paper profiles are simulated. Two scenarios are tested. In one scenario, the Nyquist limit cannot be met and in the other case, the Nyquist condition is satisfied. Nyquist limit cannot be met on the paper profiles shown in Fig. 3.7 and Fig. 3.8.

Fig. 3.8 depicts the same signal as in Fig. 3.7 with noise added to it.
A bar graph of the error is shown in Fig. 3.9.

When the Nyquist condition is not met, the lowest error is given by collecting random measurements and reconstructing the signal with compressive sensing. Other methods yield high errors in comparison. Out of the remaining methods, the lowest error is given by scanning a noiseless signal and reconstructing it using compressive sensing. Exponential filtering yields very high errors when the Nyquist rate is not met. Let us now consider a situation in which the Nyquist condition is satisfied. A plot of the original paper profile is shown in Fig. 3.10 and then from a set sample size, reconstruction is performed with exponential filtering and compressive sensing.

A bar graph of the error is shown in Fig. 3.11.
3.6. Simulated Data

Figure 3.8: Simulated paper sheet when the Nyquist condition is not met with noise, SNR=28.89

From Fig. 3.11, it can be observed that compressive sensing yields a better approximation than exponential filtering whether the Nyquist criterion is met or not. When the Nyquist rate is met, the lowest error is still given by collecting random measurements and reconstructing the signal through compressive sensing. Other methods yield high errors in comparison.

Out of the remaining methods, the lowest error is given by scanning a noiseless signal and reconstructing it with exponential filtering, followed by the same method with a little noise. When data is collected by a scanner, compressive sensing does not yield a low error. This is due to the fact that when data is not collected randomly, the RIP and incoherence conditions of compressive sensing are not met. By choosing a random projection matrix, both the RIP and incoherence conditions are met and a reasonable approximation is feasible.
3.6. Simulated Data

Figure 3.9: Relative error when the Nyquist condition is NOT met using exponential filtering and compressive sensing in the Fourier basis

The key issue here is to note that any regular periodic scanning and sampling scheme will always be unable to separate MD and CD variations when there are significant frequency components present that are closely related to the scanning or sampling pattern. It is by exploiting random sampling that we are able to break this weakness, and bypass the normal Nyquist bandwidth constraints.
3.6. Simulated Data

Figure 3.10: Simulated paper sheet when the Nyquist condition is met

Figure 3.11: Relative error when the Nyquist condition is met using exponential filtering and compressive sensing in the Fourier basis, SNR of noisy signal is 26.1222
3.6. Simulated Data

3.6.3 Noise Performance

The performance of compressive sensing in the presence of noise distribution is discussed here. A noisy simulated signal has been used with 2 periodic signals in the machine direction and one periodic signal in the cross direction.

![Original noisy signal](image)

Figure 3.12: Original noisy signal

The recovered signal using basis pursuit and basis pursuit with denoising is shown in Fig. 3.13. The denoised recovered signal contains less noise, as expected. However, this improvement is not apparent in the error plot, Fig. 3.14. The first point in Fig. 3.14 shows the error with a $\sigma$ value of zero, which does not denoise and performs exact recovery.

In practice, it is very difficult to have an optimal value for the denoising
3.6. Simulated Data

parameter $\sigma$ since the noise level cannot be known. The result of Fig. 3.14 is inconclusive and no trend is observed in the error as the $\sigma$ varies. Future work on compressive sensing and denoising is needed.

Without denoising the 1-norm error is 129.93 and 2-norm error is 127.62. With denoising, the 1-norm error is 120.94 and 2-norm is 116.25. The signal to noise ratio is 12.08 in this example.

Now, consider the denoising process with a different signal with a simple underlying structure shown in Fig. 3.15. There is a periodic signal in each dimension. Basis pursuit with denoising and different sigma values is applied to the signal and the error is illustrated in Fig. 3.16.

It can be seen in Fig. 3.16 that the error does not follow a trend with varying sigma values.

The MD and CD separation is shown in Fig. 3.17.

Without denoising the 1-norm error is 112.73 and 2-norm error is 34.3. With denoising, the 1-norm error is 112.73 and 2-norm is 41.689. The signal to noise ratio is 32.8.
3.6. Simulated Data

Figure 3.13: Recovered signal, using Fourier basis with and without denoising.
3.6. Simulated Data

Figure 3.14: Error with increasing sigma values

Figure 3.15: The noisy signal with 2 periodic signals
3.6. Simulated Data

Figure 3.16: Error with increasing sigma
3.6. Simulated Data

Figure 3.17: The MD and CD are separated from the recovered and original signals
3.6. Simulated Data

3.6.4 Random Speed Sensors vs Random Sampling

As has been discussed, compressive sensing requires random sampling. The practical implementation of random sampling is considered with simulated data in this section. A random speed scanner and random sampling are compared with similar numbers of samples.

Figure 3.18: Original Signal
3.6. Simulated Data

The measurement matrix is shown in Fig. 3.19. This matrix shows two sensors with random speed, each traversing half of the sheet. The sensors scan a random distance at a random speed during each scan. The sheet is shown in red and the squares are the sampled points. The measurement matrix contains ones at the location of samples and zeros elsewhere.

![Measurement Matrix](image)

Figure 3.19: Path of two random speed sensors

Even though the sampling motion is not completely random, since there is a sufficient number of samples, the compressive sensing algorithm can be applied. It was previously shown that for the signal being reconstructed here, two or three sensors are needed. In this simulation example, two sensors are used. In practice it would be desirable to have three sensors or more, since more measurements result in a better reconstruction.
3.6. Simulated Data

The signal to noise ratio is 14.38. The samples collected by the two random scanners yield a 1-norm error of 195.54 and a 2-norm error of 202.27.

The same signal is used with completely randomly collected samples and yields a 1-norm error of 201.92 and a 2-norm error of 210.67.

It was observed that both random sampling and random speed scanners yield similar results for the same number of samples, between 45 to 50% for 184 samples out of 4096 points.

It can be seen that with the same number of samples, Fourier yields similar results using completely random samples or samples gathered by random speed scanners. However, with completely random samples, most wavelets yield slightly lower error as can be seen in Fig. 3.20 and Fig. 3.21.

The recovered signal is separated into its CD and MD components as shown in Fig. 3.22 (samples of random speed scanners) and in Fig. 3.23.
3.6. Simulated Data

Figure 3.21: Error with 184 completely random samples

tally random samples).
3.6. Simulated Data

Figure 3.22: CD and MD variation in the recovered Signal with two random speed scanners
3.6. Simulated Data

Figure 3.23: CD and MD variation in the recovered Signal, completely random samples
3.7 Industrial Data

3.7.1 Honeywell Basis Weight and Moisture Data

A large scale stationary section of a roll of unbleached paper was tested in an experiment carried out by Honeywell Inc. Detailed machine direction and cross direction measurements were gathered by a specially programmed scanning sensor.

There were five measurements gathered per point in the machine direction via a Honeywell Da Vinci production scanner. The average of the five measurements was used. The sensors measured moisture and basis weight variations. The paper was advanced manually by approximately 100 mm between MD positions, which led to sheet wandering. Sheet wandering can occur on a paper machine in production, however, it would not be as severe as seen in this data set. The CD measurement boxes were 12.18 mm wide.

The compressive sensing algorithm was tested with two sets of industrial data. This industrial data is depicted in Fig. 3.24 and Fig. 3.27 corresponding to the basis weight and moisture content of the roll after averaging the five machine direction scans. Let us examine the basis weight data. The industrial data represents the properties of the entire sheet of paper. To simulate what happens in the paper mill, samples are collected from this sheet. There are 4096 data points in the original industrial sheet. We can choose the number of samples to use for reconstruction.

With only 64 samples, compressive sensing is evaluated. As can be seen in Fig. 3.25, compressive sensing using a Fourier basis yields an impressively low error of about 36%.
3.7. Industrial Data

Figure 3.24: Basis weight measurements are collected at 4096 grid points

The basis functions are all assessed using random spatial sampling. Fourier basis is also tested with sampling by a simulated scanner along a zigzag path. Sampling random points yields a better result as the measurement matrix will satisfy the RIP property with higher probability. Since the variation in paper quality is typically smooth and of a sinusoidal nature, Fourier yields the best approximation. Also, as expected, the error decreases as sample size is increased. The error is calculated by comparing the estimated signal with the true signal, shown in Fig. 3.24.

The compressive sensing method estimates the properties of the entire sheet of paper. Since the actuators are mounted across the width of the sheet, in many situations CD variations are potentially controllable. A filter is used to separate CD and MD variations. The CD portion is fed to the CD actuators and the MD portion is fed to the MD actuators. Fig. 3.26 de-
3.7. Industrial Data

Figure 3.25: Error calculated comparing the recovered and industrial basis weight data

picts the isolated CD variations filtered out from the signal estimated by the compressive sensing technique using a Fourier basis function and a Symlet basis function.

The 1-norm of the error is 30.497 and the 2-norm of the error is 31.26. Next, data that represents moisture variation across the sheet, shown in Fig. 3.27, will be considered.

Then, the error is calculated in a similar way to basis weight. We calculate the error for various sample sizes. With only 64 samples, compressive sensing using a Fourier basis yields a surprisingly low error of approximately 13%. Once we increase the sample size, the other basis functions also result
3.7. Industrial Data

Figure 3.26: CD recovered from industrial basis weight data

In order for the approximation to be useful, we need to separate out the CD variations and feed them to the CD actuators. This can be done in a manner similar to the basis weight situation (Fig. 3.29).

The norm-1 of the error is 46.3349 and the norm-2 of the error is 46.47. The performance of compressive sensing using various basis functions was verified with industrial data and it was shown that compressive sensing is capable of estimating the signal with reasonable accuracy.
3.7. Industrial Data

Figure 3.27: Moisture measurements are collected at 4096 grid points

Figure 3.28: Error calculated comparing the recovered and real moisture data
3.7. Industrial Data

Figure 3.29: CD recovered from industrial moisture data
3.7.2 Industrial Moisture Data from METSO

Compressive sensing was used with data supplied by Calvin Fu from METSO Automation. The results were very satisfactory with 2 rapid sensors gathering 2 to 3 data points per MD location. On a 256 by 256 portion of the sheet, the error obtained via compressive sensing using a Fourier basis is 0.2242, or 22%. Other basis functions do not reconstruct the signal in this case and yield errors close to one.

Figure 3.30: Moisture values of the paper signal, 256 × 256, normalized
3.7. Industrial Data

The recovered signal using random samples with a Fourier basis function is shown Fig 3.31.

The recovered signal was estimated with only 1% of the total data with an error of 22%. The same signal was sampled at a slower rate and only 256 samples were collected. The collected samples yielded an error of 27% using compressive sensing and an error of 92% using exponential filtering.

When a larger portion of the sheet is used, the error is even smaller. With an MD which is twice as large (512 MD points, 256 CD points), the error using a Fourier basis is 0.1894, or 18%.

Figure 3.31: recovered signal, 256 × 256
3.7. Industrial Data

In the following 2 figures, a larger portion of the sheet is considered and estimated.

Figure 3.32: Moisture values of the paper, $256 \times 512$, normalized
3.7. Industrial Data

The reconstructed signal with a Fourier basis, has an error of 0.1894.

Figure 3.33: recovered signal, 256 × 512

The 1-norm of the difference between the original and recovered signals is found to be 37.5640. The 2-norm of the difference between the original and recovered signals is 27.3558.

The error obtained from the Metso moisture data is consistent with the error from the Honeywell moisture data and both are around 20%.

If there is no random sampling, with the same number of samples collected by constant speed scanners, the error is 94%. The 1-norm of the error is 133.4 and its 2-norm is 88.82. The 2-norm is a measure of the difference in...
3.7. Industrial Data

distance between the recovered and original signals and is a good estimate of the difference between the two signals.

Figure 3.34: recovered MD and CD
3.8 Advantages and Disadvantages of Compressive Sensing

In this section, we will discuss the advantages and disadvantages of compressive sensing.

3.8.1 Advantages

Breaking the Nyquist limit is the advantage of compressive sensing. The minimum number of samples is suggested by equation 2.10. The compressive sampling theorem is reliable for compressible signals as long as the conditions of sparsity, incoherence and RIP are met. By exploiting known a priori characteristics of the signal, it may be completely represented by far fewer samples than bandwidth based sampling would require.

3.8.2 Disadvantages

Compressive sensing depends on the conditions of sparsity, incoherence and RIP for successful signal estimation. In order to satisfy the conditions of sparsity and incoherence, some testing and signal analysis may be required prior to the estimation in order to obtain information about the underlying nature of the signal. This information is necessary in choosing a proper sparsifying basis that meets the conditions of sparsity and incoherence. Moreover, it is often difficult to calculate whether the RIP condition is met since the sparsity level of the unknown signal is not known with certainty. If a suitable sparsifying basis is not known, more samples are needed for successful recovery as the sparsity number will be larger in equation 2.10.
3.8. Advantages and Disadvantages of Compressive Sensing

Compressive sensing requires the sampling to be done in a random fashion in order to satisfy the RIP condition. Achieving random sampling is impossible in some applications. For example, it is not possible with a constant speed scanning sensor traversing the full width of the sheet.

There are also implementation challenges depending on the solver and hardware in use. The sample set of the initial signal needs to be a power of two with the Spgl1 solver used in this thesis. The wavelet operators used by Spgl1 and Sparco only accept signals with dimensions equal to powers of 2. When a signal does not fit this size condition, extra zeros can be appended to the matrix, or a truncated signal may be used, to meet the size requirement.

The size of the measurement matrix can get very large. The number of rows in the measurement matrix is equal to the total number of samples collected. The number of columns of the measurement matrix is the product of the number of rows and columns in the original signal. More work on compressive sensing theory and implementation is needed to reduce the size of the measurement matrix. A recursive form for the estimation is desirable.
Chapter 4

Compressive Sensing in Practice

More work needs to be done to implement compressive sensing in real time on an operating paper machine. Faster, more durable and cost effective sensors need to be developed with the concept of random sampling in mind. Compressive sensing can be used in other signal estimation applications within and outside of the pulp and paper field.

4.1 Profile Reconstruction in Realtime Using a Solver

The compressive sampling method used to estimate the entire paper profile needs to be implemented in real time at a paper mill to be useful.

We will now discuss some ideas for the real time implementation. Since the reconstruction is done on a square matrix, every time the roll of paper moves, a column of the matrix is thrown away and a new column is introduced and all the columns in between are shifted by 1. The function calling the solver can utilize a recursive loop with a start and end time to solve the
4.1. Profile Reconstruction in Realtime Using a Solver

estimation problem with spgl1 corresponding to the actual start and stop
times in paper production. This recursive approach has already been tested
as part of this thesis in the appendix section "moisture data in a loop".

To reconstruct the profile in real time, the best approach would be to
use a customized solver adapted to real time use. The changing nature of
the time-infinite signal can then be dealt with as the solution is formed, and
the current solution can act as an initial guess point for the next update.

Furthermore, one can develop a solver that takes into account the model
of the paper machine and the actuator dynamics and uses the most effective
mathematical method to find a unique sparse solution. In the absence of
such a solver, a generic solver can be used recursively.

One approach would be to wait for a sufficient number of new columns of
the paper sheet matrix, for example ten percent of the columns, to run the
solver. Another option is to divide the new columns into smaller submatrices
and then run the solver on the smaller submatrices. The solution to these
submatrices would then be merged into the larger solution.

Finally, the sheet reconstruction data must be separated for use by CD
and MD controllers. The speed of the solver, number of iterations taken
to find the sparse solution and also the speed of the paper machine and
the sensors used, all need to be taken into account. Further work on the
real time implementation of compressive sensing with a customized solver is
suggested.
4.2 Future Work: Implementing Random Sampling

4.2.1 Stationary Array of Sensors

There are non-scanning sensors available that contain a large array of stationary sensors. These sensors can be randomly scheduled to achieve random sampling. A non scanning sensor by Metso Automation, IQInsight, one such commercial product.

4.2.2 Very Rapid Scanning Sensors

It would be possible to use one high speed sensor to collect the minimum number of samples in a random fashion. Current sensors are not fast enough for this purpose. Future work on sensor development and compressive sensing implementation is suggested.

4.2.3 Multiple Random Speed Sensors

The number of samples needed may be determined based on an offline measure of the paper, using equation 2.10. The number of sensors needed to collect the minimum number of necessary samples may then be determined. Place the sensors across the cross direction. Assign each portion to one sensor. Have the sensors cross the width of their own section up to a random distance. In other words, instead of crossing the full width each time, the sensors should only cross a random distance of their section’s width. If the sensors are not capable of scanning a random width of the sheet, more
4.2. Future Work: Implementing Random Sampling

sensors are needed.

A random speed sensor can be used to achieve random sampling. Such sensors have been developed by Metso Automation (PaperIQ Select). This intelligent random speed scanning sensor performs better in avoiding MD-CD aliasing, as shown in [24]. The result of using this sensor was tested with simulated data. Error with random speed sensors with 179 samples out of 4096 points was found using a Fourier basis. Random speed sensing has been discussed in a previous section.
4.3 Future Applications

Compressive sensing is a relatively new technique used for estimating signals from a few samples, and potentially is applicable in many different industries. It is currently used in certain applications such as data compression, image processing and earth science. This technique can be used to reconstruct compressible signals whenever random sampling is possible and certain mathematical conditions are satisfied.

It is important to keep in mind that the conditions of sparsity, incoherence and RIP need to be met with high probability for reliable accuracy. Also, the accuracy of compressive sensing needs to be tested to ensure that it is suitable for the particular application. Some initial testing is required to ensure that compressive sampling is the preferrable signal estimation method for the application in mind. Examples of industries where compressive sensing may be used are pulp and paper, oil and gas, resources, mining, textiles, medicine, material engineering, marketing, economics, music, food, agriculture, and biology.

The signal can be one dimensional or multi-dimensional. Process knowledge can lead to appropriate sparse representation. Compressive sensing can also be used to compress known signals of a large size. The known signal can be sampled and then discarded. The few samples can later recover the original signal. This method is very useful in situations where transmitting and storing data is expensive or challenging.
Chapter 5

Conclusion

The effectiveness of the compressive sensing method with paper sheet data was illustrated and it was shown that this method is successfully able to estimate paper variation using both simulated and industrial data. The data represented variation in paper weight and moisture.

The simulated data was generated using a paper model which included the actuator dynamics and sinusoidal variations in both directions as well as noise. In practice, a different model might be suitable for each mill depending on many different factors including the actuator characteristics and dead time.

The industrial data yielded acceptable reconstruction with the Fourier basis. Since the Fourier basis function resulted in the most accurate estimation, we can conclude that Fourier is a suitable basis function for the paper profile and specifically moisture and basis weight data. However, more work can be done to find out if there is a better basis function which would be similar to Fourier but slightly different.

Compressive sensing requires randomly collected measurements. PaperIQ Select by Metso Automation is an intelligent scanner with randomly varying speed that can be used as an alternative when absolutely random
samples are not possible.

Care must be taken in implementing compressive sensing. Due to the nature of the inverse problem, the Spgl1 solver recovers coefficients for square matrices only and signal interpolation and averaging may be necessary in some cases. Further work is needed on the implementation of compressive sensing on paper machine data in real time. The speed of the paper machine, solver and sensor all need to be considered and customized for each paper machine. Also, it is advisable to test the algorithm with off line data to double check its accuracy and reliability.

In many situations it is important to estimate the properties of a resource based on a small sampled set. Compressive sensing can be applied to other areas of manufacturing. In the resource industry, logging, and mining, compressive sensing can be used as a technique for both data compression and signal estimation. Compressive sampling can also be used as a way of compressing information efficiently for better storage and transmission. A known signal can be sampled and discarded to be later reconstructed from a much smaller sample set.

Based on the results of industrial and simulated data, we can conclude that compressive sensing is suitable for estimating the paper profile.
Bibliography


Appendix A - Programs

A. 1 METSO Automation

% Parisa Towfighi
% #: 36716009

% Introduction to the code:
% In this program, we load a data file. We take random samples
% of the signal Y. To do this, we have a measurement matrix
% called MM. Its number of rows is equal to numberofsamples
% variable and its number of columns is equal to
% MD*CD. We reshape the original signal to be a row vector
% of MD*CD elements. The resulting data gathered is in the
% form of a column vector.
% We wrap the measurement matrix and use it with four basis
% functions of Daubechies, Haar, Symmlet and Fourier. We use spgl1 to
% reconstruct the paper profile using compressive sensing in l1.

clc
close all
clear all

% Due to the nature of spgl1 inverse problem I have a square paper
% with more variation in MD vs. CD.
MD = 256*2;
CD = 256;
Y = zeros(CD,MD); %initialize paper profile without noise

load IQinsightMoisture.mat
Y= pfa2d(1:CD,1:MD);
Y= mat2gray(Y);
levels = 5;
% creating operators with Daubechies, Haar, Symlet and Fourier. Both MD
% CD need to be divisable by 2^ level
mode2 = 2;
WH = opHaar2d(MD,CD,levels);
WS = opWavelet(MD,CD,'symmlet',4,levels); % use symlet wavelets
WD = opWavelet(MD,CD,'daubechies',8,levels); % use Daubechies wavelets
WF = opFFT2d(MD,CD); % use fourier

%figure
%surf(Y);
%title('Original data');
%xlabel('MD');
%ylabel('CD');
%saveas(gcf, 'fuoriginal.jpg');

numberofsamples=MD;
% the range of the #of samples is from MD to MD*6
M= ceil(MD*CD.*rand(numberofsamples,1));
MM= zeros(numberofsamples, MD*CD);
for ii=1:numberofsamples
    MM(ii,M(ii))=1;
end
MSave= M;
original = (reshape(Y,MD*CD,1)); % The 2d profile is now a column vector
bpure = MM* original; % measured points

disp('size of data');
size(bpure) % bpure is the actual samples taken from the paper
disp('size of Measurement Matrix');
size(MM)

% D stands for daubechies, H for haar, S for symmlet and F for Fourier.

% create a measurement matrix operator
MM = opMatrix(MM);

%using MyWavelet function to reconstruct the paper profile with spgl1
%solver with Daubechies, Symmlet and Haar as basis functions.
[recoveredH] = MyWavelet(WH,MM,bpure,MD,CD);
[recoveredF] = MyWavelet(WF,MM,bpure,MD,CD);
[recoveredS] = MyWavelet(WS,MM,bpure,MD,CD);
[recoveredD] = MyWavelet(WD,MM,bpure,MD,CD);

%figure
%surf(recoveredF);
%title('recovered data');
%xlabel('MD');
%ylabel('CD');
%saveas(gcf, 'furecovered.jpg');

%error based on norm 2 for different sample numbers

diffH = (norm(Y - recoveredH,'fro') )/norm(Y,'fro')
diffF = (norm(Y - recoveredF,'fro') )/norm(Y,'fro')
diffS = (norm(Y - recoveredS,'fro') )/norm(Y,'fro')
diffD = (norm(Y - recoveredD,'fro') )/norm(Y,'fro')
diff = [diffF, diffH, diffS, diffD]

figure
bar(diff);
legend('Fourier', 'Haar', 'Symlet', 'Daubechies');
ylabel('norm(Y-recovered)/norm(Y)');
xlabel('184 samples');
saveas(gcf,'fuerror.jpg');

J1= norm(recoveredF - Y,1)
J2= norm(recoveredF - Y, 'fro')

figure
plot(MSave,'*g');

mdrecovered = mean(recoveredF);
cdrecovered = mean(recoveredF,2);
mdorig = mean(Y);
cdorig = mean(Y,2);

figure
subplot(2,1,1);
plot(cdrecovered,'r'); hold on;
plot(cdorig);
title('CD of the recovered and original signals');
ylabel('amplitude'); xlabel('CD');
legend('Recovered CD','Original CD');
subplot(2,1,2);
plot(mdrecovered,'r'); hold on;
plot(mdorig);
title('MD of the recovered and original signals');
ylabel('amplitude'); xlabel('MD');
legend('Recovered MD','Original MD');
saveas(gcf, 'fuMDCDseparate.jpg');
A. 2 Exponential Filter

function [CDf, MDf] = expfilt3(y, wp, MD, CD)
%
% This function performs the exponential averaging of the input matrix.
% This is the new filtering function which has been debugged. ...% Do not use the MDf output as the MD, take the MD=mean(y);
%
% INPUTS:
% y matrix of measurements
% a weighting factor for the exp. filt. z(i) = a*x(i)+(1-a)*z(i-1)
% MD, CD size of MD and CD
%
% OUTPUTS:
% CD CD profile

[n, m] = size(y);
MDf = mean(y'); % calculate the MD profile
MDf = MDf(ones(1, m),:); % create 2D MD profile
y_1 = y - MDf; % remove the scan averages from the measurements
CDf(1,1) = y_1(1,1);
for i = 2:n
    CDf(i) = a * y_1(i)+(1-a)*CDf(i-1);
end

A. 3 Scanner

function [MM] = MyScanner(MD, CD)
% Parisa Towfighi
% This is a function that accepts two input variables MD and CD.
% The function can work for a paper of any size.
% It returns MM, which is a matrix of MD, MD*CD, with ones where
% the measurements would be. This MM result is meant to be
% applied to a signal column vector to gather data points.
% The scanner follows a zigzag path on the paper. Each scan
% contains CD points in it. We are assuming a speed
A. 4. Wavelet

% of one MD point per sample time.

jj=1;
while (jj < MD)
    for ii=CD:-1:1
        M(jj)= ii;
        jj=jj+1;
    end;
    for ii=2:1:CD-1
        M(jj)= ii;
        jj=jj+1;
    end;
end;

MM= zeros(MD, MD*CD);
N= zeros(CD, MD);
for kk=1:MD
    N(M(kk), kk)=1;
end
[r,c,v]= find(N);
for kk=1:MD
    MM(kk,(r(kk)-1)*MD + c(kk))=1;
end

function [recoveredY]= MyWavelet(WW,MM,data,MD,CD)
    opWW = classOp(WW); %convert wavelet to a class
    AW = opFoG(MM,WW); % Form product
    opts = spgSetParms('verbosity',0); % Turn off the SPGL1 log output
    [recoveredD ,R,G,INFO] = spgl1(AW, data, 0, 1e-3, [], opts); % spgl1 compressive sensing solver
    recoveredDd= opWW * recoveredD;
    % take the recovereds signal to original basis
    recoveredY= reshape(recoveredDd,CD,MD);
    % reshape the recovered signal to be 2 dimensional

function [recoveredY,INFO]= MyWaveletbpdn(WW,MM,bpure,MD,CD,sigma)
A. 5. Simulated Paper

```matlab
%convert wavelet to a class
opWW = classOp(WW);

% Form product
AW = opFoG(MM,WW); 

% Turn off the SPGL1 log output
opts = spgSetParms('verbosity',0);

% spgl1 compressive sensing solver
[recoveredD ,R,G,INFO] = spg_bpdn(AW,bpure,sigma,opts);

% take the recovereds signal to original basis
recoveredDd = opWW * recoveredD;

% reshape the recovered signal to be 2 dimensional
recoveredY = reshape(recoveredDd,CD,MD);
```

A. 5 Simulated Paper

A. 5.1 Simulated Signal with Random Speed Scanner

```
% Parisa Towfighi
% #: 36716009

% Introduction to the code:
% In this program, we generate a paper matrix.
% We take random samples of the signal Y.
% To do this, we have a measurement matrix called MM.
% Its number of rows equals the numberofsamples variable
% and its number of columns is equal to MD*CD. We reshape
% the original signal to be a row vector of MD*CD
% elements. The resulting data gathered is in the form
% of a column vector. We wrap the measurement matrix and
% use it with four basis functions of Daubechies, Haar,
% Symmlet and Fourier. We use spgl1 to reconstruct the
% paper profile using compressive sensing in l1.

clc
close all
clear all

MD = 64; 
CD = 64;
Y = zeros(CD,MD); %initialize paper profile without noise
```
A. 5. Simulated Paper

\[ Y_{\text{Noise}} = \text{zeros}(CD,MD); \hspace{0.5em} \% \text{initialize paper profile with noise} \]
\[ U = \text{zeros}(CD,MD); \hspace{0.5em} \% \text{used in generating paper profile} \]
\[ X = \text{zeros}(CD,MD); \hspace{0.5em} \% \text{used in generating paper profile} \]
\[ \text{noise} = \text{zeros}(CD,MD); \hspace{0.5em} \% \text{initialize noise} \]
\[ d = \text{randn}(CD,MD); \hspace{0.5em} \% \text{used in noise generation} \]

\% We are generating a profile with sinusoids in % both MD and CD directions. It will be called \( Y \) and % will be used to calculate the error.

\% used in paper profile
\[ B_t = \text{toeplitz([-0.0814 \hspace{0.5em} -0.0455 \hspace{0.5em} -0.0047 \hspace{0.5em} 0.0017 \hspace{0.5em} 0.0003 \hspace{0.5em} \text{zeros}(1,CD-5)])}; \]
\[ E = \text{toeplitz([0.8221,\hspace{0.5em} \text{zeros}(1,CD-1)]}); \hspace{0.5em} \% \text{used in noise} \]
\[ H = \text{toeplitz([0.9990,\hspace{0.5em} \text{zeros}(1,CD-1)]}); \hspace{0.5em} \% \text{used in noise} \]

\[ \text{myones} = \text{ones}(CD,1); \]
\[ x = 0: 1: (CD-1); \]
\[ t = 0: 0.25: (MD/4 - 0.25); \]
\[ \text{wcd} = 0.4/(2*\pi); \hspace{0.5em} \% \text{frequencies in CD direction} \]
\[ \text{wcdp} = 0.3/(2*\pi); \]
\[ \text{sincd} = (1.1*\sin(2*\pi* \text{wcd} \hspace{0.5em} x)); \hspace{0.5em} \% \text{frequencies in CD direction} \]
\[ \text{wmd} = 5.5/(2*\pi); \hspace{0.5em} \% \text{frequencies in MD direction} \]
\[ \text{wmdp} = 0.4/(2*\pi); \]
\[ \text{sinmd} = (\sin(2*\pi* \text{wmd} \hspace{0.5em} t)); \]
\[ \text{sinmdp} = (\sin(2*\pi* \text{wmdp} \hspace{0.5em} t)); \]
\[ \text{for} \ i=4:MD \]
\[ \text{noise}(:,i) = (H*\text{noise}(:,i-1) + d(:,i) - E*d(:,i-1))*0.8; \hspace{0.5em} \% \text{noise} \]
\[ Y(:,i) = B_t*U(:,i-3) + 0.8221*Y(:,i-1) + \text{sinmd}(1,i) * \text{myones} + \ldots \]
\[ \text{sinmdp}(1,i) * \text{myones} ; \hspace{0.5em} \% \text{No Noise} \]
\[ \text{YNoise}(:,i) = B_t*U(:,i-3) + 0.8221*\text{YNoise}(:,i-1) + \text{noise}(:,i) + \ldots \]
\[ \text{sinmd}(1,i) * \text{myones} + \text{sinmdp}(1,i) * \text{myones}; \hspace{0.5em} \% \text{with Noise} \]
\[ U(:,i) = \text{sincd} ; \hspace{0.5em} \% \text{generating CD variation} \]
\[ \text{end} \]
\[ \text{YY=YY}; \]
\[ \text{Y} = \text{YNoise}; \]
\[ \text{levels} = 5; \]

% creating operators with Daubechies, Haar, Symlet and Fourier. Both MD % CD need to be divisible by 2^ level
\[ \text{mode2} = 2; \]
A. 5. Simulated Paper

WH = opHaar2d(MD,CD,levels);  
WS = opWavelet(MD,CD,'symmlet',4,levels); % use symlet wavelets  
WD = opWavelet(MD,CD,'daubechies',8,levels); % use Daubechies wavelets  
WF = opFFT2d(MD,CD); % use fourier

figure  
surf(Y);  
title('Y');  
xlabel('MD');  
ylabel('CD');zlabel('Amplitude');  
saveas(gcf,'randomsignal.jpg');  

numberofsamples= 184;  
M=[1; 64; 66; 127; 126;  
64*2+60; 64*2+61; 64*3+4; 64*3+57;  
64*3+58; 64*4+5; 64*4+56; 64*5+6; 64*5+54;  
64*5+55; 64*6+7; 64*6+53; 64*6+54; 64*7+8;  
64*7+52; 64*8+7; 64*8+8; 64*8+51; 64*8+52;  
64*9+3; 64*9+4; 64*9+50; 64*4+; 640+47;  
64*11+5; 64*11+47; 64*11+48; 64*12+6; 64*12+46;  
64*13+8; 64*13+44; 64*13+45; 64*14+7; 64*14+8;  
64*14+43; 64*15+4; 64*15+41; 64*15+5; 64*16+3;  
64*16+40; 64*16+41; 64*17+38; 64*17+1; 64*18+3;  
64*18+4; 64*18+36; 64*19+6; 64*19+7; 64*19+38;  
64*20+8; 64*20+35; 64*21+10; 64*21+32; 64*21+33;  
64*22+11; 64*22+12; 64*22+34; 64*23+13; 64*23+36;  
64*23+37; 64*24+15; 64*24+16; 64*24+38; 64*24+37;  
64*25+17; 64*25+40; 64*25+41; 64*26+18; 64*26+19;  
64*26+40; 64*26+41; 64*27+20; 64*27+21; 64*27+38;  
64*27+39; 64*28+23; 64*28+24; 64*28+38; 64*29+39; 64*32+32;  
64*29+40; 64*29+25; 64*29+26; 64*30+27; 64*30+28;  
64*30+42; 64*31+42; 64*31+30; 64*32+44; 64*32+31; 65*32; 696;  
64*33+31; 64*33+32; 64*33+45; 64*34+46; 64*34+47; 64*34+29;  
64*35+48; 64*35+29; 64*35+28; 64*36+48; 64*36+49; 64*36+27;  
64*37+51; 64*37+25; 64*37+52; 64*38+53; 64*38+23; 64*38+24;  
64*39+54; 64*39+55; 64*39+22; 64*40+56; 64*40+20; 64*40+21;  
64*41+57; 64*41+16; 64*41+17; 64*41+58; 64*42+59; 64*42+60;  
64*42+17; 64*43+61; 64*43+14; 64*43+15; 64*44+63; 64*44+62;  
64*44+12; 64*44+13; 64*45+64; 64*45+11; 64*46+11; 64*46+64;  
64*46+63; 64*47+12; 64*47+62; 64*47+61; 64*48+10; 64*48+59;  
64*48+58; 64*49+11; 64*49+55; 64*49+56; 64*49+57; 64*49+58;  
64*50+10; 64*50+55; 64*50+56; 64*51+9; 64*51+53; 64*51+54;  
64*52+8; 64*52+59; 64*52+60; 64*53+7; 64*53+61; 64*54+6;  
64*54+62; 64*54+63; 64*55+5; 64*55+64; 64*56+5; 64*56+64;  
64*57+6; 64*57+63; 64*58+7; 64*58+62; 64*59+8; 64*59+61;
A. 5. Simulated Paper

64*60+9; 64*60+60; 64*61+10; 64*61+11; 64*61+59; 64*62+12; 64*62+58; 64*63+13; 64*63+14; 64*63+57;
AA = zeros(CD,MD);

MM= zeros(numberofsamples, MD*CD);
    for ii=1:numberofsamples
        MM(ii,M(ii))=1;
        cc=mod(M(ii),64);
        if (cc== 0)
            cc= 64;
        end;
        AA(ceil(M(ii)/64),cc)=1;
    end
AA= AA';
MSave= M;
figure;
contourf(AA);
colormap hsv;
title('Measurement matrix');
ylabel('CD');
xlabel('MD');
saveas(gcf,'RSmeasurementmatrixhsv.jpg');

original = (reshape(Y,MD*CD,1)); % now a column vector
bpure = MM* original; % measured points

disp('size of data');
size(bpure) % bpure is the actual samples taken from the paper
disp('size of Measurement Matrix');
size(MM)

% D stands for daubechies, H: haar, S: symmlet and F: Fourier.

% create a measurement matrix operator
MM = opMatrix(MM);

%using MyWavelet function to reconstruct the paper with spgl1
%solver with Daubechies, Symmlet and Haar as basis functions.
[recoveredH] = MyWavelet(WH,MM,bpure,MD,CD);
[recoveredF] = MyWavelet(WF,MM,bpure,MD,CD);
[recoveredS] = MyWavelet(WS,MM,bpure,MD,CD);
[recoveredD] = MyWavelet(WD,MM,bpure,MD,CD);
%error based on norm 2 for different sample numbers
A. 5. Simulated Paper

diffH = (norm(Y - recoveredH,'fro') )/norm(Y,'fro');
diffF = (norm(Y - recoveredF,'fro') )/norm(Y,'fro');
diffS = (norm(Y - recoveredS,'fro') )/norm(Y,'fro');
diffD = (norm(Y - recoveredD,'fro') )/norm(Y,'fro');
diff=[diffF, diffH, diffS, diffD];

figure
bar(diff);
legend('Fourier','Haar','Symlet','Daubechies');
ylabel('norm(Y - recovered)/norm(Y)');
xlabel('184 samples');
saveas(gcf,'randomerror.jpg');

[ oCD1F, oMD1F ] =expfilt3(recoveredF,0.8,MD,CD);
[ oCD1H, oMD1H ] =expfilt3(recoveredH,0.8,MD,CD);
[ oCD1S, oMD1S ] =expfilt3(recoveredS,0.8,MD,CD);
[ oCD1D, oMD1D ] =expfilt3(recoveredD,0.8,MD,CD);

mdorig=mean(real(Y));cdorig=mean(real(Y),2);
mdrecovered=mean(real(recoveredF))
cdrecovered=mean(real(recoveredF),2)
J1=norm(recoveredF)
J2=norm(recoveredF,'fro')

figure
subplot(2,1,1);
plot(cdrecovered,'g');hold on;
plot(cdorig);
title('CD of the recovered and original signals');
ylabel('amplitude'); xlabel('CD');
legend('Recovered CD','Original CD');
subplot(2,1,2);
plot(mdrecovered,'g');hold on;
plot(mdorig);
title('MD of the recovered and original signals');
ylabel('amplitude'); xlabel('MD');
legend('Recovered MD','Original MD');
saveas(gcf,'randomMDCD.jpg');

A. 5.2 Simulated Data

% Parisa Towfighi
% #: 36716009
% Introduction to the code:
% In this program, we load a data file. We take random samples
% of the signal Y via a measurement matrix called MM. Its
% number of rows is equal to numberofsamples variable and
% its number of columns is equal to MD*CD. We reshape the
% original signal to be a row vector of MD*CD elements.
% The resulting data gathered is in the form of a column vector.
% We wrap the measurement matrix and use it with four basis
% functions of Daubechies, Haar, Symmlet and Fourier. We use spgl1 to
% reconstruct the paper using compressive sensing in l1.

clc
close all
clear all

% Due to the nature of spgl1 inverse problem I have a square paper profile
% with more variation in MD vs. CD.
MD = 64;
CD = 64;
Y = zeros(CD,MD); %initialize paper profile without noise
YNoise = zeros(CD,MD); % initialize paper profile with noise
U = zeros(CD,MD); % used in generating paper profile
X = zeros(CD,MD); % used in generating paper profile
noise = zeros(CD,MD); % initialize noise
d = randn(CD,MD); % used in noise generation

% In the following section we are generating a profile with sinusoids in
% both MD and CD directions. The profile generated will be called Y and
% it will be used to calculate reconstruction error with original Fourier
% and Loewen methods.
% used in paper profile
Bt = toeplitz([-0.0814 -0.0455 -0.0047 0.0017 0.0003 zeros(1,CD-5)]);
E = toeplitz([0.8221,zeros(1,CD-1)]); % used in noise
H = toeplitz([0.9990,zeros(1,CD-1)]); % used in noise

myones = ones(CD,1);
x = 0: 1: (CD-1);
t = 0: 0.25: (MD/4 - 0.25);
wcd = 0.4/(2*pi); wcdp = 0.3/(2*pi);

%frequencies in CD direction
sincd = ( 1.1*sin(2*pi* wcd* x));sincdp= ( 0.3*sin(2*pi* wcdp* x));
A. 5. Simulated Paper

\[ w_{md} = \frac{5.5}{(2\pi)}; \quad w_{mdp} = \frac{0.4}{(2\pi)}; \quad w_{md1} = \frac{0.006}{(2\pi)}; \]
\[ w_{mdp1} = \frac{0.04}{(2\pi)}; \]
\[ \sin(m_{md}) = (\sin(2\pi w_{md} t)); \]
\[ \sin(m_{mdp}) = (\sin(2\pi w_{mdp} t)); \% frequencies in MD direction \]
\[ \sin(m_{md1}) = (\sin(2\pi w_{md1} t)); \]
\[ \sin(m_{mdp1}) = (\sin(2\pi w_{mdp1} t)); \% frequencies in MD direction \]

for \( i=4:MD \)
\[ \text{noise}(i,i) = (H \times \text{noise}(i,i-1) + d(i,i) - E \times d(i,i-1)) \times 0.8; \% \text{generating} \]
\[ Y(i,i) = Bt \times U(i,i-3) + \sin(m_{md}(i,i)) \times \text{myones} + \sin(m_{mdp}(i,i)) \times \text{myones}; \% \text{No Noise} \]
\[ Y_{\text{Noise}}(i,i) = Bt \times U(i,i-3) + 0.8221 \times Y_{\text{Noise}}(i,i-1) + \text{noise}(i,i) + \ldots \]
\[ \sin(m_{md}(i,i)) \times \text{myones} + \sin(m_{mdp}(i,i)) \times \text{myones}; \% \text{with Noise} \]
\[ U(i,i) = \text{sincd}; \% \text{generating CD variation} \]
end

YY = Y;
Y = Y_{\text{Noise}};
levels = 5;

% creating operators with Daubechies, Haar, Symlet and Fourier. MD and
% CD need to be divisible by 2^level
mode2 = 2;
WH = \text{opHaar2d}(MD,CD,levels);
WS = \text{opWavelet}(MD,CD,'symmlet',4,levels); \% use symlet wavelets
WD = \text{opWavelet}(MD,CD,'daubechies',8,levels); \% use Daubechies wavelets
WF = \text{opFFT2d}(MD,CD); \% use fourier

% bpure is the actual samples taken from the paper
for tt=1:3
% in this for loop, we gradually increase the number of samples taken
% and plot the difference in the norm of the original signal and the
% reconstruction for different sample numbers.
numberofsamples(tt) = MD + (tt-1) * 80;
% the range of the #of samples is from MD to MD*6
M = \text{ceil}(MD*CD.*rand(numberofsamples,1));
MM = \text{zeros(numberofsamples, MD*CD)};
for ii=1:numberofsamples
\[ \text{MM}(ii,M(ii)) = 1; \]
end
MSave = M;
original = (\text{reshape}(Y,MD*CD,1)); \% now a column vector
bpure = MM \times \text{original}; \% measured points

\text{disp('size of data');}
\text{size(bpure)} \% bpure is the actual samples taken from the paper
disp('size of Measurement Matrix');
size(MM)

% Use Daubechie wavelets as the basis function in which signal is sparse
% D stands for daubechies, H for haar, S for symmlet and F for Fourier.

% create a measurement matrix operator
MM = opMatrix(MM);

% using MyWavelet function to reconstruct the paper profile with spgl1
% solver with Daubechies, Symmlet and Haar as basis functions.

[recoveredH] = MyWavelet(WH,MM,bpure,MD,CD);
[recoveredF] = MyWavelet(WF,MM,bpure,MD,CD);
[recoveredS] = MyWavelet(WS,MM,bpure,MD,CD);
[recoveredD] = MyWavelet(WD,MM,bpure,MD,CD);

% error based on norm 2 for different sample numbers

diffH(tt) = (norm(Y - recoveredH,'fro') )/norm(Y,'fro');
diffF(tt) = (norm(Y - recoveredF,'fro') )/norm(Y,'fro');
diffS(tt) = (norm(Y - recoveredS,'fro') )/norm(Y,'fro');
diffD(tt) = (norm(Y - recoveredD,'fro') )/norm(Y,'fro');

end;

figure
plot(numberofsamples,diffF,'*g');
hold on;
plot(numberofsamples,diffH,'*');
hold on;
plot(numberofsamples,diffS,'+g');
hold on;
plot(numberofsamples,diffD,'or');
legend('Fourier','Haar','Symmlet','Daubechies');
ylabel('norm(Y - recovered)/norm(Y)');
xlabel('sigma count');
%xlabel('number of samples')

[ oCD1F, oMD1F ] = expfilt3(recoveredF,0.8,MD,CD);
[ oCD1H, oMD1H ] = expfilt3(recoveredH,0.8,MD,CD);
[ oCD1S, oMD1S ] = expfilt3(recoveredS,0.8,MD,CD);
[ oCD1D, oMD1D ] = expfilt3(recoveredD,0.8,MD,CD);

figure
subplot(2,1,1);
A. 5. Simulated Paper

surf(real(oCD1F));
title('Value of LP filter for CD variation, Fourier, BW');
ylabel('CD'); xlabel('MD');
subplot(2,1,2);
surf(oCD1S);
title('Value of LP filter for CD variation, Symlet, BW');
ylabel('CD'); xlabel('MD');

A. 5.3 Simulated Data with Varying Sigma

% Parisa Towfighi
% #: 36716009

% Introduction to the code:
% In this program, we generate a data file.
% We take random samples of the signal Y.
% To do this, we have a measurement matrix called MM.
% Its number of rows is equal to numberofsamples variable
% and its number of columns is equal to MD*CD. We reshape
% the original signal to be a row vector of MD*CD elements.
% The resulting data gathered is in the form of a column vector.
% We wrap the measurement matrix and use it with four basis
% functions of Daubechies, Haar, Symmlet and Fourier. We use spgl1 to
% reconstruct the paper profile using compressive sensing in l1.

clc
close all
clear all

% Due to the nature of spgl1 inverse problem I have a square paper
% with more variation in MD vs. CD.
MD = 64;
CD = 64;
Y = zeros(CD,MD);  %initialize paper profile without noise
YNoise = zeros(CD,MD);  % initialize paper profile with noise
U = zeros(CD,MD);  % used in generating paper profile
X = zeros(CD,MD);  % used in generating paper profile
noise = zeros(CD,MD);  % initialize noise
d = randn(CD,MD);  % used in noise generation

% In the following section we are generating 2 sinusoids in
% both MD and CD directions. The profile generated is called Y
% It will be used to calculate error with original Fourier
A. 5. Simulated Paper

% and Loewen methods.
% used in paper profile
Bt = toeplitz([-0.0814 -0.0455 -0.0047 0.0017 0.0003 zeros(1,CD-5)]);
E = toeplitz([0.8221,zeros(1,CD-1)]); % used in noise
H = toeplitz([0.9990,zeros(1,CD-1)]); % used in noise

myones = ones(CD,1);
x = 0: 1: (CD-1);
t = 0: 0.25: (MD/4 - 0.25);
wcd = 0.4/(2*pi); wcdp = 0.3/(2*pi);

% frequencies in CD direction
sincd = ( 1.1*sin(2*pi* wcd* x));
sinwcdp= ( 0.3*sin(2*pi* wc dp* x));
wmd = 5.5/(2*pi);wmdp= 1.4/(2*pi);wmd  =0.006/(2*pi);
wmdp1= 0.04/(2*pi);
sinmd = (sin(2*pi*wmd*t));
sinmdp=0;% (sin(2*pi*wmdp*t));% frequencies in MD direction

for i=4:MD
    noise(:,i)=(H*noise(:,i-1)+d(:,i)-E*d(:,i-1))*0.15; % generating
    Y(:,i)= Bt*U(:,i-3)+ 0.8221*Y(:,i-1)+ 3*sinmd(1,i)*myones ;% No Noise
    YNoise(:,i)=Bt*U(:,i-3)+0.8221*YNoise(:,i-1)+ noise(:,i)+ ... 3*sinmd(1,i)*myones;
    U(:,i)= sincd ; % generating CD variation
end
YY=Y;
Y=YNoise;
levels = 5;

% creating operators with Daubechies, Haar, Symlet and Fourier. Both MD
% need to be divisable by 2^ level
mode2 = 2;
WH = opHaar2d(MD,CD,levels);
WS = opWavelet(MD,CD,'symmplet',4,levels); % use symlet wavelets
WD = opWavelet(MD,CD,'daubechies',8,levels); % use Daubechies wavelets
WF = opFFT2d(MD,CD); % use fourier

figure
surf(YY);
title('noiseless signal');
xlabel('MD');
ylabel('CD');
A. 5. Simulated Paper

saveas(gcf,'Noisefreetausignal.jpg');

figure
surf(Y);
title('Y');
xlabel('MD');
ylabel('CD');
saveas(gcf,'DNoriginal.jpg')
for sigmacount=1:8

sigma(sigmacount)=0.0005*(sigmacount-1)^2;

numberofsamples= MD*2.5;
M= cell(MD*CD.*rand(numberofsamples,1));
MM= zeros(numberofsamples, MD*CD);
  for ii=1:numberofsamples
    MM(ii,M(ii))=1;
  end
MSave= M;
original = (reshape(Y,MD*CD,1)); % now a column vector
bpure = MM* original; % measured points

disp('size of data');
size(bpure) % bpure is the actual samples taken from the paper

disp('size of Measurement Matrix');
size(MM)

% D stands for daubechies, H for haar, S for symmlet and F for Fourier.

% create a measurement matrix operator
MM = opMatrix(MM);

%using MyWavelet function to reconstruct the paper profile with spgl1
%solver with Daubechies, Symmlet and Haar as basis functions.

[recoveredH, INFOH] = MyWaveletbpdn(WH,MM,bpure,MD,CD,sigmacount);
[recoveredF, INFOF] = MyWaveletbpdn(WF,MM,bpure,MD,CD,sigmacount);
if sigmacount==1
  recoveredFnodn= recoveredF;
elseif sigmacount==8
  recoveredFdn= recoveredF;
end;

[recoveredS, INFOS] = MyWaveletbpdn(WS,MM,bpure,MD,CD,sigmacount);
[recoveredD, INFOD] = MyWaveletbpdn(WD,MM,bpure,MD,CD,sigmacount);
% error based on norm 2 for different sample numbers

diffH(sigmacount) = (norm(Y - recoveredH,'fro') )/norm(Y,'fro');
diffF(sigmacount) = (norm(Y - recoveredF,'fro') )/norm(Y,'fro');
diffS(sigmacount) = (norm(Y - recoveredS,'fro') )/norm(Y,'fro');
diffD(sigmacount) = (norm(Y - recoveredD,'fro') )/norm(Y,'fro');
end;

mdrecoverednodn= mean(recoveredFnodn);
cdrecoverednodn= mean(recoveredFnodn,2);
mdrecovereddn= mean(recoveredFdn);
cdrecovereddn= mean(recoveredFdn,2);
mdoriginal= mean(Y);
cdoriginal= mean(Y,2);

figure
plot(sigma, diffF,'--g');
hold on;
plot(sigma, diffH);
hold on;
plot(sigma, diffS, '-.g');
hold on;
plot(sigma, diffD, 'r');
legend('Fourier','Haar','Symlet','Daubechies');
ylabel('norm(Y - recovered)/norm(Y)');
xlabel('sigma');
title('Error with increasing noise threshold, sigma');
saveas(gcf,'DNerror.jpg');

figure
subplot(2,1,1);
plot(real(cdrecovereddn),'--g');
title('CD recovered with and without denoising');
hold on;
plot(real(cdrecoverednodn),'b');
hold on;
plot(real(cdoriginal),'-.r');
legend('CD using denoising', 'CD without denoising','original CD');
subplot(2,1,2);
plot(real(mdrecovereddn),'--g');
hold on;
plot(real(mdrecoverednodn),'b');
hold on;
plot(real(mdoriginal),'-.r');
legend('MD using denoising', 'MD without denoising','original MD');
A. 6 Honeywell Industrial Data

A. 6.1 Basis Weight

Parisa Towfighi
#: 36716009

Introduction to the code:

In this program, we load a data file. We take random samples of the signal \( Y \) via a measurement matrix called \( MM \). Its number of rows is equal to \( \text{numberofsamples} \) variable and its number of columns is equal to \( MD \times CD \). We reshape the original signal to be a row vector of \( MD \times CD \) elements. The resulting data gathered is in the form of a column vector. We wrap the measurement matrix and use it with four basis functions of Daubechies, Haar, Symmlet and Fourier. We use spgl1 to reconstruct the paper profile using compressive sensing in \( l1 \).

clc

```
title('MD recovered with and without denoising');

% [ oCD1F, oMD1F ] = expfilt3(recoveredF, 0.8, MD, CD);
% [ oCD1H, oMD1H ] = expfilt3(recoveredH, 0.8, MD, CD);
% [ oCD1S, oMD1S ] = expfilt3(recoveredS, 0.8, MD, CD);
% [ oCD1D, oMD1D ] = expfilt3(recoveredD, 0.8, MD, CD);
figure
plot(MSave, '*g');

figure
subplot(2,1,1);
surf(real(recoveredFnodn));
title('Recovered signal with no denoising, Fourier basis');
ylabel('CD'); xlabel('MD');
subplot(2,1,2);
surf(real(recoveredFdn));
title('Same sample set as above with denoising, Fourier basis');
ylabel('CD'); xlabel('MD');
saveas(gcf, 'DNrecovered.jpg');
SNR = snr(YY, Y);
J1dn = norm(recoveredFdn - Y, 1)
J2dn = norm(recoveredFdn - Y, 'fro')

J1nodn = norm(recoveredFnodn - Y, 1)
J2nodn = norm(recoveredFnodn - Y, 'fro')
```
close all
clear all
% Due to the nature of spgl1 inverse problem I have a square paper
% with more variation in MD vs. CD.
MD = 64;
CD = 64;
Y = zeros(CD,MD); %initialize paper profile without noise
levels = 5;

% creating operators with Daubechies, Haar, Symlet and Fourier. MD
% CD need to be divisible by 2^ level
WH = opHaar2d(MD,CD,levels);
WS = opWavelet(MD,CD,'symmlet',4,levels); % use symlet wavelets
WD = opWavelet(MD,CD,'daubechies',8,levels); % use Daubechies wavelets
WF = opFFT2d(MD,CD); % use fourier

load MDCDSeparationData
k=1;
BW= bwt_last_scan(:,40:158);
for jj=1:5:350
    B(k,:)=(BW(jj,:)+BW(jj+1,:)+BW(jj+2,:)+BW(jj+3,:)+BW(jj+4,:))/5;
    k=k+1;
end
BC= B(1:CD,1:MD);
Y = BC;
Y = mat2gray(Y);
original = (reshape(Y,MD*CD,1));
figure
surf(Y);
title('Y');
xlabel('MD');
ylabel('CD');
% We get the coefficients of the original profile

for tt=1:5 % in this for loop, we gradually increase the
% number of samples taken and plot the difference
% in the norm of the original signal and the
% reconstruction for different sample numbers.

numberofsamples(tt)= MD+(tt-1)*600; % the range of the #of samples
M = ceil(MD*CD.*rand(numberofsamples(tt),1));
MM = zeros(numberofsamples(tt), MD*CD);
    for ii=1:numberofsamples(tt)
\[ \text{MM}(\text{ii}, \text{M(ii)}) = 1; \]
end

\[ \text{bpure} = \text{MM} \ast \text{original}; \quad \% \text{measured points} \]
\% gathered data from profile sparse in fourier basis
\text{disp}('\text{size of data}');
\text{size(bpure)} \% \text{bpure is the actual samples taken from the paper}
\text{disp}('\text{size of Measurement Matrix}');
\text{size(MM)}

\% D stands for daubechies, H: haar, S: symmlet and F: Fourier.

\% create a measurement matrix operator
\text{MM} = \text{opMatrix(MM)};

\%using MyWavelet function to reconstruct the paper with spgl1
\%solver with Daubechies, Symmlet and Haar as basis functions.
\text{recoveredH} = \text{MyWavelet(WH,MM,bpure,MD,CD)};
\text{recoveredF} = \text{MyWavelet(WF,MM,bpure,MD,CD)};
\text{recoveredS} = \text{MyWavelet(WS,MM,bpure,MD,CD)};
\text{recoveredD} = \text{MyWavelet(WD,MM,bpure,MD,CD)};

\%error based on norm 2 for different sample numbers
\text{diffH(tt)} = (\text{norm}(\text{Y} - \text{recoveredH},'fro')) / \text{norm}(\text{Y},'fro')
\text{diffF(tt)} = (\text{norm}(\text{Y} - \text{recoveredF},'fro')) / \text{norm}(\text{Y},'fro')
\text{diffS(tt)} = (\text{norm}(\text{Y} - \text{recoveredS},'fro')) / \text{norm}(\text{Y},'fro')
\text{diffD(tt)} = (\text{norm}(\text{Y} - \text{recoveredD},'fro')) / \text{norm}(\text{Y},'fro')
end;

\% Doing compressive sensing with the scanner
\text{MM} = \text{MyScanner(MD,CD)};
\% The 2d profile is now a column vector
\text{bpure} = \text{MM} \ast \text{original}; \quad \% \text{measured points}

\% create a measurement matrix operator
\text{MM} = \text{opMatrix(MM)};
\text{recoveredB} = \text{MyWavelet(WF,MM,bpure,MD,CD)};
\text{diffB} = (\text{norm}(\text{Y} - \text{recoveredB},'fro')) / \text{norm}(\text{Y},'fro')

\% We plot the different errors that we calculate with\% different number of samples
\text{figure}
\text{plot(MD,diffB, '+r'); hold on;}
\text{plot(numberofsamples,diffF,'*g'); hold on;}
\text{plot(numberofsamples,diffH,'*'); hold on;}
\text{plot(numberofsamples,diffS,'+g');}
plot(numberofsamples,diffD,'or');
legend('Fourier, Scanner','Fourier random',...'Haar random','Symlet','Daubechies');
ylabel('norm(Y - recovered)/norm(Y)');
xlabel('number of samples');
save csdatam diffF diffH diffS diffD

%load csdatam.mat diffF diffH diffP diff diffS diffD

mdorig= mean(real(Y));cdorig=mean(real(Y),2);
mdrecovered=mean(real(recoveredF));
ccd=mean(real(recoveredF),2);
J1= norm(recoveredF)
J2= norm(recoveredF,'fro')

figure
subplot(2,1,1);
plot(cdrecovered,'--g'); hold on
plot(cdorig);
title('CD of the recovered and original signals');
ylabel('amplitude'); xlabel('CD');
legend('Recovered CD','Original CD');
subplot(2,1,2);
plot(mdrecovered,'--g'); hold on
plot(mdorig);
title('MD of the recovered and original signals');
ylabel('amplitude'); xlabel('MD');
legend('Recovered MD','Original MD');
saveas(gcf,'Jadid.jpg');

A. 6.2 Moisture

% Parisa Towfighi
% #: 36716009
% Introduction to the code:
% In this program, we load a data file. We take random samples
% of the signal Y via a measurement matrix called MM. Its
% number of rows is equal to numberofsamples variable and
% its number of columns is equal to MD*CD. We reshape the
% original signal to be a row vector of MD*CD elements. The
% resulting data gathered is in the form of a column vector.
% We wrap the measurement matrix and use it with four basis
% functions of Daubechies, Haar, Symmlet and Fourier. We use
% spgl1 to reconstruct the paper using compressive sensing in 11.
clc
close all
clear all
% Due to the nature of spgl1 inverse problem I have a square paper
% with more variation in MD vs. CD.
MD = 64;
CD = 64;
Y = zeros(CD,MD); %initialize paper profile without noise
levels = 5;

% creating operators with Daubechies, Haar, Symlet and Fourier. MD &
% CD need to be divisable by 2^ level
WH = opHaar2d(MD,CD,levels);
WS = opWavelet(MD,CD,'symmlet',4,levels); % use symlet wavelets
WD = opWavelet(MD,CD,'daubechies',8,levels); % use Daubechies wavelets
WF = opFFT2d(MD,CD); % use fourier
load MDCDSeparationData
k=1;
MW= moi_last_scan(:,40:158);
for jj=1:5:350
   M(k,:)=(MW(jj,:)+MW(jj+1,:)+MW(jj+2,:)+MW(jj+3,:)+MW(jj+4,:))/5;
k=k+1;
end
MC= M(1:CD,1:MD);
Y = MC;
Y = mat2gray(Y);

original = (reshape(Y,MD*CD,1));

figure
surf(Y);
title('Y');
xlabel('MD');
ylabel('CD');
% We get the coefficients of the original profile

for tt=1:5
%in this for loop, we gradually increase the number of samples taken
% and plot the difference in the norm of the original signal and the
% reconstruction for different sample numbers.

numberofsamples(tt)= MD+(tt-1)*600; % the range of the #of samples
M = ceil(MD*CD.*rand(numberofsamples(tt),1));
\[ \text{MM} = \text{zeros}(\text{numberofsamples}(\text{tt}), \text{MD} \times \text{CD}); \]
\[ \text{for } \text{ii}=1: \text{numberofsamples}(\text{tt}) \]
\[ \quad \text{MM}(\text{ii},\text{M}(\text{ii}))=1; \]
\[ \text{end} \]

\[ \text{bpure} = \text{MM} \times \text{original}; \] % measured points
\[ \% \text{gathered data from profile sparse in fourier basis} \]
\[ \text{disp('size of data')}; \]
\[ \text{size(bpure)} \] % bpure is the actual samples taken from the paper
\[ \text{disp('size of Measurement Matrix')}; \]
\[ \text{size(MM)} \]
\[ \% \text{D stands for daubechies, H: haar, S: symmlet and F: Fourier}. \]

\[ \% \text{create a measurement matrix operator} \]
\[ \text{MM} = \text{opMatrix}(\text{MM}); \]

\[ \% \text{using MyWavelet function to reconstruct the paper with spgl1} \]
\[ \% \text{solver with Daubecies, Symmlet and Haar as basis functions.} \]
\[ \text{recoveredH} = \text{MyWavelet}(\text{WH},\text{MM},\text{bpure},\text{MD},\text{CD}); \] %Haar
\[ \text{recoveredF} = \text{MyWavelet}(\text{WF},\text{MM},\text{bpure},\text{MD},\text{CD}); \] %Fourier, random
\[ \text{recoveredS} = \text{MyWavelet}(\text{WS},\text{MM},\text{bpure},\text{MD},\text{CD}); \] %Symlet
\[ \text{recoveredD} = \text{MyWavelet}(\text{WD},\text{MM},\text{bpure},\text{MD},\text{CD}); \] %Daubechies

\[ \% \text{error based on norm 2 for different sample numbers} \]
\[ \text{diffH}(\text{tt}) = (\text{norm}(\text{Y} - \text{recoveredH},'fro') )/\text{norm}(\text{Y},'fro') \]
\[ \text{diffF}(\text{tt}) = (\text{norm}(\text{Y} - \text{recoveredF},'fro') )/\text{norm}(\text{Y},'fro') \]
\[ \text{diffS}(\text{tt}) = (\text{norm}(\text{Y} - \text{recoveredS},'fro') )/\text{norm}(\text{Y},'fro') \]
\[ \text{diffD}(\text{tt}) = (\text{norm}(\text{Y} - \text{recoveredD},'fro') )/\text{norm}(\text{Y},'fro') \]
\[ \% \text{Doing compressive sensing with the scanner} \]
\[ \text{MM} = \text{MyScanner}(\text{MD},\text{CD}); \]
\[ \% \text{The 2d profile is now a column vector} \]
\[ \text{bpure} = \text{MM} \times \text{original}; \] % measured points

\[ \% \text{create a measurement matrix operator} \]
\[ \text{MM} = \text{opMatrix}(\text{MM}); \]
\[ \text{recoveredB} = \text{MyWavelet}(\text{WF},\text{MM},\text{bpure},\text{MD},\text{CD}); \]
\[ \text{diffB} = (\text{norm}(\text{Y} - \text{recoveredB},'fro') )/\text{norm}(\text{Y},'fro') \]

\[ \% \text{We plot the different errors that we calculate with} \]
\[ \% \text{different number of samples} \]
\[ \text{figure} \]
\[ \text{plot(MD,diffB,'+r'); hold on;} \]
\[ \text{plot(numberofsamples,diffF,'*g'); hold on;} \]
\[ \text{plot(numberofsamples,diffH,'*'); hold on;} \]
A. 6. Honeywell Industrial Data

plot(numberofsamples,diffS,'+g');
plot(numberofsamples,diffD,'or');
legend('Scanner','Fourier random','Haar random','Symlet','Daubechies');
ylabel('norm(Y - recovered)/norm(Y)');
xlabel('number of samples');
save csdatam diffF diffH diffS diffD

mdorig= mean(real(Y));cdorig=mean(real(Y),2);
mdrecovered= mean(real(recoveredF))
cdrecovered=mean(real(recoveredF),2)
J1= norm(recoveredF)
J2= norm(recoveredF,'fro')

figure
subplot(2,1,1);
plot(cdorig,'r'); hold on;
plot(mdorig);
title('CD of the recovered and original signals');
ylabel('amplitude'); xlabel('CD');
legend('Recovered CD','Original CD');
subplot(2,1,2);
plot(cdorig,'r'); hold on;
plot(mdorig);
title('MD of the recovered and original signals');
ylabel('amplitude'); xlabel('MD');
legend('Recovered MD','Original MD');
saveas(gcf,'Jadid2.jpg');

A. 6.3 Moisture Data in a Loop

% Parisa Towfighi
% #: 36716009
% Introduction to the code:
% In this program, we generate a data file. We take random
% samples of the signal Y, via a measurement matrix called MM.
% Its number of rows is equal to numberofsamples variable
% and its number of columns is equal to MD*CD. We reshape
% the original signal to be a row vector of MD*CD
% elements. The resulting data gathered is in the form of
% a column vector. We wrap the measurement matrix and use
% it with four basis functions of Daubechies, Haar, Symmlet
% and Fourier. We use spgl1 to reconstruct the paper profile
% using compressive sensing in l1.
clc
close all
clear all
% Due to the nature of spgl1 inverse problem I have a square paper
% with more variation in MD vs. CD.
MD = 64;
CD = 64;
Y = zeros(CD,MD); %initialize paper profile without noise
levels = 5;

% creating operators with Daubechies, Haar, Symlet and Fourier. MD and
% CD need to be divisable by 2^ level
WH = opHaar2d(MD,CD,levels);
WS = opWavelet(MD,CD,'symmlet',4,levels); % use symlet wavelets
WD = opWavelet(MD,CD,'daubechies',8,levels); % use Daubechies wavelets
WF = opFFT2d(MD,CD); % use fourier

load MDCDSeparationData
k=1;
MW= moi_last_scan(:,40:158);
for jj=1:5:350
    M(k,:)= (MW(jj,:)+MW(jj+1,:)+MW(jj+2,:)+MW(jj+3,:)+MW(jj+4,:))/5;
    k=k+1;
end
MC= M(1:CD,1:MD);
Y = MC;
Y = mat2gray(Y);
for timet= 1:5
    for ii=1:(MD-1)
        Y(:,ii)= Y(:,ii+1);
    end
    Y(:,MD)= Y(:,timet);
end
original = (reshape(Y,MD*CD,1));

for tt=1:1
    numberofsamples(tt)= MD*3;
    M = ceil(MD*CD.*rand(numberofsamples(tt),1));
    MM = zeros(numberofsamples(tt), MD*CD);
A. 6. Honeywell Industrial Data

for ii=1:numberofsamples(tt)
    MM(ii,M(ii))=1;
end

bpure = MM* original; % measured points
% gathered data from profile sparse in fourier basis
disp('size of data');
size(bpure) % bpure is the actual samples taken from the paper
disp('size of Measurement Matrix');
size(MM)
% Use Daubechies wavelets as the basis function in which signal is sparse
% D stands for daubechies, H for haar, S for symmlet and F for Fourier.

% create a measurement matrix operator
MM = opMatrix(MM);

%using MyWavelet function to reconstruct the paper profile with spgl1
%solver with Daubechies, Symmlet and Haar as basis functions.
recoveredH = MyWavelet(WH,MM,bpure,MD,CD); %Haar
recoveredF = MyWavelet(WF,MM,bpure,MD,CD); %Fourier, random
recoveredS = MyWavelet(WS,MM,bpure,MD,CD); %Symlet
recoveredD = MyWavelet(WD,MM,bpure,MD,CD); %Daubechies

%error based on norm 2 for different sample numbers
diffH(tt) = (norm(Y - recoveredH,'fro') )/norm(Y,'fro')
diffF(tt) = (norm(Y - recoveredF,'fro') )/norm(Y,'fro')
diffS(tt) = (norm(Y - recoveredS,'fro') )/norm(Y,'fro')
diffD(tt) = (norm(Y - recoveredD,'fro') )/norm(Y,'fro')
end;
% Doing compressive sensing with the scanner
MM = MyScanner(MD,CD);
% The 2d profile is now a column vector
bpure = MM* original; % measured points

% create a measurement matrix operator
MM = opMatrix(MM);
recoveredB = MyWavelet(WF,MM,bpure,MD,CD);
diffB = (norm(Y- recoveredB ,'fro') )/norm(Y,'fro')
end
% We plot the different errors that we calculate with
% different number of samples
figure
plot(MD,diffB, '+r'); hold on;
plot(numberofsamples,diffF,'*g'); hold on;
plot(numberofsamples,diffH,'*'); hold on;
A. 6. Honeywell Industrial Data

plot(numberofsamples,diffS,'+g');
plot(numberofsamples,diffD,'or');
legend('Scanner','Fourier random','Haar random', ... 
'Symlet','Daubechies');
ylabel('norm(Y - recovered)/norm(Y)');
xlabel('number of samples');
save csdatam diffF diffH diffS diffD

%load csdatam.mat diffF diffH diffP diff diffS diffD
[ oCD1F, oMD1F ] =expfilt3(recoveredF,0.8,MD,CD);
[ oCD1H, oMD1H ] =expfilt3(recoveredH,0.8,MD,CD);
[ oCD1S, oMD1S ] =expfilt3(recoveredS,0.8,MD,CD);
[ oCD1D, oMD1D ] =expfilt3(recoveredD,0.8,MD,CD);

figure
subplot(4,1,1);
surf(real(oCD1F));
title('Value of LP filter for CD variation, Fourier, MOI');
ylabel('CD'); xlabel('MD');
subplot(4,1,2);
surf(oCD1H);
title('Value of LP filter for CD variation, Haar, MOI');
ylabel('CD'); xlabel('MD');
subplot(4,1,3);
surf(oCD1S);
title('Value of LP filter for CD variation, Symlet, MOI');
ylabel('CD'); xlabel('MD');
subplot(4,1,4);
surf(oCD1D);
title('Value of LP filter for CD variation, Daubechies, MOI');
ylabel('CD'); xlabel('MD');

figure
subplot(4,1,1);
surf(real(oMD1F));view([90 90]);
title('Value of LP filter for MD variation, Fourier, MOI');
ylabel('CD'); xlabel('MD');
subplot(4,1,2);
surf(oMD1H);view([90 90]);
title('Value of LP filter for MD variation, Haar, MOI');
ylabel('CD'); xlabel('MD');
subplot(4,1,3);
surf(oMD1S);view([90 90]);
title('Value of LP filter for MD variation, Symlet, MOI');
ylabel('CD'); xlabel('MD');
A. 6. Honeywell Industrial Data

```matlab
subplot(4,1,4);
surf(oMD1D);view([90 90]);
title('Value of LP filter for MD variation, Daubechies, MOI');
ylabel('CD'); xlabel('MD');
```
Appendix B - Least Squares Method

B. 1 Least Squares Method

A simplified least squares problem is solved in this section based on the method outlined in [5] to gain more insight about the estimation that we are about to undertake.

We have setup a very simple paper profile. The paper is 4 units wide in cross direction and 16 units long in machine direction. It has a sparse representation. It is defined in the following two lines of code.

\begin{verbatim}
paper= zeros(16,4);
paper(8:12, 2:3)=1;
\end{verbatim}

A zig zag function crosses the paper and records its value as shown in the image below.

Figure B. 1: Paper with ones in the middle and x where samples are taken.
B. 1. Least Squares Method

The data collected is \( b = [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]' \). As you can see, we have 16 data points while there are a total of 64 points in the paper. Then we tried to use the method of least squares to solve the system.

\[
\begin{pmatrix}
A_1(x_0) & A_2(x_0) & \cdots & A_{64}(x_0) \\
A_1(x_1) & A_2(x_1) & \cdots & A_{64}(x_1) \\
\vdots & \vdots & \ddots & \vdots \\
A_1(x_{15}) & A_2(x_{15}) & \cdots & A_{64}(x_{15})
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_{64}
\end{pmatrix}
=
\begin{pmatrix}
b_1 \\
b_2 \\
\vdots \\
b_{16}
\end{pmatrix}
\] (1)

\( A \) is the measurement matrix; \( x \) is the paper and \( b \) is the collected data. This is the formula for the least squares method.

\[
A^T A x = A^T b
\] (2)

This method is applicable to overdetermined systems, i.e. systems of equations in which there are more equations than unknowns. However, here there is an underdetermined system, i.e. more unknowns than equations.

To deal with this underdetermined system and to make the system less sensitive to sampling noise, we use linear regularization. The cross direction and machine direction are assigned Haar wavelet basis. The Haar wavelet is scaled down and shifted in both the cross direction and the machine direction.

```
% Haar basis for cross direction
cbase=[1 1 1 1; 1 1 -1 -1; 1 -1 0 0; 0 0 1 -1]';
% Haar basis for machine direction
mbase= [ones(1,16);
ones(1,8) -ones(1,8);
ones(1,4) -ones(1,4) zeros(1,8);
zeros(1,8) ones(1,4) -ones(1,4);
ones(1,2) -ones(1,2) zeros(1,12);
zeros(1,4) ones(1,2) -ones(1,2) zeros(1,8);
zeros(1,8) ones(1,2) -ones(1,2) zeros(1,4);
zeros(1,12) ones(1,2) -ones(1,2);
1 -1 zeros(1,14);
zeros(1,2) 1 -1 zeros(1,12);
zeros(1,4) 1 -1 zeros(1,10);
zeros(1,6) 1 -1 zeros(1,8);
zeros(1,8) 1 -1 zeros(1,6);
zeros(1,10) 1 -1 zeros(1,4);
zeros(1,12) 1 -1 zeros(1,2);
```
zeros(1,14) 1 -1]’;

We take the kronecker tensor product of the cross direction and machine
direction matrices.

\[
fbase = \text{kron}(cbase, mbase);
\]

\[
H = \text{inv}(fbase);
\]

H takes the measurement matrix to the wavelet basis.

\[
Hx = y \quad (3)
\]

\[
Ax = b \quad (4)
\]

\[
AH^\top THx = b \quad (5)
\]

\[
HA^\top TAH^\top Ty = HA^\top Tb \quad (6)
\]

\[
HA^\top TAH^\top Ty + kIy = HA^\top Tb \quad (7)
\]

Here k is a constant that we can choose.

\[
HA^\top TAx + kIHx = HA^\top Tb \quad (8)
\]

\[
A^\top TAx + kH^\top THx = A^\top Tb \quad (9)
\]

We solve equation 9 for x and that gives us an estimate for the paper
profile.

The reconstructed profile using the least squares method with a Haar
wavelet is shown next from two different views.

This was an example of using a Haar wavelet to solve a simple least
squares problem.
B. 1. Least Squares Method

Figure B. 2: Paper Profile

Figure B. 3: reconstructedPaper
B. 1. Least Squares Method

Figure B. 4: reconstructedPaper3d