Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Lars Gjardar Musæus

Fractal Analysis and Its Application on Time-Series Data - An Innovative Method for Condition Monitoring of Hole Cleaning Operations

Master's thesis in Cybernetics and Robotics
Supervisor: Professor Adil Rasheed
June 2022

NTNU
Norwegian University of
Science and Technology

Lars Gjardar Musæus

Fractal Analysis and Its Application on Time-Series Data - An Innovative Method for Condition Monitoring of Hole Cleaning Operations

Master's thesis in Cybernetics and Robotics
Supervisor: Professor Adil Rasheed
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Fractal Analysis and Its Application on Time-Series Data - An Innovative Method for Condition Monitoring of Hole Cleaning Operations

Lars Gjardar Musæus

Department of Engineering Cybernetics — NTNU 2022

Abstract

Within hole cleaning in the oil drilling industry, has the recent development of sensors made it possible to monitor the state of the cleaning operation while it is running. This entails a vast array of data and measured features that can be analyzed and used to potentially reduce the economic and environmental costs and increase the safety of the drilling operation. Fractal analysis is a promising yet relatively uncharted method for time-series data analysis. To our knowledge, there are not many research papers covering its applications, and the ones that exist do not describe in detail how it is implemented. This research project will propose a complete workflow on how to implement fractal analysis as a tool and apply the resulting method to synthetic and to hole cleaning data from a real drilling operation. The project has resulted in a well-working fractal analysis tool, which has shown promising results both for time-series data analysis in general and a suiting method for analyzing hole cleaning data specifically.

Sammendrag

Innenfor hullrensing i oljeboring bransjen, har ny og moderne utvikling av sensorer gjort det mulig å monitorere tilstanden til en boring- og renseoperasjon mens den er i drift. Dette medfører en stor mengde data som kan analyseres og potensielt brukes til å effektivisere operasjonen både økonomisk, miljø-, og sikkerhetsmessig. Fraktal analyse er en lovende, dog relativt lite utforsket metode innenfor tidsserie data analyse. Til vår forståelse er det ikke mange publikasjoner som dekker bruken av metoden, og de som finnes beskriver i lite detalj hvordan metoden kan implementeres. Dette forskningsprosjektet vil foreslå og argumentere for en implementasjon av et fraktal analyse verktøy, og anvende det resulterende produktet på både syntetisk data og hullrense data fra en ekte oljeboring operasjon. Prosjektet resulterte i et velfungerende fraktal analyse verktøy, som har vist å være nyttig både som et verktøy for tidsserie analyse, og for analyse av hullrensning data spesifikt.

Preface

This research project serves as a master's thesis, and concludes a five-year integrated Master of Science study program with the title "Cybernetics and Robotics" at the Norwegian University of Science and Technology, NTNU. The thesis takes the reader through the process of implementing fractal analysis as a data analysis tool and experiments with its application on real field data from a hole cleaning operation.

I want to thank SINTEF for providing hole cleaning data and Roar Nybø, Knut Steinar Bjørkevoll, and Mandar Tabib for sharing their expertise and knowledge in the oil drilling domain.

Lastly, I want to thank my supervisor, Professor Adil Rasheed, for his guidance and heavy involvement in this master's thesis, which has been paramount for both the results and motivation.

Trondheim, June 2022
Lars Gjardar Musæus

Contents

List of Figures

List of Tables

Nomenclature

PCA Principal Component Analysis

PRM Piecewise Regression Model

ECD Equivalent Circulating Density

SSE Sum of Square Error

D_r Fractal Dimension

σ Standard Deviation

σ^2 Variance

1. Introduction

There has been a fast-growing selection of models, tools, and techniques for analyzing data in the last decade. At the same time, many small and large businesses sit on considerable amounts of data but cannot find a suiting method for capturing faults and anomalies in their data. A domain-independent challenge is that known problems occur in a system, but they cannot detect them by analyzing the signal amplitudes. The concept of fractal analysis is that one could potentially be able to capture domain-related issues that generally are not detectable by analyzing the signal amplitudes alone.

1.1. Background and Motivation

The concept of fractals and fractal analysis is not new; its origin dates back to the late 60s when Benoit Mandelbrot wanted to measure the length of Britain's coastline and discovered that fractal dimensions could be used to determine the roughness of the coastlines (Kappraff, 1986). The idea of having a deterministic measure of roughness could later be applied to other surfaces as well as time-series signals. Now there are plenty of different use cases for fractal analysis. According to Landgraf and Hansmann (2019), fractal analysis is an innovative approach for evaluating the condition of railway tracks. With the use of fractal analysis on measurement data, one can see indications of deterioration of ballast and signs of weak subsoil, which previously has been a great challenge. In the realms of image texture processing, fractal analysis has proven to be a new and highly efficient method for feature extraction, and challenging the best options of today (Costa et al., 2012). Another image processing-related application is in the confectionery industry, more specifically in the production of chocolate. Here, fractal analysis has been used to find a relation in how fat in chocolate blooms to the surface during storage as this is a big issue because a bar of chocolate that does not appear clean and shiny does not sell (Quevedo et al., 2005). Typical for most of the use cases is that analyzing the roughness of a surface or a signal could uncover properties that other methods would not capture, and exploring its application to new data and domains is highly interesting.

The specialization project conducted in the fall of 2021 was regarding condition monitoring of railway tracks and exploration of different methods for predicting maintenance in the railway domain, focusing on machine learning models (Musæus, 2021). Fractal analysis came to light during the phase of finding relevant literature concerning methods used for data analysis in the named domain. However, as it was not a method based on machine learning, it was not prioritized in the specialization project. After reviewing available literature regarding fractal analysis and realizing that it is not a well-known method including that not many papers cover its applications or implementation, it seemed like a suitable subject for a master's thesis. With access to hole cleaning data from the oil drilling industry, we had the prerequisites to experiment with the method's potential in the oil drilling domain.

The drilling process is costly both economically and environmentally, and efficient hole cleaning plays a significant factor in the equation. The established practice uses sophisticated physics-based calculations to model the operation prior to its start. The increasing development of sensors and data-transmission technology enable sensors along the drill string that measures the hole cleaning state with higher precision and reliability than the physics-based models. This opens up a new realm of possible methods to be introduced in the operation. Accordingly, this project proposes to develop a complete workflow for the use of fractal analysis for an efficient and improved monitoring of the hole cleaning process during drilling operations.

1.2. Research Objectives and Research Questions

To confine the work to be done, a set of research objectives has been defined, as well as research questions contributing to reach the objectives.

1.2.1. Research Objectives

Primary Objective: Evaluate the potential of fractal analysis for condition monitoring using time-series data.
Secondary Objectives:

- Develop a full workflow for the application of fractal analysis for time-series data.
- Demonstration of fractal analysis for optimizing hole cleaning.
- Educate the reader in the implementation and use of fractal analysis.

1.2.2. Research Questions

- How can a full fractal analysis workflow be implemented?
- How can fractal analysis be utilized for analysing a hole cleaning process?
- How can the reader learn and make use of fractal analysis themselves?

1.3. Structure of the Thesis

The thesis is divided into five chapters, including the introduction. Chapter 2 presents the background and theory needed to understand the methods utilized in the thesis. Chapter 3 introduces the data used, the preparation of data needed, and the setup required to replicate the work done in the thesis. In Chapter 4, fractal analysis is applied to the data presented in Chapter 3 through various experiments, and the results are presented and discussed. The final chapter, Chapter 5, concludes the findings by answering the research questions raised in the introduction. Lastly is a section on personal reflection through the process of this master's thesis and proposed future work if the project is to be continued.

2. Theory

2.1. Fractal Analysis

2.1.1. Background - The Coastline Paradox

Fractal analysis is a rather modern concept within data analysis; the first modern paper on it is from 2002 (Hyslip, 2002), where the technique was used to analyze railway tracks. However, the underlying methodology was developed by Benoit B. Mandelbrot back in 1967. Back then, Mandelbrot wanted to estimate the length of the coastline of Great Britain because the best estimate at the time was found using Euclidean geometry, but it was not accurate enough. Mandelbrot used a methodology that later would be exploited in fractal analysis called the divider method (Andrle, 1992). By using a fixed-size ruler(divider) and drawing straight lines between the edges of the ruler around the perimeter of the map of Great Britain, one would quickly discover that decreasing the ruler size leads to a more precise replication of the map, as well as a longer total perimeter. See Figure 2.1 for illustration.

Mandelbrot ended up with the following relation between the number of segments, the segment lengths, and the total length of the curve

$$L(\lambda) = N * \lambda^{1-D_r} \tag{2.1}$$

where:

L(λ): total length of polygonal line as a function of λ

λ: length of one section

N: number of subdivisions

D_r: fractal dimension

Mandelbrot quickly realized that measuring the coastline is not as simple as it first appeared and that the length of the coastline depends on the length of the ruler used to measure it. Thus as the measuring ruler becomes infinitely small, the length of the coastline increases to infinity. This phenomenon is known as the Coastline Paradox (Mandelbrot et al., 1983).

2.1.2. Divider Method and Fractal Dimension, D_r

Using the same approach Mandelbrot used for calculating the coastline, but applying it to a time-series signal, one would be able to sample the signal with precision based on the ruler's length, which is from here on called divider width. As Mandelbrot discovered that the coastline was impossible to determine, he found a deterministic way of describing the roughness of a line, the so-called *fractal dimension*, D_r (Hyslip, 2002). With further exploitation of Equation 2.1, if the base 10 logarithm is applied to the equation you will get

$$logL(\lambda) = (1 - D_r)log\lambda + logN \tag{2.2}$$

Figure 2.1.: Illustration of Mandelbrot's divider method for measuring the coastline of Britain. By decreasing the divider lengths(orange lines), a more precise reconstruction of the map is achieved. Illustration adapted from Landgraf and Hansmann (2019).

By inspection, it is visible that the resulting expression is on the same form that expresses a line,

$$y = mx + b \tag{2.3}$$

Thus a linear relation with the slope of the line appears. If the length of one section, λ, is plotted against the total length of the polygonal line as a function of λ, $L(\lambda)$, with both axes in a logarithmic scale, a linear relation appears. This linear relation can be written as

$$m = 1 - D_r \tag{2.4}$$

and as a result; the expression for the fractal dimension is:

$$D_r = 1 - m \tag{2.5}$$

With the correlation that a smaller divider length, λ, corresponds to a more precise sample of the signal, a lower λ will also be able to capture smaller changes in the signal. With this in mind and the fact that the fractal dimension determines the rate of change of the correlation between measured signal length and the divider width, we can conclude that the fractal dimension is a measure of the roughness of the signal. As an example, if we have two arbitrary lines, A and B, where A is smoother than the latter, then the difference in the measured line length against different divider lengths would not vary much in the smoother signal. Thus the fractal dimension, D_r, will be lower in signal A compared to signal B. A plot of the divider length versus the total line length with both axes in log scale is called a Richardson plot (Hyslip, 2002).

2.1.3. Modified Divider Method

The original divider length method, as described in Section 2.1.2 uses the absolute length of the segment lines when referring to the divider length, λ. The modified divider method is based on the same equation as the original, except that instead of having equal divider lengths, the whole data is segmented into segments of equal widths. This divider width substitutes the original λ. Figure 2.2 and 2.3 illustrates the difference in how the two methods assign λ and $L(\lambda)$.

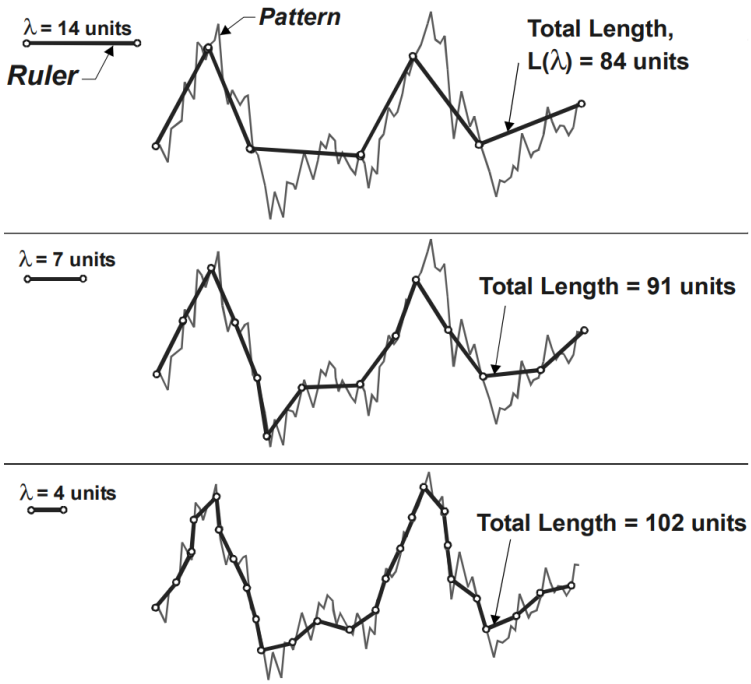


Figure 2.2.: Illustration of original divider method, where λ is a constant ruler length, and the total length, $L(\lambda)$ is the sum of the ruler lengths. Figure adapted from (Hyslip, 2002).

2.1.4. Fractal Elements and Properties

Depending on what kind of signal is being analyzed, the resulting Richardson plot can reveal a plot with either a singular slope or multiple slopes. These slopes are called fractal elements, and are illustrated in Figure 2.4 with a single fractal element and with multiple fractal elements in Figure 2.5. In case of multiple slopes will each slope reveal a fractal dimension, D_r, which is calculated using Equation 2.5. The horizontal axis on the Richardson plot corresponds to the divider width of a given divider method run; thus, the different fractal elements appear in different ranges of divider widths in the analysis. The fractal elements on the right side, usually called D_1, responds to changes over longer periods, like structural changes, while the middle slope, D_2, is more affected by higher frequency textural changes. The divider width ranges for D_1 and D_2 are dependent on the data used and its domain. More on this in Section 2.1.6.

2.1.5. Spline Interpolation

Interpolation in the context of data analysis is a realm of methods for filling in missing data in a data set. It is also used for curve fitting to eliminate outliers, which has been its primary purpose in this thesis.

Spline is a piece-wise interpolation method aiming to make smooth curves through the data points and is less prone to over-fitting compared to classic polynomial interpolation. To fit a line through a set of n data points with polynomial interpolation, a polynomial function of $n-1$ turning points is needed. For instance, if we have 100 points, it takes 99 turns to fit a line through the data with polynomial interpolation. A problem that ap-

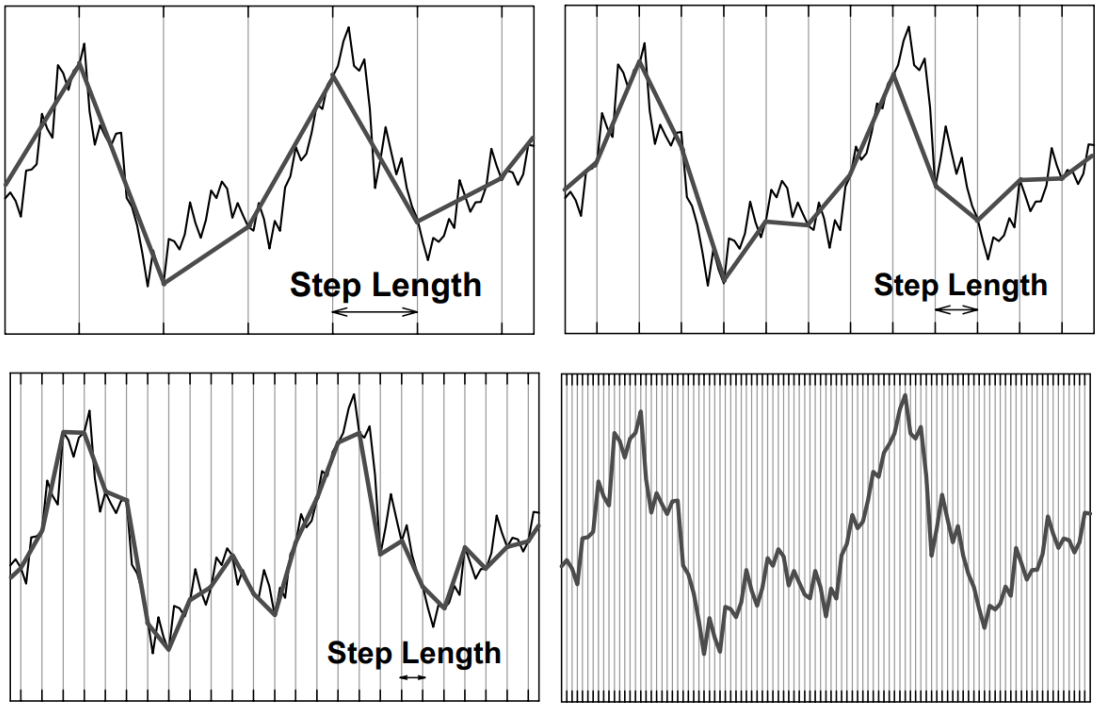


Figure 2.3.: Illustration of modified divider method. Here λ is a step length dividing the signal into segments of equal width. The total length, $L(\lambda)$, is the sum of the length of each ruler in each segment which, in contrast to the original divider method, varies from segment to segment. Figure adapted from (Hyslip, 2002).

pears when using polynomial interpolation to fit through a large number of data points is a phenomenon called polynomial wiggle or Runge's phenomenon(Epperson, 1987). It fits the data very well in the middle; however, there are often huge fluctuations in the edges that do not fit very well with the data. What Splines do, instead of creating one nth degree polynomial to fit through the data, it pieces together local polynomials between one point to the next. Then to make the resulting fitted line smooth, it makes sure that both the first derivative, which is the slope in a point, as well as the second derivative, which is the change in slope, matches at every point. The result is a smooth fit, as shown in Figure 2.7.

Using Figure 2.6 as aid, we have a set of $n+1$ data points (x_0, y_0) (x_1, y_1) ... (x_n, y_n). To calculate the spline cubic between the kth and the $k+1$th interval we have:

$$\begin{aligned} S_k(x) = S_{k,0} + S_{k,1}(x - x_k), x \in [x_k, x_{k+1}] \\ + S_{k,2}(x - x_k) \\ + S_{k,3}(x - x_k) \end{aligned} \tag{2.6}$$

This shows that each cubic/spline piece has four unknown variables , $S_{k,0}$ to $S_{k,3}$, which has to be determined. The constraints are as follows:

- $S_k(x_k) = y_k$ Each spline has to go through the data point.
- $S_k(x_{k+1}) = S_{k+1}(x_{k+1})$ To be continuous, the boundaries has to match.

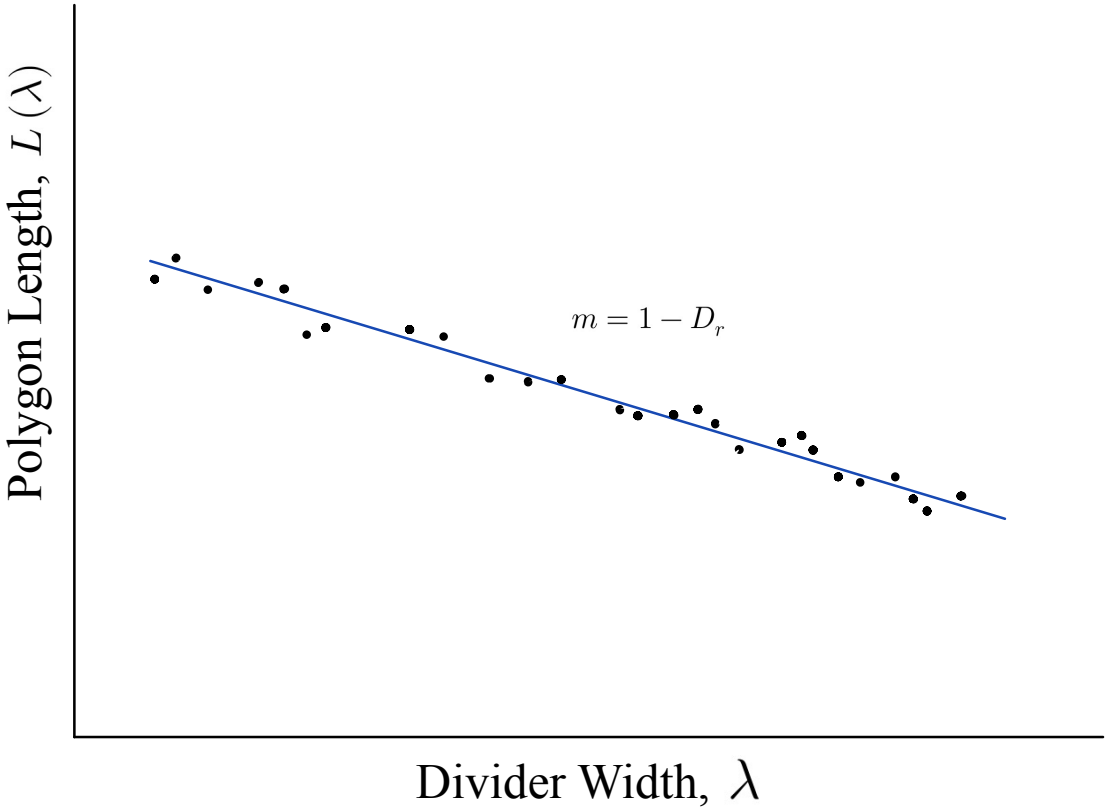


Figure 2.4.: This is a typical Richardson plot when doing fractal analysis with the divider method. It consists of the black data points with logarithmically scaled axes where the horizontal axis is the divider width, and the vertical axis is the resulting polygon length. The blue line illustrates the slope known as a fractal element, which is used to derive the fractal dimension.

- $S'_k(x_{k+1}) = S'_{k+1}(x_{k+1})$ To make the line smooth, both first- and second derivatives has to match.
- $S''_k(x_{k+1}) = S''_{k+1}(x_{k+1})$

Thus with $n+1$ data points, we have n-intervals with 4 unknowns each, which gives us a total of $4n$ unknowns. The total number of constraints are:

- $S_k(x_k) = y_k$, has to go through all data points: $n+1$.
- $S_k(x_{k+1}) = S_{k+1}(x_{k+1})$, only yields for the interior points: $n-1$
- $S'_k(x_{k+1}) = S'_{k+1}(x_{k+1})$, only yields for the interior points: $n-1$
- $S''_k(x_{k+1}) = S''_{k+1}(x_{k+1})$, only yields for the interior points: $n-1$

This gives us a total of $4n-2$ constraints, thus resulting in an under-determined system. It is typical to enforce two more conditions, for instance, set the second derivative at the edges to zero, which gives us the two extra constraints needed to solve for the unknown variables (McKinley and Levine, 1998).

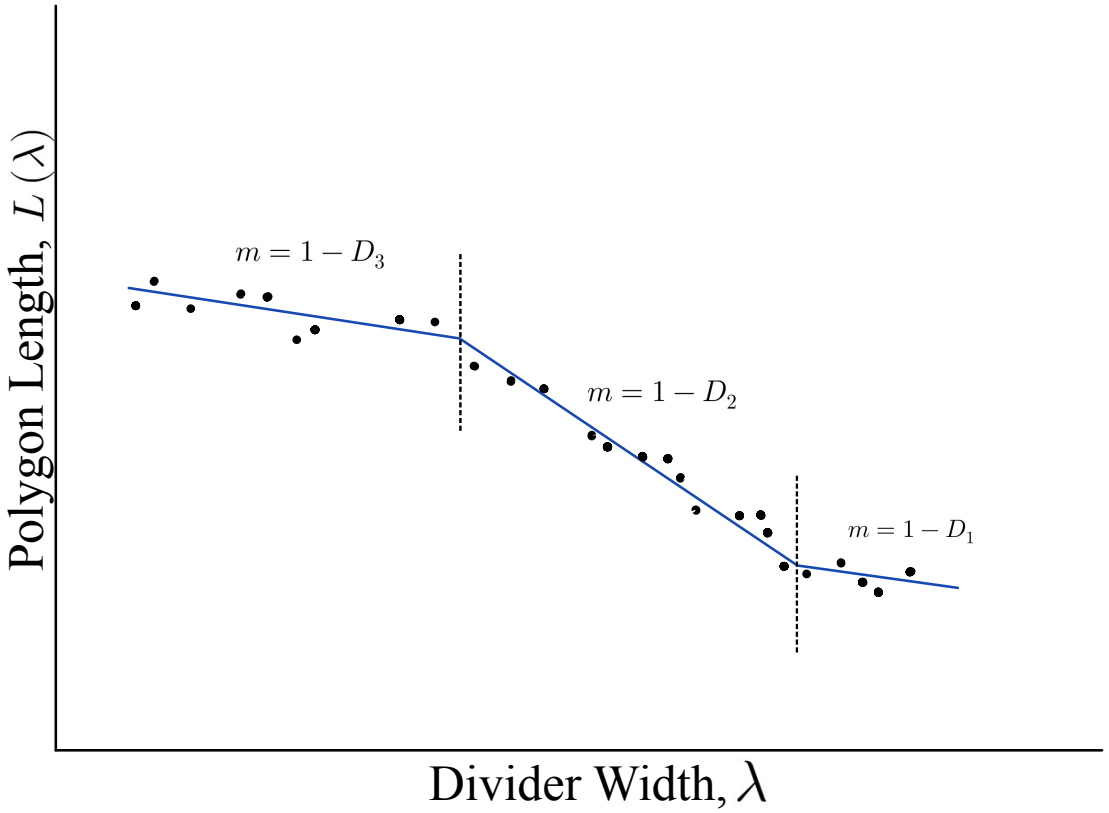


Figure 2.5.: This is a typical Richardson plot when doing fractal analysis with the divider method. It consists of the black data points with logarithmically scaled axes where the horizontal axis is the divider width, and the vertical axis is the resulting polygon length. The blue lines show that multiple slopes can often be derived from the scatterplot. The slopes signify different fractal elements. The stippled lines are breakpoints/knots.

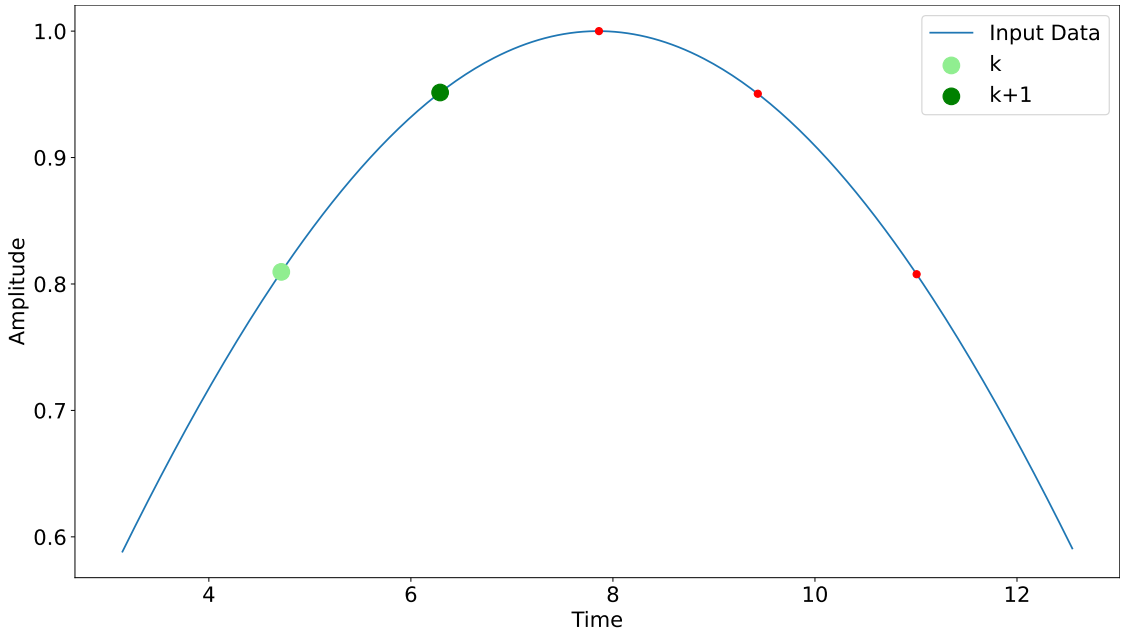


Figure 2.6.: Illustration of how knot placement would look for a arbitrary signal using spline.

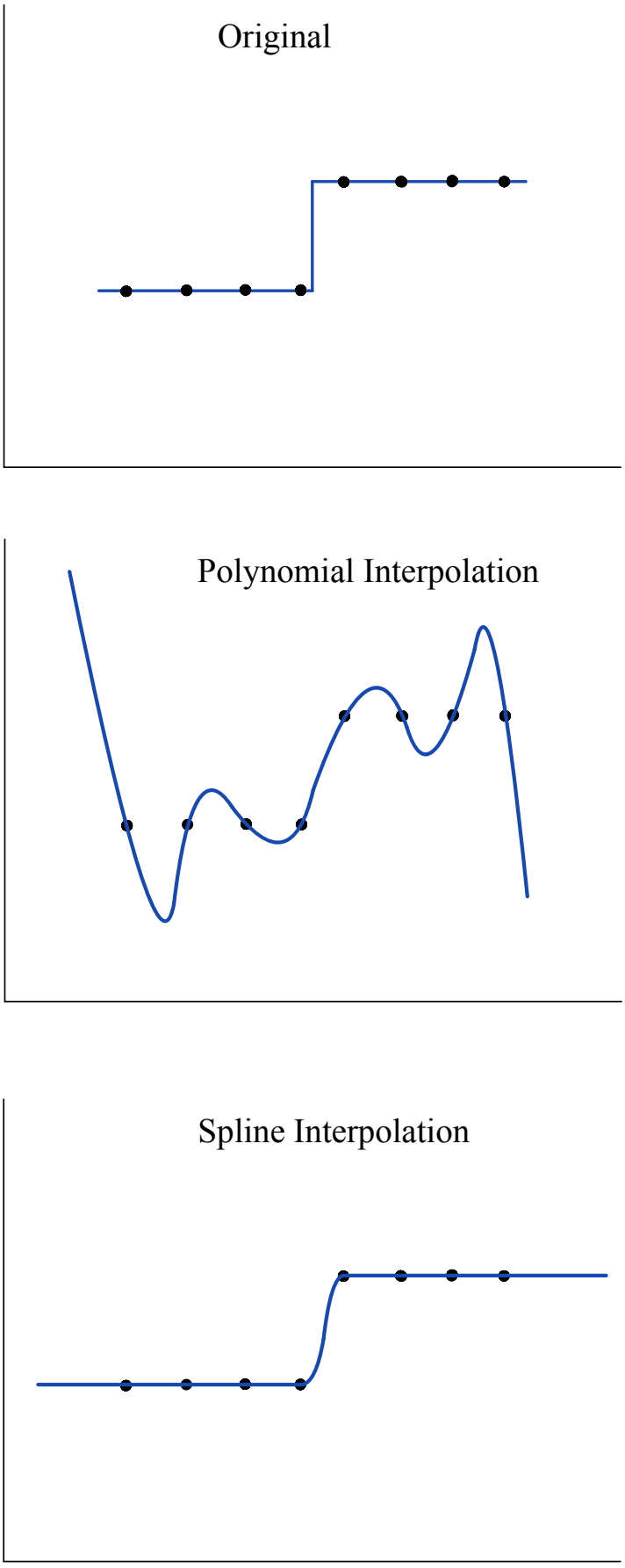


Figure 2.7.: Illustration of a signal and how best lines would be fitted using polynomial interpolation and spline, respectively.

2.1.6. Piecewise Regression Model

The piecewise regression model is a nonlinear regression model that is efficient when the trend of a scatter plot quickly changes direction, which is typically in the resulting Richardson plot when doing fractal analysis like in Figure 2.5. The idea is to fit two or more linear regression lines through the data points instead of one quadratic or cubic line, which fits our goal of extracting the slopes/fractal dimensions from the Richardson plot. The location of change is commonly known as the breakpoint or the knot, and the model supports multiple knots. The breakpoints or knots can be estimated visually, but they can also be estimated iteratively by minimizing the resulting sum of square error(SSE). When using fractal analysis, the resulting location of the knot(s) will most likely have a practical sense in the domain in which it is applied. For instance, they use three knots when doing fractal analysis in the railway domain. The range of divider widths that separate the knots tells the analyst if a problem is a sub-structural problem or a problem concerning the ballast bed.

Let us say we have a set of data points that seems suitable for the piecewise regression model with one knot, i.e., it has one change of trend throughout the data, like Figure 2.8a. Then the construction of the piecewise regression starts by creating a dummy variable for each side of the knot, which tells us if a given data point is on the left side of the knot, or the right side of the knot:

$$x_k = \begin{cases} 0, \text{ if } x_1 & \leq x^{(k)} \\ 1, \text{ if } x_1 & > x^{(k)} \end{cases}$$

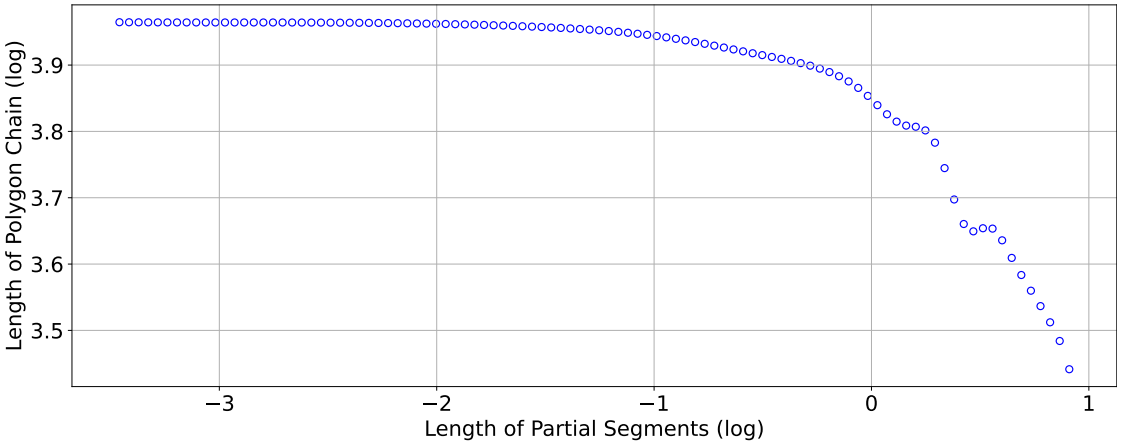
where

- x_1 = value of the independent variable, i.e. divider width in the Richardson plot.
- $x^{(k)}$ = the value at the chosen breakpoint/knot position (note that k is notation, not a exponent).
- x_k = the knot dummy variable.

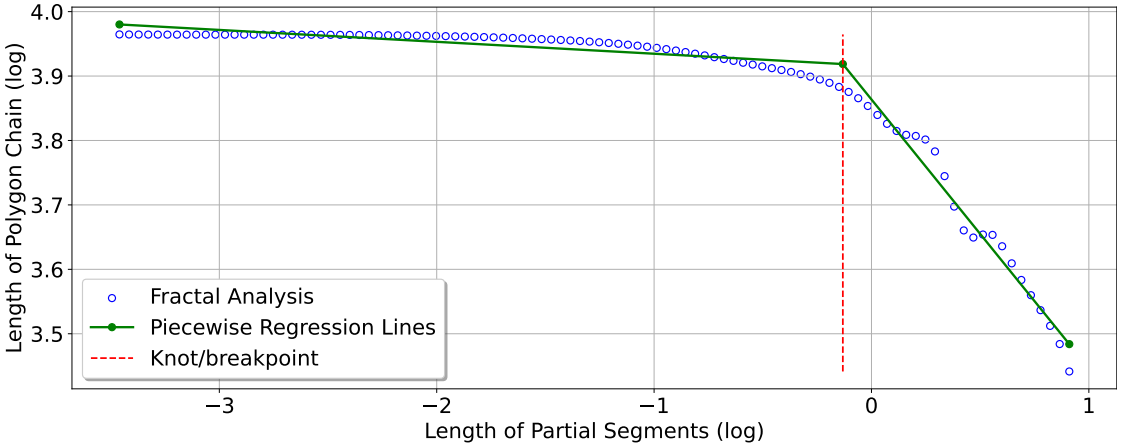
This signifies that the knot dummy variable, x_k, is zero when the value of the independent variable, x_1, is less than or equal to the value of the knot. Otherwise, the knot dummy variable's value is one if the independent variable's value is greater than the value of the knot. The piecewise regression equation is as follows:

$$\hat{y} = b_0 + b_1 x_1 + b_2 (x_1 - x^{(k)}) x_k \tag{2.7}$$

Where b_0 is the intercept coefficient, $b_1 x_1$ is the first linear term consisting of a coefficient multiplied by the value of the independent variable. Lastly is the coefficient b_2 multiplied by the knot-term for the right side of the piecewise linear regression. The variables x_1, $x^{(k)}$, and x_k are the same as listed above. The idea behind the construction of the equation is that for every x_1 location up to the knot value, $x^{(k)}$, the value of the dummy knot variable, x_k, is zero, and thus the last linear term goes away. For every x_1 after the knot location, the dummy knot variable is 1, and thus the last linear terms take effect, creating the second regression line. Figure 2.8b shows a plot of how piecewise regression would look like with one knot.



(a) Figure shows a Richardson plot with divider width on the horizontal axis, and the resulting polygon length on the vertical axis.



(b) Figure shows how piecewise regression can be used to extract the different slope values which later on can be converted to fractal dimensions using Equation 2.5. Here the knot/breakpoint is estimated to be around -0.13. The horizontal axis shows the divider width used in the analysis with the resulting total polygon length on the vertical axis.

Figure 2.8.: Two figures (a) and (b) illustrating a Richardson plot and how piecewise regression draws its linear line through the data.

2.1.7. Principal Component Analysis

Principal Component Analysis (PCA) is a well-known and widely used multivariate data analysis method, where its most common application is the reduction of dimensionality (features) while losing the least amount of information (variance) possible (Abdi and Williams, 2010).

When a data set has many possible correlated quantitative variables/features with the indicative existence of redundant information, will PCA allow us to reduce them to a smaller number of transformed variables, called principal components, that explain much of the variability in the data. Each dimension or principal component generated by PCA will be a linear combination of the original variables, and they will also be independent/uncorrelated from/with each other. The principal components generated are often used in supervised learning methods, and will in this thesis be used in fractal analysis.

PCA provides a hierarchical coordinate system based on data to represent the statistical variation in the data set. I.e., it is a coordinate system based on data directions that capture the maximal variance. To find the principal components, let us say we have a data matrix

$$X = \begin{bmatrix} \cdots & x_1 & \cdots \\ \cdots & x_2 & \cdots \\ & \vdots & \\ \cdots & x_n & \cdots \end{bmatrix}$$

where each row vector x_i are measurements from a single experiment, for instance, each feature in an oil drilling data set. We are going to assume that the data matrix X has some statistical distribution, i.e., it is not deterministic, and that there is some statistical variability. The goal is to uncover the dominant combinations of features that describe as much of the data as possible, and can be done in six steps:

1. Compute the mean row:
$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2. Calculate the average matrix (matrix of mean rows):
$$\bar{X} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} \quad \bar{\boldsymbol{x}} \quad \end{bmatrix}$$

3. Subtract mean from data matrix, giving the mean centered data:
$$B = X - \bar{X}$$

4. Compute the covariance matrix of rows of B:
$$C = B^T B$$

5. Calculate the eigenvalues and eigenvectors of C, which results in:
$$CV = VD$$
where V is the eigenvectors and D is the eigenvalues.

6. Finally we get the principal components, T:
$$T = BV$$
where the vectors V of the eigenvectors are called loadings.

Essentially by decomposing the data matrix X into the principal components vector T which holds the directions of maximum variance, and the loadings, V, are how much of each principal components it is within each experiment x_n (Brunton and Kutz, 2022)(Jolliffe, 2016). The eigenvalues, λ, in D give an indication of the amount of variance in the data set X that is being captured by the principal components. For instance, if we want to calculate how much of the variance is being captured by the first r principal components, we exploit the fact that each eigenvalue $\lambda = \sigma^2$, i.e., it is the variance of the principal component in the given data set. Thus the calculation of the cumulative captured variance for the first r principal components is done by:

$$\frac{\sum_{k=1}^{r} \lambda_k}{\sum_{k=1}^{n} \lambda_k}$$

which tells us what fraction of variance is captured by the first r eigenvalues divided by the total captured by all the eigenvalues, which is all the variance in the data. PCA is a euclidean-based method, and therefore it is important to normalize the data before applying the method. In this project, data is normalized with the following equation:

$$Z = \frac{x - \mu}{\sigma}$$

where x is an input column in our data, μ is the mean of the column, and σ is the standard deviation (Kumar et al., 2014).

3. Data, Methods and Setup

This chapter is divided into two sections; the first section introduces the reader to what data is used in this thesis and how it was generated, pre-processed, and prepared. The second section covers how the fractal analysis tool was implemented and what methods and tools were utilized.

3.1. Data

Both synthetic data and real field data have been used in this thesis. They serve two purposes; firstly, the synthetic data acts as a reference/benchmark data set when developing the fractal analysis tool and seeing if it works as intended. The field data is from an actual hole drilling operation and is used to explore if this analysis method can be helpful in the domain.

3.1.1. Synthetic Data

When implementing a tool for fractal analysis, it is helpful to have complete control of the behavior of the signal. Furthermore, when testing the method's capabilities, it is both quick and unambiguous to add anomalies. In theory, fractal analysis could be useful for analyzing signals composed of multiple wavelengths; therefore, three different signals were used during the implementation of the divider method. The three signals $A(t)$, $B(t)$, and, $C(t)$ consists of the following functions:

$$A(t) = sin(\omega_A * t) \tag{3.1a}$$

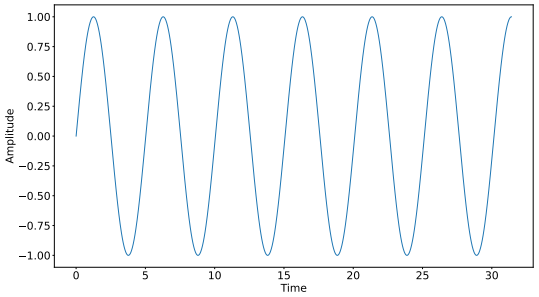
$$B(t) = sin(\omega_B * t) \tag{3.1b}$$

$$C(t) = A(t) + B(t) \tag{3.1c}$$

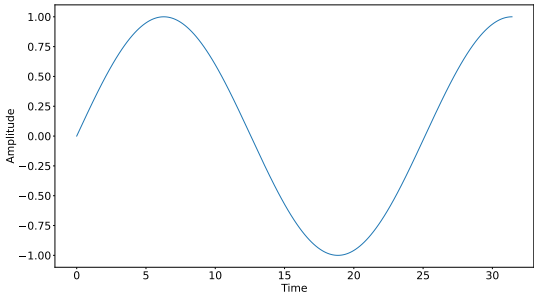
Signal $A(t)$ and $B(t)$ are independent sine waves, while signal $C(t)$ is the superposition of the first two. The wavelength for signal n is described with ω_n, where $\omega_A >> \omega_B$ to capture the effects of very different wavelengths. The superimposed signal $C(t)$ is used to see if the fractal analysis tool can handle signals composed of different wavelengths, which effectively is a more realistic signal. Figure 3.1 shows the three signals.

Adding Anomalies

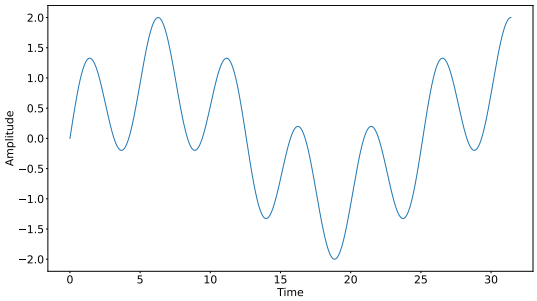
Adding anomalies to the synthetic data is, as intended, a straightforward job. When running experiments on the synthetic data, we cut out a segment in the original signal and replace it with a different signal composition, as shown in Figure 3.2. The result is an anomalous signal in which we are in perfect control over the anomaly, making experimenting with anomalies easier.



(a) $A(t) = sin(\omega_A * t)$, High frequency

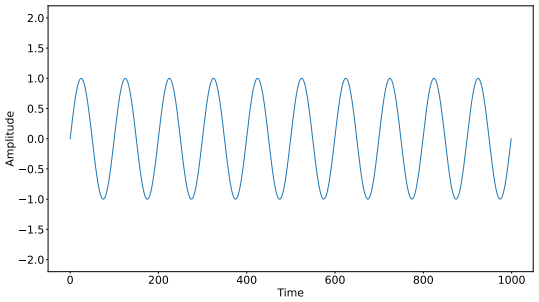


(b) $B(t) = sin(\omega_B * t)$, Low Frequency

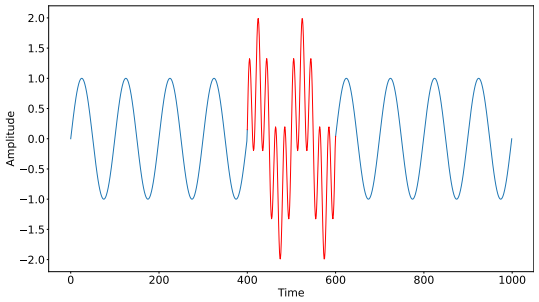


(c) $C(t) = A(t) + B(t)$, Superposition of $A(t)$ and $B(t)$

Figure 3.1.: Three signals, A, B, and C, where A and B are sine waves of different frequency, while signal C is the superposition of A and B.



(a) Original sine signal free of anomalies used for position based analysis.



(b) This plot shows the same signal as figure 3.2a, but with the middle section replaced with a different signal composition acting as an anomaly.

Figure 3.2.: Here are two plots showing how an anomaly is added to the synthetic signal.

3.1.2. Field Data

Efficient use of data is beneficial; in the motivation section, it is ascribed just how important efficient use of sensor data is in the hole drilling sector. As fractal analysis is not a typical method in the analysis stack in the oil drilling domain, did we get access to sensor data from an actual hole cleaning operation.

Hole Cleaning

Hole cleaning is the process of drilling through the soil and extracting the cuttings, creating a hole where an oil pipe can go. A rotating drill string with a bit at the end drills through the soil and leaves cuttings. To remove the cuttings, a drilling fluid is

pumped through the drill string, which, combined with the drill string's rotation, creates circulation in the hole, forcing the cuttings to move up and out. Modifying the flow rate and the rotation speed to correct values facilitates good circulation, and thus cuttings move out of the hole. Figure 3.3 illustrates what a hole cleaning operation looks like, where the blue part of the figure is the drill string with the drill bit at the end. A set of sensors are located at three different positions, where among other measurements, the depth, pressure, and equivalent circulating density (ECD) are measured.

The Dataframe

Hole cleaning is a complex process, and if not done correctly, it is expensive, time-consuming, and potentially dangerous. Thus having a good overview of the operation through sensorial monitoring and excellent models and simulations is essential.

The data provided is measurement data taken from sensors during a hole cleaning operation in the period 14.04.2020 to 29.04.2020 and consist of 85 features. The features vary from what the provider refers to as different degrees of how useful they are, and some are near-duplicates of each other. Table 3.1 shows all the different features, their unit, and a description, where the features written in bold are the most useful ones, according to the provider.

Table 3.1.: Table lists all the features in the hole cleaning data set. The most used features are written in bold, and the colored cell correspond to the sensor position illustrated in Figure 3.3.

Log/Feature	**Unit**	**Description**
TIME	yyyy-MM-dd"T"HH:mm:ss.fffzzz	Time (index)
ASMATN1-T	gn	Mean tangential acceleration
ASMDEP1-T	m	Depth ASM sensor 1
ASMDEP2-T	m	Depth ASM sensor 2
ASMDEP3-T	m	Depth ASM sensor 3
ASMDOFF-T	m	Depth offset ASM sensors
ASMECD1-T	g/cm3	ECD ASM sensor 1
ASMECD2-T	g/cm3	ECD ASM sensor 2
ASMECD3-T	g/cm3	ECD ASM sensor 3
ASMMALM1-T	gn	Mean lateral local acceleration
ASMPAM1-T	bar	Annular pressure ASM sensor 1
ASMPAM2-T	bar	Annular pressure ASM sensor 2
ASMPAM3-T	bar	Annular pressure ASM sensor 3
ASMPIM1-T	bar	Internal pressure ASM sensor 1
ASMRGM1-T	rpm	Mean rotation (gyro) ASM sensor 1
ASMTVD1-T	m	True vertical depth ASM sensor 1

ASMTVD2-T	m	True vertical depth ASM sensor 2
ASMTVD3-T	m	True vertical depth ASM sensor 3
ASMAAM1-T	dsec2	Mean angular acceleration
ASMAAZM1-T	gn	Mean z acceleration
BDTV	m	Bit TVD
BPOS	m	Block position
BSZ	in	Bit size
CKP1	bar	Choke pressure
CSTK	stks	Cumulative stroke counts
CTDA	dega	Azimuth
CTDI	dega	Inclination
DEP	m	Bit depth
DHT001 DEPTH	m	Depth of DHT001 sensor
DHT001 DEPTH OFFSET	m	Depth offset of DHT001 sensor
DHT001 ECD	g/cm3	ECD at downhole tool
DHT001 EMW	g/cm3	EMW at downhole tool
DHT001 ESD	g/cm3	ESD at downhole tool
DHT001 NOV TVD	m	TVD of DHT001 sensor
DHT001 PRESS ANN MEAN	bar	Annular pressure at downhole tool
DHT001 PRESS INT MEAN	bar	Internal pressure at downhole tool
DHT001 ROT GYRO MEAN	rpm	Rotation of downhole tool
DHT001 TEMP PCB MEAN__ASM_TIME	degC	Temperature ASM sensor
DHT001 TEMP PCB MEAN__DOWNHOLE_TIME	degC	Temperature at downhole tool
DMIAVG	kg/m3	Mud density in
DMOAVG	kg/m3	Mud density out (CVE1 Cor Dens)
EWTEMP-T	degC	Downhole/annular temperature
FLIAVG	m3/min	Flow rate in
FLOAVG	m3/min	Flow rate out
GDSS-T	unitless	Stick-slip indicator
GDTVSS-T	unitless	Stick-slip indicator
GHCAVG	unit	Hydrocarbon gas to surface
GHCMAX	unit	Hydrocarbon gas to surface
HDEP	m	Hole depth
HDTV	m	Hole TVD
HKLDAV	kg	Hook Load
IBARDENS	g/cm3	Mud density (BaRT)
IBARDENST	degC	Mud temperature (BaRT)

IBARRHEOT	degC	Mud temperature (BaRT)
IBARVIS100	unitless	Viscosity 100RPM
IBARVIS200	unitless	Viscosity 200RPM
IBARVIS3	unitless	Viscosity 3RPM
IBARVIS300	unitless	Viscosity 300RPM
IBARVIS6	unitless	Viscosity 6RPM
IBARVIS600	unitless	Viscosity 600RPM
ISLS	unitless	In slips status
LAGDEP	m	Lag depth
LAGTIM	h	Bottoms up time
OBS	unitless	On bottom status
PIP-T	bar	Internal pressure
PITC	m3	Pit volume change
PITT	m3	Pit volume total
PWEA-T	kg/m3	EMW (PWD)
PWPA-T	MPa	Annular Pressure (PWD)
PWTA-T	degC	Downhole/annular temperature (PWD)
RISFLO	L/min	Riser flow rate
ROPA	m/h	ROP averaged
ROPI	m/h	ROP instantaneous
RPMBAVG	RPM	RPM total avg
RPMMAVG	RPM	RPM motor avg
RPMSAVG	RPM	RPM surface avg
RPOS	m	Riser position
RSPD	m/min	Running speed
RT_DEPTH_RETURNS_T	m	Depth of returns
SPPAVG	MPa	Standard Pipe Pressure (SPP)
STATECODE	unitless	State
TDA	unitless	Torque/Drag activity
TMIAVG	degC	Temperature in
TMOAVG	degC	Temperature out
TQABAV	kN.m	Torque
TRPT	m3	Trip tank volume
WOBAVG	kN	Weight on bit

Feature Selection

Two experiments will be conducted with the hole cleaning data, one with hand-selected features and one with the use of principal component analysis. This subsection covers the hand-selected features.

The list of features is lengthy (Table 3.1), and as the goal of this thesis is to indicate the possible use of fractal analysis, only a selection of the features is utilized. As mentioned in the section about hole cleaning, Section 3.1.2, a flow of liquid, which is essentially mud, is pressured through the drill string to create circulation in the hole and force cuttings up and out of the hole. The equivalent circulating density, **ECD**, is

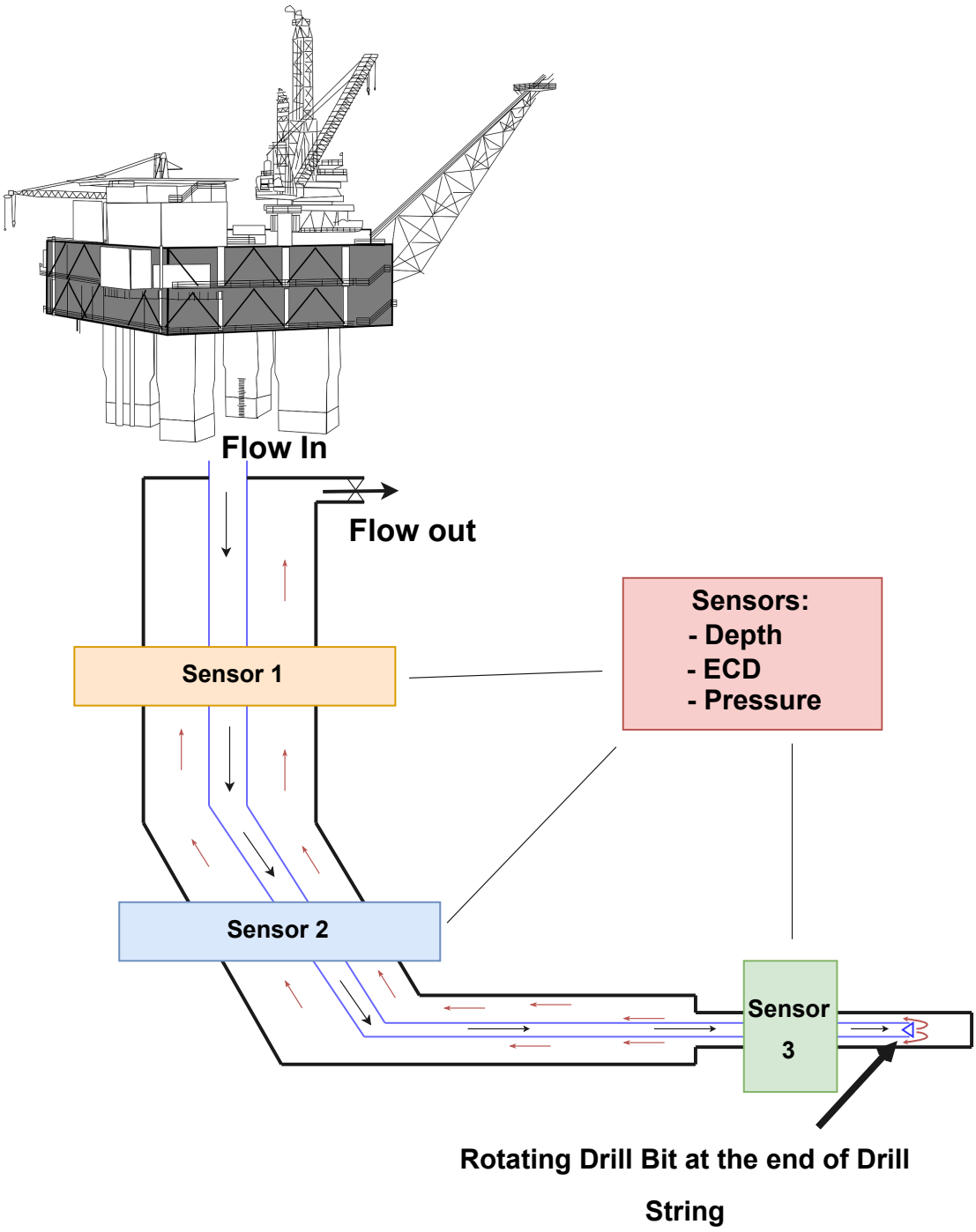


Figure 3.3.: Figure illustrates a drilling operation. The blue pipe illustrates the drill string where drilling fluid/mud is pumped down and out of the rotating drill bit and circulates back up dragging along cuttings.

the dynamic density of the circulating liquid. When the flow of liquid is emitted out the end of the drill string and moves in the annular space along the drill string, drag is created by friction against the hole wall. The pressure loss from this friction is converted to density and summed with the initial fluid density from the input liquid, and the resulting density is the ECD (Raabe and Jortner, 2022). Figure 3.4 illustrates the end

of the drill string, where the input flow creates the circulation that forces cuttings away.

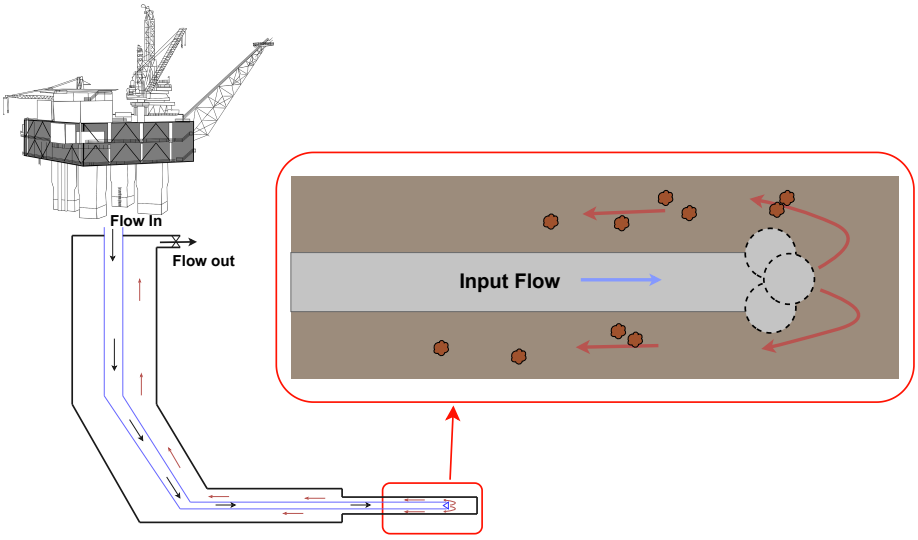


Figure 3.4.: Figure illustrates the end of the drill string where the input flow of liquid creates a circulation that extracts the cuttings(small brown pieces) up and out of the hole.

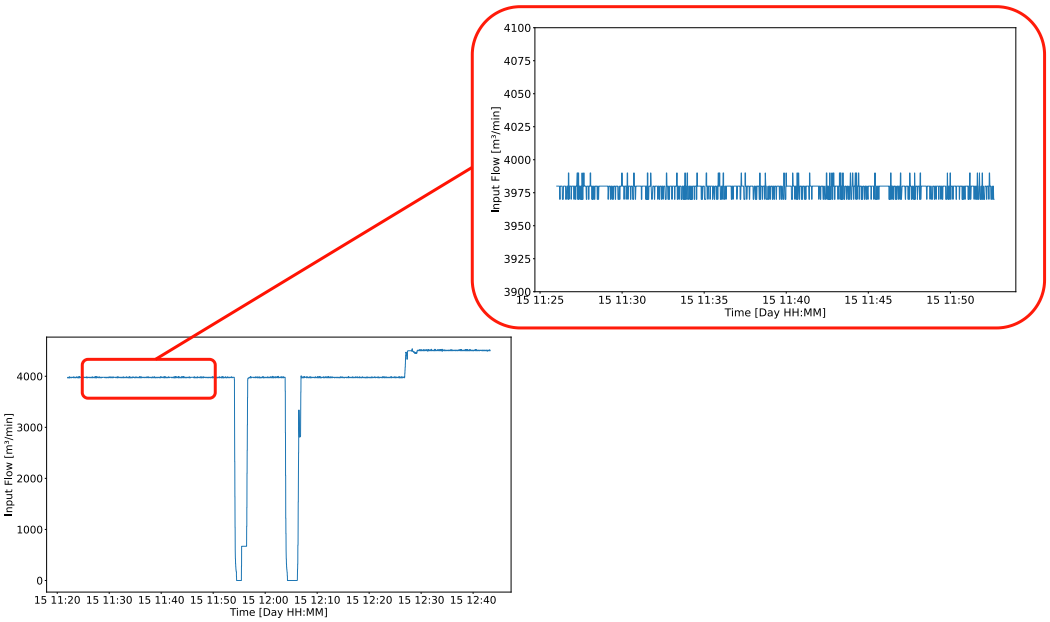
The second selected feature is the **FLIAVG**, which is the flow rate of the liquid pumped into the drill string. These two features are selected because we can create a scenario that occurs from time to time and would be highly valuable if detected. Since the ECD value is correlated to the input flow, FLIAVG, and during a regular operation with constant input flow, the ECD should be relatively stable. An issue that occurs occasionally is when the ECD suddenly increases even though the input flow is constant. This is vital to detect because a high ECD could indicate that cuttings are starting to clog the annular space between the drill string and the hole walls (Zhang et al., 2017). To investigate if fractal analysis can be used to detect an event like this, we extract a section of the ECD data where we know that the input flow is constant, as shown in Figure 3.5.

Adding Anomalies

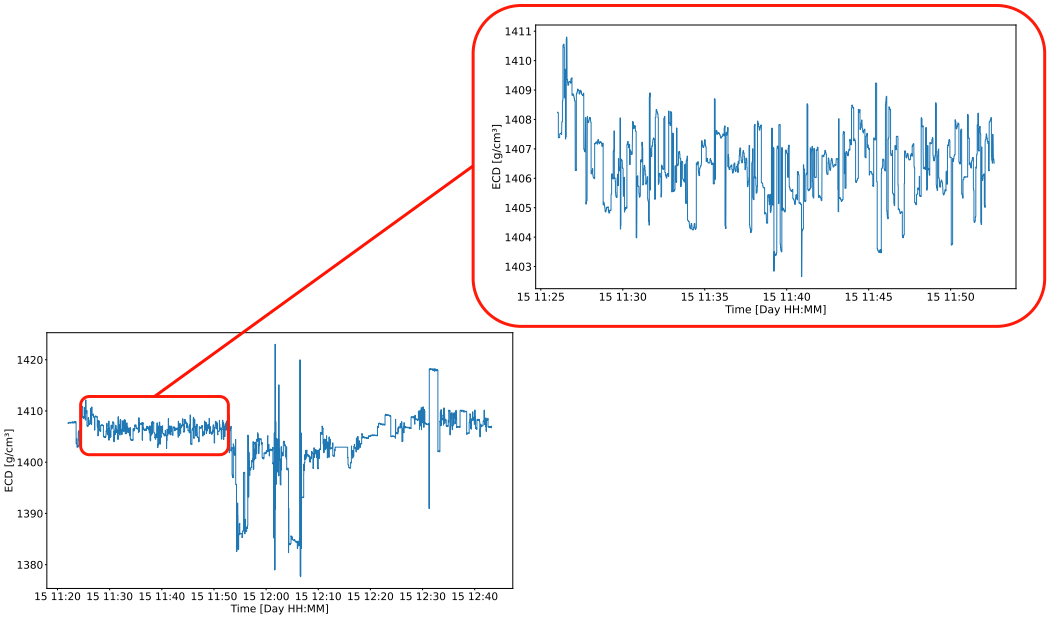
Even though an unexpected increase in the ECD measurements happens from time to time, are we not in possession of data where we know for a fact that this is happening. Consequently, we have to modify the segment by increasing the ECD manually. This has its obvious downside that it is not 100% real. However, it is close to reality, and it gives us complete control over where the anomaly is, and we can therefore confirm if the anomaly is detected correctly or not. By using the same section as seen in Figure 3.5, an anomaly can be added by elevating the values at the given section. To test the robustness of the analysis method is the anomaly added to three different positions. The anomaly profile and placements are shown in Figure 3.6.

PCA

In the theory section about PCA(Principal Component Analysis), Section 2.1.7, it is explained that in a data set, it is not always that every feature is necessary because information could be redundant, and thus only a handful of features are needed to get

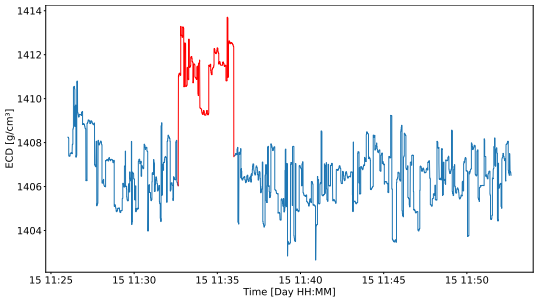


(a) This figure illustrate the region with constant input flow that was used for analysis.

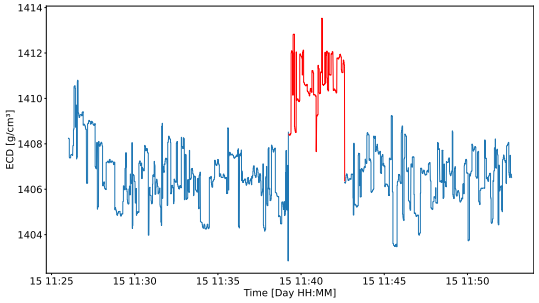


(b) This figure illustrate the measured equivalent circulating density(ECD) in the region with constant input flow that was used for analysis.

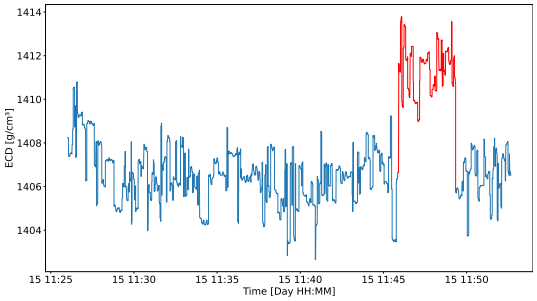
Figure 3.5.: Two figures showing how the selection of ECD segment was justified.



(a) How the measured ECD would look if a sudden increment would occur in the beginning of the signal.



(b) How the measured ECD would look if a sudden increment would occur in the middle of the signal.



(c) How the measured ECD would look if a sudden increment would occur towards the end of the signal.

Figure 3.6.: These three figures shows the measured equivalent circulating density(ECD) where an anomaly is introduced in the region marked in red.

most of the information out from the data set. In a multivariate system like the hole cleaning data process, it is not always known what causes an anomaly. Therefore, if anomalies are captured by a principal component using fractal analysis one could potentially only need to run fractal analysis on a handful of the principal components to detect anomalies in the data set instead of all the features, which is a lot more time consuming.

The hole cleaning data set provided for this thesis consist of 85 columns(features). Running a principal component analysis on the data set shows that 72% of the information (variance) is explained by the five principal components, and around 46% by the first principal component alone. Table 3.2 and Figure 3.7 show the cumulative variance explained by the principal components, while Figure 3.8 shows plots of the resulting principal component both with and without an anomaly added to the ECD measurements.

Number of Components	Explained Variance (%)
1	45.5
2	59
3	64
4	68
5	72
...	
20	94

Table 3.2.: Table shows the number of principle components needed for a given percentage of explained variance in the hole cleaning data set.

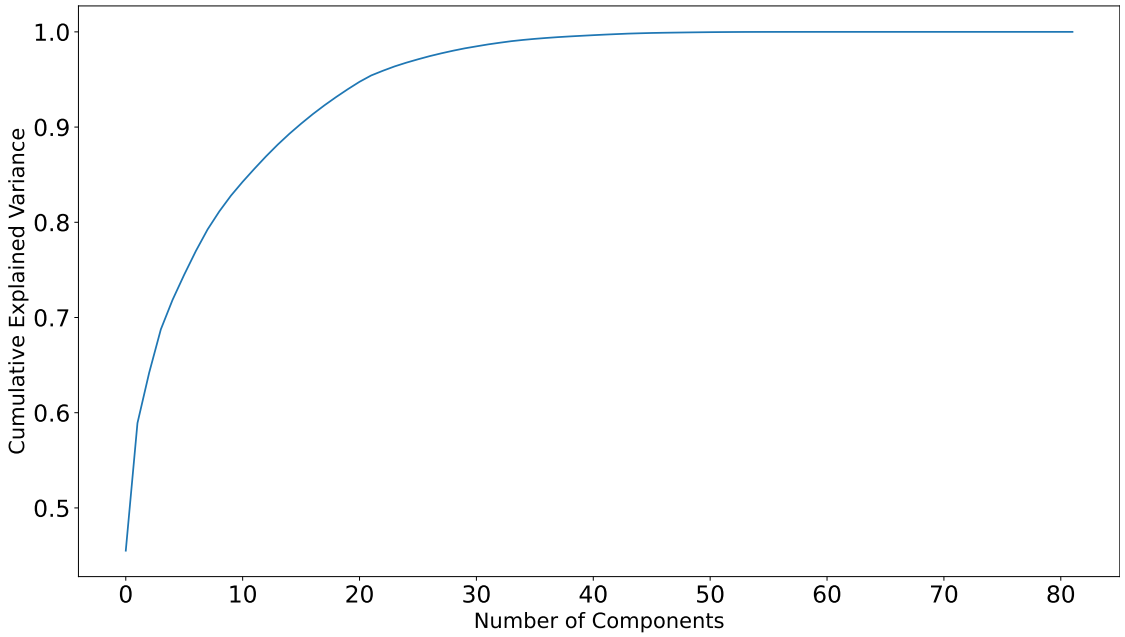
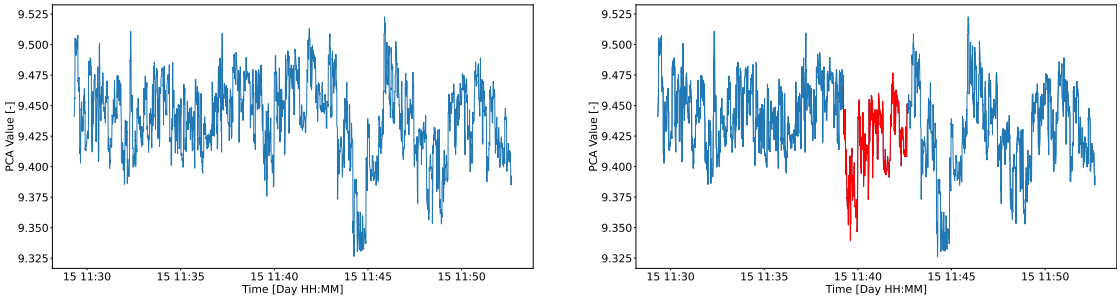


Figure 3.7.: Figure is a plot of Table 3.2, where the number of principal components is on the horizontal axis, and the resulting cumulative explained variance on the vertical axis.



(a) This figure shows the primary latent variable produced by the PCA free of anomalies.

(b) This figure shows the primary latent variable produced by the PCA after an anomaly was added to the ECD measurements. The anomalous region is colored in red.

Figure 3.8.: Two plots showing the primary principal component in the same time section when the input flow is constant.

3.2. Methods and Setup

As there are, to our understanding, not any publicly available tools for doing a fractal analysis on time-series data, will we in this section cover how it can be implemented. The tools used in this thesis were mainly Python and the publicly available libraries Numpy(Harris et al., 2020), Pandas(pandas development team, 2020), Matplotlib(Hunter, 2007), and Scipy(Virtanen et al., 2020). Numpy was used for numerical calculations, Pandas handled and structured the data, and Matplotlib was used to plot the data and results. Scipy was used for Spline interpolation to clean up the data output and piece-wise regression. All of which will be described in this section. To understand the concepts and explanations in this chapter, the reader is advised to go through the theory chapter first. A figure showing the proposed process flow for fractal analysis is shown in Figure 3.9.

3.2.1. Fractal Analysis - A Proposed Workflow

Implementing the Divider Methods

In theory Section 2.1.2, two divider methods are covered. Both methods take a data set and a divider length as input, and they output the resulting polygon length, $L(\lambda)$. We want to define a constant ruler length, λ, and a data signal as inputs in the original divider method. With the ruler length defined, each ruler's intersection/end position is found using trigonometry. The start position is where the previous ruler intercepts the input signal, as illustrated in Figure 3.10. When all the rulers are found, the length of the resulting polygon is calculated. Each ruler has the same length in the original divider method, λ. Consequently the resulting polygon length, $L(\lambda)$, is calculated as the divider length times the number of dividers, N, Equation 3.2:

$$L(\lambda) = N * \lambda \tag{3.2}$$

The fractal analysis tool implemented for this thesis utilizes the modified divider method

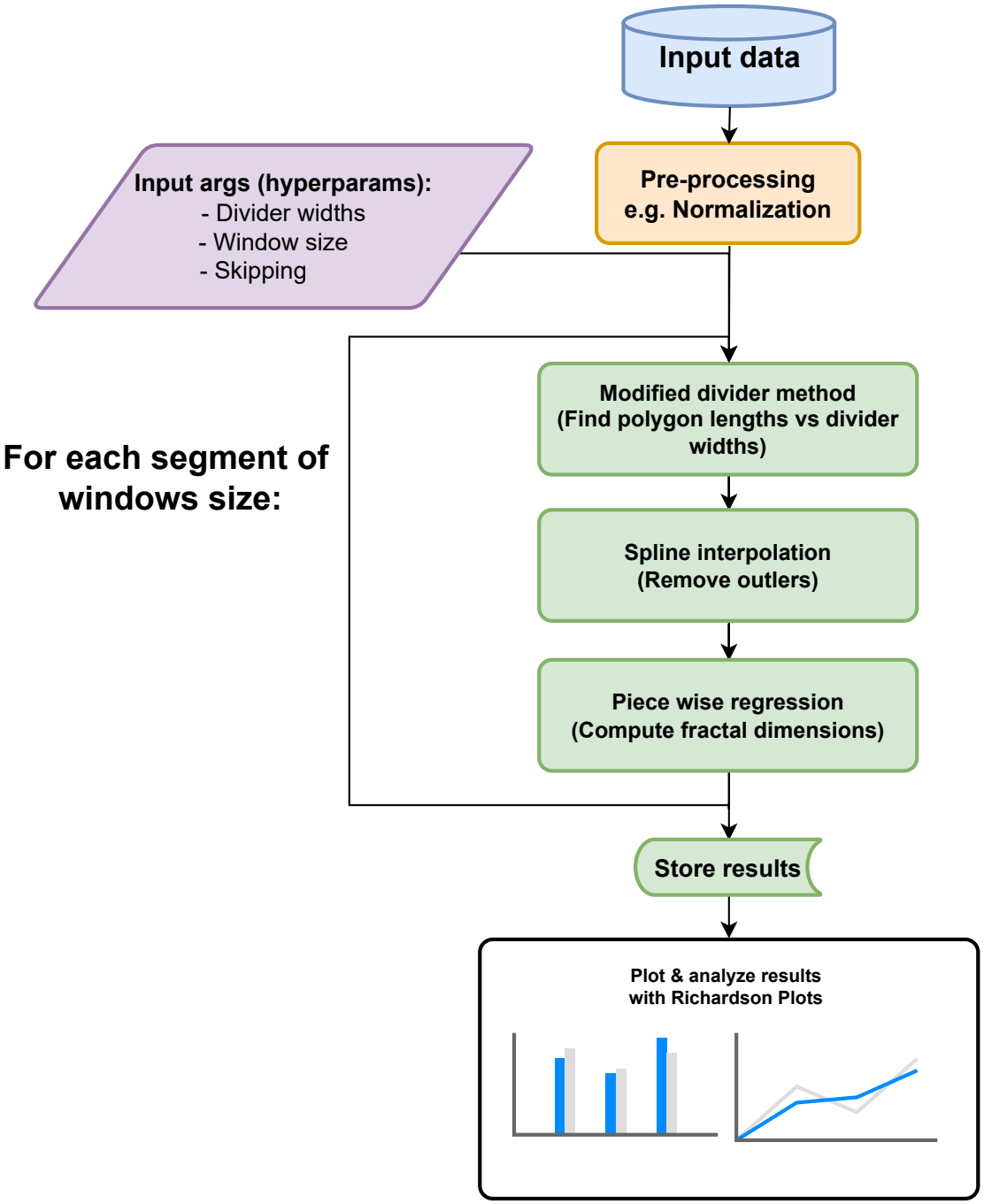


Figure 3.9.: Overview of proposed process flow for fractal analysis. Note that the pre processing block is dependent on what data set is being used, and thus varies from data set to data set.

because it is easier to realize. In the modified divider method, the input data is segmented into equal-sized sections with the width λ, and the sampling lines are drawn from start to end of each section where the section divider intersects with the input signal, as previously shown in Figure 2.3. The divider is scaled in relation to the input data to support the smallest possible divider width. That is, the smallest possible increment is the difference between two adjacent data points, given that the data is

evenly distributed. When the data is divided into sections, the slope of the sample lines is calculated and drawn. Then the length of each line is calculated using Pythagoras Theorem. The resulting total polygon length, $L(\lambda)$, is the sum of the lengths of the sampling lines. An illustrative plot of how the sample lines are drawn is seen in Figure 3.11.

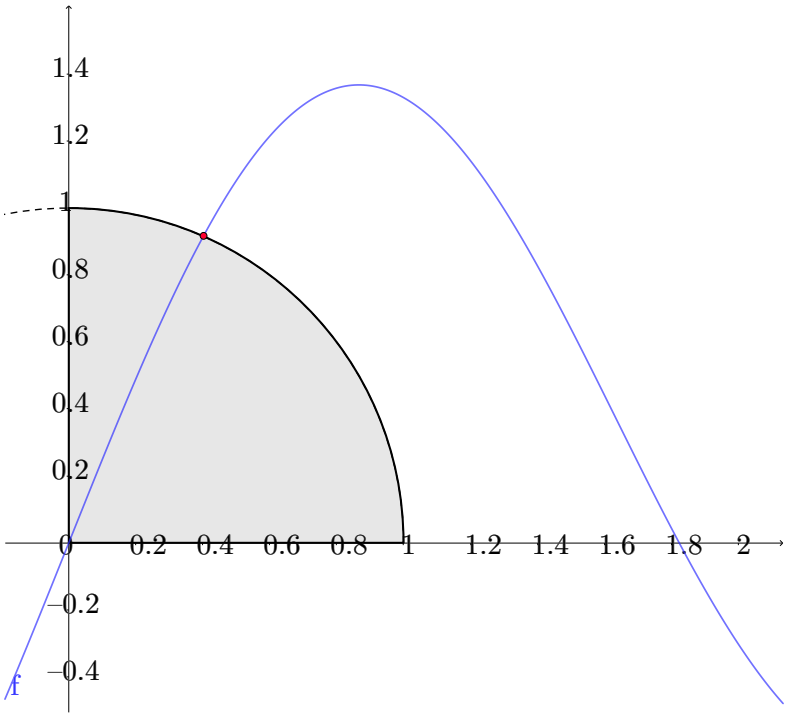


Figure 3.10.: Illustration on how the rulers start/endpoint are found in the original divider method. From the start point, in this example (0,0), a circular arc is drawn with the radius equal to the divider length (here 1 unit). The point of intersection between the arc and the input data (in blue) will be the endpoint at the given iteration. If there are multiple intersections, it is the intersection with the greatest horizontal value that will be chosen.

Finding the Fractal Dimension, D_r

With a working implementation of the modified divider method that takes a data set and a divider length as inputs, the groundwork for implementing a method that calculates the fractal dimensions is done. In theory Section 2.1.2 it is described that the fractal dimension can be found by first running the divider method on the same data segment but with different divider lengths. Then plot the divider lengths against the resulting polygon lengths in a logarithmic plot, a Richardson plot. From the Richardson plot, one can, in theory, identify one or multiple fractal dimensions by extracting the slopes of the scattered data. If we take a simple sine signal like Figure 3.12a and run the divider method on it with increasing divider width and plot the output, it comes out like Figure 3.12b. For the most part, it looks all right, but as we want to extract the slopes of the trends in the signal, it would be preferable to get rid of the outliers. There

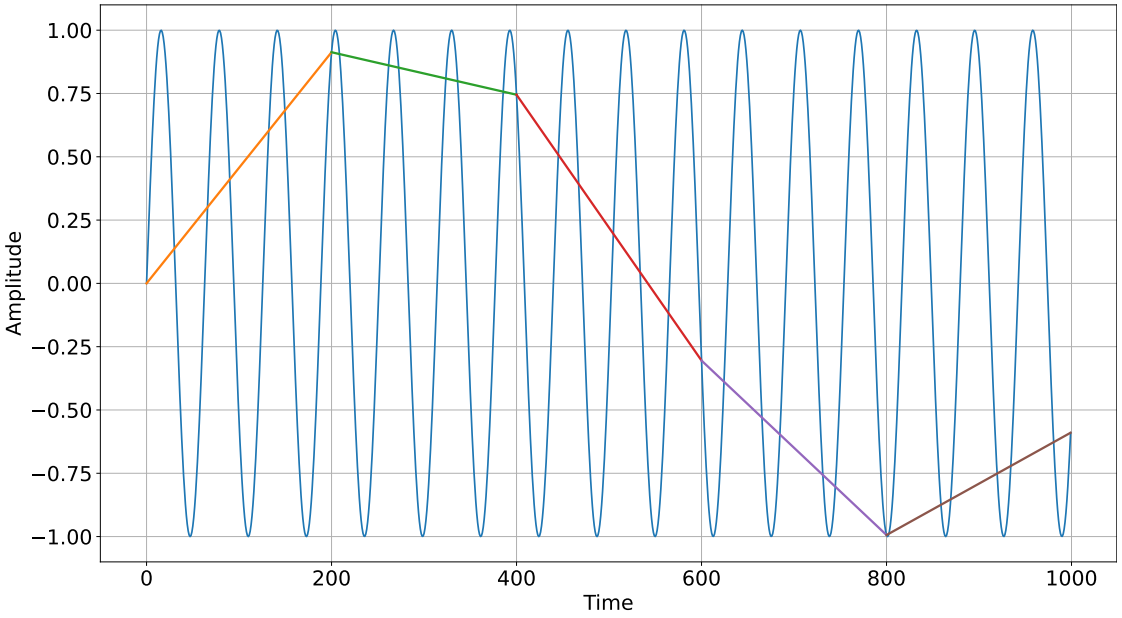


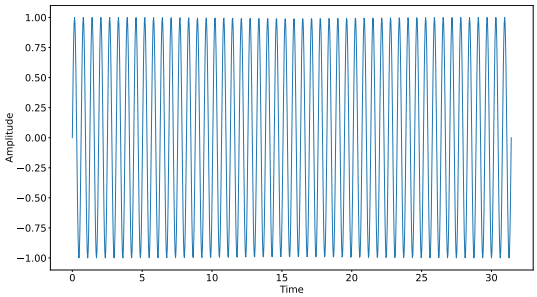
Figure 3.11.: Exampled output plot of the modified divider method. The blue wave is the input signal, the colored lines are the sample lines which are of different lengths, but they all intersect at the same interval on the horizontal axis. In this example the divider length is 200, thus the sample lines start and end exactly at each 200 mark on the horizontal axis.

are several different techniques and methods to remove outliers, and in this thesis, it is the Spline interpolation covered in theory Section 2.1.5 that is utilized. When applying Spline Interpolation on the logarithmic data to get rid of outliers, one ends up with a plot like in Figure 3.12c.

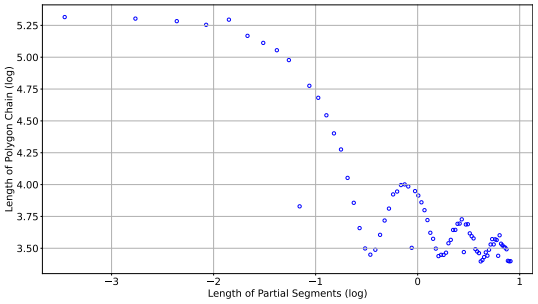
When the outliers are removed using Spline interpolation, the slopes are ready to be extracted and converted to fractal dimensions. As there are potentially multiple slopes and thus multiple fractal elements, as mentioned in Section 2.1.4, a piecewise regression model(PRM) is used to find the slope values. The underlying theory for PRM is covered in Section 2.1.6. When applying PRM on the Spline cleaned signal, the regression model iteratively finds three slope lines that fit best with the three signal trends. The resulting output plot after applying PRM is seen in Figure 3.13. Three distinct slopes are found and are converted to fractal dimensions using Equation 2.5.

Position Based Fractal Analysis/Sliding Window method

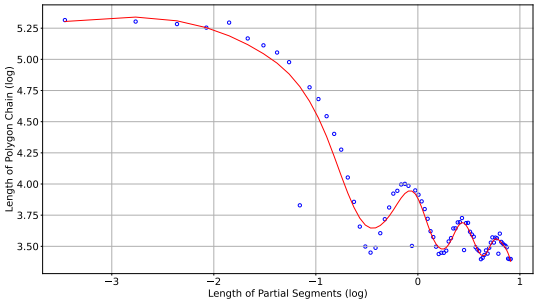
To use the proposed methods and end up with a fractal dimension, one needs a series of data for each fractal dimension calculated. When applying fractal analysis on railway data (Landgraf and Hansmann, 2019) they found the fractal dimension for every data point, i.e., position in the measurement data, and then compare the fractal dimension for each position in the data and see how it develops over time. Our proposition to implement this is by defining a fixed segment size that will work as a ”sliding window”. For each data point, run fractal analysis on the data section within the current window, then slide the window N data points and repeat the process. The variable N is the number of data points between the current window and the next. The skipping steps, N, and the window size are hyperparameters that can be tuned for a given analysis and



(a) Plot shows the input signal $A(t) = sin(\omega_A * t)$ with a large ω_A for high frequency.



(b) Plot shows the raw output when using the modified divider method on signal **A**. It has some outliers, for instance around (-1.3, 3.8) which would be beneficial to have removed.



(c) Plot of how a best fit line and interpolation would look using Spline. Here The original data points are scattered in blue, while the Spline fitted line is drawn in red.

Figure 3.12.: Three figures that illustrates the process for outlier removal in the fractal analysis.

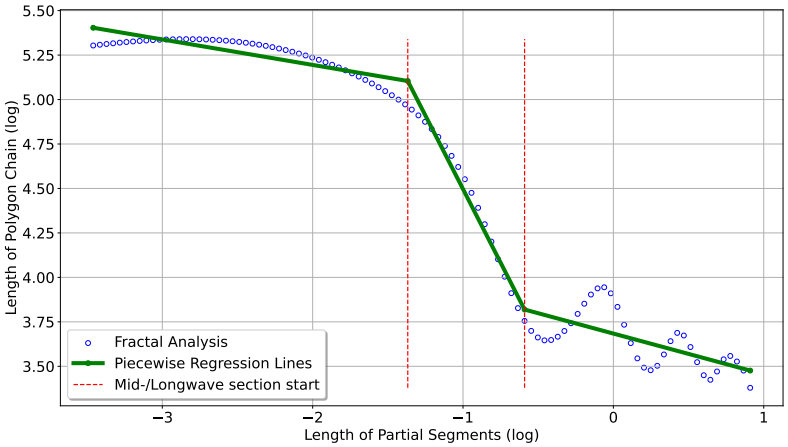


Figure 3.13.: Plot illustrates how the piecewise regression model finds three best fitting lines for the three trend sections. The red stippled lines marks the start/end of the middle-waved section.

will be covered in the next section.

Hyperparameters

For most methods in data analytics, a set of hyperparameters has to be set to adjust the method to the given data. This applies to our fractal analysis tool as well. The proposed solution has three hyperparameters, namely *divider widths*, *window size*, and *skipping*, and are shown in Table 3.3.

Table 3.3.: Table shows a list of the tuneable hyperparameters for the fractal analysis and a short description.

Hyperparameter	**Description**
Divider Widths	Range of divider widths used for each window.
Window Size	Horizontal width of the window.
Skipping	Amount of data points to skip when sliding the window. 1 signifies no skipping.

When setting the divider widths-hyperparameter, one defines the range of divider widths used in the analysis. This parameter not only defines the range of divider widths, but implicitly the amount of times the divider method is run as one invocation of the divider method function is called for each divider width. Thus a greater range of divider widths covers more wavelengths, but it is more computationally heavy.

The window size-hyperparameter defines, as its name states, the size of the interim data set of the sliding window method. This hyperparameter does not affect the number of calculations because the input data will be covered in its entirety no matter the size of the sliding window. However, the size of the window affects the resulting fractal dimension; if the window size only covers a segment with little variation, the resulting fractal dimension will be low compared to if the window covered greater regions and thus got more variation in each window. The window size also affects the analysis capability of detecting anomalies/effects that covers greater distances.

The proposed solution has a third hyperparameter, but in reality, it is rarely used. The parameter is called *skipping* and has the function of deciding the gap between each window. If the skipping parameter is set to 1, then a window will be formed at every single point on the horizontal axis; if it is 2 then it is every second point, and so on. The hyperparameter is created to reduce the number of computations if the input data is high resolution and determining the fractal dimension at each point is redundant.

Selecting the optimal hyperparameters is not a trivial task even for the most popular models to date because it is highly dependent on the problem to solve and the data at hand. To our knowledge, we have not been able to find an optimal approach for hyperparameter selection based on other papers; thus, we had to experiment on our own. The goal here was to find values that prove that the method works and not necessarily the optimal values for the hyperparameters. The hyperparameter values used in this thesis are shown in Table 3.4 and were found by experimenting with different values and iteratively running analyses on the given input data.

The process of finding suitable hyperparameter values was the same for the different experiments, therefore is only how they were found for the hole cleaning data covered in detail here. With the hole cleaning data, more precisely the ECD measurements with an anomaly added, as explained in Section 3.1.2, we have a controlled set of data where

Table 3.4.: Table shows a list of hyperparameter values used in the experiments.

Input Signal	**Divider Widths**	**Window Size**	**Skipping**
Synthetic Sine W/ Anomaly	[1,50]	200	1
ECD Measurements	[1,200]	1000	1
PCA Latent Variable	[1,60]	600	1

Table 3.5.: Table shows the hyperparameters tested on the ECD measurement data with anomaly inserted, and a description of how it performed with the given values.

Tuning of Hyperparameters			
Divider Widths	**Window Size**	**Anomaly Position and Magnitude**	**Description of Result**
[1,25]	500	Left, +10	Anomaly detected in both middle-waved and long-waved.
[1,25]	500	Left, +5	Anomaly partly detected in both ranges.
[1,25]	500	Left, +3	Anomaly not detected.
[1,60]	500	Left, +3	Anomaly partly detected in long-waved result.
[1,60]	400	Left, +3	Anomaly not detected.
[1,60]	600	Left, +3	Anomaly partly detected in middle-waved result.
[1,60]	1000	Left, +3	Anomaly partly detected in both wave ranges.
[1,60]	1000	Left, +5	Anomaly fully detected in both wave ranges, however, it seems random. Moving anomaly in next test.
[1,60]	1000	Middle, +5	Anomaly not detected, thus previous result was in fact random.
[1,200]	1000	Left, +3	Anomaly fully detected in both wave ranges. Moving the anomaly in next test.
[1,200]	1000	Middle, +3	Anomaly fully detected in long-waved, partly detected in middle-waved. Moving the anomaly in next test.
[1,200]	1000	Right, +3	Anomaly fully detected in long-waved, not detected in middle-waved.
[1,200]	**1000**	**Left, Middle, Right, +5**	**Anomaly fully detected in both wave ranges and anomaly positions. This could be ideal values.**
[1,200]	500	Right, +5	Anomaly fully detected in long-waved, not at all in middle-waved. Output has more noise.
[1,25]	1000	Right, +5	Anomaly fully detected in long-waved, not at all in middle-waved. Output has more noise.

we know where the analysis should point out the anomaly. Figure 3.6 shows the ECD measurements used for tuning the parameters. In the signal, an anomaly is added at different positions to ensure that if a selection of hyperparameters results in the analysis detecting the anomaly, it detects all the anomalies. Different magnitudes of the anomaly were tested as well; by reducing its amplitude, we can see how sensitive the tool is. Table 3.5 lists the values tested as well as a description of the result with the given values.

4. Results and Discussion

With the methods and data covered in the previous chapter, will this chapter present a series of experiments to determine if the method and the proposed implementation work as intended. Starting with the synthetic data, then the data from a the hole cleaning operation, and lastly a summary of the findings.

4.1. Experiment 1 - Synthetic Signals, Synthetic Anomalies

To verify and learn about the behavior of the fractal analysis method as a tool, a set of synthetic tests and scenarios where the tester has complete insight into the composition of the signals is a good starting point. For this reason, will the first experiment section utilize the synthetically created signals and anomalies from Section 3.1.1.

4.1.1. Three Sine Waves, Different Resulting Fractal Dimension

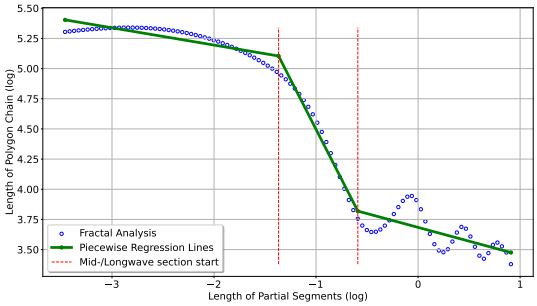
The goal of the first experiment is to see how the calculated fractal dimension corresponds to different frequency signals and a combination of frequencies. As the signals in this experiment are self-repeating, will each window be the same, and the resulting fractal dimension will be the same throughout the entirety of the signal. Therefore, a single-window analysis is sufficient to get the fractal dimensions for each signal. Starting by finding the fractal dimensions for signal $\mathbf{A}(\omega_A = 10)$ and $\mathbf{B}(\omega_B = 2)$. We expect from the theory that a signal of high frequency will have a higher fractal dimension in the middle-waved section, while a lower frequency signal will have a higher fractal dimension in the long-waved section (theory Section 2.1.2). By comparing the results from analysing signal **A** and **B** with their respective Richardson plots Figure 4.1a and Figure 4.1b does indeed the results concur with the theory. The resulting Richardson plot for signal C, seen in Figure 4.1c, has both a higher slope in the middle-waved section as well as a high slope in the long-waved section. The result could be an indication that the method captures signatures from both signal **A** and **B**, which is desirable. The next experiment is to change the wavelength of signal **B** and see how it affects the slopes in the Richardson plot from the analysis of signal **C**.

4.1.2. Different Wavelengths for Signal B

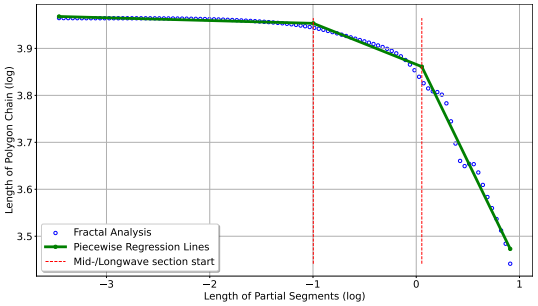
Since signal **C** is the superposition of signal **A** and **B** and to further investigate how the composition of different wavelengths affects the Richardson plot of signal **C**, will this experiment iteratively change the wavelength of Signal **B** and compare the resulting fractal dimensions for signal **C**. The setup will be as follows:

$$C_i(t) = A(t) + B_i(t) \tag{4.1a}$$

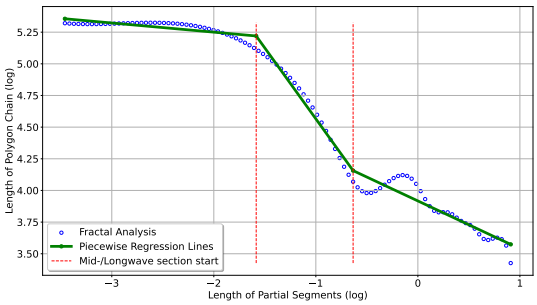
We have a set of different **C** signals, where its suffix expresses what ω_B was at that iteration, making signal **B** different for each iteration, while signal **A** remains the same. In the first experiment, the frequency affecting variable ω_B was set to 2.0, where a lower



(a) Resulting Richardson Plot for the high frequency sine waved signal **A**. Fractal elements indicated in green lines, note that the middle-waved section (middle) has the highest slope here, which concurs with the theory.

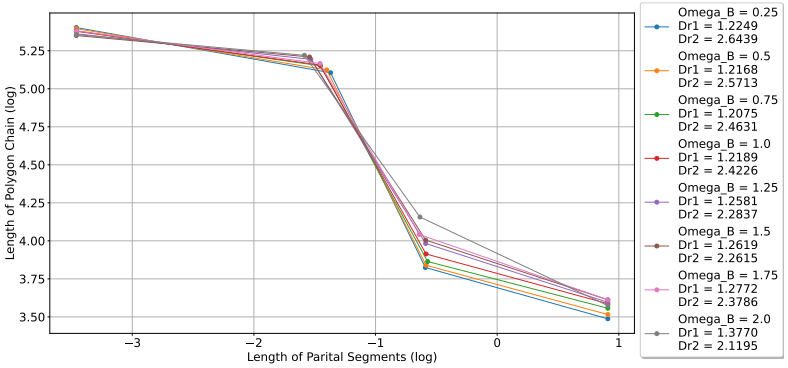


(b) Resulting Richardson Plot for the low frequency sine waved signal **B**. Fractal elements indicated in green lines, note that the long-waved section (rightmost) has the highest slope here, which concurs with the theory.

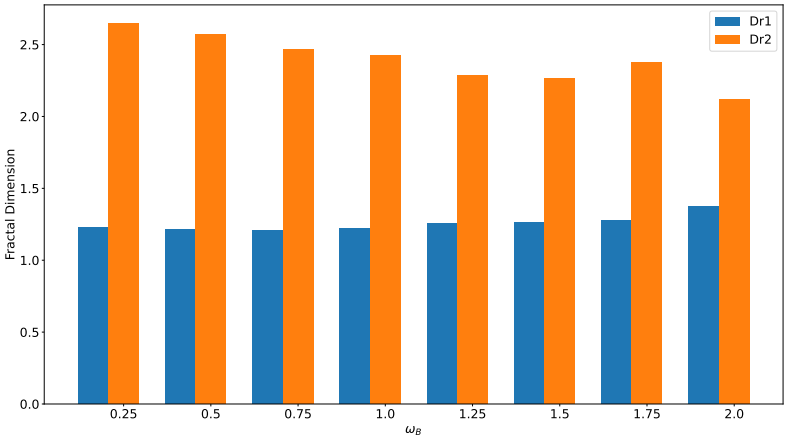


(c) Resulting Richardson Plot for signal **C**, which is a superposition of **A** and **B**. It has high slopes in both middle- and long-waved section.

Figure 4.1.: Three figures showing the resulting Richardson Plot for the synthetic sine wave signals.



(a) Richardson Plot of Signal C with ω_B ranging from 0.25 to 2.0



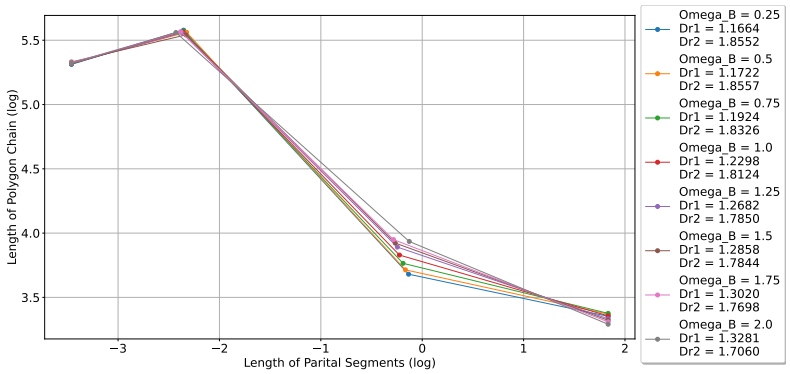
(b) Bar plot comparing the fractal dimensions where Dr_1 and Dr_2 is long-waved- and middle-waved section, respectively.

Figure 4.2.: Two figures comparing the resulting fractal dimensions when the wavelength of the input signal increases.

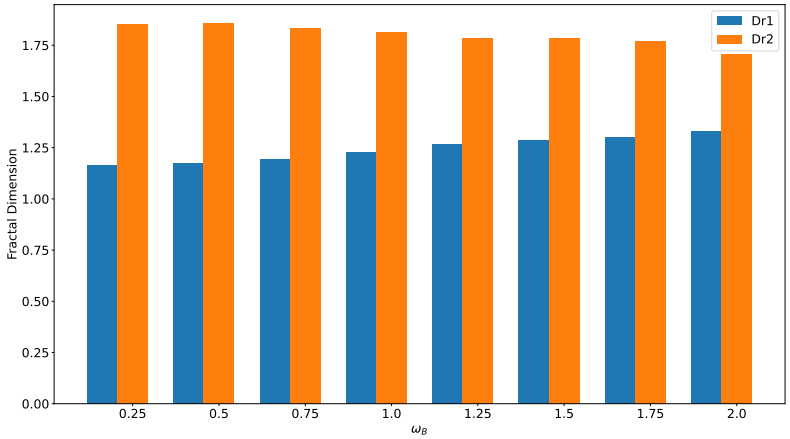
ω_B results in a longer wavelength. To test out its behavior on low frequencies and see if it is the middle-waved- or the long-waved section that is affected the most, the values were 0.25 to 2.0 with a step of 0.25 totaling in 8 different runs. A plot with the eight sets of regression lines as well as a bar plot comparing the fractal dimensions are shown in Figure 4.2. From theory, we expect that by changing the wavelength of the low-frequency component in the super-positioned signal **C** it is the long-waved section, i.e., Dr_1, that will change the most. However, the middle-waved section, Dr_2, is the most affected in our analysis. The range of divider widths for a given analysis is a hyperparameter, as explained in Section 3.2.1. Consequently, by increasing the range of divider widths for the analysis, will we end up with the results shown in Figure 4.3. The results agree with the theory that the long-waved section, Dr_1, is affected the most when the wavelength of the low-frequency signal changes.

4.1.3. Position Based Analysis with Synthetic Data

After the previous experiment confirmed that the fractal dimension responds to frequency changes, it is time to examine how the system responds when performing a position-based analysis with the sliding window method as described in Section 3.2.1. The goal for this exercise is to use the sliding window method on a standard sine signal shown in Figure 4.4a, then add a segment in the signal with different wave composition

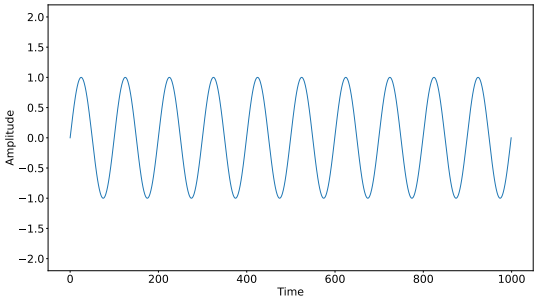


(a) Richardson Plot of Signal C with ω_B ranging from 0.25 to 2.0. Here the range of divider widths is increased for better performance.

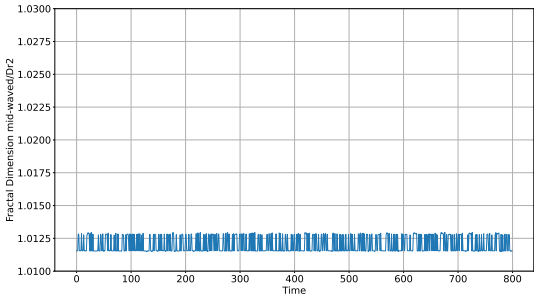


(b) Bar plot comparing the fractal dimensions where Dr_1 and Dr_2 is long-waved- and middle-waved section, respectively. Here the range of divider widths is increased for better performance.

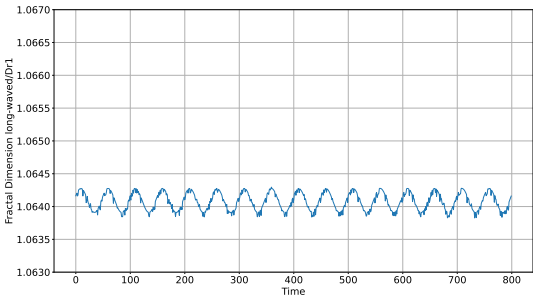
Figure 4.3.: Two figures comparing the resulting fractal dimensions when the wavelength of the input signal increases.



(a) Clean sine signal used for position based analysis.



(b) This plot shows the mid-waved fractal dimension, Dr_2, calculated at every position using the sliding window method.

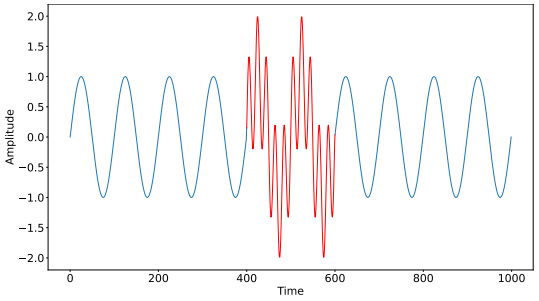


(c) This plot shows the long-waved fractal dimension, Dr_1, calculated at every position using the sliding window method.

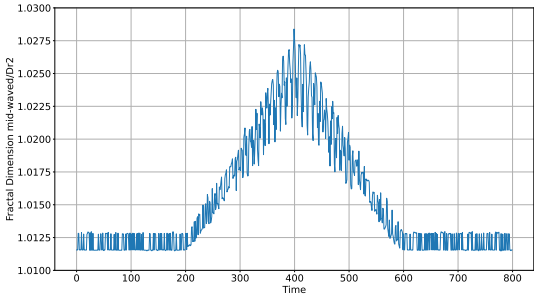
Figure 4.4.: Three plots showing the input signal and the resulting mid- and long-waved analysis results.

that will function as an anomaly, and see how it affects the results. We expect the mid-waved and long-waved fractal dimensions to be reasonably constant without any huge spikes when applying the sliding window method on a stable sine signal. The results justify the expectations, as shown in Figure 4.4b and Figure 4.4c.

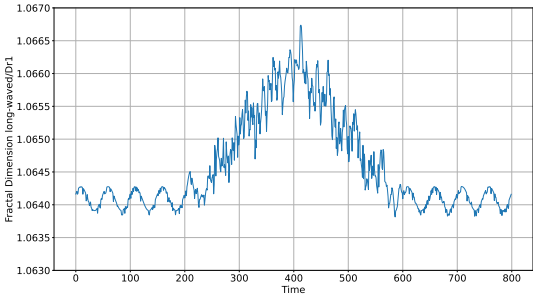
Suppose the same signal is modified by inserting a segment with varying wavelength, as shown in Figure 4.5a. In that case, we can demonstrate how the position-based fractal analysis responds to sudden changes in the signal. The resulting plots showing the fractal dimension for each position both long-waved and mid-waved are shown in Figure 4.5c and Figure 4.5b. It is clear that the method does indeed detect the sudden change in frequency. The peak value in the fractal dimension plot, Figure 4.5b, corresponds to the section in the analyzed signal with the highest fractal dimension. Highlighting this section in the original signal, as done in Figure 4.6, shows that the peak corresponds to the same area where the anomaly was positioned.



(a) Modified sine signal used for position based analysis. The red segment act as an anomaly in this experiment.



(b) This plot shows the mid-waved fractal dimension, Dr_2, calculated at every position using the sliding window method.



(c) This plot shows the long-waved fractal dimension, Dr_1, calculated at every position using the sliding window method.

Figure 4.5.: Three plots showing the input signal and the resulting mid- and long-waved analysis results.

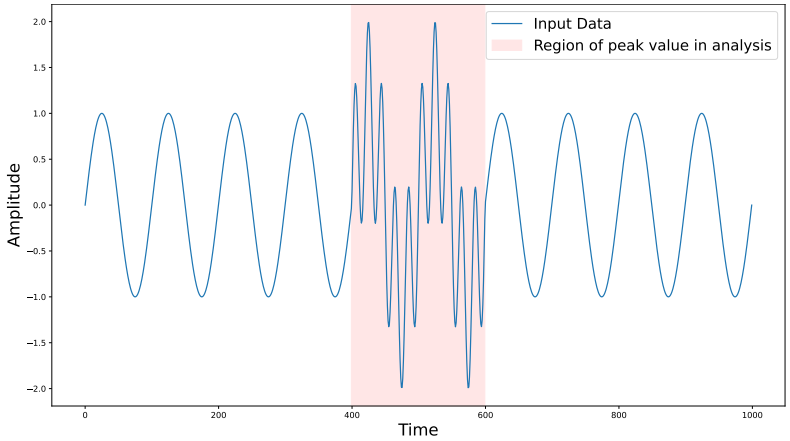


Figure 4.6.: This plot shows the original sine signal with an anomaly where the red region corresponds to the section at which the analysis found the highest fractal dimension.

4.2. Experiment 2 - Analysis of Hole Cleaning Data

Now that the proposed implementation of fractal analysis has been tested on synthetic data, it is time to experiment with the hole cleaning data. Therefore will we in this chapter conduct experiments on data from an actual hole cleaning operation.

4.2.1. Input Flow/ECD - Scenario

In Section 3.1.2 it is described how vital the ECD measurements are and that they are expected to be more or less constant during constant input flow. It is also mentioned that a related issue is when the ECD suddenly increases even though the input flow is considered constant. To be able to detect this would be of good use. For this reason, will this experiment run a fractal analysis on the measured ECD in a region where the input flow rate is constant, and later on increase the ECD and evaluate how well the fractal analysis detects it.

No Anomalies

A logical approach to evaluating the method's efficiency is to start by running an analysis on the signal without any anomalies, then introduce different anomalies in the form of increased ECD at different regions and compare the results. The signal we are analyzing in its pure form without anomalies is seen in Figure 4.7. Even though the measured ECD is in a period where the input flow rate is constant, is the ECD high-frequent but considered stable. To make it easier to follow when referring to locations in plots, will the figures in this section mostly have the data index on the horizontal axis rather than time.

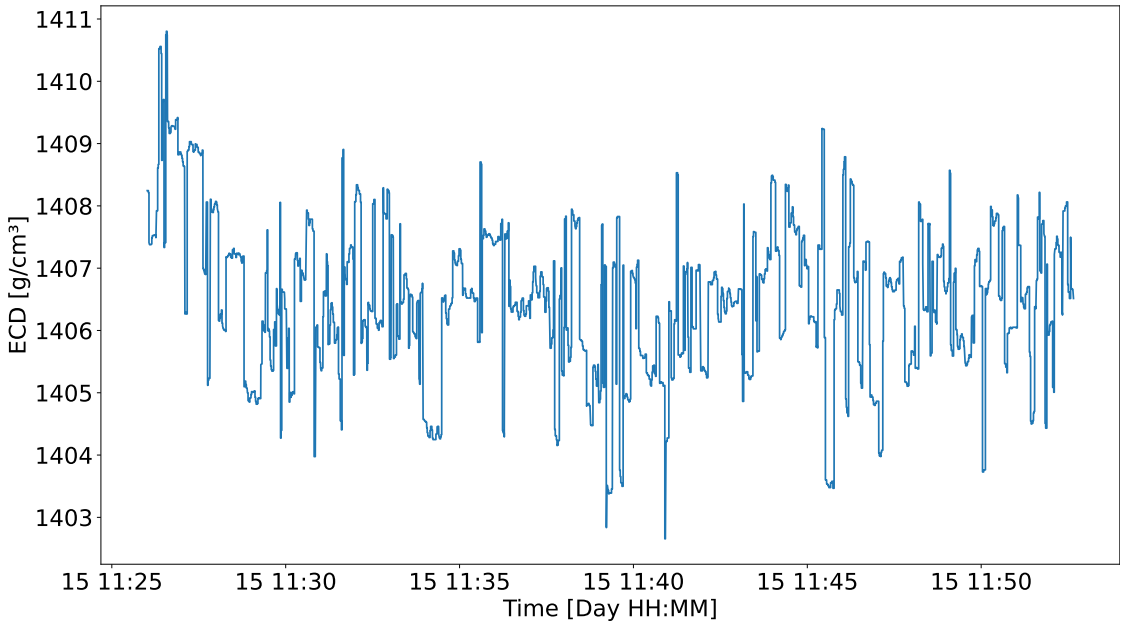
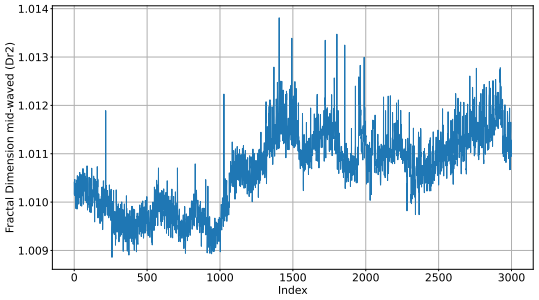


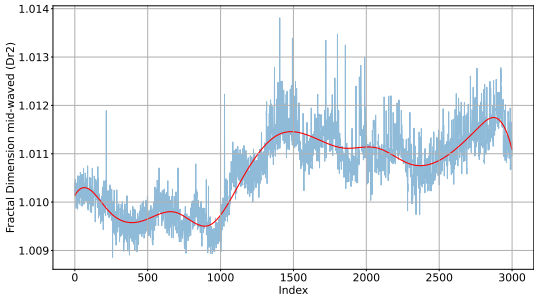
Figure 4.7.: Plot shows the measured ECD in the well in a region where the input flow is constant.

Running a fractal analysis on the ECD data with the sliding window method leaves us with two fractal dimensions, one for the middle-waved fractal dimensions and one for the long-waved fractal dimensions. Starting with the middle-waved, shown in Figure 4.8a,

we can immediately note that it is sensitive to the minor high-frequency variations in the signal. By using Spline interpolation on the output, seen in Figure 4.8b, the signal trend is easier to interpret. Since the analyzed region has stable input flow and it is known that the ECD is behaving normally, we do not get much information from the middle-waved fractal values.



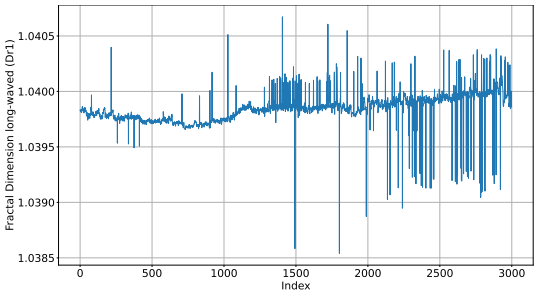
(a) This figure shows the calculated middle-waved fractal dimension for each point in the ECD data.



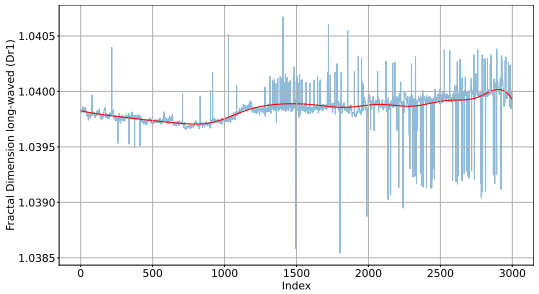
(b) This figure shows the calculated middle-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.

Figure 4.8.: Two figures showing the middle-waved Dr_2 fractal dimension results from the analysis of the ECD measurements without anomalies.

From the long-waved fractal values, on the other hand, seen in Figure 4.9a and cleaned in Figure 4.9b, we see a more desirable result where the fractal values are not varying as much as with the middle-waved values. This is because the input signal varies less in the overall amplitude, and no significant structural changes occur; thus, the output is more stable and desirable.



(a) This figure shows the calculated long-waved fractal dimension for each point in the ECD data.

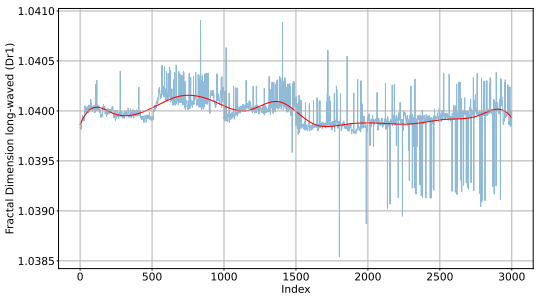


(b) This figure shows the calculated long-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.

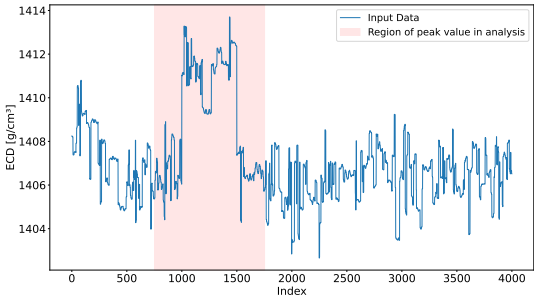
Figure 4.9.: Two figures showing the long-waved Dr_1 fractal dimension results from the analysis of the ECD measurements without anomalies.

Introducing Anomalies

With an idea of how the fractal analysis output looks when running on the ECD data without any form of anomalies, will we, in this experiment, introduce sudden increases in the ECD and examine how well the fractal analysis detects it. To make sure that it is not random if it detects the region of increased ECD, will we run the same analysis



(a) This figure shows the calculated long-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.



(b) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.

Figure 4.10.: Two figures showing the long-waved Dr_1 fractal dimension results from the analysis of the ECD measurements with anomaly on the left part of the signal.

three times with the anomaly at different positions, as shown in Figure 3.6a, 3.6b and 3.6c. To ease the readers' process, are the analysis results summarized in Table 4.1, where the columns describe anomaly placement, whether the anomaly was detected in the long-waved or middle-waved analysis, as well as a pointer to the figure of the long-waved analysis. The complete analyses with both middle-waved as well as long-waved are in the appendix A. It is clear that with the use of fractal analysis on signals like those given in these experiments, one can detect and capture sudden irregularities in the signal.

Anomaly Placement	**Mid-Wave Detection**	**Long-Wave Detection**	**Figure (LW)**
Left	No	Yes	4.10
Middle	Yes	Yes	4.11
Right	Yes	Yes	4.12

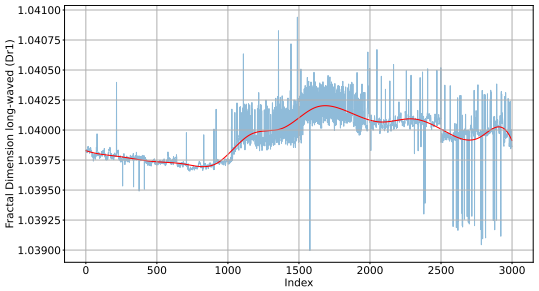
Table 4.1.: Overview of the different anomaly placements in the ECD measurements, and if the anomaly was detected by the mid- and/or long-wave analysis.

4.2.2. Analysis of Latent Variables - PCA

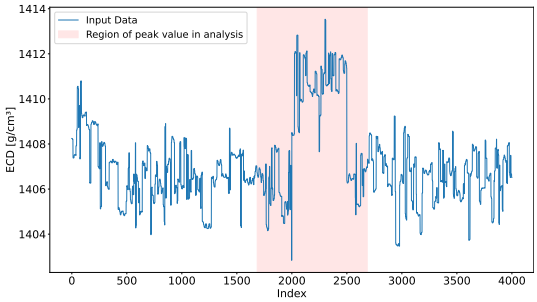
Fractal Analysis of the Primary Principal Component

With the principal components found in Section 3.1.2 will we, by using the same manner as in the previous experiments, first conduct a fractal analysis of the primary latent variable when the data set is free of anomalies, then add a sudden increase in the ECD measurement data with the same magnitude as in Section 4.2.1, and see if the analysis captures it. As we can see from the plot of the primary latent variable in Figure 3.8a, it is a signal with a lot of quick changes. If we compare the signal to the latent variable when an anomaly is added to the ECD measurement, Figure 3.8b, we can see that the anomaly of this magnitude does not have a significant impact and is barely visible to the naked eye.

Starting by running a fractal analysis on the latent variable without the insertion of anomalies results in a relatively noisy signal. However, interpolating the output using

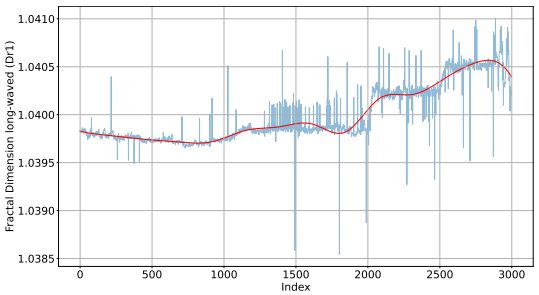


(a) This figure shows the calculated long-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.

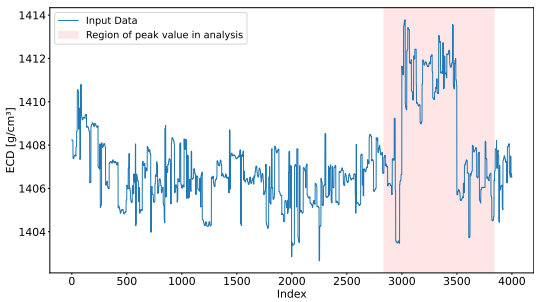


(b) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.

Figure 4.11.: Two figures showing the long-waved Dr_1 fractal dimension results from the analysis of the ECD measurements with anomaly on the middle part of the signal.

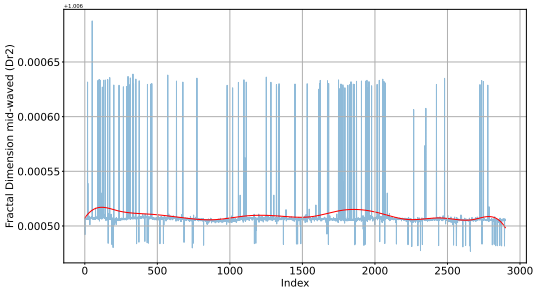


(a) This figure shows the calculated long-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.

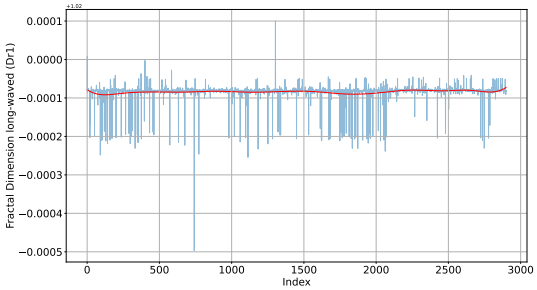


(b) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.

Figure 4.12.: Two figures showing the long-waved Dr_1 fractal dimension results from the analysis of the ECD measurements with anomaly on the right part of the signal.

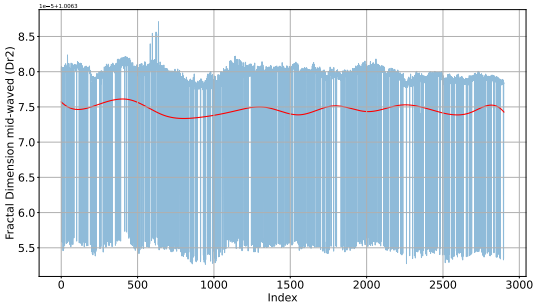


(a) This figure shows the middle-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA without anomalies in the data set.

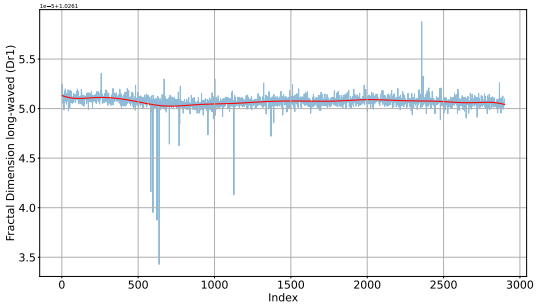


(b) This figure shows the long-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA without anomalies in the data set.

Figure 4.13.: Two plot showing the resulting fractal dimensions at each index position on the primary latent variable without anomalies in the data set.



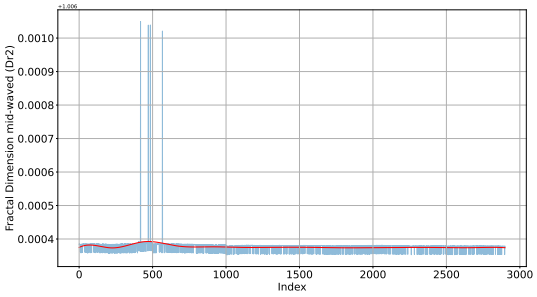
(a) This figure shows the middle-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with an anomaly in the ECD data.



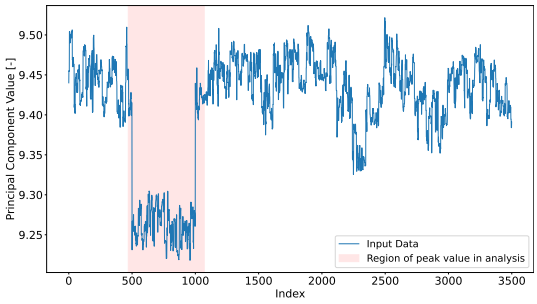
(b) This figure shows the long-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with an anomaly in the ECD data.

Figure 4.14.: Two plot showing the resulting fractal dimensions at each index position on the primary latent variable with anomaly added to the ECD data.

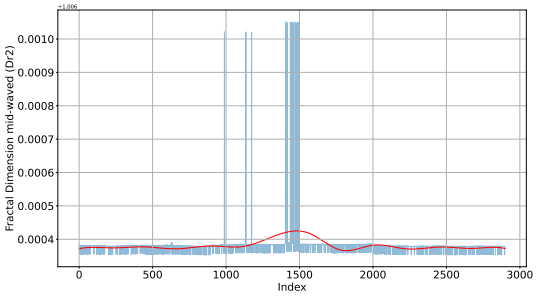
Spline to capture the trends in the analysis will make the results appear more stable, as shown in Figure 4.13. When running an analysis on the latent variable where an anomaly is inserted into the ECD measurements, Figure 4.14, we end up with no detection of the region of anomaly. This is because the anomaly in this scenario was too small to make a noticeable change in the latent variable for the fractal analysis to detect it. Nonetheless, by increasing the anomalies magnitude, the middle-waved fractal analysis is able to detect it accurately. A series of plots where the anomaly both increased and was placed at three different horizontal positions can be seen in Figure 4.15. It is clear that the method does capture the anomaly in these situations, and that it is the middle-waved part that detects it has a logical explanation. The hyperparameters for this experiment were set to values that keeps the middle-waved fractal dimension stable when the analysis is run on the signal without anomalies. Since the signal is as high frequent by nature, and the anomaly is an quick change in a signal with these characteristics, it is the middle-waved fractal dimension that changes the most. A complete list of plots of all the results, both middle-waved and long-waved is located in Appendix B.



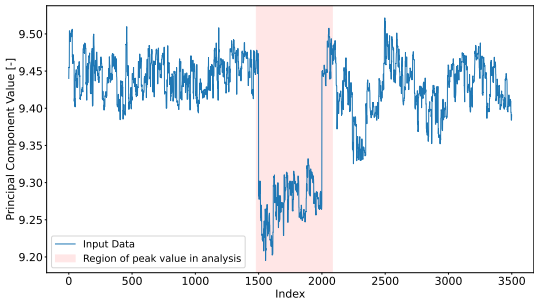
(a) This figure shows the middle-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 500-1000.



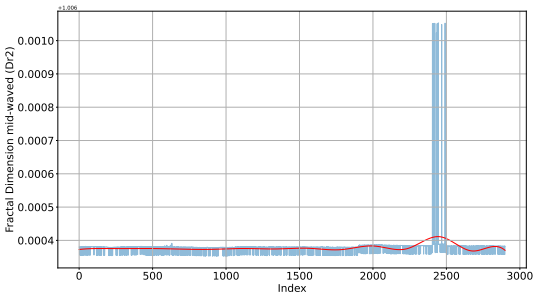
(b) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.



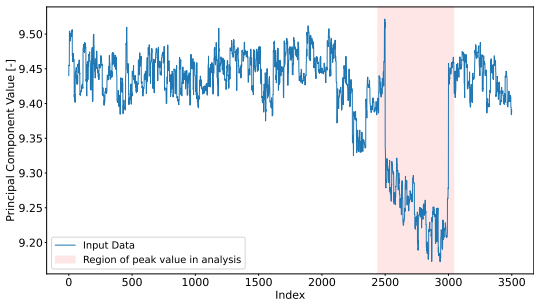
(c) This figure shows the middle-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 1500-2000.



(d) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.



(e) This figure shows the middle-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 2500-3000.



(f) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.

Figure 4.15.: Six plots showing the resulting fractal dimensions at each index position on the primary latent variable with a greater anomaly inserted at different positions in the ECD data.

4.3. Summary of Experiments

Experiment 1

The goal of the first experiment was to confirm the theory that a signal of high frequency will have a higher fractal dimension in the middle-waved section, while a lower frequency signal will have a higher fractal dimension in the long-waved section. By running the analysis on the signals composed of a single sine wave (**A** and **B**), we could both confirm the theory, and it gave us a good indication that the implemented tool is working. When analyzing the superimposed signal, **C**, which was a combination of the two sine signals, it indicated that the different fractal elements in a signal could be found because its Richardson plot showed signatures from both the low- and high-frequency signals. However, we had to conduct more experiments because one simple analysis alone was not sufficient to be sure. This led us to run multiple analyses of the superimposed signal **C**, where the frequency of one of the underlying signals was increased in each analysis. Since it was the underlying low-frequency signal, **B**, that was altered, we expected from the theory that it mainly was the long-waved fractal dimension, Dr_1, that would change in the resulting Richardson plot. At first, the results were the opposite. With the increase of frequency, the mid-waved fractal dimension, Dr_2, was changing. After increasing the range of divider widths in the analyses, we got the results we wanted and learned the importance of correct hyperparameter values.

After affirming that the implementation was working as expected, the following experiment was synthesized to see if the method could capture sudden irregularities in a signal using a sliding window method, as this would be its primary purpose when doing experiments on real measurement data. This experiment showed promising results with how it perfectly captures the irregularity in both the long-waved and middle-waved analysis, shown in Figure 4.5 and 4.6.

Experiment 2

The second set of experiments focused on the use of real measurement data from a hole cleaning operation. It was divided into two parts: The first was directly running analysis on the equivalent circulation density measurements (ECD), both with and without anomalies. In the second part, a principal component analysis (PCA) was conducted to see if abnormalities could be detected from the resulting latent variables from the PCA, enabling the tool's users only to run one analysis on the latent variable instead of analyzing individual features.

When running an analysis of ECD measurements free of anomalies, the resulting middle-waved fractal dimension plot, Figure 4.8, was both noisy and not very constant. This could be because the ECD measurement was relatively inconsistent, even though it was from a region of constant input flow. It also should be taken into consideration that not every hyperparameter combination was tested. Nonetheless, with the hyperparameter values that were used, the long-waved fractal dimension plot, Figure 4.9, was seemingly stable. After introducing anomalies, it detected them in all cases, which is very promising.

The last part of Experiment 2 concluded the experiments by fractal analysis of the latent variable returned by the principal component analysis of the hole cleaning data.

The fractal dimension plots for both middle- and long-wave were noisy yet consistent, as seen in Figure 4.13. Nonetheless, when analyzing the latent variable after adding the same irregularity to the ECD measurement as in the previous experiment, it was not detected at all, as seen in Figure 4.14. It is nearly impossible to see the anomaly by visually inspecting the latent variable if one does not know in what region it is located. More optimal hyperparameters might improve the results. For instance, if the sliding window size was large enough to cover both a regular and irregular region simultaneously, it might detect a difference. The downside of this approach is that it would be less precise because if it were to detect some irregularity, it would be in such a large region/time span that it would be hard to find its source amongst all the measured features. When the magnitude of the anomaly was increased, it was detected by the analysis in every attempt, as seen in figure 4.15. A verdict is that the method captures irregularities in the latent variable signal, given that the abnormality is odd enough.

5. Conclusion and Future Work

5.1. Conclusion

In the introduction to this master's thesis, a primary research objective was set, and secondary objectives led to the primary objective. Through the experiments in this thesis, we have ended with the following findings:

- *How can a full fractal analysis workflow be implemented?*
  As the referenced literature does not clearly communicate how one can implement fractal analysis, we had to develop the whole workflow from scratch. Starting with converting theory about divider methods to code, then outliers became an issue, thus sidetracking us on outlier removal. This led us to the use of Spline interpolation for outlier removal. As the slope of the trends of the signals had to be extracted, we had to figure out a robust way to do this. Piece-wise regression showed to be a good candidate for this. After tangling all the pieces together, we ended up with a fully working tool for fractal analysis.

- *How can fractal analysis be utilized for analysing a hole cleaning process?*
  Fractal analysis has proven useful through experiments with actual hole cleaning data when handpicking features like the measured equivalent circulating density under constant input flow. Fractal analysis can also be applied to the latent variable returned by a principal component analysis, albeit with poorer performance than with handpicked features. We have proven that using publicly available tools and synthetic data, a fully working and reproducible workflow for conducting fractal analysis on time-series data is realizable.

- *How can the reader learn about and make use of fractal analysis themselves?*
  To our knowledge, there are no available pedagogical sources on how fractal analysis is used or implemented for time-series data. For this reason, we have created a fully working and publicly available Github repository (Musæus, 2022), with examples and code that can be used to reproduce the results presented in this thesis. The implementation supports any kind of time-series data, and accordingly, the reader can experiment with data from new domains.

5.2. Reflection and Future Work

This master's thesis has personally been a great journey, from coming across the theory of fractal analysis to having a working implementation and conducting helpful experiments for industry-leading analysts. With that being said, there have been some bumps on the way. The initial plan for the project was to use railway data. Unfortunately, the first month or more was spent waiting for the data to be prepared, which is often the reality when cooperating with the industry. For them, the time horizon often is longer than a "short" project like a master's thesis. If I were to do the project a second time, I would not hesitate to change the data provider to someone that readily has the data.

By doing so, I would avoid wasting weeks on frustrating bureaucracy and could start working from the get-go. On the other hand, the time spent waiting for data was not wholly wasteful because, during this period, I got to read and explore more about fractal analysis. By the time we were done experimenting with synthetic data and were ready to test the method on field data, we got hold of the hole cleaning data, which ended up being a great candidate for us and the provider.

Through the works of this master's thesis and as asserted in the conclusion, we have developed and implemented a workflow for conducting fractal analysis on time-series data. Nonetheless, there are challenges in the current implementation, and as time was not sufficient, are we left with some exciting topics worth looking into if the project is continued.

Hyper-parameter tuning and optimization is an active field of research within data science and machine learning. As we did not have time to create a systematic and sustainable approach to find optimal hyper-parameters, is this highly advisable to look into. A solid starting point would be to get a better physical interpretation of the results, by finding a strong causality and correlation between the fractal dimension returned from the analysis and its physical implication.

With the data used in this thesis, anomalies were manually added to test the method. However, if one has access to data of scenarios where anomalies are known, it could make even higher quality experiments. Lastly, the current implementation has a computational cost of $O(n)$; however, it is possible to multi-thread it to make it faster. This would make the process of conducting experiments more efficient, enabling the analyst to do more experiments in less time, and the possibility of running the method on a system in real-time would be realistic.

Bibliography

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Robert Andrle. Estimating fractal dimension with the divider method in geomorphology. *Geomorphology*, 5(1):131–141, 1992. ISSN 0169-555X. doi: https://doi.org/10.1016/0169-555X(92)90061-R. URL https://www.sciencedirect.com/science/article/pii/0169555X9290061R. Fractals in Geomorphology.

S.L. Brunton and J.N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2022. ISBN 9781009098489. URL https://books.google.no/books?id=rxNkEAAAQBAJ.

Alceu Ferraz Costa, Gabriel Humpire-Mamani, and Agma Juci Machado Traina. An efficient algorithm for fractal analysis of textures. In *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 39–46, 2012. doi: 10.1109/SIBGRAPI.2012.15.

James F. Epperson. On the runge example. *The American Mathematical Monthly*, 94 (4):329–341, 1987. doi: 10.1080/00029890.1987.12000642. URL https://doi.org/10.1080/00029890.1987.12000642.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

James P. Hyslip. Fractal analysis of track geometry data. *Transportation Research Record*, 1785(1):50–57, 2002. doi: 10.3141/1785-07. URL https://doi.org/10.3141/1785-07.

Cadima J. Jolliffe, I. T. Principal component analysis: a review and recent developments. 2016. doi: 10.1098/rsta.2015.0202. URL https://doi.org/10.1098/rsta.2015.0202.

Jay Kappraff. The geometry of coastlines: a study in fractals. *Computers Mathematics with Applications*, 12(3, Part 2):655–671, 1986. ISSN 0898-1221. doi: https://doi.org/10.1016/0898-1221(86)90417-7. URL https://www.sciencedirect.com/science/article/pii/0898122186904177.

Sudesh Kumar et al. Efficient k-mean clustering algorithm for large datasets using data mining standard score normalization. *Int. J. Recent Innov. Trends Comput. Commun.*, 2(10):3161–3166, 2014.

M Landgraf and F Hansmann. Fractal analysis as an innovative approach for evaluating the condition of railway tracks. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 233(6):596–605, 2019. doi: 10.1177/0954409718795763. URL https://doi.org/10.1177/0954409718795763.

B.B. Mandelbrot, W. H. Freeman, and Company. *The Fractal Geometry of Nature*. Einaudi paperbacks. Henry Holt and Company, 1983. ISBN 9780716711865. URL https://books.google.no/books?id=SWcPAQAAMAAJ.

Sky McKinley and Megan Levine. Cubic spline interpolation. *College of the Redwoods*, 45(1):1049–1060, 1998.

Lars Gjardar Musæus. Railway track condition monitoring and data-driven predictive maintainance. 2021.

Lars Gjardar Musæus. Fractal analysis github repository, 2022. URL https://github.com/larsgmu/fractal_analysis.

The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL https://doi.org/10.5281/zenodo.3509134.

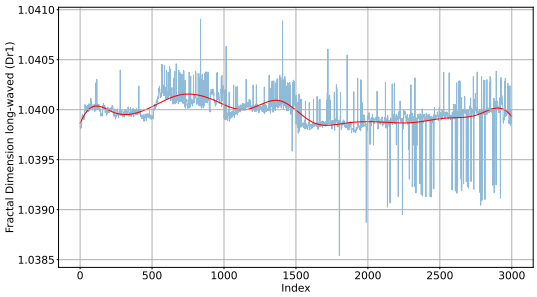
Roberto Quevedo, Christopher Brown, Pedro Bouchon, and José Aguilera. Surface roughness during storage of chocolate: Fractal analysis and possible mechanisms. *Journal of the American Oil Chemists' Society*, 82:457–462, 06 2005. doi: 10.1007/s11746-005-1093-2.

Gerald Raabe and Scott Jortner. Chapter one - well control discussion and theories. In Gerald Raabe and Scott Jortner, editors, *Universal Well Control*, pages 1–77. Gulf Professional Publishing, 2022. ISBN 978-0-323-90584-8. doi: https://doi.org/10.1016/B978-0-323-90584-8.00001-0. URL https://www.sciencedirect.com/science/article/pii/B9780323905848000010.

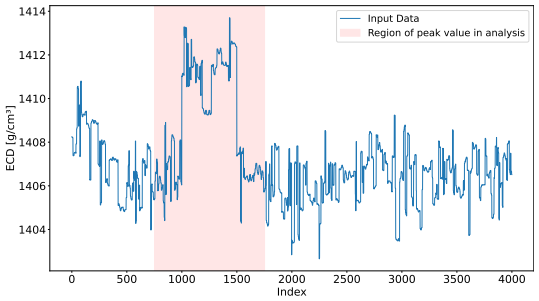
Pauli Virtanen et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Feifei Zhang, Andrey Filippov, Stefan Miska, and Mengjiao Yu. Hole cleaning and ecd management for drilling ultra-long-reach laterals. 03 2017. doi: 10.2118/183786-MS.

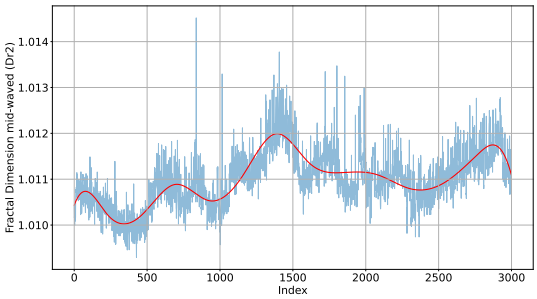
A. ECD Analysis Results



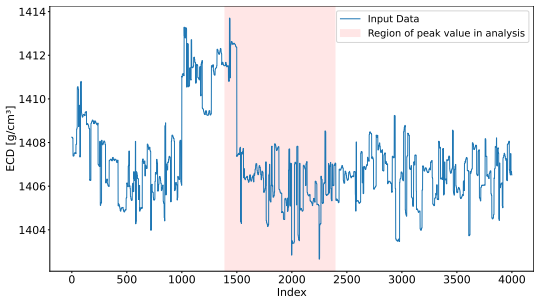
(a) This figure shows the calculated long-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.



(b) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.

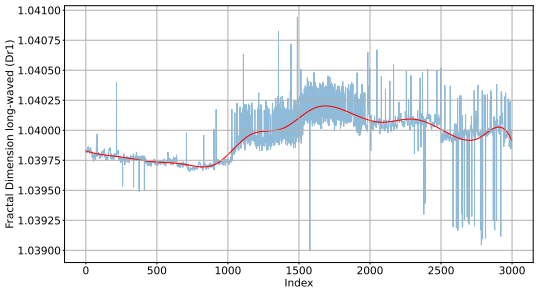


(c) This figure shows the calculated middle-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.

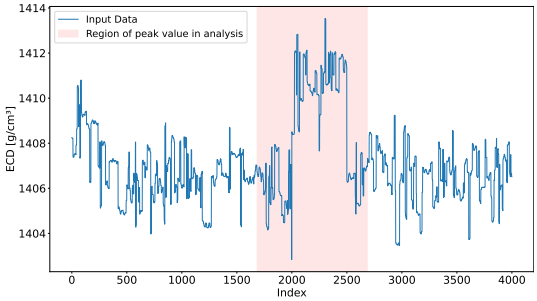


(d) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.

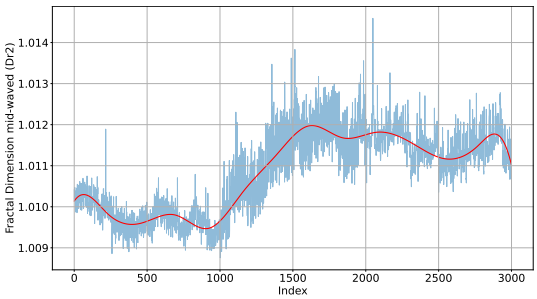
Figure A.1.: Four figures showing the middle-waved Dr_2, and long-waved Dr_1 fractal dimension results from the analysis of the ECD measurements with an anomaly on the left part of the signal.



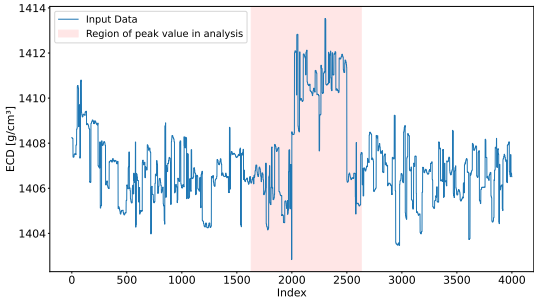
(a) This figure shows the calculated long-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.



(b) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.

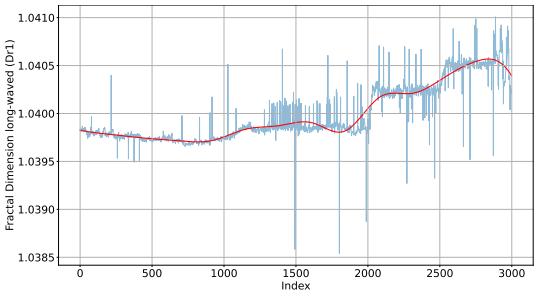


(c) This figure shows the calculated middle-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.

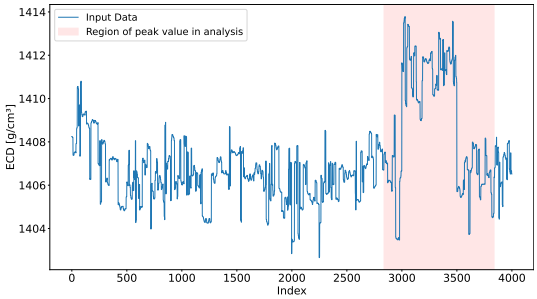


(d) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.

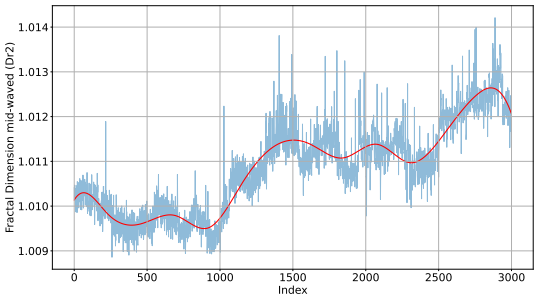
Figure A.2.: Four figures showing the middle-waved Dr_2, and long-waved Dr_1 fractal dimension results from the analysis of the ECD measurements with an anomaly on the middle part of the signal.



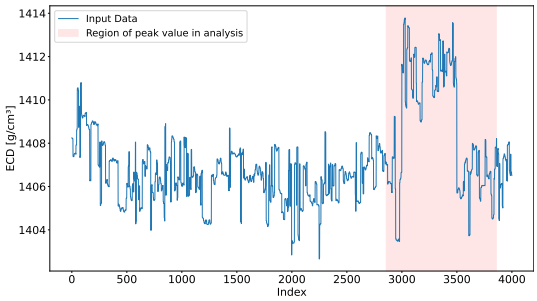
(a) This figure shows the calculated long-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.



(b) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.



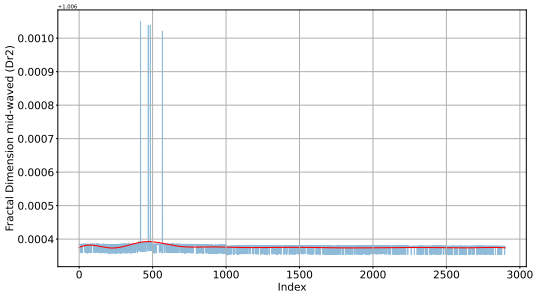
(c) This figure shows the calculated middle-waved fractal dimension for each point in the ECD data in blue, and the spline interpolation of the results in red.



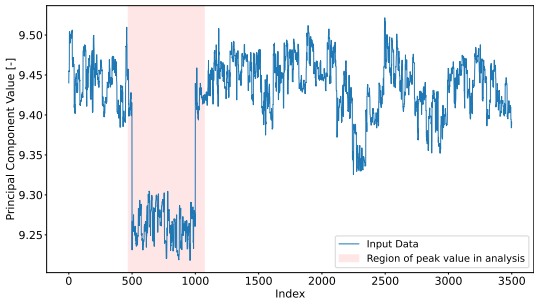
(d) This figure takes the peak fractal value in the analysis and highlights the segment which produced said value.

Figure A.3.: Four figures showing the middle-waved Dr_2, and long-waved Dr_1 fractal dimension results from the analysis of the ECD measurements with an anomaly on the right part of the signal.

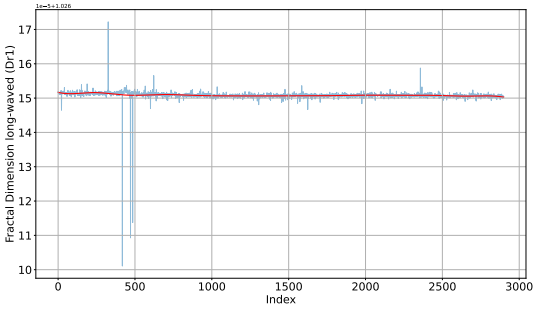
B. Latent Variable Analysis Results



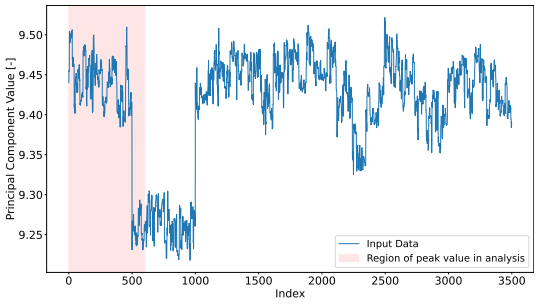
(a) This figure shows the middle-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 500-1000.



(b) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.

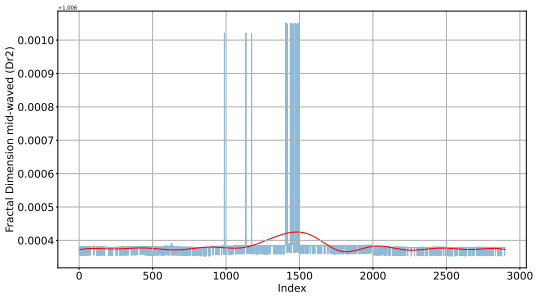


(c) This figure shows the long-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 500-1000.

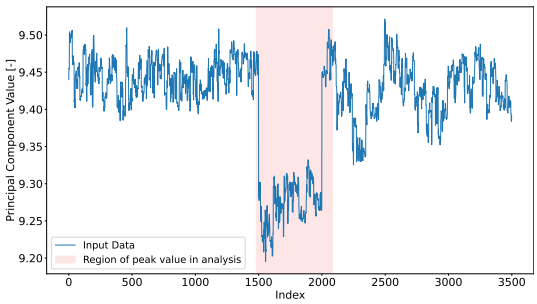


(d) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.

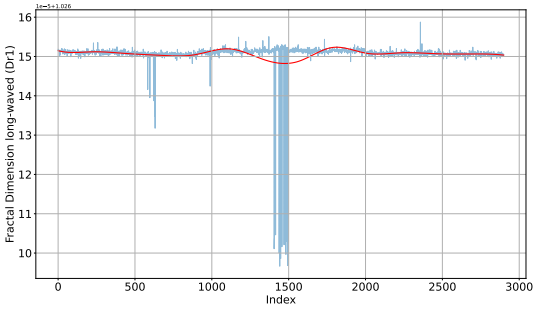
Figure B.1.: Four plots showing the resulting fractal dimensions at each index position on the primary latent variable with an anomaly inserted at the beginning of the ECD data.



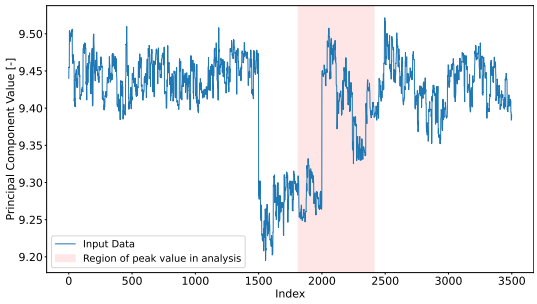
(a) This figure shows the middle-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 1500-2000.



(b) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.

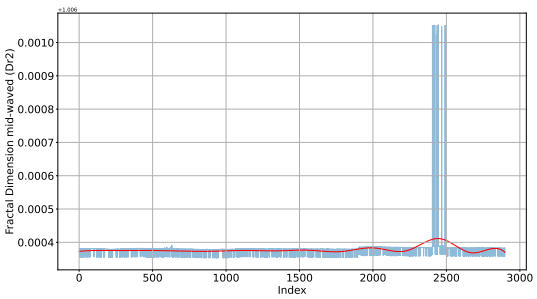


(c) This figure shows the long-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 1500-2000.

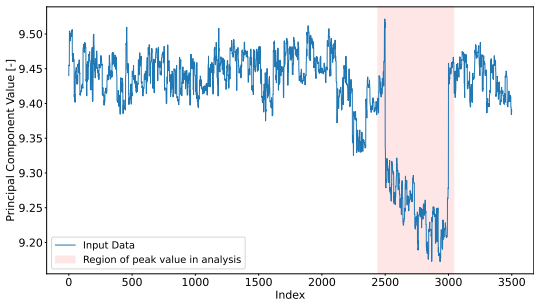


(d) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.

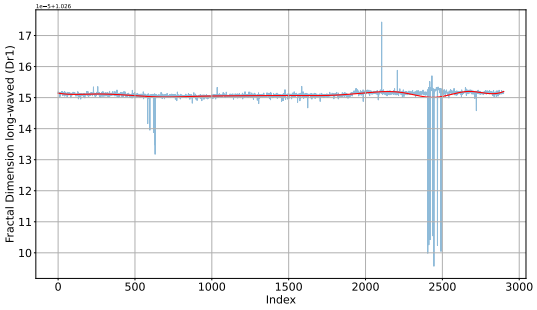
Figure B.2.: Four plots showing the resulting fractal dimensions at each index position on the primary latent variable with an anomaly inserted in the middle of the ECD data.



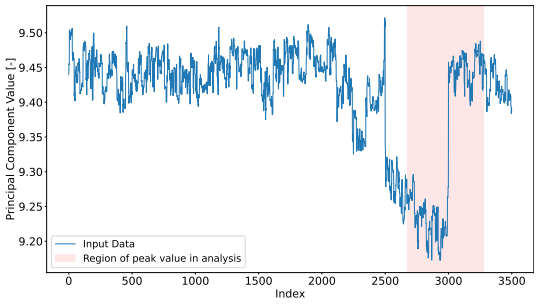
(a) This figure shows the middle-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 2500-3000.



(b) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.

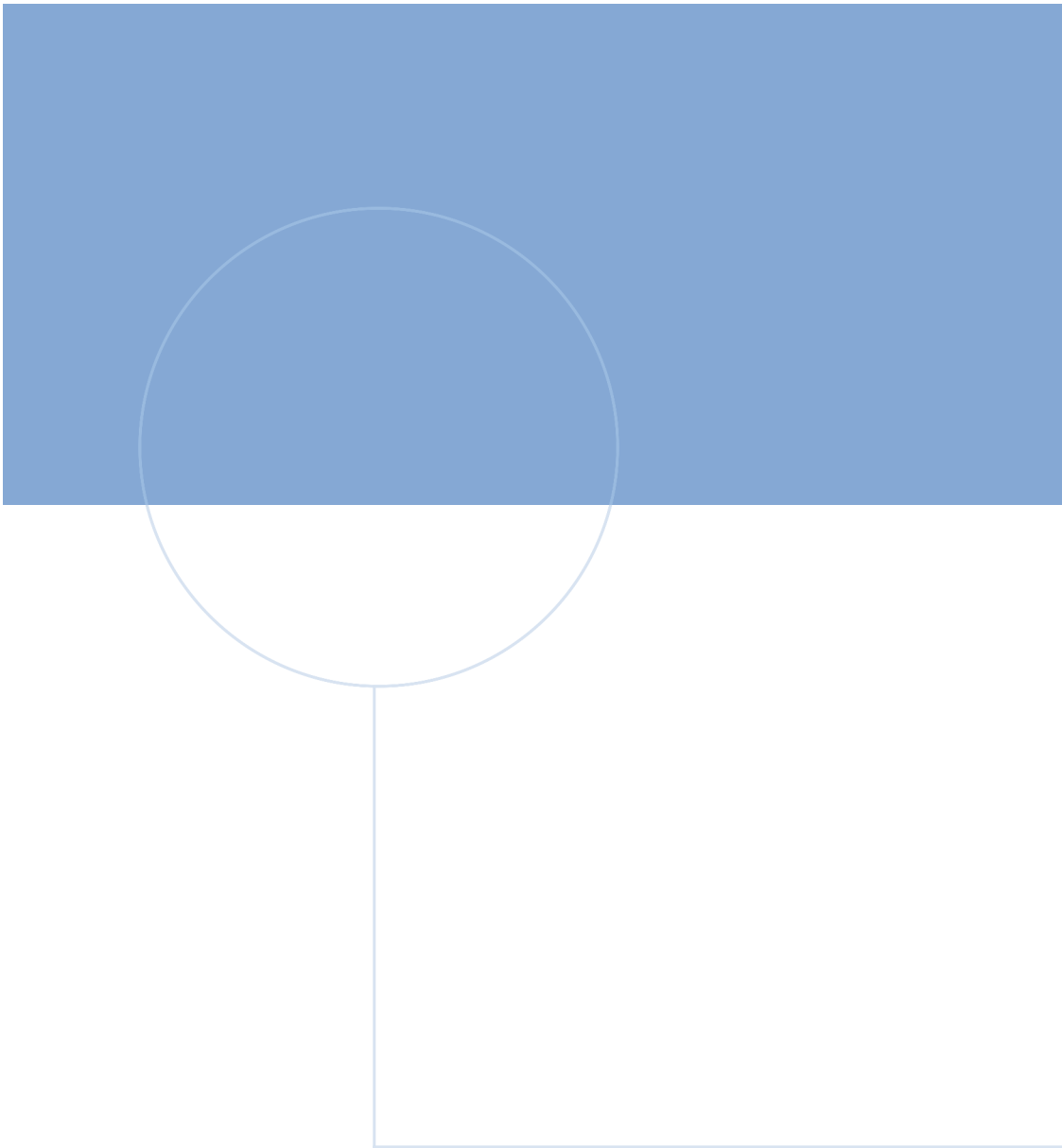


(c) This figure shows the long-waved fractal dimensions after running a fractal analysis on the primary latent variable produced by the PCA with anomalies in the data set at positions 2500-3000.



(d) This figure shows the latent variable of the anomalous data set produced by the PCA, where the red highlighted region is where the fractal analysis found its peak value.

Figure B.3.: Four plots showing the resulting fractal dimensions at each index position on the primary latent variable with an anomaly inserted towards the end of the ECD data.



NTNU

Norwegian University of
Science and Technology