

LUNDS TEKNISKA HÖGSKOLA

EXAMENSARBETE

A parametric approach to data cleaning of
ion current measurements

Fredrik ANDRÉEN BLOMDELL

andreenblomdell@gmail.com

September 2, 2017

Handledare LTH:
Andreas JAKOBSSON
aj@maths.lth.se

Examinator:
Johan LINDSTRÖM
johanl@maths.lth.se

Abstract

In this thesis, a few possible methods to clean nuisance signals from ion current measurements are explored. Ion current measurements can be used to optimise a combustion engine, but interference from the ignition system coupled with clipping caused by limitations in the measurement equipment limit the amount of data that can be extracted. This thesis develops a model which is used for simulation of the ion current measurements, that is then used to construct an algorithm to remove the interference. The algorithm developed uses a parametric approach as the interference has a distinct structure that can be exploited. A corner stone of the developed algorithm is the HTLS-algorithm, which estimates frequency and damping coefficients of uniformly sampled data. The thesis implements a recursive HTLS to exploit the structure of the interference efficiently. This method performs well on simulated data, but testing on real data shows that it needs further refinement if it is to be used commercially.

Keywords: Ion current measurements, Data cleaning, Coil ringing

Acknowledgements

I'd like to extend special thanks to my supervisor Andreas Jakobsson, for lending me his expertise. To Jakob Ängeby and Kristina Hellström, for always having answers to my questions. To Johan Swärd and Filip Elvander for discussing different problems I have had along the way. My dear friends Emma and Emelie, who proofread the report and gave me good advice along the way. To Foffe, for sharpening my table tennis skills during our longer-than-necessary lunch breaks.

Contents

Glossary	5
1 Introduction	6
1.1 Objective	6
1.2 Limitation	6
1.3 Structure of the report	6
2 Background	8
2.1 Internal Combustion Engines	8
2.1.1 Basic engine operation	8
2.1.2 Optimisation of internal combustion engines	10
2.2 Ion current	10
2.2.1 Spark generation	11
2.2.2 The combustion process	12
2.2.3 Ion current measurements	12
3 Theory	15
3.1 Sinusoid	15
3.2 Least Squares	15
3.2.1 Ordinary Least Squares	15
3.2.2 Amplitude and phase estimation of damped sinusoid with ordinary least squares	16
3.3 HTLS	17
3.4 Digital filters	18
3.4.1 Finite impulse response (FIR) filters	18
3.5 Monte Carlo simulation	20
3.6 RMS error	21
4 Method	22
4.1 Simulation of ion current	22
4.1.1 Analysis of ion current measurements	22
4.1.2 Implementation of ion current model	27
4.2 Low pass filtering with clipped data	29

4.2.1	Methodology	30
4.2.2	Results and discussion	33
4.2.3	Conclusions	37
4.3	Estimating ringing parameters	37
4.3.1	Methodology	37
4.3.2	Conclusion	41
4.3.3	Conclusion	42
5	Algorithm	44
5.1	Proposed algorithm	44
5.1.1	Algorithm settings	46
5.2	Testing - Simulated data	47
5.2.1	Methodology	47
5.2.2	Results	48
5.3	Testing - Real data	50
5.3.1	Methodology	51
5.3.2	Results	51
6	Discussion	55
6.1	Methodology	55
6.1.1	Simulation model	55
6.1.2	Low pass filter tests	56
6.1.3	HTLS	56
6.2	Algorithm	57
6.2.1	Implementation	57
6.2.2	Future work	57
7	Conclusion	59
	Appendices	62
	A Simulate ion current script	63
	B Algorithm	65
	C Algorithm - parameters	68

Glossary

AFR Air Fuel Ratio.

BDC Bottom Dead Center. The bottom most position of the piston..

RMSE Root Mean Square Error.

RPM Revolutions Per Minute. Essentially engine speed.

SNR Signal-to-Noise-Ratio, a measure of how strong a signal is compared to the noise..

TDC Top Dead Center. The top most position of the piston..

Chapter 1

Introduction

In modern combustion engines, there are several systems in place to optimise the combustion process. The optimisation is typically based on several sources of information and this thesis will work with one such source, namely, ion current measurements. By measuring the concentration of ions in the combustion chamber, information concerning the combustion can be extracted. This task is complicated by interference from other electric systems in the engine. If these perturbation signals could be removed, an engine with better fuel economy could be achieved.

1.1 Objective

The key objective of this thesis has been to investigate how much information can be extracted from a part of ion current measurements obscured by high frequency perturbations.

1.2 Limitation

This thesis will only seek to remove one of the characteristics of the perturbations; namely the perturbation that can be described as a damped sinusoid.

1.3 Structure of the report

The report has been divided into 7 chapters, followed by a number of appendices. The first chapter outlines the objective of the thesis as well as the limitations. It also provides a short introduction to the subject.

The rest of the report can roughly be divided into 3 parts.

Chapter 2 and 3 seeks to establish terminology and briefly explain some of the theory behind the tools used in the thesis.

Chapter 4 and 5 contain the work done in the thesis. In chapter 4, a lot of testing is done to create a foundation for the algorithm developed in the thesis. The results from these tests are then used to design the algorithm. Chapter 5 outlines the algorithm design and tests it to evaluate its performance.

Chapter 6 discuss the methodology used in this work and suggests future work that could be done. Chapter 7 summarizes the work and answers the objective.

Chapter 2

Background

This chapter establishes a setting for the thesis as well as introduce some terminology that will be used in the report. An introduction to what ion currents are as well as the complications associated with measuring them are also given.

2.1 Internal Combustion Engines

This section provides a short introduction to the combustion engines this thesis is based on. Deep knowledge is by no means necessary to understand the work but a basic understanding will provide a setting for the report. Furthermore, this section presents a short introduction to the goals of engine optimisation along with some of the difficulties associated with this task.

2.1.1 Basic engine operation

The type of engines this work has focused on are so called internal combustion engines. These will in the following passages be referred to as combustion engines or simply the engines. These type of engines are heat engines where the combustion of fuel occurs in a combustion chamber. Internal combustion engines can further be stratified into spark ignition engines, which use a spark plug to initiate the combustion, and compression-ignition engines, where the combustion is initiated spontaneously by the heat and generated from compression of the combustion chamber. This thesis has exclusively worked with the former. The combustion chambers are referred to as cylinders and an engine used in vehicles typically has more than one. This helps to deliver a smooth power output by having the combustions occur at different times in the different cylinders. The combustion cycle is described in more detail below.

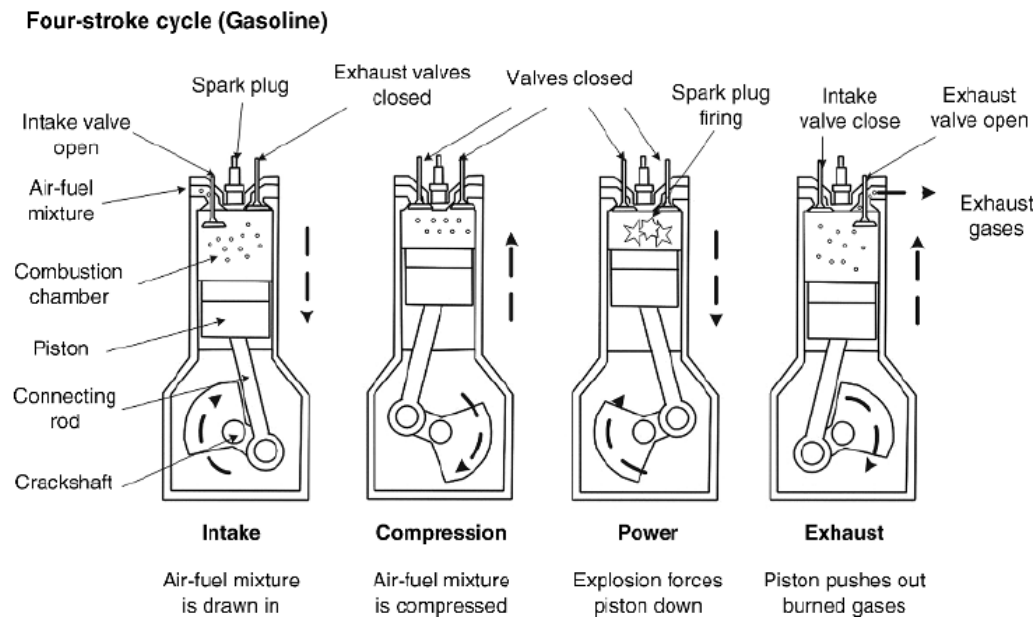


Figure 2.1: A schematic view of the strokes of a four stroke engine [1].

The figure above depicts a so called 4-stroke engine. A stroke here refers to one passage the piston makes inside the cylinder, going either up or down, so that each subfigure in figure 2.1 represents one stroke. The top position of the piston is referred to as *top dead center* (TDC) and the position furthest from TDC is referred to as *bottom dead center* (BDC). In a 4-stroke engine, the piston completes two rounds around the crank shaft for each combustion. Other designs exist, such as the 2-stroke engine, but this thesis has exclusively worked with the former. The cycle of the 4-stroke engine is typically divided into four different intervals, namely:

- **Intake** The piston start in TDC position. As the piston moves down towards BDC, the volume of the cylinder is increased, allowing the air-fuel mixture to enter the combustion chamber.
- **Compression** The piston starts in the BDC position and is on its way towards TDC. As it does so, the air-fuel mixture is compressed. Just before the piston reaches TDC, ignition begins as the spark plug discharge a spark.
- **Power** The rapid combustion of the air-fuel mixture causes a high pressure which pushes the piston downwards towards BDC. Just before the piston reaches BDC the exhaust valve opens to let out the now burnt fuel.
- **Exhaust** The piston start in BDC and as it advances towards TDC it pushes the combustion gases out through the open exhaust vent. Close to TDC, the exhaust valve closes and the intake valve opens and the cycle is repeated.

2.1.2 Optimisation of internal combustion engines

In modern combustion engines, there are numerous systems in place to control the combustion and optimize the process such that the engine operates efficiently. Traditional methods rely on the engine operating in or close to some predefined conditions which can then be stored as a table for the engine control circuit to make adjustments based on ambient conditions such as temperature and humidity, as well as engine parameters such as speed and load. The big advantage of these methods is that they are quick to find adjustments at any given time. However, as the method relies on calibration, the accuracy will depend on how well the real world conditions align with what has been calibrated, this does typically not include things such as wear and tear in the engine or varying fuel quality. There could therefore be much to gain if more live information about the engine operations as well as combustion effectiveness could be incorporated into the control circuit of the engine.

The objective when optimising the engine can briefly be summarised as delivering maximum power to the piston at the most beneficial moment. Force is generated by the expanding gas inside the combustion chamber. The objective when optimising the engine is therefore to time the combustion such that maximum pressure, occurs shortly after the piston has reached TDC after the compression phase of the 4-stroke cycle. The moment of maximum pressure is referred to as peak pressure location (PPL) and if this occurs too early, for instance in the compression phase, it will counteract the engine rotation. If it occurs too late, all the power generated from the combustion will not be absorbed by the piston. Unfortunately, the solution to this is not as easy as to release the spark at the exact moment maximum pressure is desired. The combustion process takes time to develop and the exact position of the PPL relative to the release of the ignition spark is determined by a number of factors, including air humidity, AFR, engine speed (RPM), fuel quality etc. Typically, the timing of the spark is such that the spark is released at a few degrees before TDC, BTDC.

2.2 Ion current

This section provides a background to what an ion current is, and how it can be used for optimisation of the engine. It also gives a short description of the ignition system used in this work with a focus on how the system affects the ion current measurements in terms of perturbations. In the report, the term *ion current* is used when only the current resulting from the ion concentration in the cylinder. *Ion current measurements* on the other hand is used to reference the data obtained when the ion current is measured, this includes for instance interference from the ignition system and clipping.

2.2.1 Spark generation

The spark is generated by the spark plug along with some specialised circuitry and is used to initiate the combustion process. The circuitry and spark plug is generally referred to as the ignition system of which there are several different types. The spark generated by the systems used in this work is of AC-variety (alternating current) as opposed to DC (direct current) [2].

In this work, the data comes from a capacitive ignition system exclusively. Other systems exist, such as inductive ignition systems, but as data on these systems were unavailable at the time, they are not included. This work will not delve deep into the circuitry design of these systems but will rather study the effects of them. Some background is however provided below.

The spark is generated by applying a high voltage to the spark plug gap. To generate the high voltage required, a circuit charges up a capacitor which releases its charge at a given moment, this moment is referred to as ignition timing. The released charge is then transformed so that the voltage is multiplied enough to bridge the spark plug gap and generate the ignition spark. The circuit that transform the voltage has two sides, one low voltage side and one high voltage side, these are referred to as primary and secondary side respectively. As the stored energy discharge in the spark, the current is gradually lowered until it is now longer high enough to bridge the spark plug gap.

When the voltage is too low to bridge the spark plug gap, the spark plug can essentially be seen as a capacitor. The capacitor-like spark plug in combination with the coil gives the circuit the characteristics of an LC-circuit. As energy oscillate between the coil and the capacitor in such a circuit, the voltage over the capacitor will have a sinusoidal behaviour. The frequency of this sinusoid depend entirely on the coils, capacitors and the resistance of the circuit, and as these component are the same in each ignition, the frequency should be constant. The amplitude and phase of the sinusoid however depend on how much energy is left in the circuitry, and in extension, how much was released by the spark. This is essentially random and depend on how much energy was stored in the capacitor before release, how much the resistance is over the spark plug gap, how quickly the combustion starts etc.

After a fixed time interval, the primary circuit is disengaged from the secondary as a new spark is being prepared by yet again charging the capacitor on the primary side [3]. The time interval is designed to be enough to allow the spark to discharge completely. However, some energy is generally still present on the secondary side. This energy will oscillate between the coil and the capacitor (spark plug) and its amplitude will decrease over time. The voltage over the spark plug gap takes on the appearance of a damped sinusoid in this phase of the ignition. The frequency and damping coefficient of the damped sinusoid is determined much in the same way as above by the coil, capacitor and resistance of the circuitry. However, as the primary circuit is disengaged, one can not expect these parameters to be the same as above. The amplitude and phase yet again depend on how

much energy is still stored in the circuit at the instance the primary circuit is disengaged, which, following the same reasoning as above, is random.

2.2.2 The combustion process

After the discharge of the spark, the air-fuel mix is ignited and the combustion starts. The combustion process is essentially made up of chemical reactions between the fuel and the air mixture. Different chemical reactions will dominate the combustion process at different times of the process and the combustion process is generally divided into chemical and thermal ionisation. The reactions generate heat as well as a lot of ions, i.e., particles with a net charge different from zero. As ions will form and recombine over the course of a combustion cycle, the ion concentration will vary during a cycle [4].

2.2.3 Ion current measurements

The fact that the ion concentration vary over the course of the combustion process would allow one to get information on the combustion by measuring this concentration. This means that by measuring the ion concentration in the combustion chamber, the current stage of the combustion can be estimated [4].

By applying a voltage to the spark plug and measuring the resulting current, the conductivity of the gas mixture surrounding the spark plug can be calculated. The current over the spark plug gap is referred to as the ion current. The conductivity of the gas is in turn determined by the ion concentration in the gas such that a higher ion concentration results in a larger current and vice versa. The ion current has been shown to provide information on a number of topics including misfire and knock detection, optimisation of ignition timing and estimating fuel quality [4], [5].

Measurements of the current over the spark plug are made at a fixed rate. This rate is typically determined in crank degrees so that a measurement are made at every few crank degrees. For instance, a typical set-up performs a sample every 0.2 crank degrees between the intervals -54° and 125° . The sample rate will therefore depend on how quickly the crank turns. As the measurements are performed based on crank degrees rather than at a fixed sampling rate, a perfect synchronisation is obtained between measurements and crank angle. However, this also means that the sample rate will vary with engine speed, RPM (Rounds Per Minute). For instance, with a sampling interval of 0.2 crank degrees and an engine speed of 5000 RPM, the sample frequency (f_s) becomes

$$\begin{aligned} f_s &= \frac{(\text{Number of samples per round}) \times (\text{Rounds per minute})}{\text{Seconds per minute}} = & (2.1) \\ &= \frac{(360/0.2) \times 5000}{60} = 150\,000 \text{ Hz} \end{aligned}$$

However, with an engine speed of 1000 RPM, the sample frequency instead becomes $30\,000\text{Hz}$. This is important to keep in mind as this translates to 5 times as many data points for the 5000 RPM case compared to the 1000 RPM case. This is because the frequency of the interference is independent of engine speed. As a lot of data analysis methods perform better with more data, one can expect better performance the higher the engine speed is.

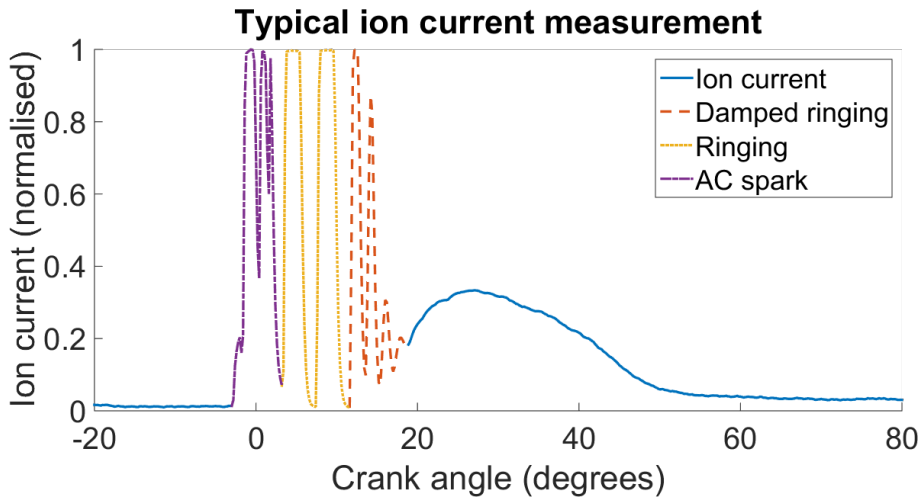


Figure 2.2: An example of a typical ion current measurement, the data has been normalised to the interval $[0, 1]$. The different parts of the perturbation has been highlighted in different colours. The AC spark can be seen in the interval -2.8 to 3.2 crank angle degrees. From 3.2 to 11.6 is the undamped ringing. Between 11.6 and 18 crank angle degrees the damped ringing can be seen. Clipping can be seen to be present in the signal as the signal is limited to the interval $\sim [0, 1]$, this mainly affect the interval where the ignition system is active.

A consequence of measuring the ion current over the spark plug, is that a lot of unwanted signals are recorded. These signals include the ignition spark as well as the following sinusoids. Due to their large amplitude compared to the ion current, they are a large problem as they cover parts of the ion current that has good informational value.

To further complicate things, the range at which the measurement equipment operate is selected so that it is well fitted to the range the ion current is expected to fall within. The perturbations from the ignition system push the current outside of this range, causing so called clipping. The clipping limits the signal to the measurement equipments permitted range by replacing values above this range with the upper value of the range. The opposite happens when the signal goes below the range. Data points affected by clipping can only be said to be larger than (or smaller than) the threshold value. They are therefore generally considered to be lost entirely as it is impossible to say anything about *how much* they exceed the permitted range. Uniformly sampled data is data collected with a fixed interval between measurement points. Data with missing samples have varying time intervals

between samples and is therefore referred to as non-uniformly sampled.

A typical example of an ion current measurement can be seen in figure 2.2. The figure illustrate the perturbations created by the ignition system as well as the clipping. The problem with the perturbations from the ignition system coupled with the clipping is what this work aims to find a solution to.

Chapter 3

Theory

This chapter establishes theory for the tools used in the rest of the thesis. For readers with knowledge of the tools, the chapter can be skipped completely.

3.1 Sinusoid

A general form of a signal consisting of N damped sinusoids can be seen below.

$$y(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{f}, t) = \sum_{i=1}^N \alpha_i e^{i2\pi f_i t - \beta_i t} \quad (3.1)$$

here $\alpha_i \in \mathbb{C}$ is the complex amplitude, $f_i \in (-0.5, 0.5]$ the frequency and $\beta_i \in \mathbb{R}$ is the dampening factor of the signal. Different values for these parameters will give the signal different characteristics.

3.2 Least Squares

Least squares is method in regression analysis which approximates the solution to an overdetermined system, i.e., when the number of equations are larger than the number of unknowns. The name "Least Squares" comes from how the solution given by the method minimizes the sum over the squared residuals.

3.2.1 Ordinary Least Squares

By defining a linear model as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{e} \quad (3.2)$$

where \mathbf{y} are measurements of the process, \mathbf{X} the input variable with dimensions $[M \times N]$, where M is the model order, $\boldsymbol{\theta}$ is the model parameters, \mathbf{e} is the residuals, and $*$ denotes the hermitian transpose formed by taking the transpose and conjugate of the matrix.

$$\mathbf{y} = [y_1, \dots, y_N]^T \quad (3.3)$$

$$\mathbf{X} = \begin{bmatrix} x_{11}, x_{12}, \dots, x_{1N} \\ x_{21}, x_{22}, \dots, x_{2N} \\ \vdots, \vdots, \ddots, \vdots \\ x_{M1}, x_{M2}, \dots, x_{MN} \end{bmatrix}^* \quad (3.4)$$

$$\mathbf{e} = [e_1, \dots, e_N]^T \quad (3.5)$$

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_M]^T \quad (3.6)$$

For the derivation of the estimator, the reader is referred to *An Introduction to Time Series Modeling* by Jakobsson [6]. The resulting estimator is shown below.

$$\boldsymbol{\theta} = (\mathbf{X}^* \mathbf{X})^{-1} \mathbf{X}^* \mathbf{y} \quad (3.7)$$

3.2.2 Amplitude and phase estimation of damped sinusoid with ordinary least squares

Given a set of data, \mathbf{y} that consist of M damped sinusoids, with known frequencies and damping factors ($\mathbf{f} = [f_1, f_2, \dots, f_M]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_M]$), the amplitude and phase can be estimated using linear least squares. This is done by forming a linear model the same way as in equation 3.2 above, with the following matrices

$$\mathbf{y} = [y_1, \dots, y_N]^T \quad (3.8)$$

$$\mathbf{X} = \begin{bmatrix} \omega_1^{t_0} & \omega_1^{t_1} & \omega_1^{t_2} & \omega_1^{t_3} & \dots & \omega_1^{t_{N-1}} \\ \omega_2^{t_0} & \omega_2^{t_1} & \omega_2^{t_2} & \omega_2^{t_3} & \dots & \omega_2^{t_{N-1}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_M^{t_0} & \omega_M^{t_1} & \omega_M^{t_2} & \omega_M^{t_3} & \dots & \omega_M^{t_{N-1}} \end{bmatrix}^*, \quad (3.9)$$

$$\mathbf{e} = [e_1, \dots, e_N]^T \quad (3.10)$$

where $\omega_n = e^{-2\pi i f_n - \beta_n}$. The complex amplitudes are then found as in equation 3.7.

3.3 HTLS

HTLS (Hankel Total Least Squares) is a method for estimating frequency and damping coefficient pairs, introduced in 1994 by Vanhuffel, Chen, Decanniere, *et al.* as a way to improve upon HSVD (Hankel Singular Value Decomposition). The method uses the fact that a model described by equation 3.1 can be expressed on state-space form according to

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n \quad (3.11)$$

$$\mathbf{y}_n = \mathbf{C}\mathbf{x}_n. \quad (3.12)$$

This can in turn be rewritten as $\mathbf{y}_n = \mathbf{C}\mathbf{A}^n\mathbf{x}_0 = \sum_{k=1}^K c_k z_k^n$, where $\mathbf{A} = \text{diag}(z_1, \dots, z_K)$ and $\mathbf{C} = [c_1, \dots, c_K]$. c_k are the complex amplitudes and $z_k = e^{(-d_k + i2\pi f_k)t_n}$ contains frequency and damping. From the parameters \mathbf{A} and \mathbf{C} , an estimate of the parameters $\boldsymbol{\alpha}, \mathbf{d}, \mathbf{f}$ can be calculated. For a deeper understanding of the theory, the reader is referred to the original article [7].

The HTLS algorithm is described in four steps in the article, these are summarised below.

Given. N uniformly sampled data points, $y_n, n = 0, \dots, N-1$ and a model order K .

Step 1. Form the Hankel data matrix, by organising the measurement data as below.

$$\mathbf{H}_{L \times M} = \begin{bmatrix} y_0 & y_1 & y_2 & \dots & y_{M-1} \\ y_1 & y_2 & & & \\ y_2 & & & & \\ \vdots & \vdots & & \ddots & \vdots \\ y_{L-1} & & \dots & & y_{N-1} \end{bmatrix}. \quad (3.13)$$

Compute the singular value decomposition, SVD, of the Hankel data matrix.

$$\mathbf{H}_{L \times M} = \mathbf{U}_{L \times L} \boldsymbol{\Sigma}_{L \times M} \mathbf{V}_{M \times M}^*, \quad (3.14)$$

where the superscript $*$ denoted the hermitian transpose.

Step 2. Truncate \mathbf{H} to a matrix \mathbf{H}_K of rank K :

$$\mathbf{H}_K = \mathbf{U}_K \boldsymbol{\Sigma}_K \mathbf{V}_K^*, \quad (3.15)$$

where $\mathbf{U}_K, \boldsymbol{\Sigma}_K$, and \mathbf{V}_K are the first K columns of $\mathbf{U}, \boldsymbol{\Sigma}$ and \mathbf{V} , respectively.

Step 3. Compute the solution \mathbf{Q} of the set

$$\mathbf{U}_K \mathbf{Q} \approx \bar{\mathbf{U}}_K, \quad (3.16)$$

where \mathbf{U}_K and $\bar{\mathbf{U}}_K$ is formed by omitting the last and first row of \mathbf{U}_K , respectively.

Now, if equation (3.16) is solved using least squares, the algorithm is called HSVD. By solving using total least squares, the HTLS algorithm is achieved.

The frequencies and damping coefficients can now be found as the eigenvalues, λ_k , of \mathbf{Q} such that

$$\lambda_k = z_k = \exp[(-d_k + i2\pi f_k)\Delta t] \quad (3.17)$$

Step 4. Calculate \mathbf{C} , the complex amplitudes, as the least squares solution to

$$\mathbf{A}_{N \times K} \mathbf{C} \approx \mathbf{Y}_{N \times 1}. \quad (3.18)$$

The difference between the methods is that HTLS employs total least squares in step 3, whereas HSVD employs ordinary least squares. This difference has been shown to decrease variance of the estimated parameters as well as to decrease the bias [7]. By using TLS instead of LS, the efficiency of the method is however somewhat decreased, the article found the difference in efficiency to be a factor of 1.01.

In the abstract of the article the method is referred to as a black-box method, meaning that no prior knowledge of parameter values is taken into account in the estimation. This could be a drawback over other methods which possibly take advantage of prior knowledge. The method is however computationally very efficient, which could outweigh these drawbacks.

3.4 Digital filters

Digital signal processing (DSP), as opposed to analogue signal processing, is signal processing performed with the use of computers, whereas analogue signal processing typically use electrical circuitry for analysis. The advantage of DSP lies in the fact that the results of a certain algorithm are always predictable. Electrical components typically has a tolerance which will affect the performance as well as being subject to quality degradation over time [8].

3.4.1 Finite impulse response (FIR) filters

Finite impulse response filters has a number of advantages, such as:

- highly regular structure, which makes for efficient implementations.
- Stability criterion is always fulfilled.
- the filter quiets down if now signal is present.
- perfect linear phase response is easily achievable.

While simple, the above advantages of the FIR filter makes it a popular filter design choice. The filter is constructed as a weighted linear combination of the current and M past samples [8]. This can be represented as a block diagram, see below.

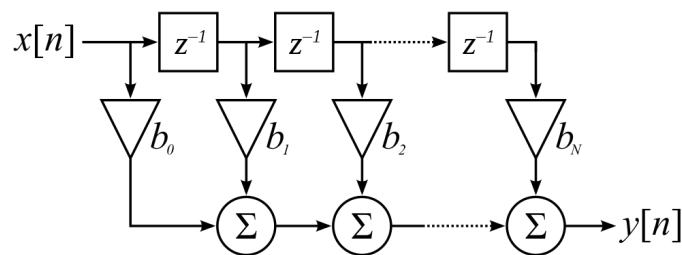


Figure 3.1: Block diagram describing a FIR-filter. The output, y_n , of the filter is constructed of the weighted sum over M past samples and a number of different filters can be designed by choosing weights in certain ways.

The flowchart in 3.1 uses z-transform notation, where z^{-1} represents the unit time lag operator. The operator is generalised as $x(n)z^{-l} = x(n-l)$, $l \in \mathbb{N}_0$, and $\mathbb{N}_0 = \{0, 1, 2, \dots\}$. Using mathematical notation, the output can be written as a function of the input in the following way:

$$y(n) = \sum_{l=0}^N b_l x(n) z^{-l} = \sum_{l=0}^N b_l x(n-l) \quad (3.19)$$

The filter is entirely defined by the set of coefficients and the filter can be given a large variety of different characteristics by customizing these coefficients. The design of these parameters has been made simple with different filter design toolboxes, such as the Signal Processing Toolbox in MATLAB.

These filter design tools typically takes a cutoff frequency, filter order and some specification of desired filter properties. For instance, using the MATLAB function `FIR1` to design a low pass filter can be done in the following way:

```
b = fir1(n, Wn, 'Low')
```

where `n` is the filter order, `Wn` the cutoff frequency and the third argument, `'Low'` specifies that the filter should be a low pass filter. An ideal low pass filter will have a frequency response equal to 1 for signal components with a frequency lower than the cut-off frequency,

and 0 for signal components with frequencies that are higher. In practice, the frequency response will never be perfect and it is therefore necessary to choose ω_n so that some headroom is provided to the signal components of interest. Another consideration is the order of the filter, denoted N in figure 3.1 above. A longer filter (higher filter order) will result in a frequency response closer to an ideal filter. However, a longer filter also means more computation and in some cases the length of the signal limits the possible length of the filter.

The filter length should be kept shorter than half the length of the signal, if possible much shorter. The reason stems from an effect called transient response and which is present in every filter. This response is the filter's response to sudden changes in the input signal such as a frequency component being added abruptly at some time point, or for that matter, stopping abruptly at some time point. The reason the transient response exists comes from the way the filter is applied to the input signal which is done through convolution. For more information on the subject the reader is referred to literature on the subject, for instance *Digital Filters: Basics and Design* by Schlichthärle [8, section 3.1]. The transient response will corrupt the output of the filter for a set number of samples, until the filter has reached what is called "steady state". The duration of the transient response depends on the length of the filter such that the transient response has a duration of $N - 1$ samples, where N is the filter length [9, chapter 5].

3.5 Monte Carlo simulation

Monte Carlo methods are used in this thesis to evaluate and compare different methods. As the input domain is multi-dimensional and very large, it is impossible to evaluate the methods for every possible point of this domain. By instead drawing a sufficiently large set from the input space, and evaluate different methods for a given input, the methods efficiency will be estimated in a reliable way.

Monte Carlo methods is a category of methods which rely on random sampling to obtain results. One important set of problems that can be solved using Monte Carlo methods is to find the expectation of some function $\phi(x)$

$$\mathbb{E}_f[\phi(x)] = \int_{\mathbb{D}} f(x)\phi(x)dx, \quad (3.20)$$

where \mathbb{E}_f is the expectation value taken with respect to the distribution f , x is a random variable, defined on the space \mathbb{D} with some probability density $f(x)$, and $\phi(x)$ is the so called objective function.

Generally, Monte Carlo methods follow the same basic pattern:

1. Define a range of possible input values
2. Draw values from this range, according to some possibility distribution

3. Calculate the function value for the drawn input
4. Collect the results

To for instance estimate the expected value, the average over the results is calculated. A confidence interval is also typically calculated for the expected value. With all Monte Carlo methods, the quality estimated statistic will benefit from more samples, i.e., the variance of the estimate will decrease with a higher sample count. The simplest Monte Carlo method is simply an average over the objective functions output, when this is given the randomly drawn input values. The estimator is formulated below, with N denoting the total number of samples drawn from the input domain.

$$\tau = \mathbb{E}_f[\phi(x)] \approx \frac{1}{N} \sum_{i=1}^N \phi(x_i), \quad x \sim f(x) \quad (3.21)$$

3.6 RMS error

The RMS (Root Mean Square) error is a method that can be used to estimate the error between two data sets. The RMSE between two vectors was calculated using the following formula

$$rms(x) = \frac{1}{\sqrt{N}} \|\mathbf{x}\|_2 = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i \cdot \bar{x}_i}, \quad (3.22)$$

where N is the number of data points in the measurements set y , \hat{y} is an estimate of y and $\|\mathbf{x}\|_2$ is the 2-norm of the vector \mathbf{x} , $\bar{\mathbf{x}}$ denotes the conjugate of \mathbf{x} . The RMS measure can be interpreted as the square root of the average squared error and while similar to the average error, they are not equal. As the measure squares the error, the method is sensitive to outliers. As least squares, which minimizes the squared error, is a central part of several methods in this thesis, the RMS error is a quite suitable measure.

Chapter 4

Method

This chapter performs a lot of initial analysis on the ion current measurements, which result in a model that can be used to simulate measurements. Testing of some methods are also conducted to see how they perform on simulated data, the conclusions drawn from these tests are then used in chapter 5 to design an algorithm.

4.1 Simulation of ion current

The goal of this section is to produce a model that can be used when simulating the ion current. Emphasis will be on reconstructing the perturbation such that it resembles the real perturbation as closely as possible.

4.1.1 Analysis of ion current measurements

To be able to construct a model suitable for simulation, the properties of measured ion currents was characterised using both literature and tools for signal analysis. As the focus of this thesis is to remove the damped sinusoid, more time was spent on analysis of this part than on the underlying ion current. The signal was divided into two components: ion current and high frequency perturbation associated with the spark and spark generation circuitry. The perturbation can further on be stratified into AC spark, undamped ringing and damped ringing.

Ion current

The signal resulting from the ions forming during the combustion process. As the focus of this work has been to remove the damped ringing signal from the measured data, the model describing the ion current was kept simple. Several sources describe the ion current

as being approximately shaped as the sum of two gauss functions [5], [10], [11], this has therefore been used in simulations.

High frequency perturbation

This section focuses on the high frequency perturbations that occur as the combustion process is initiated. An isolated measurements of an ignition system in operation can be seen below.

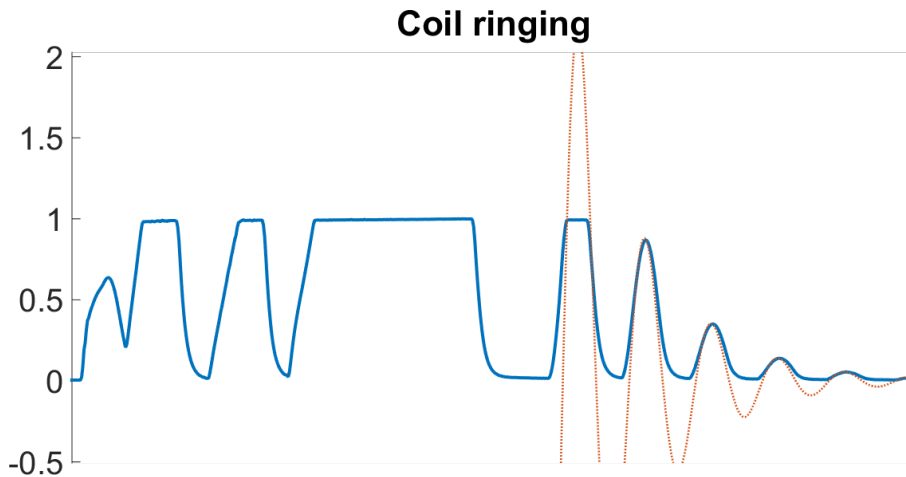


Figure 4.1: In the figure above, the data set recorded when running only the ignition system can be seen as the blue, solid line. The red, dotted line is a damped sinusoid added on top of the data to illustrate that this part of the ringing is indeed modelled well by a damped sinusoid. The measured signal fits the peaks of the damped sinusoid well, but is less responsive.

Stratification of the interference As was discussed in section 2.2.1, the spark and following sinusoid is present in the signal for during an interval of fixed length before the damped sinusoid starts. The first step of this analysis will be to try to verify this as well as to find out how long this interval is. In the data files provided by SEM AB, a number of engine settings and measurements are available, including RPM, ion current measurements and ignition timing. With this data, a script was written that loops over the recorded ion current. By then measuring the distance between the ignition timing and the first peak of the damped ringing, an estimate of the distribution over where the circuitry changes characteristics can be made.

In theory, the first peak of the damped ringing will be located in an interval as long as the period, $T = 1/f$, of the sinusoid. The reason for this spread comes from the fact that the primary circuit might disengage the secondary circuit at different phases of the damped sinusoid. If the phase at which the break point occur is approximately uniformly

distributed, the distribution of the first peaks location will be uniformly distributed, $x \in \mathbb{U}(b, b + T)$, with the following density

$$f(x|b, T) = \begin{cases} \frac{1}{T}, & x \in [b, b + T] \\ 0, & \text{otherwise.} \end{cases} \quad (4.1)$$

Here, b is the point at which the primary circuit disengage, measured as time after the ignition timing, and T is the period of the sinusoid, x is an observation of the peak location. Given a set of N observations of the peak location, $\mathbb{X} = \{x_1, x_2, \dots, x_N\}$, an unbiased estimator of b is $\min(\mathbb{X})$.

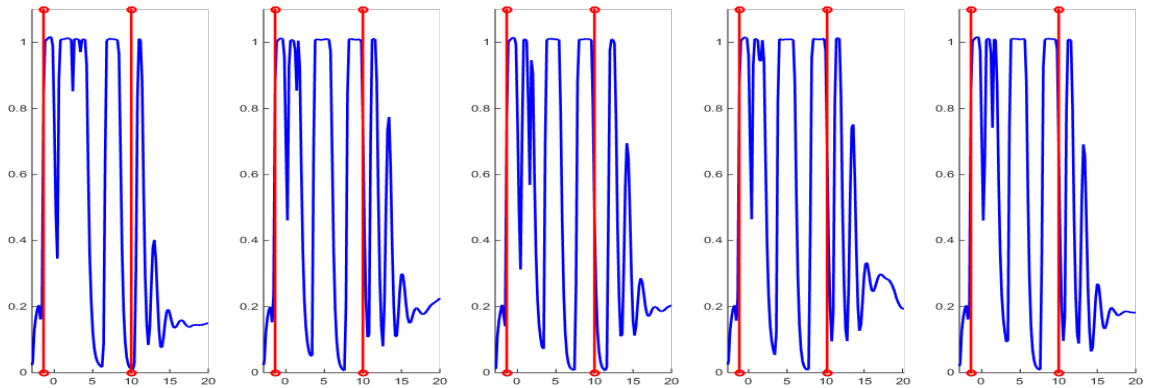


Figure 4.2: A few samples of the circuitry timings found with the above method. The blue, solid line shows the ion current measurements, and the red vertical lines show ignition timing and the estimated break point. The first vertical line is the ignition timing and the second is the estimated break point. As can be seen from the samples above, the estimated breakpoint fits the data well.

AC spark The spark produced with the ignition systems used in this work is of the so called AC variety, as opposed to a DC spark. AC stands for alternating current and DC for direct current. With the circuitry used, the spark is stochastic in such a way that one cannot modelled its behaviour to be able to remove it. Furthermore, as the combustion process starts after the spark has discharged, there is no ion current in this part of the signal [13].

As the spark bridges the gap of the spark plug, energy is discharged, lowering the total charge in the ignition system. Energy is discharged until the voltage over the spark plug is no longer enough to create a spark and the ignition circuit enters the next stage.

Undamped ringing Here the perturbation has a sinusoidal behaviour, marked in yellow in figure 4.3, however, as large parts of the signal is affected by clipping it is difficult to characterize this part of the signal reliably. Also, as the focus of this thesis has been the

removal of the damped ringing rather than this part of the signal, a limited amount of work was put into analysis of this signal component.

Damped ringing The circuitry has now disengaged the primary circuitry to begin charging in preparation of the next ignition. This changes the characteristics of the perturbation yet again with a damped sinusoid as a result. This can be seen in red in figure ??.

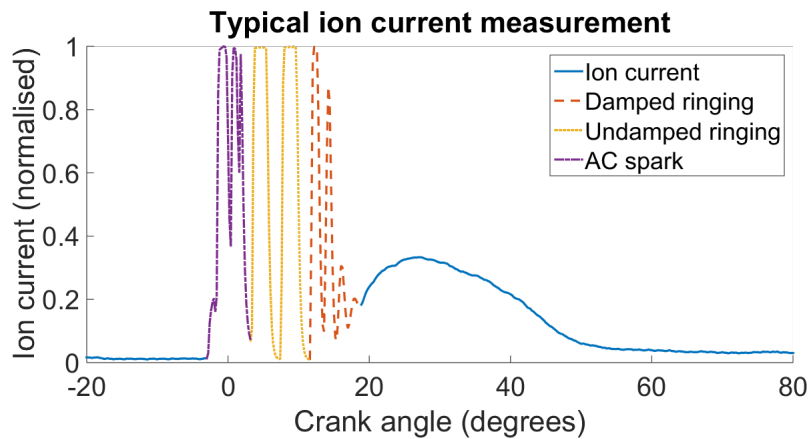


Figure 4.3: A typical ion current measurement. The different parts of the ion current and its different perturbations are highlighted.

From figure 4.1, the conclusion can be drawn that a damped sinusoid fits the data well. The clipping adds some perturbation to the lower half of the sinusoid, but apart from that the model should work fine. Further conclusions that can be drawn about the damped sinusoid is that, as the electric components in the ignition system determine the frequency and damping coefficient, these parameters are constants. As an engine experiences very high temperatures, one should expect thermal drift to affect the electric components. This will likely translate to a drift in frequency and damping coefficient as well, but for a short time interval, this drift is assumed to be negligible.

Clipping

Clipping occurs due to limitations of the measurement equipment. The equipment used to record the ion current consists of an amplification circuit as well as an ammeter (Ampere meter). These two components produce clipping at roughly the zero-level and some upper level which is determined by the specifications of the ammeter. The upper value will therefore vary between different engines. The clipping also adds some non-linear behaviour in the lower half of the signals. This can be seen in figure 4.1 where the measured data deviate from the fitted damped sinusoid in the lower half of the oscillations. The precise reason for this behaviour is unknown at this time but the exponential decay after a peak

is similar to how a charged capacitor would discharge. Seeing as the ignition circuitry has several capacitors this seems like a reasonable explanation.

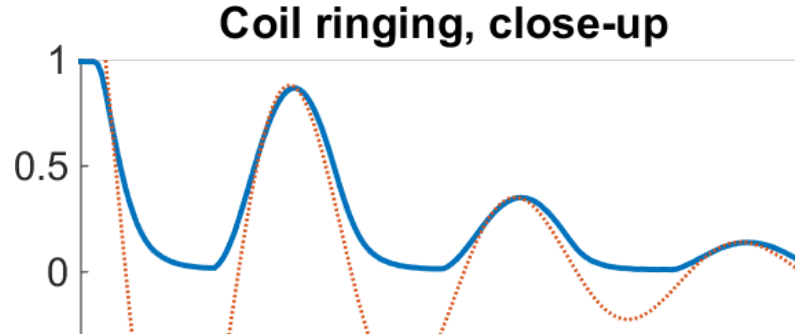


Figure 4.4: A close up of figure 4.1 above to illustrate how the lower clipping level affects the measured signal. The solid blue line represents the real measurements and the dotted red line represents a damped sinusoid. The exponential decay following the peaks of the blue, solid data set is reminiscent of how a charged capacitor would discharge its energy over time.

The level at which clipping occurs can be estimated by looking at the data, and the level can then be adjusted so that the data points above the level can be detected as clipped. This method heavily depend on which subset of the data is studied and large variance is therefore to be expected. A more robust method is described below.

Take the entire data set, sort it. Due to the clipping, a plateau is now found at the start of the data (data sorted descending). Two linear functions are then fitted to the data, one using the data points of the plateau, and on using the data points following the plateau. The clipping level can then be estimated as the intersection point of the two linear functions through:

$$\begin{aligned}
 y_1 &= k_1x + m_1 \\
 y_2 &= k_2x + m_2 \\
 x_{intersect} &= \frac{m_2 - m_1}{k_1 - k_2}
 \end{aligned}
 \tag{4.2}$$

where k and m are parameters defining the linear function. The intervals used to define the two linear functions are chosen somewhat arbitrarily. Generally a lower, more conservative estimate of the threshold is better, as it is important to not include any clipped data points in the analysis.

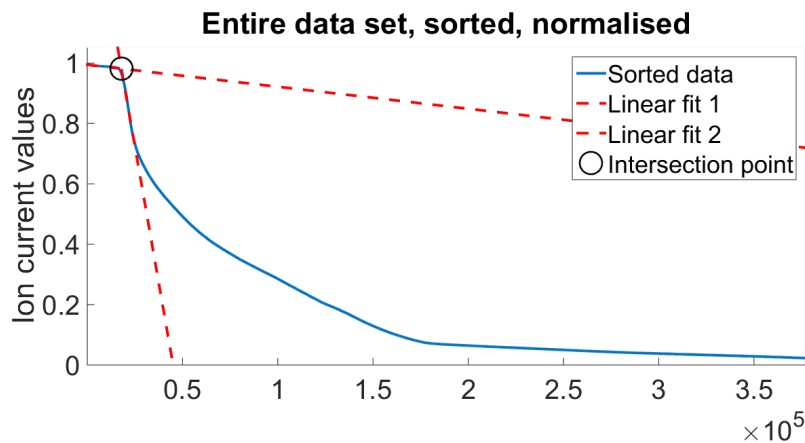


Figure 4.5: An illustration of the method used for finding the upper clipping level. Two linear functions (red, dashed) are fitted to the sorted data (blue, solid). The clipping level can then be estimated as the intersection point of the two lines (circle, black).

4.1.2 Implementation of ion current model

Working in a simulated environment has big advantages when it comes to signal separation:

- the reconstructed signal can be compared directly to the underlying source.
- Another advantage is that there is no longer a limit on the amount of data available as new data can be simulated
- parameters can be adjusted freely to examine the effectivity of the algorithm under different conditions.

Working on simulated data has disadvantages

- If the model is incorrect, the algorithm will perform poorly on real data.
- Simulation is done with very "clean" data, i.e. a distinct damped sinus and 2 Gaussian shaped signals

Using simulated data, an algorithm can be tested in more extensively to find weaknesses and to be able to optimize it for when it is applied to real data. As the data is assumed to follow the characteristics described above, this was used for simulation.

Simplifications

With the above analysis, a number of simplifications were made to reduce complexity of the model. These can be stratified into the same categories as the analysis has been divided into, namely *Ion current*, *High frequency perturbations* and *clipping*. What the simplifications are, and the motivation for them are described below.

Clipping Clipping was added to the signal by simply replacing values above (or below) a threshold value by that threshold value, so that the signal is limited to the interval [Lower threshold, Upper threshold]. This simplification removes some of the variation that is present in the clipping levels of real measurements. The main issue with adding clipping this way is that the exponential decay leading up to the lower clipping is not present in the simulation.

Ion current The ion current has been modelled as the sum of two Gaussian shaped signals, with different parameters. This simplification is expected to affect the resulting algorithm very little, as the algorithm will focus on removing the high frequency noise. However, parameters has been set so that the model ion current has similar properties to that of the real ion current.

High frequency perturbations As the focus of this thesis has been the removal of the damped sinusoid, the simulated high frequency interference was simulated as a single damped sinusoid, starting at the ignition timing and extending all the way to where the damped sinusoid would end in real measurements. Furthermore, the frequency and damping was kept at fixed values, so that any thermal drift in these values was ignored.

Model formulation

The following model was implemented using MATLAB

$$g_1(A, \mu, \sigma, x) = Ae^{-(x-\mu)^2/2\sigma^2} \quad (4.3)$$

$$g_2(\alpha, \beta, f, \psi, x_0, x) = \alpha e^{-\beta(x-x_0)} e^{2\pi f i(x-x_0) + \psi i} \theta(x - x_0) \quad (4.4)$$

$$f(A_1, \mu_1, \sigma_1, A_2, \mu_2, \sigma_2, \alpha, \beta, f, \psi, x_0, x) = \quad (4.5)$$

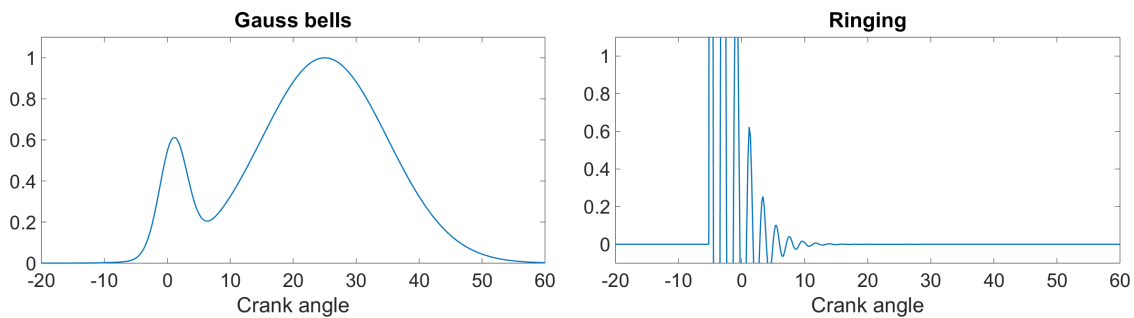
$$g_1(A_1, \mu_1, \sigma_1, x) + g_1(A_2, \mu_2, \sigma_2, x) + g_2(\alpha, \beta, f, \psi, x_0, x)$$

Equation (4.3) define a Gaussian shaped signals with centre at μ , max value A and σ describes the width. Equation (4.4) define a damped sinusoid with amplitude α , damping coefficient β , frequency f , phase ψ and a start point x_0 . White gaussian noise is also added to the signal to simulate for instance electric interference. The clipping is added with the following MATLAB script

```
y(y>threshold_high) = threshold_high;
y(y<threshold_low) = threshold_low;
```

where y is the data simulated using equations (4.5) above and `threshold_high` and `threshold_low` are the threshold values.

The full script used for simulation can be found in appendix A.



(a) The simulated ion current, here represented as the sum of two Gauss functions. The data is normalised.

(b) The simulated ringing, represented as a single damped sinusoid. The plot is windowed so as to only show the range $[-0.1, 1.1]$, but the ringing extends both above and below this range.

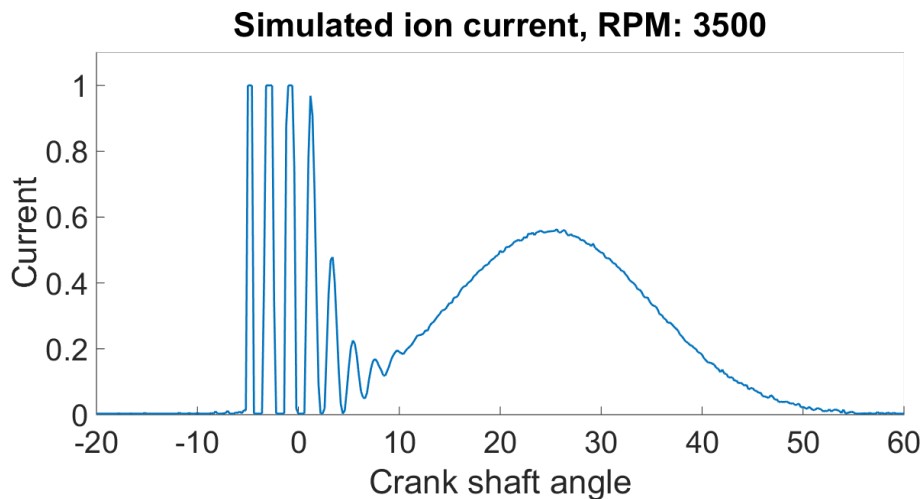


Figure 4.7: The resulting simulated ion current measurement.

This model was then used to design an algorithm for removing the interference signals from the ignition system.

4.2 Low pass filtering with clipped data

An intuitive approach to the problem of removing the high frequency components of the signal would be to apply a low pass filter to extract the ion current. This section explore the possibilities of such a strategy by applying a low pass filter to simulated clipped signals and comparing the output to the source for a few different scenarios.

4.2.1 Methodology

The ion current measurements consist of a low frequency component, the ion current, and a few high frequency components, the perturbations, well separated in frequency. As the low frequency components are separated from the high frequency components, a low pass filter should be able to separate the ion current from the perturbations. However, as the signal is subject to clipping which essentially replace data points with constants (zero frequency), it is uncertain how well this approach would work. The clipping is a result from limitations in the measurement equipment, which is discussed in more detail in section 2.2. In addition to the clipping, the effect of the transient response of the filter is also unknown. The aim of this section will therefore be to try to quantify the effects of the clipping and the transient response. These effects are studied by simulating a signal consisting of the sum of one low frequency component and one high frequency component. Simulated data was used as it is then possible to measure the exact effect of the clipping. The simulation parameters were chosen so the signal composition is similar to the ion current measurements. The parameters can be seen in table 4.1 below.

α_1	β_1	f_1	f_1/f_s	ψ_1	α_2	β_2	f_2	f_2/f_s	ψ_2	f_s	<i>Offset</i>
1	0	1 000	0.0667	0	7	0	10 000	0.667	0	150 000	α_1

Table 4.1: Parameters used when simulating a signal consisting of one low frequency component and one high frequency component. The two signal components and the sample frequency, f_s , are similar to what is typically found in the data. The offset is a constant added to the signal to simulate the property that the low frequency component (mimicking the ion current) is always positive.

With these parameters, a sequence of 900 data points was generated in MATLAB, 900 data points is a typical sequence length for the measurements. To mimic the behaviour of the measurement data, the simulation was modified in two separate ways: by adding clipping and by limiting the high frequency component to a shorter time window using heaviside functions. The added clipping will give an indication of the effect clipping has on the filtered data and the heaviside functions will give an indication of how the transient response affects the filter output. These modifications were combined in four different ways. The scenarios used for evaluation were therefore

1. No clipping, no heaviside
2. No clipping, heaviside
3. Clipping, no heaviside
4. Clipping, heaviside

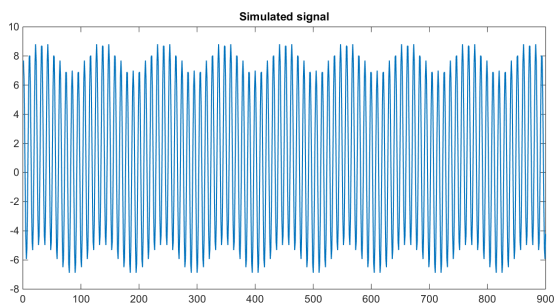
The low pass filter was configured with a cut off frequency of $5000/f_s$ which should provide sufficient headroom between the two frequency components. To study the effect the filter order has on the output, a few different filter orders were used, these were:

$$n = \{ 10, 12, 14, 16, 18, 20, 22, 26, 30, 34, 38, 42, 50, 58, 66, 74 \}.$$

The tested filter orders were chosen such that all realistic alternatives were tested. A lower filter order than 10 is unlikely to be able to produce good outputs. Filter orders larger than 74 starts to be so much longer than the high frequency component that the transient response is all that can be seen in the output.

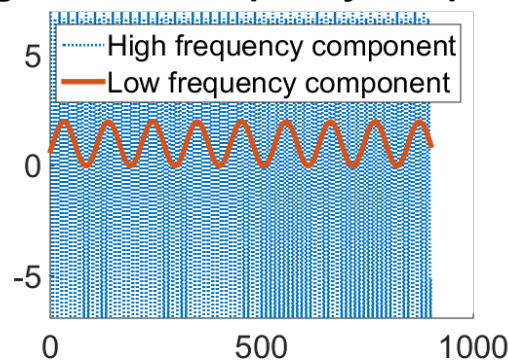
1. No clipping and no heaviside

This scenario was used to create a baseline for what could be considered as a best case scenario for the performance of the low pass filter. No clipping is added to the signal and the high frequency component is present during the entire data set, removing the issue of the transient response. The simulated signal along with the two signal components can be seen below.



(a) A sample of the simulated signals in scenario 1. No clipping is present in the signal and the high frequency component is present the entire duration of the signal.

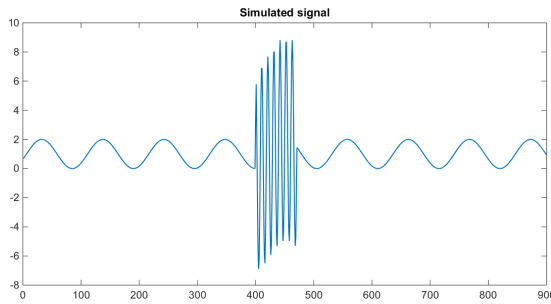
High and low frequency component



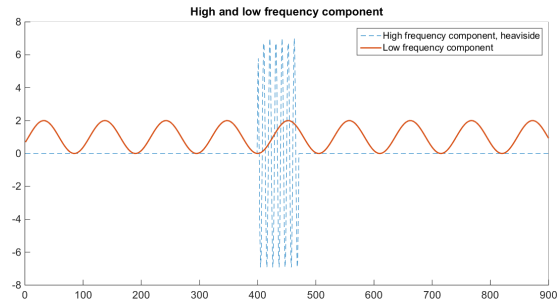
(b) The two signal components used in simulation, one high frequency (representing noise) and one low frequency (representing the ion current).

2. No clipping, with heaviside

This scenario strives to examine the effects the transient response has on the signal separation. The heaviside functions are set up so that the high frequency component is active for 70 samples. The signal starts randomly somewhere between 400 and 430, consequently ending somewhere between 470 and 500. The length of the interval is similar to the duration of the ringing in measurement data. Simulating with these specifications result in the following signals.



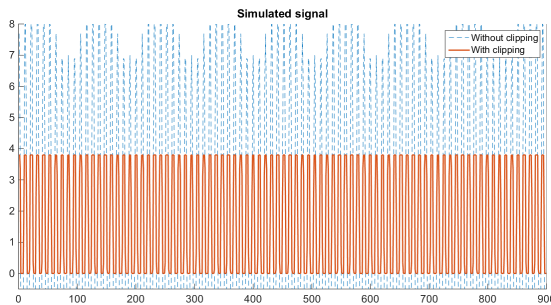
(a) A sample signal from scenario 2. As the actual ringing found in ion current measurements are short, this scenario tests what effect this will have on the filter output by limiting the high frequency component to a certain interval.



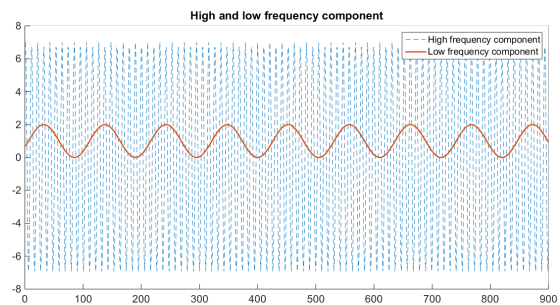
(b) The two signal components used to generate the signal. The high frequency component is limited to a certain interval, here $t = 400$ to $t = 470$ using heaviside functions.

3. Clipping, no heaviside

Here, clipping is added to the signal such that values above an upper threshold are replaced with the upper threshold value, and values below a lower threshold are replaced with the lower threshold value. The simulation is identical to scenario 1 with the addition of limiting the signal to the range $[0.0, 3.8]$, which is a typical range for ion current measurements.



(a) A sample signal from scenario 3 (red, solid) and the underlying unclipped signal (blue, dashed). Here, clipping is added to the signal such that values below 0 are replaced with 0, and values above 3.8 are replaced with 3.8.

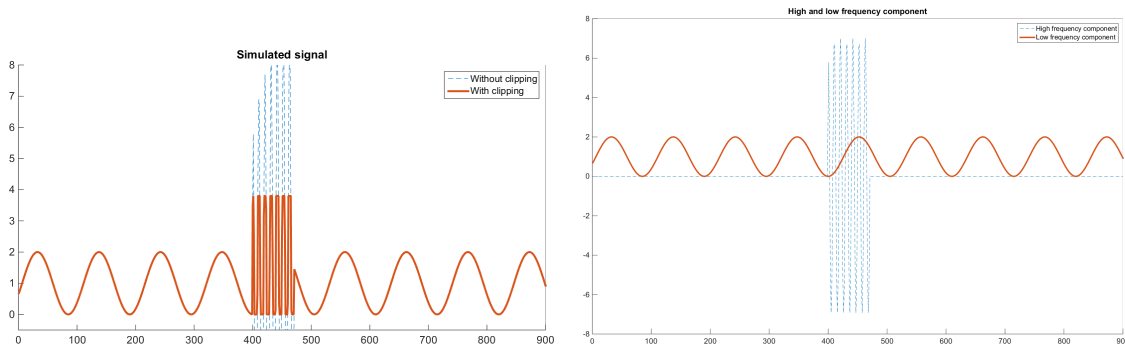


(b) The two signal components used in simulation, one high frequency (representing noise) and one low frequency (representing the ion current). The components are identical to those used in scenario 1.

4. Clipping, with heaviside

Scenario 4 is the simulation which resembles the true ion current the closest as it has both clipping and a limiting heaviside function. Some differences still remain, such as having the low frequency component oscillate for the entire duration of the data set as well as having no damping factor in the high frequency component. The high frequency component of

these simulations is also of a constant frequency, whereas the true ringing perturbation shifts in character, as is described in section 2.2. A sample of the simulated datasets can be seen below.



(a) A sample signal from scenario 4 (red, solid) and the underlying unclipped signal (blue, dashed). Here clipping is added to signal in the same way as in scenario 3, and the high frequency component has been limited to the same interval as in scenario 2.

(b) The two signal components used to generate the signal. The high frequency component is constrained to a short interval using heaviside functions.

Low pass filtering of simulated data

The low pass FIR-filters were generated using the MATLAB function `fir1` with both low pass properties and high pass properties, all using a cut off frequency of $5000Hz$. As the filter length is expected to influence the results, a few different filter lengths was compared in the tests. The way the performance was measured was through RMS error between the filter output and the true source, so the output of the low pass filter was compared to the simulated low frequency component and vice versa for the output from the high pass filter. In addition an optical comparison was made by plotting a few of the simulation sets alongside the filter outputs.

4.2.2 Results and discussion

The results from the simulation above are summarized as the average RMS error for each filter length in table 4.2 below.

Filter order	Average root-mean-square error			
	Scenario 1	Scenario 2	Scenario 3	Scenario 4
10	2.2459	0.6406	1.2037	0.3257
12	1.5921	0.4705	1.0795	0.2892
14	1.0526	0.3340	1.0023	0.2648
16	0.6466	0.2365	0.9630	0.2520
18	0.3695	0.1751	0.9468	0.2453
20	0.2003	0.1416	0.9416	0.2417
22	0.1129	0.1254	0.9405	0.2409
26	0.0757	0.1125	0.9413	0.2388
30	0.0791	0.1077	0.9426	0.2385
34	0.0786	0.1055	0.9438	0.2383
38	0.0789	0.1050	0.9452	0.2395
42	0.0814	0.1053	0.9465	0.2396
50	0.0844	0.1062	0.9489	0.2412
58	0.0848	0.1051	0.9507	0.2411
66	0.0806	0.1019	0.9516	0.2390
74	0.0734	0.0970	0.9515	0.2389

Table 4.2: The average RMS error between the low frequency source and the output from the low pass filter (lower is better). The results are divided into the four different scenarios described in section 4.2.1 above and a few different filter orders are displayed. The average was obtained using Monte Carlo sampling with 10000 samples.

Table 4.2 provides an overview of the performance of the filters in the different scenarios. To elaborate some on these results, a few samples of the filter outputs are discussed in the following sections. In the following plots, the filter outputs of two filters of different order ($n = 20$ and $n = 74$) are compared to the low frequency source. The phase difference of the filter output has been compensated for when plotting the output.

1. No clipping and no heaviside

Without clipping and without limiting the high frequency component to a certain interval, the filters output is very close to the source signal. The filter struggles to reproduce the source signal for filter orders less than ~ 20 but performs well for filter longer than that. Remnants of the high pass filter can be seen as ripples in the output of the filter of order $n = 20$. Below is a plot of the average RMS error as a function of filter order.

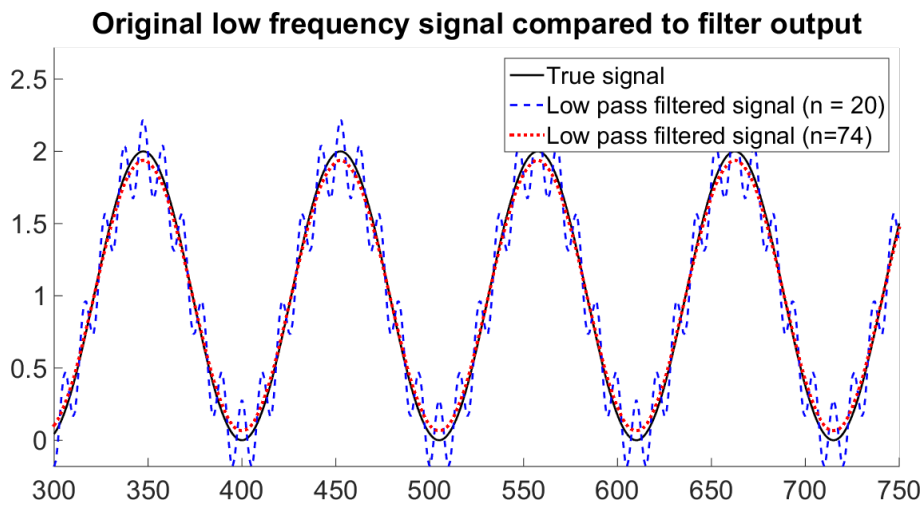


Figure 4.12: The output from the low pass filter in scenario 1, using two different filter orders, $n = 20$ (blue, dashed) and $n = 74$ (red, dotted), the original data is included for comparison (black, solid).

2. No clipping, with heaviside

When the high frequency signal is limited to a short interval using heaviside functions, the long transient response for the longer filter comes into play. This can be seen in 4.13 as a deviation from the source between $t = 440$ to $t = 480$. Errors are present around $t = 400$ as well but not as prominent. The high frequency component is active between $t = 400$ to $t = 470$ in this case.

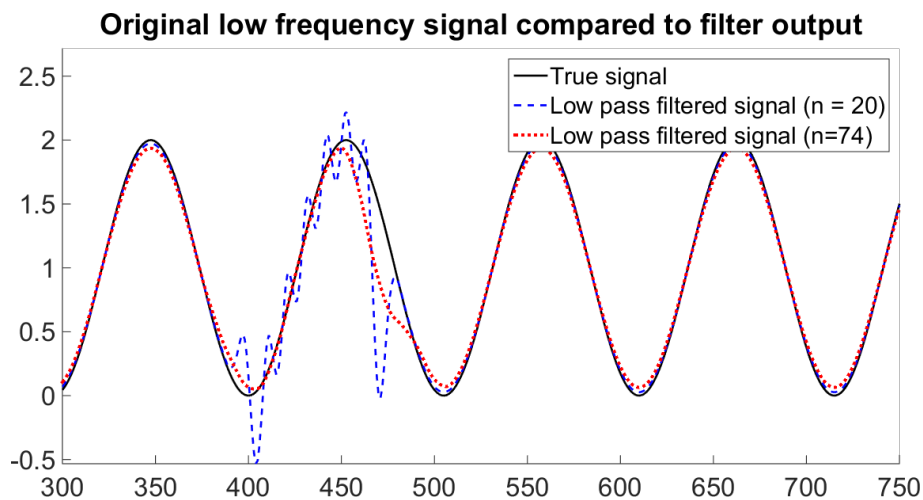


Figure 4.13: The output from the low pass filter in scenario 2, using two different filter orders, $n = 20$ (blue, dashed) and $n = 74$ (red, dotted), the original data is included for comparison (black, solid).

3. Clipping, no heaviside

With clipping added to the signal there is a drastic difference between the filter output and the source. The output still manages to match the frequency of the source but there is a large bias in the amplitude estimate.

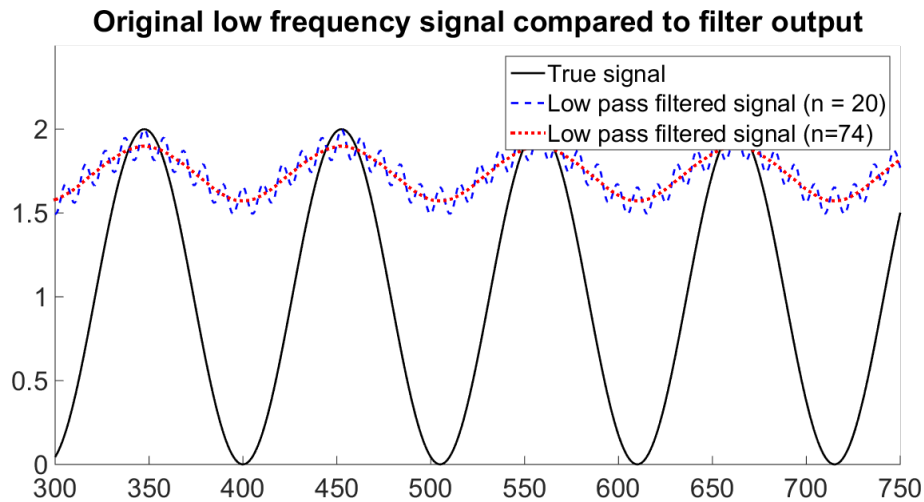


Figure 4.14: The output from the low pass filter in scenario 3, using two different filter orders, $n = 20$ (blue, dashed) and $n = 74$ (red, dotted), the original data is included for comparison (black, solid).

4. Clipping, with heaviside

In figures 4.15 the error can be seen clearly. Comparing the filter output to those in scenario 3 (with clipping, no heaviside), the outputs are practically identical in the interval $t \in [400, 470]$, i.e. where the high frequency component is active. The similarities to scenario 2 are not as striking which seems to indicate that the clipping is a larger problem than the transient response.

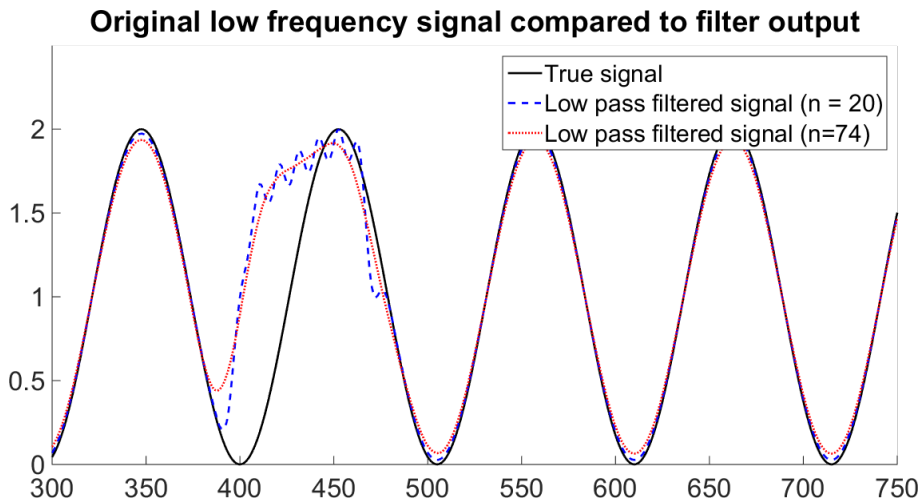


Figure 4.15: The output from the low pass filter in scenario 4, using two different filter orders, $n = 20$ (blue, dashed) and $n = 74$ (red, dotted), the original data is included for comparison (black, solid).

4.2.3 Conclusions

While filter designs other than FIR exist, that possibly could give slightly better results, they cannot be expected to be extremely superior to FIR. Transient responses is a problem affecting all filters and the only way to decrease its effect is to reduce the filter order. Regarding clipping, this basically adds an offset to the signal by removing negative values, it is unlikely that a low pass filter can reconstruct the source of a signal affected by clipping. With this, it is clear that other methods for removing the ringing must be pursued.

4.3 Estimating ringing parameters

As the clipping in the signal cause the filter output to have large errors, another method is needed. As the perturbation is known to be a relatively "clean" damped sinusoid, a parametric approach could possibly give good results. A parametric approach strives to fit the perturbation to a model, as long as the model can describe the signal well, this could give good results. This section strives to investigate how well the algorithm HTLS can be expected to perform on ion current measurements, by testing it on simulated data.

4.3.1 Methodology

The choice of algorithm used to estimate frequency and damping parameters fell on the HTLS largely because of its efficiency, which is important in the high speed application that is an engine. As with all estimations, there will be an error which is random in its nature.

By studying the error in a simulated environment, its characteristics can be described. By doing so, a method for refining the estimate can be developed.

An outline of the method for analysing the performance of the estimates can be seen as a flow chart in figure 4.16 below.

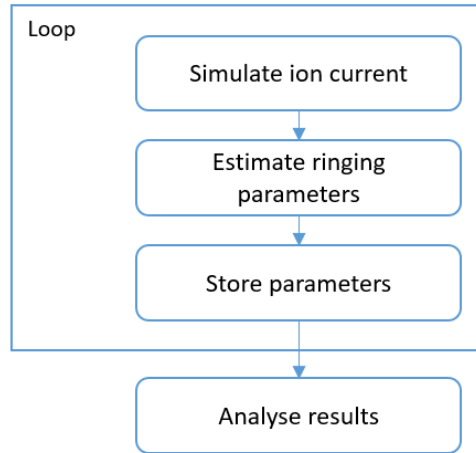


Figure 4.16: An overview of the method used for analysing the estimates. The loop was run 1 000 000 times and the results analysed with regards to average error, standard deviation and minimum and maximum values.

When simulating the ion current measurements, frequency and damping of the damped sinusoid was kept as constants between simulations Amplitude and phase was drawn randomly from the range specified in table 4.3 below. The motivation for this can be found in section 2.2.1 and 4.1.1. As the focus of the test was to examine the performance of HTLS when estimating the damped ringing, the parameters of the Gaussian shaped signals were kept at a fixed value.

A	d/f_s	f/f_s	ψ	f_s
$\in [27, 33]$	0.0857	0.0952	$\in [0, 2\pi]$	105 000

Table 4.3: The parameter intervals used to simulate the damped ringing in the test of the HTLS algorithm. The damping coefficient and frequency are kept at fixed values while the amplitude and phase are drawn from a uniform distribution with the limits described in the table above. The frequency and damping coefficients are normalised with the sampling frequency, f_s .

In the simulation, a few slightly different HTLS-based methods were used to estimate the frequency, damping coefficient, amplitude and phase of the ringing. The difference stems from whether the data used for estimation has been high pass filtered or not. High pass filtering was shown to give bad results when clipping is present in the signal, so the idea is

here to use high pass filtering after the damped ringing has decreased enough in amplitude so clipping has stopped. By using HTLS on high pass filtered data, there is a possibility that low frequency components that could corrupt the estimates are removed, which could result in better estimates. As frequency-damping pairs and complex amplitudes are estimated separately, this results in four different methods that are compared to each other. A more detailed view of one pass through the loop in figure 4.16 can be seen below.

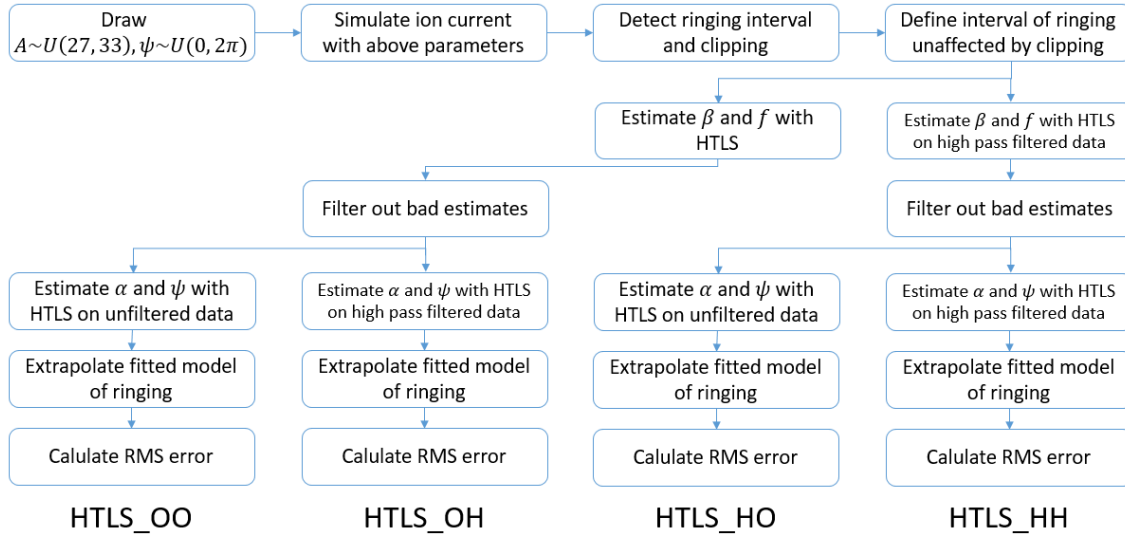


Figure 4.17: Description of the method used to study the estimation error associated with using the HTLS algorithm. The methods are run 1 000 000 times and estimated frequencies, damping coefficients and RMS error are stored after each estimate.

Standard HTLS

To find the error distribution, the above methods was run 1 000 000 times and for each simulation the estimated frequency, damping coefficient and RMS error were stored. The results can be seen below. The different methods are referred to as HTLS_XX, where XX takes on one of the values OO, OH, HH or HO, depending on the path the method takes through figure 4.17. O stands for "Original data set" and H for "high pass filtered data", so OH means that the method uses the original data set when estimating frequency and damping coefficient and high pass filtered data when estimating amplitude and phase with least squares.

	$f - \hat{f}$	$\text{std}(f - \hat{f})$	$d - \hat{d}$	$\text{std}(d - \hat{d})$
HTLS_OO	-0.000165	0.000570	-0.001509	0.005886
HTLS_OH	-0.000165	0.000570	-0.001509	0.005886
HTLS_HH	0.000038	0.000588	-0.000247	0.005036
HTLS_HO	0.000038	0.000588	-0.000247	0.005036

Table 4.4: A summary of the error of the estimated frequencies and damping coefficients for the different methods (lower is better). As the frequency is estimated in the first stage of the algorithm, the frequencies will be identical for HTLS_OO and HTLS_OH, as well as for HTLS_HH and HTLS_HO.

By comparing the estimated frequencies and damping coefficients with the true parameter values, one can get a sense of how the methods stack up against each other. Looking only at frequency and damping estimates, the method using the high pass filtered data seems to be the better alternative.

RMS error for the different methods				
	Average	std	min	max
HTLS_OO	1.034754e+10	1.033574e+13	6.629500e-02	1.033574e+16
HTLS_OH	8.864252e+09	8.863516e+12	9.446110e-02	8.863516e+15
HTLS_HH	1.573176e+12	1.570404e+15	7.604387e-02	1.570402e+18
HTLS_HO	3.639563e+12	3.615476e+15	1.742397e-02	3.615395e+18

Table 4.5: A summary of the RMS error of the four different methods (lower is better).

Table 4.5 shows the average RMSE which is a good way to compare methods against each other, but the measure can be somewhat misleading, especially when there is a large spread in the values. Therefore, as a complement, the empirical cumulative distribution function of the RMSE is also shown below, calculated with the built-in MATLAB function `ecdf`.

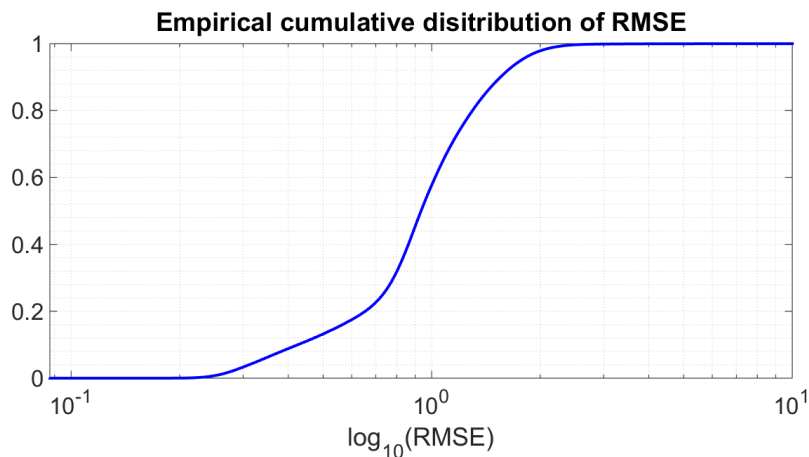


Figure 4.18: The empirical cumulative distribution function for the RMSE. The plot is limited to the interval 0 to 10, there are 248 values of the RMSE above 10.

From figure 4.18 above it can be deduced that approximately 60% of the RMS errors are below 1. This might seem like a good result at first but as the ringing is present in only approximately 70 samples out of 900, this is actually fairly high. As all data points outside of the ringing interval is zero, all errors comes from the short interval where the high frequency component is present. For the RMSE to be above 1, the sum over the errors squared actually needs to be larger than 900, and that is from only about 70 values. This means that the error on average is $\sqrt{900/70} \approx 3.5$, which is quite large.

4.3.2 Conclusion

From the tables and figure above, it is clear that the methods is not satisfactory. The large errors stem from the estimated damping coefficients, if this estimate has too large errors, the error of the estimated signal will grow exponentially when the data is extrapolated. One way to remedy this problem would be to impose some sort of discrimination criteria onto the estimated damping coefficients. By setting up some criterion the estimates that are obviously wrong could be eliminated. However, as the frequency and damping coefficients are assumed to be constant, there are better methods.

Recursive HTLS

If the estimated frequency-damping pairs are treated as noisy observations of the true parameters and if the error is assumed to be zero-mean, the average of the observations will converge to the true parameter values. By looking at the calculated average frequency and damping in table 4.4 above, the error can be concluded to not be *exactly* zero-mean. How big error this will contribute to remains to be investigated. By replacing the estimated

frequency-damping pairs with the average over all observations, the variance of the estimator will be reduced, and over time, the average will converge to a value close to the true parameter values.

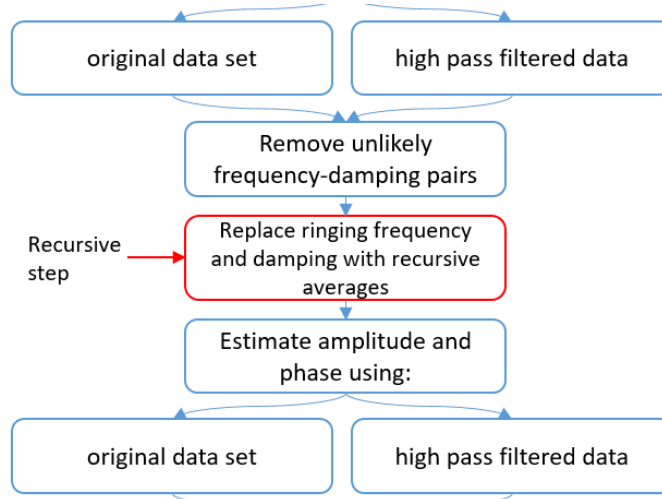


Figure 4.19: The recursive HTLS method. This method differs from the standard HTLS method in that it uses a recursive average of the ringing frequency and damping coefficient instead of using the estimates given by HTLS directly. This is illustrated with an extra step, as compared to figure 4.17.

This method was implemented in MATLAB and the results can be seen in table 4.6 below.

RMS error for the recursive methods				
	Average	std	min	max
HTLS_OO	0.298122	0.052964	0.028590	4.841331
HTLS_OH	0.296910	0.044563	0.092068	2.553825
HTLS_HH	0.303192	0.051764	0.038921	5.257370
HTLS_HO	0.562949	0.226573	0.005111	12.513195

Table 4.6: The RMS error for the different recursive methods (lower is better). The highlighted values are the best in each category.

4.3.3 Conclusion

Which of the methods above that is preferred over the others depends on the application as well as how the rest of the algorithm performs. One conclusion could be that HTLS_HH has the lowest average error which seems to indicate that this method performs better, the averages however differ with very little between the four methods. It could be that the

max error is a more relevant measure of the performance of the method as a large error could in extension lead to poor optimisation of the engine. In this case HTLS_OH would be the best candidate. This method however relies on the data passing through a high pass filter which in real world applications adds a phase shift of the signal, which translates to longer computation time. This is an advantage for HTLS_OO which does not rely on a high pass filtering of the data, while maintaining decent results.

For this reason, the final algorithm designed to clean the ringing from the ion current measurements will use the HTLS_OO version.

Chapter 5

Algorithm

This chapter outlines the proposed algorithm and evaluates the implementation using simulated data as well as real ion current measurements. As foundation for the algorithm design is the tests performed and conclusions drawn in chapter 4.

5.1 Proposed algorithm

The foundation for the proposed algorithm design comes from the previous chapters and chapter 4 in particular. The most critical conclusions are summarised below.

- a low pass filter approach will not work due to clipping
- a damped sinusoid model describes the perturbation well in the interval of interest
- the frequency and damping coefficient of the perturbation are constants
- HTLS produce good estimates of frequency and damping coefficients on average

The resulting algorithm is briefly described below in nine steps. The full algorithm is disclosed in appendix B.

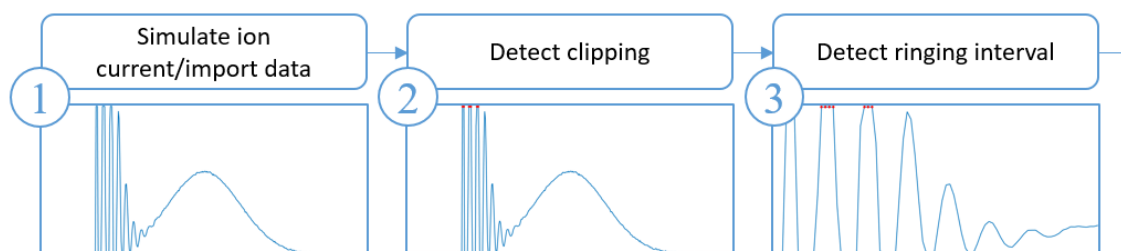


Figure 5.1: Step 1 to 3 of the algorithm. **Step 1** simulates an ion current measurements, or, if the algorithm is run on real data, imports the measurements. **Step 2** uses threshold values to detect data points affected by clipping. Such points are highlighted in red. **Step 3** finds the interval affected by ringing.

Step 1 The ion current measurements are simulated using the method described in section 4.1 above, alternatively loaded for when real data is used.

Step 2 In this step, the data points outside the measurement range are marked as affected by clipping. The script is keeping track of whether the data point is affected by clipping from above or below.

Step 3 The interval affected by damped ringing is detected and the indices are stored for further processing. Deciding where the ringing ends is not trivial, but it is important to define a range where good information of the ringing can be extracted. In this implementation, a simple approach is taken, where the ringing interval is defined to start at the first data point clipped from above, and ends at some predefined crank angle.

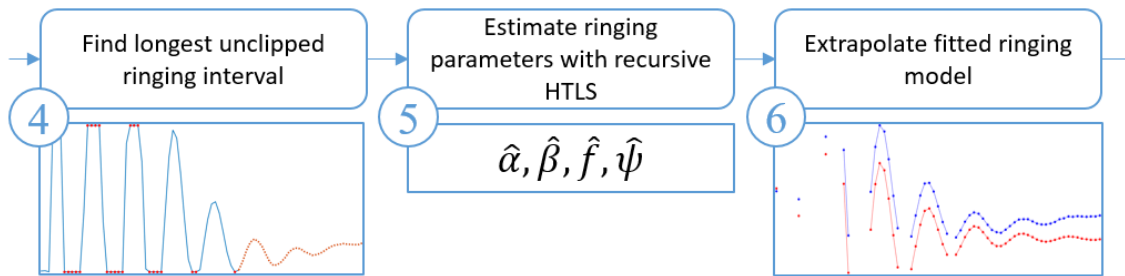


Figure 5.2: Step 4 to 6 of the algorithm. **Step 4** finds the longest interval unaffected by clipping, this will be used for parameters estimation. The longest interval is plotted using red dots above. **Step 5** uses the recursive HTLS algorithm to estimate the parameters of the damped ringing. **Step 6** estimates the damped ringing using the parameter estimates from step 4.

Step 4 As more data means better estimates of ringing parameters, and as HTLS requires uniformly sampled data, the longest interval unaffected by clipping is found.

Step 5 The ringing parameters are here estimated, using the data interval found in step 4. The method used for estimation is the recursive HTLS algorithm developed in section 4.3.

Step 6 Using the estimated parameters from step 5, an estimate of the ringing can be obtained.

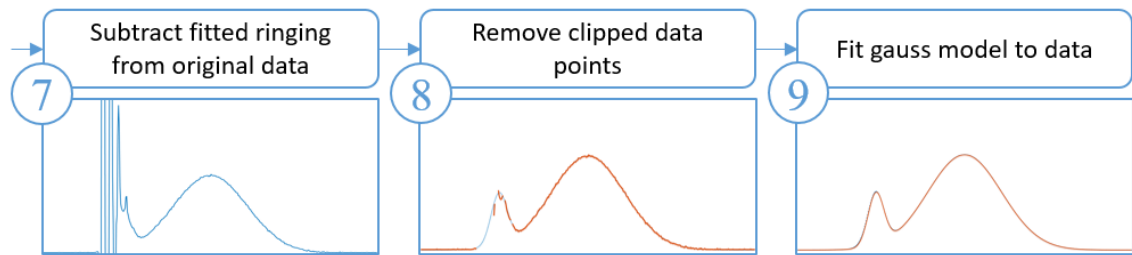


Figure 5.3: Step 7 to 9 of the algorithm. Step 9 is only performed when simulated data is used. **Step 7** subtracts the estimated ringing from the ion current measurements. **Step 8** cleans the data points affected by clipping from the ringing interval. **Step 9** fits a gauss model to the data. The fitted model can be compared to the simulated source to evaluate the algorithm. If real ion current measurements are used, this step is skipped.

Step 7 The estimated ringing is subtracted from the ion current measurements. In figure 5.3, step 7 above, some high frequency interference can still be seen where the ringing was. This is because the estimated ringing is subtracted from the points affected by clipping, this creates new interference. However, this only affects the data points that already are rendered useless due to clipping.

Step 8 In this step the data points affected by clipping are cleaned from the data set as they have no further informational value. If real ion current measurements are used, this is the final step of the algorithm. This is because fitting a gauss model to the data, which is performed in step 9, provides no additional information on the performance of the algorithm when working with real ion current measurements.

When the algorithm is evaluated on real ion current measurements, other methods has to be used. These are described in section 5.3 below.

Step 9 When the algorithm is evaluated using simulated data, a model consisting of a sum of two gauss functions are fitted to the data. As the underlying true gauss model is known, the fitted data can be compared to the source as a measure of the performance of the algorithm.

5.1.1 Algorithm settings

The algorithm also takes a few different input values, where applicable, these has been described below.

Step 3 - Detect ringing interval In this step, the interval containing the ringing is defined. With simulated data, this is defined as the interval from the first data point

affected by clipping from above, crank angle 11° .

Step 5 - Estimate ringing parameters with recursive HTLS As this method is based on HTLS, it needs the same parameters as the ordinary HTLS method. The recursive implementation adds a few parameters. All parameters required for this step are a model order estimate; filtering interval for the frequency estimate; filtering interval for the damping coefficient.

Step 9 - Fit gauss model to data This part of the algorithm attempts to fit a gauss model to the data with MATLAB's built-in function `fit()`. This function is an implementation of non-linear least squares and the calculation can become quite cumbersome. To increase efficiency, some restrictions are imposed on the space searched by the algorithm.

A complete summary of the parameters used by the algorithm can be found in appendix C.

5.2 Testing - Simulated data

This part of the testing the algorithm uses simulated data. With simulated data, a direct comparison can be made between the algorithm output and the source components.

5.2.1 Methodology

When testing the algorithm on simulated data, the algorithm was run multiple times, on data simulated with varying input parameters. By varying the parameters in realistic intervals, an estimate of how the algorithm will perform on real data can be obtained. The closer the simulated data resembles the true ion current measurements, the better the estimated performance will be, it is therefore important to choose the parameter intervals realistically. The intervals chosen for the simulation can be seen in table 5.1 below.

		Simulation parameters	
Ringing parameters		Gauss bell 1	Gauss bell 2
f	$= 10\,000\text{ Hz}$	A_1	$\in [0.5, 1.5]$
β	$= 0.08571\text{ s}^{-1}$	μ_1	$\in [-0.2, -3]^\circ$
α	$\in [27, 33]$	σ_1	
ψ	$\in [0, 2\pi)$		A_2
x_0	$\in [-6, -4]^\circ$		$\in [1.5, 2.4]$
			$\mu_2 \in \mu_1 + [14, 25]^\circ = \in [15, 28]^\circ$ [11]
			σ_2

Table 5.1: The parameter intervals used when testing the algorithm on simulated data. The naming convention of the parameters comes from the model formulated in equations (4.3) to (4.5).

Where external sources have been used to establish reasonable intervals, this is indicated by citing the source on the intervals. Where citation is not given, the intervals has been established previously in the report. Parameters are all drawn independently from each other from a uniform distribution with the endpoints defined by the intervals in table 5.1.

The evaluation of the algorithm was performed by repeatedly generating simulated ion current measurements. The algorithm was then applied to the data after which the estimated parameters where stored along with the RMS error.

5.2.2 Results

There are many possibilities to choose from when analysing the results. The RMS is a good way to get a sense of the general performance. Furthermore, as all parameters are stored, they can be compared directly to the parameters used in simulation to get more details of the performance. In this section, the results presented are mainly based on the RMS error of the fit, and the estimates of μ_1 and μ_2 . μ_1 and μ_2 is the position of the centre of the peaks of the gauss functions.

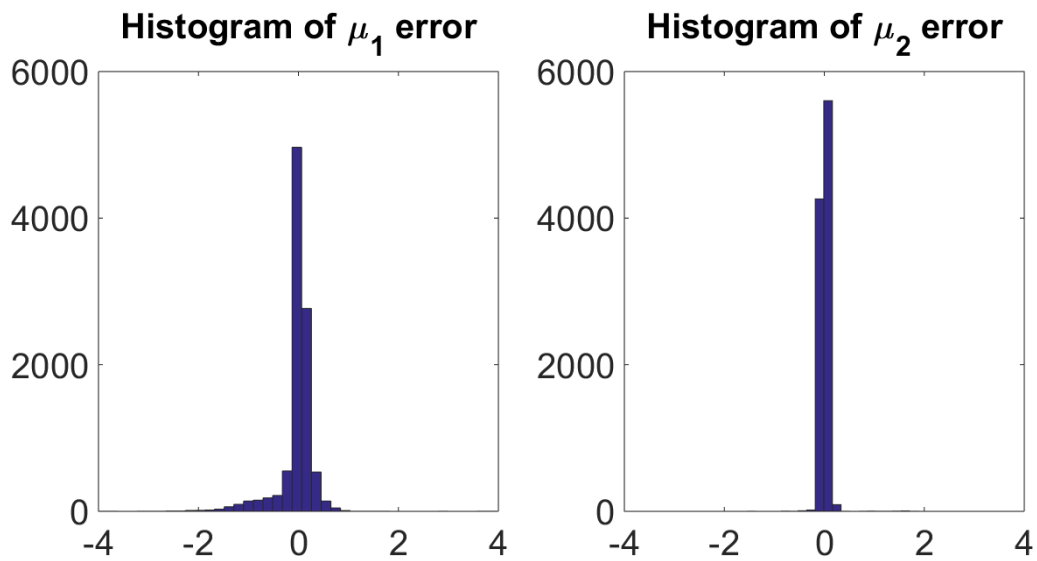


Figure 5.4: A histogram showing the distribution of the estimation errors of μ_1 and μ_2 . μ_1 corresponds with the gauss peak covered by ringing, and μ_2 with the second peak. A distinct difference can be seen between the two distributions in that the error for μ_1 has larger variance.

As the simulation parameters are stored, parameter sets which result in bad output from the algorithm can be analysed closer. By doing so, a few key categories where the algorithm struggle to produce usable output was identified, these follow below.

Low amount of data points to HTLS

For some configurations of simulation parameters, very few data points can be used by HTLS, this can cause bad parameter estimates.

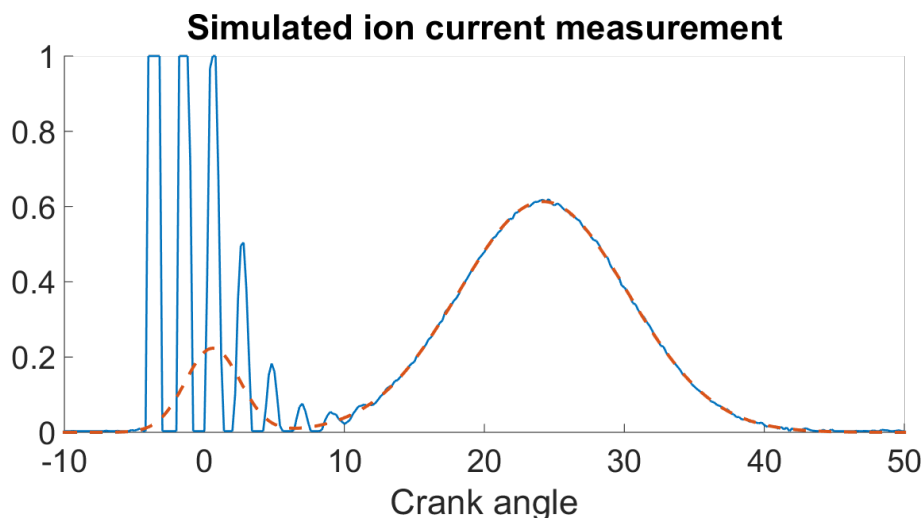


Figure 5.5: Simulated ion current measurement (blue, solid) and underlying ion current model (red, dashed). The source signal has its peaks well separated which causes the ion current component to almost go to zero between the two gauss bells at around 5-8 crank degrees. After the ringing is added, a lot of clipping is therefore present in this interval.

Low amplitude on chemical ionisation peak

If the simulated ion current has low peak separation coupled with a small amplitude for the first peak, the algorithm generally has troubles finding good estimates.

Unfortunate clipping

As clipped data points cannot be used for analysis, the amount and places where clipping occur will affect the output. Which points are affected by clipping is random due to the noise added to the signal previous to clipping. Because of this, a set of parameters can produce a near perfect output one time, and complete garbage another. This random effect is more of a problem when the data for instance has few data points to pass to HTLS, or when the first gauss peak lacks definition.

5.3 Testing - Real data

In this section, the algorithm is applied to real data as opposed to simulated. As the actual underlying signal, the ion current, is unavailable for comparison, the quantification of the results are not as straightforward as it is with simulated data. Therefore, the results are presented as samples from the algorithm output, which are commented and discussed.

5.3.1 Methodology

The algorithm is essentially run in the same way as in the case of simulated data. The only difference being that data is loaded instead of simulated, and the ringing interval is selected manually. When it comes to analysis of results, step 9 of the algorithm is skipped when real data is tested.

5.3.2 Results

Below are some of the results obtained by applying the algorithm to the data provided by SEM AB.

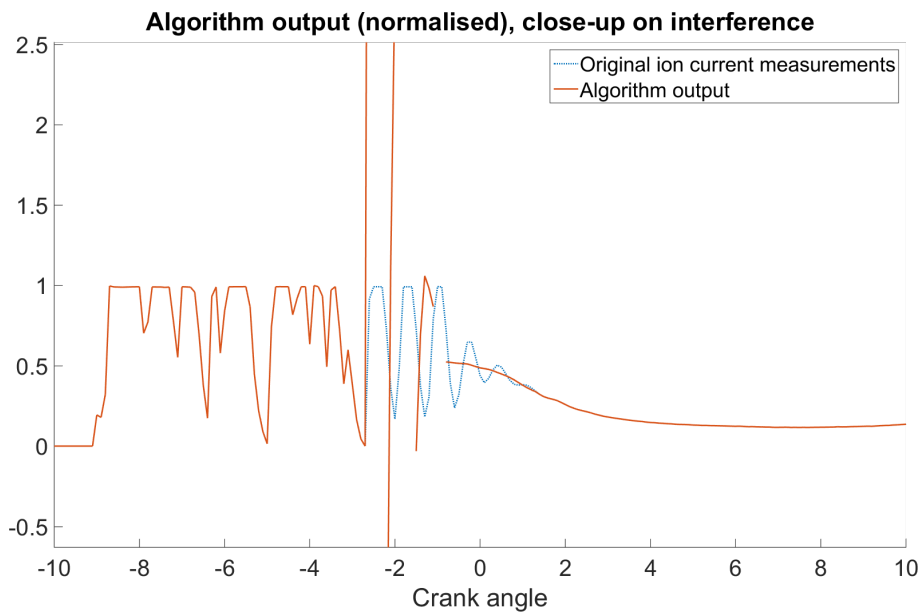


Figure 5.6: An example of the output of the algorithm when it is applied to real ion current measurements. The damped ringing starts at -2.6 crank degrees and has faded away completely at approximately 1.3 crank degrees. The algorithm seems to produce bad values between -2.6 and -0.8 crank degrees, after this the output seems reasonable. Gaps are added to the algorithm output to indicate that clipping is present at that data point.

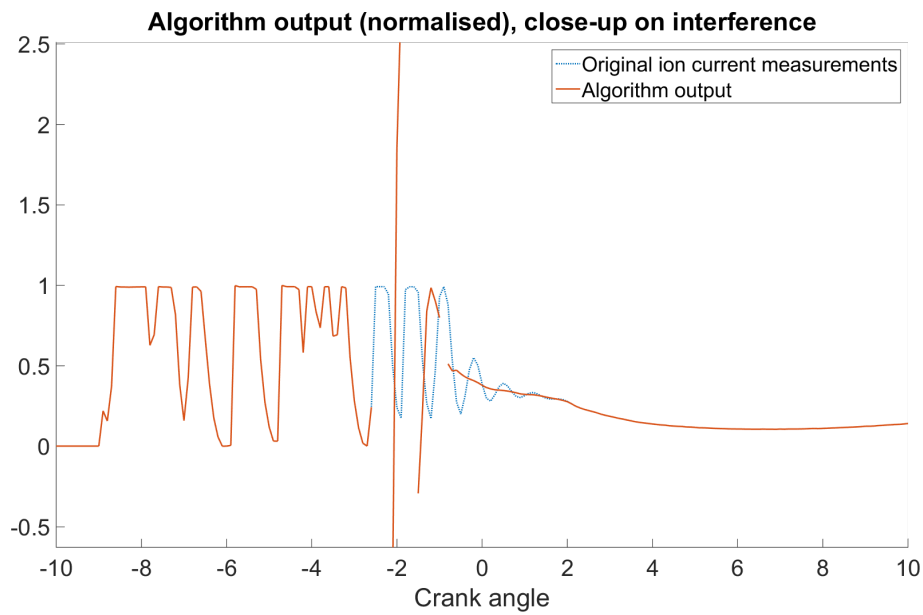


Figure 5.7: An example of the output of the algorithm when it is applied to real ion current measurements. Gaps are added to the algorithm output to indicate that clipping is present at that data point.

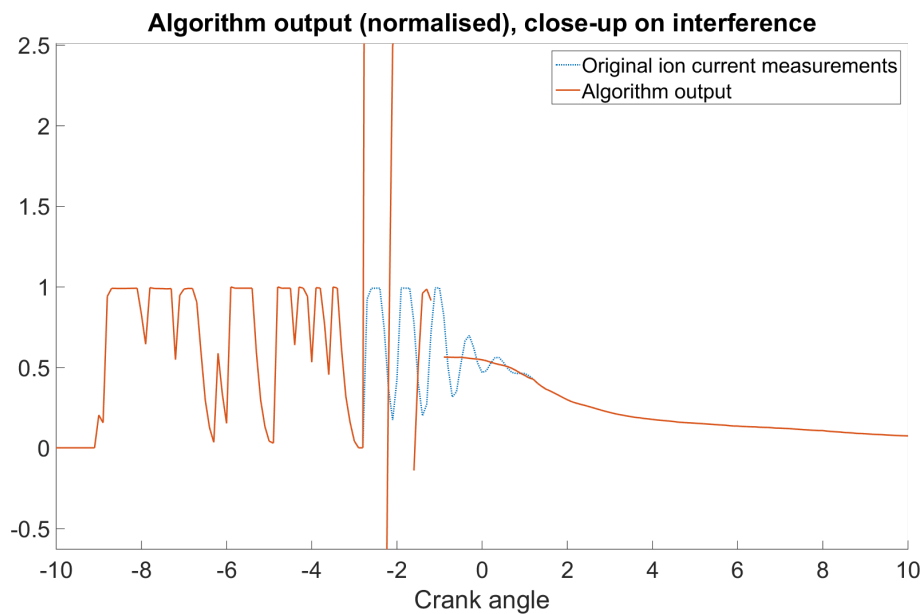


Figure 5.8: An example of the output of the algorithm when it is applied to real ion current measurements. Gaps are added to the algorithm output to indicate that clipping is present at that data point.

The plots in figures 5.6 - 5.8 above indicate that the algorithm has some flaws, in that it produce bad values in the first part of the damped ringing. However, the output seems reasonable from approximately -1 crank degrees and forward. The bad values seem to be present in the output when a lot of the data points are affected by clipping. This is likely due to the effects added to the ringing around the clipped data points which are described in section 4.1.1.

In this context, the meaning of the work reasonable says little about whether the output is true to the ion current signal. In this setting it only means that the output is relatively smooth and stays within a range that can be considered a plausible range for the ion current.

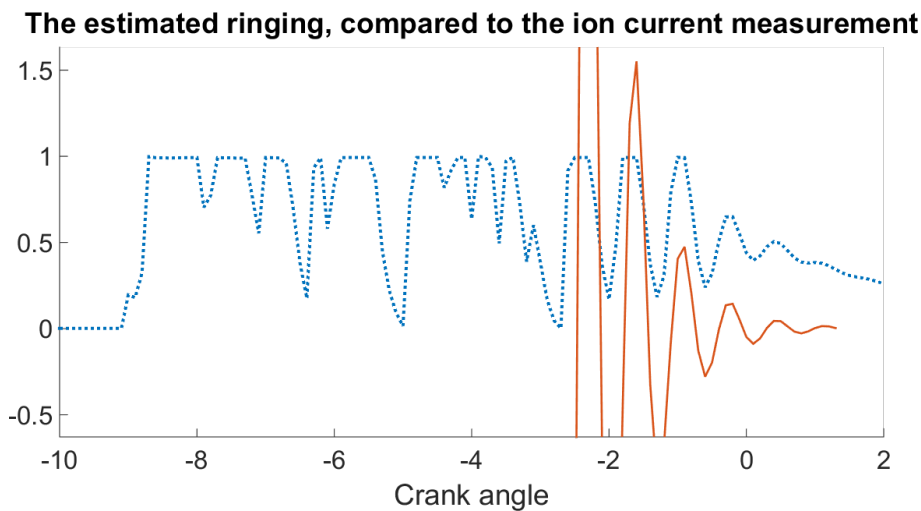


Figure 5.9: The estimated ringing (red, solid) compared to the ion current measurement (blue, dotted).

Looking at a close up of one of the estimated damped sinusoids compared to the ion current might provide some insight as to why the output varies in quality. In figure 5.9, the peaks of the estimated ringing can be seen to align well with the peaks of the measured ion current at the end of the ringing interval. The two first peaks however, do not align as well. The cause of the misalignment is unknown at this time, but it does not seem to be caused by a shift in frequency, as the period remains stable.

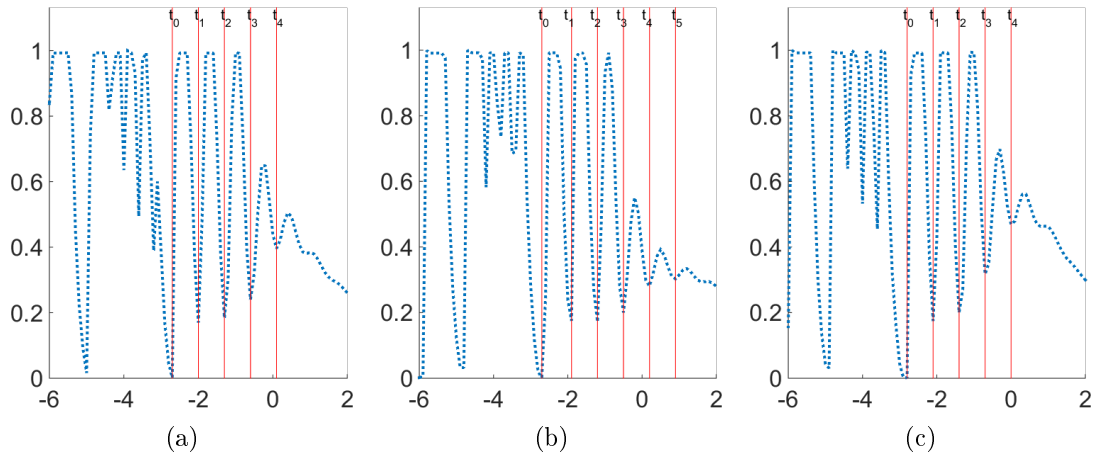


Figure 5.10: A close-up on the ion current (blue, dotted) with the positions of the troughs highlighted and labelled (red, solid). The ion current is the same as in figure 5.6, 5.7 and 5.8 above.

From figure 5.10 above, the period of the oscillations and corresponding frequency can be calculated.

	$T_n = t_n - t_{n-1}$			$f_n = 1/T_n$		
	Fig: 5.10a	Fig: 5.10b	Fig: 5.10c	Fig: 5.10a	Fig: 5.10b	Fig: 5.10c
$n = 1$	$0.1061 \cdot 10^{-3}$	$0.1061 \cdot 10^{-3}$	$0.1061 \cdot 10^{-3}$	$9.4286 \cdot 10^3$	$9.4286 \cdot 10^3$	$9.4286 \cdot 10^3$
$n = 2$	$0.1061 \cdot 10^{-3}$	$0.1061 \cdot 10^{-3}$	$0.1061 \cdot 10^{-3}$	$9.4286 \cdot 10^3$	$9.4286 \cdot 10^3$	$9.4286 \cdot 10^3$
$n = 3$	$0.1061 \cdot 10^{-3}$	$0.1061 \cdot 10^{-3}$	$0.1061 \cdot 10^{-3}$	$9.4286 \cdot 10^3$	$9.4286 \cdot 10^3$	$9.4286 \cdot 10^3$
$n = 4$	$0.1061 \cdot 10^{-3}$	$0.1061 \cdot 10^{-3}$	$0.1061 \cdot 10^{-3}$	$9.4286 \cdot 10^3$	$9.4286 \cdot 10^3$	$9.4286 \cdot 10^3$
$n = 5$		$0.1212 \cdot 10^{-3}$			$8.2500 \cdot 10^3$	

Table 5.2: A summary of the recorded periods in some of the ion current measurements. The results are approximate but seem to indicate that the period and hence the frequency is stable for the duration of the damped ringing.

Figure 5.10 and table 5.2 indicate that the period remain stable for the duration of the damped ringing. One instance of longer period can be seen for $n = 5$, but this is likely a result of the samples misaligning with the actual trough of the ringing, rather than a case of unstable period. Although the period remain stable, the shape of the sinusoid can be seen to be distorted early on in the damped ringing.

Chapter 6

Discussion

This chapter attempts to address and motivate some of the choices made throughout the thesis, as well as reason around how these choices have affected the results. Some alternative methods are discussed and how they might have performed in comparison with the proposed algorithm. Lastly, some suggested future improvements are discussed. The chapter is divided into two sections which correspond to chapter 4 and 5 respectively.

6.1 Methodology

This section discuss some of the work that was presented in chapter 4.

6.1.1 Simulation model

Using simulated ion current measurements to design an algorithm has a lot of advantages, as it is much easier to evaluate the performance of a certain method. With simulated data, lack of data is also never a problem as it is generated on demand.

There are however disadvantages one should be aware in the form of simulation model design. As the method developed will be based on simulated data rather than real data, it is important that one is aware of the simplifications that has been made. In the model developed in this thesis, simplifications has been made both concerning the clipping, the ion current, and the perturbations originating from the ignition system. Of these three, the one concerning the ion current are considered to have a relatively small impact compared to the others.

The simplification of the perturbation involves using only a damped sinusoid for the entire interference interval. As the objective of the thesis has been to remove these types of signals, extending this interval so that it makes up the entire perturbation should be fine. The only drawback should be that the element of detecting the start of the damped sinusoid

in the midst of the interference is not added to the algorithm. The switch into damped sinusoid was however shown to be fairly easy to detect due to the fact that this happens after a fixed time interval.

The simplification concerning the clipping can not easily be dismissed. The effect of the clipping change the shape of the interference such that a damped sinusoid model might no longer fit the data. These effects have a fairly large impact on the algorithm output and they should probably be incorporated into the model in a future version. One way of doing so could possibly be to run the estimated ringing through a mathematical model of a capacitor. This way, it could be possible to achieve the exponential decay present after a peak.

6.1.2 Low pass filter tests

The section testing low pass filters with clipped data, and data where the high frequency component is present for a short duration, does the tests to investigate how well a low pass filter would work with ion current. The signals used to test this is however simplified in a few different ways, how these potentially affect the outcome of the tests and therefore the conclusions drawn will be discussed below.

In the tests, two perfect sinusoids were used, one representing the ion current, and one representing the high frequency perturbation. Neither the ion current, nor the high frequency interference can be described well by these types of signals. However, by using perfect signals, the filter is given the best possible conditions, and any error in these simulations would likely be worse when using real data. This way, if the low pass filter performs badly in the tests, it can safely be concluded to perform worse with real data which is not as "nice". The simplifications made in these tests can therefore be concluded to not affect the decision to not use low pass filters.

6.1.3 HTLS

The method used for estimating the ringing parameters was HTLS. The main issue with this method is that it requires uniformly sampled data, which, with all the clipping affecting the signal is rare to come by. The search for unclipped intervals often result in the algorithm resorting to using data points at the end of ringing. The data quality at this part of the ringing is often not optimal, as the signal to noise ratio (SNR) has decreased as a consequence of the damping. Other methods that handle non-uniformly sampled data could possibly have worked better. Unfortunately, the methods found that handle non-uniformly sampled data assumed no damping was present in the signal. A method could possibly have been constructed in the scope of this thesis, but this was decided against for a couple of reasons. Time restrictions was one such reason, and having HTLS that seemed to work fairly well was another. The main reason however comes from how the clipping is added. As has been shown previously in this work, the points leading up to, and directly

following clipping from below are corrupted. The potential benefit from getting more data points to work with would therefore likely be outweighed by the disadvantage of including bad data.

6.2 Algorithm

This section discuss some of the work that was presented in chapter 5.

6.2.1 Implementation

In step 3 of the algorithm, the ringing interval is selected. The way this is done with simulated data is very static, in that the interval is selected as starting at the first clipped peak and extending to 11 crank degrees. The 11° point is selected somewhat arbitrarily so that for most simulations the ringing has faded out by then. The static interval has drawbacks in that the ringing for some instances has faded out completely well before the 11° mark. As the algorithm often selects the ends of the interval for estimations, data with very low SNR is passed to the HTLS algorithm which can lead to bad estimates. On most data sets however, the data selected has good informational value.

When testing the algorithm on real data, the ringing interval was selected manually so the issue of passing data with low SNR to the HTLS algorithm is avoided. This is however not a realistic procedure. However, it has been shown that the start of the damped ringing can found easily. By using the estimated damping coefficient, a good estimate of where the damped ringing ends can then be found.

6.2.2 Future work

The perhaps most important further improvement that is needed for the algorithm is to develop a method to deal with the effects around the clipping. These effects are likely one of the largest contributions to why the algorithm fails to clean the interference from the ion current measurements. One method to do so, could possibly involve very much the same algorithm design, but with the addition of filtering the estimated damped ringing through some filter to achieve a similar distortion as is present in the measurements.

In the implementation, thermal drift was disregarded. For the data set used, this is likely not problematic, as the data is recorded during a short time interval (~ 5 minutes) and at constant RPM. However, for more realistic conditions, RPM will vary rapidly as the car accelerates and decelerate, resulting in fluctuations of temperature. Thus, thermal drift of the parameters can likely not be disregarded. A proposed method of dealing with the drift would be to implement a different method for updating the estimated frequency and damping coefficient. In the current implementation, this estimate is given by the average over all observations. This has the disadvantage that it weights all observations equally.

A method that weights observations from long ago less, would make the algorithm more responsive to change in frequency and damping parameters. Methods such as recursive least squares, or a Kalman filter are examples of online estimators that likely would perform well. They also have the advantage of being able to tune the responsiveness of the estimates depending on how much the thermal drift affect the parameter.

Furthermore, by making the algorithm track the frequency and damping parameters, it is possible that information on for instance the wear and tear of the ignition system can be acquired. How this drift develop as a function of temperature and wear and tear, remains to be investigated.

In this thesis, the data used has exclusively come from a capacitive ignition system. These systems has three different types of interference coming from the ignition system, where the damped sinusoid makes up roughly a third of the total interference. However, other systems exist, such as inductive ignition systems, where the damped sinusoid start directly after the spark, and stay active for longer. With these types of systems, the potential gain from the proposed algorithm is naturally larger.

Chapter 7

Conclusion

The objective of this thesis has been to investigate how much information that can be extracted from ion current measurements in the intervals obscured by coil ringing. The work has concluded that the interference can be divided into three different intervals, where this work has focused on the last interval. This interval contains a signal that can be well described by a damped sinusoid model.

In addition to being obscured by the high frequency signals, the signal is affected by clipping where the signal exceeds the range of the measurement equipment. In the close proximity of the data points clipped from below, the damped sinusoid is warped in such a way that it in these parts of the signal it is no longer described well by a damped sinusoid model.

The method developed in this thesis works well with the simulated data and recovers the underlying ion current in a satisfactory way. The algorithm is furthermore efficient and while it in its current state does not run as quick as an engine at full speed, it is close enough that with further improvements it could be made efficient enough.

With that being said, the results on real data shows that more work is needed. The algorithm performs well after the clipping has subsided and the uncovered data looks legitimate. When clipping is present in a close proximity to the data points however, the output of the algorithm suffers.

Bibliography

- [1] May 22, 2017. [Online]. Available: <http://www.xbhp.com/talkies/attachments/general-biking-discussion/204930d1455784547-making-kawasaki-ninja-300-san-4-stroke-diagram.jpg>.
- [2] M. W. Stockel, M. T. Stockel, and C. Johanson, *Auto Fundamentals: workbook*, Workbook. Goodheart-Willcox Co, Jan. 2005, ISBN: 9781590703267. [Online]. Available: <http://amazon.com/o/ASIN/159070326X/>.
- [3] T. Karlsson and K. Hellström.
- [4] N. A. Henein, W. Bryzik, A. Abdel-Rehim, and A. Gupta, "Characteristics of ion current signals in compression ignition and spark ignition engines," *SAE Int. J. Engines*, vol. 3, pp. 260–281, Apr. 2010. DOI: 10.4271/2010-01-0567. [Online]. Available: <http://doi.org/10.4271/2010-01-0567>.
- [5] J. Angeby, "Method for the estimation of combustion parameters," US Patent App. 11/887,185, May 2009. [Online]. Available: <https://www.google.ch/patents/US20090132145>.
- [6] A. Jakobsson, *An Introduction to Time Series Modeling*. Studentlitteratur AB, 2015, ISBN: 9789144108360. [Online]. Available: <https://books.google.se/books?id=K6nLjwEACAAJ>.
- [7] S. Vanhuffel, H. Chen, C. Decanniere, and P. Vanhecke, "Algorithm for time-domain nmr data fitting based on total least squares," *Journal of Magnetic Resonance, Series A*, vol. 110, no. 2, pp. 228–237, 1994, ISSN: 1064-1858. DOI: <http://dx.doi.org/10.1006/jmra.1994.1209>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1064185884712095>.
- [8] D. Schlichthärle, *Digital Filters: Basics and Design*, 2nd ed. 2011. Springer, Apr. 2011, ISBN: 9783642143243. [Online]. Available: <http://amazon.com/o/ASIN/3642143245/>.
- [9] J. O. S. III, *Introduction to Digital Filters: with Audio Applications*. W3K Publishing, Oct. 2007, ISBN: 9780974560717. [Online]. Available: <http://amazon.com/o/ASIN/0974560715/>.
- [10] H. Klövmark, P. Rask, and U. Forssell, "Estimating the air/fuel ratio from gaussian parameterizations of the ionization currents in internal combustion si engines," SAE Technical Paper, Tech. Rep., 2000.
- [11] L. Nielsen, L. Eriksson, and M. Glavmo, "Metod för bestämning av läget för maximalt tryck under en förbränning i en förbränningsmotor," SE PRV. SE504197 C2, 1996.

[Online]. Available: http://was.prv.se/spd/patent?p1=3bDF-b90Qyd7eM42P9NdVA&p2=S7_FgZDKVLk&tab=2&applicant=sem+ab&range=10&hits=true&lang=en&hitsstart=30&start=33.

[12] SEM AB, *Ion current measurements*.

[13] J. ÄNGEBY, May 24, 2017.

Appendices

Appendix A

Simulate ion current script

```
function [ IonCurrent , y_ringing , y_gauss , CrankAxis ] = ...
    SimulateIonCurrent( gaussparams , ringparams , rpm , sigma_static )
%SimulateIonCurrent Simulates an ion current with the characteristics
%defined by the parameters specified
% INPUT:
%     gaussparams:    [6x1] Parameters for gaussbells
%                     [amp1, mu1, sigma1, amp2, mu2, sigma2]
%     ringparams:    [5x1] Parameters for ringing
%                     [amp, damp, f (Hz), phase, ringstart]
%     rpm:           Speed of engine, measured in rounds per minute
%     sigma_static   St.Dev. for the static noise
% OUTPUT:
%     IonCurrent:    Ion current including characteristic perturbation
%     y_ringing:     Ringing signal component
%     y_gauss:       Gauss component
%     CrankAxis:     Crank axis degrees corresponding to the ion current

N = 900;
t = (1:N).';
CrankAxis = linspace(-54, 125.8, N)';
fs = rpm2fs(rpm);
threshold_high = 3.226;
threshold_low = 0.01;
ringstart = ClosestValue(CrankAxis, ringparams(5));
gaussparams([2,5]) = ...
    arrayfun(@(val) t(ClosestValue(CrankAxis, val)), gaussparams([2,5]));

model_ringing = @(x, t) ...
    real(x(1)*...
```



```

        exp(-x(2)*(t-x(5))).*(exp(2*pi*x(3)*1i*(t-x(5)) + x(4)*1i))...
        .*logical heaviside(t-x(5)));
model_gauss = @(x, t) ...
    real(x(1)*exp(-(t - x(2)).^2 / (2*x(3)^2)));

% ===== Setup ringing =====
y_ringing = model_ringing([ringparams(1:4), ringstart], t);
% ===== Setup gauss bells =====
y_gauss = model_gauss(gaussparams(1:3), t) + ...
    model_gauss(gaussparams(4:6), t);
% ===== Setup static noise =====
noise_static = randn(N, 1)*sigma_static;

% ===== Add sources together, add clipping =====
IonCurrent = real(y_ringing + y_gauss + noise_gauss + noise_static);
IonCurrent = clipping(IonCurrent, threshold_high, threshold_low)

end

function [ y_clipping, index, index_h, index_l ] = ...
    clipping( y, threshold_high, threshold_low )
%clipping(y, threshold) Adds clipping to the input data vector. Replaces
%data points larger than the provided threshold with the threshold value.
% INPUT:
%   y:          Data vector
%   threshold:  Clipping threshold
% OUTPUT:
%   y_clipping: Data with large values replaces by the threshold value
%   index:      Index of the data points subject to clipping

index_h = y >= threshold_high;
index_l = y <= threshold_low;
y_clipping = y;
y_clipping(index_h) = threshold_high;
y_clipping(index_l) = threshold_low;
index = logical(index_h + index_l);
end

```

Appendix B

Algorithm

```
function [IonCurrent_ringRemoved_NaN, ringparams, configs] = ...
    RingingRemoval(IonCurrent, CrankAxis, configs)
% RingingRemoval Removes the ringing from simulated ion current measurements.
% IonCurrent      Ion current measurement
% CrankAxis       Crank angle synchronised with ion current
% configs         Contains engine settings, HTLS settings etc.

% Engine settings
rpm = configs.Engine.rpm;
fs = rpm2fs(rpm);
threshold_high = configs.Engine.threshold_high;
threshold_low = configs.Engine.threshold_low;

% Define ringing model
DSinus = configs.DSinus;

%HTLS settings
f_lim = configs.HTLS.f_lim;
d_lim = configs.HTLS.d_lim;
w_RecMean = configs.HTLS.w_RecMean;
d_RecMean = configs.HTLS.d_RecMean;
NumRecMean = configs.HTLS.NumRecMean;
K = configs.HTLS.K;

N = length(IonCurrent);
t = 1:N;

%% Detect clipping
[~, clipping_ind, clipping_ind_high] = ...
```

```

    clipping(IonCurrent, threshold_high, threshold_low);
%% Detect ringing
RingInterval = [CrankAxis(find(clipping_ind_high, 1)), 11];
RingInterval_idx = ClosestValue(CrankAxis, RingInterval);

t_ringing = RingInterval_idx(1):RingInterval_idx(2);
ringing_clipping_ind = clipping_ind(t_ringing);
clipping_idx_ringing = t_ringing(ringing_clipping_ind);

%% Detect longest ringing interval without clipping
tmp = [clipping_idx_ringing, t_ringing(end)];
[~, b] = max(diff(tmp));
if any(clipping_idx_ringing == tmp(b+1))
    t_longest_noclipping = (tmp(b)+1):(tmp(b+1)-1);
else
    t_longest_noclipping = (tmp(b)+1):(tmp(b+1));
end

%% Recursive HTLS
M = ceil(length(t_longest_noclipping)/2);
if M > K
    [w,d] = ...
        htls(IonCurrent(t_longest_noclipping), M, K);
    % Filter out based on limits
    filterfreq = ...
        logical((w > (f_lim(1)/fs)) .* (w < (f_lim(2)/fs)) .* (d > d_lim(1)));
    % If one parameter set matches filter
    if any(filterfreq)
        % Find parameter set closest to average
        w_RecMean
        [tmp_i, w_closest] = ClosestValue(w, w_RecMean)
        d_closest = d(tmp_i)
        % Update average
        configs.HTLS.w_RecMean = ...
            (1/(NumRecMean+1))*(NumRecMean*w_RecMean + w(tmp_i));
        configs.HTLS.d_RecMean = ...
            (1/(NumRecMean+1))*(NumRecMean*d_RecMean + d(tmp_i));
        configs.HTLS.NumRecMean = NumRecMean + 1;
        tmp_i = find(abs(w) == w_closest);
        % Replace best estimate with recursive average
        w(tmp_i) = configs.HTLS.w_RecMean .* sign(w(tmp_i));
        d(tmp_i) = configs.HTLS.d_RecMean .* sign(d(tmp_i));
    else

```

```

        % If too few data points are available , the algorithm is ended
        IonCurrent_ringRemoved_NaN = [];
        ringparams = [];
        return;
    end
else
    IonCurrent_ringRemoved_NaN = [];
    ringparams = [];
    return;
end
[A_,ph_]=linpara(IonCurrent(t_longest_noclipping),...
    K, w, d,...
    t_longest_noclipping-t_longest_noclipping(1));
% tmp_i(2) corresponds with the positive frequency
ringparams = ...
    [A_(tmp_i(2)),...
    mean(d_RecMean),...
    mean(w_RecMean),...
    ph_(tmp_i(2)),...
    t_ringing(1)];
ringing_recon = 2*DSinus(ringparams, t_ringing-t_longest_noclipping(1));

%% Subtract fitted ringing from data
IonCurrent_ringRemoved = ...
    model_subtraction(IonCurrent, t, ringing_recon', t_ringing);

IonCurrent_ringRemoved_NaN = IonCurrent_ringRemoved;
IonCurrent_ringRemoved_NaN(clipping_idx_ringing) = NaN;
end

```

Appendix C

Algorithm - parameters

```
% HTLS
configs.HTLS.f_lim = [4000 16000]; % Interval for ringing
configs.HTLS.d_lim = [0 1]; % Interval for damping coefficient
configs.HTLS.w_RecMean = 0.095598333236509; % Frequency
configs.HTLS.d_RecMean = 0.085255629989495; % Damping coefficient
configs.HTLS.NumRecMean = 1; % Starting recursive mean
configs.HTLS.K = 6; % Model order

% Define ringing model
configs.DSinus = @(p, x) ...
    real(p(1)*exp(-p(2)*x).*exp(2*pi*1i*p(3)*x)*exp(p(4)*1i));
```