THE EFFECT OF SAMPLING RATE AND SIGNAL-TO-NOISE RATIO ON METHODS FOR THE AUTOMATED DETERMINATION OF SUSTAINED MAXIMUM AMPLITUDES IN VIBRATION SIGNALS

A Master’s Thesis
Presented to
The Academic Faculty

by

Nathaniel DeVol

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
August 2021

COPYRIGHT © 2021 BY NATHANIEL DEVOL
THE EFFECT OF SAMPLING RATE AND SIGNAL-TO-NOISE RATIO ON METHODS FOR THE AUTOMATED DETERMINATION OF SUSTAINED MAXIMUM AMPLITUDES IN VIBRATION SIGNALS

Approved by:

Dr. Christopher Saldana, Co-advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Katherine Fu, Co-advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Thomas Kurfess
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: April 21, 2021
ACKNOWLEDGEMENTS

I want to start by thanking my advisors, Dr. Christopher Saldana and Dr. Katherine Fu. Without your expertise and guidance, I would not have gotten to this point. I also greatly appreciate, Dr. Thomas Kurfess for his continued guidance and support in addition to serving on my committee. I’m truly glad that I decided to come to Georgia Tech and work alongside each of you.

I am also grateful for our partners at Moog Inc. Our cooperation inspired this thesis, and their support and technical insights ensured that it was a success. I am pleased to have been part of the team.

Next, I want to thank my fellow labmates for their support and entertainment. They were a joy to work with and made my experience truly unique.

Lastly, I want to thank all my family and friends that have been with me along this journey. Your encouragement kept me going in the toughest of moments.

This work was supported in part by DE-EE0008303.
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ........................................................................................................................iii

LIST OF TABLES ........................................................................................................................................vi

LIST OF FIGURES ......................................................................................................................................viii

LIST OF SYMBOLS AND ABBREVIATIONS ..........................................................................................xi

SUMMARY ..................................................................................................................................................xii

CHAPTER 1. Introduction ......................................................................................................................... 1
  1.1 Motivation ....................................................................................................................................... 2
  1.2 Problem Statement ....................................................................................................................... 2
  1.3 Structure of this Thesis ............................................................................................................... 4

CHAPTER 2. Background ............................................................................................................................. 6
  2.1 Optical Encoders ........................................................................................................................... 6
  2.2 Signal Processing .......................................................................................................................... 7
    2.2.1 Time Domain ....................................................................................................................... 7
    2.2.2 Time-frequency Domain ..................................................................................................... 14
  2.3 Burst Detection ............................................................................................................................... 17

CHAPTER 3. Signal Processing Methods ................................................................................................. 22
  3.1 Moving RMS .................................................................................................................................. 23
  3.2 STFT Peaks ..................................................................................................................................... 25
  3.3 Peak Envelope .............................................................................................................................. 29
  3.4 LP Rectifier Envelope .................................................................................................................. 30
  3.5 Peak Distances ............................................................................................................................. 31
  3.6 Expire Timer .................................................................................................................................. 33
  3.7 Peak Height at Width .................................................................................................................... 37
  3.8 Summary of Signal Processing Methods ....................................................................................... 38

CHAPTER 4. Methodology .......................................................................................................................... 39
  4.1 Experimental Data ........................................................................................................................ 39
  4.2 Simulated Data .............................................................................................................................. 40
  4.3 Signal Disturbances ....................................................................................................................... 42
  4.4 Ground Truth .................................................................................................................................. 43
  4.5 Signal Processing Parameters ....................................................................................................... 45
  4.6 Design of Experiment .................................................................................................................... 45

CHAPTER 5. Results .................................................................................................................................... 48
  5.1 Stationary Signals .......................................................................................................................... 48
    5.1.1 Sampling Rate ...................................................................................................................... 48
    5.1.2 Signal-to-Noise Ratio ........................................................................................................... 49


**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Haar wavelet decomposition.</td>
<td>18</td>
</tr>
<tr>
<td>Table 2</td>
<td>Experimental data collection parameters</td>
<td>40</td>
</tr>
<tr>
<td>Table 3</td>
<td>Gaussian parameters for introduced disturbance.</td>
<td>43</td>
</tr>
<tr>
<td>Table 4</td>
<td>Summary of parameters used in each signal processing method.</td>
<td>45</td>
</tr>
<tr>
<td>Table 5</td>
<td>Design of experiment.</td>
<td>46</td>
</tr>
<tr>
<td>Table 6</td>
<td>Percent error of each method while varying sampling rates on a stationary signal with no noise.</td>
<td>48</td>
</tr>
<tr>
<td>Table 7</td>
<td>Percent error of each method while varying SNR on a stationary signal with a 10 kHz sampling rate.</td>
<td>49</td>
</tr>
<tr>
<td>Table 8</td>
<td>Average percent error of each method across multiple disturbance types while varying sampling rates with no noise.</td>
<td>51</td>
</tr>
<tr>
<td>Table 9</td>
<td>Standard deviation of each method across multiple disturbance types while varying sampling rates with no noise.</td>
<td>51</td>
</tr>
<tr>
<td>Table 10</td>
<td>Average percent error of each method across multiple disturbance types while varying SNR.</td>
<td>53</td>
</tr>
<tr>
<td>Table 11</td>
<td>Standard deviation of each method across multiple disturbance types while varying SNR with a 10 kHz sampling rate.</td>
<td>53</td>
</tr>
<tr>
<td>Table 12</td>
<td>Error in the presence of different disturbances with a 7.5 kHz sampling rate and no noise.</td>
<td>55</td>
</tr>
<tr>
<td>Table 13</td>
<td>Error in the presence of different disturbances with a 7.5 kHz sampling rate and a SNR of 10.</td>
<td>55</td>
</tr>
<tr>
<td>Table 14</td>
<td>Timing results for each of the signal processing methods in milliseconds.</td>
<td>56</td>
</tr>
<tr>
<td>Table 15</td>
<td>Number of tuning parameters for each signal processing method.</td>
<td>61</td>
</tr>
<tr>
<td>Table 16</td>
<td>Minimum detectable frequencies for the tuning parameters used in this work.</td>
<td>62</td>
</tr>
<tr>
<td>Table 17</td>
<td>Summary of the performance across all test cases.</td>
<td>63</td>
</tr>
</tbody>
</table>
Table 18 – All results for test case a with no disturbance. 79

Table 19 – All results for test case b: a gaussian disturbance with an amplitude of 0.5 and standard deviation of 0.005. 80

Table 20 – All results for test case c: a gaussian disturbance with an amplitude of 0.5 and standard deviation of 0.01. 81

Table 21 – All results for test case d: a higher-order gaussian disturbance with an exponent of 3, amplitude of 0.5 and standard deviation of 0.005. 82

Table 22 – All results for test case e: a higher-order gaussian disturbance with an exponent of 3, amplitude of 0.5 and standard deviation of 0.01. 83
LIST OF FIGURES

Figure 1  – Components of an optical linear encoder. 7

Figure 2  – The effect of the RMS window size and phase angle on the RMS value of a sine wave. 9

Figure 3  – Percentage error of RMS for different window sizes and phase angles. 10

Figure 4  – Peak spline envelope of a signal with a frequency-swept from 10 to 100 Hz. 12

Figure 5  – Reduced bandwidth (V_{DD}) signal generation system. (a) Initial pass and (b) iterative algorithm. 13

Figure 6  – Low-pass rectifier envelope of a signal with a frequency-swept from 10 to 100 Hz. 13

Figure 7  – Hilbert transform amplitude for a signal with a frequency-swept from (a) 50 to 100 Hz and (b) 10 to 100 Hz. 17

Figure 8  – Schematic of (a) wavelet tree and (b) shifted wavelet tree. 19

Figure 9  – An example of spike train data. 19

Figure 10 – Test signals used to demonstrate methods. One (a) is comprised of a single sine function. The other (b) is comprised of two sine functions. 23

Figure 11 – Example implementation of the Moving RMS method bounding (a) a 70 Hz signal and (b) a composite 70 and 140 Hz signal, both with two gaussian disturbances. 25

Figure 12 – (a) Example 70 Hz signal with two gaussian disturbances enveloped by the Hilbert transform. (b) 70 Hz and 140 Hz signal bounded by the Hilbert transform. 26

Figure 13 – A signal comprised of two sine waves with amplitudes 1.0 and 0.5 and frequencies 800 and 1000 Hz (a) raw signal and (b) Fourier transform. 27

Figure 14 – A signal comprised of two sine waves with amplitudes 1.0 and 0.5 and frequencies 800 and 1600 Hz (a) raw signal and (b) Fourier transform. 28
Figure 15 – Example implementation of **STFT Peaks method**

Figure 16 – Example implementation of **Peak Envelope method**

Figure 17 – Example implementation of **LP Rectifier Envelope method** bounding (a) a 70 Hz signal and (b) a composite 70 and 140 Hz signal, both with two gaussian disturbances.

Figure 18 – Algorithm for implementing **Peak Distances method**.

Figure 19 – Example implementation of **Peak Distances method** applied to a test signal with (a) two gaussian disturbances and (b) a single gaussian disturbance.

Figure 20 – (a) Test 70 Hz signal with two gaussian disturbances, (b) the result of the first step of the **Expire Timer method** applied to the test signal, and (c) an illustration of the two timers used to test the signal.

Figure 21 – Algorithm for the implementation of the **Expire Timer method**.

Figure 22 – Figure 22 – Experimental set-up with (a) accelerometer attached to EMCO where arrow indicates sensitivity direction, and (b) the hardware used to collect the data.

Figure 23 – Fourier transform of the experimental test data.

Figure 24 – Simulated signal used to test the signal processing methods.

Figure 25 – Disturbance types applied to a (a) stationary, 800Hz signal. (c) \( n = 1, a = 0.5, \sigma = 0.005 \), (c) \( n = 1, a = 0.5, \sigma = 0.01 \), (c) \( n = 3, a = 0.5, \sigma = 0.005 \), (e) \( n = 3, a = 0.5, \sigma = 0.01 \).

Figure 26 – Algorithm for iterating through peak data to find ground truth value.

Figure 27 – Comparison of the performance of each method on a stationary signal with varying sampling rates and no noise.

Figure 28 – Comparison of the performance of each method on a stationary signal with varying SNR.

Figure 29 – Comparison of the average performance of each method across multiple disturbance types with varying sampling rates and no noise.

Figure 30 – Detailed result of the average performance of each method across multiple disturbance types with varying sampling rates and no noise.
noise. Error bars represent the range in performance across the different disturbance types. (a) *STFT Peaks*, (b) *Peak Envelope*, (c) *LP Rectifier Envelope*, (d) *Peak Distances*, and (e) *Expire Timer*.

Figure 31 – Comparison of the average performance of each method across multiple disturbance types with varying SNR.

Figure 32 – Detailed result of the average performance of each method across multiple disturbance types with varying SNR. Error bars represent the range in performance across the different disturbance types. (a) *STFT Peaks*, (b) *Peak Envelope*, (c) *LP Rectifier Envelope*, (d) *Peak Distances*, and (e) *Expire Timer*.

Figure 33 – *Peak Distances* method result with 10 kHz sampling rate, SNR of 5 and (a) disturbance case b and (b) disturbance case c.
# LIST OF SYMBOLS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNC</td>
<td>Computer Numeric Control</td>
</tr>
<tr>
<td>DAQ</td>
<td>Data Acquisition</td>
</tr>
<tr>
<td>DES</td>
<td>Discrete Energy Statistics</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>EMD</td>
<td>Empirical Mode Decomposition</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>HAT</td>
<td>Hardware Attached on Top</td>
</tr>
<tr>
<td>IIoT</td>
<td>Industrial Internet of Things</td>
</tr>
<tr>
<td>IMF</td>
<td>Intrinsic Mode Functions</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>LP</td>
<td>Low-Pass</td>
</tr>
<tr>
<td>LPF</td>
<td>Low-Pass Filter</td>
</tr>
<tr>
<td>RMS</td>
<td>Root-Mean Square</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>STFT</td>
<td>Short-Time Fourier Transform</td>
</tr>
<tr>
<td>WT</td>
<td>Wavelet Transform</td>
</tr>
</tbody>
</table>
SUMMARY

Machine condition monitoring has been proven to reduce machine down time and increase productivity. State of the art research uses vibration monitoring for tasks such as maintenance and tool wear prediction. A less explored aspect is how vibration monitoring might be used to monitor equipment sensitive to vibration. In a manufacturing environment, one example of where this might be needed is in monitoring the vibration of optical linear encoders used in high precision machine tools and coordinate measuring machines. Monitoring the vibration of sensitive equipment presents a unique case for vibration monitoring because an accurate representation of the maximum sustained vibration is needed, as opposed to extracting trends from the data. To do this, techniques for determining sustained peaks in vibration signals are needed. This thesis fills this gap by formalizing and testing methods for determining sustained vibration amplitudes. The methods developed are tested on simulated signals based on experimental data. Results show that processing the signal directly with the novel Expire Timer method produces the smallest amounts of error on average under the various test conditions. Additionally, this method can operate in real-time on streaming vibration data.
CHAPTER 1. INTRODUCTION

With rising environmental concerns, there is an increased need for sustainable manufacturing with respect to maintenance programs and resource efficiency [1]. The recent rise in the Internet of Things (IoT) and the Industrial Internet of Things (IIoT) have helped to address these needs by providing insights into manufacturing processes. The IoT is a network of objects that share information and work together toward a common goal [2]. When applied to manufacturing or industrial applications, this paradigm is referred to as the IIoT [3]. The ability for manufacturing systems to communicate allows for predictive and prescriptive operations that can improve performance and realize invisible issues [4]. The most common maintenance program that uses the IIoT is condition-based maintenance. Condition-based maintenance uses condition monitoring of a machine to predict when a machine will require maintenance [5]. IIoT also has many applications in lean manufacturing, including optimizing supply chains [6], just-in-time material delivery [7], and predicting part quality [8]. While part quality and machine condition are traditionally studied separately, it is clear they are strongly interrelated [9]. In precision manufacturing, one example of where these two areas merge is in the machine’s encoders. High precision machine tools and coordinate measuring machines rely on readings from their encoders to properly position themselves. When the encoders experience vibrations, and the scanning head is vibrating in relation to the scale, the vibration movement is registered as spindle movement, resulting in positional error [10, 11]. In more extreme conditions, the vibrations can cause temporary or permanent loss of position [12]. These errors directly translate to dimensional errors in the parts being produced. With the help of
the IIoT, the vibrations at the encoder reader heads can be monitored in real time to produce alarms and notify users when errors may be present.

1.1 Motivation

Encoder manufacturers specify acceptable vibration levels with two components: shock and sustained. The shock limit is higher than the sustained limit, has no time constraint associated with it, and can be easily determined by checking for signal values that cross a threshold. The sustained limit has a time associated with it for which an amplitude must persist before it is considered sustained. Because vibration signals are oscillating, the sustained maximum of a signal cannot be determined by simply detecting if the incoming signal remains above the threshold for the specified time. Additionally, current vibration-based condition monitoring techniques rely on extracting trends from the data. In the case of monitoring glass scales, these methods are not sufficient because an accurate representation of the sustained maximums is needed. For these reasons, methods for extracting the sustained maximums from vibration data need to be developed and tested so that they can be used on edge devices for condition monitoring of optical encoders.

1.2 Problem Statement

The objective of this thesis is to formalize and test methods for extracting sustained amplitudes from vibration signals so that algorithms for monitoring optical encoders can be implemented on edge devices. From this objective, four main research questions are to be answered:

1. How can maximum sustained amplitudes in vibration data be detected?
2. What is the effect of sampling rate and signal-to-noise ratio on methods available for determining maximum sustained amplitude?

3. How does the performance of methods available for determining maximum sustained amplitude change for non-stationary signals?

4. How feasible are the methods available for determining maximum sustained amplitude for implementation on an edge device?

To answer the first research question, two categories of methods are formalized and tested. The first category of methods envelope the signal and determine the sustained maximums from the envelope; these methods will be referred to as the envelope methods. The second category process the signal directly and will be referred to as the non-envelope methods. In the envelope category, four methods to envelope the signal are investigated. The envelope functions are based on prior work, but using these envelopes to detect sustained peaks presents a new application. The first envelope method uses the relationship between a sine wave and its root-mean square (RMS) to envelope the signal. The second forms the signal envelope by summing the peaks in its short-time Fourier transform (STFT). The third fits a spline to the peaks, and the fourth uses a series of rectified low-pass filters subtracted from the original signal. Two methods in the non-envelope category are novel methods developed in this work. The first of these methods searches the time between peaks in the signal, and the second applies a threshold to the signal and incorporates two timers to determine if the sustained time limit is reached.

The second research question is answered by testing the methods on a simulated signal created with varying sampling rates and signal-to-noise ratios. This simulated signal is based on a tooth pass frequency and two harmonics of it. The performance of each
method is evaluated by how close the sustained maximum they output is to the calculated
ground truth value. The envelope functions are expected to have the lowest sensitivity to
sampling rates, but the highest sensitivity to noise.

To address the second research question, disturbances are added to the same
simulated signal and compared to the baseline signal. It is expected that the RMS envelope
and STFT envelope methods will perform worse in the presence of disturbances. This is
expected because when the disturbance is partially within the RMS or STFT window, it
will cause odd effects.

For the final research question, the computation time for each method is recorded.
These times are used together with the number of parameters required to tune each method
to draw conclusions about the feasibility of implementing each method on a
microprocessor. The methods which process the signal directly, rather than forming an
envelope, are expected to have the shortest computation times since fewer steps are
required. The STFT envelope is expected to take the longest since it requires the Fourier
transform to be computed.

1.3 Structure of this Thesis

The remainder of this thesis is organized as follows. Chapter 2 details signal
processing methods that are used throughout the work, as well as a review of optical
encoders and burst detection methods. In Chapter 3, the six signal processing methods are
formalized. Chapter 4 explains how these methods are tested and analyzed to answer the
research questions. Chapter 5 contains all results for how the tested methods preformed in
comparison to the ground truth, and a discussion of these results are presented in Chapter
6. Chapter 7 reviews the contributions of this thesis, and Chapter 8 presents future work in the area. Chapter 9 discusses limitations of the work. In Chapter 10, conclusions are drawn and recommendations for when each method should be used are presented.
CHAPTER 2. BACKGROUND

2.1 Optical Encoders

Machining parts with high precision requires the machine tool to be able to properly position itself with a high degree of precision. A prior survey found that optical encoders account for the majority of sensors used for precision positioning [13]. Optical encoders are comprised of a reader head and a scale that can be either linear or rotary. The scale has graduations that the reader head uses to measure displacement [14]. Figure 1 illustrates the components of a linear optical encoder. Errors from a machine’s encoder are translated into errors in the part’s dimensions so it is important to consider sources of error for encoders. One source of error for optical encoders is related to mechanical effects from deformations, temperature variation, or vibration [12]. Vibrations of optical linear encoders are important because when the scanning head is vibrating in relation to the scale, the vibration movement is registered as spindle movement resulting in positional error [10, 11]. In extreme conditions, the vibrations can cause temporary or permanent loss of position [12]. Optical encoder manufacturers specify the acceptable vibration levels as two parts. The first is the maximum amplitude for short peaks, and the second is the maximum amplitude which persists for a given amount of time and is considered sustained.
2.2 Signal Processing

To develop methods for determining sustained maximum amplitudes from streaming vibration data, an understanding of common signal processing techniques is needed. The following section reviews signal processing techniques, which are used in Chapter 2 to develop methods for extracting sustained maximums from vibration signals. The goal of this work is to identify sustained maximums that persist for a prespecified amount of time. Because time information is needed, the following review focuses on the time and time-frequency domains, as they are the two available solution spaces.

2.2.1 Time Domain

2.2.1.1 Root Mean Squared

When the peaks of a vibration signal are desired without regard for frequency information, the signal can be interrogated in the time domain. To determine the maximums in a sinusoidal function the moving root mean square (RMS) is first considered. It is well understood from following Equations 1 through 4 that the RMS of a sine wave $x(t)$ with
amplitude $a$ will be equal to $\frac{a}{\sqrt{2}}$. The amplitude of a sine wave could thus be approximated by taking the value of its RMS and multiplying by the square root of two. This is restated in Equation 5.

$$x(t) = a \sin(t)$$  \hspace{2cm} (1)

$$x_{rms} = \sqrt{\frac{1}{2\pi} \int_{0}^{2\pi} (a \sin t)^2 \, dt}$$  \hspace{2cm} (2)

$$x_{rms} = \sqrt{\frac{a^2}{2\pi} \frac{1}{2} \left( t - \sin t \cos t \right)_{0}^{2\pi}}$$  \hspace{2cm} (3)

$$x_{rms} = a \sqrt{\frac{1}{2\pi} \frac{1}{2} 2\pi} = \frac{a}{\sqrt{2}}$$  \hspace{2cm} (4)

$$a = x_{rms} \sqrt{2}$$  \hspace{2cm} (5)

In developing this relationship, the RMS was taken over the period of the sine wave.

If the RMS window is not the period, or a multiple of it, then Equation 5 will no longer hold. Equation 3 can be adapted for arbitrary starting and ending points $p_1$ and $p_2$, and the result is shown in Equation 6.

$$x_{rms} = \sqrt{\frac{a^2}{2 \left( p_2 - p_1 \right)} \left( p_2 - \sin p_2 \cos p_2 - (p_1 - \sin p_1 \cos p_1) \right)}$$  \hspace{2cm} (6)

Equation 6 is used to determine how the RMS value relates the ratio of the window size to the period for three starting points $p_1$, and the result is shown in Figure 2. Changes
to the starting point $p_1$ is equivalent to varying the phase $\phi$ of the sine wave and will henceforth be referred to as the phase. Using the $a/\sqrt{2}$ as the ground truth, the error for each RMS value can be calculated. The result is shown in Figure 3 and will be referenced when selecting an RMS window size.

Figure 2 – The effect of the RMS window size and phase angle on the RMS value of a sine wave.
2.2.1.2 Envelope

Another approach to extracting the sustained maximums from a vibration signal is built around forming the envelope of the signal in the time domain. One such method for forming the envelope involves fitting a curve to the peaks in a signal. This method is used in the initial step of the empirical mode decomposition (EMD). EMD decomposes a data set into a finite number of intrinsic mode functions (IMF) that can be used to reconstruct the signal [15]. The EMD is a popular technique that has a wide array of applications, including roller bearing fault identification [16], characterizing oscillations in control loops [17], and analysing esophageal manometric time series in gastroesophageal reflux disease [18]. When the Hilbert transform is performed on the IMFs, it is known as the Huang-Hilbert Transform [19]. Discussion of the full details of the EMD are beyond the scope of this work, but the method used to envelope the signal will be reviewed. Interested readers can find more information in [15]. The original method developed by Huang et al used a

Figure 3 – Percentage error of RMS for different window sizes and phase angles.
cubic spline to envelope the data [15]. Because of the popularity of the technique, researchers have tried different methods for improving the envelope performance by changing the curve type to a rational Hermite [20] and B-spline [21], and different interpolation methods such as trigonometric interpolation [22] and piecewise polynomial interpolation [23]. While these methods were found to improve the overshooting and undershooting seen in the cubic spline, the complexities are beyond the scope of this work, so only discussion of the cubic spline will be included. This envelope technique is applied to a test signal $x(t)$ created by taking a chirp with a frequency that increases linearly in time from $f_0 = 10$ to $f_1 = 100$ and applying an amplitude modulation. Producing the signal in this way allows for the techniques to be tested on varying frequencies within a single signal. The equation to generate this signal is given in Equation 7, where $c$ is the chirp rate calculated using Equation 8 where $T$ is the time it takes to sweep from $f_0$ to $f_1$. The result is shown in Figure 4. As it applies to this work, the envelope of the magnitude of the signal is of interest so only one spline is shown.

$$x(t) = (1 + 0.5 \times \sin(2\pi \times 3 \times t)) \times \cos\left(2\pi \times \left(\frac{c}{2} \times t^2 + f_0 \times t\right)\right) \quad (7)$$

$$c = \frac{f_1 - f_0}{T} \quad (8)$$
Jeong et. al. developed another method for forming the envelope of an oscillating signal for use in envelope tracking power amplifiers to increase the efficiency of the supply modulator [24]. The method involves first sending the original envelope of the signal through a low-pass filter (LPF1) and subtracting the resulting signal from the original envelope ($V_{env}$). This signal is then rectified and sent through a second low-pass filter (LPF2) before being added back to the output of LPF1. The result is the reduced bandwidth signal ($V_{DD}$). This result is not guaranteed to satisfy the condition in Equation 9, so the process may need to be repeated until Equation 9 is satisfied [24]. The technique is illustrated in Figure 5 below.

$$V_{DD}(t) \geq V_{env}(t)$$ (9)

Figure 4 – Peak spline envelope of a signal with a frequency-swept from 10 to 100 Hz.
Figure 5 – Reduced bandwidth ($V_{DD}$) signal generation system. (a) Initial pass and (b) iterative algorithm [24].

When applying this method to vibration signals, the original envelope will be the absolute value of the signal under test. This envelope technique is applied to the same amplitude modulated chirp signal from Equation 7, and the result is shown in Figure 6.

Figure 6 – Low-pass rectifier envelope of a signal with a frequency-swept from 10 to 100 Hz.
2.2.2 Time-frequency Domain

2.2.2.1 Short-Time Fourier Transform

The other solution space available is the time-frequency domain. With few exceptions, a real signal can be represented as a summation of sinusoidal components. This is known as the Fourier series. When considering a discrete-time signal \( g(n) \) with \( N \) samples, the Discrete Fourier Transform (DFT) is used to calculate its Fourier series \( G(k) \). The forward DFT and inverse DFT are shown in Equation 10 and Equation 11.

\[
G(k) = \frac{1}{N} \sum_{n=0}^{N-1} g(n) * e^{-j2\pi kn/N} \tag{10}
\]

\[
g(n) = \sum_{k=0}^{N-1} G(k) * e^{j2\pi kn/N} \tag{11}
\]

The fast Fourier transform (FFT) is an implementation of the DFT, which factorizes the matrix implementation of the DFT to reduce the total number of complex operations from \( O(N^2) \) to \( O(N \log_2 N) \), while maintaining all the original properties of the DFT [5].

The Fourier series that results from a Fourier transform integrates over all time, so all the time information is lost. A simple approach to retain some time information is to perform the Fourier transform over a short, moving window. This will record how the frequency shifts over time and is called the short-time Fourier transform (STFT). It is achieved by multiplying the signal \( g(n) \) by a window function \( h(t) \) to produce a modified signal \( g_t(n) \). This modified signal is shown in Equation 12. When preforming a STFT, the
frequency resolution is inversely related to the window length and time resolution, so the trade-off must be carefully considered [25].

\[ g_t(n) = g(n) * h(n - t) \]  \hspace{1cm} (12)

The window function can take on many forms. The Gaussian function was used for the window when the STFT was first introduced [25]. Other common window functions include the rectangular window, which simply leaves the local view unaltered and sets the rest of the signal to zero, and the Hann window, which is a raised cosine window [26]. In addition to the type of window used, another parameter that arises is the hop size, or the distance the window moves along the signal before the FT is preformed again. For efficiency reasons, the hop size is usually greater than the increment \( g(n) \) is sampled in.

2.2.2.2 Wavelet Transformation

Wavelet transformation (WT) is another way to represent the signal in the time-frequency domain which seeks to resolve some of the resolution issues that are encountered with the STFT. The WT represents a signal by a set of wavelets that are derived by scaling and translating a mother wavelet. The result of the WT is like that of the STFT, but it has better time resolution at high frequencies and better frequency resolution at lower frequencies [27].

2.2.2.3 Hilbert Transform

The Hilbert Transform is used to find the corresponding function \( y(t) \) for a real function \( x(t) \), so that \( z(t) = x(t) + i * y(t) \), such that \( z(t) \) is an analytic function. In
signal processing, the Hilbert transform is computed by first taking the Fourier transform of the real valued function $x(t)$. The negative frequencies are then negated, and the positive frequencies are doubled. Finally, the inverse Fourier transform is computed. This results in a complex valued function, where the real part is the original signal and the imaginary part is the Hilbert transform [28]. The Hilbert transform can be calculated using Equation 13, where $F$ is the Fourier transform and $U$ is the unit step function.

$$z(t) = F^{-1}(F(x(t)) \ast 2U) = x(t) + i \ast y(t)$$ (13)

The amplitude of the analytic function $z(t)$ that results from the Hilbert transform can be used to envelope a signal when the input signal has a narrow bandwidth. Another interesting property of the resulting Hilbert transform, although less relevant to this work, is that the phase derivative is the instantaneous frequency [28]. When the incoming signal has a wide bandwidth, such as a vibration signal with many harmonic frequencies, then the effectiveness of this envelope is diminished. This is illustrated in Figure 7, where the envelope of an amplitude modulated chirp signal from 50 to 100 Hz is compared to an amplitude modulated chirp signal from 10 to 100 Hz signal. In the broader bandwidth signal, the envelope becomes noisy when the higher frequencies are introduced.
2.3 Burst Detection

Detecting peaks and burst in streaming data has widespread interest, including for financial data [29], sun spot data [30], network traffic [31], early detection of a disease outbreak [32], and meme-tracking [33]. It is important to distinguish between peaks and bursts. A local peak is said to exist if a value is (a) a local maximum within a window and (b) is isolated (limited points have the same value within the window) [34]. A burst differs from a peak in that a wide region of high values exist around the local maximum. This work applies to detecting bursts that persist for a prespecified amount of time. For the purposes of monitoring linear optical encoders, a burst is considered present if it exists for at least 20 ms because this is the limit imposed by the manufacturer.

Different ideas for burst detection have been proposed from a variety of fields, and a summary of them are presented here. The most common area in which burst detection work is seen is in data mining. Zhu and Shasha investigated an elastic burst detection method intended for detecting Gamma Ray Bursts in astrophysical data and demonstrated
the technique by testing it on data from the New York Stock Exchange [35]. A brute force, sliding window approach that that examines a time series of \( n \) values would require \( n \) sliding windows, and thus would have a time complexity of \( O(n^2) \). Zhu and Shasha used Haar wavelet decomposition to reduce the complexity of the elastic window model. The wavelet coefficients for the Haar wavelet decomposition are shown in Table 1. When using the wavelet decomposition for burst detection, the sum aggregates are of interest, and the differences are disregarded. This reduces the wavelet decomposition to that shown in Figure 8a. A shortcoming of this decomposition is that windows on the same level do not overlap, so a burst that traverses a window may be missed. To account for this, the shifted wavelet tree is used, which adds half overlapping windows. A schematic of the shifted wavelet tree is shown in Figure 8b. Using the shifted wavelet tree reduced the complexity of the burst search to \( O(n) \) time complexity [35].

Table 1 – Haar wavelet decomposition [35]

<table>
<thead>
<tr>
<th>Level</th>
<th>( \frac{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8}{2} )</th>
<th>( \frac{a_1 + a_2 + a_3 + a_4 - (a_5 + a_6 + a_7 + a_8)}{2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 3</td>
<td>( \frac{a_1 + a_2 + a_3 + a_4}{2} )</td>
<td>( \frac{a_5 + a_6 + a_7 + a_8}{2} )</td>
</tr>
<tr>
<td>Level 2</td>
<td>( \frac{a_1 + a_2 + a_3 + a_4}{\sqrt{2}} )</td>
<td>( \frac{a_5 + a_6 + a_7 + a_8}{\sqrt{2}} )</td>
</tr>
<tr>
<td>Level 1</td>
<td>( \frac{a_1 + a_2}{\sqrt{2}} )</td>
<td>( \frac{a_3 + a_4}{\sqrt{2}} )</td>
</tr>
<tr>
<td>Level 0</td>
<td>( a_1 )</td>
<td>( a_2 )</td>
</tr>
</tbody>
</table>
Another area in which burst detection is of interest is in neuron firing. The action potential of a firing neuron is referred to as a spike, and the firing of a neuron over time is referred to as a spike train. Bursts in spike trains, or periods where the spike frequency is relatively high, are of interest because patterns such as these are used to draw meaning from the spike train [36]. An example of spike train data is shown in Figure 9.

One method used for detecting bursts in spike trains constructs a histogram of the time interval between spikes. This histogram is used to identify the threshold between bursts and non-burst periods. This threshold is then used on each interval in the dataset to identify the start and ends of bursts. Finally, a chi-squared test is performed to compare the
number of spikes to the average expected number \([36]\). Another method for detecting bursts in spike trains uses what is termed the Poisson surprise. The Poisson surprise \(S\) is defined as the negative logarithm of the probability that a burst under investigation would occur in a random (Poisson) spike train \(P\), with the same average spike rate\(r\) containing \(n\) spikes in time integral \(T\) \([37]\). The equations for finding the Poisson surprise and the Poisson probability are shown in Equation 14 and 15 respectively.

\[
S = -\log P \tag{14}
\]

\[
P = e^{-rT} \sum_{i=n}^{\infty} \frac{(rT)^i}{i!} \tag{15}
\]

These methods for detecting bursts in spike trains are of interest here because the on-off nature of the spike train resembles that of oscillating data such as vibration and will be referred to in Chapter 3.

The data structure of spike trains parallels other areas in which burst detection is of interest. One such area is text data mining in which structure is desired to be gleaned from document streams, such as email and news articles. In this case, the data being analyzed is like a spike train, in that it is a collection of times when a word appears and is zero otherwise. Kleinberg approaches the problem using an infinite-state automation model. The states in the model correspond to exponential models, which determine the time between messages. If data is determined to be in a higher state it means that it is coming more rapidly. To avoid short bursts causing the model to move up in state, there is a transition cost associated with moving up states but not down \([38]\).
Dealing with vibration data directly, Polycarpou et al. investigated vibration bursts as a noninvasive method for determining breaker state [39]. They were not concerned with amplitudes of bursts, but rather the shapes of the envelopes and the short-time spectra curves. These shape characteristics were used to create vibration signatures that could identify changes in power circuit breaker operation. In the work, a Discrete Energy Statistics (DES) envelope was used to envelope the amplitude of the vibration signal. The DES is faster to compute than the Hilbert Transform because it does not require the evaluation of FFTs. The DES $S(i)$ of a signal $s(i)$ is given in Equation 16 [39]. The concept of fitting an envelope around the vibration signal is used in this thesis.

$$S(i) = \sqrt{s^2(i) - s(i - 1) \cdot s(i + 1)}$$  \hspace{1cm} (16)

Oscillating signals were also considered by Zhou and Dagle when investigating how to detect sustained oscillations in power grids. They did so by looking at the self-coherence spectrum of a single channel of data. The self-coherence spectrum is created by taking the coherence spectrum between the signal $x_t$ and the time delay signal $x_{t + \Delta t}$ where $\Delta t$ is the time delay [40]. The method that they developed is useful for identifying sustained frequencies, but it does not return any amplitude information. For this reason, it is not used in this work. Drawing inspiration from the prior work and background presented in this chapter, Chapter 3 describes formalized methods for the determination of sustained amplitudes in vibration data.
CHAPTER 3. SIGNAL PROCESSING METHODS

One of the main thrusts of this work was to establish and test methods for the determination of sustained maximum amplitudes in vibration signals. An amplitude is considered sustained if it persists for a prespecified amount of time $t_{sustained}$. For the motivating example, $t_{sustained}$ was set at 20 ms, since this is the specification set by the glass scale manufacturer, Heidenhain. All methods developed here can be extended to other applications with different $t_{sustained}$ values. Throughout the development of the methods in this chapter, two signals were used for demonstration purposes. Both signals consisted of a base signal with two Gaussian disturbances. The equations used to generate the base test signals are given in Equation 17 and Equation 18. The base frequency of 70 Hz was chosen because it is the tooth pass frequency of the experimental data which the simulated data was based on. In the second signal, a harmonic created by doubling the 70 Hz base frequency was added to illustrate how the presence of harmonics may affect the performance of the methods developed here.

$$signal1 = \sin(70 * 2 * \pi * t)$$ \hspace{1cm} (17)

$$signal2 = \frac{1}{2} \sin(70 * 2 * \pi * t) + \frac{1}{2} \sin(140 * 2 * \pi * t + \pi)$$ \hspace{1cm} (18)

These equations were used to create a 0.4 s signal sampled at 10,000 Hz. Two Gaussian disturbances were then applied. A narrow gaussian disturbance with a standard deviation of 0.005 was applied to the base signals at 0.1 s. A wide gaussian disturbance
with a standard deviation of 0.02 was applied to the base signals at 0.3 s. The created signals are shown in Figure 10.

![Figure 10](image)

**Figure 10** – Test signals used to demonstrate methods. One (a) is comprised of a single sine function. The other (b) is comprised of two sine functions.

### 3.1 Moving RMS

Using the knowledge reviewed in the background chapter of how the RMS of a sine wave relates to its amplitude, a method which relies on the moving RMS was developed and tested. From Equation 5, it was established that the RMS of a signal of infinite length will be $a/\sqrt{2}$, where $a$ is the signal’s amplitude. This serves as the basis for the first technique used to extract the sustained amplitudes from a vibration signal. The effect of the moving window size and the phase angle of the underlying signal were also evaluated and the importance in selecting the moving RMS window size was demonstrated. If the RMS window is excessively long, then short spikes can cause errors in determining sustained amplitudes. Consider the case when the RMS window length is equal to $t_{sustained}$. If the signal is near the sustained limit, it is easy for a single spike in the data to
take the RMS value above the limit. If the window is too short, a large amount of error is introduced into the signal, as demonstrated in Figure 3.

This illustrates the need to specify a minimum frequency $f_{\text{min}}$ that can be detected accurately using the RMS method. From Figure 3, it was determined that the window size should be twice the period of $f_{\text{min}}$ so that the calculated RMS over the moving window is within 5% of the RMS of a signal of infinite length. This relation is shown in Equation 19. Because $f_{\text{min}}$ and the RMS window size are inversely related, if $f_{\text{min}}$ is too small, then the resulting window size will be large and sensitive to small disturbances.

$$RMS_{\text{window}} = 2 \times \frac{1}{f_{\text{min}}}$$

(19)

When a moving RMS is taken of a signal, the influence of a single point is twice the length of the window size, since the point can occur at the beginning and end of the window. If $f_{\text{min}}$ is set to 50 Hz, $RMS_{\text{window}}$ will be 40 ms, which is equal to $2 \times t_{\text{sustained}}$. Because it is twice $t_{\text{sustained}}$, small disturbances will have an influence over multiple windows. This indicated that the moving RMS window is likely not well suited for lower frequency applications.

This was implemented in Python using the NumPy package. Details about NumPy can be found in [41], and the Python code used is provided in Appendix A.1.

An example of the RMS method applied to the two test signals is shown in Figure 11. From these initial results, it appears that the moving RMS works well on the signal

24
with a single frequency, but a large amount of error is present when the second frequency is introduced. This figure also illustrates how the effect of small disturbances is lengthened.

Figure 11 – Example implementation of the *Moving RMS* method bounding (a) a 70 Hz signal and (b) a composite 70 and 140 Hz signal, both with two gaussian disturbances.

Once the signal is bound by the *Moving RMS* method, the sustained amplitudes are determined from the bounded signal as if it were non-oscillating. The process of doing this is shared between a few of the methods, and the process by which it is executed is detailed in Section 3.7. Throughout the remainder of this work, this method will be referred to as the *Moving RMS* method.

### 3.2 STFT Peaks

When considering the time-frequency domain, a clear candidate for forming the envelope of the signal, and extracting the sustained maximum, is the amplitude of the Hilbert transform. As discussed in the background, however, this approach only works well for narrow banded signals. When dealing with a vibration signal with multiple harmonics of the base frequency, the raw signal will contain a broadband of frequencies, so the Hilbert transform is not expected to work well. When the example 70 Hz signal with two gaussian
disturbances was considered, the amplitude of the Hilbert transform does a good job forming the envelope around the signal. This can be seen in Figure 12a. If the signal was modified so that it is comprised of two frequencies at 70 Hz and 140 Hz, each with an amplitude of 0.5, then the Hilbert transform no longer forms the desired envelope. This is shown in Figure 12b. When envelope analysis is performed for bearing analysis the signal is band-pass filtered centered on the frequency of interest to avoid this issue [42]. Since this will reduce the maximum peaks in the signal it was not considered a viable solution in this work.

Because of this challenge with the Hilbert transform, the STFT was instead used to envelope the signal in the time-frequency domain. When multiple sinusoidal functions are present, the Fourier transform will return the amplitude and frequency for each of the sinusoidal functions. The amplitudes of each peak can thus be summed to get an idea of the peak amplitudes of the aggregate signal. One potential shortcoming of this method is that if many sinusoidal functions are present in the signal, the amplitude obtained by summing all the peaks in the Fourier transform may not be realized in the signal. To
illustrate this effect, consider the data presented in Figure 13 and Figure 14. Both signals were generated by summing two sine waves with amplitudes of 1.0 and 0.5 g. In Figure 13, the frequencies of the two sine waves are 800 and 1000 Hz, and the sum of their Fourier transform peaks (1.5 g) was realized in the signal.

![Figure 13](image1.png)

Figure 13 – A signal comprised of two sine waves with amplitudes 1.0 and 0.5 and frequencies 800 and 1000 Hz (a) raw signal and (b) Fourier transform.

In Figure 14, the frequencies of the two sine waves are 800 and 1600 Hz, and the sum of their Fourier transform peaks (1.5 g) was not realized in the signal. This indicates that this approach may not be appropriate for signals with harmonic frequencies present.
When using the STFT, it is important to appropriately select values for the window and hop sizes as well as the window function itself. In this work, the Hann window was used because it attenuates boundary effects that the rectangular window does not, and it offers compact support that the Gaussian window does not. For the sample signals used in this section, a window size of 256 samples is used with a hop size of 64 samples. These sample signals are sampled at 10,000 Hz, so these STFT parameters will offer a resolution of 0.0064 s, which was sufficient for detecting sustained periods longer than 0.02 s. Since the STFT is padded, the STFT peaks must remain above an amplitude for the full duration of $t_{sustained}$ before that amplitude is considered to be sustained. The final parameter to select is the threshold which constitute a real peak. Because noise will add small peaks to the Fourier transform, peaks which are less than 10% of the highest peak are discarded. This STFT Peaks method was applied to the two samples from the beginning of this chapter, and the result are shown in Figure 15.

Figure 14 – A signal comprised of two sine waves with amplitudes 1.0 and 0.5 and frequencies 800 and 1600 Hz (a) raw signal and (b) Fourier transform.
Figure 15 – Example implementation of STFT Peaks method bounding (a) a 70 Hz signal and (b) a composite 70 and 140 Hz signal, both with two gaussian disturbances.

This was implemented in Python using the NumPy and SciPy packages. Details about NumPy can be found in [41], SciPy information is available in [43], and the Python code used is provided in Appendix A.2.

Once the signal is bound by the STFT Peaks method, the sustained amplitudes are determined from the bounded signal as if it were non-oscillating. The process of doing this is shared between a few of the methods, and the process by which it is executed is detailed in Section 3.7. Throughout the remainder of this work, this method will be referred to as the STFT Peaks method.

3.3 Peak Envelope

Another method that forms an envelope around the data is based on the technique reviewed in the background section where a spline is fit to local maxima. In this work, the Python package SciPy was used to find the maxima in the data. The method works by comparing neighboring values [43]. To avoid high frequency noise, the local maxima are separated by a set number of samples. If the number of samples is too small, the desired
amount of demodulation may not be achieved. If the number of samples is too large, then the result will be too smoothed out. Once the peaks are found, SciPy is used to fit a univariate spline to the local maxima [43]. The result of applying this technique to the two test signals is shown in Figure 16. For reference, the local maxima that the spline is fit to are shown in Figure 16. To avoid high frequency noise in the envelope, the local maxima used were separated by at least $\frac{t_{sustained}}{2}$.

Figure 16 – Example implementation of Peak Envelope method bounding (a) a 70 Hz signal and (b) a composite 70 and 140 Hz signal, both with two gaussian disturbances.

Once the signal is bound by the Peak Envelope method, the sustained amplitudes are determined from the bounded signal as if it were non-oscillating. The process of doing this is shared between a few of the methods, and the process by which it is executed is detailed in Section 3.7. Throughout the remainder of this work, this method will be referred to as the Peak Envelope method.

### 3.4 LP Rectifier Envelope

The last method that forms an envelope around the signal is based on the bandwidth reduction approach used by Jeong et al [24]. In its implementation in this work, the absolute
value of the signal was used as the starting envelope. The results of applying this method to the two example signals used in this section are shown in Figure 17. It is important to note that the parameters of the bandwidth reduction must be carefully selected. For the examples presented here, the first low pass (LP) filter had a cutoff of 10 Hz and order five. The second low pass filter had a cutoff of 200 Hz and order five. The process was completed twice to ensure that the envelope fully bounds the input signal.

Figure 17 – Example implementation of **LP Rectifier Envelope** method bounding (a) a 70 Hz signal and (b) a composite 70 and 140 Hz signal, both with two gaussian disturbances.

Once the signal is bound by the **LP Rectifier Envelope** method, the sustained amplitudes are determined from the bounded signal as if it were non-oscillating. The process of doing this is shared between a few of the methods, and the process by which it is executed is detailed in Section 3.7. Throughout the remainder of this work, this method will be referred to as the **LP Rectifier Envelope** method.

### 3.5 Peak Distances

All previously discussed methods have involved bounding or forming an envelope around the signal and determining the maximum sustained amplitude from the envelope.
The following methods consider the raw signal directly. In the absence of noise, and at a sufficient sampling rate, the distances between peaks can be used to find the maximum sustained amplitudes. To execute this method, the peaks must first be located. The list is then sorted according to descending amplitude and iterated through, beginning with the second list item (second highest peak). The time between the peak under investigation and all peaks higher on the list, which will have higher amplitudes than the peak under investigation, are calculated. If the time is greater than $t_{sustained}$, then a sustained amplitude at the level of the peak under investigation is said to be present. Pseudo-code representing the execution of this method is shown in Figure 18.

Figure 18 – Algorithm for implementing Peak Distances method.

One shortcoming of this method is that only two peaks need to be present, and this makes it susceptible to noise. It also does not account for multiple disturbances that may exist within a segment of signal under analysis. For example, when the same 70 Hz signal
with two gaussian disturbances used in Figure 11 was applied here, the sustained amplitude was shown to span the two disturbances in Figure 19a, but the method was successful when a single disturbance is present as shown in Figure 19b. For lower sampling rates, it is also likely that peaks will be missed. If this happens at the sustained amplitude peak, then reported sustained amplitude will be an under estimation.

![Figure 19](image.png)

Figure 19 – Example implementation of Peak Distances method applied to a test signal with (a) two gaussian disturbances and (b) a single gaussian disturbance.

### 3.6 Expire Timer

The final method developed and tested in this work resembles the Peak Distances method but incorporates an expire timer, so that if a long gap exists between two amplitudes above a specified threshold, it will not be registered as a sustained amplitude. It will be seen shortly that this resolved the issue seen in Figure 19a. Similar to the Moving RMS method, the length of the expire timer \( t_{\text{expire}} \) will be set by the desired minimum detectable frequency \( f_{\text{min}} \). The \( t_{\text{expire}} \) must be at least the length of time between peaks in \( f_{\text{min}} \). Because the absolute value of the signal is analyzed, two peaks will occur in each period, the relation between \( f_{\text{min}} \) and \( t_{\text{expire}} \) is given in Equation 20.
\[ t_{\text{expire}} = \frac{1}{2 * f_{\text{min}}} \]  

In the Moving RMS section, \( f_{\text{min}} \) was chosen to be 50 Hz, which made the window length \( 2 * t_{sustained} \). If the same minimum frequency is used for the expire timer, \( t_{\text{expire}} \) will be 0.01 ms or half of \( t_{sustained} \). Once the length of the expire timer is set, the maximum sustained amplitude is determined by closing in on it in an iterative way. The value is closed in on by storing two values: \( \text{lastSustained} \) which stores the highest amplitude that has been determined to reach the sustained time criteria, and \( \text{lastTooHigh} \) which stores the lowest amplitude where that was considered not to be a sustained amplitude. The maximum of the signal used to initialize \( \text{lastTooHigh} \) and \( \text{lastSustained} \) is initialized to zero. In each iteration, the average of \( \text{lastTooHigh} \) and \( \text{lastSustained} \) is tested. If it is determined to be a sustained amplitude, then \( \text{lastSustained} \) is updated. If not, \( \text{lastTooHigh} \) is updated. The process can continue until either a specified number of iterations are completed or until the change in the test value between iterations is below a threshold.

To determine if the amplitude under consideration is sustained or not, the value is used to threshold the rectified signal. The result is thus a signal that is one if it is at or above the test amplitude, or zero if it is below. An example of this applied to the test signal at an amplitude of 1.1 g is shown in Figure 20a and b.
Figure 20 – (a) Test 70 Hz signal with two gaussian disturbances, (b) the result of the first step of the *Expire Timer* method applied to the test signal, and (c) an illustration of the two timers used to test the signal.

At this point, the signal in Figure 20b resembles that of the spike trains and text data mining reviewed in Chapter 2. Because the application differs, this work takes a different approach to determine if a sustained amplitude exists in this signal. From this thresholded signal, it is determined if the test amplitude is sustained or not by scanning each data point. If the value is zero, it moves to the next point. If the value is one, two timers are created. One timer is counting to determine if $t_{sustained}$ is reached and is referred to as the sustained timer. The second timer is counting to determine if $t_{expire}$ is reached and is referred to as the expire timer. The expire timer is reset each time a value of one is seen. The sustained timer is reset only when the expire timer reaches its limit. The functions
of these two timers are illustrated in Figure 20c. At the start of the first Gaussian disturbance, the sustained timer is started. For each nonzero value, the expire timer is reset as the signal is scanned. When the portion of the signal that no longer crosses the threshold is reached, the expire timer will run out, and the sustained timer will be reset. When the signal passes the threshold again at the start of the second disturbance, the sustained timer is reset. This time, the sustained timer is able to complete without the expire timer canceling it, so a sustained amplitude at the test amplitude of 1.1 g is determined to be present. Pseudo-code representing the execution of this method is shown in Figure 21.
Given : x[1..n], t[1..n], replicates, sustainedTimerLimit, expireTimerLimit
Return: sustained_amplitude

//set initial values
lastTooHigh = MAX(x);
lastSustained = 0;

FOR i = 1 TO replicates
  sustained = false;
  RESET expireTimer;
  RESET sustainedTimer;

  testValue = AVERAGE lastTooHigh AND lastSustained;
  thresholded = THRESHOLD x AT testValue;
  FOR j = 1 TO n
    IF thresholded[j] = 1
      RESET expireTimer;
      IF sustainedTimer NOT started
        START sustainedTimer;
      ENDIF
      IF sustainedTimer > sustainedTimerLimit
        sustained = true;
      ENDIF
    ENDIF
  ENDFOR
  IF expireTimer > expireTimerLimit
    RESET sustainedTimer;
  ENDIF
ENDFOR
IF sustained
  lastSustained = testValue;
ELSE
  lastTooHigh = testValue;
ENDIF
ENDFOR

sustained_amplitude = lastSustained;

Figure 21 – Algorithm for the implementation of the Expire Timer method.

3.7 Peak Height at Width

For the Moving RMS, STFT Peaks, Peak Envelope, and LP Rectifier Envelope methods, it is necessary to analyze the envelope and determine the maximum amplitude where the amplitude persists for the critical threshold of \( t_{sustained} \). In this work, this is
completed by iteratively calling SciPy’s peak_width function [43]. The peak_widths function returns the width and height, where the width occurs given a relative height. The relative height used is updated and retested until the returned width is within two percent of \( t_{sustained} \). If no peaks exist in the envelope signal, which corresponds to a signal with no disturbances, the average of the envelope will be returned as the maximum sustained amplitude.

3.8 Summary of Signal Processing Methods

In this chapter, six techniques for determining the maximum sustained amplitudes from vibration data were formalized using the knowledge reviewed in Chapter 2. Of these six techniques, four of them envelope the signal and use the envelope to determine the sustained maximum, while the remaining two deal with the raw signal directly. These six methods answer the first research question: “How can maximum sustained amplitudes in vibration data be detected?” Now that they have been formalized, Chapter 4 will detail how these methods will be tested to answer the remaining research questions about how these methods perform under different circumstances.
CHAPTER 4. METHODOLOGY

To test the performance of the methods under different sampling rates and noise levels, a simulated signal was used. Testing a simulated signal that was comprised of a sum of cosine waves allowed for the ground truth to be determined. The simulated signal was based on experimental vibration data collected on the outside of a 3-axis computer numeric control (CNC) machine.

4.1 Experimental Data

The experimental data that the simulated data was based on was collected on an EMCO E350 3-axis CNC machine. Vibration data was collected using an Analog Devices ADXL203EB-ND accelerometer. The accelerometer was connected to a voltage measurement data acquisition (DAQ) hardware attached on top (HAT) of a Raspberry Pi 3B+. The DAQ HAT was part number MCC118 from Measurement Computing Corporation. The purpose of this data was to get a rough approximation of how the vibrations move through the machine, so the accelerometer was mounted on the outside of the housing above the spindle. The accelerometer placement is shown in Figure 22a, and the box used to collect the data from the sensor is shown in Figure 22b. The data collection parameters for the sensor and machining process are shown in Table 2.
Figure 22 – Experimental set-up with (a) accelerometer attached to EMCO where arrow indicates sensitivity direction, and (b) the hardware used to collect the data.

Table 2 – Experimental data collection parameters.

<table>
<thead>
<tr>
<th>Category</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machining</td>
<td>Material</td>
<td>D2 Tool Steel Annealed</td>
</tr>
<tr>
<td></td>
<td>Spindle Speed</td>
<td>4,220 RPM</td>
</tr>
<tr>
<td></td>
<td>Tool</td>
<td>Kennametal KICR ½” Single Insert</td>
</tr>
<tr>
<td></td>
<td>Insert</td>
<td>Kennametal SDEB 2615</td>
</tr>
<tr>
<td>Sensor</td>
<td>Sampling Rate</td>
<td>10,000 Hz</td>
</tr>
</tbody>
</table>

4.2 Simulated Data

To get a simple simulated signal to run the tests on, the Fourier transform was performed on the experimental data. The result of the Fourier transform is shown in Figure 23. In it there are two dominating peaks that fall at six times and twenty-three times the tooth pass frequency. The frequencies of these peaks were used with the tooth pass frequency to create the simulated signal $s(t)$ shown in Equation 21. The phases ($\varphi_1$ and $\varphi_2$) assigned to the harmonics are generated using a random number generator.
Figure 23 – Fourier transform of the experimental test data.

\[ s(t) = \cos(70 \times 2\pi \times t) + \cos(6 \times 70 \times 2\pi \times t + \varphi_1) \]
\[ + \cos(23 \times 70 \times 2\pi \times t + \varphi_2) \]

Figure 24 – Simulated signal used to test the signal processing methods.
4.3 Signal Disturbances

In addition to stationary signals, like the one created by Equation 21, the signal processing methods were tested with disturbances. The disturbances tested are modeled after a higher-order Gaussian. The equation for the higher-order Gaussian is given in Equation 22, where the parameter $a$ is the height, $b$ is the center location, $\sigma$ is the standard deviation, and $n$ is the power. When $n$ is one, the equation becomes that of a standard Gaussian [44]. When $n$ is greater than one, the Gaussian becomes steeper and flatter. The time series for each of the equations was increased by one so the original signal can be multiplied by them, as shown in Equation 23, to produce the signal with the disturbance.

$$g(t) = 1 + a \cdot e^{- \left( \frac{(t-b)^2}{2\sigma^2} \right)^n}$$

(22)

$$s(t) = s(t) \cdot g(t)$$

(23)

Various values for the Gaussian parameters were tested to introduce variability in the shape of the disturbance. The Gaussian parameters that were tested are given in Table 3. All combinations of these parameters were tested, so in total, four different burst types were tested with the stationary signal for a total of five different base signals. To illustrate each of these, each of the five different disturbances were applied to a stationary signal and shown in Figure 25. Subjecting the base signals to different disturbances and comparing their performance to the stationary signal with no disturbance answered the third research question: “How does the performance of methods available for determining maximum sustained amplitude change for non-stationary signals?”
Table 3 – Gaussian parameters for introduced disturbance.

<table>
<thead>
<tr>
<th>Disturbance Parameter</th>
<th>Values Tested</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power ( (n) )</td>
<td>1, 3</td>
</tr>
<tr>
<td>Height ( (a) )</td>
<td>0.5</td>
</tr>
<tr>
<td>Standard Deviation ( \sigma )</td>
<td>0.005, 0.01</td>
</tr>
</tbody>
</table>

Figure 25 – Disturbance types applied to a (a) stationary, 800Hz signal. (c) \( n = 1, a = 0.5, \sigma = 0.005 \), (c) \( n = 1, a = 0.5, \sigma = 0.01 \), (c) \( n = 3, a = 0.5, \sigma = 0.005 \), (e) \( n = 3, a = 0.5, \sigma = 0.01 \).

4.4 Ground Truth

To evaluate the performance of each of the algorithms, the ground truth was extracted from the simulated signals. The maximum sustained amplitude of the simulated signal was the amplitude at which two or more peaks reach or surpass the amplitude and were separated by at least \( t_{\text{sustained}} \). The true peaks of the simulated signal were found by setting the first derivative of the signal to zero and solving for values of \( t \) in the interval under investigation. This is shown in Equation 24. Once the peaks were determined, they were sorted and iterated through to find the highest amplitude where the condition above
exists. Pseudo-code showing how the peak data was searched through is given in Figure 26.

\[
\frac{ds}{dt} = 0 = a_1 \cdot f_1 \cdot 2\pi \cdot \cos(f_1 \cdot 2\pi \cdot t_1 + \varphi_1) + \cdots + a_n \cdot f_n \cdot 2\pi \cdot \cos(f_n \cdot 2\pi \cdot t_n + \varphi_n) \quad \text{for } t \in [0,0.4] \tag{24}
\]

Given: peaks[1..n], t[1..n], time_threshold
Return: sustained_amplitude

//sort amplitudes and times as single matrix
peaks2 = SORT([peaks; t]);

FOR i = 2 TO n
    //check time to preceding peaks
    FOR j = 1 TO i-1
        time = ABSOLUTE(peaks2[i,2] - peaks2[j,2])
        IF time >= time_threshold
            sustained_amplitude = peaks2[i,1];
            BREAK  //exit
        ENDIF
    ENDFOR
ENDFOR

Figure 26 – Algorithm for iterating through peak data to find ground truth value.

This method for extracting the ground truth value is similar to the Peak Distances signal processing method, but the key difference is that when used on the ground truth signal, all peak values are true values. Because they are true peaks, a sustained amplitude can be accurately determined to exist after just seeing two values at that level. When using the approach as a signal processing approach, peaks can be missed due to aliasing, or false peaks may be present due to the presence of noise.
4.5 Signal Processing Parameters

All the signal processing methods detailed in Chapter 3 were tested in this work.

The parameters used to set-up the methods are summarized in Table 4 below.

Table 4 – Summary of parameters used in each signal processing method.

<table>
<thead>
<tr>
<th>Signal Processing Method</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>All methods</td>
<td>Sustained time</td>
<td>20 ms</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>RMS window width</td>
<td>40 ms</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>Window function</td>
<td>Hann</td>
</tr>
<tr>
<td></td>
<td>Window width</td>
<td>25.6 ms</td>
</tr>
<tr>
<td></td>
<td>Hop size</td>
<td>6.4 ms</td>
</tr>
<tr>
<td></td>
<td>Peak Threshold</td>
<td>10% of highest peak</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>Minimum distance between peaks</td>
<td>10 ms</td>
</tr>
<tr>
<td></td>
<td>LPF1 Cut-off</td>
<td>10 Hz</td>
</tr>
<tr>
<td></td>
<td>LPF1 Order</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>LPF2 Cut-off</td>
<td>200 Hz</td>
</tr>
<tr>
<td></td>
<td>LPF2 Order</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Replicates</td>
<td>2</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>No parameters</td>
<td></td>
</tr>
<tr>
<td>Expire Timer</td>
<td>Expire timer duration</td>
<td>2 ms</td>
</tr>
<tr>
<td></td>
<td>Replicates</td>
<td>5</td>
</tr>
</tbody>
</table>

4.6 Design of Experiment

The test signals created using the equation for the simulated data and the disturbances detailed above are subject to various levels of Gaussian white noise and sampling rates, and their performance was compared to the ground truth value. The values for the sampling rates and signal-to-noise ratios (SNR) tested are summarized in Table 5.

The SNR is defined as the ratio of the power of the signal to the power of the noise.
Table 5 – Design of experiment.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Rate [Hz]</td>
<td>100,000</td>
</tr>
<tr>
<td></td>
<td>10,000</td>
</tr>
<tr>
<td></td>
<td>7,500</td>
</tr>
<tr>
<td></td>
<td>5,000</td>
</tr>
<tr>
<td>Signal to Noise Ratio (SNR)</td>
<td>Infinite</td>
</tr>
<tr>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

The results from testing each of the methods at these varying sampling rates and signal to noise ratios answered the second research question: “What is the effect of sampling rate and signal-to-noise ratio on methods available for determining maximum sustained amplitude?” The values for the sampling rate are based on the frequencies used to generate the data. The minimum sampling rate (5 kHz) was chosen so that its Nyquist frequency (2.5 kHz) captures the highest frequency component in the simulated signals are based on. The 100 kHz sampling rate was included as an ideal case and was only used when no noise was present. If large amounts of error exist under this test case, it can be concluded that the method was not well-formed. The remaining sampling rates and SNRs are tested by full factorial design, so all possible combinations were tested. This results in 13 test conditions for the 9 base signals. A table of all tested signal parameters is included in Appendix B.

For each combination tested, the percent error of each method relative to the ground truth was recorded to track their accuracy. The computation time that each method required was also tracked. Recording the time, in addition to comparing the number of method parameters, answered the fourth research question: “How feasible are the methods
available for determining maximum sustained amplitude for implementation on an edge device?”
CHAPTER 5. RESULTS

To understand how each of the tested variables independently affect the accuracy of the signal processing methods, the results are broken up into three sections. First, the results are presented for the test cases where no disturbances are present. Next, the average result across all test disturbance cases is presented. Finally, the computational time required to run each method is compared. Tables detailing all results are provided in Appendix C.

5.1 Stationary Signals

5.1.1 Sampling Rate

To isolate the variables under test, the effect of the sampling rate and signal-to-noise ratio are considered separately. First the effect of the sampling rate is investigated by comparing results when no noise is present in the signal. These results are shown in Table 6 and represented graphically in Figure 27. Because the Moving RMS result has a high bias, it was not included in Figure 27. The highest sampling rate is also not included in the plot, because it was meant to serve as a reference and would skew the scale of the x-axis.

Table 6 – Percent error of each method while varying sampling rates on a stationary signal with no noise.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling Rate [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100,000</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-42.1%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>-9.2%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>-1.1%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>-5.7%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.0%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-1.9%</td>
</tr>
</tbody>
</table>
5.1.2 Signal-to-Noise Ratio

Next, the effect of the SNR is considered. The full results are shown in Table 7 and represented graphically in Figure 28. The Moving RMS is again omitted from the plot because of its high bias. The case with no noise is also not plotted because it leads to an infinite SNR.

Table 7 – Percent error of each method while varying SNR on a stationary signal with a 10 kHz sampling rate.

<table>
<thead>
<tr>
<th>Signal-to-Noise Ratio</th>
<th>Infinite</th>
<th>100</th>
<th>10</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving RMS</td>
<td>-42.1%</td>
<td>-41.6%</td>
<td>-38.7%</td>
<td>-35.9%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>-4.0%</td>
<td>-1.9%</td>
<td>22.9%</td>
<td>61.4%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>-2.0%</td>
<td>-0.7%</td>
<td>14.4%</td>
<td>21.1%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>-5.8%</td>
<td>-2.9%</td>
<td>9.4%</td>
<td>18.8%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.0%</td>
<td>5.5%</td>
<td>25.4%</td>
<td>39.3%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-2.1%</td>
<td>2.3%</td>
<td>13.2%</td>
<td>21.7%</td>
</tr>
</tbody>
</table>
Figure 28 – Comparison of the performance of each method on a stationary signal with varying SNR.

5.2 Non-Stationary Signals

5.2.1 Sampling Rate

Next, the average performance across the stationary signal and the four test signals with disturbances are considered. The average values are given in Table 8 and graphically represented in Figure 29. The standard deviations are reported in Table 9. The Moving RMS and 100 kHz sampling rate were not plotted as to not skew the graph. To understand the variation in performance of each method, each method is broken out in Figure 30 with error bars. The error bars represent the range of error across the five test cases.
Table 8 – Average percent error of each method across multiple disturbance types while varying sampling rates with no noise.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling Rate [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100,000</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-43.0%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>-0.6%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>3.7%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>-1.1%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.0%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-0.5%</td>
</tr>
</tbody>
</table>

Table 9 – Standard deviation of each method across multiple disturbance types while varying sampling rates with no noise.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling Rate [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100,000</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>0.039</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>0.078</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>0.058</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>0.071</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.000</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>0.008</td>
</tr>
</tbody>
</table>

Figure 29 – Comparison of the average performance of each method across multiple disturbance types with varying sampling rates and no noise.
5.2.2 Signal-to-Noise Ratio

Next, the average performance across the stationary signal and the four test signals with disturbances are considered. The values are given in Table 10 and graphically represented in Figure 31. The standard deviations are reported in Table 11. The Moving RMS and no noise were not plotted as to not skew the graph. To understand the variation in performance of each method, each method is broken out in Figure 32 with error bars. The error bars represent the range of error across the five test cases.
Table 10 – Average percent error of each method across multiple disturbance types while varying SNR.

<table>
<thead>
<tr>
<th>Method</th>
<th>Signal-to-Noise Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Infinite</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-43.0%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>5.4%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>2.2%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>-1.1%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>-0.3%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-0.9%</td>
</tr>
</tbody>
</table>

Table 11 – Standard deviation of each method across multiple disturbance types while varying SNR with a 10 kHz sampling rate.

<table>
<thead>
<tr>
<th>Method</th>
<th>Signal to Noise Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Infinite</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>0.039</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>0.049</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>0.062</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>0.072</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.003</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>0.007</td>
</tr>
</tbody>
</table>

Figure 31 – Comparison of the average performance of each method across multiple disturbance types with varying SNR.
Figure 32 – Detailed result of the average performance of each method across multiple disturbance types with varying SNR. Error bars represent the range in performance across the different disturbance types. (a) **STFT Peaks**, (b) **Peak Envelope**, (c) **LP Rectifier Envelope**, (d) **Peak Distances**, and (e) **Expire Timer**.

5.2.3 Disturbance Characteristics

To understand which disturbances are the most difficult to classify, the results of the disturbances are considered. In total, five disturbance cases were evaluated. Case a is the stationary signal, case b is a narrow Gaussian, case c is a wide Gaussian, case d is a narrow third order Gaussian, and case e is a wide third order Gaussian. Table 12 presents the error produced by each method with 7.5 kHz sampling rate and no noise. Table 13 presents the error produced by each method with 7.5 kHz sampling rate and a SNR of 10.
Table 12 – Error in the presence of different disturbances with a 7.5 kHz sampling rate and no noise.

<table>
<thead>
<tr>
<th>Disturbance</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving RMS</td>
<td>-42.1%</td>
<td>-41.8%</td>
<td>-49.8%</td>
<td>-41.2%</td>
<td>-40.5%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>-5.2%</td>
<td>10.6%</td>
<td>5.9%</td>
<td>7.6%</td>
<td>8.5%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>-2.7%</td>
<td>-3.8%</td>
<td>6.4%</td>
<td>-2.7%</td>
<td>11.2%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>-5.7%</td>
<td>-6.7%</td>
<td>3.4%</td>
<td>-5.7%</td>
<td>9.4%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.0%</td>
<td>-0.6%</td>
<td>-0.5%</td>
<td>0.0%</td>
<td>-0.5%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-2.5%</td>
<td>-0.7%</td>
<td>-0.6%</td>
<td>-0.7%</td>
<td>-0.5%</td>
</tr>
<tr>
<td>RMS (excluding moving RMS)</td>
<td>3.4%</td>
<td>4.6%</td>
<td>4.5%</td>
<td>4.5%</td>
<td>7.9%</td>
</tr>
</tbody>
</table>

Table 13 – Error in the presence of different disturbances with a 7.5 kHz sampling rate and a SNR of 10.

<table>
<thead>
<tr>
<th>Disturbance</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving RMS</td>
<td>-39.3%</td>
<td>-38.9%</td>
<td>-37.4%</td>
<td>-38.2%</td>
<td>-36.6%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>31.5%</td>
<td>45.8%</td>
<td>33.5%</td>
<td>47.2%</td>
<td>37.3%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>2.1%</td>
<td>1.0%</td>
<td>-14.2%</td>
<td>2.1%</td>
<td>-14.5%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>1.5%</td>
<td>0.5%</td>
<td>3.1%</td>
<td>1.5%</td>
<td>3.1%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>23.4%</td>
<td>22.7%</td>
<td>8.5%</td>
<td>24.0%</td>
<td>14.7%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>2.7%</td>
<td>5.3%</td>
<td>8.0%</td>
<td>2.8%</td>
<td>14.5%</td>
</tr>
<tr>
<td>RMS (excluding moving RMS)</td>
<td>18.1%</td>
<td>22.8%</td>
<td>14.4%</td>
<td>20.4%</td>
<td>20.6%</td>
</tr>
</tbody>
</table>

5.3 Computation Time

To compare the time performance, each method is executed 1,000 times, and the average time for each method is saved. This was repeated three times and the best result is reported in Table 14. This process was repeated for two sampling rates, 10,000 Hz and 100,000 Hz. A 0.1 second signal was analysed so the methods were executed on 1,000 and 10,000 samples. This was done to give an indication of how the time to execute the method
scales with sampling rate. These results were produced on a laptop with an eighth generation Intel i5 chip.

Table 14 – Timing results for each of the signal processing methods in milliseconds.

<table>
<thead>
<tr>
<th>Method</th>
<th>Samples</th>
<th></th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1,000</td>
<td>10,000</td>
<td></td>
</tr>
<tr>
<td>Moving RMS</td>
<td>0.19 ms</td>
<td>0.67 ms</td>
<td>3.5</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>2.27</td>
<td>9.96</td>
<td>4.4</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>0.44</td>
<td>1.46</td>
<td>3.4</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>98.7</td>
<td>812</td>
<td>8.2</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.13</td>
<td>0.33</td>
<td>2.5</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>2.52</td>
<td>25.1</td>
<td>9.9</td>
</tr>
</tbody>
</table>
CHAPTER 6. DISCUSSION

The first result to consider is the performance of the RMS Envelope method. Because of the large amount of error across all test cases, it is clear that the relationship between the RMS and the peaks in a single sine wave does not hold for a signal that is comprised of a base frequency and harmonics of it. To understand why, calculations for the RMS value of a signal with two and three sine waves are included in Appendix D. These calculations show that the relationship of the RMS value and the peak value changes depending on the number of sine waves comprising the signal. This could be corrected for in the test signal with three sine waves, but the result could not be applied generally to other signals. For this reason, discussion of the RMS Envelope method is largely omitted from the remainder of this chapter.

6.1 Stationary Signals

When it came to the performance on a stationary signal, the sampling rate was found in Figure 27 to have the greatest effect on the Expire Timer and Peak Envelope. For these two methods, reducing the sampling rate led to under estimations. This is expected, as reducing the sampling rate makes it more likely for peaks to be missed. The other methods were less sensitive to changes in the sampling rate. When evaluating the effect of noise, in Figure 28, the STFT Peaks method was found to have the greatest sensitivity followed by the Peak Distances method. The effect was reduced on the other methods, but all methods saw that increasing the noise in the signal led to over estimations. This is expected since noise increases the amplitude of the input signal. The presence of noise adds
peaks into the Fourier transform, so it was expected that the STFT Peaks method would lead to larger amounts of error in the presence of noise.

6.2 Non-Stationary Signals

After disturbances were added to the signals, Figure 30 showed that the LP RectifierEnvelope was found to have the largest range in performance across all sampling rates. It was also seen that the STFT Peaks had the greatest amount of error, and the Expire Timer and the greatest decrease in performance as the sampling rate was reduced. With the sampling rate constant and varying SNR, Figure 32 showed that the performance of all methods under varying noise levels and in the presence of disturbances, followed similar trends to when disturbances were not present. When looking at the performance on the 7.5 kHz signal without noise in Table 12, the effect of the disturbance was seen to be minimal. Disturbance e, which has a sharpest rise and wide plateau, had the highest error on average. This is expected, since the sharp rise and fall are more difficult to envelope, and the envelope methods had greater error than the non-envelope methods. When looking at the effect of the disturbances on the 7.5 kHz signal with a SNR of 10 in Table 13, no disturbance case was significantly worse than the others, and case c performed better than the average. The addition of noise helps reduce the effect of the steep rise so case e is no longer the worst case. This additional noise also increases the predicted output of case c, which was initially underestimated, ultimately making it more accurate.

One effect that this work did not address is how the presence of noise and a disturbance can cause a method to predict the wrong location of the sustained maximum. This effect is illustrated in Figure 33, which shows the sustained peak that the Peak
Distances method found in the signal. In Figure 33a, when the disturbance is narrow, there is a large amount of error because the algorithm detects the sustained amplitude from noise outside the disturbance and the middle of the disturbance. This occurs to a lesser degree in Figure 33b when the disturbance is wide.

![Figure 33](image)

**Figure 33 – Peak Distances** method result with 10 kHz sampling rate, SNR of 5 and (a) disturbance case b and (b) disturbance case c.

It should be noted that just because a method does worse with noise does not mean it is not a viable solution in applications where noise is expected. Instead, filtering should be used to eliminate the noise and drive the signal processing methods performance closer to their no-noise results.

### 6.3 Computation Time

In the timing experiment, it was seen that when the methods were fed 1,000 samples representing 100 ms of data, all the methods were able to complete the computation within 100 ms, allowing it to fully complete before a new batch of data becomes available when processing in real time. The *LP Rectifier Envelope*, however, barely met this requirement and failed to process 10,000 samples representing 100 ms of data sampled at 100 kHz. For
this reason, if processing data in real time, a method other than the LP Rectifier Envelope should be considered. While the time performance of the remaining methods allows them all to operate on streaming data, some of them require batches of data, while others can truly operate on data streams. The envelope methods all require a batch processing method, since the envelope is first formed, then processed to determine the sustained maximums. The non-envelope methods are split. The Peak Distances method requires the collection of peaks in the signal to be saved and processed. The peaks can be collected in real time, but they must be looped back upon and analysed. In this work, the Expire Timer method iteratively ran on the data to close in on the maximum sustained amplitude. The method could be implemented in a way that only looks at the data once if it only needed to check a single amplitude. This can allow it to run efficiently on a stream of incoming data.

When looking at how the time performance scaled with a ten times increase in the input data, the LP Rectifier Envelope and Expire Timer both saw a roughly ten times increase in computation time, while the other methods all saw between a three and four times increase. While this is not as important for processing streaming data, it is important to consider if processing large amounts of historical data. In this context, methods other than the LP Rectifier Envelope and Expire Timer should be considered in order to reduce the total computational time.

6.4 Parameters

When considering how generalizable each of these methods are to other applications, it is important to consider the number of tuning parameters for each method. The number of parameters in each method are summarized in Table 15. Many of these
parameters relate to the $t_{\text{expire}}$, but it is still important to consider that they may need to be adjusted to obtain the optimal results for various applications. From this, it is seen that most of the methods require one to two parameters, with the exception of the \textit{STFT Peaks} method, \textit{LP Rectifier Envelope} method, and \textit{Peak Distances} method. One of the parameters for the STFT is the window function, which will likely not change with application, and another, the hop size, it tied to the window length, so it will be considered to have the same number of parameters as the others. This leaves the \textit{LP Rectifier Envelope} method as the only method that requires significantly more parameters; this indicates that it has the least generalizability. The \textit{Peak Distances} method also stands out, since it has no tuning parameters; this indicates that it is the most generalizable.

Table 15 – Number of tuning parameters for each signal processing method.

<table>
<thead>
<tr>
<th>Signal Processing Method</th>
<th>Number of Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving RMS</td>
<td>1</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>4</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>1</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>5</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>2</td>
</tr>
</tbody>
</table>

In the development of the \textit{Moving RMS}, \textit{STFT Peaks}, and \textit{Expire Timer} methods, there is a mathematical basis for a minimum detectable frequency that the methods will not be expected to perform reliably below. This is important to consider if they are being applied to a broadband signal, where low frequencies are expected. In this work, the parameters used lead to the minimum detectable frequencies shown in.
Table 16 – Minimum detectable frequencies for the tuning parameters used in this work.

<table>
<thead>
<tr>
<th>Signal Processing Method</th>
<th>Minimum Detectable Frequency [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving RMS</td>
<td>50</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>39</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>50</td>
</tr>
</tbody>
</table>

6.5 Summary

A summary of performance of each method across all the test conditions is given in Table 17; this table gives a granular view of how each method generally performs. For the stationary test cases, the methods are evaluated by the magnitude of the error they produce. If the magnitude of the error is less than five percent, it is considered acceptable; if it is greater than ten percent, it is considered unacceptable. Between five and ten percent, the magnitude of the error is considered borderline. For the non-stationary test cases, the standard deviation is also considered so that methods which produce large amounts of variation across the different disturbances are penalized. Standard deviations below three percent are considered unacceptable, and standard deviations above six percent are considered unacceptable. The limits for the magnitude of the error and the standard deviation are chosen from the results so that variation is seen. This high-level summary indicates that the Peak Envelope generally produces the least error. The Peak Envelope methods, however, had the most error at low sampling rates in stationary signals, so if this is the kind of signal that is expected, other methods should be considered.
Table 17 – Summary of the performance across all test cases.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling Rate [kHz]</th>
<th>SNR</th>
<th>Sampling Rate [kHz]</th>
<th>SNR</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving RMS</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>o</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>+</td>
<td>+</td>
<td>o</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>+</td>
<td>+</td>
<td>o</td>
<td>+</td>
<td>o</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>+</td>
<td>+</td>
<td>o</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

Key:
- $|\text{error}| < 7\%$ and $\sigma < 3\%$
- $|\text{error}| > 15\%$ or $\sigma > 6\%$
- $|\text{error}| < 15\%$ and $\sigma < 6\%$
CHAPTER 7. CONTRIBUTIONS

This thesis contributes an understanding of how to determine the sustained amplitudes in vibration data. This contribution is realized by considering a novel application for vibration envelope techniques in extracting the sustained maximums from a vibration signal. Additionally, two novel techniques that process the signal directly were formalized. The performance of the envelope techniques in this novel application, as well as the performance of the proposed methods, are quantified by testing them on signals with varying sampling rate and signal-to-noise ratio. Additionally, this thesis provides insights into which methods are best suited for real-time applications by comparing the computational time and the number of hyper parameters.
CHAPTER 8. FUTURE WORK

This work developed methods based on a theoretical framework and tested them on a simulated signal. Future work includes repeating the tests on experimental data collected at a linear glass scale reader head. In addition to learning how well the methods perform on an application, this test will also reveal how universal the tuned parameters are.

To improve the accuracy, future work should consider penalizing methods that do not accurately locate where the sustained maximums occur in time. This suggestion comes after seeing how some methods accurately return the amplitude in the case where a disturbance and noise are present, but they do so by drawing from the disturbance and noise outside the signal. Penalizing this can give a more complete evaluation of the performance of each method.

Future work can further optimize the time performance of each of the methods. For example, the function used to process the enveloped signals can be adapted to use Haar wavelets, as reviewed in the background section [35]. This would improve the time performance of the envelope methods. For the Expire Timer, the method of iteratively closing in could be optimized for time performance. In its current implementation, this method moves linearly, but quicker movements could reduce the number of iterations that it takes to close in on a number.

The methods tested in this work all looked at what had occurred. Future work can extend this to predict how the signal is trending. This can be done by looking at how the reported values are trending, or by taking the kurtosis of the signal and looking at where
the allowable limit falls on the tail of the distribution. These predictions may be sensitive
to short bursts, but it can be useful if the data is trending toward the allowable limit.
CHAPTER 9. LIMITATIONS

One limitation of this work is that the methods were tested on simulated data rather than experimental data. Because of this, there is increased potential for error when tested on a more complex experimental signal. Future work will extend the experiments in this thesis to include experimental data to address this.

In the simulated data used to test the methods, the signal was comprised of a base tooth pass frequency and the harmonics of it as it is transmitted through the machine. This makes for a broadband signal, and the methods were not tested on a narrowband signal. If these methods are being applied to a narrowband application, further testing should be done before implementing.

Finally, the disturbances tested in this work are modeled after a Gaussian and third order Gaussian. This assumes that the disturbances seen in application will be similar to this. If the actual disturbances are significantly different, than the results may vary. Additionally, in the non-stationary signals tested, the frequencies present were not varied with time. Instead only the amplitude changes made it non-stationary.
CHAPTER 10. CONCLUSION

The goal of this work was to formalize methods for the extraction of sustained amplitudes from vibration data. In doing so, the following research questions were addressed:

1. How can maximum sustained amplitudes in vibration data be detected?

2. What is the effect of sampling rate and signal-to-noise ratio on methods available for determining maximum sustained amplitude?

3. How does the performance of methods available for determining maximum sustained amplitude change for non-stationary signals?

4. How feasible are the methods available for determining maximum sustained amplitude for implementation on an edge device?

In answering the first research question, six methods were formalized in two categories: envelope and non-envelope. The envelope methods included the Moving RMS, STFT Peaks, Peak Envelope, and LP Rectifier Envelope. The non-envelope methods included the Peak Distances and Expire Timer. The remaining research questions dealt with analyzing the performance of these methods under differing circumstances.

To address the second research question, a simulated signal was used so that the sampling rate and noise level could be changed. This simulated signal was based on a base tooth pass frequency and two harmonics of it. Sampling rate was found to have the greatest effect on the Expire Timer and Peak Envelope. When evaluating the effect of noise, the Peak Distances and STFT Peaks methods were found to have the greatest sensitivity.
To answer the third research question, disturbances were added to this simulated signal. After disturbances were added to the signals, the *LP Rectifier Envelope* and *Peak Distances* were found to have the largest range in performance at low sampling rates, but the smallest range in errors at higher sampling rates. With the sampling rate constant and varying SNR, the *LP Rectifier Envelope* and *Peak Envelope* methods showed the greatest range in errors at low SNR.

To answer the fourth research question, the time to execute each method was recorded. This was conducted at two sample levels to see how computation time scales for larger datasets. The test revealed that all methods except for the *LP Rectifier Envelope* are feasible for use on streaming data. Additionally, the *Expire Timer* and *LP Rectifier Envelope* time scaled at the same rate as the increase in the size of the data, while the others scale at a lower rate. For this reason, methods other than the *Expire Timer* and *LP Rectifier Envelope* should be considered for applications where a large amount of historical data is being processed and time is important.

Together, the answers to these research questions inform users on how to process vibration signals to determine the sustained amplitudes. These signal processing methods can be used on edge devices to monitor equipment that is sensitive to vibration. This is valuable in a manufacturing application because optical encoders, which are sensitive to vibrations, are used to determine the location of CNC and coordinate measuring machines. Monitoring their vibration levels allows operators to be informed about the vibration levels and realize when error may be introduced.
APPENDIX A. CODE

This appendix contains Python code used to implement the methods for the detection of sustained maximum amplitudes in vibration signals that are explained in CHAPTER 3 and tested throughout the work.

A.1 Peak Height at Width

```
1. import numpy as np
2. from scipy.signal import peak_widths
3.
4. def peak_height_at_width(s,w):
5.     ....
6.     Parameters
7.     -------
8.     s: input signal
9.     w: desired width
10.
11.     Returns
12.     -------
13.     h_eval:    height where desired width is met
14.     left_ips:  left position
15.     right_ips: right position
16.     '....
17.     peaks, _ = find_peaks(s, width = w)
18.
19.     if len(peaks) > 0:
20.         # loop through peak_widths function to find rel_height where the peakWidth is reached
21.         rel_height = 0.5
22.         last_high = 1
23.         last_low = 0
24.         widths = [0]      # initialize so that while loop can be used
25.         count = 0
26.         while np.abs(w - widths[0]) > (w * 0.025):
27.             count += 1
28.             widths, h_eval, left_ips, right_ips = peak_widths(s, peaks, rel_height=rel_height)
29.             ind = np.argmax(h_eval)
30.             if widths[0] > w:
31.                 last_high = rel_height
32.                 rel_height = (rel_height + last_low) * 0.5
33.             elif widths[0] < w:
34.                 last_low = rel_height
35.                 rel_height = (rel_height + last_high) * 0.5
36.             else:
37.                 break
38.             if count > 20:
39.                 break
40.             h_eval = h_eval[ind]
41.             left_ips = left_ips[ind]
```
right_ips = right_ips[ind]

else:
    h_eval = np.average(s)
    left_ips = 0
    right_ips = len(s)-1

return h_eval, left_ips, right_ips

A.2 Moving RMS

import numpy as np
def moving_RMS(s,t,fs,lims,fmin=400):
    
    Parameters
    ----------
    s: signal
    t: corresponding time series
    fs: sampling rate
    lims: sustained limits
    fmin: minimum accurate frequency [Hz]
    
    Returns
    -------
    peak: sustained peak
    start: beginning of peak
    end: end of peak (or end of signal)
    s_rms: envelope rms signal
    t_rms: corresponding time series
    
    # set window size
    windowSize = 2 / fmin             # s
    windowSize = int(windowSize*fs)   # samples

    # perform rms
    rms = s**2
    rms = np.cumsum(rms, dtype=float)
    rms[windowSize:] = rms[windowSize:] - rms[:-windowSize]
    rms_env = rms[windowSize - 1:] / windowSize
    rms_env = rms_env**(1/2)

    # move rms to the level of the peaks of a sine function
    rms_env = rms_env * np.sqrt(2)
    windowSize = 2 / fmin
    rms_t = t[windowSize-1:]
    peakWidth = int(lims['sustainedTime']*fs)
    peak, left_ips, right_ips = peak_height_at_width(rms_env, peakWidth)

    start = t[int(left_ips+windowSize-1)]
    end = t[int(right_ips+windowSize-1)]

    return peak, start, end, rms_env, rms_t

A.3 STFT Peaks

import numpy as np
import numpy as np

from scipy.signal import find_peaks
from scipy.interpolate import UnivariateSpline


def peak_envelope(s, t, fs, lims, d=4):
    ...

    Parameters
    ----------
    s:    signal
    t:    corresponding time series
    fs:   sampling rate
    lims: sustainedLimits
    d:    window size [samples]

    Returns
    -------
    peak: sustained peak
    start: beginning of peak
    end: end of peak (or end of signal)
    stft_env: enveloped signal
    stft_t: corresponding time series
    ...

    >>> f, stft_t, Zxx = stft(s, fs, nperseg=w, noverlap=w/2)
    23. Zxx = 2 * np.abs(Zxx)  # account for energy in negative

    for i in range(len(stft_t)):
        buff = np.concatenate(([0],Zxx[:,i]))
        peaks, _ = find_peaks(buff)  # find peaks
        if len(peaks) != 0:
            idx = np.argsort(buff[peaks])
            j = 1
            complete = False
            while not complete:
                if buff[peaks[idx[-j]]] < buff[peaks[0]]*0.1:
                    complete = True
                else:
                    j += 1
                    if j > len(idx):
                        complete = True
            peaks = peaks[idx[-(j-1):]]
            stft_env[i] = np.sum(buff[peaks])
        # find maximum sustained amplitude from the envelope
        peakWidth = int(lims['sustainedTime']*fs / (w/2))
        peak, left_ips, right_ips = peak_height_at_width(stft_env, peakWidth)
        start = t[int(left_ips * (w/2))]
        end = t[int(right_ips * (w/2))]

    return peak, start, end, stft_env, stft_t

A.4 Peak Envelope
A.5 LP Rectifier Envelope

```python
import numpy as np
from scipy.signal import butter, filtfilt

def LP_rectifier_envelope(s, t, fs, lims, a=20, b=50, c=1):
    '''
    Parameters
    ----------
    s: signal
    t: corresponding time series
    fs: sampling rate
    lims: sustainedLimits
    a: LPF1
    b: LPF2
    c: iterations

    Returns
    -------
    peak: sustained peak
    start: beginning of peak
    end: end of peak (or end of signal)
    peaks_env: enveloped signal
    peaks_t: corresponding time series
    '''
    def low_pass(data, cutoff, fs, order=5):
        nyq = 0.5 * fs
        normal_cutoff = cutoff / nyq
```
numerator_coeffs, denominator_coeffs = butter(order, normal_cutoff)

filtered_signal = filtfilt(numerator_coeffs, denominator_coeffs, data)
return filtered_signal

envelope = np.absolute(s)
LPR_env = low_pass(envelope, a, fs)
for i in range(c):
envelope2 = envelope - LPR_env
envelope2 = np.multiply((envelope2 > 0), envelope2)
LPR_env = LPR_env + low_pass(envelope2, b, fs)
if sum(LPR_env < envelope) <= 20:
    # 20 is 1% of samples
    break
LPR_t = t

# find maximum sustained amplitude from the envelope
peakWidth = int(lims['sustainedTime']*fs)
peak, left_ips, right_ips = peak_height_at_width(LPR_env, peakWidth)
start = t[int(left_ips)]
end = t[int(right_ips)]
return peak, start, end, LPR_env, LPR_t

A.6 Peak Distances

import numpy as np
from scipy.signal import find_peaks
def peak_distances(s, t, lims):
    #
    Parameters
    ----------
    s:     signal
t:     corresponding times
lims:  sustainedLimits

    Returns
    -------
    peak:  sustained peak
    start: beginning of peak
    end:   end of peak (or end of signal)
    
    # find peaks then sort in descending order
    peaks_signal, _ = find_peaks(np.absolute(s))
    peaks_signal_t = t[peaks_signal]
    peaks_signal_a = np.absolute(s)[peaks_signal]
    peaks_signal = np.transpose(np.stack((peaks_signal_t,peaks_signal_a)))
    peaks_signal = peaks_signal[np.argsort(peaks_signal[:,1])[::-1]]

    # iterate through peaks to find maximum sustained amplitude
    complete = False
    n = 1
    while not complete:
        distBuffer = np.zeros((1,n))
        for i in range(n):
            distBuffer[0,i] = np.absolute(peaks_signal[n,0] - peaks_signal[i,0])
        # convert to boolean
        distBufBool = (distBuffer > lims['sustainedTime'])
        if np.sum(distBufBool) > 0:
            complete = True
end = np.argmax(distBufBool)

else:
    n += 1

# save output values
peak = peaks_signal[n,1]
start = peaks_signal[n,0]
end = peaks_signal[end,0]

return peak, start, end

A.7 Expire Timer

import numpy as np

def expire_timer(s, t, fs, lims, fmin=250, replicates=5):
    '''
    Parameters
    ----------
    s:     signal
    t:     corresponding time series
    fs:    sampling rate [Hz]
    lims:  sustainedLimits
    
    Returns
    -------
    peak:  sustained peak
    start: beginning of peak
    end:   end of peak (or end of signal)
    '''
    expireTime = 1 / (2 * fmin)

    attempts = np.zeros(replicates)
    attempts[0] = np.max(np.absolute(s))  # first attempt is the max
    exp = []

    lastTooHigh = attempts[0]
    lastSustained = 0

    # try replicates iterations to close on max
    for j in range(replicates):
        sustained = False
        sustainedFaultStarted = False
        startTime = 0
        expBuf = []
        num = -1

        # check for sustained offences account for positive and negative values
        check = (s >= attempts[j]) | (s <= -attempts[j])

        for i in range(len(check)):
            sampleTime = i / fs

            if check[i]:
                # expire timer is reset every time a reading is above the limit
                expireStart = sampleTime
                if startTime == 0:
                    startTime = sampleTime
                posStart = i  # for graphing save possible start value
if not sustainedFaultStarted and (sampleTime - startTime) >= lims['sustainedTime']:
    sustainedFaultStarted = True
    num = num + 1
    expBuf.append((attempts[j], posStart, i, len(s)-1])
    sustained = True
elif sustainedFaultStarted and (sampleTime - expireStart) > expireTime:
    # sustained was started but expired
    expBuf[num][3] = i  # for graphing save end
    sustainedFaultStarted = False
    startTime = 0
elif startTime != 0 and (sampleTime - expireStart) > expireTime:
    # no alarm occurred reset start time
    startTime = 0

# set the next attempt
if sustained or sustainedFaultStarted:
    lastSustained = attempts[j]
    exp = expBuf
else:
    lastTooHigh = attempts[j]

if j < (replicates-1):
    attempts[j+1] = (lastTooHigh + lastSustained) / 2
if len(exp) == 0:
    exp.append((0, 0, 0, 0))

# save output values
peak = exp[0][0]
start = t[exp[0][1]]
end = t[exp[0][2]]

return peak, start, end
APPENDIX B. DESIGN OF EXPERIMENT

The table below details all the test conditions used in this work.

<table>
<thead>
<tr>
<th>Sampling Rate [Hz]</th>
<th>SNR</th>
<th>Disturbance (if present)</th>
<th>Power (n)</th>
<th>Amplitude (a)</th>
<th>Standard deviation (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,000</td>
<td>Inf.</td>
<td>No disturbance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inf.</td>
<td>No disturbance</td>
<td>1</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td>10,000</td>
<td>100</td>
<td>No disturbance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0.5</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>0.5</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>10,000</td>
<td>10</td>
<td>No disturbance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0.5</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>0.5</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>10,000</td>
<td>5</td>
<td>No disturbance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0.5</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>0.5</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>10,000</td>
<td>Inf.</td>
<td>No disturbance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0.5</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>0.5</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>Sampling Rate [Hz]</td>
<td>SNR</td>
<td>Disturbance (if present)</td>
<td>Power (n)</td>
<td>Amplitude (a)</td>
<td>Standard deviation (σ)</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----</td>
<td>--------------------------</td>
<td>-----------</td>
<td>--------------</td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>No disturbance</td>
<td>1</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td>7,500</td>
<td>10</td>
<td>No disturbance</td>
<td>1</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>No disturbance</td>
<td>1</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td>Inf.</td>
<td>100</td>
<td>No disturbance</td>
<td>1</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td>5,000</td>
<td>10</td>
<td>No disturbance</td>
<td>1</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>No disturbance</td>
<td>1</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>0.5</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.01</td>
</tr>
</tbody>
</table>
### APPENDIX C. ALL RESULTS

Table 18 – All results for test case a with no disturbance.

<table>
<thead>
<tr>
<th>Sampling Rate [Hz]</th>
<th>SNR</th>
<th>Moving RMS</th>
<th>STFT Peaks</th>
<th>Peak Envelope</th>
<th>I.P Rectifier Envelope</th>
<th>Peak Distances</th>
<th>Expire Timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>Inf.</td>
<td>-42.1%</td>
<td>-9.2%</td>
<td>-1.1%</td>
<td>0.0%</td>
<td>-1.9%</td>
<td>-1.9%</td>
</tr>
<tr>
<td>10000</td>
<td>Inf.</td>
<td>-42.1%</td>
<td>-4.0%</td>
<td>-2.0%</td>
<td>0.0%</td>
<td>-2.5%</td>
<td>-2.5%</td>
</tr>
<tr>
<td>7500</td>
<td>Inf.</td>
<td>-42.1%</td>
<td>-5.2%</td>
<td>-5.8%</td>
<td>0.0%</td>
<td>-2.1%</td>
<td>-2.1%</td>
</tr>
<tr>
<td>5000</td>
<td>Inf.</td>
<td>-41.8%</td>
<td>-4.1%</td>
<td>-5.9%</td>
<td>0.0%</td>
<td>-5.4%</td>
<td>-5.4%</td>
</tr>
<tr>
<td>1000</td>
<td>Inf.</td>
<td>-41.6%</td>
<td>-6.6%</td>
<td>-2.9%</td>
<td>0.0%</td>
<td>-6.4%</td>
<td>-6.4%</td>
</tr>
<tr>
<td>1000</td>
<td>Inf.</td>
<td>-41.6%</td>
<td>-4.4%</td>
<td>-4.6%</td>
<td>0.0%</td>
<td>-4.6%</td>
<td>-4.6%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0%</td>
<td>1.9%</td>
<td>5.5%</td>
<td>5.5%</td>
<td>5.5%</td>
<td>5.5%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0%</td>
<td>4.1%</td>
<td>4.6%</td>
<td>4.6%</td>
<td>4.6%</td>
<td>4.6%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0.0%</td>
<td>22.9%</td>
<td>25.4%</td>
<td>25.4%</td>
<td>25.4%</td>
<td>25.4%</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>0.0%</td>
<td>31.5%</td>
<td>36.5%</td>
<td>36.5%</td>
<td>36.5%</td>
<td>36.5%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0%</td>
<td>14.5%</td>
<td>25.4%</td>
<td>25.4%</td>
<td>25.4%</td>
<td>25.4%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.0%</td>
<td>61.4%</td>
<td>39.3%</td>
<td>39.3%</td>
<td>39.3%</td>
<td>39.3%</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.0%</td>
<td>67.8%</td>
<td>23.4%</td>
<td>23.4%</td>
<td>23.4%</td>
<td>23.4%</td>
</tr>
</tbody>
</table>

Method

<table>
<thead>
<tr>
<th>Sampling Rate [Hz]</th>
<th>SNR</th>
<th>Moving RMS</th>
<th>STFT Peaks</th>
<th>Peak Envelope</th>
<th>I.P Rectifier Envelope</th>
<th>Peak Distances</th>
<th>Expire Timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>Inf.</td>
<td>-42.1%</td>
<td>-9.2%</td>
<td>-1.1%</td>
<td>0.0%</td>
<td>-1.9%</td>
<td>-1.9%</td>
</tr>
<tr>
<td>10000</td>
<td>Inf.</td>
<td>-42.1%</td>
<td>-4.0%</td>
<td>-2.0%</td>
<td>0.0%</td>
<td>-2.5%</td>
<td>-2.5%</td>
</tr>
<tr>
<td>7500</td>
<td>Inf.</td>
<td>-42.1%</td>
<td>-5.2%</td>
<td>-5.8%</td>
<td>0.0%</td>
<td>-2.1%</td>
<td>-2.1%</td>
</tr>
<tr>
<td>5000</td>
<td>Inf.</td>
<td>-41.8%</td>
<td>-6.6%</td>
<td>-2.9%</td>
<td>0.0%</td>
<td>-5.4%</td>
<td>-5.4%</td>
</tr>
<tr>
<td>1000</td>
<td>Inf.</td>
<td>-41.6%</td>
<td>-4.4%</td>
<td>-4.6%</td>
<td>0.0%</td>
<td>-6.4%</td>
<td>-6.4%</td>
</tr>
<tr>
<td>1000</td>
<td>Inf.</td>
<td>-41.6%</td>
<td>-4.4%</td>
<td>-4.6%</td>
<td>0.0%</td>
<td>-4.6%</td>
<td>-4.6%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0%</td>
<td>1.9%</td>
<td>5.5%</td>
<td>5.5%</td>
<td>5.5%</td>
<td>5.5%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0%</td>
<td>4.1%</td>
<td>4.6%</td>
<td>4.6%</td>
<td>4.6%</td>
<td>4.6%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0.0%</td>
<td>22.9%</td>
<td>25.4%</td>
<td>25.4%</td>
<td>25.4%</td>
<td>25.4%</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>0.0%</td>
<td>31.5%</td>
<td>36.5%</td>
<td>36.5%</td>
<td>36.5%</td>
<td>36.5%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.0%</td>
<td>14.5%</td>
<td>25.4%</td>
<td>25.4%</td>
<td>25.4%</td>
<td>25.4%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.0%</td>
<td>61.4%</td>
<td>39.3%</td>
<td>39.3%</td>
<td>39.3%</td>
<td>39.3%</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.0%</td>
<td>67.8%</td>
<td>23.4%</td>
<td>23.4%</td>
<td>23.4%</td>
<td>23.4%</td>
</tr>
</tbody>
</table>

Method
Table 19 – All results for test case b: a gaussian disturbance with an amplitude of 0.5 and standard deviation of 0.005.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling</th>
<th>100000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>Inf.</td>
<td>Inf.</td>
<td>Inf.</td>
<td>Inf.</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-41.8%</td>
<td>-41.8%</td>
<td>-41.8%</td>
<td>-41.8%</td>
<td>-41.3%</td>
<td>-41.5%</td>
<td>-41.6%</td>
<td></td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>-4.8%</td>
<td>6.9%</td>
<td>10.6%</td>
<td>7.3%</td>
<td>7.4%</td>
<td>11.6%</td>
<td>8.5%</td>
<td></td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>-0.8%</td>
<td>-3.1%</td>
<td>-3.8%</td>
<td>-7.6%</td>
<td>1.3%</td>
<td>-3.4%</td>
<td>-6.0%</td>
<td></td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>-6.7%</td>
<td>-6.8%</td>
<td>-6.7%</td>
<td>-5.4%</td>
<td>-4.0%</td>
<td>-4.7%</td>
<td>-5.5%</td>
<td></td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.0%</td>
<td>-0.5%</td>
<td>-0.6%</td>
<td>-1.1%</td>
<td>5.5%</td>
<td>6.0%</td>
<td>5.3%</td>
<td></td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-0.1%</td>
<td>-0.7%</td>
<td>-0.7%</td>
<td>-7.6%</td>
<td>5.4%</td>
<td>-4.7%</td>
<td>-6.0%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-38.4%</td>
<td>-38.9%</td>
<td>-39.3%</td>
<td>-35.6%</td>
<td>-36.2%</td>
<td>-36.7%</td>
<td></td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>37.6%</td>
<td>45.8%</td>
<td>38.4%</td>
<td>59.6%</td>
<td>84.3%</td>
<td>29.3%</td>
<td></td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>13.2%</td>
<td>1.0%</td>
<td>20.6%</td>
<td>21.5%</td>
<td>11.0%</td>
<td>22.9%</td>
<td></td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>8.2%</td>
<td>0.5%</td>
<td>0.9%</td>
<td>17.5%</td>
<td>7.5%</td>
<td>5.3%</td>
<td></td>
</tr>
<tr>
<td>Peak Distances</td>
<td>24.7%</td>
<td>22.7%</td>
<td>18.9%</td>
<td>37.9%</td>
<td>35.4%</td>
<td>27.2%</td>
<td></td>
</tr>
<tr>
<td>Expire Timer</td>
<td>18.2%</td>
<td>5.3%</td>
<td>7.7%</td>
<td>24.4%</td>
<td>10.7%</td>
<td>16.4%</td>
<td></td>
</tr>
</tbody>
</table>
Table 20 – All results for test case c: a gaussian disturbance with an amplitude of 0.5 and standard deviation of 0.01.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling 100000</th>
<th>Sampling 10000</th>
<th>Sampling 7500</th>
<th>Sampling 5000</th>
<th>Sampling 10000</th>
<th>Sampling 7500</th>
<th>Sampling 5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>Inf.</td>
<td>Inf.</td>
<td>Inf.</td>
<td>Inf.</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-49.8%</td>
<td>-49.8%</td>
<td>-49.8%</td>
<td>-49.8%</td>
<td>-49.4%</td>
<td>-49.6%</td>
<td>-49.6%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>4.6%</td>
<td>5.7%</td>
<td>5.9%</td>
<td>5.8%</td>
<td>5.8%</td>
<td>6.9%</td>
<td>7.0%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>8.2%</td>
<td>7.5%</td>
<td>6.4%</td>
<td>0.1%</td>
<td>6.6%</td>
<td>-18.0%</td>
<td>8.0%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>3.1%</td>
<td>3.2%</td>
<td>3.4%</td>
<td>2.8%</td>
<td>4.4%</td>
<td>3.9%</td>
<td>6.6%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.0%</td>
<td>-0.5%</td>
<td>-0.5%</td>
<td>-10.3%</td>
<td>5.5%</td>
<td>-4.6%</td>
<td>-6.5%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-0.1%</td>
<td>-0.6%</td>
<td>-0.6%</td>
<td>-10.5%</td>
<td>5.4%</td>
<td>-4.8%</td>
<td>-6.6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling 10000</th>
<th>Sampling 7500</th>
<th>Sampling 5000</th>
<th>Sampling 10000</th>
<th>Sampling 7500</th>
<th>Sampling 5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-46.9%</td>
<td>-37.4%</td>
<td>-47.6%</td>
<td>-44.5%</td>
<td>-33.9%</td>
<td>-45.4%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>20.0%</td>
<td>33.5%</td>
<td>33.1%</td>
<td>57.3%</td>
<td>76.0%</td>
<td>17.3%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>14.5%</td>
<td>-14.2%</td>
<td>5.4%</td>
<td>14.7%</td>
<td>9.5%</td>
<td>-11.8%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>11.8%</td>
<td>3.1%</td>
<td>17.7%</td>
<td>17.1%</td>
<td>20.1%</td>
<td>27.7%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>12.1%</td>
<td>8.5%</td>
<td>6.6%</td>
<td>17.1%</td>
<td>22.9%</td>
<td>15.3%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>11.9%</td>
<td>8.0%</td>
<td>6.4%</td>
<td>16.7%</td>
<td>14.6%</td>
<td>15.1%</td>
</tr>
</tbody>
</table>
Table 21 – All results for test case d: a higher-order gaussian disturbance with an exponent of 3, amplitude of 0.5 and standard deviation of 0.005.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling</th>
<th>100000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td></td>
<td>Inf.</td>
<td>Inf.</td>
<td>Inf.</td>
<td>Inf.</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-41.1%</td>
<td>-41.2%</td>
<td>-41.2%</td>
<td>-41.2%</td>
<td>-40.7%</td>
<td>-40.8%</td>
<td>-41.0%</td>
<td></td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>-3.8%</td>
<td>8.0%</td>
<td>7.6%</td>
<td>7.1%</td>
<td>8.5%</td>
<td>13.4%</td>
<td>8.9%</td>
<td></td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>0.6%</td>
<td>-2.0%</td>
<td>-2.7%</td>
<td>-6.6%</td>
<td>2.4%</td>
<td>-2.4%</td>
<td>-3.6%</td>
<td></td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>-5.7%</td>
<td>-5.8%</td>
<td>-5.7%</td>
<td>-5.9%</td>
<td>-2.9%</td>
<td>-3.6%</td>
<td>-4.5%</td>
<td></td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>6.7%</td>
<td>7.1%</td>
<td>6.4%</td>
<td></td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-0.1%</td>
<td>-0.4%</td>
<td>-0.7%</td>
<td>-6.5%</td>
<td>5.5%</td>
<td>-4.5%</td>
<td>-5.2%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Sampling</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-37.7%</td>
<td>-38.2%</td>
<td>-38.6%</td>
<td>-34.8%</td>
<td>-35.5%</td>
<td>-36.0%</td>
<td></td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>28.1%</td>
<td>47.2%</td>
<td>35.6%</td>
<td>61.2%</td>
<td>84.1%</td>
<td>35.1%</td>
<td></td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>14.4%</td>
<td>2.1%</td>
<td>-0.1%</td>
<td>22.8%</td>
<td>12.2%</td>
<td>3.6%</td>
<td></td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>9.4%</td>
<td>1.5%</td>
<td>2.0%</td>
<td>18.8%</td>
<td>8.6%</td>
<td>6.4%</td>
<td></td>
</tr>
<tr>
<td>Peak Distances</td>
<td>26.0%</td>
<td>24.0%</td>
<td>20.2%</td>
<td>39.4%</td>
<td>36.9%</td>
<td>28.6%</td>
<td></td>
</tr>
<tr>
<td>Expire Timer</td>
<td>18.1%</td>
<td>2.8%</td>
<td>8.2%</td>
<td>25.9%</td>
<td>8.0%</td>
<td>16.8%</td>
<td></td>
</tr>
</tbody>
</table>
Table 22 – All results for test case e: a higher-order gaussian disturbance with an exponent of 3, amplitude of 0.5 and standard deviation of 0.01.

<table>
<thead>
<tr>
<th>Method</th>
<th>100000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling</td>
<td>Inf.</td>
<td>Inf.</td>
<td>Inf.</td>
<td>Inf.</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-40.2%</td>
<td>-40.1%</td>
<td>-40.5%</td>
<td>-40.8%</td>
<td>-39.8%</td>
<td>-40.1%</td>
<td>-40.9%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>10.0%</td>
<td>10.3%</td>
<td>8.5%</td>
<td>10.4%</td>
<td>9.4%</td>
<td>9.5%</td>
<td>11.2%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>11.6%</td>
<td>10.7%</td>
<td>11.2%</td>
<td>7.6%</td>
<td>10.2%</td>
<td>12.6%</td>
<td>15.4%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>9.5%</td>
<td>9.6%</td>
<td>9.4%</td>
<td>9.4%</td>
<td>10.6%</td>
<td>10.2%</td>
<td>12.7%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>0.0%</td>
<td>-0.5%</td>
<td>-0.5%</td>
<td>-4.6%</td>
<td>5.4%</td>
<td>-2.3%</td>
<td>-2.8%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>-0.3%</td>
<td>-0.7%</td>
<td>-0.5%</td>
<td>-4.6%</td>
<td>5.2%</td>
<td>-2.5%</td>
<td>-2.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
<th>10000</th>
<th>7500</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Moving RMS</td>
<td>-47.5%</td>
<td>-36.6%</td>
<td>-48.1%</td>
<td>-45.0%</td>
<td>-33.0%</td>
<td>-46.0%</td>
</tr>
<tr>
<td>STFT Peaks</td>
<td>28.0%</td>
<td>37.3%</td>
<td>39.8%</td>
<td>66.5%</td>
<td>71.3%</td>
<td>23.7%</td>
</tr>
<tr>
<td>Peak Envelope</td>
<td>19.3%</td>
<td>-14.5%</td>
<td>-15.4%</td>
<td>20.6%</td>
<td>23.4%</td>
<td>-12.6%</td>
</tr>
<tr>
<td>LP Rectifier Envelope</td>
<td>17.5%</td>
<td>3.1%</td>
<td>24.5%</td>
<td>24.0%</td>
<td>27.4%</td>
<td>35.5%</td>
</tr>
<tr>
<td>Peak Distances</td>
<td>18.3%</td>
<td>14.7%</td>
<td>6.5%</td>
<td>26.1%</td>
<td>25.0%</td>
<td>14.1%</td>
</tr>
<tr>
<td>Expire Timer</td>
<td>18.1%</td>
<td>14.5%</td>
<td>6.3%</td>
<td>22.9%</td>
<td>24.2%</td>
<td>13.9%</td>
</tr>
</tbody>
</table>
APPENDIX D. RMS OF MULTIPLE SINE WAVES

To understand why the Moving RMS method does a poor job forming an envelope around the signal comprised of harmonics, the relationship of the RMS value to the peak of a signal with two frequencies is derived in Equations 25 to 30. This shows that the RMS value is half of the peak value.

\[ x(t) = \frac{a}{2} \cdot (\sin(t) + \sin(2t)) \quad (25) \]

\[ x_{rms} = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} \left(\frac{a}{2} \cdot (\sin(t) + \sin(2t))^2 \right) dt} \quad (26) \]

\[ x_{rms} = \sqrt{\frac{a^2}{8\pi} \int_0^{2\pi} \left(\sin^2(t) + 2 \cdot \sin(t) \cdot \sin(2t) + \sin^2(2t)\right) dt} \quad (27) \]

\[ x_{rms} = \sqrt{\frac{a^2}{8\pi} \cdot \left(\frac{t}{2} - \frac{1}{4} \sin(2t) + \frac{4}{3} \cdot \sin^3(t) + \frac{t}{2} - \frac{1}{8} \cdot \sin(4t)\right)} \bigg|_0^{2\pi} \quad (28) \]

\[ x_{rms} = a \cdot \sqrt{\frac{1}{8\pi} \cdot 2\pi} = a \cdot \sqrt{\frac{1}{4}} \quad (29) \]

\[ a = x_{rms} \cdot 2 \quad (30) \]
This analysis is taken further by considering the relationship of the RMS value to the peak of a signal with three frequencies. This relation is derived in Equations 31 to 36. This shows that the RMS value is equal to the peak value divided by the square root of six.

\[ x(t) = \frac{a}{3} \times (\sin(t) + \sin(2t) + \sin(3t)) \tag{31} \]

\[ x_{rms} = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} \left( \frac{a}{3} \right)^2 \times (\sin(t) + \sin(2t) + \sin(3t))^2 \, dt} \tag{32} \]

\[ x_{rms} = \sqrt{\frac{a^2}{18\pi} \int_0^{2\pi} \left( \frac{\sin^2(t) + \sin^2(2t) + \sin^2(3t) + 2 \times \sin(t) \times \sin(2t) + 2 \times \sin(t) \times \sin(3t) + 2 \times \sin(2t) \times \sin(3t)}{2\pi} \right) \, dt} \tag{33} \]

\[ x_{rms} = \sqrt{\frac{a^2}{18\pi} \int_0^{2\pi} \left( t - \frac{1}{4} \sin(2t) + \frac{4}{3} \sin^3(t) + \frac{1}{2} \sin(2t) \right)^2 \, dt} \tag{34} \]

\[ x_{rms} = a \times \frac{1}{\sqrt{18\pi} \times \left( \frac{1}{2} \times 2\pi \times 3 \right)} = \frac{a}{\sqrt{6}} \tag{35} \]

\[ a = x_{rms} \times \sqrt{6} \tag{36} \]
REFERENCES


